



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE GRADO

**QuPI-GAm: Aplicación de mecánicas de
gamificación para la integración de preguntas en
un trivial multijugador masivo**

Laura Gutiérrez López de la Franca

Febrero, 2017

QUPI-GAM: APLICACIÓN DE MECÁNICAS DE GAMIFICACIÓN PARA LA INTEGRACIÓN DE PREGUNTAS EN UN TRIVIAL MULTIJUGADOR MASIVO



UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INFORMÁTICA

Tecnologías y Sistemas de Información

**TECNOLOGÍA ESPECÍFICA DE
TECNOLOGÍAS DE LA INFORMACIÓN**

TRABAJO FIN DE GRADO

**QuPI-GAm: Aplicación de mecánicas de
gamificación para la integración de preguntas en
un trivial multijugador masivo**

Autor: Laura Gutiérrez López de la Franca

Director: David Vallejo Fernández

Director: Carlos González Morcillo

Febrero, 2017

Laura Gutiérrez López de la Franca

Ciudad Real – Spain

E-mail: Laura.Gutierrez8@alu.uclm.es

© 2017 Laura Gutiérrez López de la Franca

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Se permite la copia, distribución y/o modificación de este documento bajo los términos de la Licencia de Documentación Libre GNU, versión 1.3 o cualquier versión posterior publicada por la *Free Software Foundation*; sin secciones invariantes. Una copia de esta licencia esta incluida en el apéndice titulado «GNU Free Documentation License».

Muchos de los nombres usados por las compañías para diferenciar sus productos y servicios son reclamados como marcas registradas. Allí donde estos nombres aparezcan en este documento, y cuando el autor haya sido informado de esas marcas registradas, los nombres estarán escritos en mayúsculas o como nombres propios.

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

Resumen

En los últimos años la irrupción de Internet y la familiarización que se ha producido respecto a las nuevas tecnologías han propiciado la aparición de la Sociedad 2.0, entendida como el reflejo *online* de la sociedad real. Cada vez más gente aprovecha el medio virtual para mostrarse (ya que esta es una de las vías más efectivas para ganar visibilidad, notoriedad y prestigio a día de hoy) así como para poder expresarse o bien participar en diferentes acciones de carácter colectivo que puedan tener un cierto impacto. Este último aspecto puede ser aprovechado ya que, si se consigue llamar su atención, se puede movilizar a un gran grupo de personas dispuestas a colaborar de distintas maneras, siendo una de ellas la aportación de conocimiento.

Existen diferentes contextos donde puede ser necesario reunir un cierto tipo de datos que den soporte a un fin específico y donde resultaría útil recurrir a la colaboración de personas para conseguir recopilar esa información. Centrando la atención en el ámbito de interés en que se enmarca este trabajo, todo este planteamiento puede ser aprovechado en el contexto del mercado de juegos para móviles, más concretamente el de los juegos tipo trivial basados en preguntas. Este tipo de juegos, que aun basándose en un concepto clásico siguen gozando de mucha popularidad, deben nutrirse de una amplia variedad de preguntas, las cuales podrían obtenerse solicitando la colaboración de los propios jugadores.

Lo que se plantea en este trabajo es el diseño desarrollo de un sistema potencialmente integrable en un juego con las características previamente mencionadas, que incorpore las funcionalidades necesarias para que los jugadores puedan sugerir sus propias preguntas. Para ello es necesario trabajar en dos líneas. Por un lado se emplean técnicas de gamificación con las que se busca incentivar y motivar a los jugadores a adoptar ese rol colaborativo que de ellos se espera. Por otro lado se presta atención al procedimiento de validación requerido para establecer un filtro que permita recuperar de todo el conjunto las preguntas que tengan la suficiente calidad como para pasar a formar parte del juego. Cabe destacar que el desarrollo se ha llevado a cabo en colaboración con la empresa Furious Koalas, que es dueña de *Whizz*, un trivial multijugador masivo para plataforma móvil.

Abstract

The irruption of the Internet and the familiarization with new technologies in these last years has provoked the birth of the Society 2.0 as an *online* reflect of real society. More and more people take advantage of the virtual resources in order to show themselves, attempting to acquire notoriety, as well as being able to express themselves and participate in collective activities. This last aspect can be used since, if we can obtain the user's attention, we can mobilize a crowd of people in the same direction for diverse purposes. One of these purposes is the generation of knowledge.

There are different types of contexts in where it could be necessary gather some kind of data that gives us support in order to achive some specific purposes. In this situation it could be useful to appeal to the collaboration of people in order to gather that data. Focusing on the scope of interest of this work, this approach can be used in the context of the mobile gaming market, specifically in the context of the mobile trivial games based on questions. These kinds of games are very popular even if they are based on a classic concept. These games need a huge variety of questions and these could be obtained with the collaboration of the players.

In this work it has been designed and developed a system with the necessary functionalities in order to give the players the possibility to suggest their own questions. This system has the potential to be integrated into a game with the previously mentioned characteristics. In order to achieve this goal it is necessary to work in two different development lines. On one side gamification techniques are used with the purpose of encouraging and motivating the collaborative role of the players. On the other side a validation procedure has been developed in order to filter the global set of player's questions and to obtain only those questions with enough quality to be part of the game. It should be noted that the development has been done in collaboration with Furious Koalas, a company who owns *Whizz*, a massive mobile multiplayer trivial.

Agradecimientos

Es difícil evaluar objetivamente y sin errar quien merece la gratitud de uno, así como también es difícil redactar unos agradecimientos sin caer en clichés. Esta no es sin embargo una sección que se pueda planificar, revisar o juzgar como las demás así que, más allá de cualquier valoración, la única opción posible es empezar a escribir, dejando constancia de todos aquellos pensamientos que caprichosamente vengan a mi memoria. Con el deseo de reflejar a todo aquel que lo merece de una forma impersonalmente personal, debo agradecer...

... A aquel que me recomendó por pena, una pena sin dudas muy útil y efectiva.

... A aquel que mediante un correo me abrió la puertas al universo de los koalas, que siempre me ha recibido con una sonrisa pese a todo y a quien admiro por sus cualidades como profesional y persona.

... A aquel amante de los perros que emite pitidos de vez en cuando, que me debe un pin y que más fielmente me ha acompañado durante los años de carrera.

... A aquel cerdito danés que me brindó su amistad con *muffins* y que pese a la distancia sigue apoyándome y escuchándome.

... A aquel *sparring* peculiar que tan fervientemente desea la navaja de MacGyver y que acude en mi ayuda cuando lo necesito aunque no tenga la furgoneta del Equipo A a mano.

... A aquella con la que he compartido noches blancas, que me permitió colonizar su espacio vital durante tanto tiempo y me prestó el bolso cuando lo necesité para mis escapadas.

... A aquel que hacía y sigue haciendo de cada día una travesía motorizada y si hace falta me trae una bamba de chocolate para animarme.

... A aquella que sufre más por mis desvelos que por los suyos propios y que se va modernizando a marchas forzadas para poder comunicarse conmigo.

... A aquel informático en potencia tan especial que consiguió estar junto a mi en la única noche nevada del año y que me regaló una caja mágica para afrontar el sprint final.

... A mi misma, por deleitarme con la sonata del silencio nocturno y seguir avanzando aun sin tener la certeza de que este sea el camino correcto.

Laura

*William Blake dijo: “Ningún pájaro llega demasiado alto si se eleva con sus propias alas”.
A todos aquellos que me han prestado sus alas cuando yo no podía alzar el vuelo.*

Índice general

Resumen	V
Abstract	VII
Agradecimientos	IX
Índice general	XIII
Índice de cuadros	XVII
Índice de figuras	XIX
Índice de listados	XXI
Listado de acrónimos	XXIII
1. Introducción	1
1.1. Potencialidad de la Web 2.0.	1
1.2. Necesidad de recopilar información	2
1.3. Incentivo a través de la gamificación	3
1.4. Ámbito de interés: Concursos de preguntas	4
1.5. Qué se pretende hacer	5
1.6. Contexto de desarrollo	7
1.6.1. Desarrollo basado en tecnologías web	7
1.6.2. Colaboración con empresa Furious Koalas S.L.	8
1.6.3. Estructura del documento	9
2. Objetivos	11
2.1. Objetivo general	11
2.2. Objetivos específicos	12

3. Estado del arte	15
3.1. El mercado de los juegos para móvil	15
3.1.1. El móvil como elemento de interés	15
3.1.2. Caracterización de los juegos para móvil	19
3.1.3. Juegos basados en preguntas	37
3.2. Gamificación	42
3.2.1. Concepto de gamificación	42
3.2.2. El porqué de la gamificación	43
3.2.3. Elemento de diseño	44
3.2.4. Evolución	47
3.2.5. Ámbito de investigación	53
3.2.6. Áreas de aplicación	56
3.2.7. Frameworks de diseño y desarrollo	63
3.3. Alternativas tecnológicas para el desarrollo de apps	65
3.3.1. Elección de SS.OO. de interés	66
3.3.2. Elección del tipo de aplicación	68
4. Método de trabajo	73
4.1. Elección de la metodología de desarrollo	73
4.1.1. Por qué escoger una metodología ágil	73
4.1.2. Aplicación de eXtreme Programming	74
4.2. Características hardware y software del desarrollo	77
5. Arquitectura QuPI-GAM	81
5.1. Visión general de la arquitectura	81
5.2. Autenticación del usuario	83
5.2.1. Identificación a través de Google + y Facebook	83
5.3. Soporte para la creación de usuarios y gestión de la sesión	87
5.3.1. Creación de usuarios	87
5.3.2. Procedimiento para la gestión del inicio y cierre de sesión	89
5.3.3. Autorización de acceso	89
5.4. Soporte a la internacionalización	91
5.4.1. Detección del idioma	91
5.4.2. Almacenamiento de las diferentes cadenas de texto	92
5.5. Aportación de preguntas	93
5.5.1. Constitución del formulario	93

5.5.2.	Almacenamiento de la pregunta	95
5.5.3.	Incorporación de reCaptcha	96
5.6.	Validación de preguntas	99
5.6.1.	Constitución del formulario	99
5.6.2.	Almacenamiento de la pregunta	100
5.7.	Detección de pregunta repetida	103
5.7.1.	Aproximación 1: Elaboración de un grafo de palabras	103
5.7.2.	Aproximación 2: Empleo del API de búsqueda de Google	111
5.8.	Perfil de colaborador	118
5.8.1.	Incorporación de elementos de juego	121
5.8.2.	Información del perfil	124
5.9.	Incorporación de trofeos	128
5.9.1.	Tipos de trofeos definidos	128
5.9.2.	Dinámica para la asignación de trofeos	131
5.10.	Elaboración de rankings mensuales	136
5.10.1.	Tipos de rankings	136
5.10.2.	Construcción de rankings	138
5.11.	Organización y componentes del proyecto	141
5.12.	Estructura de la base de datos	147
5.13.	Historias de Usuario	151
6.	Resultados	155
6.1.	Aspecto y funcionalidad de QuPI-GAm	155
6.2.	Distribución de trabajo	161
6.3.	Integración de código	165
7.	Conclusiones	169
7.1.	Objetivos alcanzados	169
7.2.	Dificultades encontradas	172
7.3.	Trabajo futuro	173
7.4.	Conclusiones finales	174
A.	Ficha de usuarios empleando técnica Personas	177
	Referencias	181

Índice de cuadros

3.1. Comparativa de principales juegos de preguntas	38
3.2. Mecánicas en función de las dinámicas que potencian	48
3.3. Características software del desarrollo nativo según plataforma	68
3.4. Frameworks para frontend y backend	70
3.5. Servicios en la nube para el soporte de Web Apps	71
4.1. Especificaciones del equipo empleado para el desarrollo	78
4.2. Medios software requeridos por la aplicación	78
4.3. Software de soporte al desarrollo	79
5.1. Qué se quiere despertar en el usuario (motivación intrínseca)	121
5.2. Proyección de mecánicas que podrían formar parte de QuPI-GAm	123
5.3. Rangos definidos	127
5.4. Combinaciones válidas de rankings en base a filtros	138
5.5. Vista de la organización del proyecto según jerarquía de directorios (Frontend)	142
5.6. Vista de la organización del proyecto según jerarquía de directorios (Backend)	143
5.7. Historia de Usuario - US_SUPPORT_01	151
5.8. Historia de Usuario - US_SUPPORT_02	151
5.9. Historia de Usuario - US_UPLOAD_01	152
5.10. Historia de Usuario - US_VALIDATION_01	152
5.11. Historia de Usuario - US_VALIDATION_02	153
5.12. Historia de Usuario - US_UPLOAD_02	153
5.13. Historia de Usuario - US_UPLOAD_02.1	153
5.14. Historia de Usuario - US_UPLOAD_02.2	154

Índice de figuras

1.1. Flujo de aporte y validación de preguntas	6
1.2. Logo Furious Koalas S.L.	8
1.3. Aspecto visual de Whizz	9
3.1. Datos ilustrativos del mercado de Apps (2014)	17
3.2. Compilación de soportes para videojuego representativos	22
3.3. Compilación de juegos para móvil y dispositivos móviles históricos	25
3.4. Top de juegos por ingresos diarios generados	31
3.5. Preguntados - Menú del modo Factoría de Preguntas	39
3.6. Preguntados - Aportación de pregunta nueva	40
3.7. Preguntados - Traducción de preguntas	40
3.8. Preguntados - Valoración de una pregunta	41
3.9. Preguntados - Historial de preguntas	41
3.10. Preguntados - Perfil de usuario y logros de colaboración	42
3.11. Gartner Hype Cycle para la gamificación (años 2013 y 2014)	51
3.12. Google Trends - Interés de búsqueda para “gamification” a lo largo del tiempo	52
3.13. Relación de empresas participantes en el Gamification Summit 2014	57
3.14. Code Academy - Perfil de Usuario	58
3.15. Starbucks - Perfil de consumidor en la app	59
3.16. Green Stamps - Aspecto del antiguo libro en papel	60
3.17. Trip Advisor - Perfil de colaborador	61
3.18. RespirApp - Aspecto de la aplicación	62
3.19. Unidades elementales del framework Octalysis	65
3.20. Cuota de mercado de sistemas operativos móviles en España	67
3.21. Comparativa de descargas e ingresos en Google Play y App Store	67
4.1. Panorámica de conceptos involucrados en XP	75
5.1. Representación visual de los principales componentes incluidos en QuPI-GAm	82

5.2. Comunicación establecida durante la autenticación	85
5.3. Diagrama de secuencia para llamadas principales durante la autenticación	86
5.4. Clases para constituir el soporte de base en el servidor	88
5.5. Diagrama de secuencia con el flujo de tratamiento para la sesión	90
5.6. Diagrama de secuencia para la carga del formulario de aportación y recepción de los datos introducidos	95
5.7. Ejemplo de recaptcha	97
5.8. Comunicación establecida para la integración de la recaptcha	98
5.9. Flujo que sigue la pregunta hasta que es almacenada definitivamente	101
5.10. Diagrama de secuencia para operación mediante el formulario de validación	103
5.11. Ejemplo para visualizar los componentes del grafo de palabras	104
5.12. Modelos necesarios para la constitución del grafo de palabras	105
5.13. Integración del proceso de detección de pregunta repetida para la aproximación basada en grafo	106
5.14. Integración del proceso de detección de pregunta repetida para la aproximación basada en API de búsqueda	113
5.15. Mecánicas y dinámicas de juego incorporadas en el diseño inicial	122
5.16. Puntos por nivel en el juego Pokemon GO	126
5.17. Puntos totales (absolutos) necesarios que dan acceso a cada nivel	127
5.18. Diagrama de actividad para la comprobación de trofeos conseguidos	135
5.19. Diagrama de componentes y despliegue	144
5.20. Diagrama de clases con las plantillas html manejadas	145
5.21. Diagrama de clases con las vistas manejadas en la parte cliente	146
5.22. Diagrama de clases con los modelos en la parte cliente	147
5.23. Conjunto de recursos expuestos en el servidor	148
5.24. Modelado de entidades definidas en el Datastore	150
6.1. Interfaz para la aportación de preguntas	156
6.2. Interfaz que muestra el aspecto del perfil de colaborador	157
6.3. Interfaz para la consulta de la posición en los rankings	158
6.4. Interfaz para la consulta de trofeos	159
6.5. Interfaz para la validación de preguntas	160
6.6. Interfaz para la autenticación e inicio de sesión	161

Índice de listados

5.1. Definición de los trofeos existentes en QuPI-GAm	129
5.2. Formato del listado de trofeos devuelto por el servidor	132

Listado de acrónimos

AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
BLOB	Binary Large Object
BSD	Berkeley Software Distribution
CAGR	Compound Annual Growth Rate
CSS	Cascading Style Sheets
EGDF	European Games Developer Federation
F2P	Free to play
GAE	Google App Engine
GPL	General Public License
HCI	Human Computer Interaction
HTML	HyperText Markup Language
IAP	In-app Purchases
IP	Intelectual Property
IoT	Internet of Things
IT	Information Technology
JSON	JavaScript Object Notation
MDA	Mechanics, Dynamics and Aesthetics
MMC	MultiMedia Card
MMO	Multiplayer Massive Online
MVC	Model-View-Controller pattern
OCR	Optical Character Recognition
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
PBL	Points, Badges and Leaderboards
REST	Representational State Transfer

ROI	Return of Investment
SDK	Software Development Kit
SQL	Structured Query Language
TDD	Test-Driven Development
UCD	User Centered Design
WAP	Wireless Application Protocol
XP	eXtremme Programming

Capítulo 1

Introducción

EN ciertos contextos puede ser necesario reunir un cierto tipo de datos que den soporte a un fin específico. Para ello, según la naturaleza y el tipo de datos que se busquen, puede resultar útil recurrir a la colaboración de personas que, adecuadamente motivadas, participarían por propia iniciativa. Con esta idea subyacente de base, se concreta a lo largo del capítulo cómo podría sacársele provecho en el contexto del mercado de juegos para móviles, más concretamente para juegos tipo trivial basados en preguntas.

1.1 Potencialidad de la Web 2.0.

El desarrollo, despliegue y evolución de Internet y la Web ha permitido consolidar un medio unificado e integral donde aglutinar todo lo relativo a la gestión de información en todas sus vertientes, así como contenidos y todo tipo de servicios asociados, siendo éstos cada vez más accesibles gracias a su rápida expansión (en torno a un 40 % de la población mundial en 2016 tiene conexión a Internet de acuerdo al portal *internet live stats*¹). Siendo relevante el volumen de personas que convergen hacia el uso casi en exclusividad de la Web como punto central de acceso a información, más significativa es la dualidad referente al tipo de motivación o uso aplicable, lo que entronca directamente con la denominación de Web 2.0, término oficialmente acuñado desde 2004 para denotar el cambio de perspectiva en cuanto al posicionamiento de los usuarios en la web, favorecido por los avances en la tecnología subyacente. Se pasó de una forma de comunicación *uno a muchos* con roles separados, donde el grueso de usuarios eran consumidores de contenido y un conjunto menor productores de un contenido estático (concepción de la web como escaparate), a un medio en el que cualquiera puede adoptar tanto un rol pasivo de consulta como activo, donde el **usuario aporta contenidos**.

El interés de los usuarios ha quedado patente a través de la exploración y uso masivo de las opciones que ofrece el nuevo paradigma, que través de diversos servicios y espacios de comunicación como foros, wikis, podcasts, blogs, redes sociales, microblogging, tiendas virtuales con sección de comentarios, etc. permiten interactuar, opinar, aconsejar, preguntar, compartir, colaborar y aportar.

¹<http://www.internetlivestats.com/internet-users/>

El impacto que ha supuesto la evolución hacia la Web 2.0 se ha visto reflejado en distintas áreas, como se analiza en [BVL⁺12]. Su evolución y aplicaciones futuras van de la mano de su combinación con otros paradigmas como el *Cloud Computing*, *Big Data* o *Internet of Things*, lo que se refleja en [NCWW16]. Sin embargo, de todos los aspectos derivables de esta evolución, el foco de interés en este trabajo radica en la potencialidad de la información que puede obtenerse a través de Internet y el precedente asentado por la Web 2.0. apelando al aspecto colaborativo por parte de los usuarios.

1.2 Necesidad de recopilar información

En ocasiones y bajo diferentes contextos puede requerirse la recopilación de información. Ahora bien, no referente al punto de vista de usuario, sino con el fin de dar soporte a fines específicos. Para ello Internet ha terminado constituyéndose como el medio primario a través del cual obtenerla, debido al fácil acceso y a causa de ese gran volumen de usuarios que hay detrás aportando y generando grandes cantidades de información, tanto consciente como inconscientemente. Desde esta aproximación general al problema de recopilar información pueden identificarse distintos ámbitos con distintas motivaciones, en cada uno de los cuales se requerirá un tipo de información diferente, lo que condiciona la forma de conseguirla.

Respecto a este último punto, puede concebirse la Web como un gran repositorio de documentos, aludiendo a la **información ya existente**, del cual descubrir por procedimientos automatizados la información de interés, que puede requerir procesos de extracción, preparación y análisis con implicación de técnicas como Web Scrapping o propias de Big Data y a través de Minería de Datos y Aprendizaje Automático entre otras disciplinas. Esta es una opción que, siendo potente, puede verse limitada por cuestiones semánticas, o bien porque el tipo de información requerida no se adapta a ese tipo de procesamiento. Y esto alude al hecho de que características propias de la inteligencia humana pueden ser requeridas, como puede ser la creatividad, el humor, los dobles sentidos, juegos de palabras, el ingenio, etc. difícilmente reemplazables por técnicas de inteligencia artificial. Véase como ejemplo que permita contextualizar esa tendencia ciertas emisiones (programas de televisión o radio) donde se pide la colaboración de los espectadores aprovechando la inmediatez en la comunicación que ofrece internet, utilizando como nexo de comunicación las redes sociales, para pedir contenidos en torno a una temática, ideas, sugerencias o material multimedia, obteniendo así para el programa **contenido original** y con el mínimo esfuerzo, al margen de la necesidad de clasificar el contenido recibido.

Así pues, de esto se desprende que existe una alternativa al descubrimiento de contenidos ya existentes, basada en explotar la **Web como espacio colaborativo** donde promover la generación de nueva información aportada directamente por los usuarios. En lugar de considerar los datos almacenados disponibles en la Web como los recursos aprovechables, se pasa a considerar al **usuario como activo** de interés.

A nivel general, ante la problemática de conseguir reunir información de interés, se ha determinado que a través de la colaboración de las personas se puede conseguir una fuente casi inagotable de conocimiento que podría satisfacer esa necesidad. Ahora bien, la pretendida colaboración no surgirá de manera espontánea. Es necesario elaborar una propuesta que resulte atractiva, para que el usuario quiera participar. Y ante la problemática de promover esa cooperación, la solución que se propone viene dada a través de la gamificación.

1.3 Incentivo a través de la gamificación

Desde que aparecieran en los años 70 los primeros videojuegos como alternativa de ocio, estos se han ido consolidando como una opción real y masiva de entretenimiento, atrayendo cada vez más público. Este éxito propició que se analizaran los factores que justifican este fenómeno desde un punto de vista psicológico, llegándose a la conclusión de que los videojuegos se aprovechan de ciertos rasgos o pautas del comportamiento humano para incitar y **motivar al jugador** a seguir jugando, basando el interés en la diversión generada al ir alcanzando los objetivos que se le proponen. Si se toma este hecho aislándolo del contexto de los videojuegos, se tiene que existen una serie de dinámicas y mecánicas que, trasladadas al mundo real, pueden lograr el mismo efecto, esto es, incentivar a las personas para que alcancen ciertos objetivos o realicen ciertas acciones. Esta es la idea básica en torno a la cual se centra la gamificación, que viene recogida y desarrollada a modo introductorio en [DDKN11], un artículo de referencia respecto a esta técnica.

La gamificación pretende incorporar los mecanismos extraídos de los juegos en cualquier ámbito real que requiera una cierta implicación por parte de las personas, consiguiendo que ese esfuerzo resulte atractivo para ellas. El objetivo será ligeramente diferente según el ámbito en el que se quiera incorporar, como puede ser el sanitario (aumentar la receptividad), educativo (potenciar las capacidades), laboral (incrementar productividad) o comercial (influir y atraer clientes), entre otros. Existe un ámbito específico de interés, el de las comunidades y foros, donde el objetivo que se perseguiría incorporando la gamificación es el de **potenciar la colaboración** de los usuarios. Conservando la idea relativa al enfoque colaborativo, en este punto se enlaza con el objetivo inicialmente introducido. Si se consigue llegar a un gran número de personas y se les consigue motivar por este medio, estas pueden contribuir con su conocimiento a la construcción de grandes sistemas mediante la suma de pequeños esfuerzos individuales.

Como ejemplo paradigmático de aplicación de gamificación en un ámbito de este tipo pueden destacarse los foros dedicados a resolver cuestiones relativas a la programación o el desarrollo software. En estas disciplinas existe un fuerte componente práctico y las dudas sobrepasan los conocimientos puramente teóricos. Así, con el fin de **incentivar a la comunidad** para que resuelva las preguntas planteadas por otros usuarios, estos foros incorporan técnicas de gamificación orientadas a reconocer y recompensar el esfuerzo. Como es el caso

de Stack Overflow², donde aumenta la reputación del usuario cuando recibe votos positivos para sus preguntas (por útiles) o respuestas (por aclaratorias), permitiéndole así conseguir logros y desbloquear privilegios. Microsoft Community³ es otro foro dedicado a una tecnología específica, donde se otorgan diversos distintivos que miden la actividad de cada usuario y le permiten aparecer en el muro de colaboradores destacados.

En este trabajo se apostará por un desarrollo basado en gamificación, lo que parece ser una opción recomendable según las cifras. Un estudio realizado en 2013 por la agencia Gartner⁴ postulaba la gamificación como una de las tecnologías que mayor expectación generaba. Markets and Markets⁵, tomando como base los datos generados en el año 2014, estimó una tasa anual de crecimiento del sector(CAGR) del 46.3 % para el periodo 2015-2020.

1.4 Ámbito de interés: Concursos de preguntas

De todos los ámbitos que podrían beneficiarse de la aportación de información de la forma previamente descrita, este trabajo pone el foco en el ámbito del entretenimiento. Más concretamente uno de los sectores que podría beneficiarse de esta aportación es el de los **concursos y juegos**. La base de ciertos tipos de concursos y juegos es la de poner a prueba a un conjunto de oponentes que deben demostrar su habilidad, pericia intelectual o conocimiento en base a distintos esquemas de pruebas que requieren información heterogénea para su constitución. De todos esos tipos, los basados en un esquema de tipo **preguntas/respuestas** requieren la recopilación continuada de un volumen significativo de información diversa.

Esta es un área de trabajo atractiva en tanto que ha demostrado tener la capacidad de llamar la atención del gran público. Puede tomarse **la televisión como indicador** representativo de este hecho, en la medida que desde el inicio de las emisiones hasta la actualidad, este tipo de concursos han tenido un papel protagonista. En [Mor14] se realiza un análisis de la evolución de los concursos en la televisión española, donde se destaca la relevancia de los concursos de preguntas desde los inicios en 1956. Por otro lado, en [Gue05] se desglosa con datos la producción por formatos en la televisión española desde 1996 a 2004, siendo el tiempo de emisión dedicado a concursos (entre ellos, el tipo que nos ocupa) únicamente superado por el género de ficción. En la memoria colectiva han quedado concursos como *¿Quién quiere ser millonario?* que, como otros concursos, ha sido exportado a múltiples países, lo que demuestra que el interés por estos puede considerarse común denominador a nivel global. Este parece ser un formato que, aun presentando una mecánica recurrente, no se desgasta con el paso del tiempo, e incluso en la actualidad parece haber un repunte con concursos de emisión diaria como *Ahora caigo*, con una media de millón y medio de telespectadores⁶.

²<http://es.stackoverflow.com/tour>

³<http://answers.microsoft.com/es-es/>

⁴<http://www.gartner.com/newsroom/id/2575515>

⁵<http://www.marketsandmarkets.com/PressReleases/gamification.asp>

⁶<http://ecoteuve.eleconomista.es/programa/Ahora-Caigo/audiencias/>

Dado que parte del éxito de estos concursos se debe a que permite al espectador ponerse a prueba y jugar paralelamente al programa, tiene sentido que sea este mismo **éxito trasladable a los videojuegos**, pudiéndose comprobar en plataformas como *Google Play Store*⁷ que juegos tipo trivial como *QuizUp* o *Atriviate* acumulan millones de descargas.

1.5 Qué se pretende hacer

Una vez se ha revisado la necesidad a cubrir y el ámbito de interés para el cual sería interesante proponer una solución, se presenta la idea base de este trabajo. Inspirándose en la idea del aporte colectivo, se propone la creación de un sistema web que permita construir una gran base de datos de un conocimiento específico recopilado a través de las aportaciones de usuarios, aprovechando como recurso la inteligencia colectiva. Llevando este concepto genérico a un enfoque práctico, el objetivo principal del trabajo será la creación de un sistema web que permita **recopilar preguntas**, cuyo fin último sería su integración en un juego de tipo trivial. Este sistema será operativo y funcional, aunque no es una pieza que funcione o que tenga sentido por separado, sino que está orientada a dar soporte a la aplicación móvil o web sobre la que han de volcarse las preguntas recopiladas.

A nivel básico, ha de crearse un módulo que permita agregar las preguntas que deseen aportar al sistema aquellos que quieran colaborar. Con el fin de hacer esta utilidad llamativa a los jugadores en primera instancia y de promover posteriormente un uso continuado en el tiempo, se incorporará además un fuerte **componente de gamificación**, focalizado en un perfil de colaborador a través del cual se dará salida a la estrategia definida. Se trata de conseguir mejorar el juego involucrando a los jugadores, pero siempre ofreciendo la colaboración de tal forma que esta sea percibida de forma positiva, divertida y enganando al jugador lo máximo posible.

Por otro lado, hay que tener en cuenta que el problema fundamental cuando se solicita colaboración es que la información recibida puede ser correcta o no serlo, al no resultar adecuada por no adaptarse a lo que se espera recibir, o bien porque pueda haber sido introducida malintencionadamente. De esto se desprende la necesidad de aplicar un **proceso de validación** que garantice la calidad y consistencia de la misma. Además de esto, cabe esperar que una misma pregunta pueda ser sugerida por distintas personas, lo que es un problema si no se desea contar con preguntas repetidas. Es por todo ello que además de permitir la introducción de preguntas, se incorporará un mecanismo que permita revisar, validar e integrar los aportes. Se opta así por habilitar la posibilidad de realizar una revisión manual de preguntas (no abierta a jugadores sino accesible solo a un grupo cerrado de personas dedicadas a tal fin) complementando esto con cierto procesamiento automatizado que permita detectar si una pregunta propuesta es similar a una pregunta ya aceptada.

⁷<https://play.google.com/store/apps/category/GAME?hl=es>

Se muestra de forma simplificada en la Figura 1.1 una síntesis de las ideas planteadas previamente: desde la utilidad en sí misma del sistema, permitiendo a los jugadores proponer preguntas y que éstas sean revisadas y validadas, hasta el papel que ocupa la gamificación. En este sentido, las técnicas de gamificación escogidas tratan de reforzar la idea del **reconocimiento como elemento motivador** en los jugadores. Se trata de poner de relieve la idea de que el jugador forma parte del juego y que sus aportes resultan ser útiles. A tal efecto se trata de medir y reconocer su actividad a través de la implementación de un sistema basado en **puntos de experiencia, niveles y rangos**. Además se proponen una serie de **metas a alcanzar** a modo de trofeos y una serie de **rankings de carácter mensual** para tratar de animar a la participación continuada a lo largo del tiempo.

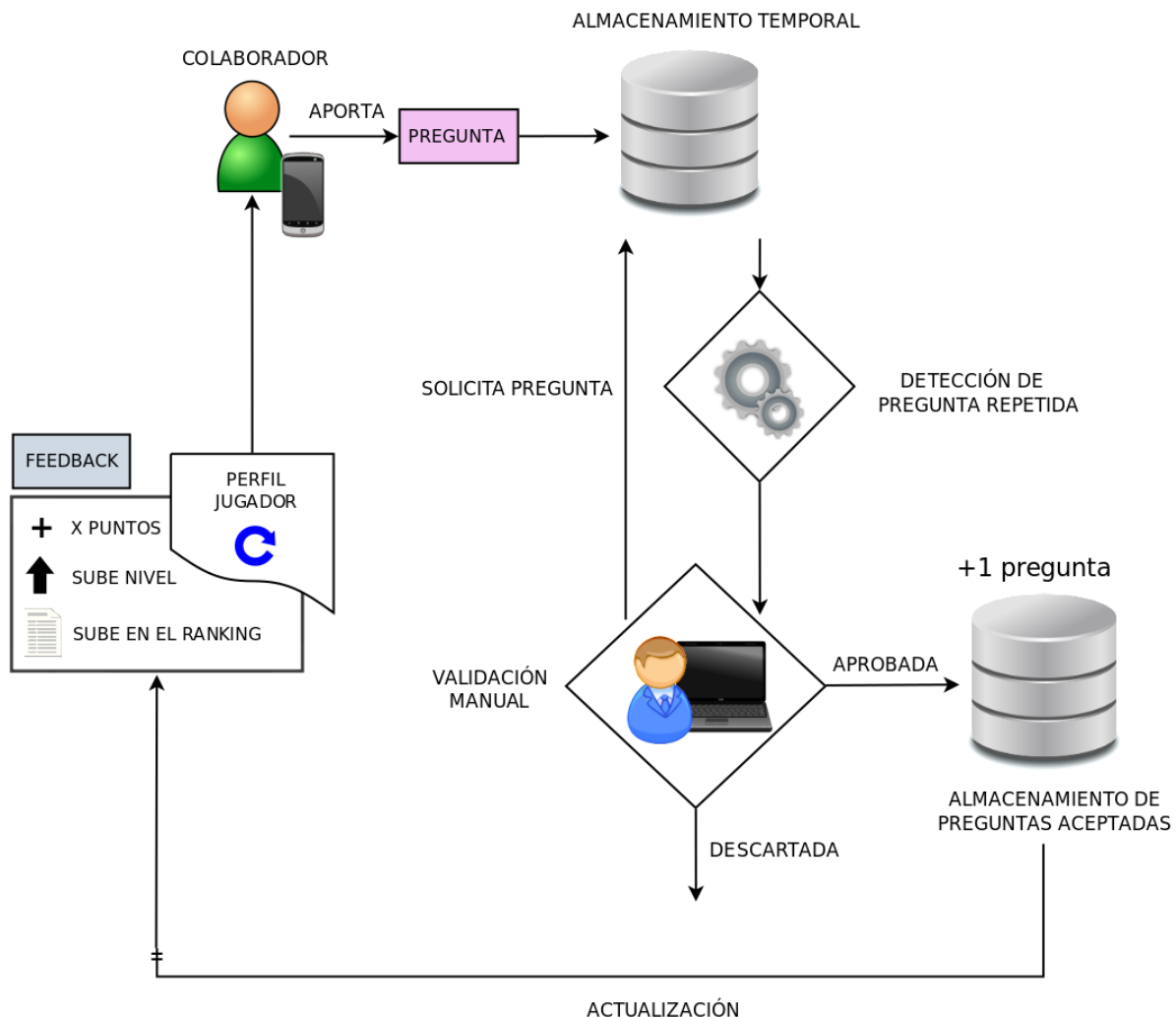


Figura 1.1: Flujo de aporte y validación de preguntas

1.6 Contexto de desarrollo

Se abordan en este apartado de forma breve aspectos relevantes que caracterizan y/o condicionan el desarrollo y la dinámica de trabajo. Por un lado, se comenta la naturaleza del desarrollo software escogido, que determina las opciones a la hora de realizar una hipotética integración posterior de la solución. Por otro lado se destaca como factor diferencial la existencia de un marco de colaboración con la empresa Furious Koalas.

1.6.1 Desarrollo basado en tecnologías web

Escoger un desarrollo web responde a la intención de abrir el abanico de posibilidades de cara a una integración posterior en un juego de preguntas. Al hacerlo de esta forma se dispone de una **aplicación web portable** a otra de la misma naturaleza, y también integrable en plataforma móvil a partir del mismo código empleando un framework que permita generar una aplicación híbrida. Dado que de acuerdo a las necesidades de la aplicación no se requeriría el acceso completo a todas las funcionalidades del móvil, este enfoque supone una buena aproximación. Según si la integración se quisiera abordar para un juego en plataforma móvil o web y la concordancia de las tecnologías empleadas por estas se podría abordar de dos formas:

- Aproximación básica. Consistiría en mantener separado el juego del sistema de recopilación de preguntas. Este último se mantendría en su propio dominio y desde el juego se promocionaría y se redirigiría a los usuarios a este. La integración requeriría una revisión del aspecto visual acorde al juego y estudiar de qué forma se integrarían las preguntas desde el sistema al propio juego (que podría ser manual). Este enfoque se adaptaría más si el juego de preguntas fuese una aplicación web o las tecnologías empleadas no fuesen compatibles.
- Aproximación integradora. Fuese el juego una aplicación web o móvil, en caso de que las tecnologías fuesen compatibles, cabría la posibilidad de integrar todo el código desarrollado en el juego. De esta forma se le estaría agregando directamente una nueva funcionalidad. También requeriría una revisión del aspecto visual y un estudio de la organización del proyecto del juego para incorporar todo el código independiente de forma coherente al diseño y descartar código asociado a funcionalidades básicas ya incorporadas en el juego.

Dadas las dos posibilidades, lo que se pretende en este trabajo es **considerar una posible integración parcial del sistema en un juego para plataforma móvil**. Esto permitiría afrontar un proceso de integración real para el caso concreto de plataforma móvil y coincidencia en las tecnologías empleadas. A tal efecto se requiere la colaboración con una empresa propietaria de un juego basado en preguntas, aspecto que se detalla en la subsección 1.6.2.

Por otro lado, la estrategia de implementación en cuanto a su vertiente tecnológica pasa por el uso de la plataforma **Google App Engine**, un servicio de computación en la nube ofrecido como PaaS por Google Cloud Platform⁸ destinado al desarrollo y alojamiento de aplicaciones web. Un punto a favor de esta tecnología es que ofrece una serie de servicios y APIs útiles que dan soporte al desarrollo. Permite desplegar la aplicación en los servidores de Google de una forma fácil y rápida, delegándose la gestión de aspectos como la escalabilidad y la disponibilidad y asignándose además unos recursos mínimos de forma gratuita bajo un dominio propio.

1.6.2 Colaboración con empresa Furious Koalas S.L.

El desarrollo de todo el proyecto se realizará en colaboración y bajo la supervisión de la empresa Furious Koalas, propietaria de la IP *Whizz*, juego para móvil que podría aprovechar el sistema implementado integrándolo⁹.

Furious Koalas S.L. es una joven spinoff de la UCLM, fundada en 2015 como empresa tecnológica dedicada al diseño, desarrollo y comercialización de videojuegos. Su principal ambición es convertirse en un referente en el mercado de videojuegos para móviles, tanto a nivel nacional como internacional. Su filosofía a nivel estratégico está basada en centrar sus esfuerzos en el lanzamiento de una o dos propiedades intelectuales, potenciando al máximo sus posibilidades comerciales, y llevando a cabo colaboraciones con otras empresas.



Figura 1.2: Logo Furious Koalas S.L.

Whizz - The Massive Quiz es en la actualidad su producto estrella, el primer juego para móvil multijugador masivo de preguntas y respuestas en tiempo real, con un alto componente social y una dinámica de concurso televisivo. Disponible para Android e iOS, cuenta con características que lo distinguen de la competencia, así como con una cuidada estética. Su aspecto visual puede apreciarse en la figura 1.3.

⁸<https://cloud.google.com/>

⁹Parte del desarrollo de este trabajo se llevó a cabo en el contexto de unas prácticas en empresa

Whizz es el único **juego Multiplayer Massive Online (MMO)** con una dinámica basada en preguntas a nivel mundial. En Whizz todos los jugadores del mundo responden simultáneamente a la misma pregunta. Permite además crear partidas en grupo sin límite de jugadores. Por otro lado, incluye diferentes ataques, penalizaciones y *power-ups* que dinamizan la experiencia de juego. Otra de sus características destacadas es que permite crear **concursos localizados** y de contenido específico, con preguntas de cierta temática programadas para franjas horarias concretas, por lo que puede llegar a emplearse también como herramienta para la generación de contenidos de interés cultural o promocional auspiciados por instituciones u organismos. En cuanto a su modelo de negocio, es **de tipo Free to play (F2P)**, obteniendo sus ingresos a partir de In-app Purchases, la contratación de micro-publicidad, canales temáticos y preguntas patrocinadas.

En el momento de comenzar el proyecto el juego contaba con aproximadamente una **base de datos de 5000 preguntas**. Una cifra considerable, pero que convendría incrementar para dotar de mayor variabilidad al juego. Resultaría muy positiva, por tanto, la adhesión de un módulo a través del cual se puedan aportar nuevas preguntas y a través del cual los jugadores refuercen su vinculación con Whizz.



Figura 1.3: Aspecto visual de Whizz

1.6.3 Estructura del documento

A continuación se listan los contenidos que pueden encontrarse a lo largo de este documento. A tal efecto se destaca brevemente qué puede encontrarse en cada capítulo:

- **Capítulo 2 - Objetivos** Se detalla a modo de listado cuáles son las metas definidas, quedando expuesto qué es lo que se pretende hacer de una forma más sistemática.
- **Capítulo 3 - Estado del Arte** Se realiza una completa revisión en torno a los temas que se tratan en el trabajo. Concretamente se recopila información relevante relativa a la gamificación, al mercado de juegos para móviles y una serie de consideraciones cuando se trata de afrontar el desarrollo de una aplicación móvil. Además se incluye el análisis de una alternativa ya existente que comparte similitudes con el desarrollo que se afronta en este trabajo.
- **Capítulo 4 - Método de trabajo** De cara a contextualizar de qué forma se ha afrontado el desarrollo se describe qué metodología se ha seguido, así como las condiciones software y hardware que lo caracterizan.
- **Capítulo 5 - Arquitectura** En este capítulo se pone atención al detalle acerca del diseño y las particularidades de implementación de la solución propuesta para cada componente del sistema. Se ofrece una panorámica general de la estructura del proyecto para su constitución, del frontend al backend, explicándose además la opción escogida para lidiar con el almacenamiento persistente.
- **Capítulo 6 - Resultados** Aquí se muestra y describe cuál ha sido el resultado final obtenido, justificando en qué medida se han logrado los objetivos inicialmente planteados. Se presenta además una panorámica de los distintos sprints afrontados que representan la planificación de tareas abordada y se perfila de qué forma se deja preparado el sistema para una futura integración en el sistema real.
- **Capítulo 7 - Conclusiones** Por último se reflexiona globalmente acerca de la utilidad del resultado obtenido y de las contingencias enfrentadas durante el camino seguido para llegar a este. Se plantean además líneas a seguir de cara a trabajo futuro.

Objetivos

DADOS el enfoque y las ideas presentadas en la introducción, en este capítulo se concreta qué se pretende llevar a cabo, sintetizando primero cuál es el objetivo general y pasando a detallar después el conjunto de objetivos específicos derivados. Se definen así una serie de hitos que dibujan el camino a seguir para alcanzar el resultado final.

2.1 Objetivo general

Con la finalidad de dar soporte a un juego para plataforma móvil de tipo trivial, se propone el diseño y desarrollo de un sistema web a través del cual los jugadores puedan tomar el rol de colaboradores y **sugerir preguntas**. Con la finalidad de animar la participación e incorporar esta funcionalidad de forma que suponga un valor añadido para el juego, se implementará un **diseño basado en técnicas de gamificación**. También se habilitará la opción de agregar un filtro mediante la posibilidad de realizar una **revisión manual de las preguntas** aportadas, con el fin de que puedan ser revisadas, corregidas, aprobadas o descartadas. Esta validación estará apoyada por una revisión automatizada que trate de **detectar preguntas repetidas**. Se aludirá en adelante a este sistema mediante el acrónimo **QuPI-GAM**¹.

El desarrollo se ha llevado a cabo en colaboración con la **empresa Furious Koalas**, quienes han lanzado *Whizz*, un juego MMO para plataforma móvil basado en preguntas que se adapta perfectamente a las necesidades que pretende cubrir QuPI-GAM. Esto permitirá en una última fase estudiar lo que supondría abordar un **proceso de integración** y dejar el sistema preparado para que esta pudiese llevarse a cabo si procede en el futuro, para lo cual se irían incorporando las distintas funcionalidades presentes en QuPI-GAM.

Utilidad de QuPI-GAM

Los objetivos pueden dirimirse en dos sentidos. Este capítulo se centra en definir qué se pretende hacer. Sin embargo también es importante justificar por qué se quiere llevar a cabo en términos del beneficio esperable. Esto entronca con los objetivos a otro nivel: los **objetivos de negocio**. Las consecuencias positivas derivables de la incorporación de esta solución en un juego de preguntas (ampliadas en la sección 5.8) se resumen a continuación:

¹*Question Integration by means of Players Input applying GAMification Mechanisms*

- El juego se ve potenciado al incorporar una nueva funcionalidad que puede llamar la atención de los jugadores y **enriquecer su dinámica** (apoyándose esto fundamentalmente en el enfoque gamificado).
- Se consigue **incrementar el número de preguntas** disponibles en la base de datos del juego, así como **diversificar la temática** de las preguntas. La calidad del juego mejorará en tanto que la experiencia de juego se verá enriquecida.
- Englobando los dos puntos anteriores se tendría un juego más completo. A efectos prácticos se traduciría en un **incremento de nuevos jugadores**, un **mayor número de sesiones** iniciadas por jugador y así como en el porcentaje de **retención**.

2.2 Objetivos específicos

Se presentan a continuación los objetivos específicos que se pretenden abordar. A grandes rasgos, estos pueden agruparse en torno a las siguientes líneas: incorporación de funciones de soporte para la constitución básica del sistema web, inclusión de funcionalidad para aportación de preguntas, inclusión de funcionalidad para la revisión/validación de preguntas y la inserción de técnicas de gamificación. A todo ello hay que sumar como broche final el volcado de parte del desarrollo en un sistema real mediante un proceso de integración.

Constitución básica del sistema web

Con el fin de contituir un marco sobre el que se puedan agregar las funcionalidades requeridas se deben acometer las siguientes tareas:

1. Realizar un estudio que permita contextualizar el entorno de trabajo. Esto supone investigar acerca de lo que implica y hasta dónde llega el mercado y el ecosistema del desarrollo de juegos para móviles. Esto se ha de volcar como estado del arte.
2. Agregar un servicio de terceros que permita gestionar la autenticación de los usuarios.
3. Incorporar un mecanismo que permita gestionar correctamente la sesión de cada usuario (creación, asignación, recuperación y cierre).
4. Introducir una política de autorización de acceso y asignación de roles que permita disponer de un control de acceso a zonas restringidas del sistema.
5. Incorporar un mecanismo para el soporte a la internacionalización.
6. Emplear tecnologías multiplataforma y diseños responsive que garanticen la correcta visualización en entornos multidispositivo.

Soporte para la aportación de preguntas

Para incorporar una interfaz a través de la cual se puedan proponer preguntas hay que:

1. Crear un formulario accesible por jugadores tanto identificados como anónimos que permita introducir una pregunta, la respuesta correcta y opciones incorrectas, así como especificar idioma de la pregunta y categoría a la que pertenece.
2. Incorporar un mecanismo de protección frente a robots en el formulario de aportación, para evitar ataques que traten de saturar de forma automatizada el sistema. Este debe ser permanente para usuarios no logueados y aparecer únicamente para la primera pregunta que aporte el jugador al inicio de cada sesión.

Estimulación a través de gamificación

De forma complementaria al soporte para la aportación, se ha de incentivar a los jugadores para que colaboren y encuentren atrayente esta funcionalidad extra. A tal efecto, se espera explorar la aplicación de distintas mecánicas y dinámicas de juego. El conjunto de objetivos a alcanzar por esta parte son:

1. Estudiar referencias de interés, artículos y libros dedicados a gamificación para tomar consciencia de los últimos avances. Toda la información de interés localizada debe ser recopilada como estado del arte.
2. Proyectar un diseño basado en gamificación y realizar un análisis que considere los factores condicionantes que entran en juego para justificar este diseño.
3. Analizar si existe en el mercado una alternativa similar a la que se pretende acometer para tomarla como referencia y/o punto comparativo.
4. Crear un perfil de colaborador, como punto central donde el jugador pueda consultar su actividad y todo el *feedback* relativo a la gamificación.
5. Definir un mecanismo que permita ofrecerle *feedback* al jugador, gracias al cual pueda conocer el impacto de su aportación en el sistema, en términos de preguntas aceptadas, agregadas al juego y descartadas. Se le ha de informar de sus avances y logros conseguidos desde la última vez que visitó el perfil.
6. Asignar al colaborador puntos de experiencia, en base a todos los parámetros de actividad medibles, como unidad base que permita a su vez incorporar un mecanismo de subida de nivel y alcance de rangos que mida su actividad y permita establecer un marco comparativo respecto a otros jugadores.
7. Establecer metas que puedan serle propuestas al jugador a modo de trofeos a conseguir.
8. Incorporar la construcción de rankings mensuales, en base a distintos parámetros combinables como número de preguntas, número de puntos de experiencia, localización del usuario y categoría de la pregunta.

Soporte a la validación de preguntas

Con el objetivo de aplicar un filtro a las preguntas recibidas para garantizar su corrección, como paso previo a que pasen a formar parte de la base de preguntas de las que se nutre el propio juego, se proponen los siguientes objetivos:

1. Crear un formulario de validación (no accesible a jugadores sino a personal destinado) a través del cual se puedan recuperar las preguntas subidas por los jugadores. El validador podrá visualizar la pregunta, corregirla y aceptarla o bien descartarla, si no se considera de valor. El validador podrá seleccionar el idioma y categoría para su recuperación, así como corregir estos parámetros.
2. Desarrollar un mecanismo automatizado que trate de detectar si ya ha sido incorporada definitivamente una pregunta al sistema que verse sobre una temática similar a una pregunta recuperada pendiente de validación.

Integración del sistema

Con la finalidad de comprender las implicaciones de un procedimiento de integración se propone el siguiente objetivo:

1. Dado el marco colaborativo que se ha establecido con la empresa Furious Koalas, no se puede plantear hacer efectiva una integración al completo en la versión comercial de (*Whizz*), pero sí que se plantea estudiar las implicaciones de llevar a cabo la integración y dejar el sistema preparado para una futura integración planteada de forma incremental.

Capítulo 3

Estado del arte

A lo largo de este capítulo se recoge una recopilación de los elementos principales sobre los que se asienta este trabajo y que constituyen su base tecnológica y de conocimiento. Dado que se pretende afrontar un desarrollo no orientado a la investigación, y que el objetivo principal no es la aportación de nuevo conocimiento desde un punto de vista teórico, la presentación no se centrará en torno a la elaboración de un estudio comparativo frente a otras soluciones similares que se pretendan mejorar. Se realizará en cambio un recorrido analizando la contribución, posibilidades y aportaciones relevantes de las diferentes áreas implicadas, justificando así la utilidad de realizar un desarrollo basado en éstas.

3.1 El mercado de los juegos para móvil

3.1.1 El móvil como elemento de interés

De acuerdo a los datos provistos por la GSMA, asociación que representa a los operadores de telefonía móvil, en 2016 existen más de **7800 millones de conexiones móviles**¹, superando este dato a la población mundial, estimada en algo más de 7400 millones de personas. Si bien no puede decirse que la distribución de conexiones sea uniforme a lo largo del globo, y por lo tanto esto no implique que cada persona sea poseedora de al menos un terminal móvil, esta comparativa sí que permite reflejar que existe una elevada demanda de tendencia creciente con respecto al uso de este tipo de líneas, y por tanto consecuentemente de dispositivos móviles como smartphones o tablets.

Esta panorámica se ve reforzada si se tiene en cuenta que los dispositivos móviles son herramientas al servicio de la comunicación, gracias a su capacidad para conectarse a distintos tipos de redes. La irrupción de Internet ha supuesto una mejora revolucionaria en este aspecto, en tanto que ha permitido tener al alcance del bolsillo **dispositivos inteligentes que pueden cumplir múltiples funciones**. Pueden adaptarse a las necesidades y gustos de su dueño, principalmente a través de la instalación y uso de aplicaciones de naturaleza diversa que requieren en la mayoría de casos conectividad.

¹<https://www.gsmainelligence.com/>

Intensa penetración y uso

Según el informe elaborado en el año 2015 por la Fundación Telefónica España [SIE15], página 66, se determinó que el **acceso a Internet a través de smartphone** había superado por primera vez al acceso realizado a través del ordenador, concretamente un 88,3 % frente a un 78,2 %. Esto refuerza la idea del uso creciente y la utilidad que se les está dando a este tipo de dispositivos. La brecha es mayor dependiendo del grupo de edad observado, sin embargo en este informe también se pone de manifiesto que cada vez se produce un acercamiento mayor a este tipo de tendencias y a la adaptación tecnológica por segmentos más maduros, por lo que la resistencia impuesta por la barrera de la edad se va debilitando.

Desde la agencia de consultoría comercial Tatum, y concretamente desde su división dedicada al desarrollo de soluciones digitales, se elaboró un estudio denominado *Mobile en España y en el Mundo 2015*² cuyo fin es poner de manifiesto lo beneficiosa que puede resultar la adaptación a lo *mobile* y lo digital por parte de las compañías. Se justifica con una serie de datos interesantes, que permiten ilustrar este punto en cuanto a la atención que los usuarios dedican a sus smartphones y tablets:

- **Irrupción de los dispositivos móviles.** En 2014 se incrementaron las ventas de smartphones un 28,4 % con respecto al año anterior, vendiéndose más de 1200 millones de terminales. Esa tendencia creciente es observada año tras año y aunque la distribución no es uniforme, la menor proporción de número de dispositivos respecto a la población total se localiza en el Sur de Asia, y alcanza un 77 %, llegando a un superávit del 139 % en Europa del Este.
- **Atención dedicada.** El tiempo de ocio que antes era cubierto por la televisión, ahora lo es cada vez más por navegación en Internet a través del móvil. Como dato de muestra, en España el 90 % de los propietarios de un smartphone se conectan cada día a Internet a través de este y 7 de cada 10 consultan el móvil un mínimo de 10 veces al día. Respecto a las tablets, es destacable el hecho de que 8 de cada 10 usuarios dedican al menos una hora al día a su uso y cada vez son más los usuarios que disponen y emplean paralelamente ambos tipos de dispositivos.
- **Tipo de uso.** Entre las actividades recreativas preferidas a realizar a través de los dispositivos móviles se encuentran las siguientes: navegación por la Web, mensajería instantánea, interacción con redes sociales, consulta de correo electrónico, visualización de vídeos, juegos y consulta de noticias. Pero lo más interesante que se refleja en este estudio es la forma en que se accede a todas esas actividades. Los usuarios manifiestan una clara predilección a realizar la conexión a través de aplicaciones específicas que ofrecen toda esa variedad de servicios. Se estima que del total de tiempo que los usuarios pasan conectados, casi el 90 % del tiempo es a través de aplicaciones, quedando la navegación directa por Internet relegada al 10 % restante. Es por ello que el catálogo

²<https://goo.gl/hG3i8Y>

de apps ha ido creciendo de forma exponencial en los últimos años hasta 2014, año en que se contabilizaban un total de 2,5 millones de apps disponibles entre las ofertadas por Google Play y el App Store. En la figura 3.1³ se recogen esta y otras cifras ilustrativas respecto al mercado de las aplicaciones móviles. Así, en base a estos datos puede concluirse que la forma más efectiva de llegar a los usuarios es a través del desarrollo de aplicaciones móviles.



Figura 3.1: Datos ilustrativos del mercado de Apps (2014)

Justificación teórica del fenómeno

Esta tendencia, revelada a través de las cifras, trata de ser justificada y comprendida también de forma teórica a través de estudios que ahondan en las motivaciones y los factores que justifican esa relación cada vez más intensa que el usuario establece con su smartphone. En [Mil14] se presenta un artículo donde se analiza el por qué de la **rápida penetración** de estos dispositivos, denominados aquí como ‘la cuarta pantalla’ por detrás del cine, los televisores y los ordenadores. Su intensa irrupción, solo comparable tecnológicamente a la penetración de los televisores en Estados Unidos durante los años 50, es analizada apoyándose en la **teoría de la mediatización**, que trata de poner de manifiesto las consecuencias más amplias de la incorporación de los medios de comunicación, plataformas digitales y contenidos multimedia en la vida cotidiana. Se postula una visión donde poco a poco la tecnología se irá fusionando con el medio, de tal forma que los paradigmas de interacción de la vida diaria se vean modificados, afectando a la realidad misma y a la connotación espacio-temporal de las comunicaciones.

³Fuente: InformeDitrendia(2015)

En ese proceso evolutivo guiado por el avance tecnológico, el smartphone se coloca como dispositivo intermediario del cambio, un catalizador a través del cual ir introduciendo innovaciones de forma regular, de modo que los usuarios se adapten a los cambios de forma natural. Llega a postularse como una llave de entrada a una posible y futura **tecnogénesis**. Asimismo, existen factores cognitivos-conductuales que pueden justificar el apego al móvil, al margen de la utilidad en sí misma que pueda derivarse de su uso, siendo algunos de ellos: considerar el smartphone como una parte extensible de uno mismo, a través del cual poder expresarse y verse representado; o el atractivo de poder desarrollar un tipo de vínculo afectivo diferente con otras personas basado en una especie de *telepresencia* virtual.

Puerta abierta a oportunidades de negocio

Puede decirse que el uso de dispositivos móviles, y más concretamente su uso ligado a actividades que requieren el acceso a Internet, es una realidad que se ha asentado en el día a día. El móvil se constituye como **herramienta cotidiana y versátil, fuertemente explotable** y de interés, tanto para los usuarios finales como desde la perspectiva de negocio. Tal ha sido el impacto de esta nueva realidad que ha llevado a que empresas y entidades redefinan sus modelos y estrategias de negocio, aprovechando la potencialidad de las aplicaciones móviles para agregar valor. Como muestra, se citan algunos de los sectores que ya se están viendo beneficiados gracias al desarrollo de aplicaciones móviles:

- **Mobile commerce.** Cada vez más negocios optan por implementar una solución de comercio electrónico a través de las cuales ofrecen sus productos y servicios de forma online. Teniendo en cuenta que la tasa de crecimiento del m-commerce en el mundo se sitúa en un 23,8 % entre 2013 y 2016, desarrollar apps adaptadas que faciliten el proceso de compra puede ser la clave que determine el éxito de la solución de comercio electrónico, aunque dependiendo del tipo de negocio, una web adaptada al entorno móvil y *responsive* puede ser una opción bien percibida y útil para los compradores.
- **Banca electrónica.** A finales de 2014, 800 millones de usuarios accedieron a su banco a través del smartphone. Se estima que esa cifra se verá duplicada para 2019 y se asume que todos los bancos deberán abordar una estrategia digital en un corto periodo de tiempo, ya que los usuarios requerirán este tipo de solución frente a la alternativa clásica de acudir físicamente a la sucursal.

Los anteriores datos reflejan la utilidad, desde el punto de vista de negocio, de emplear aplicaciones como herramienta a través de la cual llegar al cliente de una forma alternativa, que incluso puede constituirse como la principal. Como dato significativo, en el Apple Store en 2014 la categoría de negocios fue la que experimentó un aumento mayor, situado en un 210 %. Sin embargo no todas las aplicaciones deben ser concebidas como medio, sino que también pueden constituirse como un fin en sí mismo, siendo ofrecidas como producto.

En este sentido, uno de los ámbitos reinantes es el del entretenimiento, y más concretamente, los juegos para móviles. **En la tienda Google Play la categoría de juegos fue la que experimentó un aumento de apps mayor**, situándose en el 260 % en 2014, dato lógico si se asume que un alto porcentaje de dueños de smartphone reservan un hueco para el recreo que espera ser cubierto por un juego. Dado el interés justificado a lo largo de este apartado por las aplicaciones móviles, como punto de entrada a través del cual llegar a millones de usuarios, el foco de interés de este trabajo se centra en la categoría de juegos, y es por ello que en los siguientes apartados se profundiza en ese sentido.

3.1.2 Caracterización de los juegos para móvil

Antes de profundizar en los juegos para móviles, conviene dar un breve repaso a la corta pero prolífica historia de los videojuegos, como punto de partida que permita contextualizar y situar su aparición y el lugar que ahora ocupan. Puede decirse que las diferentes etapas por las que ha pasado la industria del videojuego han venido marcadas en gran parte por **las posibilidades y límites establecidos por los dispositivos hardware** sobre los cuales habían de ser ejecutados los videojuegos. Desde ese punto de vista, en un cierto punto el móvil pasó a formar parte de este conjunto de artefactos. Aspectos que han definido la evolución de los videojuegos como la exploración de diferentes géneros, mecánicas de juego y formas de interacción son dependientes en parte de los límites establecidos por el hardware.

Aunque existen otros factores condicionantes, desde las diferentes técnicas que se han ido aplicando a lo largo del tiempo (para un punto de vista del desarrollo y el aspecto gráfico) hasta los cambios en el público objetivo que van derivando en distintas tendencias, el enfoque del hardware constituye una buena perspectiva para revisar algunos de los hitos importantes en la historia de los videojuegos y delimitar diferentes etapas.

Breve repaso a la industria del videojuego

Las primeras manifestaciones de juegos no nacieron con un fin lúdico, más bien como medio de pruebas a través del cual poder **demostrar el potencial de un determinado software o hardware** en laboratorios de investigación. Se trasladaron juegos clásicos como el ajedrez, las damas o el tres en raya al entorno virtual. El punto de ruptura vino dado a finales de los años cincuenta mediante un juego denominado *Tennis for two*⁴ ejecutado en lo que puede considerarse un ordenador analógico, a través de un osciloscopio y exhibido en un laboratorio, que recreaba una partida de tenis. Este se considera el primer juego desarrollado con el **fin de entretener**, aunque estuviese destinado a los visitantes del laboratorio y no persiguiese aún un fin comercial.

⁴<https://www.bnl.gov/about/history/firstvideo.php>

Para destacar algunos hitos relevantes que ilustren el primer periodo asociado a los juegos clásicos, puede recurrirse a la línea temporal incluida en el libro [Ken10]. En 1961 **un estudiante del MIT creó *Spacewar***, considerado el primer juego interactivo para ordenador. Sin embargo esto no sentaría precedente en el arranque de la industria, ya que cuando se empezaron a perfilar los videojuegos como alternativa de entretenimiento lucrativa se apostó por la creación de máquinas específicas donde poder jugar a estos.

Conocidas compañías como SEGA, dedicadas en sus inicios a la gestión de tocadiscos (*jukebox*) y máquinas tragaperras, vieron la posibilidad de diversificarse aplicando el mismo modelo de negocio en nuevas máquinas que desafiasen la destreza del usuario, donde se debían introducir monedas para poder jugar una partida. Se considera que fue Nutting Associates quien lanzó la primera **máquina recreativa** en 1971, que permitía jugar al juego *Computer Space*, una adaptación de *Spacewar* creada por Nolan Bushnell. Este fundaría un año después la compañía Atari y lanzarían otro de los juegos míticos, el *Pong*, basado en el tenis de mesa. A lo largo de la siguientes décadas nacerían nuevas compañías dedicadas a la producción de juegos como Namco o Capcom y se popularizarían las **salas recreativas**, espacios donde público de cualquier edad podía acceder a una variedad de juegos arcade, la mayoría de los cuales hoy son considerados clásicos.

Bajo la misma perspectiva, pero tratando de introducir los juegos directamente en las casas, Ralph Baer exploró la posibilidad de emplear los televisores como un recurso aprovechable donde poder reproducir juegos. Considerado uno de los padres de los videojuegos, impulsó la creación en 1972 de la primera **videoconsola**, la Magnavox Odyssey. Fue un dispositivo que, sin contar con microprocesador ni memoria RAM y ofreciendo juegos de características muy básicas, resultó ser un éxito. La década de los 70 fue un periodo prolífico en el cual se lanzaron muchas otras consolas como Fairchild Channel F (1976), la Atari 2600 (1977) o la Color TV-Game de Nintendo (1977), así como juegos que arrasaron en ventas como *Football*, *Space Invaders* o *Asteroids* entre otros. Muchos de los juegos arcade desarrollados para máquina recreativa terminaban teniendo su versión para consola. A partir de los ochenta otros juegos míticos fueron lanzados como el *Pac-Man* de Namco (1980), *Donkey Kong* de Nintendo (1981), *E.T.* de Atari (1982), *The Legend of Zelda* de Nintendo (1987), *Final Fantasy* de Square Soft (1988) o *Super Mario Bros* de Nintendo (1990), todos ellos accesibles a través de cartuchos intercambiables ejecutables en consolas cada vez más evolucionadas como la Sega Mega Drive o la Super NES de Nintendo.

En los años noventa se produciría un punto de inflexión con la irrupción de novedades como la aparición de **consolas portátiles**. Atari y Nintendo fueron pioneras en este aspecto a partir de 1989 mediante el lanzamiento de la Atari Lynx y la línea de consolas portátiles Game Boy, seguidas un año después por el lanzamiento de la Sega Game Gear. La aparición de la consola PlayStation de Sony en 1994-1995, que requería CD y no cartucho, también sentaría un precedente de cambio respecto al soporte de los juegos.

Paralelamente al desarrollo para consola, y en parte debido a la crisis que sufrió la industria en 1983 que hizo quebrar varias compañías, cobró fuerza la idea de desarrollar juegos para los **ordenadores personales**, ya que eran computadores multipropósito pero con la capacidad suficiente como para ejecutar juegos, entre ellos el *Prince of Persia* (1989), el *Doom* (1993) o el *WarCraft* (1994) como primeras propuestas destacables. Esta dicotomía estaría destinada a mantenerse de forma permanente en el tiempo, ya que en la actualidad ambas alternativas (juegos para ordenador y para consola) siguen siendo opciones plenamente vigentes, lanzándose algunos juegos tanto para pc como para consola o bien existiendo juegos de determinados géneros que preferentemente se juegan en una u otra máquina.

De esta forma, la entrada al siglo XXI se caracterizaría por el abandono de los cartuchos y la desaparición progresiva de las máquinas recreativas como elemento habitual en los lugares de ocio. Las salas recreativas se mantendrían sin embargo algún tiempo más activas gracias a su conversión hacia las **cibersalas**, lugares donde en vez de máquinas recreativas se ponía a disposición de los jugadores ordenadores con conexión a Internet, lo que permitía jugar a los **juegos en red y en línea**, un nuevo concepto de juego multijugador introducido en los noventa que marcaría otro hito en la historia del videojuego. En la figura 3.2 se ilustran algunos de los soportes y juegos icónicos que han sido nombrados hasta ahora a lo largo de esta sección⁵.

Desde entonces han ido apareciendo nuevas generaciones de consolas y consolas portátiles con mayor potencia y prestaciones, y con nuevos periféricos que enriquecen y diversifican la experiencia de juego. A diferencia de etapas previas, **el mercado se ha visto monopolizado** desde el punto de vista hardware para este sector. Si bien inicialmente lo característico eran las desarrolladoras *first party* vinculadas a compañías dueñas de una determinada consola y para la que se hacían desarrollos exclusivos, actualmente existe un podio ocupado por tres compañías que copan el mercado en ese sentido: **Microsoft, Sony y Nintendo**. De esta forma, existen independientemente numerosas *third party* enfocadas al desarrollo y publicación de nuevos títulos.

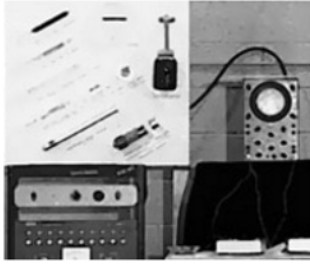
Otros aspecto característico de la última etapa es la progresiva **conversión al soporte digital** en lugar del soporte físico, lo que permite además ofertar contenidos descargables de forma separada al juego propiamente dicho a través de **plataformas** creadas para tal fin, como la Xbox Store⁶ y PlayStation Store⁷ para consola y Steam⁸ para ordenador.

⁵Fuentes: - <https://www.bnl.gov/about/history/firstvideo.php> (1)
- <https://videogamehistorian.wordpress.com/> (2)
- Wikipedia (3,6,7)
- <http://www.insertcoinclasicos.com/2007/04/13/prince-of-persia-1989/> (4)
- <http://www.atari.io/atari/atari-lynx/> (5)
- <https://www.playstation.com> (8)

⁶<https://marketplace.xbox.com/es-ES/>

⁷<https://store.playstation.com>

⁸<http://store.steampowered.com/>



1. Osciloscopio con el juego *Tennis for two* (1958)



2. Máquina recreativa con juego *Computer Space* (1971)



3. Consola Magnabox Odyssey (1972)



4. Carátulo juego para pc *Prince of Persia* (1989)



5. Consola portátil Atari Lynx (1989)



6. Consola PlayStation 1 de Sony (1994)



7. Consola Wii de Nintendo (2006)



8. Gafas de realidad virtual de Sony (2016)

Figura 3.2: Compilación de soportes para videojuego representativos

La distribución digital supone además otras ventajas como la posibilidad de crear nuevos mecanismos que permitan ofrecer mayor resistencia a la piratería. También permite reducir en gastos de fabricación, lo que redundaría en un **aumento del margen de beneficios para los desarrolladores** al eliminarse intermediarios de la cadena. Según [LBV16] en EEUU desde 2014 las ventas asociadas al modelo de distribución digital superan al tradicional (suponiendo el 56 %). También se pone énfasis en la **interacción entre jugadores**, poniendo a disposición espacios específicos como la comunidad Xbox Live⁹, donde los usuarios pueden compartir la experiencia de juego, contrastar logros y comunicarse. Esto ha supuesto un cambio de enfoque en cuanto a la forma de acceder y disfrutar los juegos, así como de

⁹<http://www.xbox.com/es-ES/live>

producirlos y rentabilizarlos.

Como última anotación, cabe destacar un fenómeno fundamental que supone otro cambio significativo en el avance de la industria y en el camino que ésta tiende a seguir es el carácter aperturista que se ha impuesto en cuanto a **público objetivo**. Si en los noventa lo habitual era dirigirse a un público más específico (no por ello escaso) amante de los videojuegos y más metido en este contexto, en la actualidad se ofrecen otras alternativas que han permitido una expansión. Ahora los juegos son una opción válida tanto para acérrimos jugadores como para **jugadores casuales** que sin necesidad de ser seguidores pueden encontrar alternativas específicas de juego que se adapten a sus gustos. Desde los simuladores que enganchan a un público más infantil hasta juegos en grupo para toda la familia, a juegos orientados a actividades saludables o hobbies como el deporte, el canto o el baile (para lo cual Nintendo abrió la veda con el lanzamiento de la consola Wii) hasta alternativas como la que ofrecen los juegos de realidad virtual, que por novedosas llaman la atención de público diverso.

En la adopción de los juegos como algo común y extendido ha ayudado el hecho de que haya generaciones que ya han crecido con ellos y que por tanto los perciben como una alternativa más de ocio de forma natural. Pero además de este factor, existe otro hecho importante que ha favorecido que millones de personas tomen contacto diariamente con los juegos: la irrupción del smartphone. Este ha supuesto un medio a través del cual los juegos se han introducido irremediamente en los bolsillos de millones de usuarios. **Los juegos para móvil constituyen toda una vía complementaria** en la industria que ha favorecido su popularización, crecimiento y masificación.

Evolución y tendencias de los juegos para móvil

La aparición del móvil como nuevo soporte donde poder ejecutar juegos supuso un nuevo hito en lo que a videojuegos se refiere, en tanto que ha permitido abarcar un **nuevo nicho de mercado**, favoreciendo la familiarización de un público mayor y más diverso. Se ha definido así un nuevo espectro de personas a las que llegar, no interesadas previamente en el concepto clásico de videojuego ni en las consolas y que sin embargo son susceptibles de ser consumidoras de los juegos para móvil.

Algunos de los factores clave que explican esta apertura están asociados a la forma en que se produce la aproximación indirecta al potencial jugador y en la motivación que estos puedan tener para jugar juegos de móvil. Respecto a la aproximación, puede considerarse que se produjo una inversión en cuanto a la forma en que el jugador entraba en contacto con el juego. Si bien previamente era el jugador el que hacía el esfuerzo de adquirir el juego, para el caso de los primeros juegos para móviles la recepción se producía de forma pasiva. El usuario adquiría el terminal para otros fines y se encontraba con una funcionalidad extra llamativa, que aun no siendo buscada inicialmente, resultaba tentador probar al venir incluida. De alguna forma puede decirse que **los juegos se colaron en el bolsillo** de los usuarios.

La inclusión de juegos en los móviles se explica por la pugna mantenida por los fabricantes por incluir funcionalidades y utilidades llamativas a medida que se comenzaron a fabricar móviles asequibles al gran público y la potencia de los nuevos terminales lo iba permitiendo, con la finalidad de llamar la atención de los potenciales compradores e incrementar las ventas, estableciendo un pulso con estos y explorando distintas vías para ver cuáles tenían mejor recepción y en qué sentido iría evolucionando el mercado. La precursora de este movimiento fue Nokia, primera empresa fabricante en incluir un juego, concretamente el clásico **juego de la serpiente para el Nokia 6110** en el año 1997. Este era un juego en pantalla monocromática, de dinámica sencilla y rescatado de 1976 cuando apareció por primera vez para máquina recreativa. Y sin embargo esa sencillez consiguió sentar un precedente y cosechar un tremendo éxito. Se estima que pueden haberse distribuido hasta 2000 millones de móviles¹⁰ con ese juego pre-instalado.

Los juegos **arcade clásicos fueron rescatados** en esta primera etapa a finales de los noventa, como si el espíritu de estos hubiese resurgido a través del móvil y gracias a él hubiesen sido redescubiertos. A partir de entonces se esperaba que todo dispositivo móvil incluyese al menos un juego. La razón del éxito de la unión móvil-juego, siendo juegos sencillos y ya conocidos, radicaba en parte en el hecho de que en las primeras etapas la motivación para jugar era diferente a la de los juegos tradicionales. El interés no era jugar al juego propiamente dicho, sino emplear el **juego como herramienta de distracción** en ciertos contextos o para cubrir momentos del día en que el dueño tenía su capacidad de acción limitada por el contexto, pero podía entretenerse por esta vía. Muchas personas a las que no compensaba llevar a cuestas un dispositivo específico para jugar como una consola portátil, descubren así las ventajas de poder jugar de forma cómoda e instantánea a un juego en cualquier momento.

Desde la aparición del juego de la serpiente, la evolución de los juegos para móvil ha ido ligada a la propia evolución de los teléfonos móviles. A continuación se listan algunos de los hechos o lanzamientos significativos ocurridos a partir del año 2000 para ilustrar puntos importantes de esta evolución, apoyado en el artículo *A Brief History of Mobile Games*¹¹ elaborado por Chris Wright y en la infografía *The evolution of Mobile Gaming*¹² elaborada por el sitio web Tombola. En la figura 3.3 se incluye una compilación con algunos de los juegos destacados y dispositivos nombrados¹³.

¹⁰<http://www.xatakamovil.com/futuro/nokia-y-la-convergencia>

¹¹<http://www.pocketgamer.biz/feature/10618/a-brief-history-of-mobile-games-introduction/>

¹²<https://www.tombola.co.uk/evolution-of-mobile-gaming/>

¹³Fuentes: - <https://johnflux.wordpress.com/2015/05/02/nokia-6110/> (1)

- <http://www.pocketgamer.biz/> (2,3,5)

- <http://java.mob.org/game/siberian.strike.html> (4)

- Wikipedia (6)

- <http://www.abdgames.com/doom-rpg/> (7)

- <http://appmoviles.net/iphone-1-caracteristicas-y-especificaciones-tecnicas/> (8)

- <http://phonesdata.com/es/smartphones/htc/dream-397/> (9)

- http://java.mob.org/game/angry_birds_mult.html (10)

- Google Play (11)



1. Nokia 6110 con el juego Snake (1997)



2. Sorcery y Alien Fish Exchange alojados en servidor (2000)



3. Space Invaders vendido físicamente (2002)



4. Siberian Strike para entorno J2ME (2002)



5. Jamdat Bowling para plataforma BREW (2002)



6. N-Gage de Nokia (2003)



7. Doom RPG, shooter primera persona (2005)



8. Iphone 1, primer smartphone (2008)



9. HTC Dream, primero con Android (2008)



10. Angry Birds (2009), primero en conseguir millones de descargas



11. Ingress (2012), con geolocalización y realidad aumentada

Figura 3.3: Compilación de juegos para móvil y dispositivos móviles históricos

- **Juegos pre-instalados.** Inicialmente y durante varios años los móviles incluían de fábrica un juego pre-instalado. No había posibilidad de adquirir e introducirles nuevos juegos. Estos juegos evolucionaron a medida de las posibilidades técnicas del dispositivo, pero inicialmente se rescatarían clásicos como *Tetris*, *Space Impact*, *Snake*, etc.

- **Navegador WAP.** El Nokia 7110, lanzado en 1999, fue el primero en incluir un navegador que emplease el protocolo Wireless Application Protocol (WAP), gracias al cual se podía realizar la conexión con una red inalámbrica, y por tanto permitía la conexión a Internet. Aun si en ese momento los escasos servicios ofrecidos en Internet y la limitada capacidad de visualización de los móviles no permitía explotar esta característica, la creación e implementación del protocolo constituiría un punto clave para que el móvil haya podido llegar a evolucionar hasta el concepto actual de smartphone y para que los juegos fuesen un paso más allá.
- **Introducción del paradigma cliente-servidor.** Gracias a la introducción del protocolo WAP los juegos dejaban de estar limitados a los contenidos grabados en la memoria del móvil de fábrica. Dada la capacidad de poder conectarse a Internet, comenzaron a desarrollarse juegos alojados en servidores con los cuales establecía comunicación el frontend instalado en el móvil. En el año 2000 ya aparecieron dos juegos que respondían a estas características: *Sorcery*, un juego basado en texto al estilo del clásico *Adventure* para ordenador (1976) en el que se enviaba la elección del usuario al servidor y se recibía la continuación de la historia; y *Alien Fish Exchange*, basado en la crianza de diferentes tipos de peces que podían venderse e intercambiarse, ejerciendo el servidor el rol de tienda mediante el cual llevar a cabo dichas acciones.
- **Primer intento de monetización.** Las compañías publicadoras y desarrolladores de juegos para móvil veían una oportunidad de negocio en este tipo de juegos para móvil basados en el paradigma cliente-servidor. Sin embargo hasta el momento las únicas que se lucraban de ello eran las operadoras, facturando gracias a los datos consumidos en la conexión a Internet por parte de los jugadores. En 2001 se llevó a cabo un primer e infructuoso intento, estableciéndose un acuerdo entre la operadora Vodafone y la desarrolladora Vivendi Universal, por el cual se crearía un portal donde Vivendi ofrecería sus contenidos y Vodafone se llevaría el 50 % de los beneficios, aunque finalmente no resulto rentable.
- **Juego como aplicación descargable.** En 2002 se consiguió introducir un entorno Java para la ejecución de aplicaciones en un prototipo de un móvil Motorola. Esto agregaba potencial al móvil desde el punto de vista de la ejecución de juegos. De esta forma ya no se requería que esta se produjera en el servidor, sino descargar el juego desde el servidor inicialmente y ejecutar este en el móvil, ya sin necesidad de conexión. Esto suponía una mejora respecto a los juegos que empleaban WAP al permitir la ejecución de animaciones, sonidos y una interactividad más fluida, lo que llevaría a que se diese paso al desarrollo de juegos arcade de acción más dinámicos que los que se habían lanzado hasta el momento.

La introducción de estos entornos permitió que la compañía Digital Bridges experimentara vendiendo juegos para móvil de forma física, a través de unas cajas que incluían un pin único. Introduciendo el pin en la url facilitada comenzaba la descarga del juego. Así por primera vez el jugador tenía la capacidad de escoger qué juego quería ejecutar en su móvil.

Sin embargo este entorno no mantendría la hegemonía y se desarrollarían otras plataformas específicas como BREW (por Qualcomm), ExEN (por In-Fusio) o Mophun (por Synergenix). Juegos representativos fueron *Siberian Strike* (desarrollado por Gameloft) para el entorno J2ME y *JAMDAT Bowling* un juego de bolos para los dispositivos que incluyesen BREW, no asociado a una marca específica fabricante de móviles, sino incluido en aquellos móviles cuya línea estaba asociada a la operadora Verizon.

- **N-Gage.** El 2003 fue el año en que se adoptó de forma más extendida la pantalla a color, característica que permitía hacer más atractivos los juegos. Pero si se trata de iniciativas llamativas, se ha de comentar la fuerte apuesta de Nokia por el juego: aunar en un mismo dispositivo un teléfono móvil y una videoconsola portátil. La idea subyacente era establecerse como competidora de las consolas portátiles ofreciendo una funcionalidad extra al llevar el teléfono integrado. Pese a que resultó ser un fracaso de ventas, la propuesta por ambiciosa, merece ser mencionada. El problema radicó en que siendo un dispositivo con dos facetas el diseño que se propuso a nivel hardware no supo unificarlas. No resultaba cómodo como teléfono por su tamaño y distribución de elementos (teclas, micrófono, altavoz). Como consola, la pantalla vertical seguía recordando a los juegos para móvil, el catálogo de juegos distribuidos en tarjetas de memoria MMC localizadas bajo la batería era limitado y ofreciendo un manejo algo incómodo hicieron que no pudiera postularse como competidora de otras consolas, como la PSP de Sony, anunciada ese mismo año y que sí que generaría mucha expectación.
- **Estancamiento.** El status quo de la industria se mantuvo durante unos años, hasta 2008, avanzando en aspectos como los gráficos 3D o en el desarrollo de juegos multijugador, entre otros. Juegos de éxito desarrollados durante estos años fueron: *Bejeweled*, juego tipo puzzle donde el objetivo era unir piezas del mismo color y con capacidad multijugador; *The fast and the furious*, juego de carreras de coches en 3D; *Downtown Texas Hold'em*, juego de póker y *Doom RPG*, un *shooter* en primera persona. Otro factor característico fue la necesidad de lidiar con los problemas de interoperabilidad, dadas las diferencias entre los distintos dispositivos que iban saliendo al mercado, que obligaba a las desarrolladoras a crear distintas versiones del mismo juego. Existían en aquel momento tres agentes involucrados: las compañías desarrolladoras, las operadoras y los fabricantes. Esa triada era liderada por las operadoras, ya que la interfaz para la descarga era gestionada por estas y un jugador solo podía acceder a los contenidos ofrecidos por su operadora de Internet.

- **Iphone.** La deriva que tomaría del mercado de móviles se vería reflejada en 2008 con la propuesta de Apple. Lanzó al mercado su propio terminal en 2007 junto con su propio sistema operativo, el iOS, sentando un precedente en cuanto al diseño y al tipo de interacción (a través de pantalla táctil) y un año después habilitaría una plataforma de acceso a aplicaciones, el iTunes App Store. Uno de los primeros juegos que destacó en esta fue *Tap Tap Revenge*, un juego musical en que había que tocar puntos en la pantalla al ritmo de la música. Ambos conceptos hacen que el Iphone pueda considerarse como el primer smartphone moderno, el que marcó la pauta de avance y el modelo a seguir del cual son reflejo los smartphones actuales.
- **Android.** Un año antes de la aparición de iOS (sistema operativo solo compatible con dispositivos maufacturados por Apple) se presentaba con el respaldo de Google el sistema operativo Android, un sistema abierto bajo licencia apache, diseñado para correr en tablets, teléfonos y relojes inteligentes. En 2008 se lanzaría el primer teléfono móvil con este sistema operativo integrado, el HTC Dream. Posteriormente muchos otros fabricantes pasarían a incluir Android en sus smartphones, siendo el primero de ellos Samsung. De igual forma que Apple, se lanzó en 2008 una plataforma para el acceso a aplicaciones denominada Android Market, que en 2009 permitiría la inclusión de aplicaciones de pago. En 2012 se remodelaría y pasaría a denominarse Play Store.
- **Markets como modelo de negocio.** Desde que aparecieran las tiendas de aplicaciones, estas se han asentado como la opción preferente para establecer el nexo entre jugadores y desarrolladores de juegos para móviles. Resulta una opción cómoda para los jugadores y satisfactoria para los desarrolladores desde el punto de vista de la monetización. A través de esta los usuarios pueden adquirir y descargar apps fácilmente. Por su parte, una desarrolladora puede acceder directamente a los jugadores eliminándose la influencia de publicadoras y operadoras, supeditándose solo a las condiciones específicas del *market*. De hecho, gracias al cambio de modelo de distribución y según el informe [EGD16] elaborado por EGDF, el margen de beneficios al que aspira el desarrollador se ha incrementado de un 25 %/50 % según la región a un 70 %.

Existen *markets* dedicados para un sistema operativo concreto, como es el caso de Google Play¹⁴ y iTunes App Store¹⁵. También existen *markets* de terceros, donde destacan Amazon Appstore¹⁶, Opera Mobile Store¹⁷ y Facebook Games. Como muestra del éxito y de la capacidad de estas plataformas para llegar a una cantidad ingente de público, cabe mencionar el éxito cosechado por juegos como Angry Birds (2009) o Candy Crush Saga (2012) cuyas versiones actuales acumulan un total de hasta 500 millones de descargas y 1000 millones de descargas respectivamente.

¹⁴<https://play.google.com/store>

¹⁵<https://itunes.apple.com/es/genre/ios/id36?mt=8>

¹⁶<https://www.amazon.es/mobile-apps/b?ie=UTF8&node=1661649031>

¹⁷<http://html5.oms.apps.opera.com/es.es/>

Se reflexionaba inicialmente acerca del rol pasivo de los primeros usuarios de móvil, que recibían sin más un juego pre-instalado en el dispositivo. Gracias a los *markets* **el rol ha vuelto a invertirse**. Ahora se dispone de un considerable catálogo de juegos, en el cual el potencial jugador puede explorar y escoger por propia iniciativa aquellos que le resulten más afines. Incluso existen consumidores que priorizan al adquirir un smartphone el hecho de que tenga las suficientes prestaciones como para poder ejecutar los últimos juegos salidos al mercado.

Dado que desde el punto de vista estructural y de negocio se estableció una solución afín a los jugadores y a las compañías desarrolladoras, los avances en la última etapa no han venido de la mano de remodelaciones del mercado. Se volvería a centrar el foco en nuevos desarrollos a través de los cuales ofrecer **nuevas experiencias de juego**, aprovechando las nuevas tecnologías integradas en los smartphone. En ese sentido, algunas de las características que han entrado en juego han sido el empleo de sensores (sobre todo el **acelerómetro**, que permite introducir una nueva forma de interacción), la capacidad de **geolocalización** y la **integración de la cámara**.

Una de las propuestas más innovadoras fue el lanzamiento de *Ingress*, cuya dinámica de juego implicaba visitar localizaciones del mundo real, lo que requería la geolocalización, incorporando además **realidad aumentada**. Esta primera propuesta, lanzada en 2012 por Niantic, asentó las bases para el lanzamiento al mercado en 2016 de un juego con una dinámica parecida, pero contextualizando su acción en torno a la historia de un juego icónico que ya en su primera versión lanzada en 1998 sería un éxito de ventas para la consola Game Boy Color: *Pokemon*. Se ofrece así una experiencia de juego innovadora donde el jugador pasa a realizar en el mundo real las acciones que se emulaban en el juego original, rompiendo en cierta medida la cuarta pared. Esta apuesta innovadora se ha visto respaldada por los jugadores, convirtiéndose en un éxito a nivel mundial, acumulando únicamente en el Play Store más de 100 millones de descargas.

De acuerdo a la revista Forbes¹⁸ **los juegos para móvil se reconocen como una de las tres tendencias** que marcan actualmente (2016) el camino a seguir en la industria del videojuego, junto con las consolas de nueva generación y la realidad virtual. Junto con la realidad aumentada, como se recoge en [LBV16] existen otras áreas que ya afectan al desarrollo de los juegos en mayor o menor medida, como es el caso del **Big Data** como herramienta para la generación de inteligencia que pueda ser volcada en el diseño, el caso del **Cloud** para poder almacenar información deslocalizada y poder portar la partida entre diferentes dispositivos o la integración con el **IoT** enriqueciendo la interacción a través de la incorporación de nuevos elementos.

¹⁸<https://goo.gl/b5dc18>

Impacto económico

El ecosistema de los juegos para móviles ha derivado en una coexistencia de juegos de distinta naturaleza, donde tienen tanta aceptación juegos clásicos que siguen resultando adictivos, como propuestas innovadoras que van dando pistas del futuro de la industria, a juegos más potentes que tratan de seguir la estela de las videoconsolas portátiles o juegos pensados para una interacción multijugador. Esto da pie a que **tanto pequeñas desarrolladoras como grandes compañías puedan adherirse al negocio.**

Para caracterizar la situación actual de la industria del videojuego desde el punto de vista del sector de los juegos para móvil, conviene recurrir a las cifras. A tal efecto se extraen algunos datos significativos. También extraído de *Essential Facts* se desprende, según su estimación, que del total de jugadores aproximadamente la mitad juegan a juegos de PC y consola (56 % y 53 % respectivamente) ocupando el tercer puesto los juegos para smartphone, al cual **le dedican tiempo un tercio de los jugadores (36 %)**. La influencia de las videoconsolas portátiles parece haberse visto reducida, ya que solo un 17 % de los jugadores le dedicarían tiempo a esta alternativa. Esto denota que en la pugna por ocupar el ocio portátil se está imponiendo el smartphone a las consolas portátiles dedicadas.

En concordancia, NewZoo, empresa dedicada a la generación de inteligencia de negocio especializada en juegos, ha elaborado el informe *2016 Global Games Market Report*¹⁹ donde se estima que del total de ingresos generados en la industria para finales de año, el 37 % del total se corresponderán a las **ganancias generadas por juegos para móvil y tablet, aportando casi 37 mil millones de dólares**, y experimentando así una tasa de crecimiento respecto al año anterior del 23.7 %. Si se atiende concretamente a los ingresos generados diariamente, las cifras hablan por sí solas. En la figura 3.4²⁰ puede observarse cómo existen apps de juego capaces de superar la barrera del millón de dólares diarios ingresados. Según NewZoo, si se atiende al interés generado para el sector del mercado *mobile* de videojuegos según un criterio geográfico, la región de Asia-Pacífico representa más de la mitad de este mercado, ya que solo China genera 6.500 millones de dólares. Por su parte, donde se ha experimentado un mayor crecimiento es en Africa-Oriente con un 83 % y en Latinoamérica, con un 73 %.

Si se compara el dinero generado por las apps de juegos respecto a las restantes apps a nivel global durante el año 2015, se obtienen datos concluyentes. Según se recoge en [LBV16] y de acuerdo a los datos recopilados por la consultora móvil App Annie, **los consumidores gastaron en total 41.100 millones de dólares en apps descargadas a través de markets digitales** considerando pago por descarga, ingresos por IAP y suscripciones (sin incluir publicidad). Del total, el 85 % del gasto global fue asumido por los videojuegos, pese a que en volumen solo representaban el 41 % del total de aplicaciones.

¹⁹<https://goo.gl/1goq5K>

²⁰Fuente:<http://dadaviz.com/i/4249/>

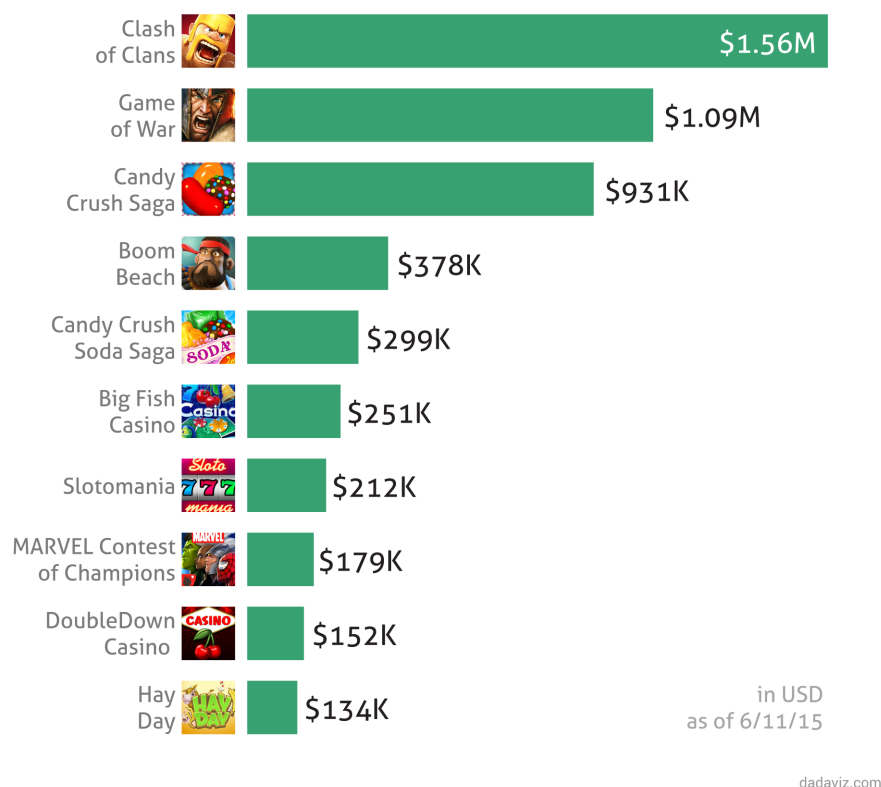


Figura 3.4: Top de juegos por ingresos diarios generados

Supercell como caso de éxito Como caso paradigmático de éxito cabe destacar los resultados obtenidos por la empresa Supercell, que con una plantilla de solo 180 empleados consiguió generar en 2015 unos beneficios de 848 millones de euros²¹ gracias a sus juegos para iOS *Hay Day*, *Clash of Clans* y *Boom Beach*. Estos resultados no son casuales. Supercell cuenta con equipos de analistas que extraen información útil de los gustos y preferencias de sus usuarios en base a los datos recopilados sobre hábitos de juego, lo que les ha permitido mantener sus juegos en los top de recaudación durante varios años.

Deriva de compañías con recorrido Prueba de que este resulta ser un sector en auge es que compañías que tradicionalmente se han dedicado al sector del videojuego en otras plataformas apuesten por introducirse también en la rama del juego para móvil. Ese es el caso de Electronic Arts, que creó la empresa subsidiaria **EA Mobile** en 2004. Más recientemente está el caso de **Nintendo**, que en asociación con la empresa DeNA ha lanzado en marzo de 2016 *Miitomo*²², su primera app a medio camino entre el juego y una app social, que permite crear un avatar y comunicarse con otros usuarios. Así mismo, tienen previsto lanzar 4 juegos más y dos de ellos asociados a dos títulos de éxito en su consola, *Fire Emblem* y *Animal Crossing*. Otro caso significativo es el de **Konami**, que siendo propietaria de títulos

²¹goo.gl/BSmraK

²²<https://goo.gl/a1XWcz>

tan relevantes para consola como la saga *Metal Gear*, *Silent Hill* o *Pro Evolution Soccer*, anunciaron en 2015 una fuerte deriva hacia el sector de los móviles, asegurando que este sería su principal plataforma ya que es ahí donde se encuentra el futuro de la industria. De 2015 a 2016 han logrado de hecho unos buenos resultados económicos gracias a juegos como *Jikkyou Pawafuru Puroyakyu 2016*²³, un juego de béisbol que ha alcanzado los 20 millones de descargas.

Situación en España Centrando la atención en España desde el punto de vista del negocio de los videojuegos, es significativo el hecho de que la mayor parte de las empresas que están destacando estén dedicadas concretamente a la plataforma móvil. Si se atiende a los datos recogidos en el Libro Blanco del Desarrollo Español de los Videojuegos (véase [LBV16]) se puede determinar la situación actual de la industria. Existen un total de **480 empresas en este sector**, habiéndose experimentado un crecimiento del 20 % respecto al año anterior, en cuyo ejercicio se generaron 510.7 millones de euros. Puede considerarse un **sector en expansión y joven**, en tanto que el 85 % de estas empresas fueron fundadas hace menos de 10 años. Analizando los datos en conjunto se aprecia que en los últimos años el sector **ha crecido gracias a la aparición de pequeños estudios independientes** (el 71 % son microempresas), animados principalmente por el boom a nivel mundial de los videojuegos para dispositivos móviles y la aparición de plataformas de distribución digitales que facilitan el acceso. En [LBV16] se incluye una lista de empresas destacadas, estando **la mitad de ellas dedicadas al desarrollo en plataforma móvil**, entre ellas Digital Legends, Genera Games, Kerad Games o Social Point. No en vano, la intensa competencia existente en estos *markets* y la aparición de motores como Unity o Unreal Engine hacen pensar que en un futuro puede verse incrementado el volumen de desarrollos dedicados a dar salida a juegos *indie* para las consolas de nueva generación.

Monetización

Parte de la capacidad de esta rama para generar beneficios se debe a los distintos modelos de explotación que se han creado, gracias al dinamismo que introduce el soporte de venta digital. Así, a la hora de **rentabilizar una app**, o un juego en este caso, es necesario conocer y escoger la estrategia que mejor se ajuste, según la naturaleza de la propia app y el tipo de público al que vaya dirigida, antes de publicarla en una plataforma de descargas. A tal efecto se revisan a continuación las diferentes alternativas para monetizar un juego, extraídas de la guía *The App Developer Business Kit*²⁴ elaborada por AdMob, compañía dependiente de Google dedicada a la monetización de aplicaciones:

²³<https://www.konami.com/games/corporate/en/news/release/20160428/>

²⁴<https://goo.gl/COZSYL>

- **App de pago.** Esta alternativa responde al enfoque tradicional de consumo. Se ofrece la aplicación como producto. De esta forma, se impone un precio para poder descargar, instalar y jugar al juego completo. La alternativa de uso previo pago requiere que la aplicación sea lo suficientemente útil y única como para que los usuarios tengan la confianza suficiente para adquirirla. Si existen alternativas gratuitas resultará muy difícil conseguir un número significativo de descargas. La ventaja de esta opción es que la aritmética es más clara y sencilla. Los beneficios son directamente proporcionales al número de descargas del juego, aunque hay que tener en cuenta que un porcentaje de cada compra será retenido por el *market* a través del cual se realice la descarga.
- **App gratuita.** Conocido como Free to play (F2P) para el caso concreto de los juegos, este modelo es una opción con una aritmética más compleja, pero que permite acercarse al usuario. Consiste en ofrecer la posibilidad de probar la aplicación de forma gratuita, agregando así un aliciente para la descarga. De esta forma, la monetización se constituye como un proceso de dos pasos, llegando primero al mayor número de receptores posibles y una vez se ha llamado su atención, instaurando el mecanismo a través del cual extraer el beneficio. A tal efecto existen diferentes vías:
 - **Dos versiones, de prueba y de pago.** La alternativa más sencilla consiste en generar dos versiones de la aplicación, una gratuita de prueba y otra completa. La idea es que el potencial jugador pueda probar una demo y constatar si el juego le gusta lo suficiente como para querer tener acceso a todas las funcionalidades. Esta opción de pago resulta solo útil en caso de estar seguros de que la aplicación tiene la calidad suficiente y la capacidad de engachar al jugador como para querer desbloquear las partes inaccesibles pagando por ello. En este caso la métrica es similar a las aplicaciones de pago, resultando de interés en cuanto a analíticas el índice de conversión, es decir, cuántas personas que accedieron a la versión de prueba finalmente descargaron la versión de pago.
 - **Inserción de anuncios.** Se trata de implantar una solución en la app que permita la inserción de anuncios de terceros. De esa forma, el jugador tiene acceso al juego completo a cambio de visualizar anuncios en ciertos momentos durante la experiencia de juego (por lo general a través de un banner, anuncios intercalados o integrados en el juego). El beneficio se obtiene por parte de los anunciantes, estableciéndose una cuota a pagar por cada ocasión en que un jugador pincha en el anuncio y visita el sitio del anunciante o bien se descarga la aplicación anunciada, en caso de que ese sea el producto anunciado. Existen redes de publicidad que permiten conectar anunciantes con desarrolladores, como es el caso de Admob de Google²⁵, Appodeal²⁶ o Chartboost²⁷, esta última específica para juegos.

²⁵<https://www.google.com/admob/>

²⁶<https://www.chartboost.com/>

²⁷<http://www.appodeal.com/>

- **Publicidad incentivada.** Un enfoque inspirado en la inserción de anuncios, pero de una forma más amigable para el usuario, consiste en llegar a acuerdos de colaboración con marcas para que estas ofrezcan promociones y descuentos. Así, ante ciertos hitos logrados en la aplicación, estas ofertas son desbloqueadas, y de esta forma, lo que para los usuarios es una recompensa para los anunciantes resulta ser una vía de promoción. Esta modalidad está empezando a ser introducida sobre todo en aplicaciones de para hacer deporte y fitness.
- **Freemium.** Otra opción es ofrecer una versión completa del juego pero incorporando la posibilidad de desbloquear opciones, suprimir limitaciones u ofrecer contenidos que mejoren o complementen la experiencia de juego. Las opciones habituales pasan por habilitar compras de bienes virtuales dentro de la propia aplicación, requerir una suscripción para acceso a contenido exclusivo u ofrecer ventajas que mejoren la experiencia como la eliminación de anuncios o un incremento en el espacio de almacenamiento. Se pretende generar a través del juego una necesidad virtual en el jugador, para lo cual es necesario en primera instancia que el juego haya conseguido engancharle.

De entre las dos alternativas **la tendencia es a apostar por la descarga gratuita**. En la plataforma Google Play, a mediados de 2015 tan solo el 31.2 % de aplicaciones eran de pago, las restantes eran gratuitas²⁸. Para el caso del Apple Store, se tienen los datos recogidos en 2014 en los informes *Unveiling the Secrets behind App Store Category Dynamics*²⁹ y *How the Most Successful Apps Monetize Globally*³⁰ elaborados por la compañía Distimo:

- Para todos los países la aportación a los ingresos totales generados por todas las aplicaciones es siempre mayor por parte de aplicaciones F2P con IAP que por parte de aplicaciones de pago. Para el país donde está más ajustado, Alemania, el 70 % de los ingresos son generados por IAP. En el extremo contrario está China, generándose por ese medio el 92 % de los ingresos. Estos datos lo que reflejan es la preferencia en la práctica de los desarrolladores por el modelo IAP.
- Si se enfoca el análisis por categorías de aplicaciones a nivel global, las de juegos son las que presentan un porcentaje mayor con un 92 % de ingresos por parte de aplicaciones gratuitas con IAP, solo superadas por aplicaciones de noticias.
- Sin embargo lo anterior no denota que las alternativas freemium resulten más rentables. De hecho en el informe se analiza como métrica el ARPD, que alude a los ingresos medios generados por descarga. En relación a los distintos modelos de explotación, se concluye que en el Apple Store, las aplicaciones de pago generan un ingreso mayor por descarga que las aplicaciones freemium.

²⁸<https://www.statista.com/statistics/266211/distribution-of-free-and-paid-android-apps/>

²⁹<https://goo.gl/nhanp4>

³⁰<http://www.lateledipenelope.it/public/54c9162c2ddfc.pdf>

El hecho de que se tienda a ofrecer aplicaciones gratuitas, aunque a nivel de aplicación la descarga pueda resultar más rentable si se ofrece de pago, responde a **la dificultad de llegar a los usuarios** en un mercado cada vez más saturado de aplicaciones. El primer paso para rentabilizar una app pasa por conseguir que los usuarios la descarguen. Manejándose dos variables como son número de descargas e ingresos por descarga, puede resultar más interesante tener un beneficio menor por descarga compensándolo al **maximizar el número de descargas** (enfoque freemium) que arriesgar a establecer un precio inicial que sea una barrera y limite las descargas.

Es importante considerar que lo usual es que dado un gran volumen de jugadores que descarguen el juego, una cantidad considerable de ellos perderán el interés en la aplicación en menos de 1 mes. De los restantes jugadores, que pueden considerarse activos, **un porcentaje mínimo será el que realmente contribuya a generar ingresos**, según la compañía Apptentive tan solo un 2.2 %³¹. Por lo que se necesita un gran volumen de descargas para que ese pequeño porcentaje resulte en una cifra significativa. Así pues, resulta importante esforzarse inicialmente por incrementar este parámetro, poniendo atención a la estrategia para anunciar y promocionar el juego, hacer que destaque respecto a otros similares y conseguir que se posicione en las recomendaciones de las principales plataformas de descargas. Una vez se ha llegado al usuario, el esfuerzo debe ir orientado a definir un diseño o estrategia a través de la cual **lograr la mayor conversión** posible. Lo que es más, si se invierte en medios para promocionar el juego, a cada descarga conseguida se habrá destinado una pequeña parte de esa inversión. Si la descarga no llega a generar beneficios como para cubrir al menos ese coste, se estarían generando pérdidas.

Un parámetro asociado al favorecimiento de la conversión, tanto para el enfoque de inserción de anuncios como para el modelo de compras integradas, y otra de las claves para incrementar el beneficio es **conseguir retención**, prolongar lo máximo posible las semanas que el jugador sigue dedicando atención al juego. Es necesario destinar esfuerzos a aumentar ese compromiso con el juego, ya que por lo que se ha explicado previamente, un jugador fidelizado puede generar mayores ingresos que 10 nuevos jugadores que solo descarguen el juego para probarlo. Existen numerosas métricas para medir la retención y el comportamiento y hábito de juego de los jugadores, algunas de ellas recogidas por SurveyMonkey Intelligence³² son:

- Ratio entre usuarios diarios activos y usuarios mensuales activos.
- La media del número de días que los usuarios usan la aplicación al mes.
- La media del total de sesiones diarias entre el total de usuarios únicos activos diarios.
- El tiempo diario invertido en la aplicación, multiplicando el tiempo medio de duración de la sesión por la media del número de sesiones por usuario por día.

³¹<https://goo.gl/LFL9P3>

³²<https://www.surveymonkey.com/business/intelligence/app-engagement-metrics/>

De esta forma puede detectarse que para ciertas aplicaciones, los usuarios acceden casi todos los días de la semana, pero durante un breve lapso de tiempo, caso típico de las aplicaciones meteorológicas. Sin embargo para otras apps, puede que el usuario dedique largas sesiones pero una o dos veces a la semana.

La retención es la clave de éxito, especialmente para ciertos tipos de juego, como es el **caso de los MMO**, ya que la experiencia de juego que perciben los usuarios depende directamente de la existencia de otros jugadores que den vida al entorno multijugador. A tal efecto se llevan a cabo indagaciones desde un punto de vista analítico para tratar de determinar los factores que favorecen la retención en este contexto. Como muestra, la agencia SuperData, dedicada a proveer inteligencia respecto al mercado de los videojuegos, publicó en 2015 datos de un estudio³³ elaborado a lo largo de cinco años, recopilando datos de diversas publicadoras, desarrolladoras y proveedores de servicios de pago respecto al comportamiento de los jugadores de diferentes juegos MMO del tipo F2P. De las conclusiones derivadas del estudio se extrajeron dos patrones: el 6.21 % de los jugadores que prueban el juego durante el primer mes de lanzamiento, siguen jugando al juego durante el siguiente año. Sin embargo solo el 0.63 % de jugadores que probaron el juego tras el primer mes de lanzamiento y durante el primer año continúan dedicándole tiempo a lo largo del primer año. Así pues, parece importante tratar de llamar la atención del mayor número posible de jugadores antes del lanzamiento del juego. Por otra parte, 24 meses es el periodo de prueba y estabilización de un juego. Si consigue superar ese tiempo, se habrá labrado una buena reputación y gracias al boca a boca llegarán nuevos jugadores de valor.

Del ejemplo anterior se desprende que **segmentar los datos** generados por los usuarios al usar una aplicación redundante en la obtención de información valiosa que, adecuadamente analizada, puede revelar qué está funcionando y qué no está funcionando tanto a nivel de producto como de explotación y medios de promoción. Existen herramientas en el mercado que facilitan el proceso de recolección y agrupación de los datos generados en torno a una aplicación móvil, entre ellas Google Analytics³⁴ y MixPanel³⁵. Algunas de las perspectivas a considerar cuando se realiza un análisis además de la retención son:

- **Actividad.** Más que atender al número de descargas hay que detectar en qué grado se produce actividad tras la descarga. Analizar los datos de las sesiones creadas, desde frecuencia a duración, permite obtener una mejor visión de la atención dedicada.
- **Adquisición.** Dado que normalmente se dedican fondos a la promoción y posicionamiento de la aplicación por distintos canales, es importante medir desde cuál de todos ellos se produce la descarga. Así, puede detectarse si no resulta rentable invertir en alguno de ellos o colocar la aplicación en una determinada plataforma de descargas.

³³<https://www.superdataresearch.com/understanding-mmo-retention/>

³⁴<https://www.google.com/analytics/>

³⁵<https://mixpanel.com>

- **Flujo de uso.** Pueden definirse eventos dentro de la aplicación para medir qué recorrido realizan los usuarios durante el uso. Si se detecta que ciertos procesos se quedan a medio camino, esto podría estar revelando un mal diseño.
- **Ingresos.** Si se han incorporado distintos mecanismos para la generación de ingresos, es importante analizar las cantidades generadas de forma individual por cada uno de ellos, con el fin de comprender cuál es el que mejor está funcionando y suprimir o mejorar aquellos que no estén generando lo suficiente.
- **Datos demográficos.** Es importante determinar a qué público está llegando la aplicación. Las nuevas estrategias de diseño y marketing pueden modificarse si se detecta, por ejemplo, que la aplicación llama la atención de una franja de edad específica o se ha popularizado en una región concreta.

Parece claro que los videojuegos para móviles es un **sector en auge, pero un sector muy competitivo** al existir millones de aplicaciones disponibles. Así pues, para incrementar las posibilidades de éxito es necesario un esfuerzo previo al desarrollo para estudiar la viabilidad o probabilidades de éxito de la propuesta que pretende llevarse a cabo, elaborando además una estrategia de distribución y explotación basada en el conocimiento ya generado acerca del funcionamiento del mercado y de las tendencias que se van observando.

3.1.3 Juegos basados en preguntas

Según los datos recopilados por SurveyMonkey Intelligence respecto al interés generado para distintas categorías de juegos para las descargas en Estados Unidos (2016)³⁶, los juegos basados en preguntas están en las últimas posiciones para métricas como número de descargas, número de usuarios activos y aportación a los ingresos totales generados. Sin embargo esto no tiene por qué implicar necesariamente una falta de interés de los jugadores, sino más bien que **se desarrollan y publican en mayor cantidad juegos de otras categorías**. Como refuerzo a esta idea, de la recopilación *Essential Facts about the Computer and Video Game Industry*³⁷ se desprende que, precisamente, el tipo de juego jugado con mayor frecuencia es aquel que pertenece al grupo de los juegos tipo puzzle, de mesa, cartas y concurso, jugando a estos un 38 % del total de jugadores, seguidos del género de acción y estrategia.

Que haya más juegos de una categoría puede denotar que esta acepta una variabilidad mayor de juego a juego, por lo que favorece el lanzamiento de múltiples alternativas diferentes que tienen la capacidad de diferenciarse de otros juegos de la misma categoría. Si bien **el problema de los juegos de preguntas puede ser que son muy similares** unos respecto a otros, existe una ventaja competitiva por el hecho de situarse en la *long tail*, y es que **no existe tanta oferta**. Así pues, escoger un juego basado en preguntas de tipo show, si se consigue ofrecer algún aliciente extra respecto a los que ya existen puede ser un acierto.

³⁶<https://www.surveymonkey.com/business/intelligence/mobile-game-statistics/>
























³⁷<http://essentialfacts.theesa.com/Essential-Facts-2016.pdf>

Whizz frente juegos competidores

Para el caso de *Whizz*, el juego con el que se ha de colaborar, el principal factor diferenciador respecto al resto de juegos competidores es su **enfoque MMO**, por el que las partidas resultan ser totalmente simultáneas. Pueden crearse partidas grupales sin límite de jugadores, donde todos responden la misma pregunta a la vez, sin turnos ni esperas. Ofrece además la posibilidad de emplear **ataques** (como reducir el tiempo de respuesta, invertir el texto, hacer que la pantalla tiemble o las opciones bailen) y *powerups* que alteran la experiencia de juego de los contrincantes. Todo ello agrega un extra de dinamismo, reforzado por su ambientación y **estética inspirada en los shows televisivos**.

Además como característica única ofrece la posibilidad de crear concursos localizados y de contenido específico, a través de la creación de un **canal con preguntas concretas**. Así, *Whizz* se postula más allá de los límites del juego como herramienta al servicio de entidades que puedan emplearlo como medio de difusión cultural o promocional. Esta supone una vía complementaria para la monetización, ya que la creación de un canal es una opción disponible previo pago. Así mismo, bajo el enfoque F2P los ingresos generados por el juego se generarán a través de las compras integradas en la aplicación y gracias a la inserción de anuncios de terceros y preguntas patrocinadas. Para esto último la ambientación del juego resulta ser un acierto, en tanto que coloca al jugador en un contexto donde le resulta familiar la visualización de anuncios, por lo que estos no entorecen la experiencia de juego.

A parte de *Whizz*, existen otras aplicaciones de naturaleza similar. Se presenta una breve revisión de sus características principales en la figura 3.1, elaborada a partir de la información ofrecida en las páginas oficiales de cada juego y los datos arrojados por SensorTower³⁸.

	Juego	Nº descargas (solo Android)	Nº preguntas	Nº categorías	Partida online	Idioma	RR.SS.	Monetización	Plataforma
	QuizUp	50.000.000	100.000	100	Sí (2 jugadores)	ES/EN		Publi	
	Atriviate	10.000.000	10.000	7	Sí (6 jugadores)	Multi4		Publi	
	Preguntados	500.000.000	90.000	6	Sí (2 jugadores)	ES/EN		Publi/ Pago/ Premium	
	Arruleados	500.000	-	-	Sí (2 jugadores)	ES	No	Fremium	
	Trivia Burst	-	30.000	30	No (1 jugadores)	EN	No	-	
	Trivial Party	50.000	10.000	7	Sí (6 jugadores)	ES		Publi	
	Pregunticas	50.000	60.000	22	Sí (2 jugadores)	ES		Publi	
	Quizoid	5.000.000	-	17	No (1 jugadores)	EN	No	Publi/ Pago	
	Who Bec.Rich	10.000.000	-	-	No (1 jugadores)	Multi6	No	Publi	

Cuadro 3.1: Comparativa de principales juegos de preguntas

³⁸<https://sensortower.com>

Precedente de éxito: Preguntados

De todos los juegos de preguntas revisados, cabe destacar *Preguntados* (*Trivia Crack* en su versión inglesa) por incluir una propuesta similar a la idea que engloba QuPI-GAm, ofreciendo a los jugadores la posibilidad de sugerir sus propias preguntas y que estas pasen a formar parte del juego. Lanzado por la desarrolladora Etermax en 2013, **alcanzó el número uno en descargas** en las principales plataformas y ha conseguido una media de 1 millón de aportaciones diarias³⁹, por lo que puede considerarse como un precedente de éxito bien recibido por los jugadores, lo que refuerza y justifica la utilidad que puede tener el desarrollo que se afronta en este trabajo.

Dispone de una modalidad denominada **‘factoría de preguntas’** cuyo objetivo es involucrar a la comunidad de usuarios permitiéndoles colaborar en el crecimiento y expansión del contenido del juego. A través de la modalidad factoría de preguntas es posible: aportar nuevas preguntas, revisar y validar las preguntas aportadas por otros usuarios, traducir preguntas ya existentes a otros idiomas y consultar el historial de preguntas aportadas o traducidas. La figura 3.5 muestra el aspecto del menú y todas las posibilidades de que dispone.

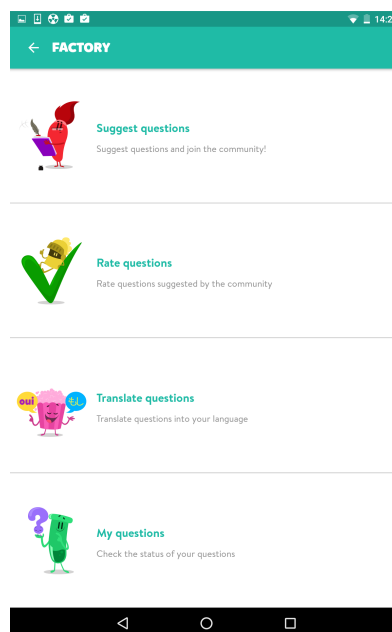


Figura 3.5: Preguntados - Menú del modo Factoría de Preguntas

- La figura 3.6 muestra el aspecto visual de las interfaces que conducen el proceso de **aporte de preguntas**. A diferencia de QuPI-GAm los jugadores también tienen la posibilidad de proponer una traducción para una pregunta subida por otro. Permite la traducción a diferentes idiomas. La figura 3.7 ilustra el proceso traducción.

³⁹<http://goo.gl/HvhsMP>

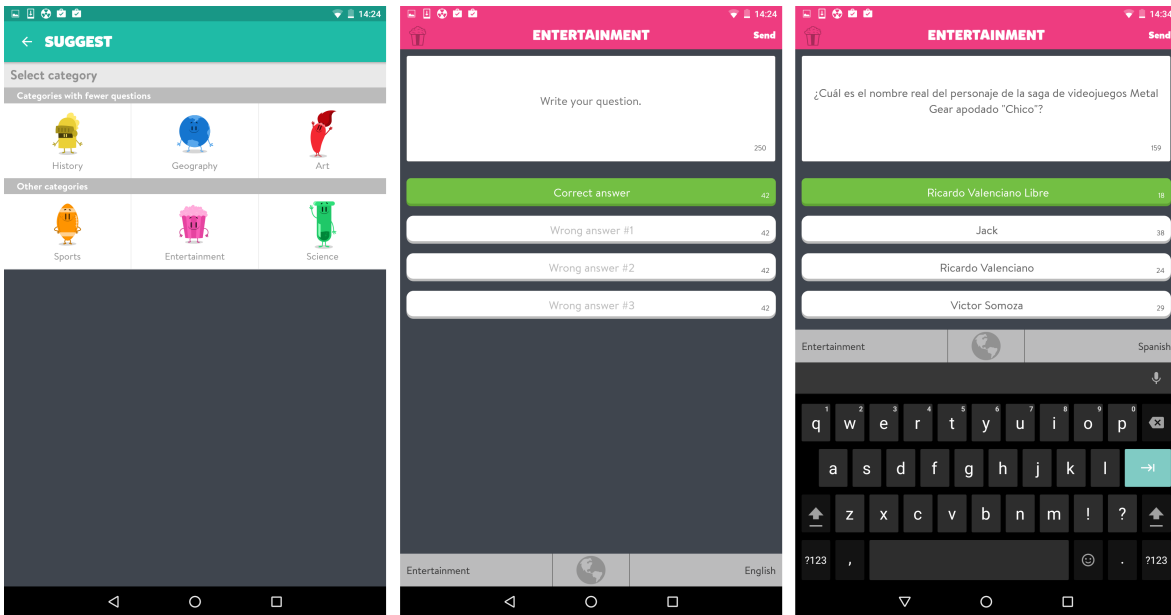


Figura 3.6: Preguntados - Aportación de pregunta nueva

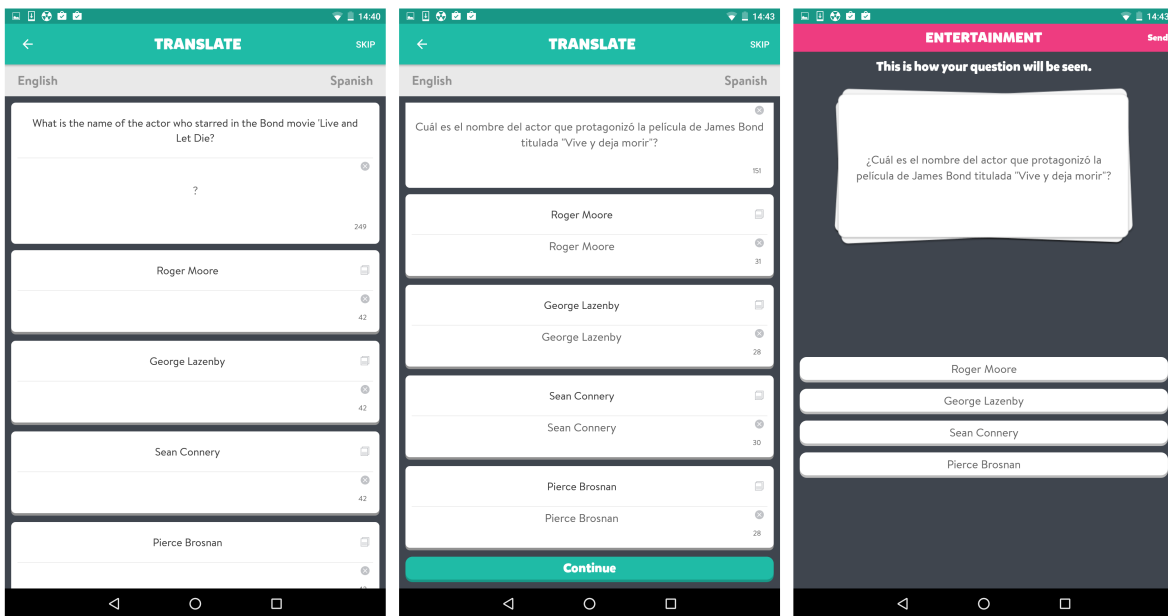


Figura 3.7: Preguntados - Traducción de preguntas

- Tanto las preguntas aportadas como las preguntas traducidas por un usuario pueden estar en tres estados: aceptadas, pendientes de aceptación o rechazadas. Para que una pregunta aportada o traducción sea aceptada y pase a formar parte del juego **se requieren un mínimo de revisiones** y es necesario que al menos el 85 % de las revisiones sean positivas. Esto también es un punto diferencial con QuPI-GAm, ya que la revisión de preguntas no es un proceso abierto, sino restringido a los jugadores y supervisado.

- A la hora de **valorar las preguntas** aportadas por otros usuarios existen tres posibles opciones, como se observa en la figura 3.8: dar la aprobación o bien indicar como indicencia que existe algún error o que la pregunta es correcta pero de poca calidad.

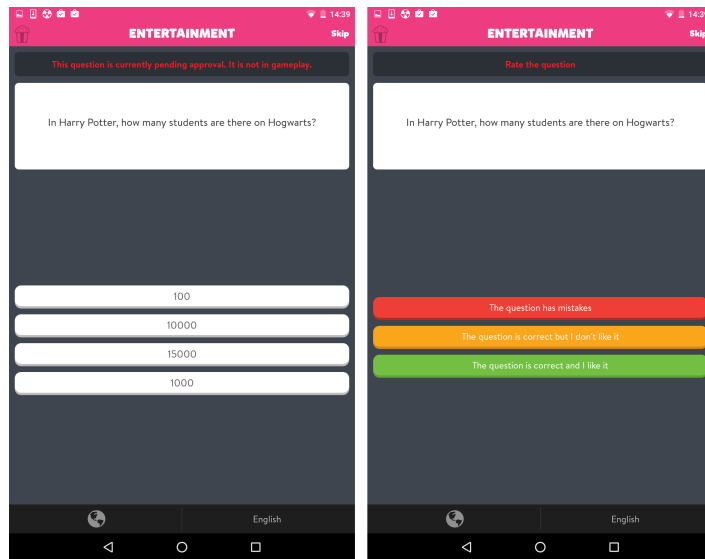


Figura 3.8: Preguntados - Valoración de una pregunta

- Todas las aportaciones realizadas por el usuario se recogen a modo de **historial**, pudiendo así consultar las preguntas aportadas y traducidas con su correspondiente estado (aceptadas, pendientes de aceptación y rechazadas). Respecto a las preguntas valoradas, es posible consultar el total asignado al usuario. La figura 3.9 lo muestra.

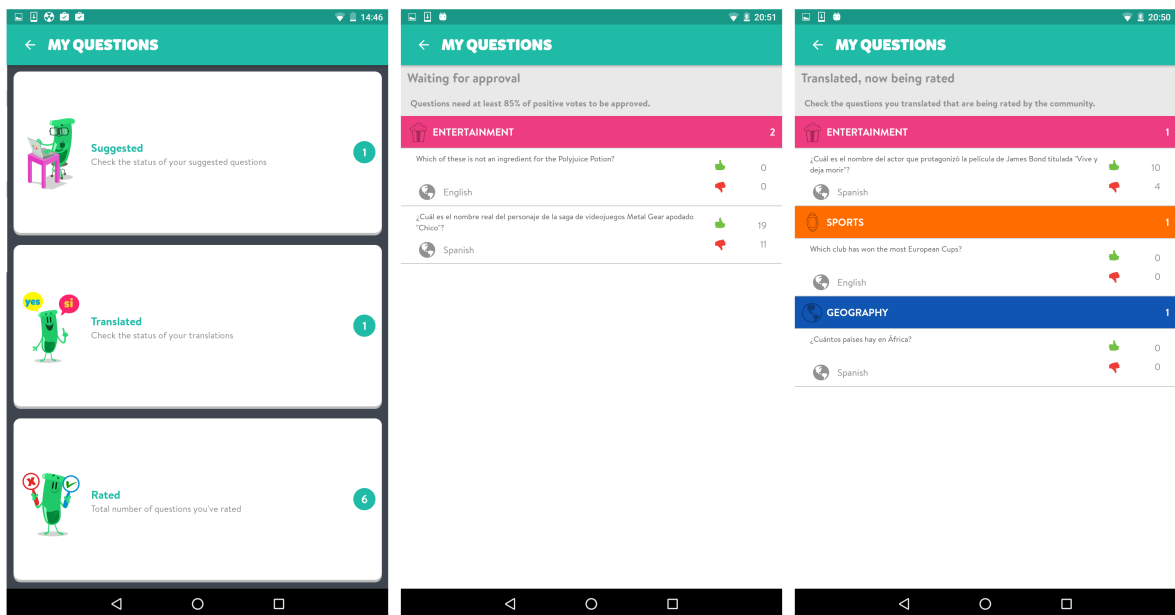


Figura 3.9: Preguntados - Historial de preguntas

- La factoría de preguntas también incorpora técnicas de gamificación con el objetivo de incentivar la participación. Concretamente **es posible obtener puntos para mejorar el perfil de usuario y logros**. Las preguntas aportadas y traducidas permiten obtener puntos una vez alcanzan el estado de aceptadas. Por otro lado no existe una diferenciación entre la faceta de jugador y colaborador, por lo que únicamente existe un perfil de usuario donde se vuelcan los logros de ambas. La figura 3.10 muestra la sección correspondiente al perfil del usuario así como el subconjunto de logros asociados.

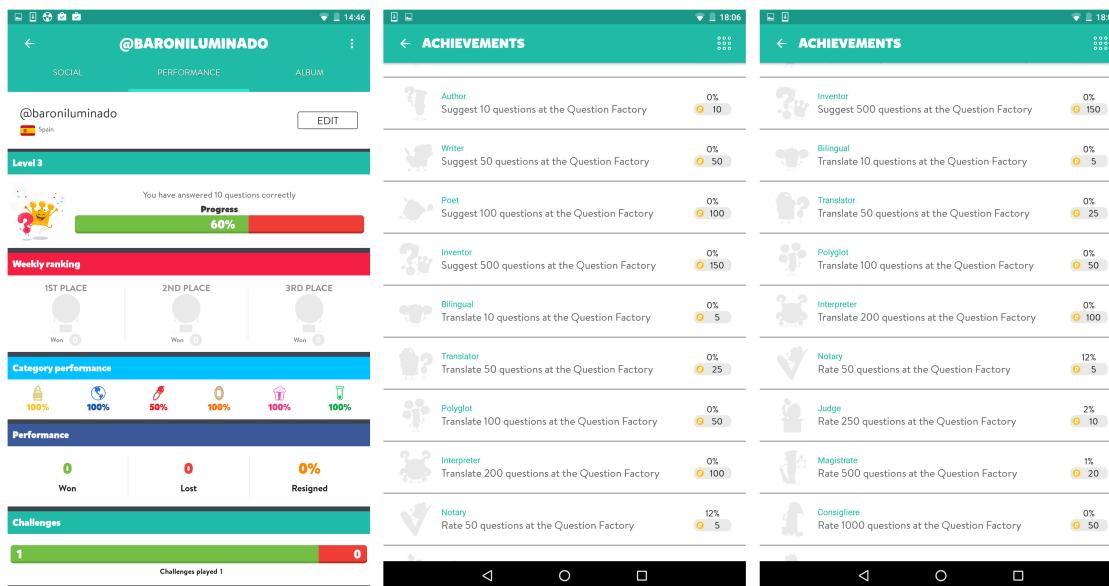


Figura 3.10: Preguntados - Perfil de usuario y logros de colaboración

3.2 Gamificación

3.2.1 Concepto de gamificación

Para comprender a qué alude el concepto y qué queda fuera conviene recurrir a una de las acepciones comúnmente referenciada y aceptada, recogida en [DDKN11]. Se propone la gamificación como una estrategia que promueve *'el uso de elementos de diseño de juegos en contextos no de juego'*.

Existe un conjunto de términos predecesores, complementarios o paralelos respecto a los cuales conviene establecer una separación, para comprender mejor qué abarca exactamente la gamificación y qué no. En primer lugar, existen dos cualidades que no deben ser confundidas: *Playfulness* y *Gamefulness*. La primera fue definida principalmente en el contexto de la HCI y comprende una concepción amplia relativa a la emulación de la experiencia de juego por medio de la forma de interacción. Esta está vinculada al disfrute generado por la propia acción de juego, más asociada a la libertad y a la exploración sin límites definidos. Por su parte, *gamefulness* apunta a la emulación de las características de juego, no como acción, sino como práctica sistematizada que implica unas reglas y objetivos a seguir.

Por otro lado, es importante no confundir la gamificación con los *Serious games* (referencia en [RCV09]). En ambos casos su finalidad no se vincula al entretenimiento, sino que se emplean como medio para alcanzar un objetivo en un contexto diferente al de los juegos. Sin embargo la gamificación solo adopta elementos de diseño característicos de los juegos, mientras que en los *Serious games* se desarrolla un juego completo, esperando alcanzar el objetivo definido a medida que se juega a este.

Un ejemplo claro para poner de manifiesto la diferencia entre ambos conceptos se tiene si se piensa en un simulador de vuelo. El objetivo es que los pilotos puedan practicar y ejecutar los protocolos de actuación frente a diferentes situaciones de una forma segura y sin riesgo. El simulador como tal se consideraría un *serious game*. Sin embargo, si al simulador se le adjuntasen una serie de módulos donde cada piloto pudiese tener un perfil donde consultar su progreso, recibiese recompensas o pudiera compararse con el resto de pilotos (entre otros) estaríamos ante un sistema (en este caso sería un juego) gamificado.

En la misma línea, tampoco ha de confundirse la gamificación con los *Pervasive games* (véase [MSW09] como referencia). Bajo esta denominación se engloban aquellos juegos que llevan más allá el concepto tradicional (en el sentido social, espacial o temporal) sirviéndose de tecnologías que permiten extender la interacción al mundo real. Como caso ejemplar donde la línea entre gamificación y *pervasive game* es algo difusa, se encuentra el acuerdo establecido entre la farmacéutica Duane Reade y la compañía Niantic Inc. Esta lanzó al mercado un juego de rol para móviles basado en realidad aumentada y gps llamado *Ingress*⁴⁰. Narra una historia cuya base es la existencia de dos facciones, donde el usuario debe posicionarse de acuerdo a sus acciones. La jugabilidad se basa en conquistar territorios y encontrar potenciadores visitando localizaciones del mundo real. Debido al acuerdo sin embargo, el juego se convierte en parte en herramienta de gamificación, en tanto que se potencia un objetivo ajeno al juego como es el de fomentar la compra en las farmacias, ya que para conseguir ciertas recompensas en el juego los jugadores pueden optar por entrar en los establecimientos y escanear un código.

3.2.2 El porqué de la gamificación

Una vez delimitado el concepto, el siguiente paso debe ir encaminado a establecer cuál es el fin de la gamificación y por qué se ha determinado que es útil la aplicación de elementos característicos de juegos en cualquier otro tipo de contexto. En los últimos años, el sector de los juegos digitales ha experimentado un *boom* remarkable. Solo en Estados Unidos, según se recoge en un estudio elaborado en 2016 por la Entertainment Software Association⁴¹, el consumo relativo a software en la industria de los videojuegos en 2015 se cifró en 16.500 millones de dólares. Esto denota el interés generado en torno a este tipo de actividad recreativa. Aun cuando el concepto de juego ha acompañado al hombre desde tiempos remotos,

⁴⁰<https://www.ingress.com/>

⁴¹<http://essentialfacts.theesa.com/Essential-Facts-2016.pdf>

la **digitalización de las plataformas** ha permitido constatar a gran escala cuan atractivos pueden llegar a resultar los videojuegos. Se observó que estos generaban un reclamo masivo, hacia lo que se puede considerar una actividad no productiva, que no en vano es capaz de generar dedicación plena, **interés y motivación**.

Y aquí es donde se revela el punto clave perseguido por la gamificación. Su fin no es otro que el de provocar en la persona, cuando se expone al sistema gamificado, un efecto similar al que se produciría si estuviese jugando a un videojuego, siendo el efecto perseguido la motivación, a través de la mejora de la experiencia de uso del servicio o aplicación con la que se esté interactuando. Se busca conseguir **inducir hacia comportamientos positivos** que permitan alcanzar determinadas metas, en ocasiones a través del desencadenamiento de ciertos efectos a nivel psicológico.

3.2.3 Elemento de diseño

Para poder aplicar gamificación, por tanto, conviene establecer qué se considera un elemento de diseño presente en los videojuegos, ya que esto asienta las bases para la elaboración de un diseño gamificado. En primer lugar Deterding propuso en [DDKN11] una distinción por niveles, de lo concreto a lo abstracto. Así, se distingue desde patrones para diseñar soluciones concretas de interacción, pasando por patrones de diseño asociables a mecánicas de juego (que deberían ser representables a través del nivel previamente mencionado), hasta llegar a líneas guía y principios de diseño, modelos de juego y metodologías y procesos a seguir para su elaboración.

Sin embargo, por lo general en los estudios revisados no se llega a tal nivel de detalle. Bajo diferentes denominaciones (como en [HKS14] la alusión a *motivational affordances* como mecánicas, en [KC16] la alusión a *human desires* refiriéndose a las dinámicas o en [Hag12] la alusión a los elementos de interacción como *building blocks*) lo común es que se listen las posibles mecánicas y/o patrones de interacción, así como las dinámicas de juego. Lo que es más, no en todos los estudios se hace referencia a cada uno de esos tres factores bajo la misma denominación, en algunos se engloba todo como elementos de diseño sin hacer distinción. En otros las mecánicas aluden también a los patrones de interacción. Tomando como referencia las clasificaciones presentes en los tres trabajos previamente mencionados así como en [SF14], [GBSTN16], [MF16] y en [PB14], se propone una recopilación de elementos de diseño característicos básicos:

Patrones de interacción. Alude a las distintas formas en que se puede dar salida a las mecánicas desde el punto de vista de la interacción con el usuario y que permiten ofrecerle *feedback* de una forma usable, atractiva y respetando el enfoque *gamefulness*. Se tiene:

- Sistema de puntuación. Permite medir o valorar cada acción realizada por el usuario de forma cuantitativa. Se puede definir distintos tipos de puntos (de experiencia, de colaboración, etc.) caracterizados en función de la forma en que se consiguieron.

- Medallas, trofeos, estampas. Son un medio claro para acreditar la los objetivos conseguidos por el usuario, y que este pueda revisarlos o compararse con otros usuarios.
- Niveles y rangos. Permiten valorar el avance del usuario y situarle con respecto al resto de usuarios y con respecto al conjunto de objetivos alcanzables en el sistema de forma objetiva y global.
- Rankings. Permiten establecer un medio comparativo entre los usuarios para las diversas actividades u objetivos específicos propuestos en el sistema, de forma separada.
- Perfil. Punto central donde el usuario debe poder consultar de forma clara toda la información relativa a su progreso. Deben incorporarse elementos gráficos como barras de progreso o diagramas, que resulten autoexplicativos y preferentemente que inciten al jugador a mejorar sus estadísticas y logros conseguidos.
- Bienes virtuales. Constituye una perfecta moneda de cambio en el mundo virtual frente a los logros conseguidos por el usuario. Para ello se debe realizar un diseño lo suficientemente atractivo como para que el usuario desee conseguirlos.
- Avatar. En muchos casos es una buena idea que el usuario tenga su representación en el medio virtual, ya que puede actuar como un catalizador del avance, al poder ser modificado y personalizado en función de los ítems conseguidos.

Mecánicas. Alude a una serie de pautas o reglas aplicables a la hora de llevar a cabo la actividad gamificada. Entre ellas:

- Restricción de tiempo. Bajo un buen diseño, si se le sugiere al usuario la consecución de objetivos en un tiempo limitado o una fecha límite, esto puede suponer un plus a la hora de evitar que aparque una actividad y la deje para otro momento. Además el empleo del tiempo como factor de diseño puede marcar la diferencia entre una gamificación estática o dinámica, siendo deseable la segunda.
- Elemento de azar. La incertidumbre de la suerte es un factor llamativo, aunque está más relacionado con la propiedad *playfulness*. El hecho de recompensar al usuario con algo no conocido de antemano, como puede ser una tirada en una ruleta que determine el premio final obtenido, fomenta el interés basado en la curiosidad de saber qué se va a obtener finalmente.
- Recursos limitados. Si al inicio se le dan al usuario unos ítems limitados, pero se le sugiere la posibilidad de conseguir ítems avanzados solo accesibles a través de la consecución de objetivos, esto incrementará el deseo por conseguirlos.
- Logros. Es importante organizar la actividad en torno a la consecución de objetivos claros, ya que permite orientarla y realizar un registro, que permite en cada momento revisar qué se ha conseguido.
- Abordaje de retos. La recomendación no es una buena herramienta para la motivación.

Sin embargo si se presenta un objetivo que deseablemente debe ser abordado bajo la perspectiva del desafío, esto resulta ser mucho más estimulante puesto que lleva implícita la idea de la superación.

- **Narrativa.** Si las actividades/objetivos a abordar van acompañados de forma paralela por una pequeña historia, esto fomenta la implicación, en tanto que las acciones del usuario pueden determinar el curso de la misma, y la curiosidad por averiguar cómo va avanzando es un buen agente motivador.
- **Estatus (para reconocimiento y comparación).** Un diseño que permita reconocer donde se sitúa el usuario respecto al resto resultará útil, ya que las personas por lo general necesitan posicionarse frente al resto. Además, la percepción de pertenecer a un determinado grupo, si se supone que este es superior al resto y ofrece privilegios, genera una sensación reconfortante.
- **Competición.** Ofrecer la posibilidad de que los usuarios se comparen entre sí resulta muy útil, ya que constatar que no se es el más virtuoso en algo, pero que se puede llegar a ser, puede ser aliciente suficiente para querer mejorar, lo que lleva a conllevar un incremento del rendimiento.
- **Coleccionables.** Fomentándose en el ímpetu de perfección, puede bastar que un conjunto de ítems digitales conseguibles se presenten como colección sin tener ninguna utilidad más allá, para que la persona tenga la necesidad de obtenerlos todos, por el simple hecho de no dejar la colección incompleta.
- **Exploración.** En ocasiones, es buena opción ofrecer algo nuevo no esperado ante un determinado avance. La sorpresa generada potenciará el deseo de inspeccionar todas las opciones de gamificación, para comprobar si hay nuevas sorpresas.

Dinámicas. Enlazando más con el aspecto psicológico, se han determinado una serie de rasgos de personalidad comunes que reflejan deseos y necesidades, de los cuales toman provecho las mecánicas de juego. Las dinámicas básicas son:

- **Recompensa.** Este es el medio que puede considerarse más explícito y directo para el incentivo. Recibir algo a cambio del esfuerzo realizado es útil cuando de inicio cuesta valorar el beneficio intrínseco de realizar la actividad.
- **Evolución.** Todo ser humano experimenta una sensación gratificante cuando puede constatar que ha progresado, ha mejorado con respecto a sus capacidades y habilidades o ha cambiado un aspecto deficiente o negativo.
- **Cooperación, altruismo.** Resulta reconfortante comprobar cómo uno es capaz de aportar soluciones que sirven de ayuda para otra persona, o bien que ha podido aportar algo a un proyecto colectivo con un fin de interés. Para explotar esta idea al máximo es importante diseñar sistemas donde se pueda medir la utilidad de la ayuda, por lo general con sistemas de votación donde los propios usuarios reconocen la labor del resto.

- Capacidad de expresión. Según el sistema al que se aporte la gamificación, la capacidad de poder manifestarse y mostrar la personalidad frente a los demás puede resultar llamativo.
- Reconocimiento. Cuando se realiza un trabajo y se obtienen resultados destacables se requiere que alguien reconozca la labor que uno desempeña, ya que resulta gratificante y es un método para corroborar que efectivamente ha sido bien realizado.
- Autodeterminación. Cuando se supera una meta, se resuelve un problema o se afronta una situación más o menos dificultosa, esto contribuye a la realización personal, ya que se presupone una evolución, una mejora.
- Reputación. Saber que los demás tienen información que nos posiciona en buen lugar resulta ser un acicate. Al igual ocurre a la inversa, lo que incita en ese caso a intentar cambiar la situación, mucho más que si nadie pudiese comprobarlo.
- Competitividad. Tener la opción de poder compararse con el resto, puede derivar en un incremento del rendimiento, por lo motivante que resulta ser mejor que otros en un cierto aspecto.
- Curiosidad. Si a una persona se le plantea una situación no resuelta, de la cual desconoce el desenlace, sea una cuestión que le afecte o totalmente ajena a ella, en muchos casos sentirá la necesidad de completar la información que le falta.

Tanto patrones de interacción, como mecánicas y dinámicas están relacionados, siendo estas últimas la base que justifica por qué las mecánicas funcionan tan bien; y dándose salida a las mecánicas a través de las diversas soluciones conocidas de interacción. Puede consultarse en Cuadro 3.2 la relación existente entre todas ellas.

Encontrar la clave de un buen sistema gamificado reside en combinar adecuadamente un conjunto de mecánicas y elaborar un diseño equilibrado. A tal efecto, se toma como referencia la visión de Brian Burke, vicepresidente de la agencia Gartner, quien sugiere que la clave reside en buscar las **3 Ms: Motivación, Momentum** (referente al equilibrio entre la capacidad del usuario y la dificultad de los retos) y **Meaning** (significado, que el fin que se pretende alcanzar aporte valor real al sujeto).

3.2.4 Evolución

La gamificación como concepto nace a raíz del impacto generado por los videojuegos, y de tratar de conseguir el estímulo que estos generan en otras áreas. Sin embargo no es cierto que la industria de los videojuegos fuese pionera a la hora de aplicar ese conjunto de estrategias que la caracterizan y que consigue promover ese interés. A lo largo del tiempo el ser humano ha tendido a organizar ciertas actividades bajo un enfoque de características similares.

Dinámica	Mecánicas	Elemento de interacción
Recompensa	Elemento de azar, Recursos limitados, Coleccionables	Sistema de puntuación, Medallas, Trofeos, Estampas, Niveles, Rangos, Bienes virtuales
Evolución	Restricción de tiempo, Recursos limitados, Logros, Abordaje de retos, Coleccionables	Sistema de puntuación, Medallas, Trofeos, Estampas, Niveles, Rangos, Rankings, Perfil
Cooperación, altruismo	Estatus	Sistema de puntuación, Medallas, Trofeos, Estampas
Capacidad de expresión	Recursos limitados, Logros, Coleccionables	Perfil, Avatar
Reconocimiento	Logros, Estatus	Sistema de puntuación, Medallas, Trofeos, Estampas, Niveles, Rangos, Perfil
Autodeterminación	Logros, Abordaje de retos, Coleccionables	Sistema de puntuación, Medallas, Trofeos, Estampas, Niveles, Rangos, Perfil
Reputación	Logros, Estatus	Sistema de puntuación, Medallas, Trofeos, Estampas, Niveles, Rangos,
Competitividad	Restricción de tiempo, Logros, Abordaje de retos, Estatus, Competición	Medallas, Trofeos, Estampas, Rankings
Curiosidad	Elemento de azar, Recursos limitados, Narrativa, Coleccionables, Exploración	Rangos, Bienes virtuales, Perfil, Avatar

Cuadro 3.2: Mecánicas en función de las dinámicas que potencian

Precedentes

Véase como ejemplo paradigmático las **competiciones deportivas**. Aquí puede detectarse claramente una motivación basada en la comparación y el reto, donde el reconocimiento basado en elementos como la posición obtenida, la puntuación conseguida o la obtención de medallas/trofeos, entre otros, suponen suficiente acicate para que el deportista entrene y quiera mejorar, sin ninguna otra recompensa tangible más que la reputación. Además la concepción de progresión y avance, clave en el deporte, va implícita en los citados marcadores. Es muy probable que de no existir las competiciones, los récords establecidos hoy día, conseguidos únicamente a base de perseverancia individual, no se hubiesen alcanzado a tal nivel. Ciertamente puede que el concepto de récord ni siquiera existiese, ya que es otro marcador que denota comparación, otro elemento más de gamificación.

Sin embargo puede considerarse que las competiciones surgieron de manera natural, derivándose de ellas de forma no premeditada el perfeccionamiento en los parámetros atléticos de los deportistas. No así con la que se considera una de las primeras manifestaciones explícitas de ideas precursoras de la gamificación, donde conscientemente se buscó aplicar una serie de **técnicas para la mejora de la productividad**. Hay que remontarse a la pri-

mera mitad del siglo XX, en la **Unión Soviética**, donde se buscó incrementar los índices de producción por una vía alternativa a la obvia y denostada por considerarse capitalista, la contraprestación económica. Como se recoge en [M.12], amparado por el ideario comunista, se implantó una **competitividad socialista** que permitía ensalzar el valor del trabajo en base a la obtención de puntos por parte de las fábricas según los objetivos alcanzados, menciones y reconocimiento por alcanzar ciertos umbrales y estampas distintivas a los trabajadores.

A lo largo del tiempo muchos de los avances han venido impulsados por la logística desplegada ante conflictos bélicos o cuestiones políticas, siendo otro claro ejemplo el sistema de reconocimiento militar basado en **condecoraciones**. Sin embargo, también son destacables las novedades impulsadas desde otros ámbitos, como el comercial, concretamente las estrategias de marketing, donde se agudiza la creatividad para llamar la atención de los potenciales clientes. Algunas de las tácticas que se aplican aún hoy día tienen su origen a principios del siglo XX, y están basadas en estrategias que sugieren una primera aproximación a la gamificación. Muy orientado al premio o recompensa, las *S&H Green Stamps*⁴² son uno de los primeros intentos de fidelización mediante la creación de un catálogo de premios, al alcance de aquellos que acumulasen puntos (sellos verdes) por compras realizadas en distintos establecimientos adheridos a la iniciativa. En los años ochenta se produjo un boom de este tipo de tácticas de fidelización, donde desde compañías aéreas hasta cadenas hoteleras, implementaban sistemas para **incentivar el consumo en base a puntos**.

Citados ya algunos ejemplos que pueden considerarse precursores de la gamificación, hay que poner el foco en los primeros intentos que se dieron por sistematizar lo observado hasta el momento. Una de las primeras obras de referencia viene de la mano de Charles Coonradt [C.84], con su libro *The game of Work* (1984), donde trató de recopilar una serie de mecánicas que parecían funcionar como herramienta motivacional en los ámbitos del ocio recreativo y el deporte. La finalidad era proponer estrategias de liderazgo en la empresa que consiguiesen que los empleados disfrutasen más de su trabajo, obteniendo consecuentemente mejores resultados a causa de una mayor implicación y eficacia.

También en los ochenta, pero en otra dirección y desde el punto de vista de las aplicaciones software, se abriría una vía complementaria que cabe ser mencionada, y que trataba de asociar la diversión generada por los juegos a aspectos relativos a las interfaces, como en [Mal82], un artículo presentado en 1982 donde se trataba de identificar qué hacía a los juegos de ordenador tan fascinantes y divertidos y proponía una serie de líneas guía en el diseño general de interfaces de usuario basado en las conclusiones extraídas. Desde entonces, se ha considerado la diversión como un factor más a requerir a la hora de diseñar interfaces de usuario en numerosas obras. Véase como referencia el libro *Funology: From Usability to Enjoyment* [BOMW04] donde se propone mejorar la experiencia de uso centrándose en aspectos como las emociones.

⁴²<https://www.greenpoints.com/>

Como se indicaba al comienzo, y como ya se ha desarrollado, **los juegos no fueron el germen de lo que supone la base de la gamificación**. Sin embargo, lo que sí puede afirmarse es que, no siendo ese su objetivo, se convirtió en un excelente campo de pruebas donde convergieron un conjunto de tácticas que demostraron ser una herramienta eficaz de estímulo y generadoras de diversión. Los años 90 fue una época de expansión donde las videoconsolas se hicieron hueco en muchos hogares, y se consolidaron como una novedosa alternativa de ocio. Esto favoreció que los **videojuegos actuaran como catalizador para comprobar, demostrar y tomar consciencia del fenómeno que arrastraban**, o que se manifestaba a través de ellos, dando pie a la formalización de la gamificación como concepto.

Formalización del concepto

El término empezó a concretarse de forma tímida a través de ciertos aportes en la primera década del siglo XXI. Se cree que la primera referencia a este fue realizada por **Nick Pelling**, cuando creó Conundra, una consultora especializada en gamificación ⁴³, según su propia definición. Fue a partir de 2008 cuando se empezó a reflexionar de forma más contundente acerca de su potencialidad y se empezó a hacer eco a través de blogs ^{44 45}.

A partir de 2010 se experimentó un boom, poniéndose el foco de atención en la gamificación desde distintos frentes. Desde la inspección más teórica y académica, siendo **Sebastian Deterding** uno de los primeros y más visibles (y a su vez críticos) exponentes hasta expertos que, en una aproximación más práctica, han creado guías de referencia en base a la experiencia. Puede citarse como ejemplo las líneas guías sugeridas por Oracle Corporation⁴⁶. También han aparecido compañías dedicadas en exclusiva a ofrecer soluciones de gamificación, algunas de las más notorias **Badgeville**⁴⁷ y **Bunchball**⁴⁸, esta última dedicada a integrar mecánicas de juego a redes sociales, webs, aplicaciones móviles desde 2007, por lo que fue una de las primeras que supo captar la potencialidad de la gamificación. En este sentido, cabe destacar el caso de la compañía **Snowfly**⁴⁹ que desde 1999 se ha dedicado a ofrecer soluciones software para incentivar a los trabajadores en el entorno de trabajo. Por otro lado, se han creado diversos puntos de encuentro de carácter internacional donde compartir el conocimiento generado y presentar nuevas soluciones, así como favorecer la creación de oportunidades de negocio. Ejemplos son la **CWG Conference**⁵⁰ y el reconocido *workshop* impulsado por Gabe Zichermann, el **Gamification Summit**⁵¹, donde acuden representantes de numerosas empresas de diversos ámbitos.

⁴³<http://www.nanodome.com/conundra.co.uk/>

⁴⁴<https://blog.oogalabs.com/2008/11/05/gamification-game-mechanics-is-the-new-marketing/>

⁴⁵<http://www.bretterrill.com/2008/06/my-coverage-of-lobby-of-social-gaming.html>

⁴⁶<http://www.oracle.com/webfolder/ux/applications/uxd/assets/sites/gamification/index.html>

⁴⁷<https://badgeville.com/>

⁴⁸<http://www.bunchball.com/>

⁴⁹<https://snowfly.com/>

⁵⁰<http://www.gwc-conference.com/es/>

⁵¹<http://gsummit.com/>

Gamificación como área de interés

La gamificación se ha convertido en un área de interés que **ha generado altas expectativas**. En 2011, atendiendo a los informes elaborados por la empresa de investigación y asesoramiento Gartner⁵², se predijo que el 50 % de las empresas que apostasen por incorporar procesos de innovación, aplicarían la gamificación en esos procesos⁵³. Si se observa el **Hype Cycle** elaborado anualmente como representación gráfica del estado de las tecnologías emergentes, puede observarse en la figura 3.11⁵⁴ que se alcanzó el pico de expectativas en el año 2013. A partir de entonces se ha entrado en una fase de normalización y según sus predicciones se espera que en no más de 5 años se hayan desarrollado todas las grandes innovaciones en este campo, entrando entonces en un periodo de estabilización.

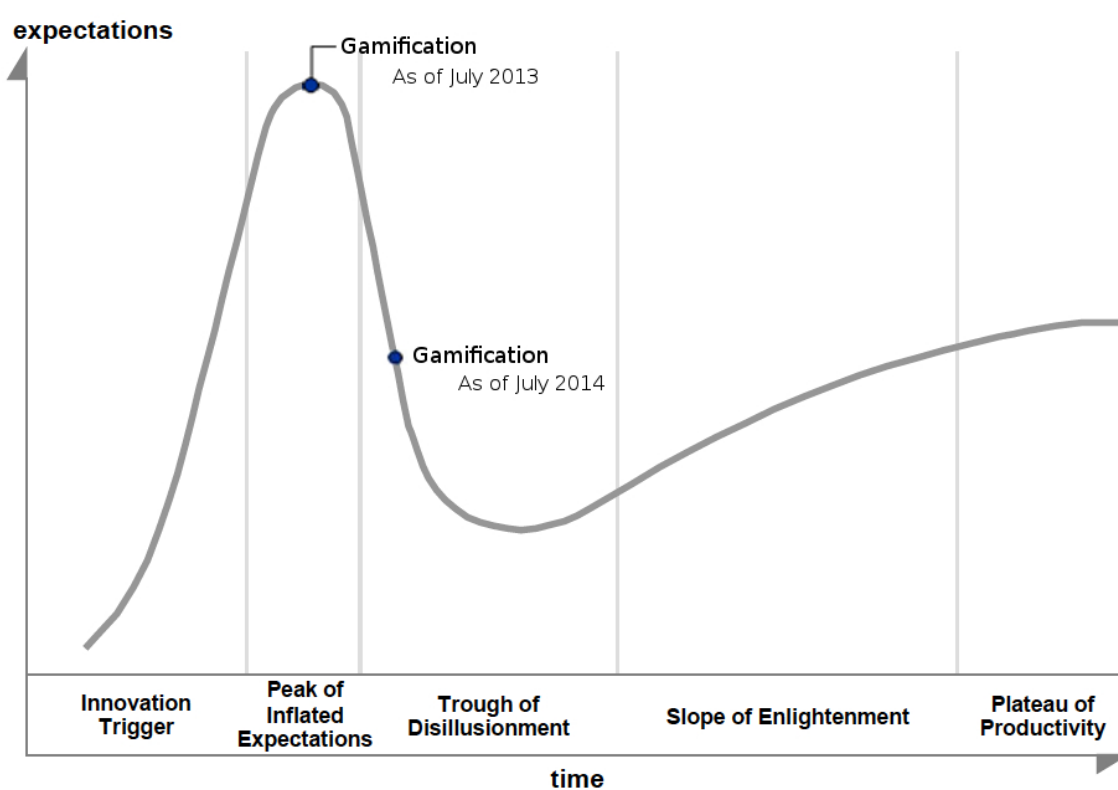


Figura 3.11: Gartner Hype Cycle para la gamificación (años 2013 y 2014)

Prestar atención a las búsquedas en Internet puede resultar útil para la determinación de tendencias. Siguiendo este enfoque, atendiendo a **Google Trends** como herramienta que permite acceder a metainformación relativa a las consultas registradas en el motor de búsqueda Google, se tiene que el término “gamification” (véase figura 3.12⁵⁵) no generó ningún tipo de interés hasta 2011. A partir de ese momento se comenzaron a registrar búsquedas de for-

⁵²<http://www.gartner.com/technology/home.jsp>

⁵³<http://www.gartner.com/newsroom/id/1629214>

⁵⁴Fuente: <http://www.gartner.com/newsroom>

⁵⁵Fuente: <https://www.google.es/trends/explore?date=all&q=gamification>

ma incremental, hasta alcanzar un pico de la **máxima popularidad (100 puntos) en 2014**. Desde entonces se ha mantenido más o menos estable con una popularidad de 80 puntos, alcanzando en noviembre de 2016 un pico de 91. De la misma fuente se extrae que las regiones donde se ha consultado más este término de forma relativa (en proporción al total de búsquedas generadas en cada una de ellas) son América del Norte y Brasil, Europa Occidental, Australia, India, Singapur y Sudáfrica.

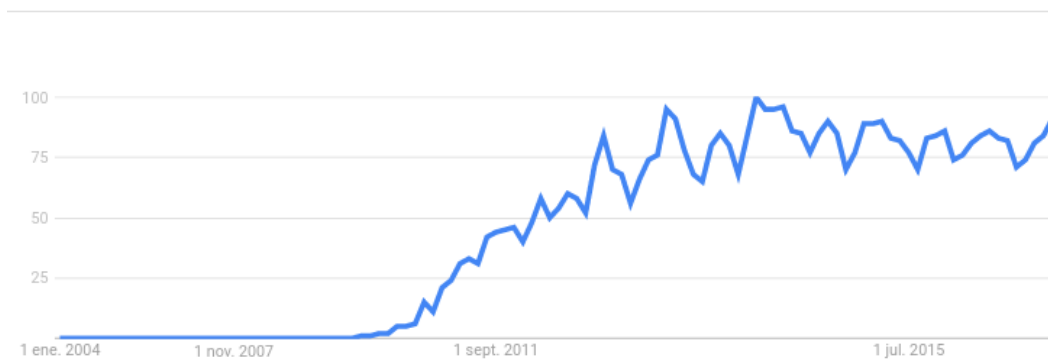


Figura 3.12: Google Trends - Interés de búsqueda para “gamification” a lo largo del tiempo

Aunque la gamificación como tal no está ligada a una tecnología concreta y teóricamente no es necesario un soporte digital para poder aplicarla, el hecho de que se haya concretado y popularizado en este momento y no antes se debe en gran parte a que **su aplicación resulta ser mucho más cómoda, fácil, práctica y accesible en soporte digital**. En relación a este punto también se teoriza acerca de que el hecho de que ya exista una **generación que ha nacido y crecido vinculada al medio digital** y acostumbrada a los videojuegos como recurso de ocio, puede haber favorecido la apuesta por la gamificación, pues se espera que estos estén más receptivos y que su interacción con sistemas gamificados resulte más natural. Aunque en la *Gamification Research Network* Deterding discutía este último punto asegurando que hasta el momento no había ningún estudio que lo demostrase (puede comprobarse en [Det12]), a partir de entonces se ha elaborado alguno donde se comprueba esto mismo. En [LA15] se realiza un experimento para comprobar si los resultados obtenidos sometiendo a un conjunto de personas a un sistema gamificado para el aprendizaje dependían del grado de interacción previo con videojuegos y su familiaridad con el manejo de la tecnología. Los resultados revelaron que sí que existía una correlación en ese sentido, donde dado un nivel de experiencia se asimilaba mejor la lección.

Aun no sabiendo hasta donde llegarán las soluciones basadas en gamificación, lo que sí puede garantizarse es que actualmente **sigue siendo un mercado en auge que genera interés**. Se siguen haciendo esfuerzos por mejorar y ofrecer soluciones cada vez más eficaces, atendiendo en parte a la experiencia acumulada en este corto aunque intenso periodo en que la gamificación ha despegado como tal. Como muestra de esto, se puede ver **el ejemplo de TalentLMS**, una plataforma de eLearning ofrecida como SaaS y que emplea la gamificación

como recurso, que en 2014 realizó una encuesta entre los usuarios de sus servicios⁵⁶, para tratar de determinar sus preferencias y qué era mejor percibido. También se espera que la introducción de tecnologías que están despuntando, como la realidad aumentada o los beacons, den soporte a un despliegue basado en gamificación que sea capaz de ofrecer una experiencia aún más atrayente.

3.2.5 **Ámbito de investigación**

La gamificación ha recibido atención académica de forma intensa y continuada sobre todo a partir del 2011. Actualmente se sigue dando salida a una cantidad considerable de **estudios y artículos que analizan el fenómeno o tratan de realizar aportaciones** útiles. Con el fin de poder comprobar en qué sentido se dirimen las líneas de investigación y desarrollo actuales, se ha escogido un artículo reciente para algunos de los enfoques típicos que se encuentran en este tipo de estudios y se presenta un resumen para cada uno de ellos.

Estudios sobre los Beneficios y la Efectividad de la Gamificación

How gamification motivates visits and engagement for online academic dissemination

Véase [KC16]

Evaluando las posibles dificultades extra que pueden surgir a la hora de incorporar gamificación en aplicaciones educativas, el trabajo se centra en **medir la mejora conseguida al introducir gamificación en una plataforma online** diseñada para la promoción y divulgación académica que involucra a estudiantes, docentes, personalidades de la industria y al público general. El objetivo de esta plataforma, llamada LOPUPA, es el de la divulgación de conocimiento académico, productos y actividades.

La evaluación consistió en la recopilación y **análisis de datos tanto cuantitativos** (a través de Google Analytics) **como cualitativos** (a través de un formulario online). Se demostró que la gamificación tiene el potencial de atraer, motivar y mantener a los usuarios, según sus conclusiones. Concretamente las técnicas de gamificación aplicadas en la plataforma consiguieron reducir el *porcentaje de rebote* (ocasiones en las que un visitante abandona la web tras ver únicamente una página) a tan solo un 20.74 %, incrementar la duración media de las visitas a 2 minutos y 14 segundos, incrementar el número de páginas vistas durante una sesión a 17.76 e incrementar el porcentaje de visitantes que vuelven a la plataforma a un 73.7 %; cifras que han alcanzado las expectativas planteadas.

Respecto al análisis cualitativo, los resultados obtenidos indicaron que para los usuarios de la plataforma los tres métodos de gamificación más importantes fueron: retroalimentación virtual gráfica (trofeos, medallas o cartas), actividades temáticas y juegos web y por último los foros de discusión(para mantener el contacto con otros usuarios y administradores).

⁵⁶<http://goo.gl/q8nqrr>

Aplicación de gamificación en ámbitos concretos

Learning healthy lifestyles through active videogames, motor games and the gamification of educational activities

Véase [GGN⁺16]

La **obesidad** se ha convertido en uno de los grandes problemas de la sociedad moderna. En España la sufre el 62 % de la población y ha sido declarada por la Organización Mundial de la Salud como la epidemia del siglo XXI. Por todo ello en los últimos años se ha convertido en el centro de atención de múltiples colectivos que tratan de frenar su expansión. En este trabajo se desarrolla un programa de entrenamiento apoyado por gamificación para tratar de prevenir la obesidad infantil en niños de entre 8 y 12 años.

El **programa de entrenamiento** se dividía en tres partes: 16 sesiones de grupo en el colegio de 90 minutos, 16 sesiones individuales en casa de 45 minutos y 90 minutos de formación para la familia. Las sesiones en grupo se llevaban a cabo dos veces a la semana y constaban de una charla por parte del profesorado (con el objetivo de concienciar a los niños sobre la obesidad y reforzar los contenidos aprendidos), actividades físicas para practicar las habilidades motoras y el uso de la aplicación TANGO:H (una aplicación diseñada para ayudar en la rehabilitación física y el entrenamiento cognitivo de niños enfermos con la particularidad de que el conjunto de actividades no está definido, es capaz de generar nuevos ejercicios a través de un simple editor).

Las **sesiones individuales en casa** también se llevaban a cabo dos veces por semana y constaban del uso de dos videojuegos. Por un lado *Pirates Island* (juego multijugador online con actividades educativas) y por otro *Wii Fit Plus* (para continuar fomentando la realización de ejercicio físico). Las mecánicas de gamificación utilizadas en el programa fueron: puntos, insignias, tablas de clasificación, tiempos, desafíos y retroalimentación positiva.

Los resultados obtenidos demostraron que el **programa de entrenamiento resultó efectivo** enseñando a los niños participantes comportamientos y hábitos de vida saludables. Además, gracias a la gamificación, se consiguió una **adhesión al programa en las sesiones en casa del 95.6 %**. Las puntuaciones, niveles y recompensas motivaron a los niños a realizar las sesiones programadas en casa.

Estudio de la base psicológicas tras la gamificación

Dentro del ámbito de la psicología existen diversas teorías y estudios que ahondan en el análisis de rasgos o características humanas que tienen una conexión directa con los factores que dan soporte a la gamificación. De esta forma se conectan los resultados esperados de esta con una base teórica y conductual. Sería en los años 40 cuando Abraham Maslow propondría su **teoría de la motivación humana**, iniciando así una discusión básica para la gamificación, ya que la motivación resulta ser el elemento clave.

A theory of human motivation

Véase [Mas43]

Buscando una respuesta que explicase qué induce a la acción genuina en el ser humano de forma intrínseca (en ausencia de elementos externos que condicionen su comportamiento), Maslow proponía las **necesidades físicas, psicológicas y afectivas como motor inductor**. Se planteaba así la existencia de distintos niveles de necesidad donde, una vez las de un nivel han sido cubiertas, el ser humano actúa con el fin de satisfacer las necesidades del nivel superior. Los primeros niveles relativos a (1) necesidades fisiológicas, (2) alcance de un cierto umbral de seguridad y estabilidad y (3) realización amorosa, afectiva y de pertenencia van orientadas a la cobertura de carencias. A partir de ahí se alude a (4) la reputación y reconocimiento ajeno y propio y la (5) autorrealización para desarrollar el propio potencial.

Self-determination theory and work motivation

Véase [GD05]

Los avances en el campo han terminado derivando en la **Teoría de la Auto-determinación**⁵⁷, algunos de cuyos máximos referentes son Edward Deci y Richard Ryan. Se analiza la necesidad como agente motivador, pero una **necesidad universal**, compartida por igual por todos los seres humanos, innata y que no puede ser satisfecha en mayor o menor medida, simplemente satisfecha. Las necesidades que se consideran bajo este enfoque son: autonomía (poder decidir), competencia (poder desenvolverse) y relación (poder interactuar). En este extenso artículo se desarrollan aspectos como la Teoría de la Auto-determinación (SDT), la Teoría de la Evaluación Cognitiva (CET) como teoría derivada y se analiza la **interacción y relación entre la motivación intrínseca y extrínseca**. Este es un punto determinante en la gamificación, ya que de alguna forma se recurre inicialmente a la motivación extrínseca como vehículo para poder desarrollar una motivación intrínseca. El empleo de elementos externos motivadores pueden resultar contraproducentes y disminuir la voluntad intrínseca. Es por ello que es necesario entender cómo ambas pueden coexistir y potenciarse.

Psychological theory and the gamification of learning

Véase [LBCA15]

En este artículo se realiza un repaso a diferentes teorías que son de aplicación cuando se trata concretamente de gamificación aplicada al aprendizaje, para lo cual se consideran dos ramas: teorías del aprendizaje y teorías motivacionales. El trabajo repasa algunos de las principales teorías de ambas ramas con el objetivo de analizar que aspectos pueden resultar de utilidad a investigadores y profesionales. Se sugiere la importancia de tener en cuenta las necesidades psicológicas de los usuarios y cómo ciertos aspectos de la gamificación les afectan. Como referencia, las teorías analizadas son las siguientes:

⁵⁷<http://selfdeterminationtheory.org>

- *Teoría del Diseño Instruccional Gamificado* (Gamified Instructional Design)
- *Teorías del Aprendizaje a través del Condicionamiento* (Learning via Conditioning)
- *Teorías basadas en las Expectativas* (Expectancy Theories)
- *Teoría del Establecimiento de Metas* (Goal-Setting Theory)
- *Teoría de la Autodeterminación* (Self-Determination Theory)

Líneas guía para el desarrollo de sistemas de gamificación

Knowledge discovery of game design features by mining user-generated feedback

Véase [GBSTN16]

Las técnicas de gamificación comprenden diversos métodos para motivar a los usuarios. Pero la gran incógnita es descubrir que conjunto de técnicas es aquel capaz de **garantizar una mayor probabilidad de éxito en cada aplicación concreta**. Para llevar a cabo este cometido los autores de este trabajo analizaron las características tanto de los juegos para móvil Android de mayor éxito como de aquellos menos atractivos, basándose en su posición en el ranking de Google Play Store, usándose finalmente un total de 60 para el estudio. Por otro lado, y apoyándose en otras publicaciones, establecieron una **clasificación con todos los métodos de gamificación relevantes** a considerar. De este modo, y a través de la generación de diferentes **modelos de datos y la utilización de algoritmos de aprendizaje automático**, trataron de averiguar qué características parecen poder relacionarse con el éxito de estas aplicaciones.

El estudio se llevó a cabo considerando las características tanto de forma individual como agrupadas en subconjuntos. Así pues la información se organizó a modo de matriz binaria, incluyendo para cada aplicación si cada una de las 24 características consideradas estaba presente o ausente, y si la aplicación era exitosa o no. Los resultados obtenidos reflejan que **ninguna de las características por sí sola es suficiente para determinar el éxito o fracaso de un juego**, siendo los puntos la característica que individualmente obtuvo un mayor valor. Por tanto **la solución pasa por combinar características**, aunque el trabajo no aporta conclusiones claras al respecto de que combinación concreta de características es aquella con una mayor probabilidad de éxito.

3.2.6 Áreas de aplicación

El interés y las expectativas en torno a la gamificación pueden constatarse fácilmente si se atiende al **gran número de empresas que manifiestan su interés**. Como muestra se adjunta en la figura 3.13⁵⁸ una relación de empresas con representación en el Gamification Summit celebrado en 2014. Por lo general, su foco de interés estará centrado en las posibles aplicaciones comerciales y/o laborales de la gamificación. Sin embargo existen muchos más ámbitos

⁵⁸Fuente:<http://sf14.gsummit.com/some-of-this-years-attendees-include/>

que también se aprovechan de su incorporación. Algunos de los más relevantes se revisan en esta sección, junto con ejemplos ilustrativos. Para localizar más ejemplos de casos de estudio donde se aplicó la gamificación de forma exitosa (atendiendo al ROI obtenido) puede consultarse la extensa lista recopilada en el sitio web del creador del framework *Octalysis*, elaborada en colaboración con Mario Malkav Colombo y Massimiliano Currolo⁵⁹.



Figura 3.13: Relación de empresas participantes en el Gamification Summit 2014

Educación

Buscando una mejora en los hábitos de estudio y un incremento en la concentración, que den como salida una mejora en el rendimiento y en los resultados derivados del proceso de aprendizaje, la gamificación puede ser un gran aliado a la hora de presentar el esfuerzo requerido de una forma más atractiva.

⁵⁹<http://yukaichou.com/gamification-examples/>

Véase el caso de **Code Academy**⁶⁰. Aprender un lenguaje de programación siguiendo guías de referencia o vídeos por propia iniciativa puede llegar a ser un tanto desmotivador y frustrante. Esta era la opinión de Ryan Bubinski y Zach Sims, motivo por el cual decidieron crear en 2011 una plataforma interactiva online que permitiese emular lecciones guiadas, como si de tomar clases se tratara, pero adaptado al ritmo del usuario y reconociendo su esfuerzo y su progreso a medida que se va registrando su avance. En la figura 3.14⁶¹ se aprecia el aspecto del perfil de usuario. Algunos de sus puntos fuertes son:

- Adquisición de puntos y medallas por metas alcanzadas dentro de cada lección. Incluye el registro del progreso del usuario, así como de las habilidades adquiridas.
- Propuesta de retos donde se han de abordar mini-proyectos, que deben realizarse en un tiempo acotado. También incluye la obtención de medallas con restricciones temporales, como el hecho de conseguir un mínimo de puntos en un mismo día.
- Posibilidad de compartir logros en redes sociales.

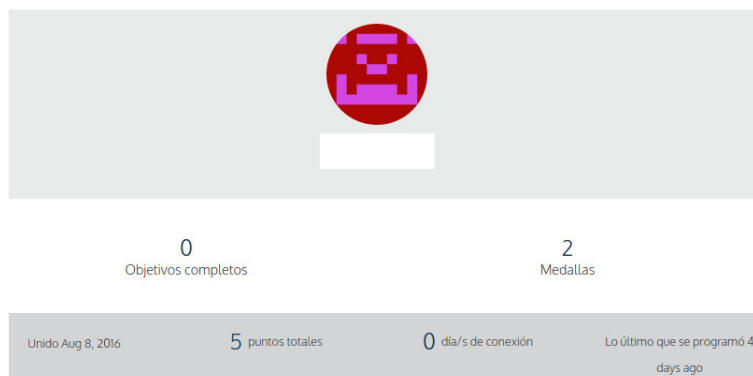


Figura 3.14: Code Academy - Perfil de Usuario

Comercial

Toda aquella iniciativa llevaba a cabo desde una perspectiva comercial tendrá como fin último tratar de incrementar los ingresos obtenidos. A tal efecto, la gamificación se posiciona como una herramienta que da soporte a tal fin mediante la modificación del comportamiento de los clientes, tratando de conseguir que consuman el servicio que se ofrece por encima del ofrecido por la competencia (fidelización, retención y diferenciación) así como aumentando la frecuencia de consumo.

Véase el caso de **Starbuds My Rewards**⁶². Este es un programa lanzado en el año 2009 con el objetivo de fidelizar los clientes de la compañía Starbuds. Se apostó por una iniciativa novedosa dentro de la industria del café, en la línea de diferenciación que le caracteriza.

⁶⁰<https://www.codecademy.com/>

⁶¹Fuente:CodeAcademy

⁶²<https://rewards.starbucks.mx/>

A través de la aplicación móvil de Starbucks el cliente puede realizar un seguimiento de las recompensas obtenidas. En la figura 3.15⁶³ se pueden ver algunas de las interfaces de la aplicación que recogen información del perfil como consumidor. Los puntos clave son:

- Por compras realizadas no solo en los locales de la cadena, sino en locales adheridos que venden sus productos, se consiguen estrellas digitales que se acumulan en el perfil de la aplicación. Las estrellas se consiguen introduciendo en la app un código incorporado a cada producto o bien a cada ticket por consumición en los locales.
- Se establece un único nivel alcanzable (nivel oro) con restricción temporal, al cual se puede acceder si se consiguen más de 30 estrellas en un periodo de 12 meses. Esto tiene su impacto en la obtención de ofertas como descuentos, la posibilidad de rellenar la bebida o comida gratis. Además la tarjeta física de cliente a través de la cual se realizan los pagos es reemplazada por un diseño exclusivo.

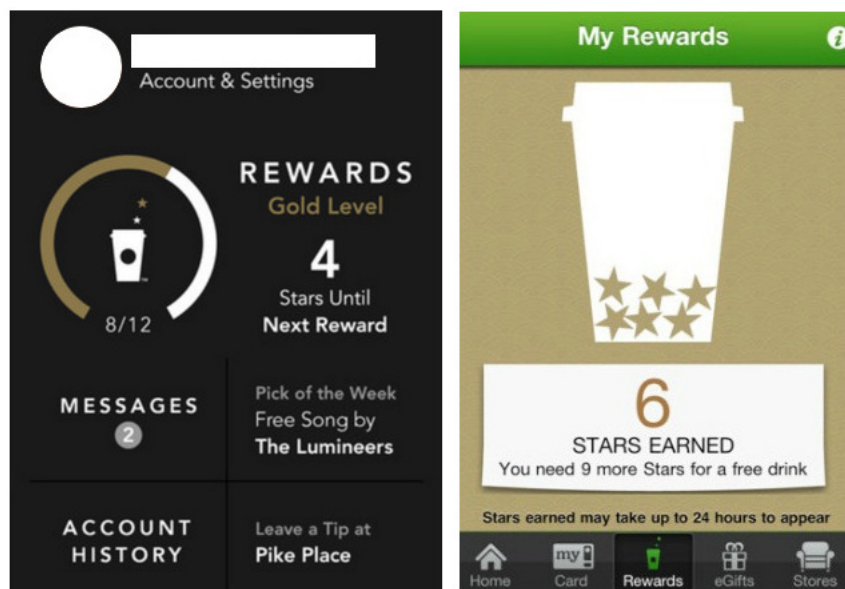


Figura 3.15: Starbucks - Perfil de consumidor en la app

Véase el caso de **Green Stamps**⁶⁴. La andadura de este programa de fidelización iniciado en 1896 aún perdura en la actualidad, lo que puede ser tomado como medida de efectividad y éxito. Auspiciado por la Sperry & Hutchinson Company, sus puntos clave eran:

- Recompensa. Se ofrecía una herramienta de fidelización a diversos negocios y establecimientos que adquirirían los sellos y los entregaban a cambio de un determinado desembolso establecido libremente. Los clientes conseguían sellos equivalentes a puntos que coleccionaban en una cartilla. Una vez obtenidos los 1200 puntos requeridos para completarla, se podía canjear por una serie de productos ofrecidos en un catálogo.

⁶³Fuente: Starbucks & <https://goo.gl/465QnX>

⁶⁴<https://www.greenpoints.com/>

- Reputación. Poder mostrar a familiares y amigos que se había conseguido realizar un gasto suficiente como para completar la cartilla, permitía demostrar que el portador tenía un cierto estatus, desde el punto de vista económico, del cual se podía presumir.

En la actualidad el sistema ha sido adaptado y renovado, trasladándose a la la plataforma digital y permitiéndo la adquisición de puntos a través de diversos canales de compra. En la figura 3.16⁶⁵ se muestra cómo era el antiguo libro en papel y los sellos.



Figura 3.16: Green Stamps - Aspecto del antiguo libro en papel

Colaborativo

Existen múltiples contextos en que la resolución de un problema puede ser únicamente abordable con la colaboración de un gran conjunto de personas, aprovechando lo que se denomina inteligencia colectiva. Dentro de esta denominación pueden encontrarse ámbitos específicos como pueden ser las comunidades y foros online, donde se emprenden discusiones, debates o se comparte el conocimiento para resolver problemas o el *crowdsourcing*, donde una entidad u organización pide colaboración voluntaria para abordar una tarea (véase [EAG12] como referencia).

Véase el caso de **TripAdvisor**⁶⁶. Como se describe más detalladamente en el análisis realizado en [Sig15], TripAdvisor es una web de viajes cuyo mayor activo son los miles de opiniones aportadas por los usuarios respecto al conjunto de servicios ubicados en el ecosistema del turismo. Fundada en el año 2000, su diseño ha ido variando hasta el momento actual, en el cual incorpora un interesante perfil de colaborador, que constituye la herramienta central complementaria a la propia utilidad de la web, en torno a la cual se incita a los viajeros a tomar parte en la misma. En la figura 3.17⁶⁷ se puede ver la información del perfil de colaborador. Los puntos clave bajo los que se sustenta son:

⁶⁵Fuente:<https://www.etsy.com/es/listing/127309616/albumes-de-4-completo-libros-s-h-green>

⁶⁶<https://www.tripadvisor.es/>

⁶⁷Fuente:TripAdvisor

- Sencillo y completo sistema para puntuar el elemento valorado, a través de una relación de características puntuables de 0 a 5 según el grado de cumplimiento, además de la posibilidad de dejar comentarios.
- Posibilidad de compartir las *reviews* en redes sociales.
- Posibilidad de valorar la utilidad de comentarios de otros usuarios, punto importante para reflejar el reconominio y la utilidad que ha generado la aportación.
- Obtención de puntos, medallas y trofeos, así como la capacidad de subir de nivel. Dispone además de un sistema que sugiere metas cercanas y las acciones que permitirían conseguirlo. A mayor nivel se tiene mejor reputación dentro de la comunidad, por lo que las opiniones reflejadas tienen más peso.

Figura 3.17: Trip Advisor - Perfil de colaborador

Salud

Con el objetivo de mejorar el estado de forma físico/mental, curar una dolencia o superar una adicción, a través de la gamificación se pueden enmascarar las pautas y recomendaciones a seguir, intentando que el receptor adopte unos hábitos de vida saludables y ayudando a superar así el desánimo inicial hasta que se consiguen resultados. Desde otra perspectiva, la gamificación también puede ayudar a la constitución de espacios colaborativos donde los facultativos puedan pedir una segunda opinión, minimizándose así el error en la valoración de una enfermedad, el diagnóstico y la forma de proceder.

Véase el caso de **RespirApp**⁶⁸. Esta es una aplicación para móviles actualizada en 2015, impulsada por la Asociación Española Contra el Cáncer con el fin de dar el apoyo definitivo a aquellas personas que requieren ayuda para dejar de fumar. En la figura 3.18⁶⁹ se muestran el aspecto de la aplicación. Sus características son:

- Dado el problema que se pretende resolver, este cuadra perfectamente con un enfoque basado en metas alcanzables. Bajo este enfoque, en vez de orientar al fumador con recomendaciones, se le orienta con retos, lo que resulta más estimulante.
- El fumador recibe feedback positivo en la forma de ayuda frente a diversas situaciones a las que se puede enfrentar, consejos y beneficios que obtendrá a medida que consiga alcanzar los objetivos.
- Dispone de un acceso directo al consultorio de la asociación, en el cual un profesional puede responder las dudas que surjan a lo largo del proceso.
- A medida que se alcanzan objetivos, se van registrando en la forma de medallas virtuales, que pueden ser compartidos en redes sociales.
- Al iniciarse la aplicación, se piden datos para poder conformar un perfil personalizado en base al cual se adaptan las metas que se proponen, comenzando con objetivos más asequibles que se tornarán poco a poco en más ambiciosos.



Figura 3.18: RespirApp - Aspecto de la aplicación

⁶⁸<https://www.aecc.es/TeAyudamos/informaryconcienciar/Programas/Paginas/Respirapp.aspx>

⁶⁹Fuente: RespirApp

3.2.7 Frameworks de diseño y desarrollo

Aspectos criticables

Si bien la potencialidad de la gamificación está mayormente aceptada en el plano teórico, esta es objeto de críticas desde el punto de vista práctico. La crítica va orientada hacia el hecho de que muchas implementaciones supuestamente basadas en gamificación no respetan lo que realmente implica. En el artículo *Gamification: Is it game over?*⁷⁰(2014) se recopilan algunos de los aspectos que, según algunas voces, la dejan al descubierto.

En algunos casos se critica la simplificación del concepto, como haría Margaret Robertson al advertir acerca de la *pointsification*, etiqueta que sirve para aludir al hecho de ver reducida la gamificación a la incorporación de puntos, niveles e insignias, bajo un simple esquema de recompensas. Esto revela la posibilidad de que aun no se haya asimilado la forma correcta de aplicar y entender el concepto y qué es lo que realmente implica. La gamificación requiere un diseño que tenga en cuenta y gestione correctamente la motivación intrínseca y extrínseca, orientándose siempre al tipo de público al que irá dirigida.

Otra corriente crítica alude a un uso negligente recogido bajo el término *exploitationware*, expuesto por Ian Bogost. Va referido al uso de la gamificación en el ámbito laboral para encubrir una estrategia que busque únicamente fomentar la explotación de los trabajadores. Siempre debe tratarse de mejorar las dinámicas de trabajo, no de crear una falsa ilusión bajo la cual se oculten condiciones de trabajo precarias. Esta puede considerarse más una crítica al sector que a la gamificación, ya que ésta abarca otras áreas donde este razonamiento no es de aplicación. Para esas otras áreas, otro riesgo considerable es la **adicción** que podría producirse según han planteado teóricos como Zichermann. Ciertos diseños podría sobrepasar la frontera de la motivación hacia un espectro de dependencia no recomendable, lo que debe ser también considerado y limitado.

La solución a algunos de esos problemas pasa por el establecimiento de **límites éticos**. En otros casos es necesario ofrecer pautas o guías para ayudar a definir **aproximaciones correctas**. A tal efecto, se han creado distintos frameworks, cuyo fin es sistematizar y formalizar el proceso de diseño en unos casos y de evaluación de sistemas gamificados en otros.

Recopilación de frameworks

A literature review of gamification design frameworks

Véase [MRGAM15]

En este artículo se recoge una completa revisión y una clasificación de diferentes propuestas. Se analizan un total de **18 frameworks** de diseño y desarrollo (algunos genéricos y otros para ámbitos específicos) en base a la atención dedicada a **19 ítems de diseño** importantes extraídos de la literatura. Estos van referidos a aspectos económicos (riesgos, viabilidad,

⁷⁰<http://www.bbc.com/future/story/20121204-can-gaming-transform-your-life>

retorno, objetivos), mecanismos de evaluación (métricas y análisis), aspectos psicológicos (sociales, éticos, motivacionales), tipo de interacción (con tecnología, empleando narrativa) y la lógica volcada (mecánicas de juego).

La conclusión general extraída es que la mayoría de frameworks consideran importante el aspecto psicológico y de interacción, realizando un diseño orientado a la persona y considerando los factores intrínsecos incluidos en la Teoría de la Auto-determinación. Se considera de valor la recopilación de información, pero no todos establecen métricas claras. Por su parte, los aspectos de lógica y económicos destacados varían según el framework escogido.

El artículo incluye un anexo con una tabla donde se destaca la dedicación de cada framework a las 10 características más relevantes que marcan la diferencia entre ellos, pudiéndose observar cómo el más completo de todos ellos es el denominado *Six steps to Gamification* (2012), propuesto por Werbach y Hunter. Muchos otros frameworks toman este como referencia, al igual que toman como pauta común el modelo para el diseño de juegos **Mechanics, Dynamics and Aesthetics (MDA)**, cuyos fundamentos pueden encontrarse en [HLZ04].

Framework destacado: Octalysis

De las múltiples alternativas presentadas en el artículo se destaca aquí el framework Octalysis (2013) propuesto por Yu-kai Chou, quien a través de su libro *Actionable Gamification* [Y.16] se pone en la piel del diseñador. Va profundizando poco a poco en las cuestiones a tratar, sirviéndose de ejemplos ilustrativos y proponiendo retos al lector. Organiza la información en torno a un **octógono**, cuyos lados representan las dimensiones a tratar, todas ellas fuertemente orientadas a la perspectiva de la motivación de la persona que interactuará con el sistema. Esos **ocho elementos básicos (Core Drives)** que han de guiar el diseño del sistema gamificado están representados en la figura 3.19⁷¹ y son: (1) Vocación por contribuir, (2) Superación personal, (3) Desarrollo de la creatividad, (4) Disfrute por posesión, (5) Interacción social, (6) Deseo de acceso a elementos exclusivos, escasos o limitados, (7) Curiosidad ante la impredecibilidad y (8) Evitar perder oportunidades.

La distribución de los elementos en el octógono no es casual. A la izquierda (**Left brain**) se localizan aquellas áreas que responden a la motivación extrínseca, asociadas más a la urgencia en incluso a la adicción, y a la derecha aquellas que se corresponden a la motivación intrínseca (**Right brain**). La diferencia entre ambas es que la primera es incentivada por factores externos, mientras en la segunda es la actividad en sí misma la que resulta motivadora. Así mismo, también existe distinción entre los elementos que se localizan en la parte superior del octógono (**White hat**) y los localizados en la parte inferior (**Black hat**), donde los primeros se consideran positivos al potenciar distintas capacidades de la persona, mientras que los segundos inciden en generar necesidad o incertidumbre.

⁷¹Fuente: <http://yukaichou.com/>

Un sistema gamificado que incida en técnicas de tipo *Left brain* y *Black hat* tendrá más posibilidades de generar una motivación a corto plazo que se extinga en poco tiempo. Por otro lado incidir en técnicas del extremo contrario en el octógono llevará a generar un compromiso a largo plazo. El propio octógono se emplea como herramienta evaluadora⁷², para lo cual un sistema es analizado dibujando la figura asociada a este en función de la dedicación a cada una de las ocho unidades.

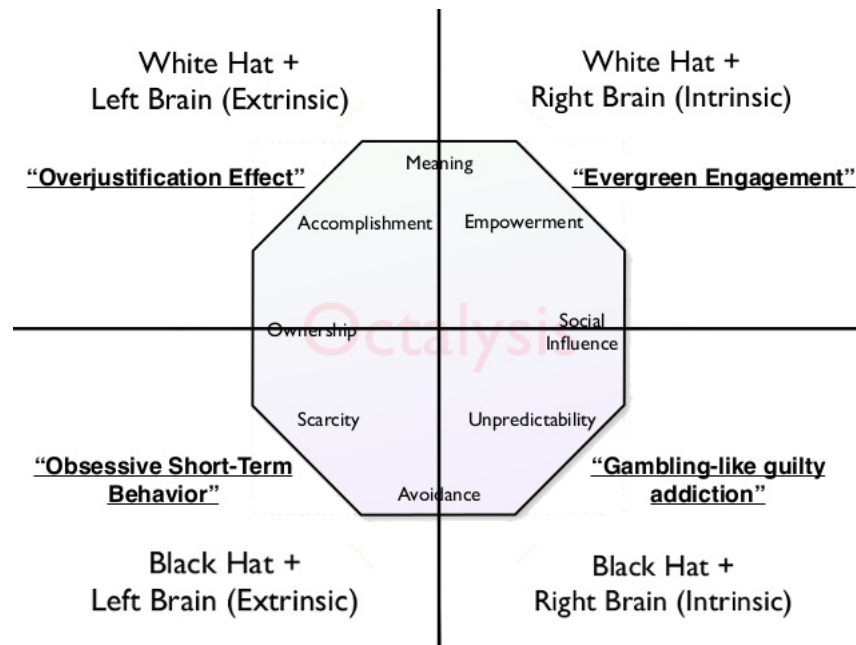


Figura 3.19: Unidades elementales del framework Octalysis

3.3 Alternativas tecnológicas para el desarrollo de apps

A grandes rasgos cuando se aborda la creación de una aplicación móvil es necesario determinar qué se quiere hacer y cómo se quiere hacer, para poder ajustar finalmente la viabilidad del proyecto. Para determinar qué se quiere hacer hay que realizar un análisis de la necesidad que se pretende cubrir o de aquello que se quiere ofrecer, pudiéndose así definir todas las funcionalidades a incluir y determinar la cantidad de trabajo requerido. El cómo implica tomar una serie de decisiones acerca de las bases que caracterizarán el desarrollo y **qué medios destinar (materiales, hardware y software)**, siendo este el punto de interés en el que se centra esta sección. Una vez determinado el esfuerzo estimado requerido y las herramientas y recursos escogidos, sería momento de trazar un estudio de viabilidad económica, lo que implicaría considerar:

- Tiempo y personal dedicados al desarrollo y mantenimiento.
- Estrategia de marketing a seguir.
- Cuota de adhesión y porcentaje de retención de los markets escogidos.

⁷²<http://www.yukaichou.com/octalysis-tool/>

- Licencias software.
- Arquitectura IT requerida (principalmente hosting o servidores desplegados).
- Mecanismos de monetización escogidos.

Como se aprecia, existen muchos aspectos a tener en cuenta, y no todos relacionados directamente con el desarrollo de la aplicación en sí. Esta constituye el primer eslabón de una cadena que termina en el usuario final. La consideración de este es fundamental. **Saber a quién se quiere llegar permite disponer de información clave** a tener en cuenta que afectará previamente al diseño y la toma de decisiones, así como a acciones posteriores relacionadas como la captación, fidelización o la comunicación con este.

Desde el punto de vista tecnológico, el mayor condicionante a la hora de elegir entre unas opciones y otras dependerá en gran medida del ecosistema en que la aplicación tenga presencia, lo que se elige en parte en función del perfil definido de usuario objetivo, ya que la **elección de un ecosistema** determina el tipo de usuario al que se podrá llegar. El ecosistema se compone esencialmente de tres elementos: **canal de distribución** elegido (asociado normalmente a los *markets*), **plataforma hardware** y **sistema operativo** empleado en el dispositivo móvil. Este último será factor crítico que condicionará la elección.

3.3.1 Elección de SS.OO. de interés

Para determinar a través de qué sistemas operativos móviles la aplicación estará accesible conviene conocer **cuáles son los principales y cuál es la cuota de mercado de cada uno**. Hace unos años existía una mayor variedad de plataformas, entre ellas Blackberry, Firefox OS, Ubuntu Touch o Simbian. Sin embargo, y tal como se reflexiona en el artículo del portal web Xataka móvil⁷³ poco a poco se ha ido produciendo una convergencia que ha derivado en la instauración de un duopolio, donde la cuota de mercado se reparte casi en su totalidad entre Android e iOS, perdiendo Windows Phone cada vez más fuerza como tercera alternativa. En ese mismo portal se recogen por países los datos acerca de la cuota asignada a cada sistema operativo. En la figura 3.20 se observan los datos para España, donde han crecido únicamente de 2015 a 2016 los usuarios de Android e iOS. Android es el claro ganador, habiendo sido elegido por un 92.4 % de usuarios, seguido por iOS con un 7.2 % y quedando relegado Windows Phone a un tercer lugar con un 0.6 %. Cabe destacar que este tipo de reparto es apreciable en el mismo sentido en el resto de países según los datos recogidos.

Así pues, asumiendo que las plataformas a considerar son únicamente tres, es necesario determinar en cuáles de ellas se desea que la aplicación esté disponible. Si se quiere llegar al mayor número de usuarios habría que elegir **Android**. Sin embargo este es un tipo de usuario que ha demostrado preferencia por las **aplicaciones gratuitas**.

⁷³<https://goo.gl/zBpa3A>

Spain	3 m/e Aug'15	3 m/e Aug'16	% pt. Change
Android	89.5	92.4	2.9
iOS	6.2	7.1	0.9
Windows	3.9	0.6	-3.3
Other	0.4	0	-0.4

Figura 3.20: Cuota de mercado de sistemas operativos móviles en España

El usuario de productos Apple se considera un tanto más selecto y puede estar dispuesto a un mayor desembolso. Según un informe elaborado por App Annie⁷⁴ en conjunto el **App Store de Apple generó el doble de beneficios** para los desarrolladores que la tienda Google Play en los dos primeros trimestres de 2016, aunque esta supera a la primera en el número de descargas registradas, como se aprecia en la figura 3.21⁷⁵. Otros factores a tener en cuenta son el tipo de usuario esperado en el país o países donde se pretenda distribuir la aplicación, o también la naturaleza de la aplicación, ya que si se ofrece una **funcionalidad muy específica** puede que esta interese solo a usuarios de un sistema operativo específico, pudiéndose considerar aquí la opción de **Windows Phone**.



Figura 3.21: Comparativa de descargas e ingresos en Google Play y App Store

⁷⁴<http://bgr.com/2016/07/20/ios-vs-android-developers-profits-app-store-google-play/>

⁷⁵Fuente: App Annie

3.3.2 Elección del tipo de aplicación

En función del presupuesto con el que se cuente y de si se quiere o no afrontar un desarrollo multi-plataforma, se optará por desarrollar una aplicación nativa o no, lo que a su vez influenciará la elección de las herramientas, entornos, frameworks y lenguajes.

Aplicación nativa

Una aplicación nativa es aquella que **se desarrolla específicamente para un sistema operativo**, lo que implica emplear el SDK correspondiente, que será diferente para cada plataforma. En principio es la opción más potente, ya que al usarse librerías que explotan directamente las funcionalidades del sistema operativo se obtiene el **mejor rendimiento** posible y la mejor adaptación al entorno, pudiéndose acceder al completo a los componentes del dispositivo. Para poder hacer uso de la app únicamente es necesario descargar e instalar el paquete correspondiente, por lo que una vez descargado no tiene por que ser necesaria una conexión a Internet, a no ser que la aplicación incorpore utilidades online.

Si se opta por el desarrollo nativo la elección del soporte tecnológico es sencilla, ya que para cada sistema operativo existen un entorno y un lenguaje específicos. En el cuadro 3.3 se especifican el lenguaje, formato de paquete, entorno y *market* específicos para los tres principales sistemas operativos.

			
OS	Windows Phone	iOS	Android
Compañía	Microsoft	Apple	Google
Market	Microsoft Store	Google Play	iTunes App Store
IDE	Visual Studio	Android Studio	Xcode
Lenguaje	.Net	Objective-C	Java
Formato	.xap	.ipa	.apk

Cuadro 3.3: Características software del desarrollo nativo según plataforma









En caso de optar por un escenario donde solo se considera una plataforma, este es el tipo de aplicación que hay que elegir. El problema viene cuando se opta por la opción multi-plataforma. Esto requeriría trabajar en **versiones diferentes de la aplicación**, multiplicándose las dificultades y el tiempo para el desarrollo y requiriéndose diferentes profesionales con conocimientos específicos para cada tipo de plataforma. Esto solo sería abordable en caso de disponer del capital suficiente. Si esta opción no es viable y solo se quiere trabajar en una única línea de desarrollo pero no se quiere renunciar al enfoque multi-plataforma existen otras alternativas.

Aplicación web

Una aplicación web es aquella que no se descarga y ejecuta directamente en el dispositivo, sino que es **accesible a través de la url donde se aloja haciendo uso de un navegador web** a modo de middleware. La aplicación no ocupa así espacio en la memoria del móvil, ya que la ejecución corre de cargo del navegador. La principal ventaja de este enfoque es que no se requieren conocimientos sobre los recursos específicos de cada plataforma. Se desarrolla una única versión de la aplicación **accesible desde cualquier dispositivo con independencia del sistema operativo** subyacente. Además se accede en todo momento a la última versión de la aplicación, a diferencia de las nativas dónde la aplicación descargada requerirá la instalación de actualizaciones.

La filosofía de desarrollo que se aplica es exactamente igual a la de un desarrollo web común, pero poniendo especial atención en la construcción de **interfaces adaptadas** al tamaño reducido de la pantalla, distribuyendo los contenidos y emulando en la medida de lo posible de la navegación típica y el *look and feel* esperable en una app. Se afronta por tanto una arquitectura cliente-servidor constituida por dos partes bien diferenciadas:

- **Frontend.** Asociado a la parte cliente, su función es la de gestionar aspecto gráfico de la aplicación, tomar los datos de entrada del usuario, solicitar los servicios requeridos al servidor y mostrar los datos de salida generados. Desde el punto de vista del desarrollo la elección de base se ciñe al modelo de estándares web, conformado por los lenguajes **HTML, CSS y JavaScript**. Sin embargo existen numerosas alternativas y herramientas software que pueden incorporarse para facilitar y agilizar el desarrollo. Se agregan librerías como jQuery, que ponen a disposición utilidades de más alto nivel, o bien directamente lenguajes de más alto nivel como CoffeeScript para el caso de JavaScript. También existen extensiones de CSS como LESS y SASS que ponen a disposición un lenguaje que facilita la creación de hojas de estilo. En cuanto a la compilación de HTML es interesante escoger un motor de plantillas, que permite aislar las etiquetas HTML del código y cargar estas junto con los datos que se le pasen. Mustache.js, Hogan.js o Handlebars.js son algunas alternativas. También relevante es la elección del framework en torno al cual se organice el desarrollo. En el cuadro 3.4 pueden consultarse algunos de los más utilizados.
- **Backend.** Asociado a la parte del servidor, constituye el motor de la aplicación y su función es la de servir peticiones que cubran la funcionalidad requerida, gestionando la lógica y las necesidades de persistencia. A diferencia del frontend, aquí existen numerosas alternativas respecto al lenguaje elegido y en función de este, ha de escogerse un framework que facilite la gestión relativa la recepción y despacho de peticiones. En el cuadro 3.4 pueden consultarse los lenguajes más populares y una selección de frameworks para cada uno de ellos.

	Lenguaje		Frameworks
Frontend		CSS	Semantic UI, Bootstrap, Foundation, Uikit, Skeleton
		JavaScript	Backbone, AngularJS, Ember, Vue, React, Nuclear
Backend		NodeJS	HapiJS, Meteor, ExpressJS, MeanJS, SailsJS
		Python	WebApp2, Django, Flask, Pyramid, CherryPy, TurboGears
		.Net	Asp.Net
		Java	Spring, Sparks, Struts, Vaadin, JSF, Play
		Ruby	Rails, Sinatra, Padrino
		PHP	Laravel, Symfony, Silex, KumbiaPHP, Kohana

Cuadro 3.4: Frameworks para frontend y backend

Aunque esta opción suprime totalmente la problemática relativa a la variedad de plataformas, implica asumir algunas renunciaciones. A diferencia de las apps nativas, se dispone de **acceso limitado a los componentes** hardware del dispositivo, el **rendimiento puede verse afectado** a causa del incremento de los tiempos de respuesta (al depender de la red y un servidor) y la experiencia de usuario no puede refinarse tanto. En ese sentido, el hecho de ofrecer el mismo estilo de interfaz en los distintos sistemas operativos puede resultar negativo, porque puede implicar una ruptura con la estética de general de la plataforma. Requieren una conexión permanente a Internet para poder ser ejecutadas, lo que puede ser inconveniente. Por otro lado, al no estar ligadas a un paquete descargable **no tienen cabida en los markets**, lo que es negativo si se tiene en cuenta que por lo general los usuarios no buscan aplicaciones concretas, sino que exploran el catálogo que ofrecen las tiendas para descubrir nuevas apps.

Infraestructura hardware Dado que la aplicación no se instala en el dispositivo del usuario sino que ha de estar alojada en un servidor, hay que contar también con las opciones disponibles a la hora de considerar el despliegue requerido. Una opción es **constituir por cuenta propia toda la infraestructura física** necesaria para poner en funcionamiento los servidores. Sin embargo cada vez toma más fuerza la opción de contratar una **infraestructura virtual, recurriendo a los servicios en la nube** ofrecidos por terceros. Estos ceden parte de su infraestructura física bajo demanda, escalando los recursos dedicados en función del tráfico generado de forma elástica. Puede contratarse únicamente la infraestructura física o bien solicitar servicios asociados con su gestión que permiten abstraerse totalmente de la configuración y centrarse en el desarrollo del producto o servicio ofrecido. En el cuadro 3.5 se recopilan algunas alternativas de diferentes proveedores, según se requiera un IaaS o un PaaS. Hay que tener en cuenta que en el caso de un IaaS únicamente se contrata el hardwa-

re, ningún software viene incluido más allá del sistema operativo. Es necesario encargarse por tanto de toda la configuración relativa a dependencias, firewalls, balanceadores de carga, base de datos y respaldo, entre otros. Supone un mayor esfuerzo, pero puede lograrse una mayor personalización y optimización que si se depende de la configuración del Paas.

Tipo de servicio	Proveedores	Servicio
IaaS	Amazon Web Services	Amazon EC2
	Google Cloud Platform	Google Compute Engine
	Microsoft Azure	Azure Virtual Machines
	OpenStack Foundation	OpenStack
Paas	Google Cloud Platform	Google App Engine
	Microsoft Azure	Azure Cloud Services
	Heroku	Heroku
	Red Hat	OpenShift Container Platform
	Century Link	AppFog

Cuadro 3.5: Servicios en la nube para el soporte de Web Apps

Aplicación híbridas

Las aplicaciones híbridas suponen un **camino intermedio** entre las apps nativas y las web apps, ya que la base del desarrollo es idéntica al de las aplicaciones web. Sin embargo se agrega un paso extra que permite tomar ciertos aspectos del enfoque nativo. Esencialmente se toma el código correspondiente al frontend y **se compila a nativo** con ayuda de herramientas dedicadas a tal fin. Se consigue generar un paquete similar al que se generaría en una aplicación nativa, pero conteniendo únicamente la gestión de la interfaz y los aspectos gráficos. Esta conversión permite acceder a algunas de las ventajas de estas últimas, entre ellas, disponer de un paquete descargable e instalable en el móvil gracias al cual se puede acceder a las tiendas de aplicaciones. Además la experiencia de usuario se ve favorecida, ya que se puede hacer uso de las API específicas que dan acceso a la utilidades propias del móvil y a los componentes hardware. Todo ello manteniendo la ventaja principal del enfoque web, el disponer de una única versión del código que puede ejecutarse en las distintas plataformas, pero también su principal desventaja, ya que el rendimiento sigue siendo inferior que el pueda obtenerse con una aplicación nativa.

Como se ha dicho, la aplicación se desarrolla inicialmente a la luz de los criterios característicos de una web app. La parte correspondiente al backend se seguirá manteniendo alojada en un servidor, constituyendo este el motor de la aplicación. Sin embargo **el código correspondiente a la parte de frontend es convertido**, empleando para ello librerías JavaScript que incluyen wrappers a través de los cuales se invoca finalmente el código necesario específico de cada sistema operativo. A tal efecto, existen frameworks de soporte como **PhoneGap** o **Titanium**, que permiten guiar este procedimiento hasta llegar a compilar y generar el paquete nativo correspondiente. Si bien la compilación a nativo suele afrontarse partiendo de los lenguajes web estándar, hay que tener en cuenta que existen alternativas que permiten aplicar el mismo enfoque pero bajo otros lenguajes. Es el caso de **Xamarin**, que ofrece una API a través de la cual se puede obtener la compilación del paquete contenedor partiendo de un desarrollo basado en C# y .NET.

Método de trabajo

A lo largo de este capítulo se pone énfasis en la forma en que se ha llevado a cabo el trabajo, bajo qué contexto y qué medios. Para ello se justifica qué metodología de desarrollo se ha seguido de base, explicando las variaciones respecto a esta. Por otro lado se caracteriza el desarrollo desde el punto de vista software y hardware, nombrando las principales tecnologías empleadas junto con ciertas herramientas de soporte.

4.1 Elección de la metodología de desarrollo

A la hora de afrontar el desarrollo se han seguido las pautas básicas establecidas por **eXtreme Programming**¹, ligeramente adaptadas a las particularidades que definen el contexto de desarrollo. Los principales condicionantes radican en el hecho de ser un desarrollo abordado por una única persona, pero requiriendo una fuerte interacción con el equipo de desarrollo de la empresa con la que se colabora. Por otro lado, la naturaleza de producto software al cual se dará salida, que puede ser integrado en un futuro en otro mayor y que se puede ir conformando a través de la incorporación de módulos diferenciados a través de los cuales va ganando valor, da soporte a la idea de emplear una metodología ágil frente a una tradicional, y mas concretamente el eXtreme Programming.

4.1.1 Por qué escoger una metodología ágil

Tomando como referencia las bases que rigen toda metodología ágil y la comparativa de éstas frente a las metodologías tradicionales que se recoge en [Let06], se justifica por qué la elección se ha decantado hacia el enfoque ágil. Escoger una metodología ágil frente a una tradicional aporta un **extra de flexibilidad** deseable. Las primeras pueden estar en principio destinadas a proyectos mayores donde existe un riesgo más marcado debido a cambios o imprevistos. Sin embargo, aun no existiendo requisitos cambiantes e imprecisos como tal para el caso que se plantea, es deseable proyectar un desarrollo creciente que permita a cada paso discernir distintas vías para evolucionar el producto. Esto permite proyectar **periodos cortos de desarrollo** donde se realiza una planificación centrada en una parte concreta del producto, evitándose así la necesidad de plantear desde el primer momento una planificación global, estableciéndose únicamente objetivos primarios y la línea a seguir.

¹<http://www.extremeprogramming.org/>

Por otra parte se desea enfatizar en un enfoque centrado en la **comunicación continua**, ya que se considera preferible abordar la toma de decisiones de forma paulatina y progresiva, en lugar de definir todos los puntos desde el inicio, ganándose en adaptabilidad. Gracias a ello se puede instaurar una mecánica de trabajo más fluida, aprovechándose las reuniones periódicas establecidas con el equipo de desarrollo de la empresa para aunar dos objetivos: la revisión de resultados por un lado y discusión de problemas, toma de decisiones y planteamiento de objetivos por otro. Cabe destacar que pese a plantearse como un trabajo individual, la connotación de trabajo en equipo está presente. Es por ello que el enfoque ágil se adapta mejor a esta necesidad de interacción.

4.1.2 Aplicación de eXtreme Programming

Iacovelli definió en [IS08] un framework para la clasificación de las principales metodologías ágiles en base a cuatro vistas centradas en distintos aspectos. Desde la vista de “uso” que recoge las salidas beneficiosas que se obtienen de la aplicación ágil, hasta la vista denominada “capacidad de agilidad” que mide qué tanto la metodología cumple con los principios ágiles, pasando por las vistas de “aplicabilidad” y “procesos y productos”, centradas en analizar la adecuación a distintos entornos y qué actividades pueden ser cubiertas bajo la metodología respectivamente, se definen una serie de atributos binarios que deben o no cumplirse y que diferencian las metodologías.

A través de la evaluación de los atributos binarios este **framework comparativo** también sirve como herramienta para definir cuál es la metodología ágil que mejor se adapta a un contexto de trabajo y proyecto específico. De tal forma, se ha escogido eXtreme Programming en base al valor de los atributos asignados a esta en la tabla comparativa que puede consultarse en [IS08], página 97. De entre los atributos a destacar que son asociados a la metodología escogida están presentes: orientación a la productividad y cumplimiento de los requisitos, donde la fecha de entrega no es crítica, interés en el abordaje de iteraciones cortas, carácter colaborativo con intercambio de conocimientos, elevada interacción con el cliente, menor interacción con el usuario final, estructura jerárquica de roles no requerida, interés en la refactorización, el testing y la integración; todo ello para el contexto de un proyecto y equipo pequeños con un riesgo y complejidad no elevados. Así mismo presenta una perspectiva un tanto más alejada de la gestión, la documentación y el control de calidad para centrarse más en la definición de requisitos, modelado y codificación, así como los test y ejecutables como artefactos de salida.

En lo sucesivo se comentan los puntos clave o principios que caracterizan la metodología, particularizando y justificando cómo se han aplicado, o bien si algunos de ellos no han sido seguidos a rajatabla. En la figura 4.1 se ofrece además una panorámica de aquellos conceptos importantes involucrados cuando se trata de XP, destacándose en negro aquellos aspectos que han sido adaptados.

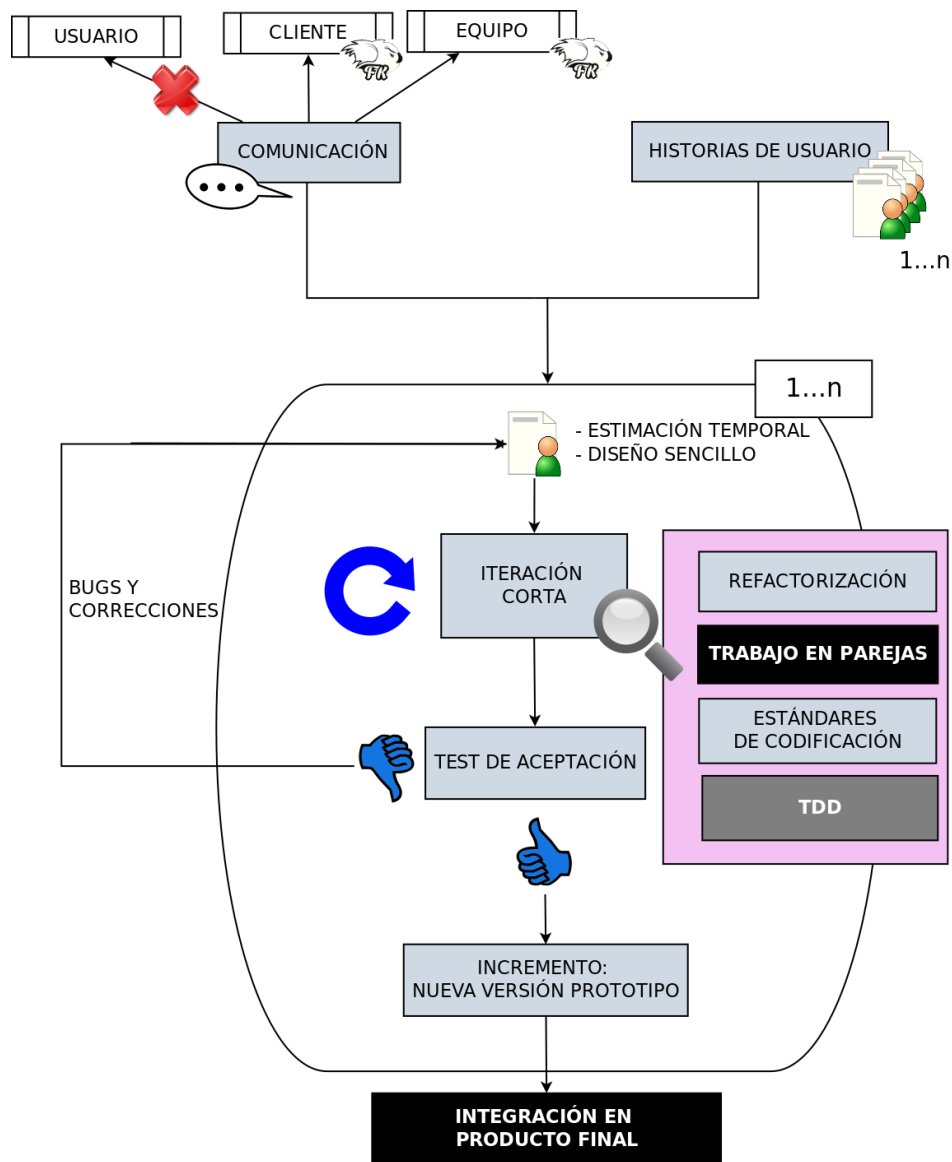


Figura 4.1: Panorámica de conceptos involucrados en XP

1. **Fomento de la comunicación.** Como ya se ha perfilado, se afronta un desarrollo individual pero que mantiene la connotación de trabajo en equipo, en tanto que se colabora con la empresa Furious Koalas S.L, y los resultados podrían llegar a ser integrados parcial o totalmente como módulo auxiliar en el juego *Whizz*. De tal forma, el contacto con la empresa pasa por la comunicación con un equipo pequeño y multidisciplinar de 4 desarrolladores, que actúan en conjunto en base a un doble rol. Actúan como cliente en tanto que son los dueños del producto que se va a ver beneficiado gracias al proyecto, pero a la vez actúan como equipo de soporte con el que interaccionar desde un punto de vista técnico. Así, se está cumpliendo ese principio por el cual el cliente debe estar siempre presente y disponible.

2. **Roles simplificados.** Según se aprecia en el anterior punto y dada la distribución de recursos de personal de la que se dispone, varios roles deben recaer en la misma persona. De esta forma, la persona encargada del desarrollo de QuPI-GAm será a la vez programador, encargado de pruebas y encargado de seguimiento. El conjunto del equipo de la empresa actuarán puntualmente como programadores en tanto que a ellos se recurre para realizar consultas específicas de carácter técnico (promoviéndose así el intercambio de conocimientos). El equipo en conjunto actúa igualmente como cliente y un representante de todos ellos asume el rol de gestor (*big boss*) como nexo con el equipo de Furious Koalas y de consultor (portando conocimiento específico acerca del juego Whizz).
3. **Proyección de iteraciones cortas.** Para favorecer el seguimiento y fomentar la comunicación se considera útil fijar iteraciones cortas de 1 o 2 semanas de duración, al cabo de las cuales se fija una nueva reunión. En esta se revisan y muestran los resultados conseguidos, se plantean correcciones o solución a incidencias detectadas o problemas que hayan frenado el avance. Asimismo se fijan los objetivos a alcanzar para la siguiente iteración. En estas reuniones se fomenta el intercambio de información útil entre los miembros participantes y el denominado *juego de planificación* es abordado conjuntamente. En XP se sugiere que el cliente establezca la prioridad de cada historia y los programadores el esfuerzo. En este caso se opta por definir ambos parámetros de manera conjunta. En eXtreme Programming, a diferencia de otras, al término de cada iteración no es estrictamente necesario entregar un ejecutable operativo cerrado, si no que se toleran revisiones posteriores si se observan deficiencias o se requieren mejoras sobre esa parte del sistema.
4. **Entregas.** En cada iteración se ha tratado de centrar el grueso de la actividad en una historia de usuario concreta, de tal forma que la suma de una serie de iteraciones den como resultado el desarrollo de un módulo específico, o una parte característica y diferencial del módulo que haga crecer el sistema, dándose así soporte a la idea de seguir un proceso de carácter incremental.
5. **Interés en la refactorización.** En XP se aboga por proyectar refactorizaciones de código, bajo la premisa de que todo diseño inicial puede llegar al punto de resultar inadecuado ante la evolución que siga el proyecto. Para el caso específico, el hecho de arrancar el desarrollo sobre tecnologías nuevas como AppEngine o Backbone puede favorecer el hecho de producir código mejorable durante el proceso de adaptación inicial, por lo que la refactorización resulta muy adecuada. Por otra parte, durante las fases de integración de código también puede ser requerida cierta adaptación de nomenclaturas o distribución de código y funcionalidades, para que la fusión de código resulte coherente.

6. **Proceso de integración continua.** Dado que al final de cada iteración se pretende obtener un incremento de valor y operativo, la integración es proyectada de forma continua en XP. Sin embargo este requerimiento se ha relajado, considerando como sistema en sí mismo QuPI-GAM. Se dejaría así la integración real en el sistema destinatario (el juego) para una fase más allá del ámbito de este proyecto, donde se puede planificar una integración en varios pasos, incorporando primeramente una versión mínima a la que se irían agregando funcionalidades.
7. **Orientación a testing.** El testing es considerado como elemento nuclear de esta metodología, ya que se aboga por un desarrollo basado en TDD. Si bien no se ha aplicado esta filosofía de desarrollo, sí que se han incorporado test unitarios y de integración.
8. **Seguimiento de estándares de programación.** Esta norma resulta de utilidad para cualquier tipo de contexto de trabajo. Así pues, desde un inicio se ha tratado de seguir el estándar impuesto en la organización con la que se colabora, siguiendo la estructura recomendada por el equipo y respetando las consignas de las guías de estilo para *javascript* y *python* sugeridas por Google.
9. **Programación en pareja.** Este principio no puede ser cumplido en tanto que el rol de programador recae en una única persona. Por lo tanto la limitación por recursos de personal requiere la supresión de este rasgo. Por esa misma razón no es de aplicación el principio por el cual cualquier parte de código puede ser modificada por cualquier integrante del equipo de programadores para evitar la existencia de imprescindibles.

4.2 Características hardware y software del desarrollo

A continuación se expone el detalle de los elementos software y hardware requeridos tanto en fase de desarrollo como para el despliegue de la aplicación.

Características Hardware

En primer lugar cabe definir cuál ha sido el contexto de desarrollo. Las características del equipo en el que se ha llevado a cabo se tienen en el cuadro 4.1. Puede observarse que este es un equipo con unas prestaciones estándar. No se ha requerido más ya que las necesidades de procesamiento no resultan ser elevadas. Para las pruebas realizadas en local el servidor es emulado en el propio equipo y la base de datos cuenta con un pequeño volumen de información. Así, agregando al equipo una conexión a Internet, se tienen todos los medios necesarios para poder afrontar el desarrollo.

Por su parte el despliegue se realiza empleando los servicios disponibles en la nube a través de un Paas (ver siguiente apartado) por lo que los servidores son parte de la infraestructura de un tercero. Con el fin de poder testear el funcionamiento de la aplicación también se requerirán en la fase de integración varios dispositivos móviles con diferentes SS.OO.

Version	
Kernel	Linux 3.19.0-15-generic (x86_64)
Distribution	Ubuntu 15.04
BIOS	
Date	01/08/2014
Version	83CN53WW(V3.00)
Board	
Name	Lenovo G505s
Vendor	LENOVO
Processors	
AMD A8-4500M APU with Radeon(tm) HD Graphics 1800,00MHz	
AMD A8-4500M APU with Radeon(tm) HD Graphics 1900,00MHz	
AMD A8-4500M APU with Radeon(tm) HD Graphics 1900,00MHz	
AMD A8-4500M APU with Radeon(tm) HD Graphics 1800,00MHz	
Memory	
Total Memory	7294996 kB
SCSI Disks	
ATA HGST HTS545050A7	
MATSHITA DVD-RAM UJ8DB	
Display	
Resolution	1366x768 pixels

Cuadro 4.1: Especificaciones del equipo empleado para el desarrollo

Características Software

En el cuadro 4.3 se recogen algunas de las principales herramientas empleadas de cara al soporte al desarrollo, así como en el cuadro 4.2 la relación de lenguajes y frameworks requeridos. De entre estos últimos se destacan y detallan brevemente aquellos que más han caracterizado el desarrollo: Backbone en el frontend y GAE en el backend.

	Cliente	Servidor
Frameworks	<ul style="list-style-type: none"> ■ Backbone ■ Bootstrap 	<ul style="list-style-type: none"> ■ WebApp2
Lenguajes	<ul style="list-style-type: none"> ■ Javascript + JQuery ■ CSS3 + HTML5 + Less 	<ul style="list-style-type: none"> ■ Python
Librerías	<ul style="list-style-type: none"> ■ Hello.js ■ Underscore.js 	<ul style="list-style-type: none"> ■ appengine.ext.ndb ■ appengine.api.search

Cuadro 4.2: Medios software requeridos por la aplicación

Tarea	Herramienta
Edición de código	<i>Emacs</i> como editor de texto para el código del cliente y <i>Geany</i> como entorno de desarrollo para el código del servidor, ambos bajo licencia GPL
Depuración	Para la depuración del código del cliente se emplearán las herramientas para desarrolladores incorporadas en el navegador Google Chrome
Organización	Como sistema de control de versiones se empleará <i>Mercurial</i> , y como sistema web para la gestión de código <i>Bitbucket</i> , teniendo este último la ventaja de no imponer limitación en el número de repositorios que se pueden crear ni en el tamaño. Por otro lado, se empleará <i>Trello</i> como herramienta para la gestión de las tareas y seguimiento del estado del proyecto.
Documentación	Para generar la documentación se empleará <i>LaTeX</i> , junto con el editor <i>TexMaker</i> . Para la elaboración de diagramas, se empleará <i>ArgoUML</i> (bajo licencia BSD), <i>Dia</i> (bajo licencia GPL) y <i>StarUML</i> para la generación de diagramas.
Utilidades	Para testear los recursos expuestos en el servidor con independencia de la interfaz de cliente, se empleará el cliente REST <i>Postman</i> incorporado en el navegador Google Chrome. Por otro lado, para testear el diseño <i>responsive</i> de las interfaces en un dispositivo móvil, se empleará la extensión <i>Responsive Inspector</i> del navegador Google Chrome.

Cuadro 4.3: Software de soporte al desarrollo

AppEngine En el marco de los servicios en la nube ofrecidos por Google Cloud Platform, Google App Engine se postula como una **alternativa en forma de Paas** para el alojamiento de aplicaciones web o backends móviles. Destacan los siguientes aspectos:

- Se pone a disposición de los desarrolladores la **infraestructura de Google**, así como una serie de servicios incorporados que permiten abstraerse de la configuración y mantenimiento de los servidores. Cuestiones como la escalabilidad y la disponibilidad pasan a ser gestionadas de forma automática según el tráfico.
- También se ofrecen otros **servicios integrados** y APIs que dan soporte al desarrollo, lo que en conjunto permite que el desarrollador se centre en la aplicación propiamente dicha. De entre esos servicios, cabe citar la base de datos NoSQL y el API de búsqueda que ofrece, ya que han sido intensamente usadas.
- Ofrece un modelo de desarrollo basado en una aproximación de tipo *sandbox*. El SDK incorpora un entorno emulado donde se simula casi al completo el entorno del AppEngine real, que permite testear la aplicación localmente durante el desarrollo. A medida que se avanza, distintas versiones de la aplicación pueden ser desplegadas fácilmente en el entorno real, que es accesible de forma **gratuita bajo unos recursos mínimos**. Una vez la aplicación es desplegada y está operativa, se dispone de una consola desde la cual se puede monitorizar toda la actividad generada (excepciones lanzadas, datos generados y almacenados en la base de datos, etc.). El libro [San15] supone un buen recurso para el desarrollo con AppEngine.

Backbone.js Es un framework que permite organizar el código por el cual se rige el comportamiento del frontend. Constituye una extrapolación del patrón MVC y permite estructurar y separar la parte relativa a la carga de las plantillas HTML, el manejo de los eventos generados, la lógica gestionada con JavaScript y la comunicación y sincronización con el servidor. Para ello se compone de 4 unidades básicas o componentes:

- **Modelos.** Los datos asociados a la lógica de la aplicación son estructurados como modelos. Se facilita la creación y destrucción de instancias, así como el establecimiento de criterios de validación previos a la creación. Si existe una correlación entre las instancias manejadas en el lado cliente y servidor, puede configurarse una sincronización por la cual todo cambio producido en el frontend sea replicado en el backend.
- **Colecciones.** Las colecciones permiten gestionar de forma conjunta una serie de instancias diferentes de un modelo dado, lo que permite emprender acciones concretas que pueden ser necesarias cuando se trata de manejar una lista de elementos.
- **Vistas.** La gestión de la interfaz se realiza a través de las vistas. Cada vista se asocia a una plantilla y a través de ella se gestionan los aspectos relativos al renderizado y al manejo de los eventos generados a causa de la interacción con el usuario.

Además de los eventos generados de parte del usuario, backbone considera otro tipo de eventos a nivel interno que facilitan la coordinación de las distintas partes (modelo/colección frente a vista). Esto permite implementar el **patrón Observador** de forma sencilla. Al igual que un evento disparado por una acción del usuario en la interfaz puede ocasionar la actualización de una instancia de un modelo determinado, el procesamiento interno que se desencadene en la aplicación puede implicar la actualización de los modelos o colecciones. En ese caso backbone permite generar eventos de cambio, de tal forma que cada vista afectada pueda capturarlos y pueda volver a renderizarse para mostrar los nuevos datos.

- **Rutas.** Tradicionalmente cada recurso solicitado al servidor era requerido a través de una url diferente. Sin embargo ahora al emplearse páginas generadas de forma dinámica mediante peticiones AJAX, se solicita de inicio una página a través de una única url que no va a cambiar a pesar de los cambios que se vayan produciendo o las zonas a las que se acceda. La definición de rutas permite vincular o enlazar ciertas localizaciones o contenidos dentro de la página a urls, emulando así el comportamiento tradicional. Para ello cuando se accede a cierta zona, se cambia la url mostrada en el navegador y cuando se introduce una ruta específica se actualizan tanto modelos como vistas para mostrar lo que corresponde. De esta forma se pueden usar marcadores o compartir un cierto estado de la página, así como recurrir a la navegación atrás/adelante.

Capítulo 5

Arquitectura QuPI-GAM

ESTE capítulo está dedicado al análisis de las distintas partes de las que se compone el proyecto. Se establece para ello una distinción de componentes orientada a las funcionalidades incorporadas, aunque considerándose también aspectos de base no funcionales necesarios para constituir la base del sistema web. En torno a este se centrará la descripción. Para cada parte, funcionalidad o componente descrito se especifica a que módulo pertenece, con qué historia de usuario está vinculado (las historias de usuario se incluyen en la sección 5.13), una pequeña descripción de la problemática que se afronta y de qué manera se pretende resolver, incidiendo finalmente en el detalle de los aspectos de diseño considerados en su resolución.

Al final del capítulo se incluyen una serie de secciones especiales que complementan las anteriores. Por un lado y con la finalidad de incrementar la comprensión global del proyecto, en el apartado 5.11 se establece un marco que permite comprender mejor la organización del proyecto a un nivel de abstracción menor que el que se encontrará en las anteriores secciones. Este está un tanto más centrado en la parte cliente y en la conexión entre cliente y servidor. Por otro lado, el apartado 5.12 permite tener una mejor comprensión acerca de la base de datos empleada, recopilando además aquí la relación de modelos definidos. Estas dos secciones pueden tomarse como referencia a medida que se vaya leyendo el capítulo, ya que en ciertos puntos se aludirá a su contenido.

5.1 Visión general de la arquitectura

Para introducir y desarrollar la arquitectura conviene seguir un **enfoque top-down**, según el cual se presentan a alto nivel de abstracción los componentes del sistema. De esta forma se ofrece una panorámica de las partes que en conjunto dan entidad al sistema diseñado y desarrollado. La delimitación de los componentes se ha establecido definiendo las funcionalidades de base que dan sentido a QuPI-GAM, fruto de lo cual se obtienen tres módulos que puede encontrarse en la figura 5.1.

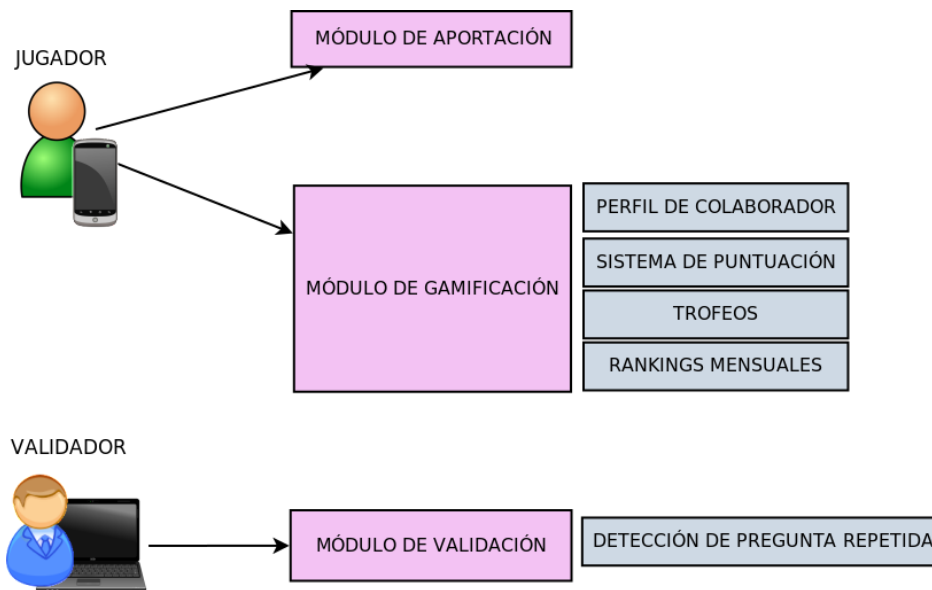


Figura 5.1: Representación visual de los principales componentes incluidos en QuPI-GAM

Lo primero que puede apreciarse es que **no todo el sistema está dirigido a un único usuario**. Se tienen dos tipos. Por un lado se tiene a los jugadores del trivial en que de forma futurible debería integrarse QuPI-GAM y que adoptan el papel de colaboradores. Por otro lado se tendría a personal seleccionado, que debe ponerse en la piel del validador. Esto ya da a entender que habrá que incorporar un mecanismo que autorize el acceso a ciertas áreas de la aplicación en función del rol.

Módulo de aportación. Si se pone el foco en el módulo de aportación, se está centrando la atención en el módulo más sencillo pero que constituye la base y esencia a nivel funcional de QuPI-GAM, ya que es el encargado de gestionar que los jugadores puedan sugerir sus propias preguntas. Incorpora algunas particularidades, como la introducción de un mecanismo de protección frente a bots, que se discutirá más adelante en detalle. A grandes rasgos, el resultado de incorporar este módulo supone poder conseguir información (de valor o descartable) por parte de los jugadores.

Módulo de validación. Otro módulo fundamental y necesario para complementar al anterior es el de validación. A través de este se establece un tamiz por el cual deben pasar todas y cada una de las preguntas sugeridas, para garantizar que tienen una calidad mínima que permite introducirlas en el juego. Se espera que la validación, al efectuarse de forma manual por expertos en las distintas materias, sea muy consistente. Sin embargo para llevar este módulo un paso más allá se incorpora un cierto procesamiento automatizado que da soporte a la validación manual revisando un aspecto que el validador por sí mismo no podría afrontar, esto es, tratar de detectar si una pregunta pendiente de validación tiene una temática muy similar a otra que ya se tiene como definitiva (y por tanto no interesaría, porque implicaría tener la misma pregunta duplicada).

Módulo de gamificación. Todo lo anterior asienta una base que se lleva un paso más allá gracias a la introducción de gamificación. Dado que el jugador de por sí no tiene por qué tener interés en colaborar y hacer crecer el juego, se le debe ofrecer algo a cambio que le enganche y que incluso le haga considerar este módulo como otra dinámica que le enganche y le divierta. A tal efecto a cada jugador se le asigna un perfil de colaborador, como punto central donde aglutinar todo lo relativo a gamificación, desde el sistema de puntuación creado hasta la consulta de los trofeos adquiridos y su posición en los distintos rankings existentes.

Aunque en la sección 5.11 se incluye un diagrama de despliegue que da una mejor perspectiva de lo que implica un desarrollo de este tipo, es importante poner de relieve la complejidad asumida al optar por un desarrollo de tipo cliente-servidor. Hay que manejar tecnologías web, distintos lenguajes, frameworks y herramientas de soporte para cada uno de los dos extremos, así como decidir cuál es la mejor alternativa para constituir el servidor y gestionar el almacenamiento persistente, que para el caso concreto se ha resuelto recurriendo a un IaaS, concretamente a App Engine.

5.2 Autenticación del usuario

Módulo	Funcionalidad base
Identificador	US_SUPPORT_01 (cuadro 5.7)
Problemática	De cara a aplicar la gamificación, se necesita vincular a cada jugador la información relativa a la actividad que ha desempeñado respecto a sus aportes. Para ello es necesario asignar a cada uno un usuario en el sistema. Y para que el jugador pueda acceder a su usuario se necesita incorporar un mecanismo de autenticación que le permita identificarse.
Enfoque	Implementar un mecanismo propio para la autenticación implica gestionar el almacenamiento de credenciales, una cuestión delicada que requiere poner atención extra en lo que a mecanismos de seguridad se refiere. Con el fin de sortear este aspecto de una forma sencilla pero eficaz, se ha optado por recurrir a la autenticación con terceros. De esta forma haciendo uso del protocolo de autorización OAuth2 y de la librería hello.js se delega todo el proceso de inicio de sesión en los proveedores escogidos: Google y Facebook.

5.2.1 Identificación a través de Google + y Facebook

Se da la posibilidad a los jugadores de que se autentiquen si desean que su actividad quede registrada y poder disponer de un perfil a través del cual recibir todo el feedback relativo a la parte colaborativa del juego. En lugar de incorporar un mecanismo de autenticación propio, se ofrece la posibilidad de iniciar sesión en el sistema a través de una cuenta de Facebook o Google +. Si el jugador ya dispone de una cuenta de Facebook y/o Google + este evita tener que crear nuevas credenciales de acceso mediante un registro. Se le ofrece así un mecanismo de inicio de sesión sencillo, rápido y que le resulta familiar.

Elección de proveedores: Si los jugadores no disponen de una cuenta con los proveedores elegidos, este puede resultar un factor de rechazo, ya que de inicio puede no compensarles tener que crear dicha cuenta de forma externa al juego. Es por ello que se han escogido dos redes con una elevada **popularidad**, por lo que se asume que la mayor parte de jugadores las conocerán y serán poseedores de una cuenta. Por otro lado, dado que al contar con un servicio de terceros se depende de la **disponibilidad** que estos ofrezcan, se han escogido servicios que a día de hoy ofrecen ciertas garantías y se espera que siga siendo así durante mucho tiempo.

Autorización empleando OAuth2

OAuth2 es un protocolo de autorización gracias al cual un usuario (el jugador en este caso) puede permitir a un tercero (que sería QuPI-GAm) el acceso a ciertos contenidos protegidos bajo credenciales gestionados por un servidor de recursos de confianza (en este caso Facebook y Google +) sin necesidad de que el usuario tenga que facilitar sus credenciales al tercero para que este pueda acceder a dichos contenidos. En este caso interesa aprovechar en sí mismo el mecanismo de autenticación que ofrecen estos servicios, accediéndose además al *nick* que ese usuario tenga asignado en cada proveedor.

De inicio y para poder hacer uso de este mecanismo, es **necesario registrar el dominio de QuPI-GAm** en cada uno de los proveedor de recursos a los que se quiere acceder de forma independiente. De esta forma se da de alta un cliente que obtiene un identificador y una clave secreta, gracias a las cuales QuPI-GAm puede identificarse como cliente válido frente al proveedor. Una vez solventado este aspecto de configuración, se está en disposición de llevar a cabo el procedimiento. A continuación se expone de forma simplificada y se representa en la figura 5.2 el flujo de comunicación y los pasos que se producen entre los tres actores implicados, siendo estos el proveedor (Google o Facebook) y el cliente que solicita acceso (QuPI-GAm) diferenciándose en este al agente de usuario que representa la parte de frontend (donde se localiza el jugador) y la aplicación del lado servidor:

- El jugador escoge uno de los proveedores e inicia el procedimiento de inicio de sesión.
- En ese momento desde el agente de usuario se envían en primer lugar al proveedor las credenciales asignadas a QuPI-GAm como cliente. En caso de ser reconocido como cliente válido continúa el proceso.
- Desde el agente de usuario el jugador accede al servicio ofrecido por el proveedor, con el que interactúa introduciendo sus credenciales.
- El proveedor recibe las credenciales del jugador. En caso de que la autenticación de los parámetros recibidos sea correcta y de que el jugador manifieste su conformidad para otorgar a QuPI-GAm los permisos que se solicitan, el proveedor envía al agente de usuario un *token* que representa la sesión iniciada en este.

- Desde el agente de usuario se envía el token al servidor de QuPI-GAm. En este punto es necesario validar el *token* recibido para garantizar que fue generado por el proveedor y no desde el extremo cliente por una entidad desconocida. Para ello desde el servidor se envía una petición al proveedor donde se incluye la clave secreta asignada, a través de la cual este confirma su validez. En caso de ser válida, se reciben los datos del jugador a los que se tendría acceso. Para el caso solo interesa su nick en la red social.
- En este punto, habiéndose confirmado que el jugador es quien dice ser, se puede iniciar sesión dentro de la propia aplicación. Para culminar el procedimiento, desde el agente de usuario se envía una petición al proveedor para que se cierre la sesión recién creada en el servicio que ofrece el proveedor, ya que una vez completado el login no se requerirá solicitarle más información.

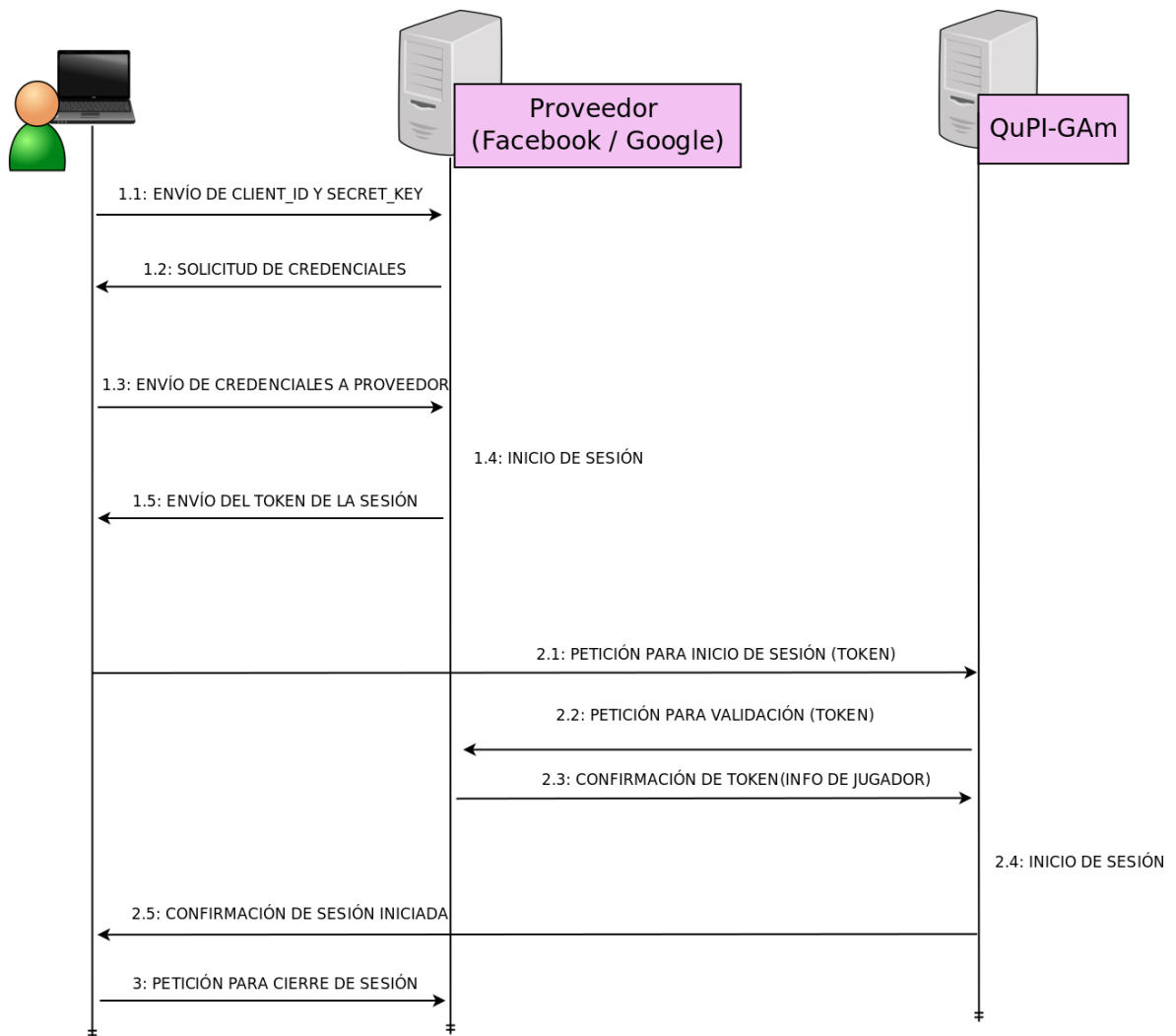


Figura 5.2: Comunicación establecida durante la autenticación

Detalles de implementación: Hello.js

Se destacan los aspectos característicos de implementación tanto en la parte cliente como en el código del servidor:

- Parte cliente. Una buena parte del procedimiento de autenticación es responsabilidad del frontend. Con el fin de facilitar la interacción con los proveedores se ha recurrido a Hello.js, un SDK escrito en Javascript que facilita y **unifica la comunicación con las distintas APIs REST** expuestas por cada uno de ellos para la autenticación mediante OAuth. Cada proveedor ofrece una API diferente, por lo que si se recurre directamente a ellas es necesario implementar distintas versiones para cada una. Hello.js ofrece una serie de funciones y define una serie de eventos que estandarizan la gestión de las llamadas, sea cual sea el proveedor (para los que ofrece soporte).
- Parte servidor. Aunque el manejador para la petición de inicio de sesión lleva a cabo más acciones, en lo que respecta a la parte de autenticación únicamente cabe destacar el hecho de que desde éste se **envía una petición de tipo GET al servidor del proveedor**, gracias a la cual se comprueba que el *token* recibido es válido. Entre otros aspectos se comprueba que efectivamente ha sido generado por el proveedor y que no haya expirado. En la figura 5.3 se recoge un diagrama de secuencia donde se muestran las llamadas que se producen entre los tres extremos.

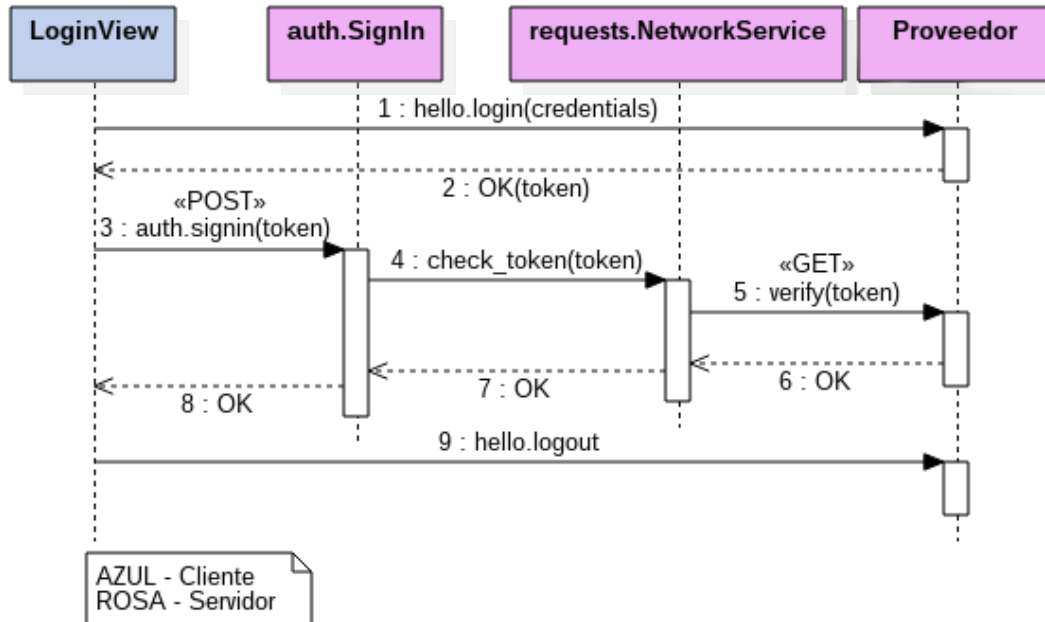


Figura 5.3: Diagrama de secuencia para llamadas principales durante la autenticación

5.3 Soporte para la creación de usuarios y gestión de la sesión

Módulo	Funcionalidad base
Identificador	US_SUPPORT_01 (cuadro 5.7)
Problemática	Dado que se pretende aplicar gamificación, debe poder hacerse un seguimiento del usuario que permita registrar su actividad. Para ello es necesario poder tenerlos identificados en el sistema, para lo que se necesita incorporar en primer lugar un mecanismo de autenticación (lo que se discute en la sección 5.2), la posibilidad de crear usuarios dentro del sistema y también un mecanismo que permita asignar una sesión a cada usuario identificado. Así mismo, hay que establecer unas fronteras de acceso, ya que no todos los servicios expuestos han de ser accesibles a los jugadores.
Enfoque	Aprovechando los recursos que ofrece WebApp2 como framework empleado para la parte asociada al backend, se recurre a sus módulos <i>auth</i> ¹ y <i>sessions</i> ² con el fin de incorporar una solución que permita asignar un usuario a cada jugador, definir unos roles que permitan limitar el acceso y ofrecer la posibilidad de iniciar sesión en el sistema.

Configuración de base

Para constituir la base de configuración del servidor en lo que a manejo de peticiones y gestión de sesiones se refiere, se han incorporado una serie de clases de base (ver figura 5.4) que hacen uso de los módulos *auth* y *sessions* ofrecidos por el framework.

5.3.1 Creación de usuarios

Dado que la autenticación se gestiona mediante terceros (véase sección 5.2) no es necesario almacenar las credenciales de los usuarios (esto es, una contraseña vinculada a cada usuario). Sin embargo sí que es necesario crear un nuevo usuario para cada nuevo jugador autenticado en el sistema. A tal efecto se define un **modelo *User*** (incluido en la figura 5.24) que hereda de la clase homónima *auth.models.User*. La información almacenada es:

- **OID.** El identificador del usuario en el sistema se construye de la siguiente forma. Se constituye de dos partes. La primera es el nombre de la red social con la que se autentica a modo de prefijo. La segunda es el identificador que le ha asignado la propia red social y que es devuelto al servidor cuando este realiza la validación del *token*.
- **Nick.** Extrapolado según su nombre registrado en el proveedor para la autenticación.
- **Rol.** De cara a implementar la limitación de acceso a ciertas áreas existen dos roles asignables, el rol de aportador y el de validador.
- **Perfil.** De forma automática se crea un nuevo perfil de colaborador y se le asigna. Para más detalles acerca del perfil, ver la sección 5.8.
- **Idioma.** Se almacena cuál es el idioma asignado al jugador.
- **Localización.** Se almacena la cadena de ubicaciones que permiten posicionarlo.

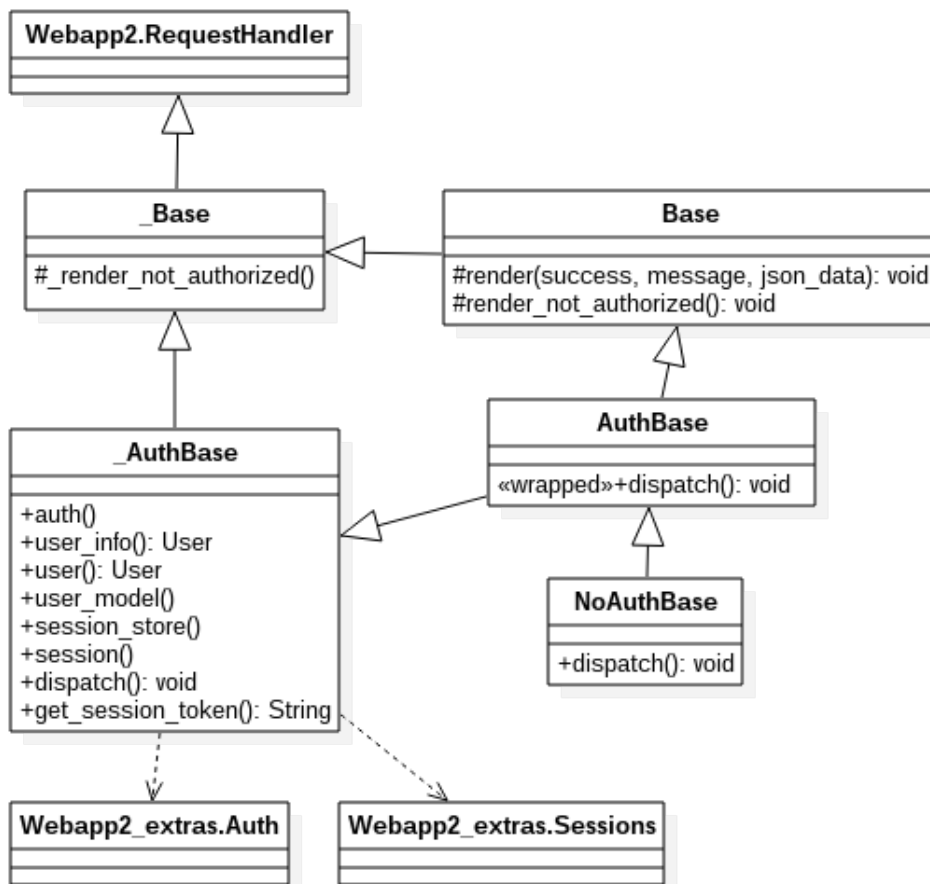


Figura 5.4: Clases para constituir el soporte de base en el servidor

Acerca de la localización Se ha considerado un diseño que maneje la localización como pieza de información disponible. Se emplea esta información, pero no se ha incorporado el mecanismo necesario para extraer el dato del usuario. Dado que no se le quiere pedir información al usuario, sería necesario emplear una librería que permitiese extraer la ubicación a través del dispositivo del jugador. Una buena alternativa para implementarlo sería emplear el API Geocoding^a de Google Maps como servicio de geocodificación inversa. Sin embargo se ha asumido que este mecanismo estará presente en el sistema final donde debiera integrarse QuPI-GAm.

^a<https://developers.google.com/maps/documentation/geocoding>

Agilizando el registro. Es importante destacar que se ha apostado por agilizar al máximo el procedimiento que permite al jugador registrarse en el sistema. Es por ello que en ningún momento se le pide introducir ningún tipo de información. La información que se requiere (que es mínima) se extrae directamente desde el proveedor para la autenticación. Siguiendo con este concepto minimalista, no se distinguen como dos procedimientos separados el registro de un nuevo usuario en el sistema o el inicio de sesión de un jugador conocido. Ante cada nueva petición de inicio de sesión se comprueba si el jugador es o no reconocido. Si no se reconoce, se inicia el procedimiento para la creación.

5.3.2 Procedimiento para la gestión del inicio y cierre de sesión

Habiéndose explicado ya el mecanismo establecido para la autenticación y las consideraciones necesarias para la creación de nuevos usuarios en el sistema se puede definir completamente cómo se ha gestionado la incorporación del inicio de sesión en el sistema:

1. Recurso **/auth/signin** Cuando el jugador manifiesta su intención de iniciar sesión se solicita este recurso. Si el jugador inicia sesión por primera vez, se le crea y asigna un usuario. En caso contrario, se recupera su usuario correspondiente. Una vez disponiendo del usuario, se crea una nueva sesión en la que se almacena la información del usuario. De esta forma para las sucesivas peticiones que este envíe, en cualquier momento se podrá recuperar su información asociada.
2. Recurso **/auth/signout** Cuando el jugador manifiesta su intención de cerrar sesión se solicita este recurso. Únicamente es necesario recuperar su sesión y eliminarla, así como eliminar las cookies que se hubiesen creado durante el transcurso de la sesión. En caso de que el jugador abandone la página sin cerrar la sesión, esta quedará almacenada en el servidor y no se destruirá. Sin embargo no es problema ya que la sesión dispone de un parámetro configurable, el *session_max_age* que permite especificar el tiempo en segundos para que esta expire.
3. Recurso **/auth/session** Cuando el jugador accede a la página lo primero que se hace es enviar una petición hacia este recurso, para comprobar si este jugador ya tiene una sesión abierta. En caso afirmativo se devuelve la información necesaria para actualizar la interfaz y mostrar que tiene una sesión abierta.

El diagrama de secuencia de la figura 5.5 muestra una recreación de las llamadas principales donde el jugador accede, inicia sesión y cierra sesión. Este diagrama viene a completar el extracto mostrado para la autenticación en la figura 5.3.

5.3.3 Autorización de acceso

Dado que se ha determinado de inicio que la comunidad de jugadores solo pueden desenvolverse en lo que a colaboración se refiere aportando preguntas y no participando de su validación, es necesario restringir el acceso a esta segunda área y los recursos expuestos para darle soporte.

Asignación de rol

Como se adelantaba en la descripción del modelo creado para representar la información de cada usuario en la sección 5.3.1, a este se le pueden **asignar dos roles**, el de aportador y el de validador. Cuando se crea un nuevo usuario en el sistema al realizarse un nuevo registro, se le asigna únicamente el **rol de aportador por defecto**.

Por otro lado, dado que los validadores serían un grupo de personas seleccionadas para tal fin, estos son creados de forma separada como usuarios de prueba por defecto para imitar lo esperado en el sistema destinatario, asignándoles a estos usuarios el **rol de validador**. Esto significa que no se ha implementado un mecanismo de registro alternativo explícito para inscribir a los validadores.

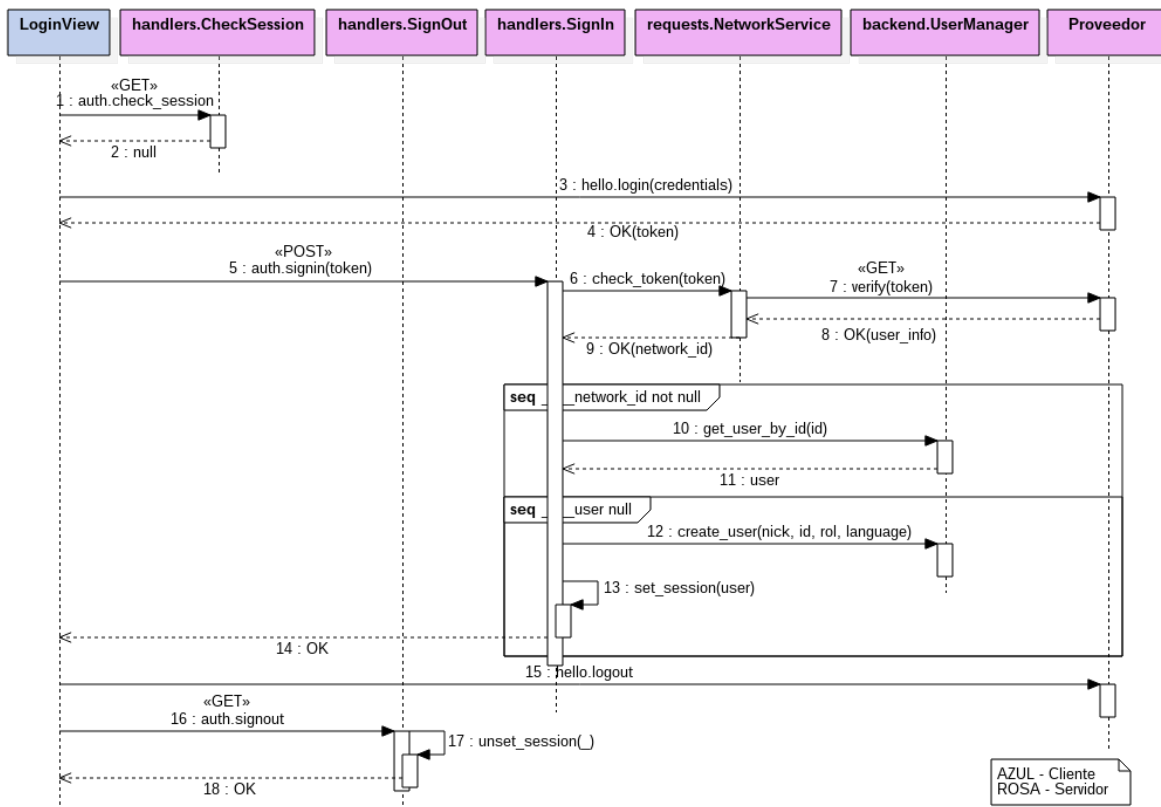


Figura 5.5: Diagrama de secuencia con el flujo de tratamiento para la sesión

Discriminación de usuarios en base a su rol

Se tiene un mecanismo de autenticación y un modelo de datos compartido por aportadores (los jugadores) y validadores (personas escogidas para ello). Esta unificación facilita la tarea de determinar si un usuario dado que quiere acceder a un área tiene permiso o no. Lo que hay que considerar es lo siguiente:

- Determinar si se requiere autenticación.** Para cada clase encargada de atender la petición hacia un recurso concreto, hay que determinar si esta ha de solicitar autenticación. Existen recursos que pueden ser solicitados por jugadores no identificados (área de aportación) otros que solo están accesibles para jugadores identificados (área del perfil) y otros que solo son accesibles por validadores (área de validación).

Para satisfacer este punto basta con crear dos clases: *AuthBase* y *NoAuthBase* (véase figura 5.4). Aquellas clases que gestionen peticiones a recursos que no requieren autenticación han de heredar de la segunda, que no debe realizar ningún tipo de comprobación. Para el caso opuesto deben heredar de *AuthBase*. Esta deberá sobrescribir la función encargada del despacho de peticiones.

- **Determinar qué rol se requiere en cada caso.** En caso de que se requiera autenticación para el acceso a un recurso, hay que incorporar un mecanismo que permita especificar qué tipo de rol es el que se requiere.

Sabiendo que cuando un usuario (sea jugador o validador) inicia sesión su información asociada está disponible en todo momento a través de la sesión asignada, incluido su rol, basta con **acoplar un wrapper** a la función de base encargada de despachar cada petición. Este *wrapper* ha de recuperar la sesión (si existe), desde esta recuperar el usuario asignado y finalmente contrastar si el rol de este coincide con el asignado a ese *wrapper*. En caso de que no exista sesión se debe informar de que el acceso está protegido. En caso de que sí exista sesión, si no hay coincidencia de roles se informa de que no se tienen los permisos requeridos.

5.4 Soporte a la internacionalización

Módulo	Funcionalidad base
Identificador	US_SUPPORT_02 (cuadro 5.8)
Problemática	Dado que se espera llegar a público de diferentes países es necesario poder mostrar la aplicación en diferentes idiomas
Enfoque	La aplicación dará soporte a dos idiomas: español e inglés. Se incorporará un diseño que permita agregar nuevos idiomas únicamente incluyendo las cadenas correspondientes. Se ha optado por diseñar una solución propia sencilla, en lugar de emplear marcos como que incorpora el propio framework a través de su módulo <code>i18n</code> .

5.4.1 Detección del idioma

Para poder mostrar la interfaz de acuerdo al idioma presente en el dispositivo del usuario es necesario detectarlo y almacenarlo. Para ello se establece el siguiente flujo:

- Cuando se solicita el acceso a un recurso que devuelve una plantilla con la definición de una interfaz, la vista principal asociada se encarga de **detectar el idioma a través del navegador**. Si no está soportado se define como predeterminado el inglés.

Cookie para el idioma Dado que la información del idioma se detecta en el lado cliente, se crea una cookie donde se almacena el idioma asignado para que esta esté disponible en el servidor

- Cada vez que el usuario inicia sesión, en el servidor se comprueba cual es el idioma detectado y si es distinto al que se le había asignado en sesiones previas **se actualiza**. Puede pasar que un usuario acceda a través de distintos dispositivos. Si cada uno tiene un idioma distinto definido, esto permite que el idioma se adapte de forma dinámica. Una vez iniciada la sesión, la información del idioma quedará almacenada y estará accesible a través de la propia sesión, por lo que la cookie ya no es necesaria. Sin embargo es necesario crearla para dar soporte a aquellos jugadores que realicen aportaciones anónimas, ya que a estos no se les asigna sesión.

5.4.2 Almacenamiento de las diferentes cadenas de texto

Una vez se conoce en qué idioma hay que devolver las cadenas de texto hay que establecer un marco que permita almacenar conjuntamente todas ellas y que permita recuperar en cada caso la cadena pertinente de una forma sencilla. Se tienen dos tipos de cadenas:

- **Archivos de constantes y mensajes.** Tanto en el cliente como en el servidor existen una serie de archivos que almacenan las cadenas predefinidas que se muestran por pantalla, así como los mensajes de información o error devueltos. Para estos casos se recurre a la definición de **diccionarios**. Cada cadena pasa a ser así un diccionario donde la correspondencia clave-valor se vincula al código del idioma y su texto asociado. Así para seleccionar una unidad de texto basta con seleccionar el nombre de su diccionario asignado, recuperando el idioma para definirlo como clave. Esto implica que la incorporación de un nuevo idioma solo implica agregar una nueva entrada en cada diccionario definido.
- **Información almacenada en el Datastore.** Información de base para el juego como son las categorías necesita de su traducción correspondiente a cada idioma. Asimismo, la información más importante del sistema, que son las preguntas, han de ser traducidas. Esto implica que en el Datastore hay que incorporar un esquema que permita almacenar y recuperar las distintas cadenas de texto vinculadas a estas piezas de información. Para ello se define el **modelo Translation** (véase figura 5.24). Cada entidad almacena el par idioma/texto. Así cada entidad en el Datastore que albergue texto que requiere ser almacenado en distintos idiomas, en lugar de definir una propiedad de tipo cadena, define una propiedad que almacene una lista de claves que apunten a las entidades de tipo *Translation* correspondientes, una por cada idioma.

Manejo de información en Datastore según modelo Translation El hecho de incorporar esta infraestructura puede condicionar ciertos aspectos de implementación asociados a la recuperación de esta información. Desde consultas que involucren estas cadenas de texto hasta la identificación universal de una categoría seleccionada dado su nombre (sabiendo que cada categoría tendrá más de un nombre, uno por cada idioma), pasando por el empleo del API de búsqueda para la detección de preguntas repetidas o la identificación y almacenamiento de una misma pregunta validada o traducida a un idioma concreto. Sin embargo estas vicisitudes se comentarán a medida que se expliquen cada uno de estos aspectos.

5.5 Aportación de preguntas

Módulo	Aportación
Identificador	US_UPLOADER_01 (cuadro 5.9)
Problemática	Se desea ofrecer al jugador la posibilidad de colaborar con el juego aportando sus propias preguntas. A tal efecto se necesita incorporar un mecanismo a través del cual este pueda introducir y enviar la pregunta.
Enfoque	Se creará un formulario que pueda rellenar tanto un jugador anónimo como uno identificado. Las preguntas recibidas se almacenarán temporalmente como preguntas pendientes de validación.

5.5.1 Constitución del formulario

Lo primero a considerar cuando se ha de constituir un formulario es la información de entrada que ha de solicitársele al jugador, lo que se recoge en el siguiente listado:

- **Idioma.** Se muestra un desplegable para que el jugador especifique en qué idioma está introduciendo la pregunta de entre los soportados por el sistema. Por defecto el desplegable tendrá fijado el idioma asociado al jugador, ya que se espera que en el mayor número de casos de forma natural exista esa correlación.
- **Categorías.** En los juegos de preguntas, estas son clasificadas o agrupadas en torno a su temática, definiéndose para ello una serie de categorías padre generales y especificando estas a través de subcategorías a modo de jerarquía. No sería práctico solicitar al jugador que clasifique su pregunta para la subcategoría específica correspondiente, ya que puede no estar familiarizado con todas las existentes, le resultaría tedioso y podría equivocarse. Ante este hecho puede optarse por dos opciones:
 - No especificar categoría. Una opción sería no pedirle que clasifique la pregunta y que esto recayese enteramente en los validadores. Sin embargo esta opción ha sido descartada.
 - **Especificar solo la categoría general.** Otra opción, la incorporada, pasa por pedir al jugador que seleccione en qué categoría encaja mejor la pregunta, pero solo

de entre aquellas categorías de primer nivel. Por lo general este grupo de categorías, al ser general y englobar a otras no será muy numeroso, por lo que el jugador no tendrá que elegir entre muchas opciones. La ventaja de requerir este pequeño esfuerzo al jugador es que en lugar de disponer de una gran amalgama de preguntas por revisar, estas se tendrán separadas. Si se tienen validadores expertos en ciertas áreas, cada uno de ellos podrá recuperar únicamente aquellas preguntas asociadas a su temática de forma sencilla.

Visualización de categorías. En cuanto a su incorporación al formulario, teniendo en cuenta que se quiere ahorrar espacio en la pantalla (de cara a la visualización en dispositivos pequeños), se ha optado por incorporar un **carousel** que muestra en cada pasada una opción de las disponibles. Así, basta con que el jugador vaya pasando opciones hasta encontrar una categoría donde cuadre la pregunta. Se asume que no existirán muchas categorías que mostrar, no más de 8, por lo que en los primeros usos el hecho de que el jugador no pueda visualizar todas las categorías juntas no debería ser problemático. Lo que es más, cuando el jugador emplee el formulario de forma repetida una serie de veces, entrará en juego la memoria a largo plazo, y podrá recordar sin esfuerzo todas las categorías.

- **Pregunta.** El jugador debe proponer el enunciado de una pregunta, junto con la respuesta correcta y tres opciones más incorrectas, todo ello en formato texto.

Revisión mínima del contenido aportado

Cuando el jugador desencadena el evento para que la pregunta sea registrada, antes de requerir al servidor el recurso encargado de su recepción y almacenamiento se aplica un **procedimiento de validación mínima**. Con esto se consigue que evitar consumir recursos del servidor en caso de que el contenido no cumpla unos requisitos mínimos necesarios para considerarlo como pregunta válida. En caso de no superarse alguno de los filtros, se informa convenientemente al jugador del fallo mediante un mensaje textual y destacando el área de conflicto. Los filtros definidos son los siguientes:

- Se comprueba (en caso de ser necesario) si ha sido **superada la prueba** establecida para verificar que la pregunta está siendo introducida por un humano. Este aspecto se detalla en la sección 5.5.3.
- No se establece como requisito que el jugador se haya autenticado, por lo que **se permiten preguntas de un jugador anónimo**.
- En el lado cliente se trata de construir un objeto de tipo *UserQuestion* a raíz de los campos introducidos en el formulario. Este modelo incorpora una **función de validación** que se encarga de aprobar o denegar su construcción en base a:

- No puede quedar **ningún campo vacío**.
- No se permiten **campos repetidos**.
- Se impone una **limitación en el número de caracteres** máximos requeridos por campo. Esto garantiza una correcta visualización en el espacio asignado en dispositivos móviles.
- El texto del enunciado de la pregunta debe incluir **al menos 3 palabras y de una longitud mínima** en términos de caracteres, ya que con menos no se puede contruir una pregunta con sentido.

5.5.2 Almacenamiento de la pregunta

En la figura 5.6 pueden consultarse de forma simplificada la relación de recursos solicitados al servidor y el flujo de comunicación para satisfacer la carga del formulario y la recepción de los datos introducidos. Esencialmente se gestiona la carga de las categorías y del widget destinado a la recaptcha (ver sección 5.5.3), se valida la información que pretende enviar el jugador y si todo es correcto se invoca el recurso que recibe esta información (`/questions/user`). Las preguntas sugeridas se almacenan de forma aislada en el datastore empleando para ello entidades de tipo *UserQuestion* que registran, además de todos los datos recibidos, la fecha de recepción y el usuario asignado al jugador que la propone (ver la figura 5.24 para el detalle de su modelado).

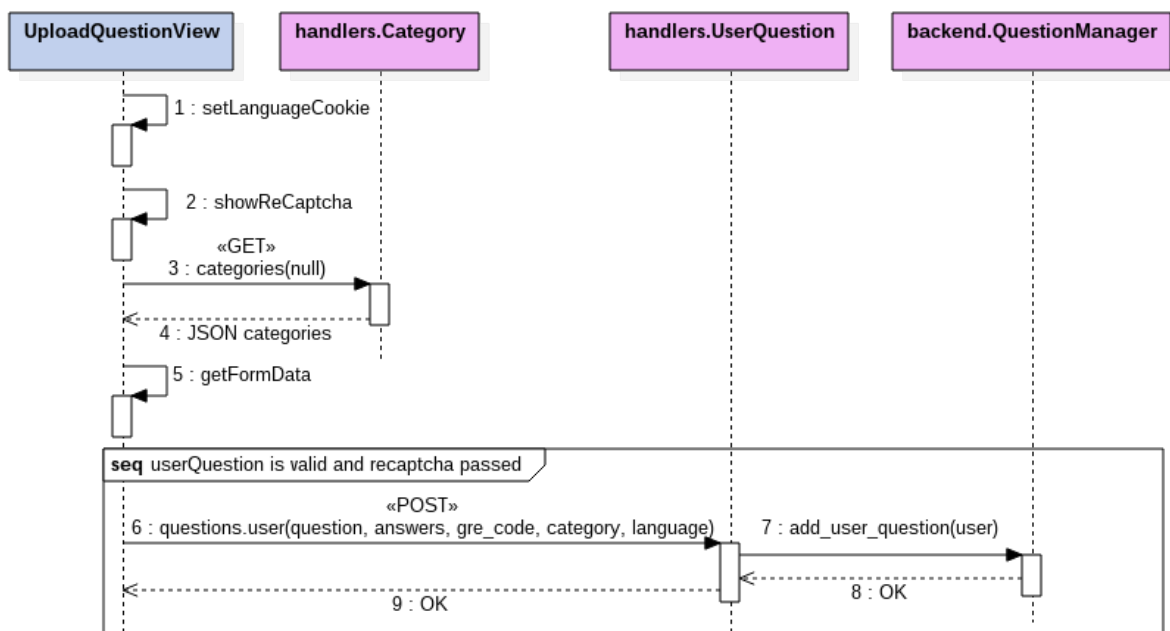


Figura 5.6: Diagrama de secuencia para la carga del formulario de aportación y recepción de los datos introducidos

5.5.3 Incorporación de reCaptcha

Dado que se permite que jugadores anónimos (no logueados) también puedan aportar preguntas al sistema, es conveniente agregar una barrera de seguridad que permita **protegerse frente ataques** lanzados de forma maliciosa a través de **bots y/o scripts**. A través de estos se podría saturar de peticiones al servidor o bien introducir en este de forma indiscriminada información que no sea de valor, malgastándose así recursos de memoria en su almacenamiento temporal. Con la finalidad de comprobar que la información que se pretende enviar al servidor ha sido generada por un jugador humano y no por medios automatizados es necesario incorporar un desafío que permita verificarlo previo envío de la información.

El primer mecanismo que se introdujo en los sitios web para combatir este tipo de vulnerabilidad era el de **incorporar una captcha**. Consiste en mostrar una imagen que contiene texto distorsionado y pedir al humano que introduzca los caracteres que se muestran, esperando que la distorsión sea una barrera fácilmente franqueable para este y que por contra dificultase la labor del robot. Sin embargo esta alternativa cada vez ha resultado ser más denostada, ya que el avance de la inteligencia artificial ha hecho que existan sistemas OCR que pueden resolver las captchas con un elevado porcentaje de exactitud. Como muestra se tiene el artículo presentado por Google donde se empleaba un sistema capaz de detectar los números de las casas en las imágenes capturadas a través de Street View con una exactitud del 99 %⁴. Lo que es más, para la mayoría de usuarios resultaban ser incómodas o incluso un factor limitante para personas con discapacidad.

A tal efecto se desarrolló una nueva versión del desafío más sofisticada: el empleo de **recaptcha**⁵. Esta solución únicamente solicita al usuario un simple click o toque en una pequeña área de la pantalla. Se analizan entonces una serie de parámetros de riesgo para evaluar si este ha sido o no ejecutado por un humano. En caso de sospecha, se propone un reto alternativo basado en asociaciones de carácter visual, poniéndose ahora en jaque las capacidades de visión por computador del robot. Este tipo de pruebas resultan ser más rápidas y cómodas para el usuario y más complejas de resolver para los robots. En la figura 5.7 se ve el aspecto de la prueba.

Así pues con el fin de incorporar esta solución ofrecida por Google, es **necesario registrar el dominio de QuPI-GAm** en este proveedor para dar de alta al sistema como un cliente válido, obteniéndose así un identificador y una clave secreta, que dan acceso al API. Una vez solventado este aspecto de configuración, se está en disposición de incorporar el procedimiento de carga y verificación del desafío. A continuación se expone de forma simplificada y se representa en la figura 5.8 el flujo de comunicación y los pasos que se producen entre los tres extremos implicados, siendo estos el proveedor del API y el cliente que solicita acceso (QuPI-GAm) diferenciándose a su vez en este las partes de frontend y backend:

⁴<https://security.googleblog.com/2014/04/street-view-and-recaptcha-technology.html>

⁵<https://www.google.com/recaptcha/intro/index.html>

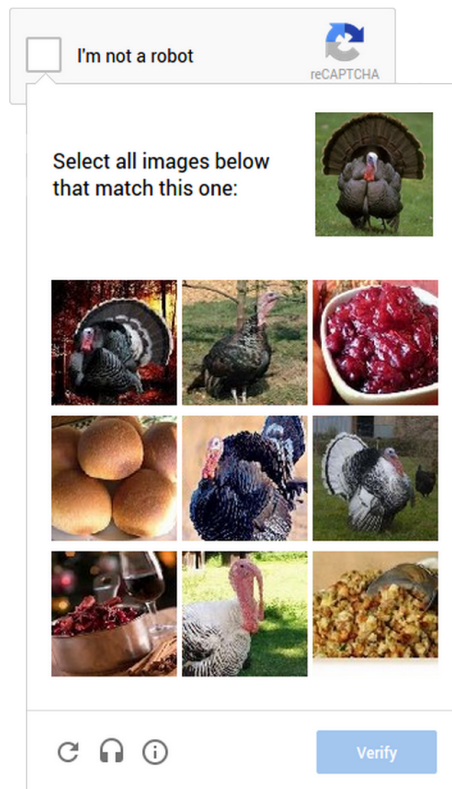


Figura 5.7: Ejemplo de recaptcha

- En la parte cliente, la vista responsable de mostrar el formulario para la introducción de la pregunta ha de emitir una solicitud para cargar el widget correspondiente a la recaptcha. Es necesario especificar en esa petición la clave de sitio web asignada y los parámetros de ajuste. El más importante es el relativo al idioma en que quiere cargarse dicho widget.
- Cuando el jugador completa la recaptcha se genera un código textual correspondiente al resultado de la evaluación. Al tratar de enviar la información introducida se recupera dicho código y se incorpora a la petición como parámetro.
- En el manejador del recurso encargado de recibir la información lo primero que se hace es verificar el código de recaptcha recibido. Para ello se emite una petición desde el servidor especificando la clave secreta asignada y dicho código hacia el servidor del proveedor del servicio. En caso de que resulte ser un código válido se devuelve la confirmación.
- Si la validación resultó exitosa se continua en el servidor procesando la información recibida. En caso contrario se corta el proceso y se devuelve un mensaje de error al lado cliente.

Por último hay que tener en cuenta que el comportamiento de la carga de la recaptcha varía en función de si el jugador es anónimo o ha iniciado sesión. Según el caso:

- En caso de ser **anónimo**, la recaptcha siempre será requerida. Es por ello que después de enviar una pregunta y ser esta confirmada, el widget ha de recargarse en el formulario, con el fin de que el jugador vuelva a pasar la verificación para cada pregunta que quiera introducir. Aun así, se espera que un jugador anónimo no tenga interés en aportar más de una pregunta de forma consecutiva.
- En caso de estar **identificado**, la recaptcha será requerida para la primera pregunta que introduzca tras el inicio de sesión. Así, cuando se le confirma que su primera pregunta ha sido aceptada, en el lado cliente ha de eliminarse el widget.

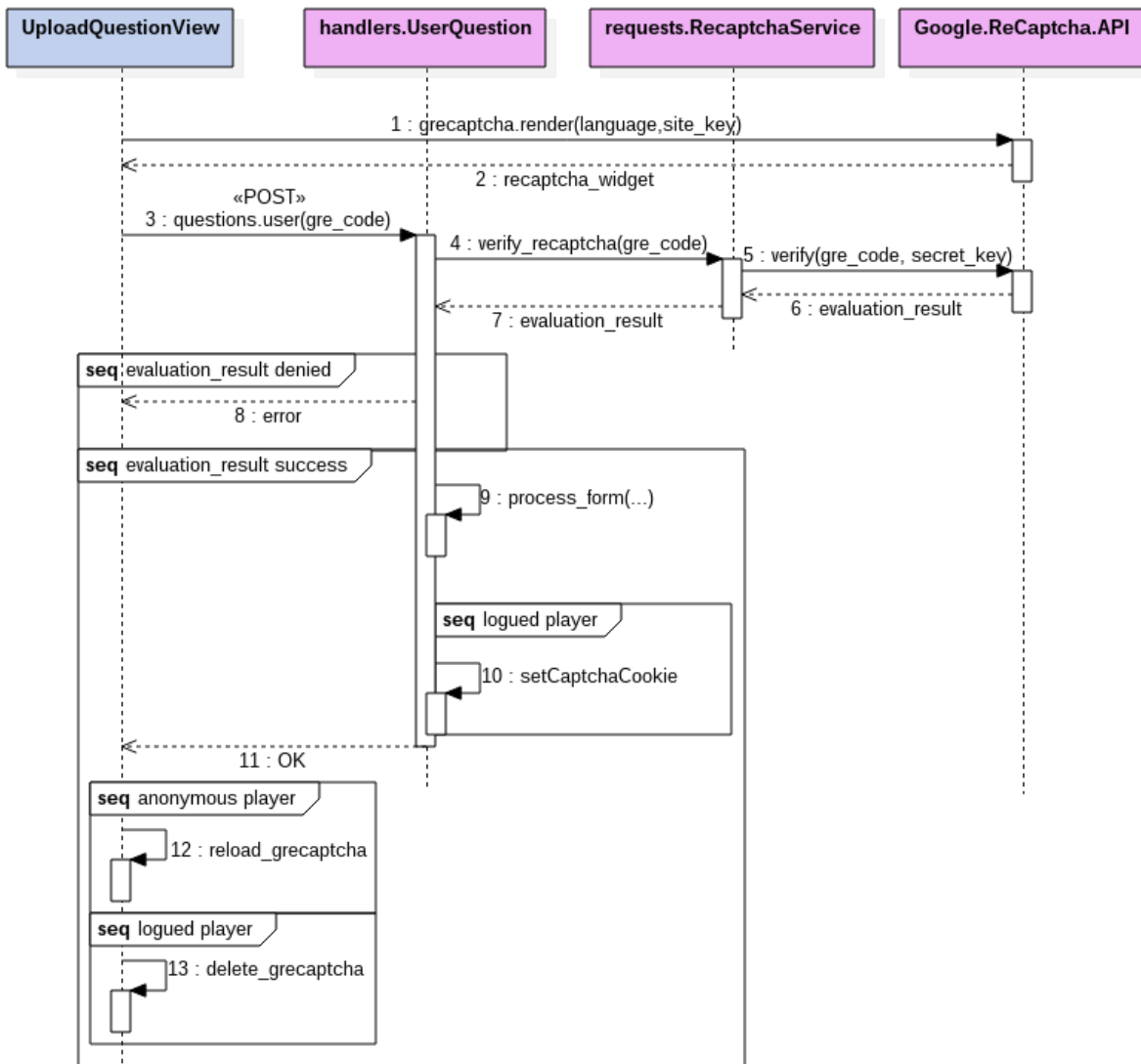


Figura 5.8: Comunicación establecida para la integración de la recaptcha

Cookie de recaptcha Este nuevo estado ha de ser conocido en el servidor para que en las sucesivas peticiones emitidas, el manejador no inicie el proceso de validación del código de recaptcha. A tal efecto, es necesario crear una cookie que sentencie si para ese jugador con sesión iniciada debe o no aplicársele ese procedimiento.

5.6 Validación de preguntas

Módulo	Validación
Identificador	US_VALID_01 (cuadro 5.10)
Problemática	Personal seleccionado debe poder validar las preguntas sugeridas por los jugadores. A tal efecto se necesita incorporar un mecanismo a través del cual se puedan recuperar, revisar, aprobar y/o descartar las preguntas.
Enfoque	Se creará un formulario accesible solo por usuarios registrados como validadores que cumpla todas las condiciones definidas en la historia de usuario correspondiente. Hay que poner especial atención en el diseño a las implicaciones del posible acceso concurrente a este formulario por parte de varios validadores.

5.6.1 Constitución del formulario

Siguiendo el mismo orden que para la explicación del formulario de aportación, lo primero a considerar es la información recuperada del datastore con que este será poblado. Esencialmente la información a manejar es la misma que para el formulario de aportación, con matices:

- **Idioma.** El validador puede escoger de qué grupo de preguntas quiere recuperar una en función del idioma. Si por alguna razón el jugador especificó mal el idioma, una vez recuperada el validador puede definir el idioma correcto en el mismo selector para que este error se corrija al guardarse como definitiva.
- **Categorías.** El validador tiene la posibilidad de definir también el grupo de preguntas en que tiene interés especificando a qué categoría general deben pertenecer. A parte de este primer filtro, debe seleccionar su árbol de subcategorías para clasificarla correctamente. Al igual que con el idioma, puede corregir la categoría general en caso de que el jugador se equivocase, ya que lo que finalmente se almacenará será la subcategoría específica que él defina.
- **Pregunta.** El validador podrá visualizar tanto el enunciado de la pregunta como las opciones (correcta e incorrectas) sugeridas. En caso de ser una pregunta válida pero esta contenga errores sintácticos o gramaticales, o bien la respuesta correcta no se ajuste a la realidad, el validador tiene libertad para editar estos campos.

Revisión mínima del contenido aportado

Al igual que con el formulario de aportación, cuando el validador manifiesta la intención de almacenar definitivamente una pregunta, se aplica un **procedimiento de validación mínima** con el fin de evitar errores humanos. En caso de no superarse alguno de los filtros, se informa convenientemente al validador del fallo mediante un mensaje textual y destacando el área de conflicto. En el lado cliente se trata de construir un objeto de tipo *ValidatorQuestion* a raíz de los campos introducidos en el formulario. Este modelo incorpora una función de validación que se encarga de aprobar o denegar su construcción en base a:

- Debe haberse recuperado previamente una pregunta pendiente de validación. Esto significa que el validador no puede introducir información en el formulario vacío y tratar de validarla. **Cada validación debe estar vinculada a una pregunta** ya almacenada.
- No puede quedar **ningún campo vacío**.
- No se permiten **campos repetidos**.
- Se impone una **limitación en el número de caracteres** máximos por campo. Esto garantiza una correcta visualización en el espacio asignado en dispositivos móviles.
- El texto del enunciado de la pregunta debe incluir **al menos 3 palabras y de una longitud mínima** en términos de caracteres, ya que con menos no se puede contruir una pregunta con sentido.
- Debe **especificarse la subcategoría** a la que pertenece la pregunta.

Además de las comprobaciones anteriores, de forma automatizada se revisan todas las cadenas de texto que se van a enviar como definitivas, **eliminando caracteres de espaciado** innecesarios que hayan podido pasar inadvertidos al validador (dos espacios en blanco juntos, tabulación, salto de línea, etc).

5.6.2 Almacenamiento de la pregunta

Se proyecta un diseño con **dos bancos de preguntas** diferenciados: uno temporal, donde se almacenan las preguntas sugeridas por los jugadores y otro definitivo, donde se almacenan todas las preguntas provenientes del primer grupo que ya han sido revisadas. De este segundo grupo ya se podrían tomar las preguntas que se muestran durante una sesión de juego. En la figura 5.9 se ilustra este punto, recuperándose además el modelo asignado a cada tipo de pregunta. En el caso del **modelo Question** se almacena la misma información contenida inicialmente por el modelo *UserQuestion* pero agregando las diferentes traducciones del texto y asignándose la fecha de validación y quién fue el validador responsable.

Descripción del flujo recuperación/validación

En la figura 5.10 se aprecia el flujo de llamadas desencadenado para el proceso completo de validación en sus distintas variantes. El proceso de validación se constituye de:

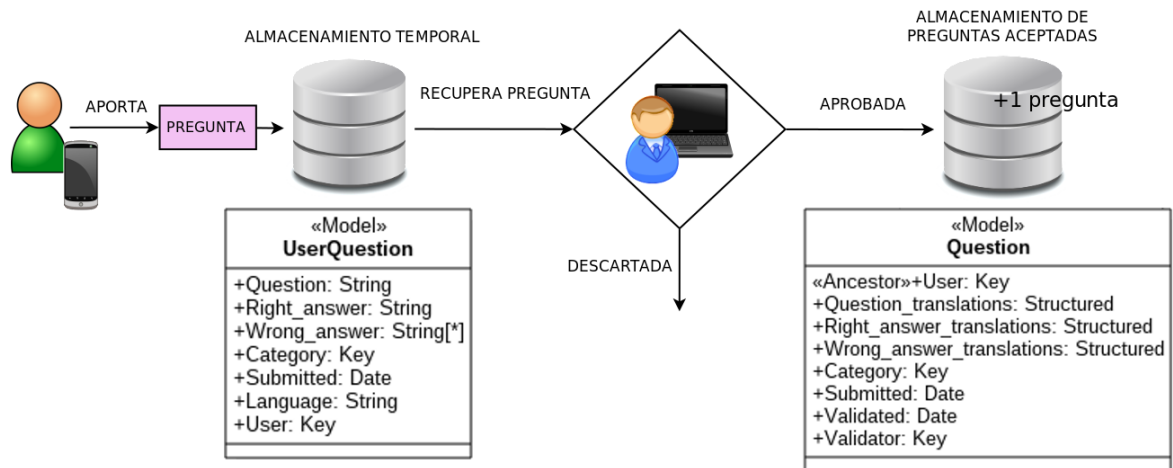


Figura 5.9: Flujo que sigue la pregunta hasta que es almacenada definitivamente

1. **Recuperación.** El primer paso consiste en recuperar una pregunta del conjunto de las pendientes de revisión, pudiendo especificar aquellas pertenecientes a una categoría general y propuestas en un idioma específico. Cuando se solicita el recurso al servidor que recupera una de estas preguntas:
 - Además de devolver la información asociada a la pregunta, **se devuelve también su identificador en el datastore**. De esta forma no se pierde la conexión entre la información que está visualizando el validador y la pregunta almacenada.
 - Cuando se devuelve la información al validador aun **no se borra del datastore** la instancia correspondiente a esa pregunta (del modelo *UserQuestion*).
 - En este punto se lanza la comprobación automatizada que comprueba si existe ya una pregunta validada similar a la que ha recuperado, con el fin de notificárselo al validador. Este aspecto se detalla en la sección 5.7.

Política de recuperación. Del conjunto de preguntas se escogerá **aquella que más tiempo lleva almacenada**. Aunque solo se requiere una pregunta para satisfacer la petición, **se recuperan diez preguntas** del datastore y se devuelve una de esas diez de forma aleatoria. ¿Por qué? Esto permite que si un validador tiene problemas con una pregunta pueda solicitar otra sin preocuparse por esa, **evitando un bloqueo**. Si siempre se devolviese la más antigua, al volver a solicitar una pregunta recibiría la misma hasta que otro validador se encargase de ella.

2. **Operación.** Una vez recibida la pregunta, el validador debe decidir si la pregunta es apta o no, categorizarla y corregirla en caso de ser necesario.

3. **Decisión.** Una vez revisada la pregunta existen dos opciones: optar por descartarla o bien dar la aprobación. Según el caso:

- Si se decide por la **aprobación definitiva**, la información vuelve a enviarse al servidor. Esta se almacena a la luz del modelo *Question*. Es en este punto cuando se elimina definitivamente la información almacenada en el datastore para esta pregunta bajo el modelo *UserQuestion*. No conviene eliminarla antes porque si no se forzaría al validador a decidir sobre la pregunta recuperada sin posibilidad de poder posponerlo. En este punto también se llevan a cabo acciones de soporte a la detección automática de pregunta repetida.

Aprobación como transacción. Es importante poner de relieve que las tres acciones conforman un todo y que si no se ejecutan todas en bloque el sistema quedaría en un estado inconsistente. Es por ello que hay que garantizar que todas las acciones derivadas de la validación han de ejecutarse como una transacción.

- Si se opta por el **descarte**, este hecho es notificado al servidor, que se dispone a eliminar la pregunta descartada.

Validadores concurrentes

Dado el enfoque que se acaba de presentar, es necesario analizar cómo impacta sobre este el hecho de que haya varias personas realizando labores de validación para una misma categoría y un mismo idioma. Esto implicaría que varias personas se están nutriendo del mismo grupo de preguntas. Supóngase que V1 recupera una pregunta. Como al recuperar una pregunta pendiente de validación esta no se elimina del datastore, un validador V2 que recupere una pregunta instantes después puede **recibir la misma pregunta**.

Sabiendo que para cada petición se recuperan las 10 preguntas más antiguas y se devuelve una de ellas aleatoriamente, el hecho de que V2 recupere una pregunta justo después no implica que vaya a recuperar la misma que V1. Si solo se tienen dos validadores concurrentes las probabilidades son bajas. En cambio si se tienen 10 validadores concurrentes la probabilidad de coincidencia se dispara. Esto sugiere que ese parámetro (10 preguntas entre las que elegir) debería ajustarse en el caso de conocer cuantos validadores se tienen realmente por categoría, incrementándolo si es necesario.

Asumiendo que V1 y V2 recuperan una pregunta con un pequeño lapso de tiempo y resulta ser la misma, ambos estarían **trabajando sobre la misma pregunta**. Entonces si V1 valida/descarta antes que V2, cuando V2 intente validar/descartar se intentará recuperar la pregunta originalmente propuesta por el jugador para eliminarla, pero esta no existirá. En caso ha sido contemplado y **manejado a través de una excepción** específica, que cuando salta desencadena que se informe de que otro validador ya se ha encargado de esa pregunta.

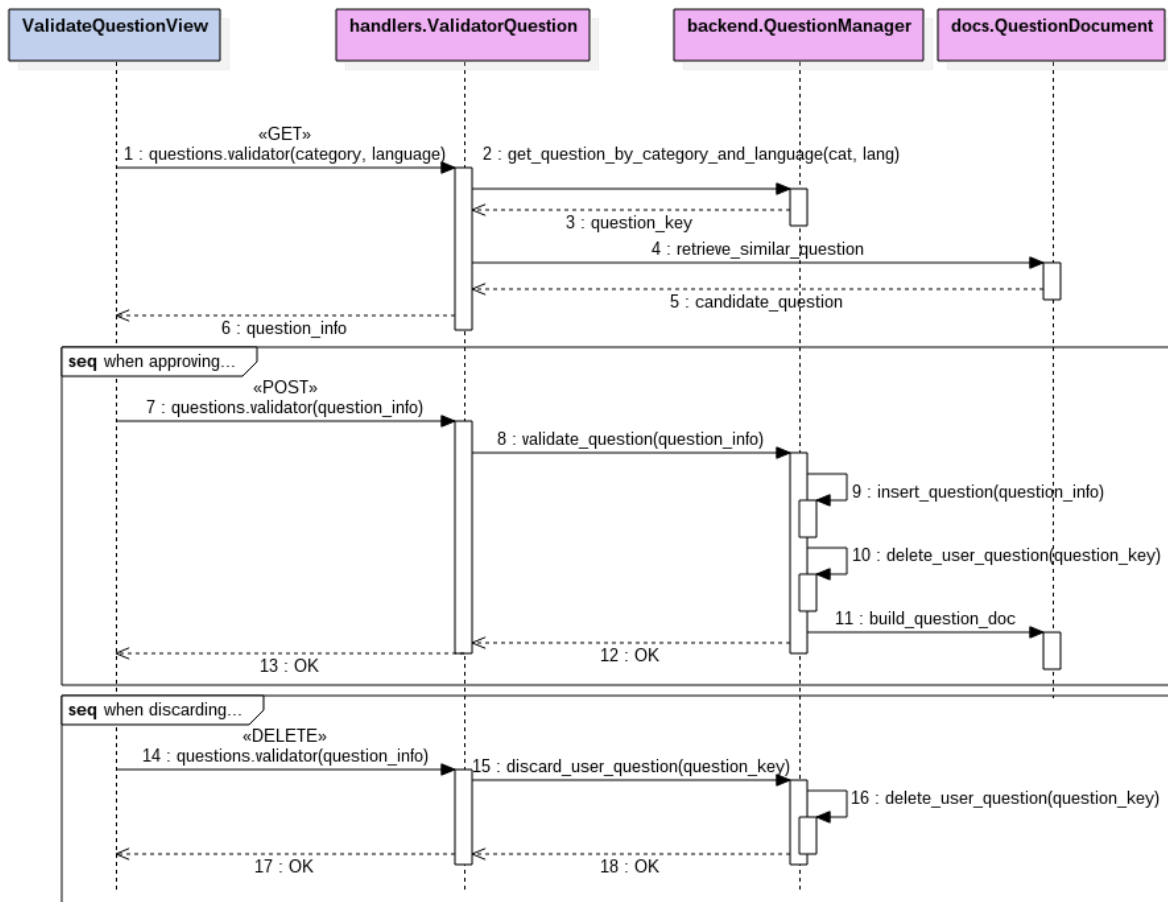


Figura 5.10: Diagrama de secuencia para operación mediante el formulario de validación

5.7 Detección de pregunta repetida

Módulo	Validación
Identificador	US_VALID_02 (cuadro 5.11)
Problemática	Uno de los problemas que hay que enfrentar al recopilar información de diversas fuentes no coordinadas es que ésta pueda ser coincidente. En este caso no es deseable contar con esta redundancia. Por ello resulta conveniente estudiar la incorporación de algún mecanismo que trate de detectar cuándo una pregunta propuesta puede haber sido incorporada previamente al sistema.
Enfoque	Se estudiaron dos aproximaciones para dar salida a este objetivo, una solución más manual que requería la elaboración de un grafo de palabras donde quedase patente la temática de las preguntas aprobadas y otra centrada en aprovechar el API de búsqueda incorporada en App Engine. A continuación se detalla el planteamiento de ambas opciones y cual fue escogida finalmente.

5.7.1 Aproximación 1: Elaboración de un grafo de palabras

La primera aproximación se basaba en la creación de un grafo que permitiese registrar la temática de las preguntas validadas a nivel de las palabras clave que la componen. De esta forma se dispondría de un grafo almacenado en el datastore cuyos nodos serían las palabras

relevantes extraídas de cada pregunta y donde las relaciones representarían la vinculación entre palabras. Puede verse en la Figura 5.11 un ejemplo del aspecto que tendría el grafo, donde A_x representa una asociación y P_x representa una palabra. La asociación A_2 representa que las palabras P_3 , P_4 y P_5 aparecen juntas en una pregunta que ya ha sido validada.

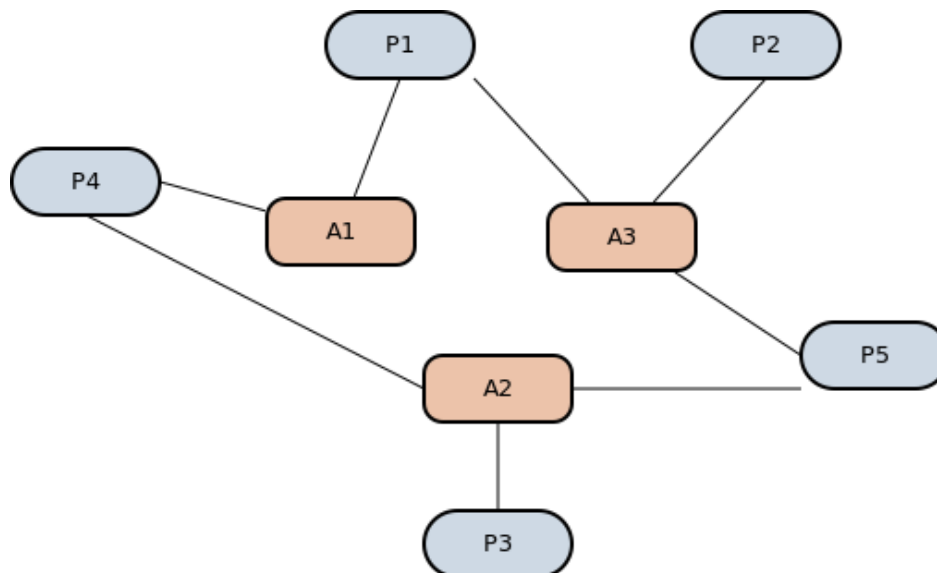


Figura 5.11: Ejemplo para visualizar los componentes del grafo de palabras

Grafo de relaciones frente a contador de ocurrencias

En un primer momento se pensó en un enfoque simplificado, según el cual se podría almacenar en una tabla tipo hash cada palabra relevante detectada y el número de ocurrencias de la misma para el total de preguntas registradas. De esta forma si la media del número de ocurrencias del conjunto de palabras que componen la pregunta superase un umbral definido, se consideraría que la pregunta está repetida. El problema de este enfoque es que lo que realmente se sabe es que todas las palabras que componen la pregunta han sido referenciadas muchas veces en otras preguntas, pero no hay forma de saber si han sido referenciadas en conjunto. Y en conjunto podían dar lugar a una nueva pregunta útil. Para el anterior ejemplo, la palabra P_5 que aparecía en la asociación A_2 también aparece en otra pregunta junto con palabras diferentes, las palabras P_1 y P_2 , según recoge la asociación A_3 . Esto denota el hecho de que la palabra P_5 estuviese altamente referenciada no asegura que pueda dar pie a una nueva pregunta válida en compañía de otras palabras que aporten un contexto diferente. Como ejemplo clarificador, véase el siguiente ejemplo:

Dada la tabla siguiente:

- París (500)
- Cantante (400)

Si el usuario introduce: «¿Cuál de los siguientes cantantes nació en París?». Las palabras “París” y “cantante” que están presentes en muchas preguntas, pueden estarlo en todos los casos por separado, por lo que no se tendría información suficiente para descartar esta pregunta.

Así pues se opta por diseñar una opción que permita registrar y consultar el conjunto de palabras relevantes asociadas a una pregunta en el grafo. A tal efecto se necesita considerar las siguientes cuestiones:

- ¿Cuál es el modelo de datos que da soporte al grafo?
- ¿De qué forma se pueden extraer las palabras relevantes de la pregunta?
- ¿De qué forma se registra en el grafo las palabras extraídas de una pregunta validada?
- ¿De qué forma se consulta en el grafo si existe una relación entre un nuevo conjunto de palabras de entrada?
- ¿En qué momento del proceso debe aplicarse este procedimiento?
- ¿Conviene que el descarte sea automatizado o siga existiendo apoyo manual?

Modelado del grafo

Respecto al modelo de datos, se propone un modelado basado en dos tipos de entidades, una que represente la **palabra** y otra que represente **asociaciones entre palabras**. Como puede comprobarse en la figura 5.12 con el tipo *QuestionLabel* se registraría cada nueva palabra, y esta contiene una lista de asociaciones en las que está presente. Cada entidad de tipo *LabelAssociation* contiene una lista de referencias a las entidades que representan las palabras que estaban presentes en una misma pregunta aprobada. Ésta supone un arco bidireccional entre cada par de palabras.

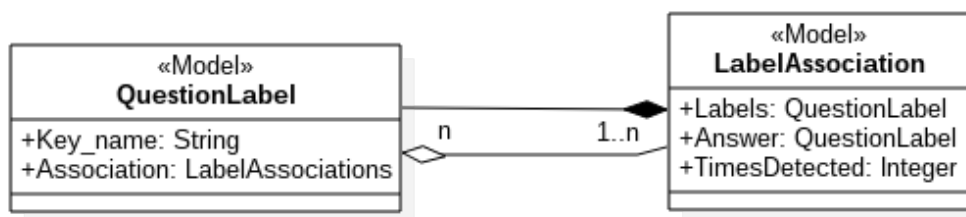


Figura 5.12: Modelos necesarios para la constitución del grafo de palabras

Integración del proceso en el flujo

Una vez establecido esto, es necesario determinar de qué forma se integraría en el flujo de recepción/aprobación este proceso, reflejado en la figura 5.13. Dado que lo ideal es que al validador solo le lleguen preguntas que no puedan ser repetidas (ya que de lo contrario estaría haciendo un esfuerzo innecesario al ser una pregunta no útil) para esta aproximación se determina que la mejor opción es **colocar la revisión y construcción del grafo en la recepción de la pregunta propuesta**, de tal forma que no mediaría en ningún momento el apoyo del validador. El proceso sería el siguiente:

1. El colaborador sube una pregunta.
2. Al recibir una pregunta se extraen las palabras clave.
3. Las palabras clave se contrastan en el grafo para determinar si existe repetición:
 - a) Si no se detecta coincidencia la pregunta es almacenada y se registra su temática en el grafo, incorporando las palabras en el grafo si no existían ya e incorporando la nueva asociación.
 - b) Si se detecta coincidencia se descartaría automáticamente.

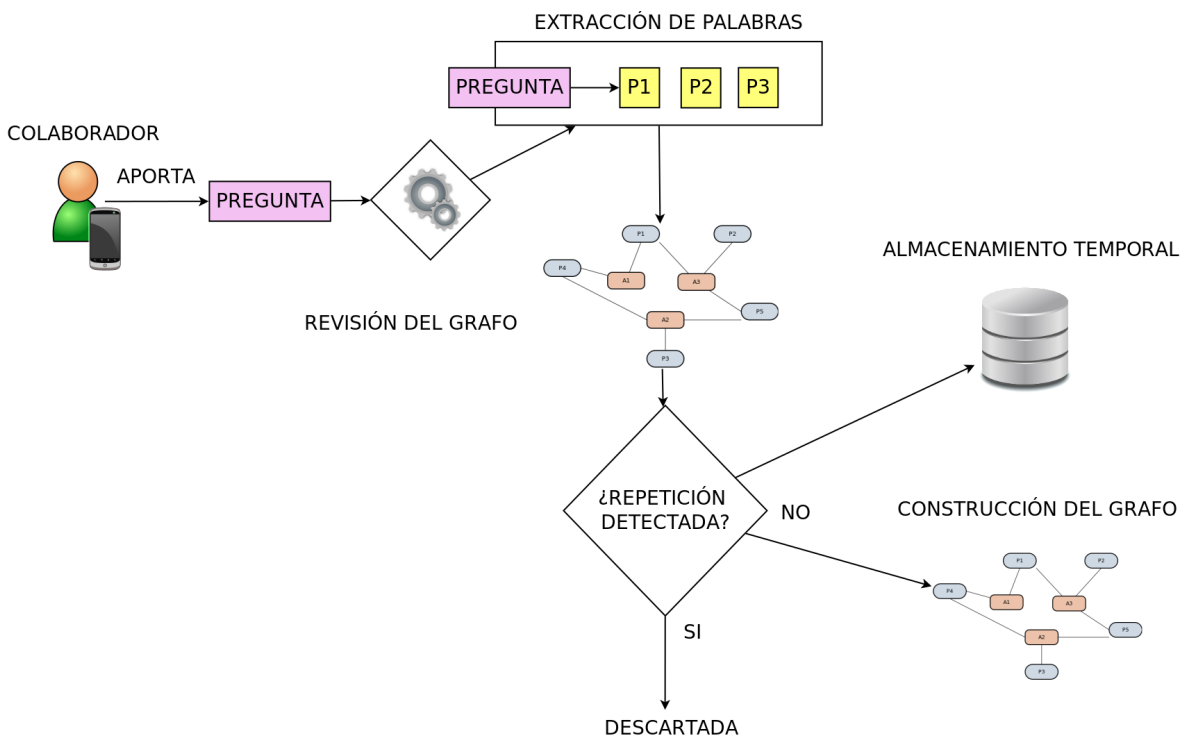


Figura 5.13: Integración del proceso de detección de pregunta repetida para la aproximación basada en grafo

Extracción de palabras

El siguiente paso consistía en determinar qué alternativas existen para realizar la extracción de palabras. Dada la naturaleza del grafo no conviene extraer todas las palabras de cada pregunta, ya que muchas de ellas no aportan significado. La opción idónea es la de **recurrir al procesamiento del lenguaje natural** ya que de esta forma, si se consigue identificar de qué tipo es cada palabra, es posible **determinar cuales resultan ser más relevantes**. Se requería por tanto una librería capaz de ofrecer funcionalidades propias del procesamiento del lenguaje natural, que pudiese ser incluida en el servidor y de libre uso.

NLTK⁶ es una herramienta que engloba una serie de recursos útiles relacionados con el lenguaje natural, entre ellos, un conjunto de librerías para el procesamiento de texto. Resultaba una opción interesante en tanto que el código fuente está escrito en python. Esta disponible bajo licencia Apache 2.0 y además incluye la posibilidad de asignar etiquetas POS (*Part of Speech*) a cada palabra de un texto (*tagging*), que es la funcionalidad requerida.

Etiquetar las palabras por su categoría gramatical es útil en dos aspectos:

- **Permite realizar una validación mínima de la pregunta**, comprobando si tiene un mínimo de palabras tolerable y si contiene alguna forma verbal (al menos 3-4 palabras son requeridas para poder formar una pregunta y siempre debe contener un verbo).
- **Permite elaborar una lista de categorías de interés**, aquellas que aporten el sentido a la pregunta, pudiéndose descartar así las palabras que no pertenezcan a estas. Por lo general las preguntas aportadas estarán formadas por diferentes formas verbales, determinantes, sustantivos, nombres propios, pronombres, conjunciones, adverbios, adjetivos y números. El aplicar el proceso de etiquetado se mantendrían únicamente verbos, sustantivos, nombres propios, adjetivos, números y las denominadas partículas -qu, que caracterizan en muchos casos el tipo de respuesta que se busca.

Revisión / construcción del grafo

Habiendo ubicado la manera en que pueden extraerse las palabras clave de una pregunta, el siguiente paso es determinar una primera aproximación al procedimiento por el cual se comprueba si un conjunto de palabras está presente en el grafo. En primer lugar es necesario identificar qué palabras están presentes en el grafo y cuales no. En segundo lugar para las palabras que ya hayan sido incluidas en el grafo previamente, **se busca en este si existe una relación entre todas ellas o algunas de ellas**, tomando como relevante aquella asociación más larga (que involucre mayor número de palabras de entre las consideradas).

Se muestra en el Listado 1 en pseudocódigo este procedimiento, donde para cada palabra extraída presente en el grafo, **para cada asociación en la que esté presente, se comprueba si las demás palabras se encuentran entre esas asociaciones**. Como se ve, el tiempo de

⁶<http://www.nltk.org/book.1ed/>

ejecución depende del número de palabras y el número de asociaciones donde esté presente cada palabra. El número de palabras no se espera que sea mayor de 5 a 10 palabras extraídas de la pregunta. El factor limitante es que para conceptos populares éstos pueden estar presentes en muchas asociaciones, lo que ralentizaría el proceso.

Algorithm 1 Pseudocódigo para detección de asociación entre palabras en el grafo

Input: question_labels

Output: Check_Repeated_Question

```
1: existing_labels = label in question_labels and label in graph
2: new_labels = label in question_labels and label not in graph
3: foreach label in existing_labels do
4:   foreach association in label.associations do
5:     chain = 1
6:     foreach remaining_label in (existing_labels - label) do
7:       if remaining_label in association then
8:         chain = chain + 1
9:       end if
10:    end for
11:    max_chain = MAX(max_chain, chain)
12:    if len(longest_association) < max_chain then
13:      longest_association = association
14:    end if
15:  end for
16: end for
```

Una vez extraída la asociación más larga se dispone de una serie de datos que dan lugar a múltiples combinaciones que determinan si la pregunta resulta o no de interés:

- **N_LABELS**, como número de palabras extraídas de la pregunta
- **N_NEW**, como número de palabras nuevas
- **N_EXISTING**, como número de palabras ya presentes en el grafo
- **N_ASSOCIATION**, como número de palabras de la asociación más larga detectada
- **MAX_CHAIN**, como número máximo de palabras bajo una misma asociación

En función de su valor se tienen distintas situaciones, dependiendo de si las palabras son **nuevas o existen** en el grafo y de si se **detecta o no una asociación** en el grafo y si es total o parcial. Todas ellas se reflejan en el Listado 2, algunas de ellas dando pie a la introducción de información nueva en el grafo, al descarte de la pregunta o a casos intermedios donde se debe decidir.

Algorithm 2 Pseudocódigo para veredicto de repetición en el grafo según casuística

Input: question_labels**Output:** Check_Repeated_Question

```
1: // Disponiendo ya de las siguientes variables
2: existing = label in question_labels and label in graph
3: new = label in question_labels and label not in graph
4: longest_association
5: max_chain
6: n_existing = len(existing)
7: n_new = len(new)
8: n_labels = len(question_labels)
9: n_association = len(longest_association)
10: if n_association == 0 then
11:   // Caso: TODAS las palabras son NUEVAS
12:   // – Existing[]
13:   // Caso: NINGUNA ASOCIACION detectada entre las palabras ya existentes en el
   grafo
14:   // – Existing[A-B-C] and NOT Association[A-B] or [B-C] or [A-C] or [A-B-C]
15:   new_association = new LabelAssociation
16:   new_association.labels = new_labels + existing_labels
17:   foreach label in new_association.labels do
18:     label.associations = label.associations + new association
19:   end for
20:   REPEATED = false
21: else
22:   if max_chain == n_existing then
23:     // Caso: TODAS las palabras EXISTENTES LOCALIZADAS en una asociacion
24:     // Subcaso: La ASOCIACION puede incluir MAS PALABRAS y puede haber
   NUEVAS o no
25:     // – Existing[A-B-C-D] (and New[F-G]) and Association[A-B-C-D-E]
26:     REPEATED = true
27:     if n_new != 0 then
28:       longest_association.labels = longest_association.labels + new_labels
29:       foreach label in new_labels do
30:         label.associations = longest_association
31:       end for
32:       REPEATED = false
33:     end if
34:   else
35:     // Caso: ALGUNAS de las palabras EXISTENTES LOCALIZADAS en una aso-
   ciacion
```

```

36:     if max_chain == n_association then
37:         // Subcaso: La ASOCIACION no aporta MAS PALABRAS
38:         // Existing[A-B-C] (and New[F-G]) and Association[A-B] or [B-C] or [C-A]
39:         longest_association.labels = new_labels + existing_labels
40:         foreach label in (new_labels + existing_labels) do
41:             label.associations = longest_association
42:         end for
43:         REPEATED = false
44:     else
45:         // Subcaso: La ASOCIACION aporta MAS PALABRAS
46:         // Existing[A-B-C] (and New[F-G]) and Association[A-B-D] or [B-C-D] or [C-
A-D]
47:         new_association = new LabelAssociation
48:         new_association.labels = new_labels + existing_labels
49:         foreach label in new_association.labels do
50:             label.associations = label.associations + new_association
51:         end for
52:         REPEATED = false
53:     end if
54: end if
55: end if

```

Dificultades que justifican el descarte

Esta aproximación resulta ser un tanto rígida ya que aunque haya coincidencia, en el momento en que la asociación localizada en el grafo aporte más palabras o queden al margen algunas de las palabras de entrada, es difícil sentenciar un descarte. A esto se suman otra serie de dificultadores que llevaron a estudiar otras opciones:

- Al tratar de integrar NLKT en el proyecto, agregándolo como una librería externa, se encontraron dificultades. App Engine solo permite agregar librerías externas escritas puramente en python. Algunas de las funciones de NLKT están desarrolladas en C para un soporte a más bajo nivel. App Engine no tolera esto para garantizar que ninguna de las aplicaciones desplegadas toma más tiempo de cpu del asignado, para evitar interferencias en el rendimiento de otras aplicaciones con las que esté compartiendo recursos. La solución pasaba por aislar el módulo de interés con aquellas funciones requeridas que no inclumplieran la restricción, subiendo esto separadamente. Esto requería estudiar algún proyecto que hubiese integrado satisfactoriamente NLTK en App Engine, no resultando ser una tarea asequible.
- Dado que se dispone de preguntas en dos idiomas, habría que haber dispuesto dos grafos separados, lo que abriría la posibilidad de tener una misma pregunta repetida dos veces, al hacerse el análisis de forma separada para cada idioma.

- Registrar las distintas formas verbales de un mismo verbo resultaba ser inconveniente pues se aumentaba el tamaño del grafo con un conjunto de palabras que representan lo mismo. A tal efecto existe en NLKT una utilidad que permite convertir una forma verbal a su correspondiente infinitivo, lo que podría haber resultado el problema.
- Por la riqueza del lenguaje, esta aproximación también es rígida en tanto que no tiene ninguna tolerancia a la comprensión de sinónimos. NLTK dispone de un módulo que sugiere sinónimos, pero considerando éstos a parte de las palabras de entrada hubiese implicado un aumento en la entrada de información al algoritmo de detección, lo que perjudicaría el rendimiento.
- Al colocar la revisión tras recibir la pregunta, puede que las palabras se reciban con fallos de ortografía. Esto implica que se metería en el grafo una palabra residual, que no va a poder ser identificada con su homóloga correcta.
- Es necesario decidir si se analiza únicamente el enunciado de la pregunta o también la respuesta correcta, dado que al ser dos unidades de información separadas, aun aportándose información más precisa, se incorpora un factor de incertidumbre mayor a la hora de decidir en qué casos se descarta la pregunta en el algoritmo.

5.7.2 Aproximación 2: Empleo del API de búsqueda de Google

App Engine ofrece la posibilidad de emplear un API que permite incorporar un mecanismo de búsqueda de texto sobre información estructurada, a través del empleo de índices temáticos y documentos. Cada pieza de información se almacena en un documento, conteniendo este campos de tipo texto, fecha, numérico y geolocalización. Incluye un lenguaje para realizar consultas en base a las cuales se localizan documentos concretos que cuadran con la búsqueda especificada. Para este enfoque se vuelven a discutir cuestiones de diseño mencionadas en la primera aproximación, tratando de corregir problemas detectados:

- ¿Qué información extra requiere ser almacenada para poder realizar la búsqueda?
- ¿Es necesario en esta aproximación llevar a cabo un proceso de extracción de palabras?
- ¿De qué forma se determina que el proceso de búsqueda ha devuelto un resultado suficientemente aproximado?
- ¿En qué momento del proceso debe aplicarse este procedimiento?
- ¿Conviene que el descarte sea automatizado o siga existiendo apoyo manual?

Información requerida

En primer lugar se discute qué información extra es requerida para dar soporte al proceso de búsqueda. A diferencia de la aproximación basada en grafo, esta resulta ser más sencilla, en tanto que basta con llevar un registro paralelo de las preguntas validadas, almacenando estas en documentos además de su correspondiente entrada en el datastore. Se trata de crear un índice de preguntas en el cual se va incorporando un documento por pregunta. El documento únicamente registra el idioma de la pregunta, un campo de texto para el enunciado y otro campo de texto para la respuesta correcta.

Integración del proceso en el flujo

El siguiente paso consiste en volver a discutir en qué parte del flujo seguido por las preguntas desde que son aportadas hasta que son validadas debe incorporarse la comprobación. Teniendo en cuenta que se tienen dos conjuntos de preguntas (uno de recibidas y otro de aprobadas) hay que definir con qué conjunto de preguntas se desea realizar la comprobación y en qué momento, para garantizar que se realiza antes de que le llegue al validador y que la comprobación sea consistente:

- **Todas las preguntas VS preguntas validadas.** Si para realizar la comprobación se van almacenando en documentos todas las preguntas recibidas, esto llevaría a realizar el contraste con preguntas que no han sido aprobadas. Esto significa que se podría descartar una pregunta a causa de otra que al llegar al procedimiento de validación podría ser descartada, y por lo tanto al no integrarse ninguna de las dos preguntas al sistema se ha perdido información. Es por ello que parece más apropiado ir registrando únicamente las preguntas definitivas validadas.
- **Comprobación tras la recepción VS antes de la validación.** En el enfoque basado en grafo se proponía inicialmente la posibilidad de aplicar la revisión justo en el momento de recibir una nueva pregunta subida por un colaborador, para así poder descartar automáticamente la pregunta antes de almacenar nada en el datastore. Sin embargo si ahora se hace eso, queda un hueco sin cubrir, en la medida que si varias preguntas similares son aportadas al sistema antes de que alguna de ellas sea validada no se detectaría la repetición para ninguna de ellas, llegando todas a la fase de validación y pasando todas ellas al conjunto de preguntas aportadas. Es por ello que resulta más adecuado realizar la comprobación justo antes de que un validador solicite la revisión de una nueva pregunta.

Necesidad de soporte manual

Según se acaba de exponer, resulta más adecuado contrastar solo con las preguntas validadas. Cuando un validador solicita la revisión de una pregunta, para la pregunta devuelta se realiza la consulta previamente y se determina si hay una pregunta candidata que se parezca. El hecho de colocar el proceso de comprobación en esta parte del flujo habilita la posibilidad de constituir el proceso de forma totalmente automatizada o como un proceso de soporte que cuente también con revisión manual.

Cuando se termina si existe una pregunta que puede ser parecida, esta podría eliminarse directamente y escoger una nueva. Sin embargo, a causa de la **dificultad de dictaminar un veredicto preciso** (como se justificará en posteriores apartados) llevaron a la conclusión de que es más razonable y seguro establecer este mecanismo como un **mecanismo de soporte**, de tal forma que en lugar de efectuar un descarte, ante la existencia de una pregunta candidata la respuesta del sistema es dar un aviso al validador, mostrándole la pregunta y delegando en él la decisión final. En la figura 5.14 se muestra cómo quedaría el flujo completo.

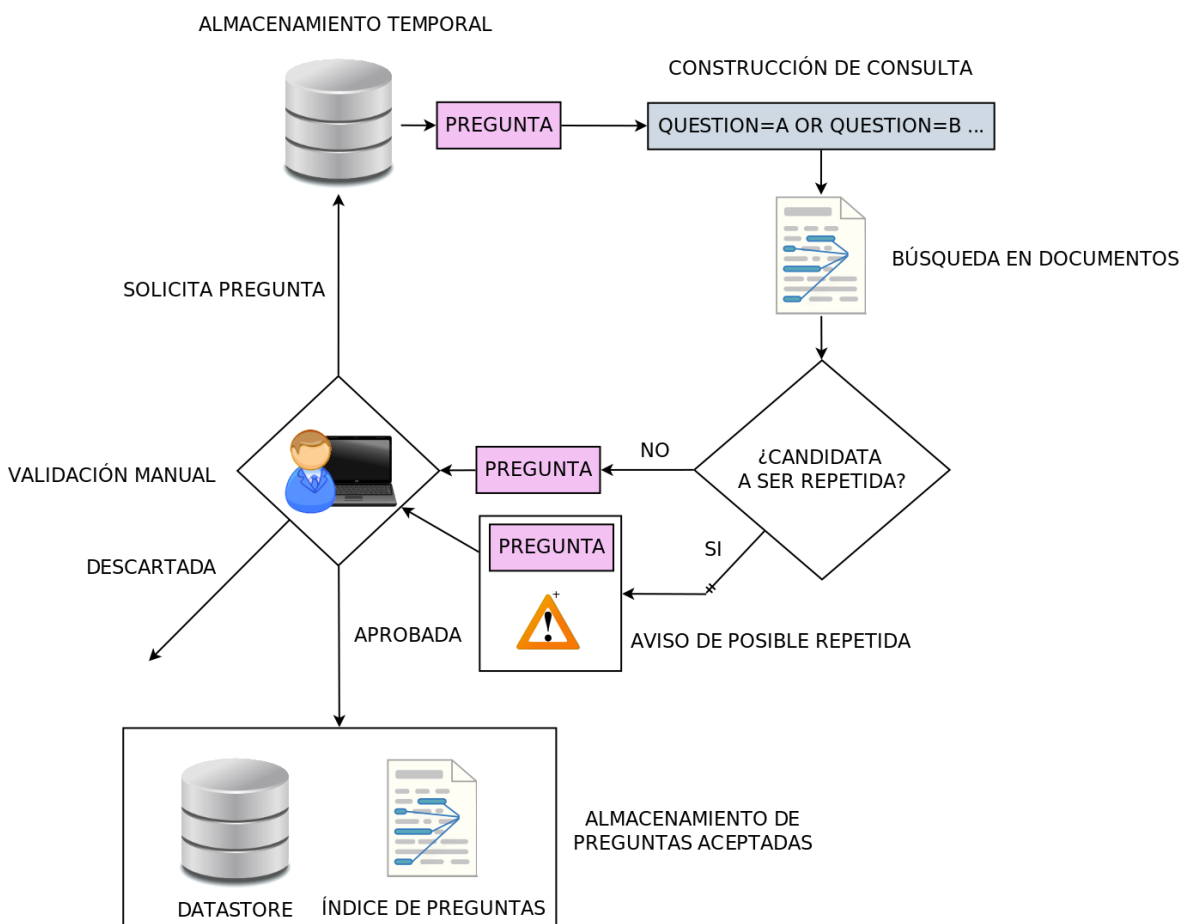


Figura 5.14: Integración del proceso de detección de pregunta repetida para la aproximación basada en API de búsqueda

Construcción de la consulta

En el caso de la aproximación basada en grafo tanto el procedimiento de construcción del grafo como el de consulta eran complejos. En este caso la constitución del índice de preguntas es sencillo. Basta con **crear un nuevo documento con la pregunta completa y la respuesta correcta** y agregarlo en el momento en que la validación se lleva a cabo. Sin embargo la construcción de la consulta y el análisis del resultado requieren más dedicación.

A la hora de constituir la consulta es necesario establecer los campos del documento a los cuales se aplicará la consulta y su valor o posibles valores, así como la combinación de estos (AND o OR). Con el fin de tener que evitar extraer palabras clave de la pregunta, inicialmente se probó introduciendo todas las palabras presentes en la pregunta y la respuesta correcta.

Dada la tabla siguiente pregunta:

- P: Cual es el récord de España de 100 metros lisos?

- R: 10.06

La consulta quedaría de la siguiente forma:

- C: "Question=(Cual AND es AND el AND récord AND de AND España AND de AND 100 AND metros AND lisos) AND Answer=(10.06)"

AND o OR. Como se ve en el ejemplo anterior es necesario decidir si se establece una inclusión de términos y campos más o menos restrictiva construyendo la consulta con AND o OR. Conceptualmente AND es más correcto, ya que lo que se busca es que todas las palabras por las que se consulta se encuentren dentro de la pregunta por la que se está realizando la búsqueda. Sin embargo a efectos prácticos esto resultaba ser demasiado restrictivo. Al exigir la existencia de todas las palabras, en pocos casos devolvía algún resultado. Es por ello que se decidió que la mejor opción es separar cada término con OR para solicitar su existencia pero sin la exigencia implícita en el AND, de tal forma que con que uno de ellos se localice en una pregunta ya se garantiza que se devuelva algún resultado en la consulta.

De igual forma se quiere un *matching* tanto de la pregunta como de la respuesta por lo que habría que separar los dos campos con AND. Pero separar ambos campos mediante AND también resulta ser demasiado restrictivo, ya que la respuesta (que normalmente se compondrá de 1 o 2 palabras) si no coincide exactamente con la palabra que se busca hará que falle la consulta, no devolviéndose nada. Este problema se da sobre todo a causa de los posibles **fallos ortográficos** que existan. Se asume que las preguntas validadas habrán sido correctamente introducidas en el sistema. Sin embargo la consulta se construye en base a la pregunta introducida por un usuario, por lo que una variación en las palabras si están mal escritas puede causar que no se detecte la equivalencia y falle la consulta. Así pues, se decide emplear el operador OR para constituir las consultas, lo que permite ganar en flexibilidad frente a los posibles fallos ortográficos.

Extracción de palabras. La mayor tolerancia que introduce el operador OR es un punto a favor, sin embargo esto causa problema al introducir todas las palabras de la pregunta en la consulta. Esto es así porque existen palabras muy comunes compartidas por muchos de los enunciados de las preguntas, lo que hace que se devuelvan como resultado muchas preguntas que ofrecen una coincidencia únicamente sustentada por esos términos comunes. Es por ello que, al no poder contar con una herramienta más potente, como resultaba ser NLTK, se introduce un procedimiento mínimo de revisión por el cual las palabras con menos de cuatro caracteres son eliminadas de la consulta. Aunque eso pueda suponer la eliminación de formas verbales o sustantivo cortos relevantes en algún caso, de esta forma al menos conjunciones, preposiciones y determinantes son eliminados tras pasar ese filtro tanto para el inglés como el español.

Así pues, recibida una pregunta, se le aplica un filtrado para la extracción de palabras, donde se eliminan de esta caracteres especiales, comas, puntos y signos de interrogación, transformando todos los caracteres a minúscula (ya que en la consulta las mayúsculas se reservan para los operadores). Se obtiene así un conjunto de palabras no repetidas a introducir en la consulta.

Se valoró la posibilidad de pedir al propio colaborador que además de aportar la pregunta y las posibles opciones de respuesta, aportase una serie de etiquetas precisas que definiesen la temática de la pregunta. Esto hubiese permitido obtener directamente el conjunto de palabras precisas que identifican la pregunta sin esfuerzo. Esta es una alternativa que se ha introducido con éxito en otros ámbitos como foros, donde los colaboradores etiquetan sus aportaciones para facilitar la búsqueda y relación de contenidos. Sin embargo, en este caso se determinó que podría ser más una barrera que causara rechazo en el jugador y que frenara la subida de preguntas. Por lo que finalmente se descartó asignarle esa responsabilidad extra.

Tolerancia a variaciones de la palabra. Existe un operador especial que agregado al inicio de una palabra permite aplicar *stemming*. Esto significa que en la búsqueda se aplica cierta tolerancia a las variantes de una palabra tales como singular/plural, masculino/femenino y distintas formas de un mismo verbo. Resulta útil, ya que para las búsquedas que se van a realizar se den coincidencias exactas de texto. Como para cada consulta no se espera disponer de muchas palabras, este operador es agregado a cada palabra. Recopilando todo lo anterior, el ejemplo anterior quedaría ahora de la siguiente forma:

Dada la tabla siguiente pregunta:

- P: Cual es el récord de España de 100 metros lisos?

- R: 10.06

La consulta quedaría de la siguiente forma:

- C: "Question=(cual OR record OR españa OR 100 OR metros OR lisos) OR Answer=(10.06)"

Análisis del resultado de búsqueda

Orden de los resultados. Lo más importante a la hora de configurar la consulta es definir cuál será el orden de los elementos encontrados en caso de que haya más de una coincidencia. Para ello se recurre al *MatchScorer*. Esta es la puntuación que se asigna a cada documento a la hora de evaluar una consulta de acuerdo a su concordancia. Dado que la consulta se configura en base al operador OR, se espera que cada consulta pueda devolver varios resultados, por lo que la estrategia a seguir es únicamente considerar el primero devuelto, ya que es el que ha conseguido una mayor puntuación.

Cabe considerar que la consulta puede ser configurada para devolver un máximo número de resultados, lo que para otro tipo de aplicaciones puede ser beneficioso desde el punto de vista de la optimización de la ejecución. Sin embargo en este caso no es aplicable, ya que al establecer un límite y llegar a este se para la búsqueda, y más allá aún podría haber algún documento con un valor mayor.

Emisión de veredicto. No es suficiente con tomar la pregunta que devuelva con una mayor puntuación ya que a causa de la configuración establecida (en base al operador OR) casi siempre devuelve al menos un documento. Y no se tiene garantía de que este se parezca lo suficiente a la pregunta que está bajo revisión. Es por ello que en este punto la clave reside en definir un umbral mínimo que debe cumplir ese documento devuelto con la mayor puntuación para considerar que podría ser similar a la pregunta de entrada y considerar entonces que ha de ser enviado un aviso al validador.

Puntuación asignada. Para ello conviene comprender en base a qué criterios se asigna la puntuación, lo que supone un problema en tanto que en la documentación disponible solo se indica que se realiza una búsqueda basada en la frecuencia de los términos. Por lo que se ha observado obtiene una mayor puntuación aquel documento donde más palabras hay presentes que concuerden con las introducidas en la consulta. La implicación de esto se muestra en el siguiente ejemplo:

Dada la tabla siguiente consulta:

- C: Question=(que OR clepsidra)

Se obtendrían las siguientes puntuaciones:

- P1: ¿Qué función tiene una clepsidra? - Puntuación por frecuencia: 2 (que*1 + clepsidra*1) - P2: ¿Qué ha hecho que se enfade tanto como para que no quiera volver? - Puntuación por frecuencia: 3 (que*3)

Como se para lo que sería deseable es que la pregunta P1 obtuviese mayor puntuación. Sin embargo obtiene mayor puntuación la pregunta P2, donde apareciendo únicamente uno de los términos, se asigna una puntuación mayor al aparecer uno de ellos repetido varias veces. Esto sería un problema. Sin embargo dada la naturaleza de las preguntas que se esperan recibir, muy cortas, es muy difícil hallar casos donde éstas cuenten con palabras repetidas (sin contar conectores, determinantes, etc. que no aportan sentido a la frase y tratan de ser removidos de la consulta). Así pues este factor de incertidumbre no es tal puesto que no se espera que haya palabras repetidas dentro de una misma pregunta.

Campos de la consulta. Lo que sí resultó ser un factor de incertidumbre era introducir ambos campos en la consulta, tanto el enunciado de la pregunta como la respuesta correcta. Se esperaba que introducir mayor detalle de información supusiera la recuperación más precisa de las preguntas. Sin embargo lo observado fue que hacía variar la puntuación obtenida para una misma consulta y un mismo documento, para lo que no se encontró explicación. Así pues, con el fin de favorecer la estabilización de las puntuaciones asignadas y viendo que no mejoraba en nada la incorporación de más información, se decide que es más adecuado modificar la consulta, únicamente estableciendo el campo relativo al enunciado de la pregunta.

Establecimiento del umbral. Teniendo en cuenta lo anterior no se puede establecer un umbral de puntuación absoluto para cualquier documento y cualquier consulta. Este debe ser relativo al número de palabras incorporadas a la consulta, ya que cada coincidencia hace incrementar la puntuación asignada. Aun sabiendo esto, parece haber otros factores que afectan a la puntuación que no se han podido deducir, por lo que no resulta sencillo definir el umbral para conseguir una fiabilidad de detección muy precisa. Es por ello que, como se indicaba anteriormente, parecía preferible incorporar este sistema como soporte y que, detectada una posible repetición, el validador determine si en efecto lo es o no.

Aunque un enfoque basado en lógica difusa podría haber aproximado mejor el resultado, definiendo en base al contexto (características de la consulta y el documento) qué se considera una puntuación elevada o baja, finalmente se definió en base al comportamiento y valores observados una constante que, multiplicada por el número de palabras de la consulta, determina el umbral mínimo a superar de puntuación para que se emita el aviso al validador, como se muestra en la Ecuación 5.1.

$$umbral = constanteAjuste * nPalabrasConsulta \quad (5.1)$$

5.8 Perfil de colaborador

Módulo	Gamificación
Identificador	US_UPLOAD_02 (cuadro 5.12)
Problemática	Al margen del propio altruismo que pueda mover a los jugadores a colaborar con el juego sugiriendo preguntas y lo que les motive el hecho de que muchos otros jugadores puedan ver con sus propias preguntas, es necesario ofrecerles algo más allá que les motive a realizar esta tarea. Es por ello que se pretende realizar una selección de patrones de diseño e interacción propios de la gamificación para potenciar la colaboración, ofreciendo a la vez al jugador una nueva funcionalidad que resulte divertida en sí misma y que actúe como un complemento más del propio juego.
Enfoque	Se pretende crear un perfil de colaborador para los jugadores registrados, como punto central donde se pueda consultar toda la información relativa a la actividad, así como logros y recompensas obtenidos. Así mismo la interacción con todas las mecánicas y recursos de gamificación se realizará a través del perfil.

Objetivos de negocio

Es común a diferentes guías y frameworks sugerir como primer paso previo al diseño el establecimiento de los objetivos de negocio. Así pues, como punto de partida conviene clarificar **qué beneficios se esperan** alcanzar con la incorporación de la solución basada gamificación (QuPI-GAm) en el juego de preguntas:

- Incorporar una **nueva funcionalidad** que resulte atrayente para los jugadores, es decir, que sea percibido por los jugadores como un añadido más a la dinámica de juego.
- **Incrementar el número de preguntas** disponibles en la base de datos del juego. Cuantos más colaboradores propongan preguntas, mayores posibilidades de obtener preguntas con la suficiente calidad como para poder ser introducidas en el juego.
- **Diversificar la temática de las preguntas** introducidas en el juego. Si se consigue llamar la atención de diferentes perfiles de personas, más posibilidades de encontrar colaboradores que propongan preguntas creativas y originales por un lado o específicas a temáticas concretas por otro. También pueden recopilarse de esta forma preguntas de carácter local, que ensalcen el conocimiento asociado a una región con el que puedan identificarse un determinado grupo de jugadores.

Todo lo anterior redundaría en una mejora del juego, al conseguirse un **producto más completo**. En gran medida el tiempo de vida de un juego de preguntas depende de su capacidad de sorprender al jugador a través de estas. Así, el hecho de disponer de más preguntas y más variadas reduce el riesgo de que el jugador se canse del juego por encontrarse pronto con preguntas repetidas. Por lo que el beneficio debe obtenerse por dos vías:

- Ofrecer un juego más completo debe mejorar la percepción que de este tienen los jugadores y su reputación. Se espera que resulte más llamativo gracias a las nuevas preguntas, y por lo tanto se registre un **incremento del número de nuevos jugadores**.

- Al incorporar una nueva funcionalidad complementaria al propio juego, esto puede repercutir en un aumento de la atención que el jugador le dedica. Esto se debe a que el jugador puede que en momentos concretos no quiera iniciar una partida, pero sí revisar su perfil y proponer alguna pregunta que se le haya ocurrido. Concretamente, **el número de sesiones abiertas por jugador se vería incrementado**, más aun si las técnicas de gamificación funcionan.
- Si el juego consigue enganchar en primera instancia al jugador cuando lo descarga y prueba, incorporar un mayor volumen de preguntas variadas debe poder incrementar el tiempo de vida del juego. De esta forma, se consigue **aumentar la retención**, lo que puede repercutir en un aumento de ingresos a causa de la publicidad incorporada en el juego e incrementar las posibilidades de que los jugadores hagan uso de las compras integradas en la aplicación.

Usuarios objetivo

Dado que la gamificación se dedica a estimular la motivación del usuario, el diseño de esta parte debe ir directamente orientado hacia este. Es por ello que en parte la aproximación a un diseño gamificado debe estar basada en la filosofía del UCD. A tal efecto, de forma previa al diseño es importante esbozar cuales son las características de los usuario que han de interactuar con el sistema. En primer lugar se manifiestan las características del público al que va destinado un juego de preguntas, para después determinar qué subconjunto de estos son los receptores ideales para un sistema como QuPI-GAm.

- **Nacionalidad** Este parámetro depende de la estrategia de lanzamiento e implantación que haya seguido el juego. Para el caso concreto de *Whizz* el interés se centra en la comunidad hispanohablante y angloparlante, lo que abarca distintas regiones. Dado que es un juego de tipo online que ha de comunicar personas de distintos lugares es difícil centrar el diseño en rasgos muy marcados y característicos de una zona. Conviene apostar por un diseño que pueda ser ampliamente aceptado.
- **Nivel de conocimientos** Dado el formato que se emplea para poder contestar las preguntas, donde se ofrecen las posibles respuestas, el nivel de conocimientos no supone una barrera ni una brecha que determine el tipo de usuario, ya que aquellos menos ilustrados pueden intentar contestar en base a las opciones que se ofrecen mediante razonamiento, descarte o intuición. Por su parte, dado que las preguntas abarcan un abánico amplio de temáticas, cualquier persona especializada en un área puede encontrar estímulo tanto en saber contestar las preguntas de su ámbito como en aprender aquellas que desconoce, una vez desvelada la respuesta correcta.
- **Franja de edad** En principio el público al que va dirigido un juego de tipo preguntas ha de ser bastante amplio, siendo accesible a cualquiera que quiera retarse a sí mismo comprobando su nivel de conocimientos o bien midiéndose frente a otros. Al estar

basado en una dinámica sencilla y entretenida, cabe esperar un público de carácter familiar, lo que significa que idealmente se tendrá un público conformado por niños, jóvenes, adultos y personas mayores. No obstante, dado que existe una brecha entre el nivel de conocimientos que puedan tener los niños frente a los restantes grupos de edad, los niños pueden descartarse como grupo mayoritario, a no ser que el juego se haya pensado específicamente para ellos y contenga preguntas acordes, (no siendo el caso de *Whizz*) en cuyo caso la exclusión iría en el sentido contrario.

- **Habilidad tecnológica** El juego de tipo preguntas es más amigable que otros al requerir interacciones simples, por lo que resulta ser una buena alternativa de juego para aquellas personas que no tienen mucho manejo con los dispositivos táctiles. Este parámetro no supone un factor crítico condicionante y por tanto se puede llegar a un público más amplio no tan familiarizado con los videojuegos.
- **Contexto de uso** Existen juegos para móvil que dan pie a sesiones de juego cortas y otros que requieren un periodo mayor para completar una partida. Por lo general, en el caso de los juegos de preguntas las partidas constan de una serie de turnos que obligan a dedicar cierto tiempo si se quieren completar. Es por ello que la actitud del jugador frente al juego implica cierta atención, es decir, se reserva un tiempo para la dedicación exclusiva al esparcimiento por esta vía.

Como se desprende del anterior análisis el público objetivo de QuPI-GAm sería de **naturaleza diversa**, lo que resulta adecuado al contribuir a la diversidad de las preguntas que han de recibirse. Siguiéndose la **técnica “Personas”** propia del UCD (véase [Blo06] como referencia), en el anexo A se han incluido las fichas de dos usuarios prototípicos específicos para QuPI-GAm, teniendo en cuenta para su elaboración las consideraciones recogidas a continuación, que acotan las características previamente mencionadas:

- Es extrapolable el factor asociado a la nacionalidad, así como la consideración de personas que puedan tener más o menos conocimientos y un mayor o menor desempeño y hábito de uso de las tecnologías y los videojuegos.
- En cuanto al factor edad **el interés se centraría en un público joven y adulto**, ya que en estos grupos cabe esperar la unificación de dos aspectos: capacidad e ímpetu por explorar las vías que ofrece un juego (ya que QuPI-GAm es un módulo complementario a la funcionalidad principal) y conocimientos en una temática concreta. Respecto a este último punto, las personas adultas se presuponen más experimentadas y pueden tener mucha información que ofrecer; las personas jóvenes a través de sus *hobbies* puede conocer datos interesantes y aportar información lúdica y de actualidad; por otro lado el espectro adolescente, al encontrarse en periodo lectivo, puede aportar un considerable volumen de datos de distintas materias.

- Respecto al contexto de uso se tiene el caso contrario, dado que esta funcionalidad complementaria presenta una dinámica diferente. Si bien un jugador debe iniciar a consciencia una nueva partida que requerirá varios turnos para ser finalizada, para el caso de QuPI-GAm debe poder proponer una nueva pregunta o consultar su perfil de forma rápida y sencilla, ya que esto favorece que haga uso de esta utilidad. Se espera así una actitud activa por parte del jugador, que esté donde esté y en cualquier momento pueda iniciar una **sesión corta** y proponer una pregunta. Para periodos cortos de tiempo, la simple consulta del perfil puede suponer una vía de distracción.

5.8.1 Incorporación de elementos de juego

A la hora de proponer una buena solución de diseño basada en la incorporación de elementos de juego, es necesario tener en cuenta que no basta con introducir estas de cualquier forma en el sistema que pretende ser gamificado. Como se pone de relieve en [Y.16] no es tan determinante la elección de mecánicas como la consideración de dónde, cuándo y por qué se introduce cada una de ellas. Esto es así en la medida que las mecánicas no constituyen un fin en sí mismo, sino que son solo un medio para alcanzar un fin. La finalidad es lo primero que hay que considerar cuando se inicia el proceso de diseño. Por lo general, siempre irá orientada a establecer qué es lo que se quiere provocar en el usuario objetivo desde el punto de vista de los sentimientos, escogiendo así en base a esto las mecánicas que permitan conducir la experiencia para conseguir generar lo esperado en el usuario.

¿Cuál es la finalidad perseguida? Como ya se ha nombrado previamente, el sistema QuPI-GAm está orientado a estimular la colaboración de los usuarios que forman parte de la comunidad de jugadores del juego de preguntas donde ha de ser integrado. Más concretamente, desde el punto de vista de la motivación intrínseca (de interés al despertar en el usuario una necesidad de acción de carácter voluntario) se quiere generar una satisfacción basada en el uso del sistema apoyada en los dos puntos que se recogen en el cuadro 5.1.

Ser parte de...

Para el jugador puede resultar satisfactorio que se le permita ser parte del juego y que sienta que su esfuerzo es útil y necesario. Esto genera un sentido de **pertenencia a la comunidad** que ha creado el juego y favorece la identificación con este. Por esta parte se trabajaría la motivación generada por un **aspecto altruista**. Así el reconocimiento resulta clave para que el jugador pueda percibirlo.

Poder expresarse...

Ofrecer la posibilidad de que el jugador trate cualquier tema es interesante puesto que se asume que de forma natural querrá proponer preguntas sobre temáticas que le resulten atrayentes. Los hobbies son unos de los ejemplos paradigmáticos en cuanto a lo que a motivación intrínseca se refiere. Dar la posibilidad al jugador de que aporte un **conocimiento que le apasiona** y que este vaya a llegar a otras personas gracias al juego resulta ser otra de las claves de la motivación. También es aplicable a **jugadores creativos** que quieran demostrar su ingenio creando preguntas originales. Desde este punto de vista, el reconocimiento vuelve a ser el elemento que puede reforzar esta conducta.

Cuadro 5.1: Qué se quiere despertar en el usuario (motivación intrínseca)

Características incorporadas. Según se ha referido anteriormente, el grueso de la estrategia a seguir ha de ir orientada a poder medir y reconocer la aportación de cada usuario. Teniendo eso en consideración se presenta en el cuadro 5.2 una recopilación de posibles mecánicas planteadas que podrían ser incorporadas al sistema QuPI-GAM. Si bien no todas han sido finalmente incorporadas al sistema (siendo estas las que aparecen en gris) se incluyen igualmente en la tabla, ya que se plantearían como opciones de cara a trabajo futuro. Así mismo, en la figura 5.15 se sintetizan los elementos que entran en juego en cuanto a la gamificación aplicada en QuPI-GAM y aquellos que quedan fuera (áreas oscurecidas).

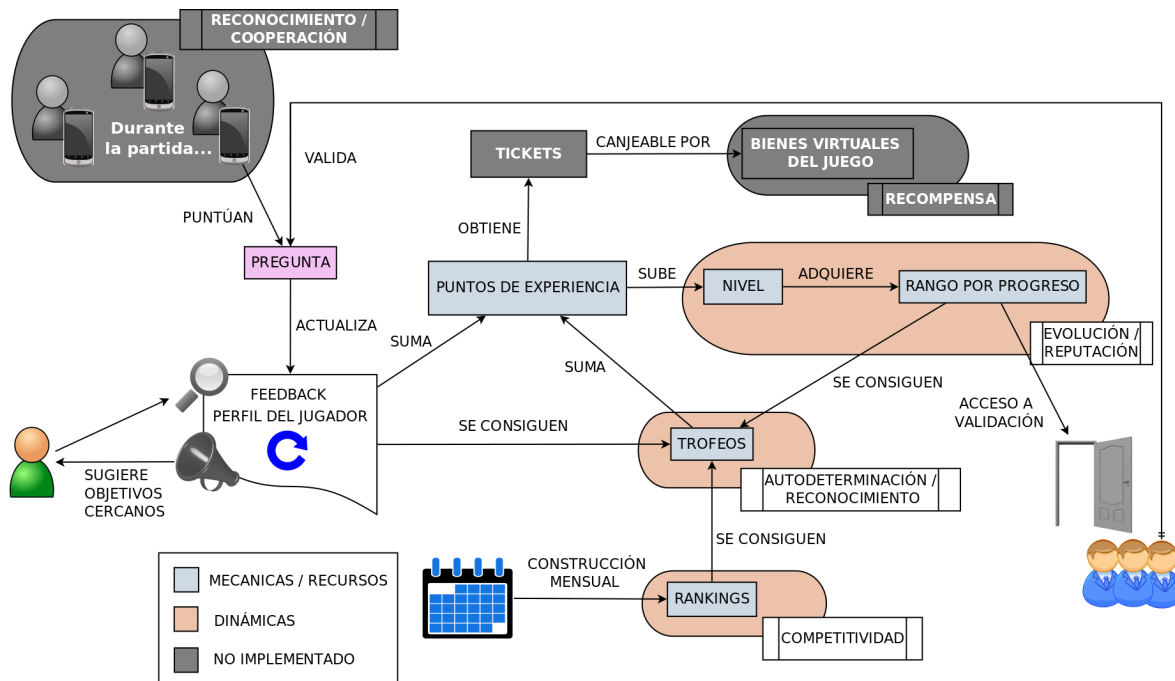


Figura 5.15: Mecánicas y dinámicas de juego incorporadas en el diseño inicial

Descripción	Dinámicas
Consultar la actividad. El jugador puede consultar su actividad en términos de preguntas aportadas al sistema, aprobadas y rechazadas. Esto permite conocer el impacto de su aportación.	Feedback
Puntos de experiencia, nivel y rango. Todas las acciones que realice el jugador como colaborador repercutirán en la adquisición de puntos de experiencia. Estos permiten medir cómo de significativa es su aportación. A medida que se consiguen puntos el jugador va subiendo de nivel y adquiriendo diferentes rangos, que le otorgan un cierto estatus respecto al resto de jugadores. Si bien en la literatura existe una fuerte crítica a los sistemas que reducen la gamificación al modelo PBL, se considera que éste sí que puede resultar útil como base para establecer un marco para la gamificación, gracias al cual el jugador puede posicionarse fácilmente.	Feedback, Evolución
Tickets canjeables. Tendiendo más hacia la motivación extrínseca, se trataría de poder adquirir tickets canjeables por bienes virtuales existentes en el juego al que se incorporase QuPI-GAm. Se ofrecería así una recompensa más explícita al jugador con la que podría mejorar su experiencia de juego, a través de la adquisición de nuevos ítems virtuales, lo que a su vez es positivo porque incita al jugador a seguir jugando al juego.	Recompensa
De colaborador a validador. Cuando un colaborador llega a cierto nivel, se le pasa a considerar como colaborador destacado y confiable, por lo que se le da acceso a un nuevo nivel de colaboración donde él mismo puede aprobar o descartar preguntas de otros usuarios.	Reconocimiento, Reputación
Trofeos. Se establecen una serie de metas que permiten guiar y registrar la progresión del jugador, guiarle y retarle en cierto sentido. El jugador debe poder consultar todos los trofeos que se pueden adquirir y comprobar aquellos adquiridos. Por cada trofeo adquirido se consiguen puntos de experiencia.	Autodeterminación, Reconocimiento, Evolución
Sugerir metas cercanas. Resulta de interés incorporar un mecanismo que sugiera al jugador qué metas (en función de los trofeos definidos) está cerca de conseguir. De esa forma se le puede animar a que realice una aportación extra.	Feedback
Rankings mensuales. Aunque la competitividad no es el sentimiento principal que quiere fomentarse en los colaboradores, resulta interesante incorporar como recurso la creación de rankings con los mejores jugadores en base a diferentes parámetros. Esto se basa en una mecánica de restricción de tiempo, en tanto que los rankings serán mensuales, para evitar que ciertos jugadores se puedan posicionar alto y que todos, al inicio de cada mes, puedan intentar conquistar el número 1 de alguno de los rankings.	Competitividad, Reputación, Reconocimiento
Recibir puntuación de otros usuarios. Al margen de la aceptación que recibía una pregunta por parte de un validador, es interesante poder medir cómo de buena le parece la pregunta a la comunidad de jugadores. Para ello se puede incorporar un mecanismo en el propio juego en la fase de integración (no implementable en el prototipo) por el cual durante la partida las preguntas puedan ser puntuadas de forma rápida y fácil por los propios jugadores. Así podría disponerse de un valor promedio que midiese la excelencia de cada colaborador en términos de su contenido, y no únicamente de su actividad.	Reconocimiento, Cooperación

Cuadro 5.2: Proyección de mecánicas que podrían formar parte de QuPI-GAm

5.8.2 Información del perfil

El perfil constituye la interfaz de entrada a través de la cual el jugador puede recibir todo el *feedback* asociado a su actividad. A la hora de incorporar la información es mejor apostar por un diseño sencillo y claro, para que el sistema sea fácil de usar y esto no suponga una barrera. De entrada la información que se muestra se centra en dos aspectos:

- **Actividad.** Se informa al jugador del número de preguntas aportadas, del número de preguntas validadas y del número de preguntas que fueron revisadas y descartadas.
- **Progreso.** Es necesario definir una unidad de medida que permita cuantificar y valorar las acciones llevadas a cabo por el jugador. A tal efecto se emplea el concepto de **puntos de experiencia (XP)**. Para definir un marco comparativo o una escala se definen también **niveles** (alcanzables acumulando puntos de experiencia) que a su vez dan acceso a **rangos**, a modo de etiquetas que sitúan al jugador dentro de un subconjunto específico de niveles.

Barra de progreso. Se logra una percepción de avance difusa incorporando una barra de progreso donde se muestran los puntos totales acumulados. No se informa al jugador exactamente de cuántos puntos debe conseguir para superar cada nivel. Así, al visualizar la barra cerca de completarse esto le puede animar a interactuar más con el sistema para alcanzar el nuevo nivel.

Registro de cambios

Algo característico del perfil es que para reforzar la sensación de avance, cada vez que se accede además de mostrar la información actualizada, **señala los parámetros que han mejorado** desde la última visita registrada al perfil. De esta forma, de un vistazo se perciben los cambios. Desde el punto de vista del diseño esto implica mantener de alguna forma en la entidad destinada a almacenar la información del perfil de un jugador dos estados del mismo. En la figura 5.24 se puede consultar la definición de la entidad *UploaderProfile*. Como puede apreciarse, para el aspecto comentado, se solventa incorporando la fecha de última visualización en el perfil y agregando un tipo de dato que admite un JSON, es decir, un diccionario donde se mapea para cada valor medible almacenado de por sí en el perfil (valores que sí estarían actualizados) los cambios de forma acumulativa. Ante una nueva visita del perfil basta con extraer sus valores para conocer en qué medida ha cambiado la variable; así como resetear el contenido del diccionario.

Constitución del sistema de puntos y niveles

Los puntos de experiencia se conciben como la **moneda de cambio** principal para recomensar a los jugadores colaboradores. Todo aquello que se consiga y se premie redundará en un aumento de puntos de experiencia, por lo que toda acción se canalizará hacia este

concepto. Los puntos de experiencia representan recompensa en dos sentidos. De forma implícita, a mayor cantidad de puntos mejor situado se está y mejor reputación se tendría en comparación a otros jugadores. De forma explícita, podrían ser canjeables de forma que cada cierto número de puntos de experiencia conseguidos el jugador obtuviese tickets canjeables a su vez por otro tipo de bienes virtuales o beneficios ofrecidos dentro del juego. Este último aspecto sin embargo solo se plantea como opción a considerar.

Ya se ha referido que para establecer un marco comparativo cada cierto número de puntos de experiencia conseguidos se sube de nivel y que cada cierto número de niveles se consigue un rango. Los dos primeros conceptos son numéricos. El tercero actúa a modo de etiqueta identificativa. Aunque en el sistema web de base no tiene un impacto directo el hecho de ser dueño de uno u otro rango, este podría ser útil de cara a otorgar privilegios en base a méritos a un cierto grupo de jugadores. La relación existente de consecución entre los tres conceptos depende del **sistema de puntuación** definido, lo que no resulta una tarea trivial.

Sistema de puntuación de Pokemon GO. Conviene tomar como reflejo un sistema de puntuación que haya resultado ser exitoso, como es el caso de Pokemon GO. A nivel general define un sistema de puntuación dinámico donde a medida que se avanza en el juego se van consiguiendo ítems y multiplicadores que permiten conseguir cada vez más y más puntos de experiencia. Teniendo esto en cuenta, tiene sentido que establezcan un criterio un tanto estricto a medida que se sube de nivel. En la figura 5.16^a se muestran los puntos que hay conseguir en cada nivel para pasar al siguiente. Puede apreciarse que los incrementos de un nivel a otro son considerables, manteniéndose en algunos casos en el 0 %, pero en otros exigiéndose un 25 %, un 50 % y hasta un 100 % más de puntos respecto al anterior nivel, como es el caso del paso del nivel 20 al 21. Sin embargo el modelo en su definición es sencillo, puesto que para cada nivel se decide un incremento porcentual concreto respecto al nivel anterior.

^aFuente:<http://www.vandal.net/guias/guia-pokemon-go/niveles-y-recompensas>

Teniendo (a grandes rasgos) el modelo anterior como referencia, no convendría sin embargo aplicar el mismo criterio por las razones que se exponen a continuación. En su caso, el sistema de asignación de puntos es dinámico, lo que significa que la forma en que se consiguen los puntos es adaptativa y da pie, por ejemplo, a multiplicar la experiencia percibida ante una acción. Sin embargo dado que el sistema que se está manejando con QuPI-GAM es más sencillo y acotado que este, no se tiene tanta capacidad para incorporar también una asignación dinámica. Así pues, se tiene un **sistema de asignación de puntos estático**. De inicio se definen las reglas para conseguir puntos y cuántos se consiguen y eso se mantiene inmutable a medida que se avanzan niveles. Dado este aspecto estático, no conviene imponer un crecimiento tan pronunciado en los requerimientos de puntos de experiencia necesarios para subir de nivel.

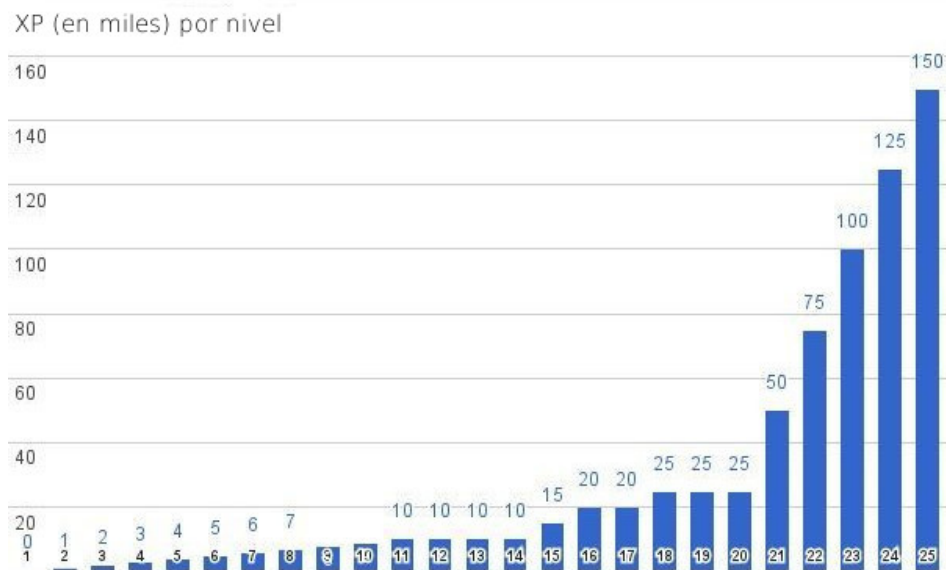


Figura 5.16: Puntos por nivel en el juego Pokemon GO

Mapeo puntos-niveles. Teniendo en cuenta lo anterior, lo que se pretende es ir incrementando muy suavemente la cantidad de puntos exigidos para alcanzar el siguiente nivel. Para definir esta progresión se plantea definir el umbral de puntos totales que hay que superar para acceder a cada nivel. En la gráfica de Pokemon GO se definen los puntos de experiencia por paquetes de forma relativa, es decir, que para subir al siguiente nivel hay que conseguir x puntos más de los que ya se tienen. Sin embargo para definir el modelo en QuPI-GAM se ha creado una fórmula que, dado el nivel, devuelve los puntos totales (de forma absoluta) que se han tenido que conseguir para alcanzarlo. Esta fórmula se expone en la ecuación 5.2.

$$XP = \sqrt{N^4} + 100 \times (\log_{10} \sqrt{N^N}) + 50 \quad (5.2)$$

En la figura 5.17 se incluye un gráfico que da idea del comportamiento de esta fórmula, donde se observa que describe una **curva creciente poco pronunciada**. Para alcanzar el nivel 100 se necesitan 30000 puntos de experiencia (muchos menos respecto a los millones de puntos requeridos en el sistema tomado como referencia). Dando alguna cifra, en cuanto a los incrementos relativos en Pokemon GO en el nivel 20 se necesitan 25000 puntos y en el nivel 21 50000 (incremento del 100 %) mientras que el modelo definido, en el nivel 20 se necesitan 211 puntos y en el 21 se requieren 216 (incremento del 2,3 %) puntos. Al escogerse una escala menor se espera que el jugador valore más cada punto recibido.

Mapeo niveles-rangos. Dado que los rangos no son infinitos como los niveles y son un número reducido, esta asignación no se realiza mediante fórmula (aunque se podría, dado que cada rango tiene un identificador numérico asignado además de su cadena de texto). Directamente se asigna a cada uno de los diez rangos definidos el nivel mínimo necesario para ser alcanzado. En el cuadro 5.3 se listan los rangos definidos.

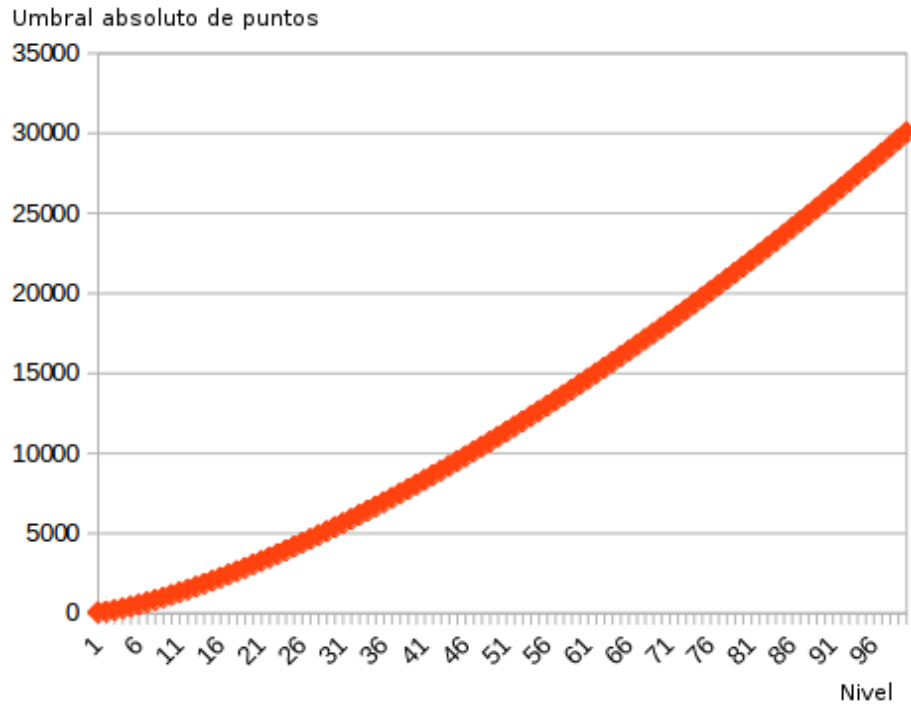


Figura 5.17: Puntos totales (absolutos) necesarios que dan acceso a cada nivel

Español	Inglés	Nivel mínimo
Iniciado	Beginner	0
Colaborador junior	Junior collaborator	4
Colaborador senior	Senior collaborator	14
Colaborador experto	Master	24
Autoridad	Authority	49
Erudito	Erudite	74
Gurú de las preguntas	Question's Guru	89
Colaborador legendario	Whizzard	99

Cuadro 5.3: Rangos definidos

Para mantener un equilibrio, es importante **balancear los puntos** conseguidos por cada acción, para conseguir retar adecuadamente al jugador respetando el *momentum*, una de las 3 Ms aludidas en el estado del arte. Es igualmente malo que resulte ser muy sencillo subir de nivel como que resulte ser una tarea demasiado ardua. Existen dos eventos que desencadenan la consecución de puntos: la aprobación de una pregunta por un lado y la adquisición de trofeos, que según de qué tipo sean aportan más o menos puntos (este concepto se comenta más adelante en la sección 5.9).

5.9 Incorporación de trofeos

Módulo	Gamificación
Identificador	US_UPLOAD_02.1 (cuadro 5.13)
Problemática	Se necesita incorporar un mecanismo que permita sistematizar las acciones o méritos que puede llevar a cabo el jugador y que le serán recompensados.
Enfoque	Se emplea la metáfora del trofeo para definir en base a un tipo, un subtipo y una serie de parámetros combinables la lista de metas medibles que se le pueden sugerir al jugador.

5.9.1 Tipos de trofeos definidos

Así como los puntos permiten medir la recompensa obtenida según las acciones realizadas, los logros constituyen una herramienta que permite establecer los objetivos alcanzables dentro de QuPI-GAm. A través de estos el jugador tiene una idea clara de lo que de él se espera. La idea de logro se ha materializado en el sistema bajo el concepto de trofeo. A lo largo de esta sección se detalla cómo han sido introducidas estas metas y cómo son mostradas.

En primer lugar es necesario definir qué tipos de trofeos se pueden incorporar. Hay que tener en cuenta que todas aquellas acciones que puede realizar el jugador y son medibles son susceptibles de tener su reflejo en uno o varios trofeos. En QuPI-GAm finalmente se han incluido tres tipos de trofeos, más un tipo que fue descartado. Estos constituyen unos trofeos de base a los que se podrían incorporar otros a medida que se proyectase la incorporación de más funcionalidades. Para explicar los trofeos existentes, primero es necesario explicar el **modelo seguido para facilitar su definición**. Es importante escoger una estructura que permita almacenar todos los tipos de trofeos y la información asociada de la forma más uniforme posible para facilitar luego la comprobación de qué trofeos se han conseguido dada una acción. A tal efecto se incorpora en 5.1 directamente la estructura de diccionarios anidados que se ha empleado.

La definición de esta estructura no resultó trivial, teniendo en cuenta que había que buscar un diseño que unificase la información de los distintos trofeos para poder comprobarlos (sean del tipo que sea) bajo un procedimiento lo más estándar posible.

Cabe destacar los siguientes aspectos acerca de la estructura:

- Se define un primer lugar un **tipo general** que diferencia claramente cada tipo de trofeo. Este tipo se asocia al concepto con el que se relaciona dentro de QuPI-GAm.
- Se definen unos **subtipos de trofeos** que pueden ser comunes a distintos tipos y que apuntan a ciertos parámetros que aplicados al tipo permiten generar distintas combinaciones de trofeos.

```

TROPHIES = {
  FROM_QUESTION: {
    TOTALS: {
      GOALS:[1,10,25,50,100,250,500]],
      KIND: { BRONZE: 25, SILVER: 100, GOLD: 500 },
      EXTRA_PARAMETERS:[GOAL]
    },
    TOTALS_BY_CATEGORY:{
      GOALS:[5,10,25,50,100,250,500], [1,5,10,25,50,100,250]],
      KIND: { BRONZE: 25, SILVER: 100, GOLD: 500 },
      EXTRA_PARAMETERS:[GOAL, CATEGORY]
    },
  },
  FROM_RANKING: {
    KIND value represents position in ranking, not goal
    TOTALS_BY_QU_CATEGORY_POSITION: {
      GOALS:[[1]],
      KIND: { BRONZE: 10, SILVER: 3, GOLD: 1 },
      EXTRA_PARAMETERS:[CATEGORY, POSITION]
    },
    TOTALS_BY_XP_LOCATION_POSITION: {
      GOALS:[[1]],
      KIND: { BRONZE: 10, SILVER: 3, GOLD: 1 },
      EXTRA_PARAMETERS:[LOCATION, POSITION]
    },
    TOTALS_BY_QU_LOCATION_POSITION: {
      GOALS:[[1]],
      KIND: { BRONZE: 10, SILVER: 3, GOLD: 1 },
      EXTRA_PARAMETERS:[LOCATION, POSITION]
    },
    TOTALS_BY_QU_LOCATION_CATEGORY_POSITION: {
      GOALS:[[1]],
      KIND: { BRONZE: 10, SILVER: 3, GOLD: 1 },
      EXTRA_PARAMETERS:[LOCATION, CATEGORY, POSITION]
    },
    TOTALS_BY_XP_GLOBAL_POSITION: {
      GOALS:[[1]],
      KIND: { BRONZE: 10, SILVER: 3, GOLD: 1 },
      EXTRA_PARAMETERS:[POSITION]
    },
    TOTALS_BY_QU_GLOBAL_POSITION: {
      GOALS:[[1]],
      KIND: { BRONZE: 10, SILVER: 3, GOLD: 1 },
      EXTRA_PARAMETERS:[POSITION]
    },
  },
  FROM_XP:{
    TOTALS:{
      GOALS:[range(len(UPLOADER_RANKS))],
      KIND: { BRONZE: 2, SILVER: 6, GOLD: len(UPLOADER_RANKS) },
      EXTRA_PARAMETERS:[GOAL]
    },
  },
}

```

Listado 5.1: Definición de los trofeos existentes en QuPI-GAm

- Dentro de cada subtipo hay que tratar de definir las distintas **metas impuestas de forma numérica**, lo que se define a modo de lista en la clave *GOALS*. Estos números tendrán una semántica asociada diferente según el tipo de trofeo que representen.
- Dado que hay metas más sencillas y metas más complejas, se definen **tres tipos de trofeos** a los que corresponde distinta puntuación en *GOALS*: bronce, plata y oro. Se incorporan en la estructura unos valores indicadores que representan para cada trofeo a partir de qué meta superada se asignaría este tipo.
- Por último, dado que el subtipo es únicamente una etiqueta semántica, es necesario **parametrizar las variables aplicadas al trofeo**, lo que se consigue mediante *EXTRA_PARAMETERS*, donde se incluyen unas etiquetas estándar que representan cada parámetro implicado.

Una vez vista la estructura pasan a describirse los tipos de trofeos considerados, teniendo en cuenta que lo que se busca es reconocer tanto la cantidad como la variedad de preguntas aportadas:

- **En base a las preguntas validadas.** Dado que las preguntas aportadas por el jugador pueden no ser de calidad, en base a la aportación no se definen trofeos. En cambio una vez validadas sí que suponen una acción de valor llevada a cabo por el jugador. Así pues, se define un trofeo que mide el número de preguntas de valor asignadas al jugador. Considerando el total de preguntas y segmentando el total en base a la categoría de la pregunta se generan dos tipos de trofeos:
 - Se han aportado un total de [X] preguntas
 - Se han aportado un total de [X] preguntas de la cat. [C]
- **En base al ritmo de aportación[DESCARTADO].** Cuando se trata de gamificación jugar con el impacto del tiempo como variable estableciendo márgenes donde se deben llevar a cabo acciones es útil para motivar esta. Inicialmente se planteó la posibilidad de definir un trofeo donde se reconociese el número de preguntas validadas dentro de un mismo mes. Este sería una especialización del trofeo basado en preguntas, en el que no se pone cota de tiempo. Sin embargo, se concluyó que podría resultar inadecuado al poder generar frustración en el jugador. Esto se debe al desequilibrio existente entre el número de jugadores y validadores(mucho menor este segundo). Un jugador podría aportar muchas preguntas dentro de un mismo mes, pero si los validadores no tienen capacidad para afrontar a buen ritmo la revisión de todas las preguntas que se van recibiendo, el jugador no llegaría a conseguir este trofeo. Esta dependencia respecto a los validadores lleva al descarte.
- **En base a los rankings mensuales.** Mensualmente se elaboran una serie de rankings combinando distintos parámetros (los mismos que afectan a los trofeos). Esos parámetros se basan en localización, categoría y un criterio (siendo este número de preguntas

o número de puntos de experiencia). Aquí la meta numérica no alude a totales (es decir, número de veces que se ha estado en el ranking) sino directamente a la posición en el ranking. Para cada tipo de ranking construido solo pueden conseguirse tres trofeos, de acuerdo a la posición lograda en este (primera posición, en el podio o en el top diez). Sabiendo esto se generarían las siguientes combinaciones:

- Se ha terminado un mes en la pos. [P] por nº preguntas (global)
 - Se ha terminado un mes en la pos. [P] de la cat. [C] por nº preguntas
 - Se ha terminado un mes en la pos. [P] de [L] por nº preguntas
 - Se ha terminado un mes en la pos. [P] de [L] para la cat. [C] por nº preguntas
 - Se ha terminado un mes en la pos. [P] por XP (global)
 - Se ha terminado un mes en la pos. [P] de [L] por XP (global)
- **En base a los puntos de experiencia.** Dado que los puntos de experiencia dan acceso a distintos niveles, y estos a su vez a distintos rangos, por cada rango alcanzada se consigue un trofeo. Este aspecto no acepta ninguna combinación con parámetros. El tipo de trofeo generado sería:
- Has alcanzado el rango [R]

5.9.2 Dinámica para la asignación de trofeos

Para hacer funcionar la operativa asociada a los trofeos hay que considerar una serie de aspectos. Se tratan estos en una secuencia ordenada según serían requeridos de acuerdo a la interacción del jugador.

Consulta del listado de trofeos disponibles

El jugador debe poder consultar la lista de todos los trofeos disponibles. Esto implica solicitar la lista al servidor. Dado que la información de los trofeos disponibles se tiene registrada de forma esquemática es necesario aplicar un primer formateo en el servidor y un segundo formateo desde el cliente a la información recibida para constituir la lista que será finalmente agrupada por tipos y desplegada.

Formato en servidor. Por parte del servidor la información se devuelve agrupada en un diccionario bajo el esquema presentado en 5.2. Para cada tipo y subtipo anidado se devuelve como valor una lista. Cada elemento de la lista representa un trofeo. Cada trofeo se representa como una lista de dos elementos. Un elemento especifica qué tipo de valoración se le da (oro, plata, bronce) y el otro es una lista con el valor de los parámetros que configuran ese trofeo específico. Los parámetros siempre se colocan en un orden preestablecido, para facilitar el mapeo de estos a su campo correspondiente en el enunciado asociado cuando se mapeen en el cliente. Para llevar a cabo este formateo se recorre el diccionario que define los trofeos por tipo y subtipo y para cada uno de ellos, se toman previamente el conjunto de valores asignables para cada parámetro y se generan todas las posibles combinaciones.

```

TROFEOS: {
  TIPO1:{
    SUBTIPO1: [[[objetivo,categoria,localizacion,posicion],oro/plata/bronze],...]
    SUBTIPO2: []
    ...
  },
  TIPO2:{
    SUBTIPO3: [],
    ...
  }
  ...
}

```

Listado 5.2: Formato del listado de trofeos devuelto por el servidor

Formato en cliente. En el cliente se ha de construir una colección de trofeos, que será la que se renderice a modo de listado. La información de cada trofeo se almacena en un objeto según el modelo *Trophy*. Cuando se recibe el listado en bruto del servidor se aplica un procedimiento para desgranar esa información en los siguientes campos:

- **Identificador.** A cada trofeo se le asigna una cadena de texto generada a partir de sus parámetros y que le identifica unívocamente respecto a los demás trofeos.
- **Tipo.** Se almacenan las etiquetas que indican el tipo y subtipo de trofeo.
- **Descriptor.** Dado el tipo y subtipo, en el lado cliente se dispone de un diccionario que contiene, para los distintos idiomas soportados, la cadena de texto correspondiente que se debe mostrar junto con unos caracteres comodín en la posición donde deben reemplazarse la lista de parámetros recibidos.
- **Tipo según valoración.** Se almacena si el trofeo es de bronce, plata u oro.

Disparadores para la comprobación de nuevo trofeo adquirido

A medida que el jugador interactúa con el sistema generará una actividad debido a la cual puede haber conseguido trofeos. Antes de entrar a describir cómo se comprueba si se ha conseguido un trofeo hay que determinar **en qué momento se lanza la comprobación**. La opción elegida es definir qué acciones pueden desencadenar la adquisición de un trofeo y cada vez que estas sean ejecutadas lanzar la comprobación. Se puede pensar que esto puede generar mucha carga de procesamiento, sin embargo hay que considerar que:

- Dado que al jugador se le muestra la información cuando entra al perfil y consulta expresamente los trofeos, se podría lanzar la comprobación justo en ese momento. Así si no consulta con frecuencia la lista de trofeos se ahorra procesamiento. El problema de este enfoque es que la información de los trofeos no está actualizada en la base de datos y se depende de la visita del jugador como disparador para efectuar la comprobación. Esto es un problema para la construcción de rankings, porque para esto se necesita tener todos los perfiles actualizados al final del mes.

- Para solventar lo anterior se podría configurar una tarea periódica que se lanzase a final de mes y que actualizase todos los perfiles. Se tendrían que considerar condiciones como ignorar aquellos perfiles que durante ese mes no han registrado actividad. Sin embargo supondría mucha carga de procesamiento si se consigue llamar la atención de muchos jugadores y por lo tanto se almacenan muchos perfiles. Considerando esto se ha optado por incorporar el enfoque de la actualización instantánea.

Una vez se ha determinado que para cada acción que pueda generar un nuevo trofeo se va a lanzar la comprobación, hay que determinar **cuáles son esas acciones**. Dado que la actividad nuclear que da sentido a QuPI-GAM es la aportación de preguntas, tiene sentido que la validación de una pregunta sea el principal desencadenante. Así pues cuando un validador aprueba una pregunta, deben comprobarse los trofeos concernientes a las preguntas y los puntos de experiencia para el jugador que la sugirió. Por otro lado dado que mediante los rankings también se reconoce a los mejores colaboradores de cada mes, los trofeos son la opción ideal para registrar este hecho. Por lo que cada vez que se constituyen los rankings se comprueban los trofeos concernientes a los rankings para los jugadores en el top.

La comprobación de todos los tipos de trofeos se encapsula bajo una misma función, por lo que se le pasa como argumento una etiqueta que identifica qué tipos y subtipos deben revisarse para la llamada concreta. También se deben especificar los parámetros que pueden determinar el trofeo concreto conseguido, ya que en la comprobación no se realiza como barrido, sino comprobando únicamente si se ha conseguido una combinación específica.

Para el caso de la pregunta validada, el proceso es el siguiente. Se toma el perfil del jugador al que corresponde la pregunta. Se le notifica esta circunstancia. En ese momento el perfil actualiza sus variables de actividad y lanza una comprobación de logros asociados. Esto implica actualizar las variables de progreso y ahora sí, comprobar si le corresponde algún nuevo trofeo. Dados los trofeos definidos, tiene sentido revisar todos los subtipos de trofeos concernientes a la actividad asociada a la pregunta y aquellos relacionados con los puntos de experiencia, especificando únicamente como parámetro condicionante la categoría de la pregunta.

Para el caso de los rankings, no se necesita al perfil como intermediario. Para cada jugador en una posición susceptible de conseguir trofeo se lanza la comprobación. Como es lógico, habría que indicar que se revisen únicamente los trofeos que tienen relación con los ranking, pero además especificando como subtipo el ranking específico en el cual se localizó al jugador. Según qué ranking sea habrá que pasar como parámetro la posición, la categoría de preguntas y la localización del jugador.

Comprobación de nuevo trofeo adquirido

El proceso de comprobación consta de los siguientes pasos, que se resumen en el diagrama de actividad de la figura 5.18:

- Se **toma el tipo(s) y subtipo(s)** especificados junto con los parámetros opcionales. Todo ello en combinación permite centrar la comprobación y únicamente evaluar si se han conseguido uno o varios trofeos específicos.
- El primer paso para el jugador considerado cual es **extraer el valor actual** que ostenta para la meta que mide el trofeo a evaluar. Según el tipo de trofeo dependerá la forma de conseguir ese valor, que en todo caso habrá sido definido de forma numérica.
- El siguiente paso es comprobar si el valor actual es susceptible de conseguir trofeo. Gracias a la estructura creada para almacenar la información de los trofeos esta comprobación es similar sea cual sea el tipo y subtipo de trofeo. En esta estructura se definen los objetivos merecedores de trofeo a modo de lista ordenada (una por cada tipo y subtipo). Así, basta con **comprobar si el valor actual está inscrito en la lista**. En caso afirmativo, se habrá conseguido un trofeo.

Particularizaciones. La única particularización viene de mano del trofeo que mide el número de preguntas aportadas por categoría. Dado que existen categorías generales (de primer nivel) que engloban a otras más específicas, los objetivos no son iguales para todas las categorías. Se definen en la estructura varias listas, una por cada nivel en la jerarquía de categorías, donde los objetivos son más asequibles para las subcategorías más profundas. El hecho de que una categoría dada sea hija de otras implica que en este caso no se realiza una comprobación, sino tantas como profunda sea la categoría, ya que si por ejemplo se considera una pregunta de la categoría "Fútbol" también se debe realizar la comprobación en la categoría "Deportes" que la engloba.

- Si se detecta que se ha conseguido un trofeo, el siguiente paso es **registrarlo en el datastore**. Este aspecto se detalla en el siguiente apartado (ver 5.9.2). Aquí existe un caso particular de mano de los trofeos de tipo ranking. Dado que en el diseño se decide que únicamente se registrará el hecho de haber estado presente en un determinado ranking (no el número de veces que esto ocurre), aunque al jugador le corresponda trofeo, si ya ha conseguido ese trofeo (cosa que se comprueba), únicamente se registra en esa entidad la fecha en la que se volvió a conseguir. También se le asignan al jugador los puntos correspondientes a este logro, pero no se registra como nuevo trofeo.
- Dado que se necesita vincular los recién creados trofeos al perfil del jugador, se **envía una notificación al perfil** mediante la cual vuelve a actualizar las variables de progreso. Curiosamente, dado que los trofeos aportan puntos de experiencia, es necesario lanzar de nuevo la comprobación en este punto, pero solo para el trofeo relativo a los XP, por si gracias a los puntos recibidos el jugador hubiese subido de nivel alcanzando un nuevo rango. Recordar que los puntos que aporta el trofeo depende de la valoración que tenga (oro, plata y bronce) lo que de nuevo es consultado en la estructura descriptora de trofeos donde se mapea cada una de ellas a unos objetivos determinados.

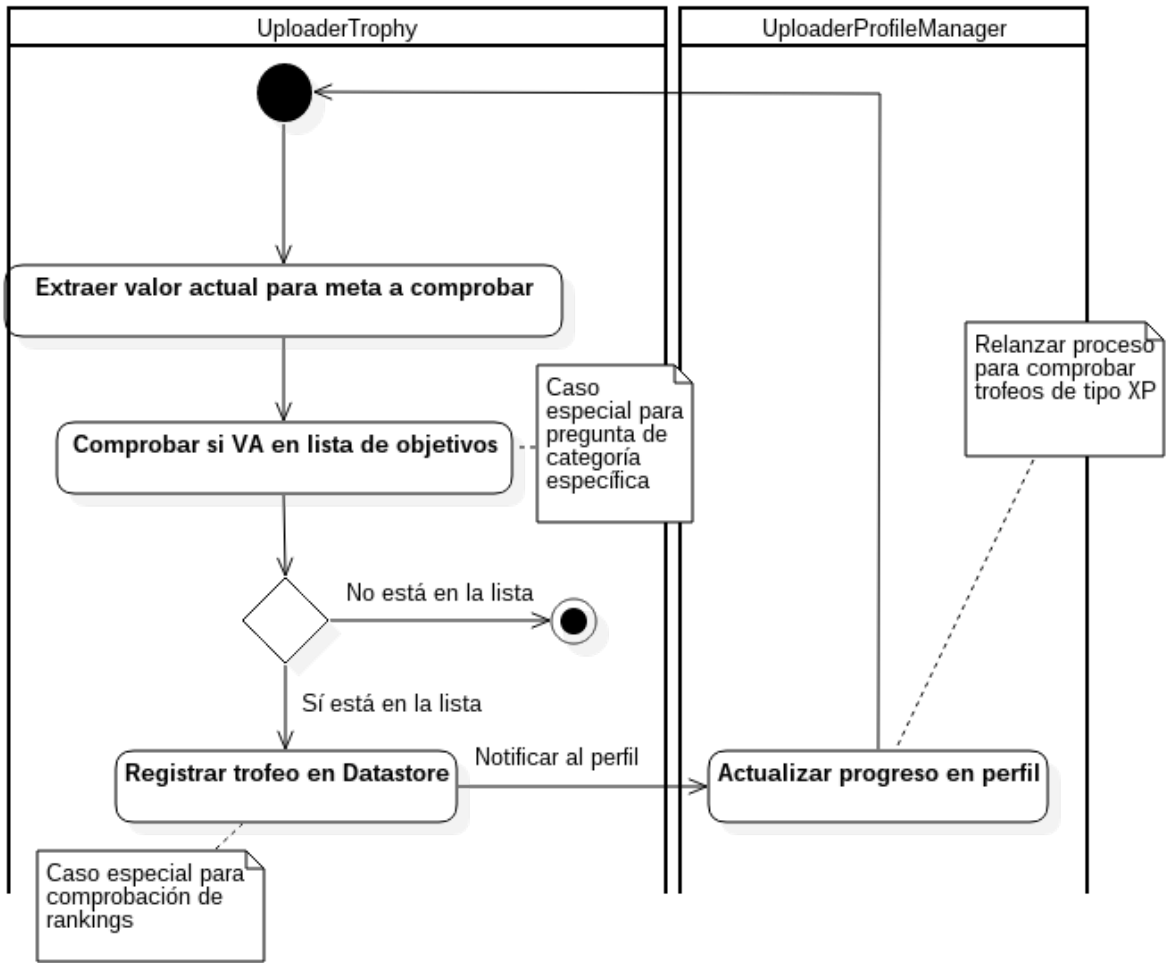


Figura 5.18: Diagrama de actividad para la comprobación de trofeos conseguidos

Almacenamiento de trofeos en el datastore

Para almacenar la información de un trofeo en la base de datos se define el modelo *UploaderTrophy* cuya especificación puede consultarse en la figura 5.24. En el modelo *UploaderProfile* se almacena una propiedad con un listado de los trofeos vinculados al jugador. Cada trofeo tiene como marcador el tipo y subtipo correspondiente e incorpora además una referencia a una propiedad estructurada donde se especifica el trofeo concreto conseguido a través de los parámetros necesarios (opcionales, según el tipo de trofeo). Además aquí se almacena una lista con las fechas en que se consiguió (lista de un elemento en todos los casos excepto en el caso del ranking, ya que se puede estar más de una vez en distintos meses en la misma posición de un mismo ranking).

Consulta de trofeos adquiridos

Inicialmente se comentó cómo se aplica el formateo tanto en el servidor como en el cliente para poder devolverle al jugador la lista completa de trofeos disponibles. Sin embargo esa es solo la primera parte del proceso, ya que una vez devuelta esta hay que marcar aquellos trofeos que ya se han conseguido. Para ello, hay que recordar que a cada trofeo se le asigna una cadena de texto a modo de identificador. Así pues es necesario recuperar todos los trofeos conseguidos por el jugador y para cada uno de ellos es necesario:

1. Obtener su cadena de texto identificativa
2. Comprobar si la fecha de obtención es posterior a la última fecha de visita al perfil. En caso afirmativo será un trofeo nuevo, lo que debe notificarse.
3. El formato de devolución consiste en un diccionario donde la clave es la cadena identificativa y el contenido es una lista que contiene la fecha en que se consiguió y opcionalmente los puntos vinculados a ese trofeo (como forma de indicar que ese trofeo es nuevo).
4. Al recibirse en el cliente, ya se dispone previamente de una colección que contiene todos los trofeos. Tomando los identificadores se localiza y actualizan fácilmente aquellos que han sido conquistados por el jugador.

5.10 Elaboración de rankings mensuales

Módulo	Gamificación
Identificador	US_UPLOAD_02.2 (cuadro 5.14)
Problemática	Se desea ofrecer al jugador algún marco que le permita compararse con otros jugadores y además recompensar a aquellos que estén a final de mes en las posiciones más altas del mismo.
Enfoque	Se implementará un mecanismo basado en consultas de ordenación que permita generar múltiples rankings en base a diferentes filtros. Estos rankings son accesibles en todo momento por el jugador, pero será a final de mes cuando una tarea periódica se encargue de construirlos y tomar las acciones pertinentes para reconocer a los jugadores bien posicionados.

5.10.1 Tipos de rankings

El fundamento de los rankings a nivel de gamificación se basa en estimular la competitividad, reputación y reconocimiento de los jugadores. Yendo un paso más allá, de cara al diseño conceptual se han tenido en cuenta dos criterios para incrementar su utilidad o la buena percepción que de ellos pueda tener el jugador:

- **Constitución de rankings mensual.** La idea es configurar una tarea periódica en el servidor que al final del mes construya todos los posibles rankings y recompense a través de la asignación de trofeos (como ya se explicó en la sección 5.9) a los jugadores bien posicionados (en el top 10). Sin embargo no se plantea la constitución de rankings

de forma absoluta en base a la actividad generada por cada jugador desde que comenzó a participar como colaborador. Si se abordase de esa forma poco a poco se generaría una diferencia inabordable por los nuevos aportadores respecto a aquellos que ya tuviesen un largo recorrido. Este sería un factor de rechazo. Así, para equilibrarlo los rankings son mensuales a todos los efectos, lo que significa que para su generación solo se considera la información del jugador relativa al último mes.

- **Multiplicidad de rankings.** Si existen muchos tipos diferentes de rankings el jugador puede localizar alguno donde le resulte más sencillo destacarse, si por ejemplo en otros más populares existe más competitividad. Es por ello que se han combinado todos los parámetros posibles para generar una cantidad considerable de rankings y que además el jugador pueda aplicar el filtro que desee.

A nivel de consulta, el jugador tiene la capacidad de consultar el estado actual de un ranking específico en cualquier momento, aunque para el reparto de trofeos solo cuenten las posiciones registradas en el momento de lanzar la comprobación periódica. El conjunto de filtros definidos para la consulta es más completo que el conjunto revisado en el reparto. A continuación se describen cuáles son:

- **Criterio.** En primer lugar hay que definir si el ranking se quiere construir en base a:
 - Número de preguntas
 - Cantidad de puntos de experiencia
- **Filtrado.** Una vez definido el criterio entra en juego la segmentación de acuerdo a:
 - **Elección de la categoría** (solo disponible si se eligió como criterio el número de preguntas). Puede especificarse una categoría de cualquier nivel.
 - **Elección de la localización.** Se puede segmentar por jugadores en base a la localización, para sus localizaciones registradas. Dado que se almacena información de localización a varios niveles (país, autonomía y provincia) se puede ajustar bastante este parámetro para tener una perspectiva de carácter local.
 - **Habilitar/deshabilitar información histórica.** Aunque los rankings válidos aptos para ser valorados son aquellos que consideran únicamente la información asociada al jugador en el último mes, al jugador se le permite ver cómo quedarían los rankings en base a toda la información acumulada desde que se inicio este módulo de aportación.

Ejemplo de consulta a la base de datos. Asumiendo que el jugador ha escogido el correspondiente ranking mensual, por número de preguntas, para una localización específica y para una categoría dada, el aspecto de la consulta emitida a la base de datos tendría el siguiente aspecto, teniendo en cuenta que **se ha empleado para su construcción GQL^a**, un lenguaje simplificado de consulta inspirado en SQL:

```
query_order = 'ORDER BY Uploader_profile.Monthly_tracking.' + cat + ' DESC'
query_filter = 'WHERE Locations + str(loc)
query = User.gql(query_filter + ' ' + query_order)
ranking_data = query.fetch(n_positions)
```

^a<https://cloud.google.com/appengine/docs/python/datastore/gqlreference>

Así pues existen 16 **combinaciones** de tipos de rankings para el caso de la consulta del jugador. Para el caso de la tarea periódica que los genera se reducirían a 6, ya que la mitad correspondiente a la información global no son considerados y además se invalidan las dos combinaciones que incluyen tanto puntos de experiencia como categoría (porque no tiene sentido). Sin embargo hay que tener además en cuenta el número de localizaciones asignadas al jugador y el número total de categorías existentes en el juego para lograr todas las combinaciones. Como ejemplo ilustrativo, suponiendo que existen en total 200 localizaciones registradas de todos los jugadores y existen 46 categorías, en la tabla 5.4 se visualizan las combinaciones válidas y puede verse que el total de rankings generados estaría cercano a los 10000.

Momento	Criterio	Localización	Categoría	Total
Mes	XP	L	C	No válida
Mes	XP	L	-	200
Mes	XP	-	C	No válida
Mes	XP	-	-	1
Mes	Q	L	C	9200
Mes	Q	L	-	200
Mes	Q	-	C	46
Mes	Q	-	-	1

Cuadro 5.4: Combinaciones válidas de rankings en base a filtros

5.10.2 Construcción de rankings

El primer paso para poder constituir los rankings consiste en poner atención a qué datos de los jugadores deben poder ser recuperados para, en base a ellos, poder generarlos. Referente a este aspecto hay que considerar la particularidad de estar empleando una base de datos no relacional. Esta cuestión se detalla en la sección 5.12 pero en resumen supone sacrificar la versatilidad de las consultas SQL que pueden construirse y ejecutarse contra una base de datos relacional a cambio de una ganancia considerable en escalabilidad. Este hecho resulta condicionante a la hora de proyectar el diseño de los modelos en base a los cuales

se almacena la información, teniendo como consecuencia directa la necesidad de mantener información redundante en la base de datos.

Teniendo presente lo anterior, la consulta que permite constituir un cierto ranking está basada en todos los casos en una ordenación en base a una propiedad específica, agregando a esta un filtrado en base a la localización en caso de ser necesario. La mayor parte de estas propiedades actúan como variables acumuladoras. Se especifica a continuación para cada tipo de ranking qué propiedad se requeriría:

- Basado en XP (para una localización o no) desde siempre. Basta con recuperar del perfil del jugador el total de puntos de experiencia que ha conseguido.
- Basado en XP (para una localización o no) en el último mes. Es necesario conocer el total de puntos de experiencia conseguidos pero en el último mes. Dado que los puntos de experiencia se consiguen por pregunta validada y por trofeo, habría que comprobar para cada jugador evaluado cuántas preguntas se le validaron en un periodo de tiempo concreto y qué trofeos consiguió, calculando la puntuación asociada al sumatorio de todos los méritos. En lugar de eso, para disponer del dato directamente se tiene una propiedad donde se van acumulando los puntos de experiencia conseguidos durante el último mes, que al inicio de un nuevo mes es reiniciada.
- Basado en número de preguntas (para una localización o no) desde siempre. Basta con mirar en el perfil del jugador cuántas preguntas se le han validado en total, lo que se tiene directamente.
- Basado en número de preguntas (para una localización o no) en el último mes. Para cada jugador habría que discriminar qué preguntas se le validaron durante un periodo de tiempo concreto. Para disponer del dato directamente se requiere un acumulador donde se vayan contabilizando las preguntas validadas y que se pondría a cero cada vez que empieza un nuevo mes.
- Basado en número de preguntas (para una localización o no) desde siempre para una categoría. En este caso para cada jugador habría que recuperar todas las preguntas que se le han validado dentro de una determinada categoría. En lugar de buscar esto para cada jugador se crea una propiedad acumuladora por cada categoría donde se van contabilizando las preguntas aprobadas. Estos acumuladores al ser globales nunca se resetean.
- Basado en número de preguntas (para una localización o no) en el último mes para una categoría. Sería necesario recuperar por cada jugador, todas las preguntas registradas y validadas dentro de un período de tiempo y para una categoría concreta. En lugar de realizar esa búsqueda de preguntas se puede incorporar una propiedad acumuladora por categoría donde ir registrando cada pregunta aprobada durante el mes.

La idea es que sin incorporar las propiedades acumuladoras la información ya está realmente incorporada en la base de datos. Sin embargo está disponible a través de consultas principalmente al tipo **Question** que dispone tanto del mapeo pregunta-jugador como de la fecha de validación y detalles de la pregunta como la categoría. Sin embargo para construir los rankings lo que interesa es realizar una consulta que devuelva jugadores, y no el conteo de preguntas. Lo que es más, se necesita incorporar el filtro basado en localización, que se corresponde al tipo de entidad del usuario del jugador. Si la consulta se hiciese sobre el tipo *Question* no se podría aplicar ese filtro. La rigidez para realizar consultas que involucran varios tipos de entidades supone un factor limitante, por el cual se necesita recurrir finalmente a esas propiedades acumuladoras accesibles a través de la entidad *User*.

Así pues, la solución pasa por referenciar desde el perfil del jugador dos modelos anidados que permitan la asignación dinámica de propiedades en tiempo de ejecución (no requieren definición previa). Estos modelos son **GlobalTracking** para los acumuladores totales y **MonthlyTracking** para los acumuladores mensuales. Tal como se ha mencionado anteriormente en cada uno de ellos se van registrando a medida que sea necesario propiedades nombradas con el nombre de la categoría asignada a la pregunta. En función del tipo de interacción del jugador estos modelos crecerán más o menos y ocuparán más espacio (si aportan preguntas de múltiples y diferentes categorías o se centran en un subconjunto pequeño). Una vez repartidos los trofeos correspondientes a los jugadores bien posicionados en cada uno de los rankings generados a final de mes todo el contenido de la entidad de tipo *MonthlyTracking* se elimina.

Flujo para la consulta del jugador

Cuando un jugador accede al área de rankings **visualiza todos los posibles filtros** que puede combinar para generar un ranking. Para cada cambio introducido en los selectores se envía una petición al servidor, que genera el ranking específico. Tanto para el caso en que sea el jugador el que desencadene el procedimiento como para el caso en que parta de una tarea periódica configurada en el servidor, el mecanismo empleado es similar. La única diferencia es que el jugador tiene más filtros habilitados (los que tienen un carácter temporal global) y que a este además de devolverse las primeras posiciones del ranking, **se le muestra su posición en el ranking** si no está incluido en esas posiciones destacadas. Para este último rasgo sin embargo se impone un límite para cortar la búsqueda en caso de que el jugador esté a mucha profundidad y existan muchos jugadores, lo que significa que rebasada la posición 1000 únicamente se especifica que está por detrás de la misma.

Al igual que ocurría con los trofeos, dada la información en bruto se aplica un **formateo a la lista de jugadores** destacados de forma previa a la devolución al cliente, con la finalidad de representar de forma compacta toda la información y que este listado se pueda renderizar directamente una vez recibido en el extremo cliente. Así, se genera una lista de listas, donde

cada lista interna incorpora en un orden definido previamente toda la información relevante que se debe devolver de cada jugador, a saber: nombre, nivel, rango y número de preguntas o puntos de experiencia gracias a los cuales están en esa posición.

Flujo para la generación de rankings mensuales

Disponiendo ya de una función que en base a unas etiquetas y parámetros reconstruya el ranking de interés, el procedimiento para la construcción de rankings al final del mes es sencillo. Basta con configurar una tarea periódica en el servidor que:

- Genere todas las posibles combinaciones de tipos de rankings en base a los filtros, retirando de la combinación aquellas no válidas.
- Para cada combinación se generen todas las posibles combinaciones de parámetros y se construya el ranking, lo que deriva en la obtención de una lista de entidades de usuarios.
- Se toma cada usuario y se le notifica a su perfil que se revise si se ha conseguido gracias a esta circunstancia un nuevo trofeo.

En cuanto a la configuración de tareas periódicas, App Engine ofrece soporte a través de las **tareas CRON**. La forma de definirlo pasa por incorporar en el servidor un servicio invocable únicamente por el propio servidor. El periodo de invocación se establece mediante un archivo específico de configuración denominado *cron.yaml*.

Modelo optimizado de construcción de rankings con AppEngine De cara a implementar mejoras a QuPI-GAM se localizó un artículo donde se describe el modelo desarrollado por Tomoaki Suzuki^a para la constitución y consulta de rankings escalables con AppEngine, que busca reducir la complejidad de $O(n)$ a $O(\log n)$. Aunque no se llegaron a volcar estas ideas en QuPI-GAM conviene guardar y referenciar y esta fuente de información por su interés y utilidad

^a<https://cloud.google.com/datastore/docs/articles/fast-and-reliable-ranking-in-datastore/>

5.11 Organización y componentes del proyecto

Se muestran a continuación una serie de cuadros y diagramas que permiten entender globalmente la estructura del proyecto, así como la interacción entre las distintas partes. Este análisis está un tanto más centrado en la descripción de los elementos de la parte cliente, ya que a la hora de exponer las funcionalidades a lo largo del capítulo se ha puesto más énfasis en la implementación del código del servidor.

Estructura general del proyecto

En los cuadros 5.5 y 5.6 se recoge la jeraquía de directorios correspondientes a la parte de frontend y backend respectivamente, junto con una descripción breve de su finalidad. Recordar que, como se describe en la sección 3.3.2, estos conceptos se emplean en desarrollo web para aludir a la parte de código alojado en el cliente encargada del aspecto visual y la interacción con el usuario (frontend) y a la gestión de la lógica y el procesamiento de los datos en la parte del servidor (backend).

Estructura del proyecto				
	/			
FRONTEND	html/	Contenedor de las plantillas que serán cargadas por el navegador web		
	static/	images/	Contenedor de imágenes mostradas en la aplicación	
		stylesheets/	Contenedor de hojas de estilo	
	scripts/	views/	Archivos que contienen la definición de las vistas de la aplicación web	
		models/	Archivos que contienen la definición de los modelos de la aplicación web	
		collections/	Archivos que contienen la definición de las colecciones de la aplicación web	

Cuadro 5.5: Vista de la organización del proyecto según jerarquía de directorios (Frontend)

Una vez expuesta una vista general a través de la jerarquía de código empleada, se muestra en la figura 5.19 un **diagrama de despliegue** con los componentes principales del sistema, para comprender de qué forma se distribuye el hardware necesario (dependiente de la infraestructura ofrecida por GAE) y la filosofía básica bajo la cual se organiza el backend.

Como puede observarse la parte de código asociada al backend es ejecutada en distintas instancias de GAE alojadas en servidores dedicados a tal fin. En esta parte, el **diseño en tres capas** es marcado mediante la definición de *handlers* y *managers*. Los *handlers* constituyen la interfaz a través de la cual se exponen los recursos y se manejan las peticiones recibidas. Estas se encargan de llamar a la lógica requerida para construir la respuesta. Para acceder a las entidades alojadas en el Datastore se recurre a los servicios de los *managers*, entendidas como clases que responden al patrón Singleton y se encargan de la interacción con la capa de persistencia. Asimismo, existen de forma separada servidores encargados de alojar los recursos estáticos de la aplicación, que están optimizados internamente para servir de forma eficiente las peticiones que solicitan este tipo de recursos.

Estructura del proyecto			
BACKEND	/		
	app.yaml	Archivo de configuración general. Se especifican aspectos como la versión de las librerías o el mapeo de código para el despachado de peticiones	
	index.yaml	Contiene la lista de índices generados para cada tipo de consulta	
	cron.yaml	Archivo de configuración. Se ajustan parámetros relativos a las tareas que son ejecutadas de forma periódica en el servidor	
	constants.py	Recopilación de valores constantes. Entre ellos: claves privadas, mensajes de información o error devueltos al usuario, etc	
	settings.py	Recopilación de valores ajustables. Permiten determinar la configuración de la lógica de la aplicación	
	messages.py	Recopilación de etiquetas. Asociadas a los campos de los JSON enviados y devueltos para cada petición/respuesta generada y a nombres de cookies	
	utils.py	Recopilación de utilidades genéricas que pueden ser requeridas desde distintas partes del código.	
	errors.py	Definición de errores que pueden ser generados durante la ejecución	
	handlers/	router.py	Archivo donde se especifican las clases que contienen los manejadores que han de gestionar cada petición
		api/	Contiene el conjunto de manejadores que atienden las peticiones emitidas por el cliente
		task/	Contiene el conjunto de manejadores que gestionan la ejecución de las tareas periódicas lanzadas por el servidor
	models/		Contiene la definición de cada tipo de entidad almacenada en el Datastore
	backend/		Contiene las clases que acceden al Datastore para gestionar las entidades
	docs/		Contiene las clases que gestionan los documentos del API de búsqueda
	requests/		Contiene las clases encargadas de construir peticiones emitidas por el servidor de la aplicación hacia otros servidores de terceros
	game_info/		Contiene archivos con información estructurada sobre características del juego que ha de ser cargada en el Datastore
	content/		Contiene el código encargado de poblar el Datastore generando datos de prueba y datos de base característicos del juego
data/		Este directorio almacena la imagen del Datastore y los documentos generados cuando se ejecuta la aplicación en local.	

Cuadro 5.6: Vista de la organización del proyecto según jerarquía de directorios (Backend)

Organización del frontend

Existen tres elementos clave en torno a los cuales se organiza la información de la cual se responsabiliza la parte cliente: plantillas, modelos y vistas y por último la gestión de las peticiones AJAX. La estructura de la interfaz y su contenido se define a través de una serie de **plantillas** que contienen código HTML y sintaxis que permite su renderizado junto

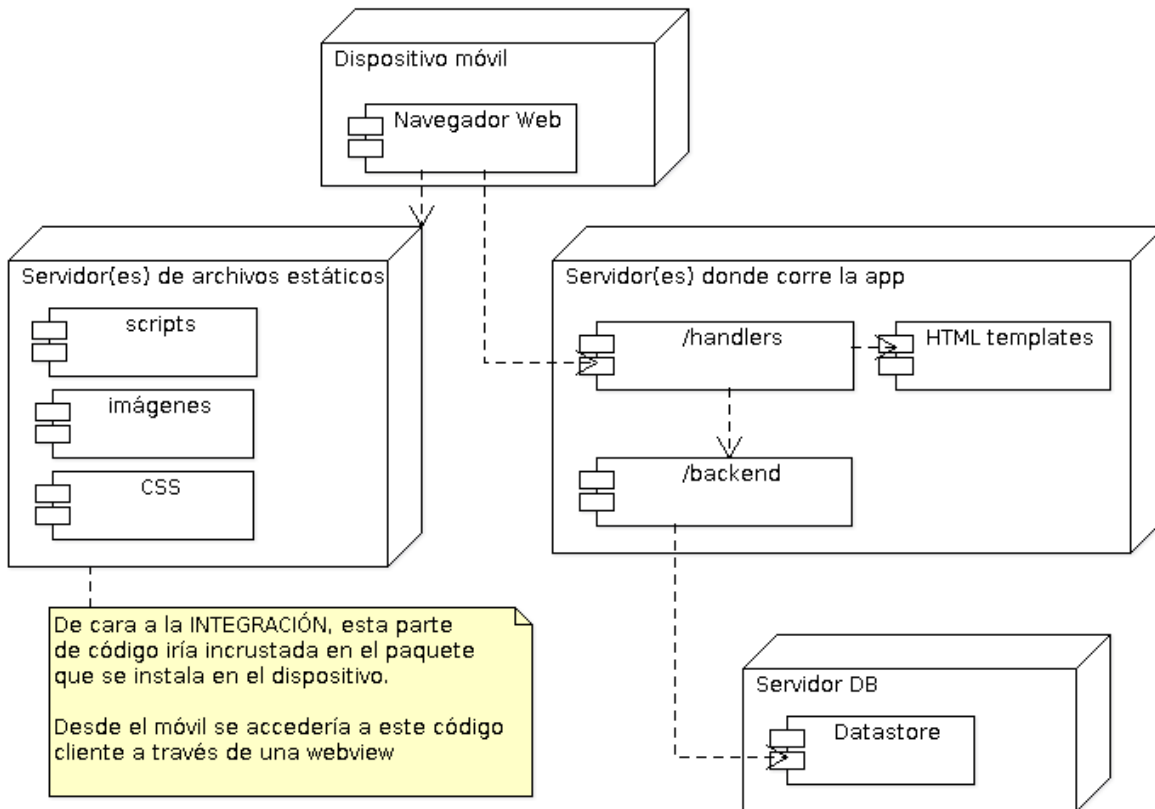


Figura 5.19: Diagrama de componentes y despliegue

con información incrustada. En la figura 5.20 se incluye la relación de plantillas empleadas, donde puede observarse la relación de inclusión entre ellas, ya que algunas de ellas actúan de marco donde cargar a su vez otras plantillas.

Para organizar el código de la parte cliente se han empleado vistas y modelos. Sobre las **vistas** recae la responsabilidad de gestionar la parte relativa a la interacción y el aspecto de la interfaz. En la figura 5.21 puede observarse un diagrama de clases con el conjunto de vistas definidas y sus relaciones. Las asociaciones incluidas en el diagrama reflejan cómo existen una serie de vistas de base que a su vez incorporan y cargan opcionalmente otras subvistas en función de la interacción que haga el usuario. A modo de referencia y si se quiere mayor detalle se incluye la versión completa de este diagrama en la documentación anexa entregada junto con este documento, donde puede verse lo siguiente:

- Cada vista está vinculada a una plantilla que se encarga de renderizar. La variable *template* establece esta vinculación.
- Por su parte cada vista ha de interactuar con una serie de modelos que almacenan la información vinculada a la interfaz. Los modelos a lo que puede afectar se incluyen como variables de la vista, incluyendo la variable *strings* el modelo que almacena las cadenas que han de cargarse junto con la plantilla según el idioma.

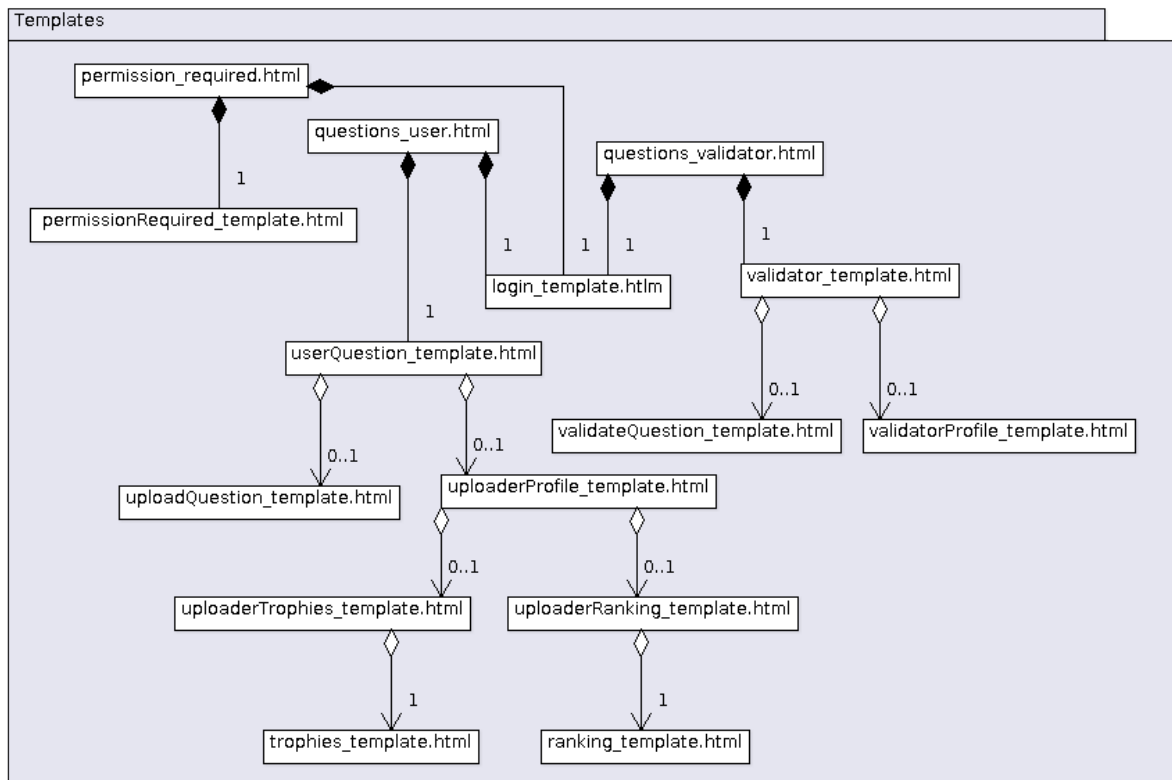


Figura 5.20: Diagrama de clases con las plantillas html manejadas

- Todas las vistas implementan una serie de funciones de base para el renderizado y la configuración inicial que solo han sido incluidas en la vista *BaseView* por simplicidad.
- Se ha colocado el estereotipo «request» para delimitar qué funciones son aquellas desde las cuales se emiten peticiones AJAX al servidor.

Los **modelos** son un equivalente a unidades que permiten manejar los datos asociados a la lógica de negocio. A tal efecto, se incluyen en la figura 5.22 el detalle de los modelos definidos. Puede observarse que existe un subpaquete con modelos destinados a alojar las distintas cadenas que han de mostrarse en cada interfaz en función del idioma definido. Asimismo, cabe destacar la existencia de dos **colecciones** para el manejo de trofeos y rankings y la validación a través de la función *validate* que se aplica para garantizar unos parámetros mínimos de correctitud sobre los modelos que han de alojar las preguntas aportadas por el usuario o bien revisadas por el validador. Para cada modelo se especifican sus atributos colocando el tipo de dato a la derecha de los dos puntos. Cuando se especifica como dato JSON, implica que el atributo almacena un diccionario de datos.

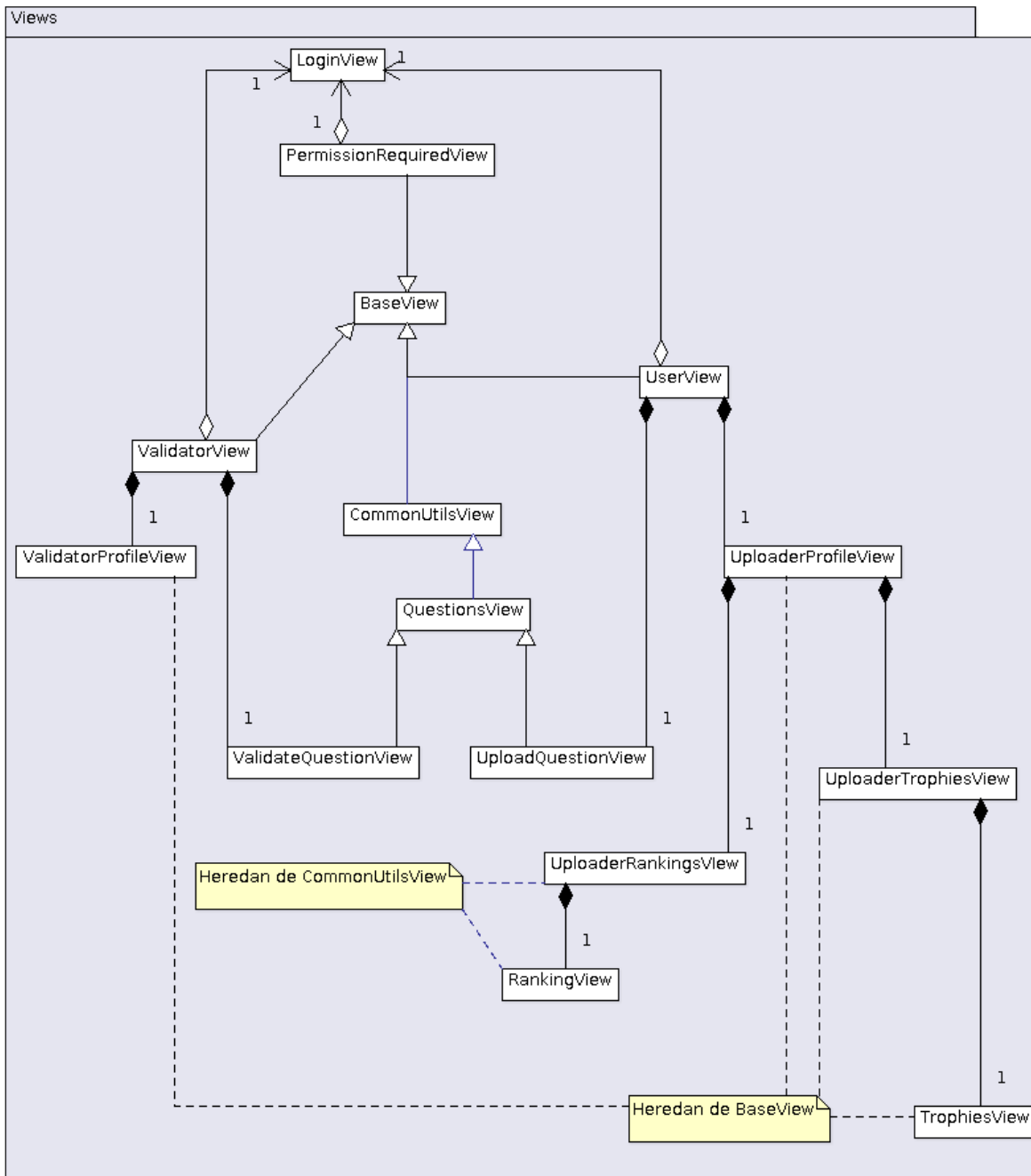


Figura 5.21: Diagrama de clases con las vistas manejadas en la parte cliente

Por último, como se puede ver en el diagrama que recopila las vistas, estas emiten una serie de peticiones a **recursos alojados en el servidor**. En la figura 5.23 puede consultarse el conjunto de recursos disponibles en el servidor (tanto aquellos públicos como aquellos ejecutados periódicamente por el propio servidor) identificados con la ruta mediante la cual son solicitados. Se especifican los parámetros de entrada (petición) y salida (respuesta) para cada uno de ellos. Aquellos sombreados son los que sirven las plantillas que definen la interfaz. Al margen de los parámetros incluidos, todas las respuestas incorporan en su *payload* un indicador de éxito o fracaso y opcionalmente un mensaje informativo.

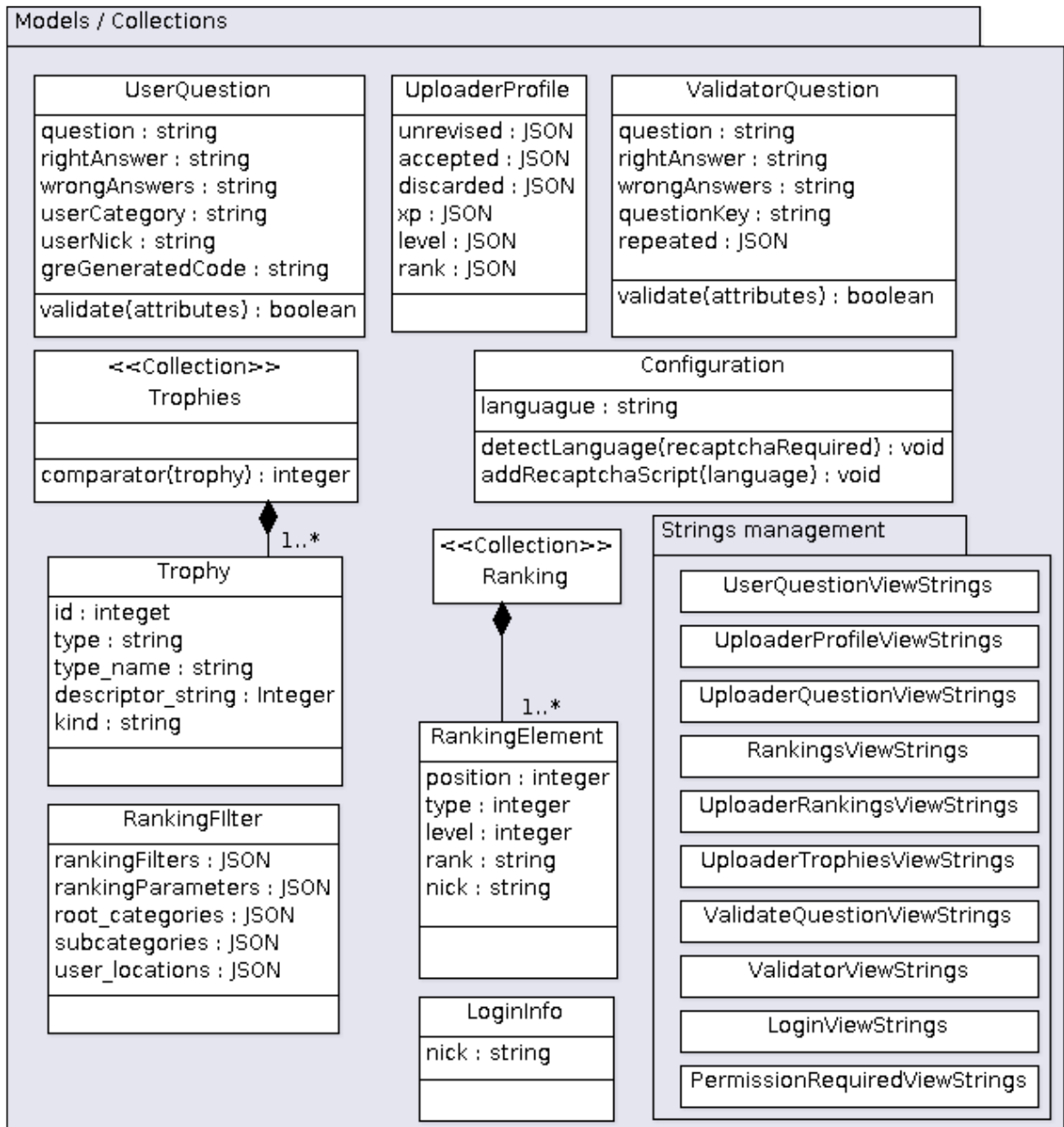


Figura 5.22: Diagrama de clases con los modelos en la parte cliente

5.12 Estructura de la base de datos

Al emplear GAE se dispone de una serie opciones para el almacenamiento de datos: Cloud Storage (para el almacenamiento de BLOBs), Cloud SQL (para sistemas OLTP), BigQuery (para sistemas OLAP) o Cloud BigTable (para información poco estructurada) además de **Google Datastore**, la opción elegida en este trabajo.

Contextualización de la base de datos: Datastore

Datastore no responde al enfoque tradicional relacional sino que es una base de datos NoSQL jerárquica orientada a objetos. Si se contrasta con una base de datos relacional:

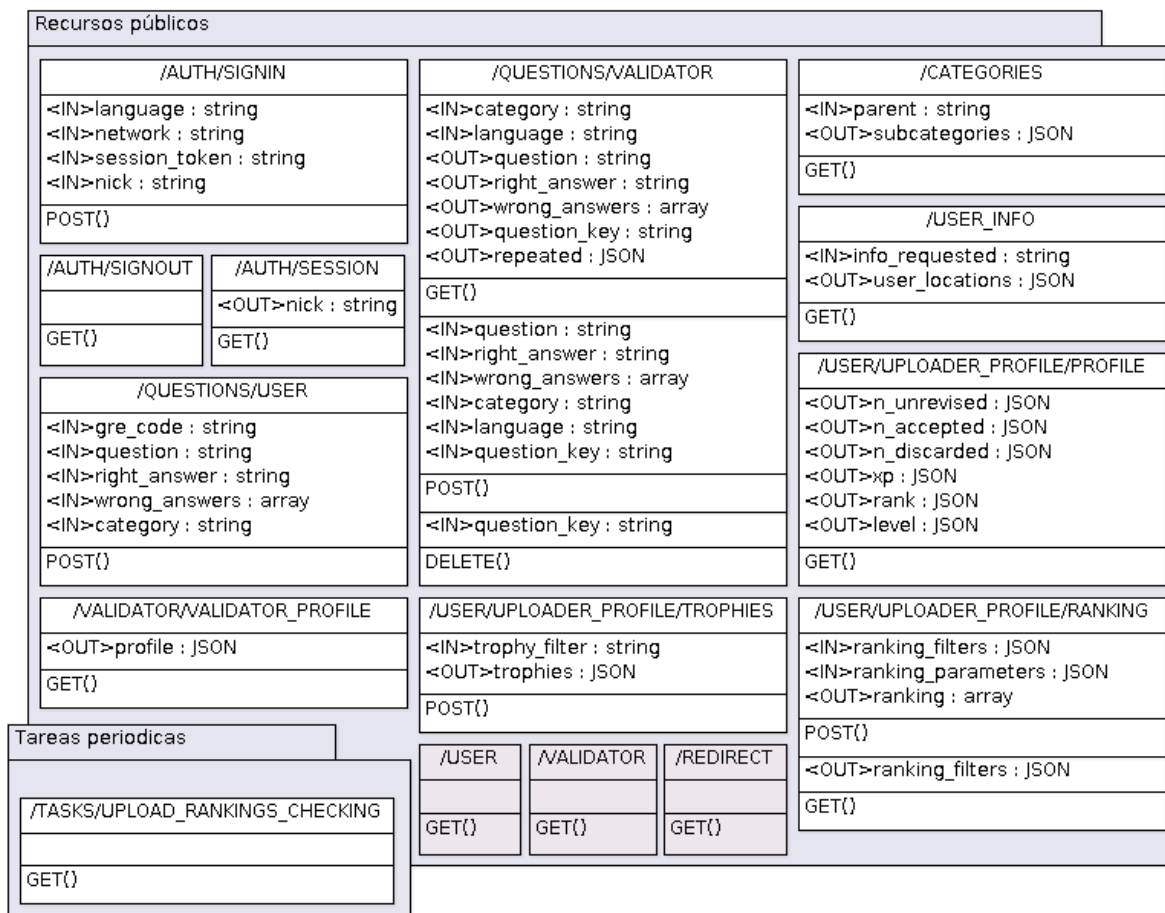


Figura 5.23: Conjunto de recursos expuestos en el servidor

- Se manejan **tipos** (equivalente a relaciones), **entidades** (equivalente a los registros) y **propiedades** (equivalente a los campos).
- Cada entidad creada dispone de una **clave** que constituye el equivalente a la clave primaria definida para cada registro y cada entidad de cada tipo puede aludir a otras entidades del mismo u otro tipo a través de una propiedad especial que permite establecer la jerarquía (lo que sería la clave ajena).
- A la hora de diseñar el modelo de datos no es necesario establecer **esquemas tan estrictos** como en el modelo relacional. Esto significa por ejemplo que no todas las entidades de un tipo tienen por qué contar con las mismas propiedades e incluso pueden definirse nuevas propiedades en tiempo de ejecución.
- Las bases de datos SQL soportan un **conjunto de datos** simple. En Datastore se tiene una mayor flexibilidad pudiéndose introducir objetos JSON o listas como valor de una propiedad. Dada su orientación a objetos se puede leer una entidad de la base de datos e instanciarla de forma cómoda y directa a un objeto a través del cual pueden accederse a sus propiedades.

- El conjunto de **consultas** que pueden ejecutarse sobre la base de datos es más restringido, ya que estas se apoyan en índices preconstruidos. Operaciones como la unión (*join*) o el filtrado de acuerdo a una subconsulta no están permitidas.

Al elegirse Datastore se pone énfasis en la capacidad de mantener un buen rendimiento cuando crece el número de datos y usuarios, ya que una de sus principales ventajas es su **capacidad para escalar** tanto para escrituras como para lecturas. En caso de las escrituras los datos son distribuidos. En el caso de las lecturas, la restricción en las consultas soportadas y el sistema de índices preconstruidos permite que una búsqueda en cientos de entidades sea igual de rápida que entre millones. Permite además ejecutar transacciones atómicas y se garantiza una alta disponibilidad y consistencia fuerte según el tipo de consulta. Por otro lado se integra muy bien con lenguajes de **scripting** y basados en objetos como Python.

Diseño redundante. La característica de escalabilidad se obtiene sacrificando las operaciones disponibles para las consultas, siendo este el **punto débil** frente al modelo relacional. Para sobrellevar este aspecto puede ser necesario afrontar un diseño donde existan **datos redundantes**. Esto no implica que existan dos entidades idénticas replicadas, sino que ciertos datos pueden estar contenidos a la vez en distintas propiedades de distintos tipos de entidades. Esta redundancia que no es deseable en el modelo relacional, se asume necesaria para el caso de las bases de datos NoSQL. No en vano, dada su buena disposición al manejo de grandes volúmenes de datos, hace que este no tenga por qué ser un factor limitante y justifica un diseño donde se incorpore cierta redundancia.

Modelos definidos

Dado que los tipos que se almacenarán en el Datastore se definen creando clases que heredan de *ndb.Model* es factible recoger la estructura definida en la base de datos a modo de diagrama de clases. En este, que puede consultarse en la figura 5.24, cada clase representa un tipo o modelo. Lo que en el modelo relacional serían claves ajenas se representan aquí mediante los tipos *Key* y *Structured*, plasmándose mediante agregaciones y composiciones respectivamente las relaciones entre los modelos. Los modelos apuntados por una propiedad de tipo *Key* tienen una existencia independiente. Aquellos apuntados por la propiedad de tipo *Structured* tienen una existencia dependiente del modelo que lo incluye.

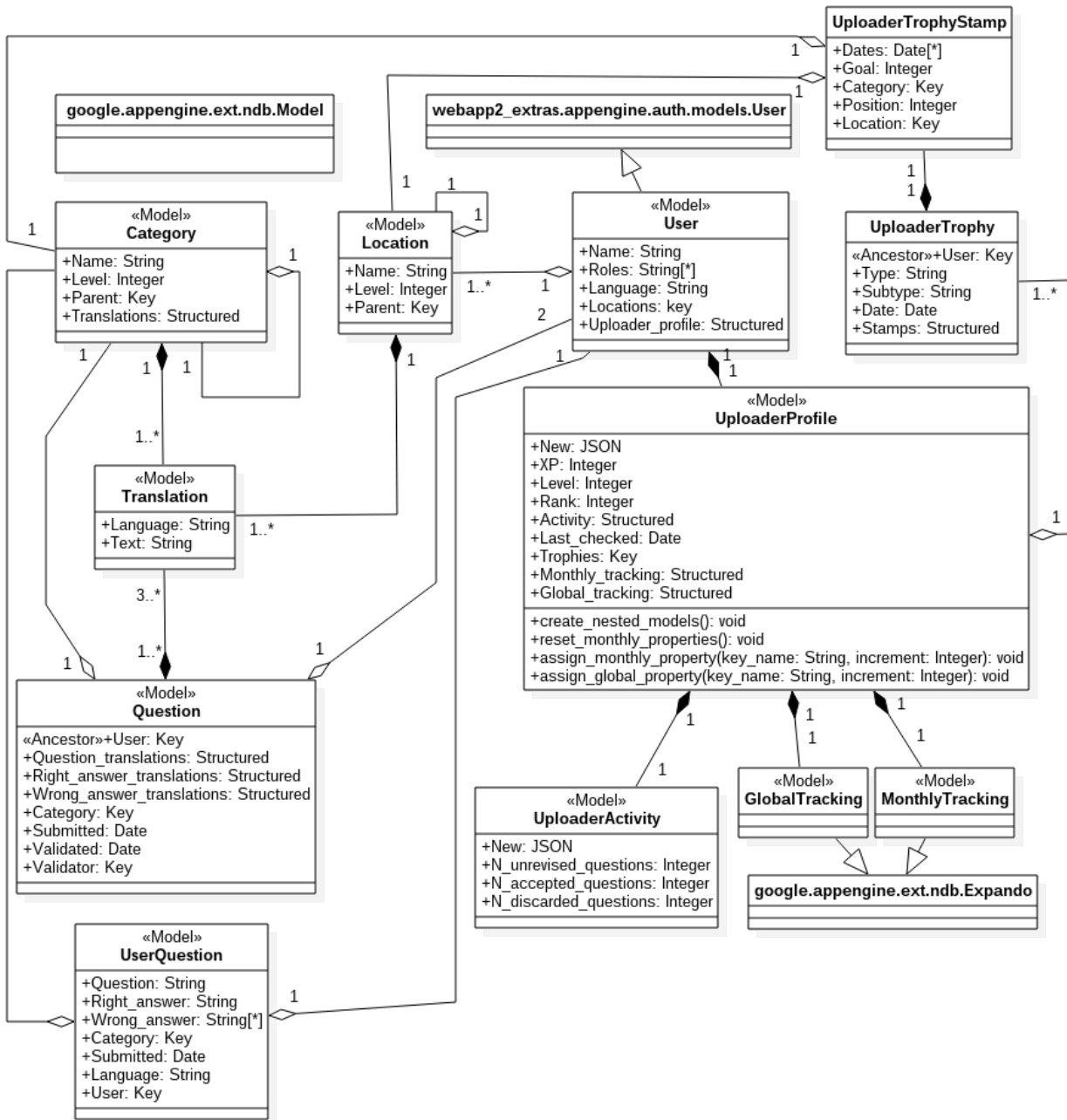


Figura 5.24: Modelado de entidades definidas en el Datastore

5.13 Historias de Usuario

En este apartado se incluyen al detalle el conjunto de historias de usuario definidas.

Identificador	US_SUPPORT_01
Rol	Como... colaborador
Funcionalidad	Quiero... poder iniciar sesión en el sistema
Utilidad	Para... estar identificado y que quede registrada toda mi actividad
Aceptación	Debe permitirse la autenticación a través de una cuenta de Google +
	Debe permitirse la autenticación a través de una cuenta de Facebook
	No debe existir un mecanismo explícito de registro de nuevo usuario . Este no tendrá que rellenar ningún formulario.

Cuadro 5.7: Historia de Usuario - US_SUPPORT_01

Identificador	US_SUPPORT_02
Rol	Como... colaborador
Funcionalidad	Quiero... poder visualizar los textos de la interfaz en el idioma apropiado
Utilidad	Para... que la experiencia de uso resulte lo más cómoda y familiar posible
Aceptación	Se incluirá soporte para el idioma español e inglés
	El colaborador no debe introducir explícitamente el idioma deseado. Este será detectado automáticamente desde el dispositivo, seleccionándose de forma co-rrelativa ese idioma para la aplicación
	Si no se tiene soporte para el idioma detectado en el dispositivo, el idioma establecido por defecto será el inglés

Cuadro 5.8: Historia de Usuario - US_SUPPORT_02

Identificador	US_UPLOAD_01
Rol	Como... jugador aportador
Funcionalidad	Quiero... poder proponer una pregunta
Utilidad	Para... poder colaborar y que esta pueda pasar a formar parte del conjunto de preguntas de las que se nutre el juego
Aceptación	Se permite el acceso sin login, es decir, los jugadores anónimos también pueden proponer preguntas
	Se ha de incorporar un mecanismo para verificar que la pregunta no fue generada a través de bots
	Se ha de aplicar una validación mínima a la información aportada para evitar gastar recursos del servidor en contenido inútil. Si no se pasa la verificación se debe informar al jugador de las causas de la denegación

Cuadro 5.9: Historia de Usuario - US_UPLOAD_01

Identificador	US_VALIDATION_01
Rol	Como... validador
Funcionalidad	Quiero... poder recuperar preguntas pendientes de revisión y determinar de forma definitiva si se consideran o no válidas
Utilidad	Para... determinar qué preguntas de las sugeridas por los jugadores pueden pasar a formar parte del conjunto de las que se muestran en el juego
Aceptación	Es un área de acceso restringido. Solo un colaborador al que se le haya asignado el rol de validador debe poder acceder a esta zona
	Se debe poder recuperar una pregunta indicando como parámetros de recuperación su idioma y la categoría general a la que pertenece (de acuerdo esto al criterio del jugador)
	El validador debe tener la posibilidad de modificar cualquier campo del formulario, en caso de que considere que la pregunta es válida pero contiene algún error que ha de ser modificado
	Si se recupera una pregunta que genera dudas y el validador no se ve capaz de emitir un juicio, debe poder solicitar una nueva pregunta, quedando esa pendiente
	El validador debe poder descartar o validar definitivamente una pregunta
	Para cada pregunta recuperada, se debe informar al validador de si existe ya una pregunta validada en el sistema que puede ser de temática similar a la que se va a revisar. En ese caso el validador recibirá un aviso y podrá leer ambas preguntas y decidir si la recuperada aporta contenido nuevo o debe ser descartada
	Se ha de aplicar una validación mínima a la información aportada para evitar un posible error por parte del validador
	Varios validadores deben poder estar ejerciendo la labor de validación de forma simultánea sin que esto suponga conflicto

Cuadro 5.10: Historia de Usuario - US_VALIDATION_01

Identificador	US_VALIDATION_02
Rol	Como... validador
Funcionalidad	Quiero... poder determinar, dada un pregunta pendiente de revisión, si existe una pregunta ya validada que comparta la misma temática.
Utilidad	Para... poder determinar si vale la pena revisar y validar la pregunta recibida o es susceptible de ser descartada al contarse ya con una pregunta similar en el sistema
Aceptación	Cuando se devuelva una pregunta pendiente de revisión, se debe devolver además el enunciado de la pregunta ya validada que más se parezca a esa, en caso de detectarse una similitud
	En caso de posible coincidencia, al validador se le mostrará un aviso y el enunciado de la pregunta susceptible de estar repetida. Si este determina que ambas preguntas aluden al mismo concepto, tiene libertad para descartar la recibida

Cuadro 5.11: Historia de Usuario - US_VALIDATION_02

Identificador	US_UPLOAD_02
Rol	Como... jugador aportador
Funcionalidad	Quiero... poder recibir feedback acerca de mi actividad como aportador sugiriendo preguntas accediendo a mi perfil personal
Utilidad	Para... poder realizar un seguimiento de mi actividad, del estado de las preguntas subidas y comprobar que consigo a medida que se validan mis preguntas
Aceptación	El acceso a este área estará restringido a jugadores autenticados que hayan iniciado sesión
	Se debe poder consultar del total de preguntas aportadas cuántas han sido aprobadas y cuántas han sido descartadas
	Se incorporará un mecanismo que permita medir y cuantificar el progreso a través de un sistema de asignación de puntos de experiencia
	Este área debe ser el punto de acceso a cualquier otro mecanismo de gamificación que requiera proveer de <i>feedback</i> al jugador

Cuadro 5.12: Historia de Usuario - US_UPLOAD_02

Identificador	US_UPLOAD_02.1
Rol	Como... jugador aportador
Funcionalidad	Quiero... que exista una lista de logros alcanzables en función de la actividad y las acciones que se pueden llevar a cabo en el sistema
Utilidad	Para... poder
Aceptación	Cuando se acceda a la sección se deben listar todos los trofeos disponibles
	Se debe marcar en la lista de trofeos aquellos conseguidos por el jugador así como aquellos nuevos que no había conseguido en la última visita al perfil
	La obtención de un trofeo debe recomensarse asignando puntos al jugador, otorgándose distinta puntuación en función de la dificultad del trofeo
	Los trofeos deben mostrarse agrupados por tipo

Cuadro 5.13: Historia de Usuario - US_UPLOAD_02.1

Identificador	US_UPLOAD_02.2
Rol	Como... jugador aportador
Funcionalidad	Quiero... poder compararme con otros consultando rankings
Utilidad	Para... poder contextualizar y medir en qué medida impacta mi aportación al sistema tomando como referencia la actividad desarrollada por otros aportadores
Aceptación	Se debe poder aplicar un filtro para localizar los rankings de interés
	Deben poder construirse rankings en función de distintos criterios. Entre ellos hay que distinguir entre el número de preguntas aportadas y los puntos de experiencia conseguidos, discriminando en este punto opcionalmente por categoría, localización y por distintos periodos temporales
	Los rankings irán variando a lo largo del tiempo. De forma automatizada y periódica (a final de mes) el sistema debe generar todos los rankings definidos y recomendar a los jugadores localizados en las diez primeras posiciones. Estos rankings se constituyen de forma relativa al mes, no absoluta, lo que significa que se construyen en base a la actividad del jugador en el último mes
	Cualquier ranking puede ser generado en cualquier momento por el jugador

Cuadro 5.14: Historia de Usuario - US_UPLOAD_02.2

Resultados

UNA vez expuesta la problemática, los objetivos planteados, el método de trabajo y las principales decisiones de diseño adoptadas a la hora de plantear la solución, se exponen los resultados conseguidos en términos de esfuerzo efectivo dedicado (cómo se ha organizado el trabajo a lo largo de diferentes iteraciones) y del producto final al que se ha dado salida. En ese sentido hay que considerar QuPI-GAM como sistema independiente potencialmente integrable con el juego *Whizz*, por lo que se comenta asimismo el estado de preparación en que ha quedado el planteamiento para afrontar una posible integración.

6.1 Aspecto y funcionalidad de QuPI-GAM

Una vez desarrollado y probado el sistema en local, donde gracias al enfoque sandbox de App Engine se puede simular todo el entorno, es momento de llevar a cabo el **despliegue en los servidores** que Google pone a disposición. La capacidad y recursos asignados están limitados para las cuentas gratuitas, pudiéndose consultar los valores concretos en la página de precios de App Engine¹. Una vez realizado el despliegue, el sistema está disponible e interaccionando con el Datastore como base de datos bajo el dominio gratuito:

<https://questions-app-3.appspot.com/>

En este apartado se repasarán brevemente las funcionalidades principales con que cuenta el sistema de cara a los colaboradores (jugadores que aportan y validadores) acompañándolo de pantallazos que muestran el aspecto de QuPI-GAM. En cuanto a este último punto se ha tratado de proyectar un diseño adaptable y sencillo, agrupando la información de forma que sea fácilmente localizable, aportándole el *feedback* necesario al usuario de forma clara.

¹<https://cloud.google.com/appengine/pricing>

Share your own question SPANISH

CATEGORY

ART

QUESTION

¿De qué famosa saga de videojuegos es creador Hideo Kojima?

ANSWERS

- ✓ Metal Gear
- ✗ Splinter Cell
- ✗ Final Fantasy
- ✗ Final Fantasy

ERROR: Fields must not be repeated

✓ I'm not a robot reCAPTCHA

Send Delete

Figura 6.1: Interfaz para la aportación de preguntas

Aportar pregunta. A través de este formulario el usuario puede subir su pregunta (compro­bándose que no existen opciones repetidas y que ningún campo está vacío) una vez superada la captcha y definida la categoría a la que pertenece.

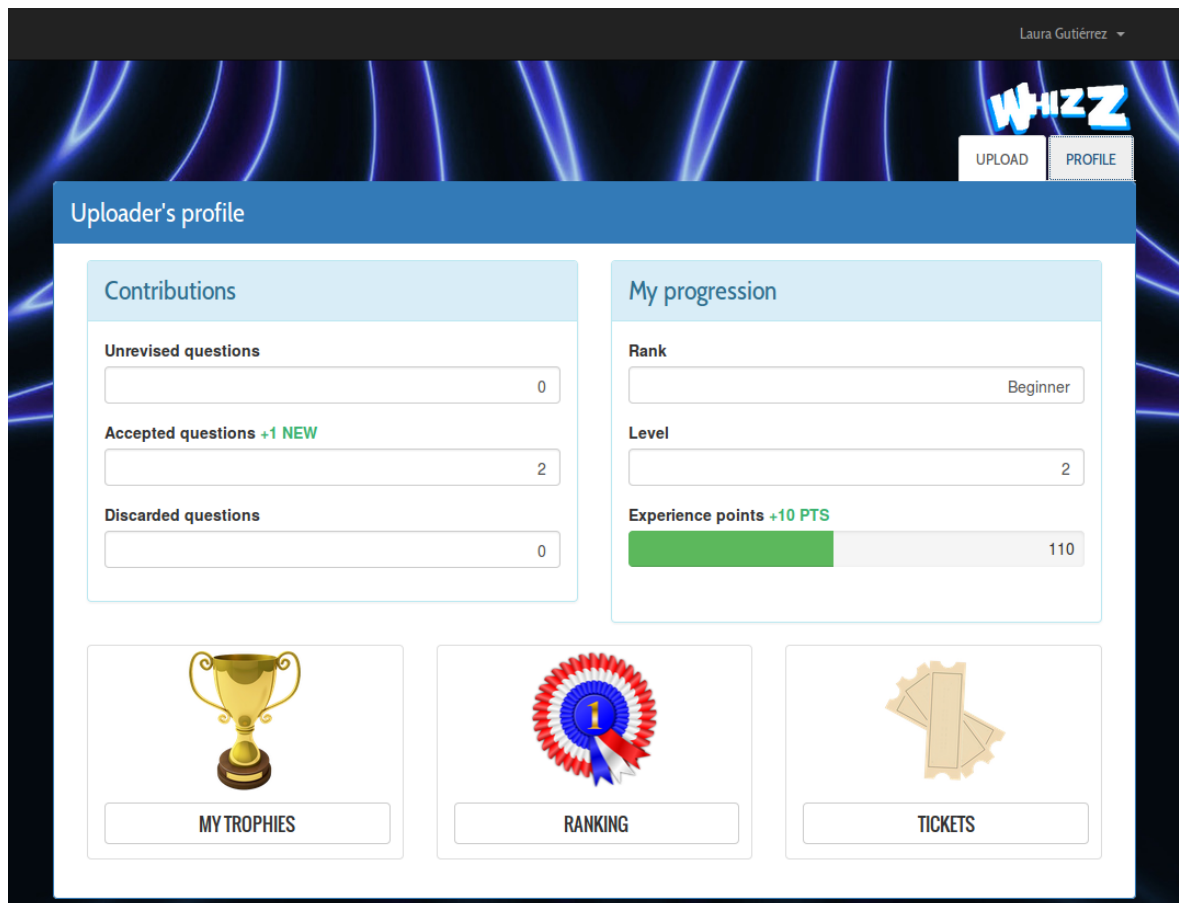


Figura 6.2: Interfaz que muestra el aspecto del perfil de colaborador

Visitar perfil. El usuario dispone de un perfil donde puede consultar su actividad, informándose de los avances conseguidos desde su última visita. Puede comprobar sus logros en términos de puntos de experiencia, nivel y rango adquiridos. Constituye un punto central a través del cual se puede acceder al área de rankings y a la galería de trofeos.

Laura Gutiérrez ▾

Wizz
 UPLOAD PROFILE

Rankings

Selection criteria

Type: Number of questions Experience points

Location: National Worldwide

Category: All Specific

Time frame: Last month Since ever

Ranking

#	Player	Level	Rank	-
1	PFGREIUHGA	18	Rank 7	493
2	MSFBOSWTXO	18	Rank 7	447
3	SUHLMPFAZQ	17	Rank 6	422
4	NZMTJBYVQZ	17	Rank 6	411
5	SBCNWEWTMY	17	Rank 6	409
6	UHIWLSFBCW	16	Rank 6	289
7	HPJXEZCOOL	16	Rank 6	265
8	UIDVMKAQZX	15	Rank 6	242
9	KLAMUQVLAC	14	Rank 6	122
10	PDNFXYTGUS	13	Rank 5	90
11	Laura Gutierrez	6	Rank 3	2
12	Lord	1	Rank 1	0

Figura 6.3: Interfaz para la consulta de la posición en los rankings

Comprobar rankings. Desde el perfil, el usuario puede acceder al área de ranking, donde puede consultar su posición en los diferentes rankings configurando una serie de filtros basados en localización, categoría de pregunta, periodo de tiempo y tipo de logro.

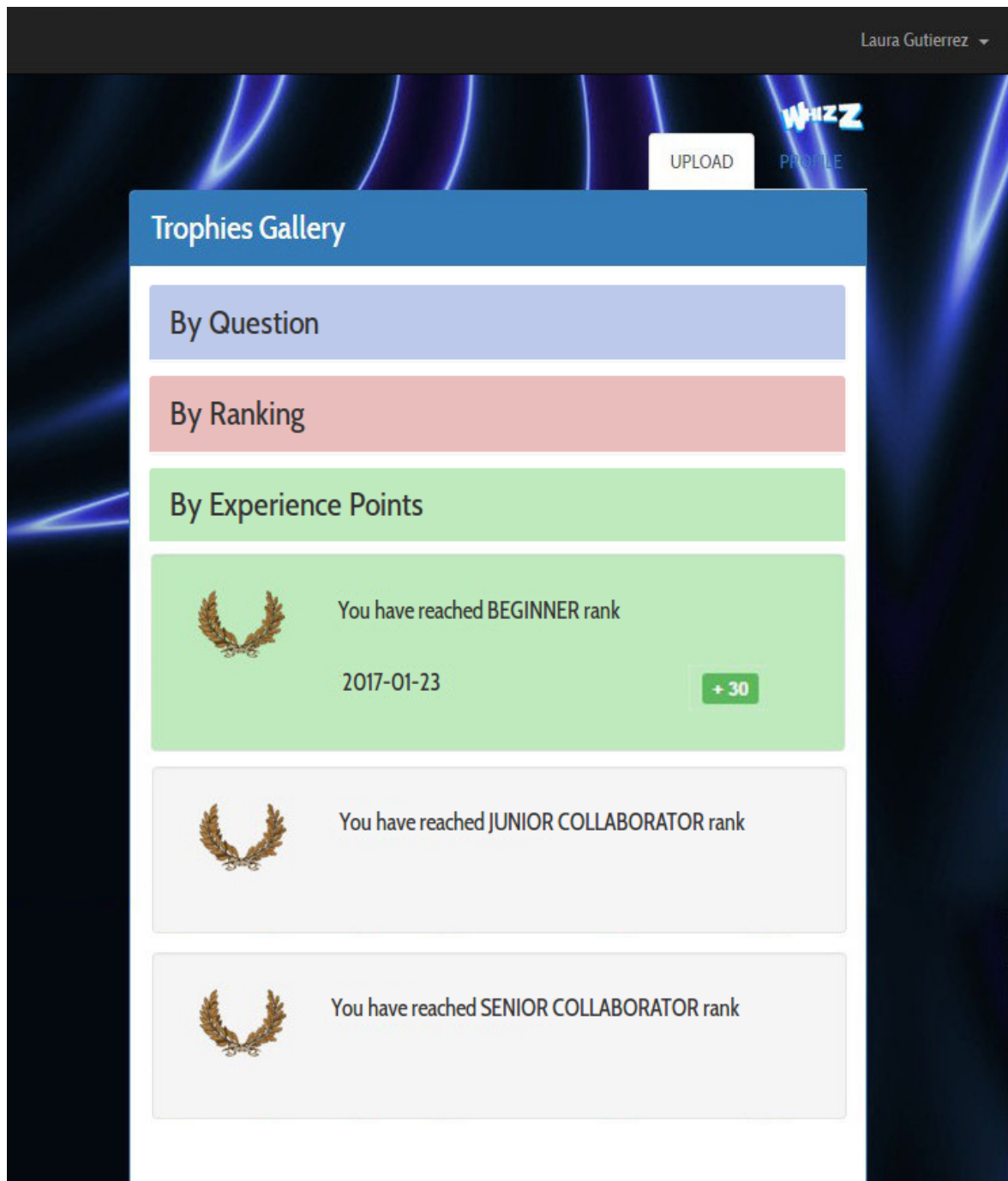


Figura 6.4: Interfaz para la consulta de trofeos

Revisar trofeos. Se proponen al usuario una serie de metas a conseguir a través de la noción de trofeo. En la galería puede consultar los trofeos existentes, así como aquellos que ya ha conseguido.

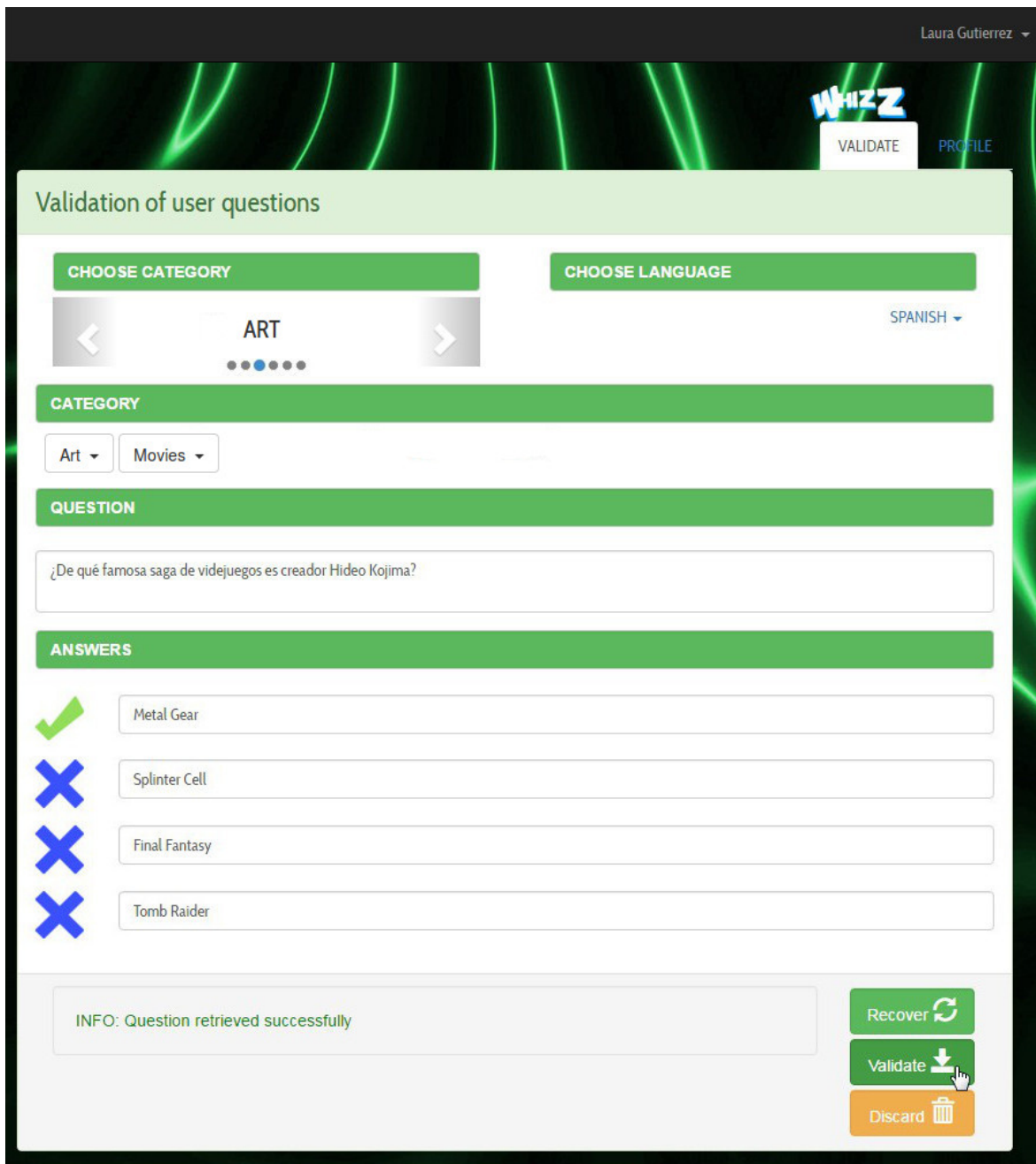


Figura 6.5: Interfaz para la validación de preguntas

Validar pregunta. Los colaboradores asignados pueden acceder al conjunto de preguntas pendientes de revisión. El validador recibe una pregunta en función de la categoría y el idioma establecidos. En este punto, puede optar por validar la pregunta una vez corregida y definida su subcategoría, descartarla o bien recuperar otra pregunta y posponer esta. En caso de que el sistema haya detectado que podría existir en la base de datos una pregunta de temática similar a la revisada se le emite un aviso.

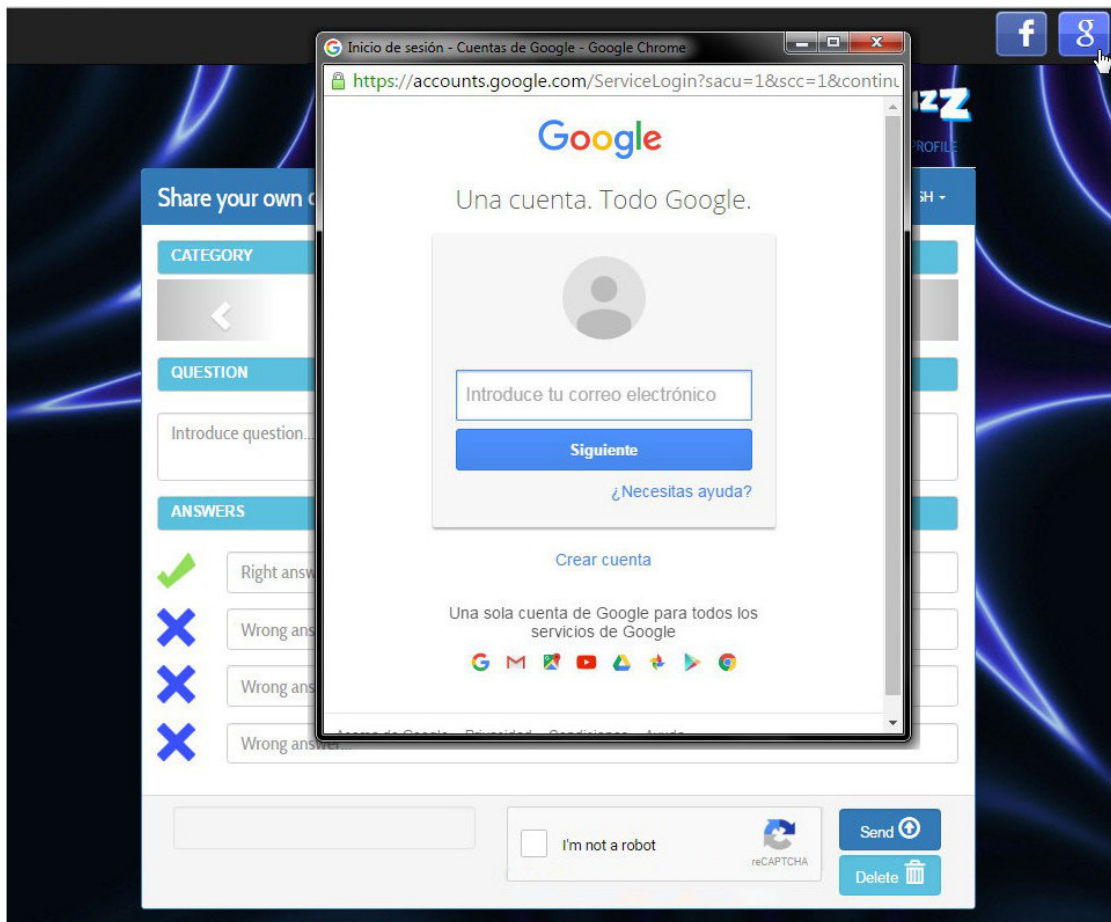


Figura 6.6: Interfaz para la autenticación e inicio de sesión

Iniciar sesión. El jugador tiene la opción (no obligatoria) de loguearse en el sistema a través de su cuenta de Google + o Facebook. El validador sí que debe iniciar sesión para estar identificado y poder tener acceso al área de validación.

6.2 Distribución de trabajo

En el cuadro presentado a continuación se describen las iteraciones en torno a las cuales se ha organizado el desarrollo. Al ser un trabajo afrontado de forma unipersonal, las iteraciones no se han solapado por lo que se han tratado de forma secuencial.

ITERACIONES DE DESARROLLO	
1# Familiarización con las tecnologías	
Historia de usuario	-
Tiempo asignado	2 semanas
<p>Se planteó como primera tarea la elaboración de un formulario sencillo en la parte del cliente a través del cual se pudiera introducir una pregunta con cuatro opciones de respuesta y que esta información se pudiera enviar y almacenar en el servidor.</p> <p>El objetivo principal era asumir una tarea sencilla que permitiese realizar una toma de contacto con las tecnologías que se emplean en la empresa. Esencialmente se centró el aprendizaje en el framework Bootstrap y los lenguajes HTML5, CSS3, Javascript y jQuery para la parte del código cliente. Del lado del servidor, se centró el estudio en el funcionamiento de la plataforma AppEngine, que permite alojar el código en los servidores de Google, así como en el funcionamiento del Cloud Datastore como base de datos orientada a objetos, empleando para todo ello Python como lenguaje de programación.</p> <p>También se consultaron las hojas de estilo seguidas en la empresa, así como la estructura bajo la cual se organizaba el código del juego en el servidor. Dado que se optó por un desarrollo independiente al juego, era importante comprender la estructura seguida para poder replicarla en el código generado de forma separada, con el fin de favorecer una futura integración.</p>	

2# Incorporación de mecanismo de autenticación	
Historia de usuario	US_SUPPORT_01
Tiempo asignado	2 semanas
<p>En esta iteración se incorporó al sistema un mecanismo que permitiera a los jugadores autenticarse e iniciar sesión en el sistema.</p> <p>Para ello, tras probar y descartar el servicio de autenticación integrado en App Engine, se incorporó la librería hello.js, que da soporte para la autenticación mediante terceros, en este caso, una cuenta de Google + o Facebook. La ventaja de este método es que no es necesario almacenar en el servidor las contraseñas de los jugadores, ya que esta parte es gestionada por los servicios previamente mencionados. A este lado es necesario únicamente validar la información de autenticación recibida en el servidor, para garantizar que esta ha sido emitida por una fuente confiable y no ha sido manipulada.</p>	

3# Constitución del formulario para subir preguntas	
Historia de usuario	US_UPLOAD_01
Tiempo asignado	2 semanas
<p>Se completó el formulario creado en #1 a través del cual el jugador puede aportar sus preguntas, incorporando la posibilidad de especificar información extra relativa a la pregunta y agregando un mecanismo de seguridad.</p> <p>Se agregó la posibilidad de que el usuario especifique el idioma en que sube la pregunta. Se incorporó además un carrusel de categorías base, para que el jugador pueda clasificar la pregunta que sube de acuerdo a su temática.</p> <p>Dado que no es obligatorio iniciar sesión para poder aportar una pregunta, el sistema podría ser vulnerable a ataques lanzados a través de bots que introdujesen información errónea. Como mecanismo de protección, se incorporó una recaptcha.</p>	

4# Incorporación de sesión en el servidor y refactorización de código	
Historia de usuario	US_SUPPORT_01
Tiempo asignado	2 semanas
<p>La tarea asignada consistió en mejorar el código del cliente en cuanto a su organización y el código del servidor en cuanto a funcionalidad.</p> <p>Hasta el momento se mantenía la información de la conexión a través de cookies. Fue necesario incorporar en el servidor el soporte a la sesión, como medio para poder almacenar y consultar de forma segura la información asociada a cada jugador que mantiene una sesión iniciada.</p> <p>En cuanto al código cliente, se estudió el framework Backbone y el sistema de plantillas Underscore, con el fin de llevar a cabo una refactorización que permitiera organizar el código de acuerdo al patrón MVC.</p>	

5# Creación del formulario de validación	
Historia de usuario	US_VALIDATION_01
Tiempo asignado	2 semanas
<p>Se creó un formulario accesible por el personal encargado de revisar las preguntas aportadas por los jugadores, con el correspondiente soporte en el servidor.</p> <p>A través de este formulario, el validador puede recuperar preguntas de una categoría y un idioma concreto. Tiene la posibilidad de corregir la redacción, así como reasignar la categoría y el idioma definido por el jugador. Puede validar la pregunta, en cuyo caso pasaría a formar parte de las preguntas que se deben mostrar en el juego o bien descartarla si no tiene la calidad suficiente. Se gestiona aquí la necesidad de controlar el acceso concurrente a una misma pregunta por parte de varios validadores.</p>	

6# Internacionalización	
Historia de usuario	US_SUPPORT_02
Tiempo asignado	1 semanas
<p>Se incorporó el soporte necesario para poder internacionalizar la aplicación. Esto permite que los textos presentes en la interfaz que visualiza el jugador se adapten al idioma detectado en su dispositivo (con soporte al español y al inglés).</p>	

7# Creación del perfil de colaborador	
Historia de usuario	US_UPLOAD_02
Tiempo asignado	1 semana
<p>Centrando el trabajo en la parte de gamificación, se determinó en este punto crear un perfil de colaborador, a través del cual el jugador pueda realizar un seguimiento de su actividad. Puede consultar el número de preguntas aportadas sin revisar, preguntas aceptadas y preguntas descartadas. También puede consultar la recompensa obtenida en términos de puntos de experiencia, nivel y rango adquiridos.</p>	

8# Definición e incorporación de trofeos	
Historia de usuario	US_UPLOAD_02.1
Tiempo asignado	2 semanas
<p>Con el fin de guiar y reconocer las acciones positivas que el jugador lleve a cabo en el sistema como colaborador, se plantea la incorporación de trofeos como metáfora para identificar los logros que vaya consiguiendo.</p> <p>Se definieron una serie de metas a conseguir por el jugador, de tal forma que pueda consultar los trofeos conseguidos (que le reportan puntos de experiencia) así como los pendientes. Esta mecánica tienen por objetivo incentivar al jugador, para que trate de mejorar como colaborador, aportando cada vez más preguntas.</p>	

9# Implementación de mecanismo para construir rankings mensuales	
Historia de usuario	US_UPLOAD_02.2
Tiempo asignado	2 semanas
<p>Se elaboró un sistema de construcción de rankings, donde el jugador puede combinar distintos filtros para consultar distintos tipos de clasificación. De esta forma puede contrastar cómo de significativa ha sido su aportación al sistema comparándose con el resto de la comunidad de colaboradores de acuerdo a la posición que ocupe en estos.</p> <p>Por otro lado se configuró una tarea periódica en el servidor dedicada a comprobar al final de cada mes qué jugadores se localizan en el top 10 de cada ranking para recompensarles con puntos de experiencia. Los rankings se elaboran de forma relativa, tomando como referencia la actividad generada por cada jugador únicamente durante ese último mes. Esto favorece que no se vayan creando diferencias insalvables entre jugadores ya que al inicio de cada mes todos parten de cero.</p>	

10# Detección de pregunta repetida	
Historia de usuario	US_VALIDATION_2
Tiempo asignado	2 semanas
<p>Al recopilarse información de diversas fuentes no coordinadas ésta pueda ser coincidente. Como no es deseable contar con esta redundancia, se estudiaron dos vías que permitiesen tratar de detectar cuándo una pregunta propuesta que va a pasar el proceso de revisión podría haber sido incorporada previamente al sistema. La opción finalmente incorporada pasó por emplear la librería <i>Search</i> incluida en el soporte de App Engine, que permite registrar cada pregunta bajo el concepto de documento y realizar búsquedas en todas ellas en base al texto que contienen.</p>	

11# Despliegue y primer contacto hacia una futura integración	
Historia de usuario	-
Tiempo asignado	2 semanas
<p>Llegados a este punto, se desplegó el sistema web en los servidores de Google sobre el dominio que gratuitamente se asigna a las cuentas de desarrollo y se realizaron pruebas para comprobar si se registraba el mismo comportamiento que en la simulación en local.</p> <p>Por último se proyectaron los pasos a dar para abordar una integración progresiva en <i>Whizz</i>. Aunque la experiencia integradora se llevó a cabo en una primera fase, esta no llegó a ser incorporada en la versión comercial del juego.</p>	

6.3 Integración de código

Como ya se ha referido anteriormente, todo el proyecto se ha realizado en colaboración y bajo la tutela del equipo de desarrollo de la empresa Furious Koalas. El grueso de la dedicación de este trabajo ha ido dedicado al desarrollo del sistema QuPI-GAm. La idea que da peso a la elaboración de este sistema es la de terminar integrándolo en su totalidad en el juego *Whizz*. Sin embargo en la última fase de desarrollo únicamente se ha optado por plantear el impacto que tendría dicha integración, definiéndose qué pasos habría que dar desde un enfoque incremental.

Integración como objetivo pendiente. Dejar el proyecto en este punto se justifica por el hecho de que llevar a cabo una integración sobre la versión comercial de *Whizz* tendría un impacto considerable en lo que se refiere a la coordinación con todo el equipo de trabajo, no asumible a día de hoy. Sin embargo esto no invalida la utilidad de este trabajo, en la medida que de acuerdo a la estrategia de la empresa la integración sí que está contemplada y el sistema desarrollado asienta las bases para poder disponer del sistema de aportación incluido en el juego en un futuro cercano.

En términos de integración, actualmente el concepto al que más se recurre en estos términos es el de **Integración Continua**. Este es un enfoque bajo el cual se pretende mejorar la calidad del software y que engloba una serie de prácticas que minimizan el riesgo de que un cambio, mejora o añadido en una parte del código tenga efectos colaterales indeseados que no se detecten en el resto de código. Esencialmente consiste en ir integrando cada cambio de forma periódica en un entorno similar al de producción, comprobando la integridad del conjunto generando de forma automatizada un *build* que lanza una batería de tests y genera los pertinentes informes.

Si bien la integración continua como tal no se ha empleado, ya que no se han ido integrando el código de QuPI-GAm en el sistema final conforme se iba desarrollando, sí que debería simularse este comportamiento de cara a la integración, es decir, se respetaría esta filosofía. Por la parte del juego objeto de la integración, se dispone de una batería de pruebas que puede ser lanzada en para comprobar que ninguna parte se ha visto afectada negativamente. Por otra parte, la integración no debería afrontarse en un único paso, sino que se debería proyectar un enfoque incremental donde se fuesen planteando fases diferenciadas, en cada una de las cuales se integrase solo un paquete específico de funcionalidades. Se tendría:

- **Primera fase.** Debería plantearse la incorporación de una **versión mínima** de QuPI-GAm donde únicamente se expusiera a los jugadores el formulario de aportación. Esto permite dar un primer paso, aproximarse al código e identificar los distintos componentes y partes del sistema final. Familiarizarse con el sistema final es importante para poder llevar a cabo la integración de manera exitosa y con el menor impacto.

- **N Fases posteriores.** Una vez superada con éxito la primera fase, se trata de seguir incorporando funcionalidades en fases sucesivas, lo que en QuPI-GAM se podría plantear en dos, una fase dedicada a integrar el apartado de validación, que al ya existir en el sistema final deberían ser fusionados y otra dedicada a incorporar toda la parte relativa a gamificación, que al ser nueva en el juego se podría incorporar de una forma más aislada y con menos riesgo.

Experiencia integradora

Por último, cabe exponer cuál ha sido la experiencia integradora, aun no habiéndose llevado a cabo ningún tipo de integración en la versión comercial del juego. Con vistas a enfrentarse a lo que implica este procedimiento, sí que se llegó a inspeccionar el código de Whizz y se realizó una integración parcial en una rama separada del proyecto. De esto pueden extraerse algunas pautas a tener en cuenta cuando se quiera afrontar la integración completa y a nivel comercial:

- Lo primero que hay que afrontar es el **análisis del código** del sistema destinatario. Esto permite principalmente **localizar recursos y utilidades compartidas**. Al realizar la integración hay que descartar esas funciones o modelos de la base de datos coincidentes e invocar directamente las incluidas en el sistema final, lo que puede requerir leves modificaciones del código a integrar, si por ejemplo el tipo de manejado o el formato de los datos no es coincidente.
- Hay que comprender el **impacto que va a causar el código a integrar**. Esto implica hacer una distinción entre aquellas partes independientes que pueden incluirse con más libertad (al complementar el código ya existente) y aquellas partes que deben establecer una relación con el código destinatario. Para el caso concreto, por ejemplo la definición del usuario impacta directamente, ya que está definido en ambos sistemas. La solución es mantener la definición del usuario del sistema final, pero agregándole una entrada que apunte al perfil de colaborador que si es definido por el código nuevo introducido.
- Fue necesario llevar a cabo una **refactorización del código** de QuPI-GAM para adaptar los nombres de las variables al estilo (semántico) empleado en el sistema final. De esta forma se mantiene la coherencia y el código resulta más mantenible.
- Desde el punto de vista del frontend, es necesario acordar con el encargado del diseño y la estética del juego el **rediseño final de los formulario y restantes interfaces**. Dado que todo el juego compone un conjunto estéticamente coherente, no pueden introducirse las interfaces tal cual se desarrollaron para el prototipo. Deben reutilizarse todos los componentes gráficos que ya se estén empleando en el juego.

- Es importante ir elaborando una traza o **log con cada cambio** que se vaya introduciendo en el sistema final a medida que se va avanzando en la integración. Esto permite, dado un problema, volver atrás en los pasos que se dieron y localizar más fácilmente la fuente de conflicto. Esencialmente hay que registrar qué nuevas funciones, clases y modelos son agregados, qué funciones ya existentes son modificadas y qué funciones del sistema destinatario son reutilizadas desde el nuevo código.
- Deben ser **reconfigurados los servicios de terceros** integrados en QuPI-GAm (re-Captcha, autenticación con Google + y Facebook) que requerían registrar el dominio de la aplicación especificando el dominio o dominios asignados al sistema final.
- Hay que considerar si existen **funcionalidades compartidas por ambos sistemas** y definir de qué forma se fusionan ambas. En este caso, el sistema final ya dispone de una interfaz independiente de acceso restringido a personal donde se pueden aportar nuevas preguntas, revisar estas y traducirlas. Esto impacta directamente con el apartado de validación desarrollado en QuPI-GAm. Para resolver esta coincidencia habría que tomar como base lo ya existente en el sistema final y readaptarlo para que integre el flujo asociado a las preguntas aportadas por parte de los jugadores hasta pasar a ser catalogadas como preguntas validadas.

Conclusiones

PARA concluir, en este capítulo se incluye una pequeña reflexión acerca de lo que ha supuesto el proyecto en su totalidad, destacándose como factor diferencial el hecho de haber sido desarrollado en el marco de un proyecto real con gran potencial y habiéndose iniciado incluso la colaboración en el contexto de prácticas en empresa. Se repasan los objetivos inicialmente expuestos, con el fin de justificar en qué medida han sido cubiertos. Se comentan también las dificultades más destacables identificadas a lo largo del desarrollo y se reflexiona acerca de líneas de trabajo que podrían seguirse de cara a plantear un trabajo futuro que permita extender este proyecto.

7.1 Objetivos alcanzados

Se revisan en esta sección los objetivos inicialmente planteados para ver en qué medida se han cumplido. Se vuelven a listar para ello el conjunto de objetivos definidos:

1. Realizar un estudio que permita contextualizar el entorno de trabajo. Esto supone investigar acerca de lo que implica y hasta dónde llega el mercado y el ecosistema del desarrollo de juegos para móviles. Esto se ha de volcar como estado del arte.

¿Alcanzado? SI. Se ha llevado a cabo un completo estudio acerca de los conceptos fundamentales en torno a esta temática, desde el el mercado de juegos para móvil hasta lo que supone el desarrollo de una app. Todo ello ha resultado en parte inspirador y ha podido reflejarse a lo largo del diseño y desarrollo.

2. Agregar un servicio de terceros que permita gestionar la autenticación de los usuarios.

¿Alcanzado? SI. Se ha habilitado la opción de autenticarse e iniciar sesión en el sistema mediante una cuenta de Google + y/o Facebook, dos proveedores de peso para los que se espera que la gran mayoría de jugadores disponga de al menos una cuenta en alguna de ellos.

3. Incorporar un mecanismo que permita gestionar correctamente la sesión de cada usuario (creación, asignación, recuperación y cierre).

¿Alcanzado? SI. Apoyándose en la librería *sessions* ofrecida por el framework WebApp2 se ha incorporado esta utilidad de base al servidor.

4. Introducir una política de autorización de acceso y asignación de roles que permita disponer de un control de acceso a zonas restringidas del sistema.

¿Alcanzado? SI. Se ha definido un mecanismo que permite comprobar, antes de servir la petición, si el usuario que la emitió tiene el rol requerido para acceder a ese servicio o al área de la aplicación solicitada.

5. Incorporar un mecanismo para el soporte a la internacionalización.

¿Alcanzado? SI. A través de la definición de diccionarios con los idiomas soportados como clave, la detección del idioma del dispositivo y un modelo *Translation* en la base de datos, se consigue cumplir totalmente este requerimiento, resultando sencilla la incorporación de nuevos idiomas en caso de que fuese necesario.

6. Emplear tecnologías multiplataforma y diseños responsive que garanticen la correcta visualización en entornos multidispositivo.

¿Alcanzado? SI. Para el desarrollo de las interfaces en el lado cliente se han empleado el framework Bootstrap, que facilita el desarrollo de un diseño adaptable.

7. Crear un formulario accesible por jugadores tanto identificados como anónimos que permita introducir una pregunta, la respuesta correcta y opciones incorrectas, así como especificar idioma de la pregunta y categoría a la que pertenece.

¿Alcanzado? SI. Este requisito ha sido satisfecho por el cumplimiento de la historia de usuario US_UPLOAD_01.

8. Incorporar un mecanismo de protección frente a robots en el formulario de aportación, para evitar ataques que traten de saturar de forma automatizada el sistema. Este debe ser permanente para usuarios no logueados y aparecer únicamente para la primera pregunta que aporte el jugador al inicio de cada sesión.

¿Alcanzado? SI. Se ha incorporado al formulario de validación una reCaptcha, un tipo de captcha evolucionada más usable y apta para dispositivos móviles y basada en imágenes.

9. Estudiar referencias de interés, artículos y libros dedicados a gamificación para tomar consciencia de los últimos avances. Toda la información de interés localizada debe ser recopilada como estado del arte.

10. Proyectar un diseño basado en gamificación y realizar un análisis que considere los factores condicionantes que entran en juego para justificar este diseño.

¿Alcanzado? SI. Se ha llevado a cabo un completo estudio acerca de los conceptos fundamentales en torno a lo que es y qué implica la gamificación. Se han estudiado algunas soluciones de gamificación específicas así como artículos académicos y se ha recopilado información acerca de frameworks que dan soporte a la elaboración de un diseño gamificado.

11. Analizar si existe en el mercado una alternativa similar a la que se pretende acometer para tomarla como referencia y/o punto comparativo.
¿Alcanzado? SI. Se ha analizado en profundidad un producto de éxito ya existente de características parecidas a QuPI-GAM: el juego *Preguntados*, que permite que sus jugadores aporten, revisen y traduzcan preguntas.
12. Crear un perfil de colaborador, como punto central donde el jugador pueda consultar su actividad y todo el *feedback* relativo a la gamificación.
¿Alcanzado? SI. Este requisito ha sido satisfecho por el cumplimiento de la historia de usuario US_UPLOAD_02.
13. Definir un mecanismo que permita ofrecerle *feedback* al jugador, gracias al cual pueda conocer el impacto de su aportación en el sistema, en términos de preguntas aceptadas, agregadas al juego y descartadas. Se le ha de informar de sus avances y logros conseguidos desde la última vez que visitó el perfil.
¿Alcanzado? SI. Este requisito ha sido satisfecho por el cumplimiento de la historia de usuario US_UPLOAD_02.
14. Asignar al colaborador puntos de experiencia, en base a todos los parámetros de actividad medibles, como unidad base que permita a su vez incorporar un mecanismo de subida de nivel y alcance de rangos que mida su actividad y permita establecer un marco comparativo respecto a otros jugadores.
¿Alcanzado? SI. Se ha incorporado un mecanismo estático de asignación de puntuación en base a las acciones realizadas por el jugador, que se apoya en un sistema de puntuación propio que define que vincula una barrera de puntos totales adquiridos a superar para subir de nivel.
15. Establecer metas que puedan serle propuestas al jugador a modo de trofeos a conseguir.
¿Alcanzado? SI. Este requisito ha sido satisfecho por el cumplimiento de la historia de usuario US_UPLOAD_02.1.
16. Incorporar la construcción de rankings mensuales, en base a distintos parámetros combinables como número de preguntas, número de puntos de experiencia, localización del usuario y categoría de la pregunta.
¿Alcanzado? SI. Este requisito ha sido satisfecho por el cumplimiento de la historia de usuario US_UPLOAD_02.2.
17. Crear un formulario de validación (no accesible a jugadores sino a personal destinado) a través del cual se puedan recuperar las preguntas subidas por los jugadores. El validador podrá visualizar la pregunta, corregirla y aceptarla o bien descartarla, si no se considera de valor. El validador podrá seleccionar el idioma y categoría para su recuperación, así como corregir estos parámetros.

¿**Alcanzado?** **SI**. Este requisito ha sido satisfecho por el cumplimiento de la historia de usuario US_VALIDATION_01.

18. Desarrollar un mecanismo automatizado que trate de detectar si ya ha sido incorporada definitivamente una pregunta al sistema que verse sobre una temática similar a una pregunta recuperada pendiente de validación.

¿**Alcanzado?** **SI**. Este requisito ha sido satisfecho por el cumplimiento de la historia de usuario US_VALIDATION_02. Se ha podido comprobar la utilidad de emplear la librería *search* para inspeccionar las palabras que contiene cada pregunta y determinar si existe una coincidencia significativa entre la propuesta y una validada.

19. Dado el marco colaborativo que se ha establecido con la empresa Furious Koalas, no se puede plantear hacer efectiva una integración al completo en la versión comercial de (*Whizz*), pero sí que se plantea estudiar las implicaciones de llevar a cabo la integración y dejar el sistema preparado para una futura integración incremental.

¿**Alcanzado?** **SI**. Aunque no se ha afrontado el proceso de integración al completo sí que se ha tenido una toma de contacto con el código del sistema final y se ha inspeccionado para plantear una futura integración.

7.2 Dificultades encontradas

De todos los objetivos planteados, como dificultad notable cabe destacar el caso referido a la historia de usuario US_VALIDATION_02, referente a la **detección de pregunta repetida**. Se afrontaron dos enfoques:

- El primero de ellos se descartó porque la librería de soporte requerida (nlkt) incluía **funciones escritas en C**, lo que viola las restricciones de App Engine. Intentar aislar de la citada librería únicamente las funciones requeridas resultó ser un tanto complejo.
- El otro enfoque se basaba en el empleo de las herramientas de búsqueda de Google con el fin de identificar preguntas que compartiesen términos similares. En la documentación de **Google apenas se ofrece información acerca de los criterios de búsqueda**, por lo que resulta difícil configurar correctamente los parámetros implicados para obtener el mejor ajuste cuando se lanza la búsqueda. Por otro lado, se obtenían resultados diferentes según si el proyecto se ejecutaba localmente o desplegado, lo que da la pista de que en local no se incluye todo el soporte de la librería *search*.

Por otro lado, dado el diseño donde todos los jugadores pueden ser potenciales aportadores y únicamente un pequeño grupo de personal escogido puede jugar el papel de validadores, se prevee que puede producirse una **descompensación en el ritmo de aportación-revisión**. De cara a la gamificación este puede ser un factor de riesgo, puesto que si se produce un cuello de botella y los validadores no revisan a buen ritmo, a los jugadores no les llega feedback de sus aportaciones.

7.3 Trabajo futuro

A continuación se presentan una serie de líneas que podrían seguirse para mejorar, evolucionar o sacar todo el partido al trabajo desarrollado en este proyecto:

- El trabajo ha servido para sentar las bases de un proyecto de largo recorrido. Dado que la integración completa de QuPI-GAM está incluida en la línea estratégica de la empresa con la que se ha colaborado, la línea de trabajo más clara es la de **afrentar una integración incremental** por la cual se incluya completamente el sistema en la aplicación móvil.
- Incorporar en QuPI-GAM funcionalidades interesantes que son directamente dependientes de la integración en el sistema final, y que por tanto deberían incluirse una vez llevada a cabo la integración, como es el caso del **canjeo de tickets** y **permitir que los jugadores valoren las preguntas** entre ellos a medida que les aparezcan en el juego (estos aspectos fueron discutidos en la sección 5.8.1).
- Crear nuevos trofeos a medida que se fuesen incorporando nuevas funcionalidades. Tomando como referencia el punto anterior por ejemplo, se podrían registrar **trofeos que midiesen el impacto que causan las preguntas** del usuarios. Esto daría pie a trofeos basados en: número de jugadores que te valoraron positivamente, número de veces que has valorado a otros jugadores, etc.
- Una vez integrado el sistema, convendría **definir una serie de métricas** que permitan esclarecer el impacto que causa el sistema, comprobándose **cómo responden los jugadores** a través de los datos de uso recopilados.
- Aplicar **lógica borrosa al mecanismo de detección de pregunta repetida** para no establecer el umbral de similitud en un único valor, lo que aporta rigidez al veredicto.
- Redefinir la participación de los jugadores aprovechando los rangos que se asignan para **permitir a ciertos jugadores adoptar el rol de validador** una vez han alcanzado un determinado rango, que da garantías sobre su fiabilidad como colaboradores.
- Permitir a los validadores que indiquen si procede que una **pregunta tiene cierto carácter local**, en cuyo caso introduciría la zona geográfica con la que se vincula. De esta forma estas preguntas serían mostradas en partidas localizadas en dicha la zona.
- **Optimizar la gestión en la constitución de rankings** siguiendo el modelo que ya se nombró en la sección 5.10.2.
- Devolver más *feedback* al jugador en cuanto a su actividad. Por una parte, si se descartó una pregunta se le deberían **exponer los motivos**. Por otro lado convendría incluir un área en el perfil y una consulta que le permitiese leer el **histórico de preguntas generadas** por él.

- Sugerir al jugador **cuál es el trofeo/s que está más cerca de conseguir** cada vez que visite el perfil.
- **Dinamizar el sistema de asignación de puntos**, incluyendo por ejemplo multiplicadores. También se podría plantear la asignación de rangos no en función del nivel, sino en función del grupo de trofeos del que se tengan más. Así, a través del rango, se definiría qué tipo de aportador es el jugador y resultaría más distintivo.

7.4 Conclusiones finales

Si bien el impacto del proyecto desarrollado no puede ser determinado objetivamente hasta que se trabaje en la línea integradora y pueda ser expuesto junto con el juego al público, se considera que el desarrollo de este proyecto ha resultado útil, ya que asienta la base de una propuesta muy interesante que, como demuestran los precedentes (ver sección 3.1.3) podría tener buena recepción y resultar ser todo un éxito.

En un sentido más allá del puramente práctico, este proyecto ha resultado enriquecedor en varios sentidos. Por un lado, poder entrar en contacto con un ambiente distinto al académico y estar cercana al desarrollo de un juego para móvil aun no habiendo participado de forma directa en su creación y desarrollo, resulta estimulante. El hecho de estar próxima a un proyecto tan interesante como es la creación de un juego multijugador masivo en línea supone un aprendizaje en sí mismo. Desde el punto de vista del aprendizaje, el haber entrado en contacto con tecnologías web y frameworks específicos supone un nuevo conocimiento adquirido, que sin duda resultará útil para futuros desempeños profesionales.

Desde el punto de vista técnico, destacaría la elección de App Engine como un acierto. A través de este IaaS se pone a disposición del desarrollador de forma gratuita tanto un servidor como una base de datos, donde el despliegue se realiza de forma sencilla y donde el desarrollo local resulta ser muy cómodo gracias a su enfoque sandbox. Desde el punto de vista organizativo, cabe destacar la utilidad de mantener reuniones cada semana o dos semanas con el equipo. Se ha podido comprobar el beneficio que aporta mantener un seguimiento y comunicaciones periódicas.

Como cualquier trabajo que se afronte, este podría haber alcanzado un grado de refinamiento mayor, aunque siempre hay que considerar la importancia de establecer un equilibrio entre el tiempo dedicado, los recursos invertidos y la calidad obtenida. Lo que si puede asegurarse es que detrás de estas páginas existe un esfuerzo real y compartido por quien escribe y por quien ha tutorizado todo el proceso. Y es que, como aseveró Napoleón Hill, “El esfuerzo sólo libera plenamente su recompensa cuando una persona se niega a abandonar”.

ANEXOS

Anexo A

Ficha de usuarios empleando técnica Personas



DATOS PERSONALES	
NOMBRE	Oliver Suárez Portillo
EDAD	43
NACIONALIDAD	Mejicano
GÉNERO	Hombre
OCUPACIÓN	Conserje de instituto
ESTUDIOS	Profesional Técnica en Administración (educación nivel medio superior)
CLASE SOCIAL	Clase media
SITUACIÓN ECONÓMICA	Casado y sin hijos. Vive de forma humilde pero holgada, pudiendo afrontar el pago de su vivienda (su mayor carga económica) gracias a su sueldo y el de su marido.
INTERESES	Amante del deporte, del campo y de la buena gastronomía. Disfruta de eventos y reuniones sociales con amigos y familiares. Viviendo en un entorno educativo, presenta interes por expandir sus conocimientos de cultura general.
HÁBITOS/RUTINAS	Cada día laboral se caracteriza por: <ul style="list-style-type: none"> - Se despierta a las 6:30 y sale a correr media hora - Desayuna mientras lee noticias en su smartphone - Se desplaza en autobús a su puesto de trabajo - Dedicar la mañana a atender las tareas pendientes o requerimientos del personal y estudiantes del centro, pudiendo disponer de huecos libres donde nadie le demanda - Come con sus compañeros de trabajo - Vuelve a casa en autobús y dedica unas horas a relajarse - El resto de la tarde lo dedica a tareas del hogar y a actividades ociosas. Sale a la calle y charla un rato con sus vecinos - Al caer la noche se relaja junto a su marido viendo en la televisión competiciones deportivas
POSICIÓN FRENTE AL PRODUCTO	
OBJETIVO	Busca una vía de entretenimiento útil. Tiene cierto pesar por no haber podido ir a la universidad, y le resulta gratificante cada vez que acierta las respuestas de las preguntas. Pretende aprovechar el juego para ampliar sus conocimientos de cultura general y poder comentarlos con los profesores del centro educativo donde trabaja. También le resulta llamativo poder plasmar en el juego todo aquello que aprende por otras vías aprovechando la posibilidad de aportar sus propias preguntas al sistema, lo que le permite reafirmar la utilidad de lo aprendido.
EXPECTATIVAS	Del producto espera poder: <ul style="list-style-type: none"> - Adquirir conocimiento - Retarse a sí mismo - Distraerse en los ratos libres que le deja el trabajo - Poder darle salida a las nuevas curiosidades que aprende - Poder difundir el folclore asociado a su tierra y sus orígenes
ACTITUD	<i>Proactividad baja-media</i> para generar contenidos En balance generará más contenidos de los que consuma
ROL	Usuario
CONTEXTO DE USO	
DISPOSITIVO	Ipad Air 2
FRECUENCIA	Realiza un seguimiento diario de su perfil de usuario en el juego. Accede varias veces al día: A lo largo de la mañana varias sesiones cortas donde propone preguntas y en la tarde-noche una sesión larga donde disfruta de una o varias partidas online.
LUGAR	Oficina de trabajo y en casa



DATOS PERSONALES	
NOMBRE	Lisa Fuentes García
EDAD	22
NACIONALIDAD	Española
GÉNERO	Mujer
OCUPACIÓN	Fotógrafa freelance
ESTUDIOS	Bachillerato de bellas artes Curso de fotografía e imagen en una escuela técnica
CLASE SOCIAL	Clase media-baja
SITUACIÓN ECONÓMICA	Proveniente de una unidad familiar humilde de 5 componentes, recientemente independizada, ha cambiando de ciudad buscando nuevas oportunidades. Dispone de un empleo como camarera a media jornada e intenta abrirse camino en su profesión de fotógrafa. Su capacidad económica es limitada.
INTERESES	Una chica activa y capaz. Con ganas de descubrir nuevas aficiones y entretenimientos, cultivar un buen círculo de contactos, disfrutar de sus amistades y ahorrar todo el dinero posible para organizar viajes.
HÁBITOS/RUTINAS	En un día laboral: <ul style="list-style-type: none"> - Se despierta a las 7:00 de la mañana - Consulta sus redes sociales y correo mientras desayuna - Se marcha en metro a su puesto de trabajo - Come algo en el bar en el que trabaja - Aprovecha la tarde para descubrir nuevos rincones de la ciudad que fotografiar, para quedar con sus amistades o para hacer cursos que le ayuden a mejorar en su profesión - Vuelve en metro a casa - Para desconectar en lo que queda de día va al gimnasio
POSICIÓN FRENTE AL PRODUCTO	
OBJETIVO	Concibe el producto como una herramienta de entretenimiento. Dado que pasa el día fuera de casa, necesita un juego que la distraiga en los trayectos y huecos del día en que está sola. Por su interés en viajar y la gran importancia que le da al hecho de tener muchos contactos y amistades, le gusta poner atención a las preguntas curiosas, ya que luego suponen un un buen comodín para dar pie a conversaciones.
EXPECTATIVAS	Con el uso del producto espera poder: <ul style="list-style-type: none"> - Distraerse en los ratos libres - Compartir lo que aprende en los diferentes cursos a los que asiste, ya que se dedica a una disciplina que le apasiona - Aprender datos de cultura general o actualidad curiosos
ACTITUD	<i>Proactividad media</i> para generar contenido En balance consumirá más contenidos de los que genere
ROL	Usuario
CONTEXTO DE USO	
DISPOSITIVO	Smartphone Huawei Honor 6
FRECUENCIA	Inicia 2 o 3 sesiones cortas durante el día para comprobar su perfil y proponer alguna pregunta que se le ocurra según las vivencias del día. Al caer la noche, dedica un rato a jugar una partida online.
LUGAR	En casa, antes de acostarse. Durante el día, en los lapsos de transporte.

Referencias

- [Blo06] S. Blomkvist. The User as a personality. A Reflection in the theoretical and practical use of personas in HCI design, 2006.
- [BOMW04] M. Blythe, K. Overbeeke, A. Monk, y P. Wright. *Funology: From Usability to Enjoyment*. Kluwer Academic Publishers, 2004. ISBN: 978-1402029667.
- [BVL⁺12] J. Becerril, A. Vallejo, A. Lumbreras, G. Chávez, A. Duk, y R. Torres. La web 2.0: Un análisis de su impacto en lo social, político, cultural y económico. *Investigación Universitaria Multidisciplinaria*, páginas 23–34, 2012.
- [C.84] Coonrad C. *The game of work. How to Enjoy Work as Much as Play*. Liberty Press, 1984. ISBN: 978-1423630852.
- [DDKN11] S. Deterding, D. Dixon, R. Khaled, y L. Nacke. From Game Design Elements to Gamefulness: Defining "Gamification". En *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, MindTrek '11, páginas 9–15, 2011.
- [Det12] S. Deterding. Experts split, hold various opinions. *Blog post*, 2012. Disponible online: <http://gamification-research.org/2012/05/experts-split-hold-various-opinions-on-possible-future-of-gamification/>.
- [DSN⁺11] S. Deterding, M. Sicart, L. Nacke, K. O'Hara, y D. Dixon. Gamification. Using Game-design Elements in Non-gaming Contexts. En *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, páginas 2425–2428, New York, NY, USA, 2011. ACM.
- [EAG12] E. Estellés-Arolas y F. González. Towards an integrated crowdsourcing definition. *Journal of Information Science*, 38(2):189–200, 2012.
- [EGD16] *The future of the European mobile game ecosystem*. Disponible online: <http://www.mobilegamearch.eu/wp-content/uploads/2013/09/MGAfinalbrochure.pdf>, European Games Developer Federation, 2016.

- [GBSTN16] A.K.B. Gopinath Bharathi, A. Singh, C.S. Tucker, y H.B. Nembhard. Knowledge discovery of game design features by mining user-generated feedback. *Computers in Human Behavior*, 60:361–371, 2016.
- [GD05] M. Gagné y E. Deci. Self-determination theory and work motivation. *Journal of Organizational behavior*, 26(4):331–362, 2005.
- [GGN⁺16] C.S. González, N. Gómez, V. Navarro, M. Cairós, C. Quirce, P. Toledo, y N. Marrero-Gordillo. Learning healthy lifestyles through active videogames, motor games and the gamification of educational activities. *Computers in Human Behavior*, 55:529–551, 2016.
- [Gue05] E. Guerrero. The competition as quality gender on Spanish television. *Comunicar Journal 25: Quality Television*, 13, 2005.
- [Hag12] P. Hagglund. Taking gamification to the next level. *Unpublished PhD thesis. Universidad de Umea.*, 2012.
- [HKS14] J. Hamari, J. Koivisto, y H. Sarsa. Does gamification work? - A literature review of empirical studies on gamification. *Proceedings of the Annual Hawaii International Conference on System Sciences*, páginas 3025–3034, 2014.
- [HLZ04] R. Hunicke, M. LeBlanc, y R. Zubek. MDA: A formal approach to game design and game research. Workshop on Challenges in Game, AAAI Press, 2004.
- [IS08] A. Iacovelli y C. Souveyet. Framework for Agile Methods Classification. páginas 91–102. MoDISE-EUS, 2008.
- [KC16] M.-S. Kuo y T.-Y. Chuang. How gamification motivates visits and engagement for online academic dissemination - An empirical study. *Computers in Human Behavior*, 55:16–27, 2016.
- [Ken10] S. Kent. *The Ultimate History of Video Games: from Pong to Pokemon and beyond*. Three Rivers Press, 2010.
- [LA15] R.N. Landers y M.B. Armstrong. Enhancing instructional outcomes with gamification: An empirical test of the Technology-Enhanced Training Effectiveness Model. *Computers in Human Behavior*, 2015.
- [LBCA15] R.N. Landers, K.N. Bauer, R.C. Callan, y M.B. Armstrong. *Psychological theory and the gamification of learning*. 2015.

- [LBV16] *Libro Blanco del Desarrollo Español de Videojuegos*. Disponible online: <http://www.dev.org.es/libroblancodev2016>, Asociación Española de Empresas Productoras Y Desarrolladoras de Videojuegos y Software de Entretenimiento, 2016.
- [Let06] P. Letelier. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). 2006.
- [M.12] Nelson M. Soviet and American precursors to the gamification of work. En *International Conference on Media of the Future, Academic MindTrek'12, Tampere, Finland, October 3-5, 2012*, páginas 23–26, 2012.
- [Mal82] T. Malone. Heuristics for Designing Enjoyable User Interfaces: Lessons from Computer Games. En *Proceedings of the 1982 Conference on Human Factors in Computing Systems, CHI'82*, páginas 63–68, 1982.
- [Mas43] A. Maslow. A theory of human motivation. *Psychological review*, 50(4):370, 1943.
- [MF16] R. McDaniel y J. Fanfarelli. Building Better Digital Badges: Pairing Completion Logic With Psychological Factors. *Simulation and Gaming*, 47(1):73–102, 2016.
- [Mil14] J. Miller. The fourth screen: Mediatization and the smartphone. *Mobile Media & Communication*, 2(2):209–226, 2014.
- [Mor14] J. Moreno. Spanish quizzes and games shows: Evolution and historical perception (1956-1975). *Estudios Sobre el Mensaje Periodístico*, 20:27–42, 2014.
- [MRGAM15] A. Mora, D. Riera, C. González, y J. Arnedo-Moreno. A literature review of gamification design frameworks. En *Games and Virtual Worlds for Serious Applications (VS-Games), 2015 7th International Conference on*, páginas 1–8, 2015.
- [MSW09] M. Montola, J. Stenros, y A. Waern. *Pervasive Games: Theory and Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009. ISBN: 978-0123748539.
- [NCWW16] R. Newman, V. Chang, R.J. Walters, y G.B. Wills. Web 2.0 - The past and the future. *International Journal of Information Management*, 36(4):591–598, 2016.
- [PB14] H. Park y J. Bae. Study and Research of Gamification Design. *International Journal of Software Engineering and Its Applications*, 8:19–28, 2014.

- [RCV09] U. Ritterfeld, M. Cody, y P. Vorderer. *Serious Games: Mechanisms and Effects*. Routledge, Taylor and Francis, 2009. ISBN: 978-0415993709.
- [San15] D. Sanderson. *Programming Google App Engine with Python: Build and Run Scalable Python Apps on Google's Infrastructure*. O'Reilly Media, Inc. First edition, 2015. ISBN: 1491900253.
- [SF14] K. Seaborn y D.I. Fels. Gamification in theory and action: A survey. *International Journal of Human Computer Studies*, 74:14–31, 2014.
- [SIE15] *La Sociedad de la Información en España 2015*. Disponible online: <http://www.fundaciontelefonica.com/artecultura/sociedad-de-la-informacion/informe-sie-espana-2015/>, Fundación Telefónica España, 2015.
- [Sig15] M. Sigala. The application and impact of gamification funware on trip planning and experiences: the case of TripAdvisor's funware. *Electronic Markets*, 25(3):189–209, 2015.
- [Y.16] Chou Y. *Actionable Gamification: Beyond Points, Badges, and Leaderboards*. Octalysis Media, 2016. ISBN: 978-1511744041.

Este documento fue editado y tipografiado con \LaTeX empleando la clase **esi-tfg** (versión 0.20170125) que se puede encontrar en:
https://bitbucket.org/arco_group/esi-tfg

[respeta esta atribución al autor]

