



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA
Departamento de Informática

PROYECTO FIN DE CARRERA
Sistema Gestor de Contenidos Web
para el aprendizaje de idiomas: BABEL

Autor: Javier Alonso Albusac Jiménez

Director: Carlos González Morcillo

Diciembre, 2005

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

Resumen

La aparición de nuevas tecnologías y la posibilidad de comunicación a través de Internet abre un amplio abanico de posibilidades en el aprendizaje de idiomas.

La mayoría de sitios en la red dedicados a la enseñanza de idiomas, son plataformas e-learning de pago por servicios o lugares donde se ponen a disposición del usuario material docente. No suelen permitir la actualización de los contenidos por parte del usuario, de una forma directa y sencilla.

Lo que se pretende conseguir con Babel es una plataforma que sirva de apoyo en el aprendizaje y práctica de idiomas, donde los usuarios puedan compartir sus conocimientos contribuyendo al crecimiento de la base de contenidos, en un entorno controlado por usuarios con diferentes roles y responsabilidades sobre la información.

*Este proyecto está dedicado a mis
Padres, a mis hermanos y a Vanesa,
por su constante apoyo y confianza.*

Agradecimientos

A Carlos González, por su apoyo, ayuda y su gran idea.

A mi madre, Manuel y Vanesa por la revisión de este documento.

A Ana, por la traducción de parte de la interfaz a francés.

Índice

1.- INTRODUCCIÓN.....	1
1.1.- ENFOQUE.....	2
1.2.- ESTRUCTURA DEL TRABAJO.....	2
2.- OBJETIVOS DEL PROYECTO.....	4
3.- ANTECEDENTES, ESTADO DE LA CUESTIÓN.....	7
3.1.- ASPECTOS FUNDAMENTALES DE LA GESTIÓN DE CONTENIDOS.....	7
3.1.1.- <i>Definiciones</i>	7
3.1.2.- <i>Tendencias en la Gestión de Contenidos</i>	8
3.1.3.- <i>Características deseables en un Gestor de Contenidos</i>	9
3.1.4.- <i>Tareas que permite un Gestor de Contenidos</i>	9
3.2.- CASOS DE ESTUDIO.....	10
3.2.1.- <i>Zope, servidor de aplicaciones web</i>	11
3.2.2.- <i>Typo3</i>	14
3.2.3.- <i>Content Management Server de Microsoft</i>	16
3.2.4.- <i>Comparativa entre Zope y Typo3</i>	18
3.3.- USABILIDAD WEB.....	19
3.4.- DESARROLLO DE WEB DINÁMICO.....	24
3.4.1.- <i>PHP</i>	24
3.4.2.- <i>PHP 5</i>	26
3.4.3.- <i>ASP y ASP.NET</i>	26
3.4.4.- <i>JSP</i>	27
3.5.- <i>CSS</i>	28
3.6.- ENSEÑANZA Y APRENDIZAJE DE IDIOMAS.....	30
3.6.1.- <i>Métodos utilizados</i>	30
3.6.2.- <i>Ejercicios prácticos</i>	33
3.6.3.- <i>Sitios en la red para el aprendizaje de idiomas</i>	34
4.- MÉTODOS Y FASES DE TRABAJO.....	38
4.1.- INGENIERÍA DEL SOFTWARE.....	38
4.1.1.- <i>Ciclo de vida del Software</i>	38
4.1.2.- <i>Diseño Multicapa</i>	44
4.2.- VISUALIZACIÓN EN DISPOSITIVOS DE DIFERENTES TAMAÑOS.....	46
4.3.- INTERFAZ MULTIIDIOMA.....	47

4.4.- ARQUITECTURA DEL SISTEMA.....	49
4.5.- MÁQUINA DE ESTADOS.....	50
4.5.1.- Roles de usuario.....	51
4.5.2.- Acciones.....	52
4.5.3.- Estados	53
4.5.4.- Transiciones de estados.....	54
4.5.5.- Permisos sobre los contenidos.....	55
4.5.6.- Máquina de estados inicial.....	56
4.6.- MÓDULOS.....	57
4.6.1.- Módulo de administración.....	59
4.6.2.- Módulo de gestión de contenidos.....	60
4.7.- CONTENIDOS.....	61
4.7.1.- Acciones posibles sobre un contenido.....	64
4.7.2.- Tareas pendientes.....	65
4.7.3.- Bloqueo de contenidos.....	67
4.7.4.- Documentos teóricos.....	68
4.7.5.- Traducciones.....	71
4.7.6.- Correcciones sobre las traducciones.....	72
4.7.7.- Ejercicios prácticos.....	74
4.7.8.- Exámenes.....	78
4.8.- BUSCADOR DE CONTENIDOS.....	79
4.8.1.- Creación de índices para los documentos.....	82
4.8.2.- Creación de índices para las traducciones.....	82
4.8.3.- Creación de índices para los ejercicios prácticos.....	83
4.8.4.- Creación de índices para los exámenes.....	84
4.9.- DICCIONARIOS.....	85
4.10.- COMUNICACIÓN ENTRE LOS USUARIOS.....	87
4.10.1.- Mensajería interna.....	88
4.10.2.- Chat.....	89
4.10.3.- Comunicación por voz.....	92
4.11.- JUEGOS.....	93
4.11.1.- El Ahorcado.....	93
4.11.2.- Juego de las preguntas.....	94
4.12.- SÍNTESIS DE VOZ.....	94

4.13.- USABILIDAD WEB.....	95
4.14.- MÉTODOS DE APRENDIZAJE.....	98
5.- RESULTADOS.....	99
5.1.- RESULTADOS OBTENIDOS.....	99
5.2.- POSIBLES COSTES DE IMPLANTACIÓN DEL SISTEMA.....	99
5.2.1.- <i>Costes anuales de hosting.....</i>	<i>100</i>
5.2.2.- <i>Costes de alquiler de un servidor.....</i>	<i>102</i>
5.2.3.- <i>Coste de adquisición de un servidor.....</i>	<i>103</i>
5.2.4.- <i>Decisiones de las tecnologías utilizadas.....</i>	<i>104</i>
5.2.5.- <i>Mecanismos de financiación del sistema.....</i>	<i>106</i>
6.- CONCLUSIONES Y PROPUESTAS.....	108
6.1.- CONCLUSIONES.....	108
6.2.- PROPUESTAS Y LÍNEAS FUTURAS.....	109
7.- BIBLIOGRAFÍA.....	111
8.- ANEXOS.....	113
8.1.- ANEXO A – MANUAL DEL USUARIO.....	113
8.1.1.- <i>Cuenta de usuario.....</i>	<i>113</i>
8.1.2.- <i>Herramientas.....</i>	<i>117</i>
8.1.3.- <i>Selección de idiomas.....</i>	<i>121</i>
8.1.4.- <i>Administración del sistema.....</i>	<i>122</i>
8.1.5.- <i>Administración de contenidos.....</i>	<i>123</i>
8.1.6.- <i>Contenidos.....</i>	<i>125</i>
8.2.- ANEXO B – DIAGRAMAS DE CASOS DE USO.....	127
8.3.- ANEXO C - DIAGRAMA DE CLASES.....	132
8.4.- ANEXO D- ESTRUCTURA DEL CÓDIGO FUENTE.....	135
8.5.- ANEXO E – INSTALACIÓN Y CD ADJUNTO.....	137

Lista de símbolos

Figura 3.1 Contenido del sitio web.....	21
Figura 3.2 Visión del usuario del sitio web a primera vista.....	22
Figura 3.3 Ejemplo de migas de pan.....	23
Figura 3.4 Lenguajes de programación del lado del servidor.....	24
Figura 3.5 Evolución del número de programadores que utilizan PHP.....	25
Figura 4.1 Diseño Multicapa.....	44
Figura 4.2 Arquitectura de Babel.....	49
Figura 4.3 Estructura de tablas de la máquina de estados.....	51
Figura 4.4 Máquina de estados inicial del sistema.....	56
Figura 4.5 Diagrama de clases de los módulos.....	58
Figura 4.6 Diseño de tablas para los módulos del sistema.....	59
Figura 4.7 Diagrama de clases sobre contenidos (abreviado).....	62
Figura 4.8 Diseño de tablas para los contenidos.....	63
Figura 4.9 Editor WYSIWYG.....	64
Figura 4.10 Bloqueo de contenidos mediante la implementación de semáforos.....	67
Figura 4.11 Diseño de documentos teóricos.....	70
Figura 4.12 Diseño traducciones.....	72
Figura 4.13 Ejemplo de corrección.....	73
Figura 4.14 Diseño de correcciones.....	73
Figura 4.15 Diseño de ejercicios prácticos.....	75
Figura 4.16 Diseño de exámenes.....	78
Figura 4.17 Creación de índices para el buscador.....	81
Figura 4.18 Ejemplo de búsqueda en diccionarios.....	87
Figura 4.19 Ejemplo de corrección realizada en el chat.....	89
Figura 4.20 Estructura de tablas de la base de datos para el Chat.....	90
Figura 4.21 Ejemplo de usuario con posibilidad de hacer llamada con Skype.....	92
Figura 4.22 Zonas del sitio web.....	96
Figura 8.1 Creación de una nueva cuenta de usuario.....	113
Figura 8.2 Cuenta de usuario.....	114
Figura 8.3 Mis contenidos.....	115
Figura 8.4 Tareas pendientes.....	116
Figura 8.5 Buzón de mensajes.....	117

Figura 8.6 Buscador de contenidos.....	117
Figura 8.7 Diccionarios.....	118
Figura 8.8 Chat.....	120
Figura 8.9 Panel de administración.....	122
Figura 8.10 Administración de contenidos.....	124
Tabla 4.1 Roles de usuario.....	51
Tabla 4.2 Acciones sobre los contenidos.....	53
Tabla 4.3 Estados en los que se puede encontrar un contenido.....	54
Tabla 4.4 Transiciones de estado.....	55
Tabla 5.1 Costes de Hosting en plataforma GNU/Linux.....	100
Tabla 5.2 Costes de Hosting en plataforma Windows.....	101

1.- INTRODUCCIÓN

La extensión y uso de Internet en los últimos años rompe límites y barreras en la comunicación entre personas que trabajan de forma colaborativa. Es posible que las personas que trabajan en un mismo proyecto, se encuentren físicamente en diferentes lugares y en distintos intervalos de tiempo. Es por esto, que cada vez es más frecuente ver a personas de diferentes nacionalidades trabajar por un objetivo común, donde la comunicación y el entendimiento entre ellos es fundamental.

Lo que se pretende con Babel es crear un sistema para el aprendizaje y práctica de idiomas, con herramientas síncronas y asíncronas que faciliten la comunicación entre personas de diferentes lenguas.

En Babel juega un papel fundamental la gestión de contenidos, por las ventajas que aporta tanto en ahorro de costes como de tiempo.

En la mayoría de sistemas de gestión de contenidos podemos encontrar un sistema de roles y flujos de trabajo entre los usuarios, que permiten un mayor control sobre la información almacenada y facilita la colaboración entre los usuarios. Se busca tanto el crecimiento como la actualización de la base de contenidos de forma distribuida, separando la información de el modo en el que se presenta, pudiendo ser compartida y reutilizada en cualquier momento.

Actualmente existen en la red, herramientas de gestión de contenidos de carácter libre como pueden ser Zope o Typo3 que gozan de soporte de estándares abiertos, y de carácter privativo como CMS (Content Management Server) de Microsoft.

1.1.- ENFOQUE

El principal objetivo del proyecto es el desarrollo de una herramienta que sirva de apoyo en el aprendizaje y la práctica de idiomas.

Para que el sistema sea flexible, existe una máquina de estados formada por diferentes elementos modificables como son roles de usuario, estados en los que se pueden encontrar los contenidos, acciones que podemos realizar y transiciones entre estados, que determinan de forma dinámica el comportamiento y la forma de operar del sistema con respecto a los contenidos.

1.2.- ESTRUCTURA DEL TRABAJO.

El documento se divide en siete capítulos y cinco anexos:

- **Capítulo 1**, Introducción, enfoque del proyecto y estructura del documento.
- **Capítulo 2**, Antecedentes, estado de la cuestión. Para este proyecto se ha realizado estudios sobre diferentes tecnologías y herramientas, en este capítulo se comenta cual es la situación actual de todas estas herramientas y tecnologías.
- **Capítulo 3**, Objetivos del trabajo. Enumeración y descripción de cada uno de los objetivos propuestos.
- **Capítulo 4**, Métodos y fases de trabajo. En este capítulo se describen los diferentes elementos que componen el sistema, métodos empleados y fases del ciclo de vida del software que se han llevado acabo para la realización del proyecto.
- **Capítulo 5**, Resultados obtenidos tras la realización del proyecto y un análisis de los posibles costes que supondrían la futura implantación del sistema.
- **Capítulo 6**, Conclusiones y propuestas de futuro. Cuales han sido las

conclusiones obtenidas, y algunas propuestas que pueden mejorar el sistema actual.

■ **Capítulo 7**, Bibliografía y referencias web utilizadas para la realización del proyecto.

■ **Anexos:**

- Anexo A: Manual del usuario.
- Anexo B: Diagramas de casos de uso donde se describe la funcionalidad del sistema
- Anexo C: Diagramas de las clases más importantes del sistema.
- Anexo D: Estructura del código fuente.
- Anexo E: Instalación y CD adjunto

2.- OBJETIVOS DEL PROYECTO

En este Proyecto Fin de Carrera se pretende conseguir el diseño y la construcción de un sistema de gestión de contenidos que sirva como herramienta de apoyo al aprendizaje y práctica de idiomas.

■ **Gestión de contenidos.**

El sistema dispondrá de una base de contenidos donde los usuarios podrán crear, actualizar y reutilizar contenidos de diferente índole.

- Documentos teóricos.
- Traducción libre y correcciones.
- Ejercicios prácticos.
- Exámenes (formados por ejercicios prácticos.)

La información almacenada en el sistema permutará entre diferentes etapas: adquisición, supervisión y finalmente publicación, buscando el mayor control posible sobre esta.

Las acciones que un determinado usuario podrá realizar en cada momento, dependerán de su rol y del estado en el que se encuentre el contenido.

■ **Comportamiento dinámico del sistema.**

Se buscará que el sistema sea lo más flexible posible, para ello se implementará una máquina de estados que determinará los diferentes flujos de trabajo. La máquina de estados estará formada por los siguientes elementos:

- Roles de usuario.
- Acciones posibles sobre los contenidos.
- Estado en el que se puede encontrar un contenido.
- Transiciones entre estados (estado-acción-estado siguiente).
- Permisos. De las acciones definidas, cuáles son las que puede realizar cada rol.

■ **Buscador de contenidos.**

En aquellos sistemas en los que su base de contenidos alcanza un gran volumen

de tamaño, acceder a la información que deseamos de forma rápida es más complicado a pesar de que esté categorizada y ordenada. Se desarrollará un buscador que permita a cualquier usuario acceder a la información a partir de un conjunto de palabras claves.

■ **Diccionarios y Síntesis de voz.**

Implantación de diccionarios en el sistema para los diferentes idiomas activos. Se realizará síntesis de voz sobre las palabras para que el usuario tenga la posibilidad de escuchar la correspondiente pronunciación.

■ **Interfaz Multiidioma.**

Debido a que el sistema será utilizado para el aprendizaje de varios idiomas, serán usuarios de diferentes nacionalidades los que puedan hacer uso de él, por lo que es requisito fundamental que la interfaz se encuentre en varios idiomas.

■ **Comunicación entre los usuarios.**

Se habilitará en el sistema un chat dotado de diferentes salas, dependiendo del idioma que desee aprender el usuario, con herramientas específicas para la comunicación entre personas de diferentes lenguas.

■ **Posibilidad de aprender jugando.**

El sistema dispondrá de dos juegos: el “Ahorcado” donde se intentará adivinar una determinada palabra de un idioma en concreto y el “Juego de las preguntas”, donde se irán mostrando de forma aleatoria ejercicios prácticos almacenados en el sistema.

■ **Usabilidad.**

Se estudiarán y aplicarán técnicas de usabilidad web, para que el sistema sea lo más intuitivo posible y sencillo de utilizar.

■ **Multiplataforma.**

El dispositivo utilizado por el usuario (dispositivos de diferentes tamaños) o su sistema operativo no debe ser un impedimento para poder utilizar el sistema.

■ **Diseño Multicapa.**

El diseño del sistema debe estar dividido en tres capas: Presentación, Dominio, y Persistencia de tal forma que facilite el desarrollo y mantenimiento futuro.

3.- ANTECEDENTES, ESTADO DE LA CUESTIÓN.

3.1.- ASPECTOS FUNDAMENTALES DE LA GESTIÓN DE CONTENIDOS.

3.1.1.- Definiciones.

Algunos de los términos relacionados con la gestión de contenidos son:

- **Dato:** Cantidad mínima de información no elaborada, sin sentido por sí misma.
- **Información:** Conjunto organizado de datos, que constituyen un mensaje sobre un determinado ente o fenómeno.
- **Conocimiento:** conjunto organizado de datos e información destinados a resolver un determinado problema.
- **Gestión del conocimiento, Gestión de la Información y Gestión de Contenidos:**

El motivo de ubicar juntos estos tres términos es porque están altamente relacionados. Muchas empresas por motivos de marketing han vendido una misma herramienta con pequeñas modificaciones como herramientas de gestión de conocimiento, posteriormente como herramientas de gestión de la información y en los últimos años como herramientas de gestión de contenidos.

Como se ha comentado anteriormente, la información es un conjunto de datos que debidamente organizados tienen un sentido y transmiten un mensaje. La información es lo que podemos gestionar, físicamente se encuentra o bien en documentos o bien en bases de datos, por tanto ya se habla de contenidos y de gestión de contenidos.

En cuanto al conocimiento es algo más complejo ya que es necesario que la

información sea procesada por una persona, conocimiento que ha adquirido con la experiencia, pero estos conocimientos para que puedan ser gestionados se plasman en documentos o en bases de datos.

3.1.2.- Tendencias en la Gestión de Contenidos.

Muchas de las organizaciones aún sigue prefiriendo el papel como medio físico para almacenar la información.

La implantación de nuevas tecnologías en las organizaciones es un proceso complejo [VILL] debido a que las personas por naturaleza, son bastantes reacias al cambio en su forma de trabajar.

La tendencia en cuanto a la forma de almacenar la información es la aceptación del documento electrónico como forma válida de documento (la oficina sin papeles). Ha influido en gran medida la utilización generalizada del correo electrónico [ECM] como forma de comunicación en los negocios. La aceptación será aún mayor con la evolución y mejora de las **firmas digitales** en los documentos donde se permita demostrar la autenticidad de los mismos.

La utilización de la tecnología como herramienta para la gestión permite mayor velocidad de trabajo con un índice de error mínimo, a lo que sumamos las redes de comunicaciones que nos dan la posibilidad de acceder a la información desde cualquier situación geográfica.

Un objetivo primordial es la **accesibilidad**, empleando información obtenida de documentos no estructurados. Actualmente existe una dificultad añadida en la gestión de contenidos con la creación de documentos dinámicos en la red, el gestor de contenidos debe tener en cuenta estas modificaciones para que la información este siempre actualizada.

3.1.3.- Características deseables en un Gestor de Contenidos.

Actualmente, los gestores de contenidos deben cumplir una serie de condiciones mínimas para tener un cierto grado de calidad.

- Disponibilidad de la información. Tener acceso a ella siempre que sea necesario.
- Poder obtener la información de una forma selectiva (utilización de filtros).
- El tiempo de respuesta del sistema no debe ser alto.
- La información puede ser presentada de diferentes formas y en distintos medios.
- Es deseable una respuesta “inteligente” por parte del gestor a la hora de seleccionar la información. No debe mostrar información que el usuario no esté buscando.
- Facilidad de uso (usabilidad).

3.1.4.- Tareas que permite un Gestor de Contenidos.

■ **Presentación de contenido dinámico**

El contenido y la apariencia de un sitio web puede depender de los datos existentes en una determinada base de datos. La modificación de estos datos con inserciones y actualizaciones por parte de usuarios con diferentes roles, evitan al personal encargado de mantener el sitio tener que modificar los archivos almacenados en el servidor.

Cuando se presenta contenido de tipo dinámico se suele ofrecer la posibilidad de personalización, indexación y búsqueda sobre éste.

■ **Administración del sitio**

Administrar un sitio web no muy amplio es algo sencillo, pero aquellos en los que se disponga de grandes volúmenes de información y varios usuarios con acceso a los datos, es más complejo.

El gestor de contenidos nos proporciona un lugar común de trabajo donde varios usuarios pueden trabajar de forma colaborativa.

■ **Manipulación de contenidos**

Permite a un usuario no experto participar en la creación de contenidos. No es necesario que conozcan algún lenguaje de programación para poder ampliar la base de contenidos, disponen de editores WYSIWYG similares a cualquier procesador de texto con los que pueden crear nuevos contenidos.

Es por tanto una de las principales características de este tipo de framework. Pueden colaborar usuarios expertos, conocedores de lenguajes de programación ampliando la funcionalidad del sistema, y usuarios expertos en otras áreas, como por ejemplo historiadores, filólogos... que pueden colaborar en la gestión de contenidos.

■ **Sistema de Seguridad**

Se encarga de la gestión de diferentes roles de usuarios con diferentes permisos [SHU03]. Posibilidad de distribución de tareas entre los diferentes tipos de usuario y acceso restringido a zonas del sistema. Por ejemplo, la parte de diseño y utilización de plantillas se puede limitar únicamente a los diseñadores, la lógica de negocio a los programadores, y la gestión de contenido a los administradores de bases de datos.

■ **Servicios de Red**

Los servicios de red proporcionan la posibilidad de aceptar peticiones de otros programas. Por ejemplo, compartir noticias entre diferentes páginas o la utilización de un buscador común.

■ **Escalabilidad**

El sistema se adapta a la demanda de los visitantes del sitio.

3.2.- CASOS DE ESTUDIO

Los siguientes casos de estudio tratan de diferentes Framework que permiten el desarrollo de aplicaciones web y la gestión de contenidos. El estudio de estos, junto con los aspectos fundamentales de los gestores de contenidos, han servido para conocer la filosofía y forma de trabajo de los gestores de contenidos.

3.2.1.- Zope, servidor de aplicaciones web.

Zope posee una serie de características [ZOP04] que hacen de él un servidor de aplicaciones web muy potente.

3.2.1.1.- Características propias de Zope.

- Zope es gratuito y distribuido bajo licencia de software libre.
- Proporciona un marco de trabajo propio.
- Es de fácil instalación.
- Dispone de numerosos servicios y módulos para el desarrollo de aplicaciones. La mayoría de estos componentes son gratuitos y de código abierto.
- Cuenta con el apoyo de la comunidad de desarrolladores, que contribuyen en su crecimiento.
- Escalabilidad lineal de las aplicaciones sin necesidad de modificación de código, con ZEO (Zope Enterprise Objects) que permite que las aplicaciones puedan estar distribuidas en diversas computadoras.
- Es altamente orientado a objetos.
- Posee una base de datos propia orientada a objetos, llamada ZODB.
- Permite la conectividad con bases de datos relacionales.
- Es multiplataforma, tan sólo es necesario la utilización de un navegador para poder trabajar en el marco de trabajo que proporciona Zope.

- Permite el trabajo colaborativo. Posee un sistema de control de versiones que permite trabajar de forma segura pudiendo recuperarse de errores.
- Posibilidad de crear contenidos dinámicos con DTML y Python.
- Zope no es una herramienta de desarrollo visual, como pueda ser Macromedia DreamWeaver o FrontPage.

3.2.1.2.- Conceptos fundamentales de Zope.

Zope posee su propio **marco de trabajo** (Framework) independiente, no necesita de ninguna herramienta externa para poder desarrollar aplicaciones web.

Se basa en la utilización de lenguajes como Python o Perl para la capa de dominio; para la parte de diseño y plantillas existen dos soluciones, una basada en XML, ZPT (Zope Page Template) y otra basada en HTML, DTML (Document Template Markup Language).

El panel de administración de Zope es conocido como ZMI (Zope Management Interface) con el que se puede acceder a los objetos del sistema, crear usuarios, roles, permisos y todas las actividades necesarias para la administración del sistema, sin necesidad de acceder al sistema de archivos ubicados en el servidor.

Orientación y publicación de objetos. Zope permite desarrollar sitios web dinámicos. Es altamente orientado a objetos (todo elemento de Zope es considerado como un objeto).

La estructura de objetos de Zope está jerarquizada, un objeto puede contener a otros objetos. Para acceder con el navegador a un objeto B que esta contenido dentro de un objeto A, tendremos que utilizar la URL /A/B.

Seguridad. Actualmente, en el desarrollo de sitios web con un tamaño considerable, participan varias personas y todas ellas especializadas en determinadas áreas. Zope fue diseñado pensando en este factor, por lo que es posible limitar el acceso a

diferentes zonas del sistema, según sea el rol del usuario.

Los diseñadores podrán dedicarse exclusivamente a la creación y utilización de plantillas, los programadores a la capa lógica, y los administradores de bases de datos a la utilización de herramientas y módulos destinados a la gestión de bases de datos.

ZODB, Zope Object Database. Todos los objetos de Zope están almacenados en la ZODB. Se trata de una base de datos transaccional orientada a objetos de alto rendimiento. Cada una de las peticiones realizadas por diferentes usuarios, es tratada por la base de datos como una transacción individual.

Es consistente debido a que si algún error se produce durante el tratamiento de la transacción, el sistema deshace los cambios realizados, quedando el estado de la base de datos como se encontraba originalmente antes de la petición.

Adquisición. La adquisición es una de las características más importantes y más potentes que posee Zope.

- Los objetos están dentro de otros objetos (jerarquía).
- Los objetos pueden heredar los atributos y los comportamientos de los objetos que están por encima en la jerarquía.

Extensibilidad. Existe la posibilidad de crear nuevos productos y añadirlos al servidor Zope. Existe un catálogo con más de 500 productos, disponible en la página oficial de Zope [ZOP]

3.2.1.3.- Arquitectura de Zope

A continuación se pasará a detallar los elementos más importantes de la arquitectura de Zope.

- **Zserver:** Servidor web propio de Zope encargado de servir el contenido a los usuarios. El contenido es accesible vía HTTP, FTP, WebDAV y XML-RPC.

- **Servidor web externo:** Zope contempla la posibilidad de poder utilizar otros servidores distintos a Zserver, como puede ser Apache o Microsoft IIS.
- **Núcleo de Zope.** Es el motor del sistema, se encarga de dar servicio a la capa funcional y la de gestión de la base de datos.
- **Bases de datos.** Base de datos propia (ZODB) y conectividad con bases de datos relacionales externas.
- **Sistema de archivos.** Zope posee su propio sistema de archivos al que puede ser accedido vía web a través de la ZMI o de una aplicación FTP.
- **ZClasses y productos:** Una de las características fundamentales de Zope es su extensibilidad. Podemos ampliar Zope añadiendo nuevos tipos de objetos con ZClasses o bien nuevos productos con clases de Python.

3.2.2.- Typo3

Typo3 es un sistema de Gestión de Contenidos (CMS) [TYP04] de código abierto y desarrollado bajo licencia GPL.

Está implementado en PHP y los datos se almacenan en bases de datos MySQL. Al igual que en Zope y la mayoría de CMS desde el punto de vista de la gestión de contenidos diferencia dos zonas: **Backend** y **Frontend**.

Desde la parte de Backend es posible gestionar todo el contenido del sitio web, con creación de nuevos contenidos, modificaciones y eliminación de los mismos.

El Frontend corresponde a la parte pública o visible. Todo el trabajo de gestión de contenidos se llevará a cabo desde el Backend, y los resultados finales se mostrarán desde el Frontend.

Es un sistema multiplataforma, tan sólo es necesario disponer de un navegador web. Al igual que en Zope todo está integrado en un único marco de trabajo.

Es extensible, existen multitud de módulos como calendarios, agendas, foros...

fáciles de añadir al sistema.

Incluye un motor de búsqueda indexada que permite la búsqueda de texto en las páginas del sitio web y dentro de archivos externos de tipo TXT, HTML, MS WORD o PDF.

Permite el trabajo colaborativo entre los usuarios del sistema, existen diferentes flujos de trabajo dependiendo del rol del usuario. Posee un sistema de control de versiones a través del cual es posible recuperar versiones anteriores de cada uno de los elementos del sistema, lo que hace de Typo3 un sistema seguro .

Existen tres clases de usuarios:

- **Editores de Contenido:** Los editores de contenidos no tienen por que ser usuarios expertos de sistemas, ni conocer ningún lenguaje de programación o estar familiarizado con actividades de desarrollo. Por ello Typo3 proporciona editores WYSIWYG para la edición sencilla de contenido.
- **Administradores:** Son los encargados de la gestión del sitio web. Permite la creación de usuarios de diferentes tipos, con sus correspondientes permisos sobre el sistema. Los permisos posibles sobre los elementos del sistema son cinco: mostrar página, editar contenido, editar página, borrar página y creación de nuevas páginas.
- **Desarrolladores:** Typo3 posee una serie de propiedades que lo hace especial e interesante para los desarrolladores:
 - Meta-lenguaje TypoScript. Lenguaje de plantillas y configuración para el diseño, propio de Typo3. Permite la separación de capas en el desarrollo de aplicaciones.
 - Plantillas en cascada. Las plantillas de diseño realizadas con TypoScript pueden ser ordenadas jerárquicamente.
 - Plantillas HTML. Existen plantillas con la estructura de documentos

comunes como la presentación de noticias o tableros de anuncios que pueden ahorrar tiempo de desarrollo.

- Gestión de extensiones. Existe en la red [TYP] un repositorio con gran cantidad de componentes para la extensión de Typo3, gestionable desde el propio sistema.
- Scripts de PHP. Reutilización de fragmentos de código que aportan funcionalidad a las aplicaciones.
- Centro de pruebas para la validación y verificación del nuevo código fuente.
- Integración multimedia. Posibilidad de incorporación de audio, vídeo y tecnología Flash a las aplicaciones.
- Motores de búsqueda. Permiten realizar búsquedas de forma cómoda sobre los contenidos del sitio web, tanto los que se encuentran almacenados en la base de datos, como los estáticos pertenecientes a un documento.

3.2.3.- Content Management Server de Microsoft

Se trata de un gestor de contenidos cuya utilización se ha visto impulsada con la tecnología .NET y que es utilizado para la administración y gestión de sitios web de gran envergadura.

Las características que posee, son muy similares a las mencionadas para los CMS de carácter libre citados anteriormente en el documento.

- Posee un entorno común de trabajo donde los propios usuarios pueden crear, publicar y administrar el contenido.
- Está integrado con otras herramientas de Microsoft como Microsoft Visual Studio .NET y Microsoft .NET Enterprise Servers.
- Proporciona contenidos en formato XML y permite la personalización de servicios Web XML.

- Preparado con herramientas y componentes para el desarrollo de sitios dedicados al comercio electrónico.
- Alta seguridad, sistema de roles de usuario y permisos sobre los elementos del sistema.
- Gestión de versiones.
- Permite el trabajo colaborativo.
- Posibilidad de utilización de plantillas para la separación de la parte de diseño y lógica de negocio.
- Posee dos herramientas fundamentales para la edición de contenidos CMS:
 - Web Author Client. Permite crear, editar y publicar contenido directamente con la utilización de un navegador.
 - Authoring Connector for Microsoft Office. Permite crear y publicar contenido directamente desde Microsoft Office.
- Persistencia de la información. Almacena el contenido en XML, HTML y objetos de contenido binario para tener la máxima flexibilidad posible. Los objetos son almacenados en un repositorio de Microsoft SQL Server.
- APIs administradas. La funcionalidad y los depósitos de contenido de Content Management Server están disponibles a través de interfaces de programación de aplicaciones COM+ administradas (APIs).
- ASP y ASP.NET utilizados como lenguajes de programación.

3.2.4.- Comparativa entre Zope y Typo3.

1. Tanto Typo3 como Zope son servidores de aplicaciones que ofrecen un espacio común para el desarrollo de aplicaciones web y la gestión de contenidos.
2. Son totalmente independientes, no necesitan de ninguna herramienta externa para poder trabajar.
3. Permiten el trabajo colaborativo y el control de versiones.
4. Recuperación de los errores en ambos casos.
5. Tan sólo es necesario la utilización de un navegador para poder trabajar.
6. Typo3 incorpora editores WYSIWYG que facilitan la inserción de nuevos contenidos, se asemejan a los procesadores de textos con los que el usuario está acostumbrado a trabajar. El gran inconveniente es que no funcionan en todos los navegadores.
7. En ambos casos se permite el trabajo con plantillas, por lo que es posible trabajar la parte de diseño por separado.
8. Tanto Zope como Typo3 disponen de gran cantidad de componentes, realizados por diferentes usuarios, con los que es posible ampliar el sistema.
9. Sistema de roles y gestión de permisos sobre los elementos del sistema.
10. En ambos casos se encuentra disponible el código fuente por lo que es posible modificarlo y ajustarlo a nuestras necesidades (reutilización).
11. Zope posee su propio sistema de archivos, mientras que Typo3 lo simula con la información almacenada en la base de datos.
12. Zope es altamente orientado a objetos, mientras que Typo3 no lo es. En Zope

todo elemento es un objeto.

13. Zope posee su propia base de datos orientada a objetos, ZODB y permite la conexión con bases de datos relacionales. Typo3 trabaja con bases de datos MySQL.
14. Zope posee una propiedad fundamental gracias a su filosofía orientada a objetos, que Typo3 no posee, se trata de la adquisición donde los objetos pueden utilizar propiedades y funciones de los objetos que los contienen.
15. Zope esta preparado para un acceso masivo de usuarios. Puede estar instalado en varias máquinas y distribuir la carga de trabajo.

3.3.- USABILIDAD WEB

La organización internacional para la estandarización posee varias definiciones para la usabilidad.

“La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso” ISO / IEC 9126

“Usabilidad es la eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico” ISO / IEC 9241.

“Cuando se simplifica la aplicación y se concentra en las principales opciones, eliminando u ocultando las opciones secundarias se facilita el aprendizaje de cualquier persona”. Jakob Nielsen [NIE02]

“Web usables son aquellas que el usuario puede “entender” (qué es y cómo usarla) sin demasiados esfuerzos ...

En Internet la competencia es sólo cuestión de hacer clic aquí o allá, de manera que si frustra a los usuarios no tardará en dirigirse a otro lado”. Steve Krug [KRU04].

La usabilidad afecta directamente a la calidad del software. Es un aspecto fundamental de cualquier sistema para que tenga éxito y sea finalmente utilizado. Existe gran cantidad de aplicaciones con una funcionalidad excelente, que dejan de ser utilizadas por no tener un grado de usabilidad alto. Cuando un sistema no es usable produce la frustración del usuario.

A partir de este momento nos centraremos en la usabilidad web y los estudios realizados por Jakob Nielsen y Steve Krug.

Según Steve Krug [KRU04], un usuario cuando accede a un sitio web debe contestar de forma sencilla a las siguientes preguntas:

- ¿ Qué sitio es este ?
- ¿ En qué página estoy ?
- ¿ Cuáles son las principales secciones del sitio ?
- ¿ Qué secciones tengo en este nivel?
- ¿ Dónde estoy en el esquema de las cosas ?
- ¿ Cómo busco algo?

A continuación se pasará a describir algunos aspectos fundamentales de la usabilidad web según Jakob Nielsen y Steve Krug.

- Dejar bien claro sobre donde se puede hacer clic. La mayoría de páginas web contienen diferentes elementos sobre los que el usuario puede interactuar. Debe quedar bien claro cuales son esos elementos.
- Saber de que trata el sitio a simple vista. Esto puede ayudar a conseguir el interés del usuario. La utilización de algún logo que represente la finalidad del sitio, o explicar los puntos importantes del sistema en la página principal es

fundamental.

- División de las páginas en zonas claramente definidas. Debe quedar bien claro cada una de las partes. Según Steve Krug los usuarios no leen las páginas sino que las hojean por los motivos siguientes:
 - Normalmente tenemos prisa: el usuario desea encontrar lo que busca en el menor tiempo posible.
 - Sabemos que no hay que leer todo el contenido de un página para encontrar lo que buscamos.
 - Sabemos que funciona. Las personas están acostumbradas a hojear los periódicos, revistas y los libros para dar con las partes que le interesa.



Figura 3.1 Contenido del sitio web

Veamos un ejemplo de lo que contiene realmente una página y lo que ve el usuario.

Y esta es la visión del usuario a primera vista.



Figura 3.2 Visión del usuario del sitio web a primera vista

- Ayuda contextual en aquellos iconos sobre los que es posible interactuar. Las imágenes utilizadas como botones deben estar relacionadas con la acción que se realizará al hacer clic sobre ellas, además es conveniente añadirle un título indicativo que se muestre cuando se sitúe el puntero del ratón sobre ella.
- Utilizar colores y tamaño de fuente legibles.
- Evitar sobrecarga de información. No es conveniente cargar una página con demasiada información innecesaria. El usuario visita un determinado sitio web con unos objetivos claros. La sobrecarga de información puede dificultar al usuario a la hora de encontrar lo que buscaba y centrarse en otros objetivos que se alejan de los iniciales.
- El usuario debe sentirse localizado en todo momento dentro del sistema. Siempre debe saber de donde viene, donde se encuentra y que es lo que puede hacer. Una buena forma de conseguirlo es la utilización de migas de pan. Las migas de pan

indican las secciones por donde ha ido pasando el usuario y donde se encuentra actualmente, pudiendo acceder de nuevo a ellas con un simple clic de ratón.

» Home » Mis contenidos » Documento » Ejercicios prácticos » Crear Ejercicios prácticos

Figura 3.3 Ejemplo de migas de pan

- Eliminar palabras innecesarias [KRU04]:
 - Reduce el nivel de ruido en la página.
 - Realza el contenido verdaderamente práctico.
 - Se acortan las páginas permitiendo ver al usuario más de cada una con sólo un vistazo y sin avanzar o retroceder.

- Creación de una jerarquía visual clara. Relación clara de los elementos que compongan la página, destacando de forma más prominente los elementos más importantes en la jerarquía.

- Utilización de pestañas. Según Steve Krug las pestañas son una buena forma de clasificar los contenidos:
 - Utilizan la metáfora de las etiquetas separadoras de libros, por lo que la mayoría de usuarios las saben utilizar.
 - Son claras y fáciles de entender.
 - Difíciles de perder de vista.

- Buscador de contenidos. Incluir un buscador de contenidos en las páginas, facilita al usuario la búsqueda de información y el acceso rápido a esta.

3.4.- DESARROLLO DE WEB DINÁMICO.

Cuando empezaba a utilizarse Internet como medio de comunicación, tan sólo era posible publicar documentos estáticos. Para variar su contenido era necesario modificar el documento con un editor y volver a subirlo al servidor.

Actualmente es posible desarrollar aplicaciones web con contenido dinámico, adquiriéndolo de bases de datos o archivos almacenados en el servidor. Este factor permite una mayor interacción por parte del usuario.

Son numerosos los lenguajes de programación que existen para desarrollar aplicaciones web dinámicas. Entre los más populares se encuentran PHP, ASP, ASP.NET y JSP. Todos ellos tienen en común que son lenguajes del lado del servidor, es decir, el código es interpretado en el servidor, quien envía un documento estático con código HTML al usuario.

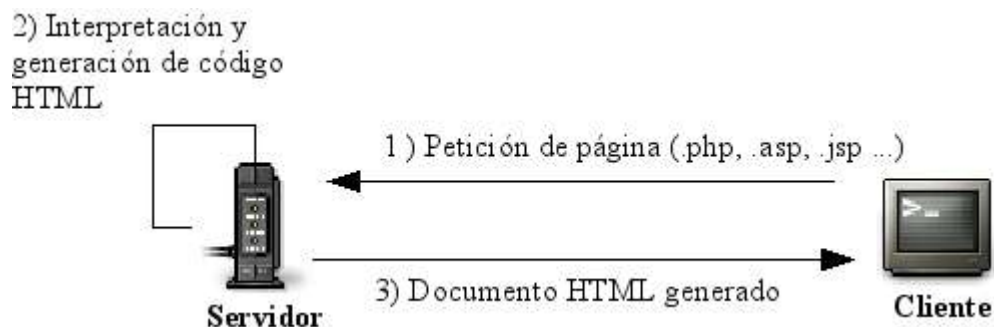


Figura 3.4 Lenguajes de programación del lado del servidor

Por otra parte existen los lenguajes del lado del cliente, como es el caso de JavaScript. Los scripts son ejecutados en el navegador del usuario sin la necesidad de la intervención del servidor. La mayoría de aplicaciones web, combinan estos dos tipos de lenguajes.

3.4.1.- PHP

“PHP es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl

con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas” [PHP]

Inicialmente fue desarrollado para la creación de aplicaciones web exclusivamente. Actualmente es posible desarrollar aplicaciones de escritorio con PHP-GTK.

Fue creado en 1994 por Rasmus Lerdorf. El crecimiento del uso de PHP por parte de la comunidad de programadores ha sido exponencial desde su creación. En el año 2002 ya era utilizado por más de nueve millones de programadores de todo el mundo, en la actualidad son más de veinte millones.

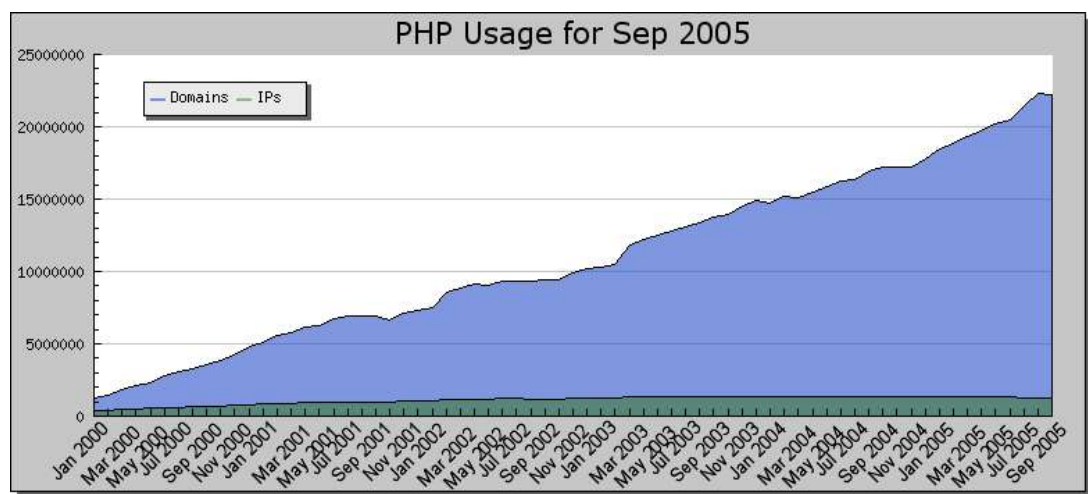


Figura 3.5 Evolución del número de programadores que utilizan PHP

Las siglas de PHP, inicialmente significaban Personal Home Page (página de inicio personal), pero se cambiaría posteriormente de acuerdo con la convención de designación de GNU pasando a significar, PHP Hipertext Preprocessor.

Algunas **características** de PHP [THO03] son:

- Multiplataforma.
- Ofrece un alto rendimiento.
- Conectividad con un número amplio de sistemas de bases de datos: InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras.
- Es un lenguaje de carácter libre. Al ser un producto de código abierto, **PHP** goza

de la colaboración de un número amplio de programadores, que encuentran y modifican rápidamente los errores.

- Sintaxis sencilla, lo que proporciona gran facilidad a la hora de aprender el lenguaje y su posterior utilización.
- Portabilidad. El código escrito en PHP en una máquina y en un sistema operativo es válido en cualquier otro sistema operativo y en cualquier otra máquina. No es necesario reescribir el código.
- Gran cantidad de librerías disponibles. Muchas de estas bibliotecas son proporcionadas por desarrolladores sin ánimo de lucro.

3.4.2.- PHP 5

PHP 5 es la evolución de PHP, se trata de la última versión de este lenguaje de programación. La principal característica es que es orientado a objetos, esto supone todas las grandes ventajas del paradigma orientado a objetos: herencia, polimorfismo, facilidad de reutilización del software ...

Novedades de PHP 5:

- Herencia.
- Polimorfismo.
- Clases abstractas.
- Interfaces.
- Objetos por referencia.
- Clonación de objetos.
- Identificadores de visibilidad: public, private, y protected.
- Tratamiento de excepciones.

3.4.3.- ASP y ASP.NET

Active Server Pages (ASP), es un lenguaje de script de Microsoft del lado del servidor que permite desarrollar páginas web dinámicas.

ASP ha pasado por cuatro iteraciones mayores, ASP 1.0 (distribuido con IIS 3.0),

ASP 2.0 (distribuido con IIS 4.0), ASP 3.0 (distribuido con IIS 5.0) y ASP.NET (parte de la plataforma .NET de Microsoft). Las versiones pre-.NET se denominan actualmente (desde 2002) como ASP *clásico*. [WIK].

Los servidores que soportan este lenguaje son de la familia de Windows NT: Personal Web Server (PWS) e Internet Information Server (IIS).

ASP.NET es la nueva versión de ASP disponible con la plataforma .NET de Microsoft. La arquitectura ha sido rehecha desde cero para facilitar al máximo la creación de aplicaciones web dinámicas, promoviendo una mejor reutilización y compartición del código.

Una de las principales características de esta nueva versión de ASP es el soporte de múltiples lenguajes compilados (Visual Basic, C++ o C# entre otros). La primera vez que se solicita un página es compilada y almacenada en memoria caché, de tal forma que las siguientes peticiones serán atendidas de forma más rápida, esto produce un incremento del rendimiento.

Al poder emplear diferentes lenguajes, como C++, C#, Visual Basic .NET es posible el desarrollo de aplicaciones bajo el paradigma de la orientación a objetos y por tanto la posibilidad de realizar un diseño multicapa.

3.4.4.- JSP

Java Server Pages (JSP) es otro de los lenguajes de programación para la generación de sitios web dinámicos desarrollado por Sun Microsystems.

Existe la posibilidad de integrar JSP con clases Java, lo que permite separar en diferentes capas las aplicaciones, teniendo desarrollado la capa de dominio en lenguaje Java puro y presentar los datos vía web con JSP.

Además tiene la gran ventaja de ser multiplataforma, pudiendo funcionar sobre diferentes sistemas operativos, permitiendo la portabilidad de las aplicaciones desarrolladas y posee soporte para la conectividad con diferentes gestores de bases de datos..

Los archivos JSP son muy similares a los servlets: es compilado la primera vez que es invocado y comienza a ejecutarse en el servidor como un servlet, proporcionando un alto rendimiento.

El coste de desarrollo de aplicaciones JSP es bajo. La tecnología Java puede ser adquirida de forma gratuita de la página oficial de Sun Microsystems, si necesidad de la adquisición de licencias para el desarrollo.

Como en cualquier otra aplicación web, deben implantarse en un servidor de aplicaciones. Tomcat es uno de los servidores más utilizados para aplicaciones JSP, incluye el compilador Jasper, que compila los JSPs convirtiéndolos en servlets. Tomcat está escrito en Java, por lo que funciona en cualquier sistema operativo que disponga de la máquina virtual de Java instalada.

Tomcat está desarrollado y mantenido por los miembros de la Apache Software Foundation y voluntarios independientes. Cualquier usuario puede acceder de forma libre al código fuente bajo los términos de la Apache Software Foundation. Las versiones más recientes son las 5.x, que implementan las especificaciones de Servlet 2.4 y de JSP 2.0. En las versiones 4.x, Jakarta Tomcat utiliza el contenedor de servlets.

3.5.- CSS

CSS (Cascading Style Sheets), hojas de estilo en cascada, es un lenguaje formal definido como estándar [CSS03] para añadir información sobre el estilo de documentos HTML y XML. Existen dos versiones publicadas en W3C, CSS1 publicada en 1996 y CSS2 publicada en 1998.

CSS1 está soportado por la mayoría de navegadores. CSS2 aporta nuevas funcionalidades con respecto a la primera versión y soporte para documentos XML, no todos los navegadores están preparados para su procesamiento.

Las hojas de estilo son utilizadas por los diseñadores para definir el aspecto de la capa de presentación. Es posible definir la estructura del documento con etiquetas HTML y el aspecto visual con las hojas de estilo. La utilización de hojas de estilo simplifica el código HTML de tal forma que hace más fácil la búsqueda de errores y el depurado de las páginas.

Para una misma aplicación web es posible utilizar diferentes hojas de estilo, adaptando la presentación a las necesidades del usuario. Por ejemplo para personas con deficiencias visuales, es posible crear hojas de estilo donde los elementos posean un tamaño mayor, mejorando de esta forma la accesibilidad.

Las hojas de estilo están definidas mediante una serie de reglas en un fichero de texto plano. Cada una de estas reglas consta del tipo de elementos sobre el cual se desea definir un estilo y un conjunto de atributos con sus valores correspondientes. Los elementos sobre los que se pueden crear estilos, son aquellos que pertenecen a cualquier estructura de un documento: tablas, fuentes, capas, fondo del documento...

Es posible definir un estilo sobre un tipo de elemento de forma general, o bien crear varias clases de un mismo tipo de elemento. Una de las grandes ventajas que proporcionan las clases de estilos, es la herencia. Un elemento que se encuentre contenido en otro, puede heredar propiedades de estilo.

Un ejemplo de definición general para un elemento sería:

```
table{  
  color:black;  
  font-size:12px;  
}
```

Todas las tablas definidas en el documento HTML tendrían un borde de color negro y un tamaño de fuente de doce pixels. Para definir una clase de un tipo de elemento se especificaría el tipo de elemento seguido de un punto y el nombre de la clase:

```
table.central{
```

```
border:1px solid #cccccc;
font-size:12px;
padding: 0px 0px 0px 0px;
background-color:#ffffff;
width:85%;
margin-left:auto;
margin-right:auto;
}
```

Para que una tabla tome las propiedades definidas en la clase “central” se definiría como: <table class= “central”>. En este caso la tabla tomaría el estilo de la clase central en lugar del definido para todas las tablas de forma general.

3.6.- ENSEÑANZA Y APRENDIZAJE DE IDIOMAS

Aprender idiomas es muy importante por las ventajas que proporciona tanto a nivel personal como profesional. Es necesario para la comunicación e integración con las personas de diferentes países.

Muchos autores coinciden en el siguiente punto: Para aprender correctamente un nuevo idioma, no es suficiente con su estudio sino que además es necesario estudiar la cultura del lugar donde se habla.

Coinciden también en el gran esfuerzo, voluntad y tiempo que debe emplear una persona para el aprendizaje del idioma.

3.6.1.- Métodos utilizados

Jack C. Richards [RIC98] y Aquilino Sánchez Pérez [AQU82] estudian diferentes métodos de enseñanza de idiomas en sus obras “Enfoques y métodos en la enseñanza de idiomas” y “La enseñanza de idiomas. Principios, problemas y métodos”. A continuación se describirá brevemente en que consiste cada uno de ellos.

3.6.1.1.- Método gramática – traducción

Los puntos de esta metodología [AQU82] podrían resumirse en:

- *“Predominio de un modelo de lenguaje derivado del legado escrito de autores consagrados. Estos se constituyen en maestros y en norma. Los hablantes se fundamentan en última instancia o en caso de duda en el 'argumentum auctoritatis'”.*
- *“Convencimiento y creencia en el hecho de que los procesos lingüísticos son procesos fundamentalmente lógicos, adquiridos por deducción. En consecuencia se hace necesario aprender primero las reglas gramaticales como mecanismos indispensables, que configuran la producción lingüística. La aplicación de las reglas permitirá la formación de las frases y oraciones bien hechas, utilizando léxico variando en combinaciones distintas”.*
- *“El concepto de lengua oral o coloquial es sinónimo de 'vulgar', de bajo nivel o calidad. No solamente no se toma como modelo sino que incluso se debe evitar en el uso y rechazar como incorrecto”.*
- *“La memorización de las reglas gramaticales por un lado y de largas listas de vocabulario por otro, son los objetivos prioritarios en una lección del método tradicional”.*
- *“La técnica de traducción directa e inversa es la más ampliamente utilizada en clase”.*
- *“El vocabulario utilizado es el que mejor sirva a la aplicación de las reglas gramaticales, no necesariamente el que más se use en la comunicación interpersonal”.*

3.6.1.2.- Método directo

Consiste en el aprendizaje de idiomas mediante la práctica oral y la lectura. Las personas aprenden la lengua materna mediante este método.

Si en el método de gramática- traducción [RIC98], se hacía mayor hincapié en obtener una mayor precisión, en el método directo se persigue obtener una mayor fluidez lingüística.

Se caracteriza por la ausencia de gramáticas en los primeros niveles y ofrecer la lengua tal cual se habla en la vida diaria.

3.6.1.3.- Métodos audio-oral y audio-visual

El lingüista W. Moulton enunciaba los principales dogmas [AQU82] de esta metodología.

- *“La lengua es la lengua oral, no escrita”.*
- *“La lengua es el resultado de un conjunto de hábitos”.*
- *“Hay que enseñar la lengua, no algo sobre la lengua”.*
- *“La lengua es lo que hablan los habitantes nativos de la misma, no lo que alguien cree que debe ser”.*
- *“Las lenguas son diferentes”.*

En este método se aplican las nuevas tecnologías. Consiste principalmente en la repetición una y otra vez por parte del usuario, de aquello que ve y escucha. El objetivo es aprender un idioma mediante la consolidación de lo repetido.

Con la utilización de imágenes se pretende evitar la lengua materna, a través de la imagen se introduce directamente la lengua que se desea aprender.

3.6.1.4.- Método nocional-funcional

El método nocional-funcional, parte de la idea de que las unidades docentes utilizadas por cada persona para el aprendizaje del nuevo idioma, deben estar estructuradas de acuerdo a sus necesidades o finalidades.

Este método es estudiado en [AQU82] :

- *“El método nocional-funcional parte de la consideración de la lengua que se va a enseñar como un instrumento de comunicación interpersonal”.*
- *“La elaboración del material docente en un libro de texto, no toma como punto de partida una organización por temas gramaticales, ni léxicos de frecuencia, ni situaciones de la vida real, sino aquellas unidades de comunicación (actos de comunicación lingüística) que forman parte integrante de la comunicación de la vida real”.*
- *“Esas unidades del lenguaje se ordenan adecuadamente de acuerdo con las necesidades de comunicación de aquellos que aprenden un idioma. (En este sentido, se debe también tener en cuenta el aprendizaje de un idioma con finalidades especiales y concretas)”.*
- *“Dado que los actos de comunicación lingüística admiten varias formas o relaciones diferentes, se debe proceder también a una selección de las formas o estructuras lingüísticas, de acuerdo con el grado de dificultad o de complejidad de las mismas y siempre teniendo en cuenta las necesidades y objetivos de quien aprende”.*

3.6.2.- Ejercicios prácticos

Existen diferentes tipos de ejercicios prácticos [MLG] utilizados en la enseñanza de idiomas para la comprensión oral y escrita del alumno. A continuación se muestra un listado de los más comunes.

- De acuerdo a un texto leído o escuchado realizar una serie de preguntas.
 - Selección múltiple.
 - Completar espacios en blanco.
 - Respuestas del tipo verdadero / falso.

- Resumen oral o escrito de lo comprendido por el alumno.
- Ejercicios de relación. Hacer corresponder texto con imágenes o texto con texto.
- Seguir las instrucciones especificadas por un mensaje que se ha realizado de forma escrita u oral.
- Juegos: El juego de las adivinanzas, el Ahorcado, sopa de letras, crucigramas ...
- Completar frases y diálogos.
- A partir de elementos desordenados construir oraciones.
- Descripción de objetos y personas.
- Composición guiada o libre.

3.6.3.- Sitios en la red para el aprendizaje de idiomas

Antes del desarrollo de Babel se han estudiado las características principales que ofrecen otros lugares en Internet destinados al aprendizaje y enseñanza de idiomas.

- www.mansioningles.com
 - ◆ Únicamente contenido para aprender Inglés.
 - ◆ Numerosos ejercicios, cursos y documentos teóricos.
 - ◆ Los ejercicios no son corregidos. El sistema muestra la solución correcta y el usuario debe comparar.
 - ◆ Contenidos creados por los webmasters del sitio, no es posible la creación de contenidos por parte de los usuarios.
 - ◆ Diccionarios y herramientas de traducción externas.

- www.aulafacil.com
 - ◆ Documentos teóricos.

- ◆ Cursos.
 - ◆ Ejercicios.
 - ◆ Chat
 - ◆ Posibilidad de aprender varios idiomas.
 - ◆ El usuario no puede crear contenidos nuevos.
- **www.home.es**
 - ◆ Sitio oficial de Home English. Posibilidad de aprender varios idiomas: Inglés, Francés y Alemán. Posee plataforma e-learning de pago para alumnos matriculados.
- **www.auladiez.com**
 - ◆ Aprendizaje de castellano.
 - ◆ Ejercicios y documentos teóricos.
 - ◆ Cursos online de pago.
- **www.e-aprender-ingles.com**
 - ◆ Aprendizaje de inglés.
 - ◆ Ejercicios, vocabulario y gramática.
 - ◆ Posee implementado un agente inteligente con tecnología Microsoft Agent Technology que ayuda al usuario en su visita.
 - ◆ Posee herramientas propias interesantes como un conjugador de verbos y un convertidor a gerundio.
 - ◆ El usuario no puede crear contenidos nuevos.
- **www.worldspeaking.com**
 - ◆ Posibilidad de aprender inglés, francés y castellano.
 - ◆ Aprendizaje personal, comunicación con profesores por teléfono o Internet.
 - ◆ Plataforma e-learning que permite el aprendizaje interactivo, profesor virtual y herramientas de ayuda.
 - ◆ El usuario no puede crear contenidos nuevos.

- www.learn-english-today.com
 - ◆ Sólo inglés como idioma.
 - ◆ Numerosos juegos y ejercicios prácticos.
 - ◆ El usuario no puede crear contenidos nuevos.

- www.speakspeak.com
 - ◆ Sólo inglés.
 - ◆ Ejercicios prácticos.
 - ◆ Plataforma e-learning de pago.

- www.learnspanish.com
 - ◆ Aprendizaje del castellano. Es necesario pagar para tener acceso a los contenidos.

- www.usingenglish.com
 - ◆ Sólo inglés.
 - ◆ Ejercicios y documentos.
 - ◆ Acceso privado para alumnos.

- www.englishtown.com
 - ◆ Plataforma e-learning de pago para el aprendizaje de inglés.
 - ◆ Lecciones por e-mail.
 - ◆ Debates supervisados por un profesor.
 - ◆ Lecciones interactivas.

- www.curso-de-aleman.de
 - ◆ Destinado al aprendizaje de alemán.
 - ◆ Gramáticas.
 - ◆ Ejemplos de voz.
 - ◆ Chat.
 - ◆ El usuario no puede crear contenidos nuevos.

Tras el análisis de las características más importantes de algunos sitios web

dedicados a la enseñanza y aprendizaje de idiomas podemos extraer las siguientes conclusiones.

- Podemos clasificarlos en dos categorías: Aquellos que disponen de contenido gratuito y plataformas e-learning de pago que permiten el contacto con el alumno y el seguimiento de su evolución.
- Los idiomas que más predominan son inglés, francés, alemán y castellano, pero la mayoría de sitios se suele dedicar a uno de ellos en concreto.
- Los usuarios no pueden contribuir con sus conocimientos. Los ejercicios y documentos son los expuestos por los desarrolladores del sitio web. El usuario tan sólo puede consultarlos y realizar los ejercicios.

4.- MÉTODOS Y FASES DE TRABAJO

4.1.- INGENIERÍA DEL SOFTWARE

En el transcurso del proceso de creación de Babel ha influido el ciclo de vida elegido y la decisión de realizar un diseño multicapa.

4.1.1.- Ciclo de vida del Software

Según la definición dada en ISO 12207-1 el ciclo de vida del software es

“Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso”.

El modelo de ciclo de vida empleado en Babel, es el Modelo de Espiral [DOM04], propuesto por Boehm en 1988. En cada una de las iteraciones de la espiral, se plantean nuevos objetivos evaluando ciertas alternativas y posibles riesgos. Al finalizar cada una de ellas se obtiene un nuevo producto que debe cumplir los requisitos establecidos en un principio y funcionar correctamente.

Para el proyecto se han realizado diez iteraciones, en cada una de las cuales se han planteado una serie de objetivos, alternativas posibles, posibles riesgos y los resultados obtenidos:

■ **Iteración 1:**

◆ Objetivos:

(a) Visualización del sistema en dispositivos de diferentes tamaños.

◆ Alternativas:

(a) Utilización de hojas de estilo.

- (b) Documentos diferentes dependiendo del dispositivo en el que se visualiza el sistema.
 - ◆ Resultados:
 - (a) Creación de dos hojas de estilo. Una para ordenadores de sobremesa y otra para PDA.
 - (b) Independencia entre la estructura de los documentos y la forma de presentación.
 - ◆ Planes:
 - (a) Diseño de una interfaz multiidioma.
- **Iteración2:**
- ◆ Objetivos:
 - (a)Diseño y creación de una interfaz multiidioma.
 - ◆ Alternativas:
 - (a) Para cada uno de los idiomas un fichero donde se guardan las etiquetas en variable.
 - (b) Etiquetas almacenadas en tablas de la base de datos.
 - ◆ Riesgos:
 - (a)Si las etiquetas se almacenaran en la base de datos, cada página puede contener un número elevado de etiquetas, que supondrían demasiadas consultas a la base de datos.
 - ◆ Resultados:
 - (a) Creación de un fichero para cada uno de los idiomas con las etiquetas que se mostrarán en la interfaz.
 - ◆ Planes:
 - (a) Soporte del sistema para la creación de usuarios.
 - (b) Roles de usuario.
 - (c) Permisos de usuarios
 - (d) Módulo de administración.

■ **Iteración 3:**

◆ Objetivos:

- (a) Soporte del sistema para la creación de usuarios.
- (b) Roles de usuario.
- (c) Permisos de usuarios
- (d) Módulo de administración.

◆ Resultados:

- (a) Implementación del módulo del administración para la gestión del sistema.
- (b) Sistema de roles y permisos de usuario con los que es posible determinar las acciones posibles de cada usuario en el sistema.

◆ Planes:

- (a) Implementación de la máquina de estados.

■ **Iteración 4:** Máquina de estados.

◆ Objetivos:

- (a) Implementación de la máquina de estados.

◆ Resultados:

- (a) Creación de una máquina de estados que determina el comportamiento del sistema hacia los contenidos. La máquina de estados esta formada por: Los roles y acciones que puede realizar un usuario, estados en los que se puede encontrar un contenido, acciones posibles sobre un contenido y las transiciones de estado.

◆ Planes:

- (a) Implementación de diferentes tipos de contenidos y gestión de contenidos.

■ **Iteración 5:**

◆ Objetivos:

- (a) Implementación de los diferentes tipos de contenidos.
- (b) Administración de contenidos.

- ◆ Resultados:
 - (a) Implementación de los tipos de contenido: documentos, ejercicios prácticos, traducciones , correcciones sobre traducciones y exámenes.
 - (b) Panel de administración de contenidos desde donde es posible la configuración de la máquina de estados.
- ◆ Planes:
 - (a) Diseño e implementación de un buscador de contenidos.

■ **Iteración 6:**

- ◆ Objetivos:
 - (a) Diseño e implementación de un buscador de contenidos.
- ◆ Alternativas:
 - (a) Cada vez que el usuario busque, realizar búsquedas sobre los contenidos de la base de datos directamente.
 - (b) Procesamiento de los contenidos y extracción de índices para realizar las búsquedas.
- ◆ Riesgos:
 - (a) Si las búsquedas se realizan directamente sobre los contenidos de la base de datos, cuando la base de contenidos posea un volumen considerable, las búsquedas pueden ser demasiado lentas.
- ◆ Resultados:
 - (a) Procesamiento de los contenidos para la generación de índices.
 - (b) Estructura de datos donde se almacenan las palabras y los códigos de los contenidos donde aparece la palabra.
 - (c) Cuando el usuario realiza la búsqueda consulta las tablas de índices.
- ◆ Planes:
 - (a) Diseño e implementación de diccionarios.

■ **Iteración 7:**

◆ Objetivos:

- (a) Diseño e implementación de un buscador de contenidos.
- (b) Utilización de diccionarios estándar en formato de texto plano.

◆ Alternativas:

- (a) Búsqueda directa sobre los ficheros de texto plano.
- (b) Extracción de palabras de los ficheros de texto plano para almacenarlo en una estructura de tablas en la base de datos.

◆ Riesgos:

- (a) Búsqueda secuencial sobre los ficheros, puede provocar que la búsqueda sea demasiado lenta.

◆ Resultados:

- (a) Extracción de palabras y definiciones a partir de los diccionarios que se encuentran en ficheros de texto plano.
- (b) Almacenamiento de las palabras y definiciones en la base de datos.
- (c) Búsqueda de términos sobre la base de datos.

◆ Planes:

- (a) Síntesis de voz.

■ **Iteración 8:**

◆ Objetivos:

- (a) Posibilidad de que los usuarios puedan escuchar la pronunciación de las palabras.
- (b) Realización de síntesis de voz

◆ Resultados:

- (a) Utilización de la aplicación Festival en el servidor para la generación de archivos WAV con la pronunciación de las palabras.

◆ Planes:

- (a) Implementación del módulo de comunicación.

■ **Iteración 9:**

◆ Objetivos:

- (a) Posibilidad de enviar mensajes privados entre los usuarios del sistema.
- (b) Creación de un chat que permita la comunicación entre usuarios que hablen diferentes idiomas
- (c) Posibilidad de practicar conversación hablada.
- (d) Debe ser posible la comunicación desde diferentes navegadores y de una forma sencilla para el usuario.

◆ Alternativas:

- (a) Utilización de tecnología de Macromedia.
- (b) Applets y librerías JMF de Java.
- (c) Utilización de PHP, JavaScript y Skype.

◆ Riesgos:

- (a) Costes elevados de los componentes Macromedia.
- (b) Fuertes políticas de seguridad de las Applets.
- (c) La instalación de plugins en el navegador para el funcionamiento del chat puede ser difícil para algunos usuarios.

◆ Resultados:

- (a) Creación de un buzón privado de mensajes para cada usuario.
- (b) Cada usuario puede enviar mensajes al resto de usuarios del sistema (comunicación asíncrona).
- (c) Implementación del chat con PHP y JavaScript, sin necesidad de la instalación de ningún plugin adicional en el navegador.
- (d) Diccionarios y posibilidad de realizar correcciones sobre los mensajes escritos por los usuarios desde el chat.
- (e) Diferentes salas dependiendo del idioma nativo del usuario, y el idioma que desee aprender.
- (f) Comunicación por voz mediante la aplicación Skype, que proporciona una alta calidad independientemente de la distancia a

la que se encuentren los usuarios.

- ◆ Planes:

- (a) Diseño e implementación de Juegos.

- **Iteración 10:**

- ◆ Objetivos:

- (a) Creación de juego del Ahorcado.

- (b) Creación del juego de las preguntas

- ◆ Resultados:

- (a) Juego del ahorcado.

- (b) Sección en el panel de administración de contenidos, donde es posible ampliar la fuente de palabras para el juego del ahorcado.

- (c) Juego de las preguntas. El sistema toma aleatoriamente ejercicios prácticos de la base de contenidos.

4.1.2.- Diseño Multicapa

El diseño e implementación de Babel se ha realizado en tres capas diferentes independientes [POL04] unas de otras: Presentación, Dominio y Persistencia.

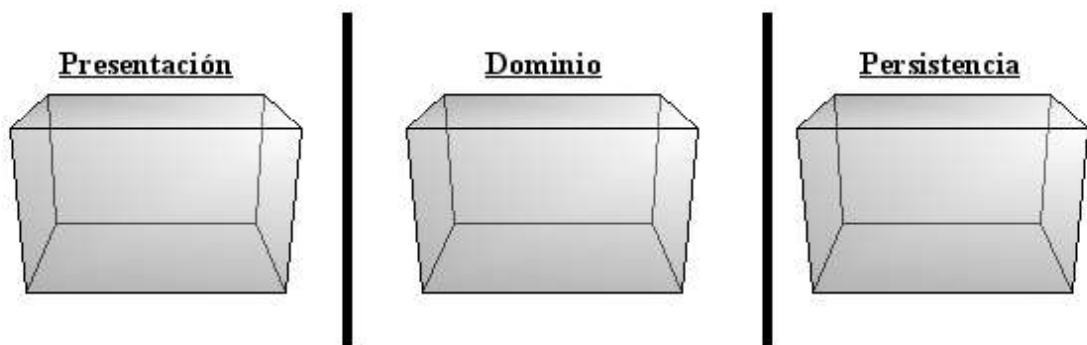


Figura 4.1 Diseño Multicapa

- **Presentación:** Se decide como deben presentarse los datos a los usuarios. Formularios, interfaces, imágenes ...
- **Dominio:** Es donde se encuentra toda la lógica de negocio y funcionalidad del sistema. Implementación de clases, algoritmos, scripts...
- **Persistencia:** Acceso y control sobre los datos que requiere la aplicación, almacenados de forma persistente. Bases de datos, ficheros ...

Cada módulo implementado en Babel tiene su correspondiente representación en la capa de presentación, las clases implementadas en la capa de dominio y su correspondiente clase “broker” implementada en la capa de persistencia para poder manipular los datos almacenados en la base de datos.

Por ejemplo si queremos crear una clase *A* , crearíamos en la capa de presentación *FA.php* , la clase *A* en la capa de dominio y *brokerA.php* en la de persistencia (todas las clases broker son subclases de la clase padre **Broker** donde se encuentran implementadas las funciones básicas para la manipulación de la base de datos: conexión, ejecución sentencia, desconexión ...).

Podemos destacar dos de las ventajas que proporciona el diseño multicapa:

- ✓ Forma cómoda de trabajo en un grupo de desarrollo formado por varios miembros: diseñadores dedicados a la capa de presentación, programadores a la capa de dominio y miembros encargados del diseño y gestión de las bases de datos con la capa de persistencia.
- ✓ Facilidad de mantenimiento: Una modificación en una de las capas no afecta a las restantes. Por ejemplo, si en un momento determinado se decide cambiar el aspecto de presentación de la información, la lógica de funcionamiento de la aplicación y la forma de obtener los datos no varía.

4.2.- VISUALIZACIÓN EN DISPOSITIVOS DE DIFERENTES TAMAÑOS

Uno de los objetivos planteados inicialmente era la posibilidad de visualizar el sistema en dispositivos de diferentes tamaños y resoluciones [COL03].

La forma de conseguirlo es a través de la utilización de hojas de estilo. Con las hojas de estilo es posible definir las propiedades de los elementos presentados en la capa de presentación.

La estructura de las páginas del sistema es la siguiente:

- **Cabecera.** Donde se muestran los elementos comunes a todas las páginas.

- **Menús de navegación:**
 - Si el dispositivo es un ordenador de sobremesa se muestran las pestañas, con los menús desplegables, y un menú en la derecha con las diferentes posibilidades que ofrece el sistema.

 - En el caso de un PDA, al disponer de un menor tamaño de pantalla, no existe menú a la derecha y las opciones son mostradas en el mismo espacio que el contenido central.

- **Contenido central,** específico de cada página.

- **Pie de página.**

La anchura, altura y tamaño de los diferentes componentes (tablas, imágenes, fuentes ...) han sido definidas mediante porcentajes en lugar de medidas puntuales como puedan ser píxeles o pulgadas. Esto permite que el sistema pueda ajustarse de forma automática a la resolución de la pantalla.

Existen dos hojas de estilo en el sistema **pc.css** y **pda.css**, en función de la

resolución detectada se utilizará una u otra.

Actualmente las versiones de los navegadores para PDA no soportan algunas etiquetas HTML como pueden ser las capas o los iframes (página dentro de otra página) y algunas de las funciones JavaScript.

En el caso de los módulos de administración, la creación, actualización y eliminación de elementos esta dividida en diferentes capas, que se muestran u ocultan dependiendo de las acciones que quiera realizar el usuario. Esto permite tener el mayor número de elementos (en una misma página) en el menor espacio posible, haciendo que la usabilidad sea mayor ya que el usuario puede realizar un mayor número de operaciones con un menor número de clics de ratón. El servidor de aplicaciones también se ve favorecido, ya que si cada acción estuviera en una página diferente, el tráfico y la carga de trabajo aumentaría.

Para el chat y el juego del ahorcado ha sido necesario la utilización de funciones JavaScript no soportadas aún por algunos navegadores para PDA.

Por tanto los módulos de administración, el Chat y el juego del Ahorcado no son visualizados en un PDA.

Esto no supone un problema demasiado grave, debido a que la mayoría de usuarios del sistema accederán a través de un ordenador de sobremesa, y aquellos que lo utilicen desde un PDA existe una alta probabilidad de que lo utilicen también en otras ocasiones con un PC.

4.3.- INTERFAZ MULTIIDIOMA

Un sistema dedicado al aprendizaje de idiomas, debe poseer su interfaz en diferentes idiomas. Para hacer posible este objetivo se pensó en dos posibles soluciones.

Una posible forma, sería tener almacenadas las etiquetas en la base de datos.

Solución no muy acertada ya que en la carga de una página en el navegador, el número de consultas para las etiquetas puede ser alto. Teniendo en cuenta que puede haber numerosos usuarios en el mismo instante de tiempo, el sistema gestor de bases de datos puede llegar a convertirse en un cuello de botella.

Otro inconveniente sería que al añadir un nuevo idioma al sistema, supondría la modificación en el diseño de tablas de la base de datos, teniendo que añadir una nueva tabla o nueva columna si se encuentran todas en una única tabla.

La solución por la que se ha optado es la de tener un archivo con las etiquetas para cada uno de los idiomas, incluyéndolo en la cabecera. Este archivo posee el mismo nombre que la clave definida para el idioma en el sistema.

Por ejemplo, si se ha definido el idioma **Español** en el sistema con clave **esp** , el archivo **esp.php** contendrá todas las etiquetas traducidas al castellano.

Por cada una de las etiquetas se crea una variable en el archivo:

```
$e_identificacion = "Identificación";
```

Para evitar posibles riesgos en la repetición de nombres de variables se han establecido una serie de normas:

- Todas las etiquetas deben comenzar por "e_".
- Etiquetas para los botones empiezan por "b_"
- Mensajes de error empiezan por "error_"
- Mensajes del sistema empiezan por "msg_";
- Mensajes largos "msgl_"
- Variables de sesión "s_"

La inclusión de los archivos de idiomas no supone un problema, ya que PHP dispone de hasta 16 MB de memoria temporal.

4.4.- ARQUITECTURA DEL SISTEMA

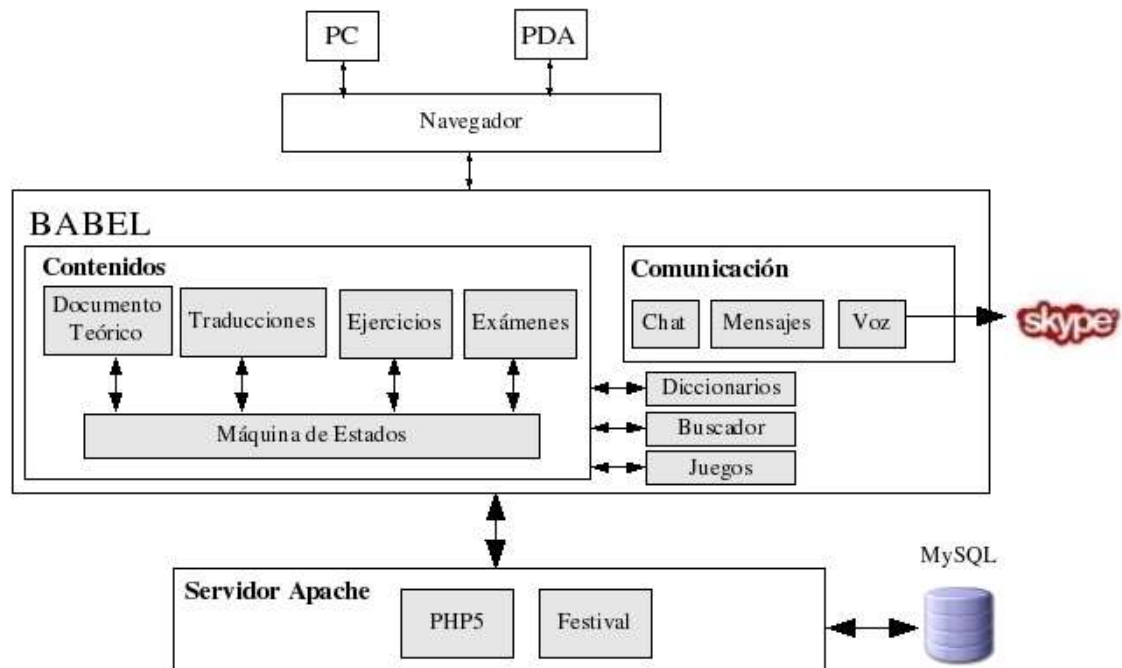


Figura 4.2 Arquitectura de Babel

El sistema puede ser accedido a través de dispositivos de diferentes tamaños como puede ser un PC o un PDA. Es suficiente con la disponibilidad de un **navegador web** y una conexión a la red.

El módulo de **contenidos** esta formado por: Documentos teóricos, traducciones, ejercicios prácticos y exámenes, y la **máquina de estados** que regula el comportamiento del sistema con respecto a los contenidos.

Directamente relacionado con los contenidos, el sistema dispone de **diccionarios** para la consulta de determinados términos, un **buscador** de contenidos que permite acceder a la información de una forma más rápida y sencilla. El sistema además cuenta con un juego de preguntas y el juego del “ahorcado”.

El módulo de comunicación consta de un **Chat** diseñado específicamente para la comunicación entre usuarios de diferentes lenguas, **envío de mensajes** entre usuarios dentro del sistema y comunicación mediante conversación hablada con la utilización de la

aplicación Skype.

El usuario a través del navegador realiza las peticiones de página a un **servidor Apache**. Es en el servidor donde se encuentra el interprete de PHP5, lenguaje en el que esta implementado Babel. Una vez interpretado el código PHP y generado el documento HTML correspondiente, se envía al navegador del usuario. La base de contenidos del sistema se encuentra almacenada en una base de datos MySQL.

La síntesis de voz, se realiza a través del software **Festival** instalado en el servidor. Desde el navegador se enviarán aquellas palabras sobre las que se desee generar el sonido correspondiente. Una vez generado el sonido en un archivo con formato WAV es enviado de nuevo al usuario para que pueda escucharlo. Si el usuario dispone de herramientas de reproducción multimedia en el navegador el sonido se reproducirá directamente.

4.5.- MÁQUINA DE ESTADOS

La máquina de estados afecta directamente al comportamiento de los contenidos en el sistema. Proporciona gran **flexibilidad** y **dinamismo**. En función de su configuración se generarán los diferentes flujos de trabajo entre los usuarios.

Está formada por los siguientes elementos:

- Roles de usuario.
- Estados de un documento.
- Acciones sobre un documento.
- Transiciones entre estados (estado-acción-estado siguiente).
- Permisos de los usuarios sobre los contenidos.

Inicialmente, cuando instalamos el sistema por primera vez, la máquina de estados posee una serie de roles, estados, acciones ... por defecto, pero cada uno de los elementos que constituyen la máquina de estados son modificables. Podemos actualizarlos, eliminarlos o crear nuevos.

Los nombres asignados a roles, estados y acciones serán mostrados posteriormente en el sistema. Al ser Babel un sistema en el que se preve usuarios de diferentes nacionalidades, se ha decidido, que estos nombres deben estar escritos en inglés. Podría crearse una tabla donde se encuentre la traducción a cada idioma de cada uno de los elementos, pero ya no sería tan sencilla la creación o modificación de los elementos de la máquina de estados, ya que cualquier cambio supondría la modificación en todos los idiomas activos del sistema.

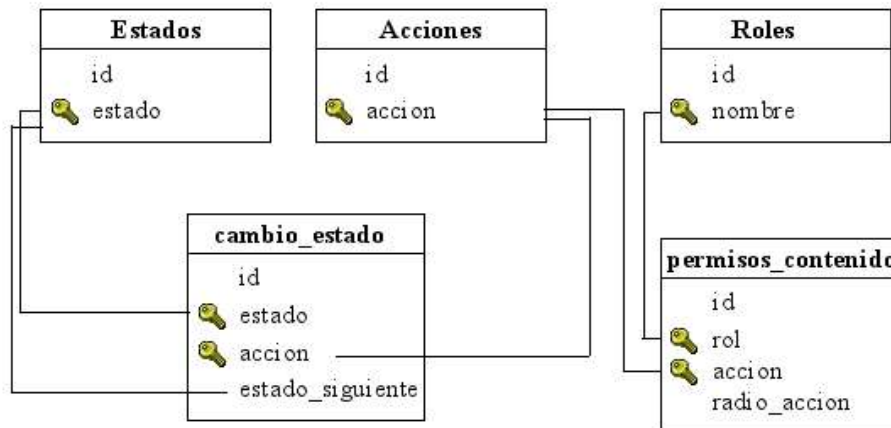


Figura 4.3 Estructura de tablas de la máquina de estados

4.5.1.- Roles de usuario

Cada uno de los usuarios dentro del sistema posee un rol que determinará que acciones podrá realizar sobre los contenidos y los módulos. Es posible crear nuevos roles y actualizar o eliminar los existentes.

Roles iniciales al instalar el sistema:

Roles
Anonymous
Editor
Reviewer
Administrator
Manager

Tabla 4.1 Roles de usuario

- **Anonymous:** Usuario anónimo, no identificado o no registrado en el sistema.
- **Editor:** Editor de contenidos. Usuario registrado en el sistema, con posibilidad de crear nuevos contenidos.
- **Reviewer:** Usuario encargado de revisar los contenidos. Puede realizar correcciones, determinar el nivel de idioma que le corresponde al contenido y publicación final del contenido.
- **Administrator:** Administrador de los parámetros de configuración del sistema.
- **Manager:** Se encarga de configurar la máquina de estados y del mantenimiento de los contenidos para el juego del ahorcado.

4.5.2.- Acciones

Sobre los contenidos almacenados en el sistema se pueden realizar un número determinado de acciones especificadas en la máquina de estados. Es posible generar nuevas acciones de forma dinámica pero a diferencia de los roles o los estados, es necesario **implementar** mediante programación la **funcionalidad** de las mismas, es decir, como debe ejecutarse la acción en cada tipo de contenido.

Para añadir una nueva acción al sistema se deben seguir los siguiente pasos:

- 1) Crear un documento php con el mismo nombre de la acción: por ejemplo, para la acción “Modificar” crearíamos “Modificar.php”.
- 2) Por cada tipo de contenido se creará un documento .php asociado a la acción: modificar_documento.php, modificar_traducccion.php, modificar_ejercicio.php y modificar_examen.php.
- 3) Modificar.php se encargará del redireccionamiento dependiendo del tipo de contenido en el que nos encontremos. Si nos encontramos en documentos teóricos y ejecutamos “Modificar”, Modificar.php redireccionará a modificar_documento.php donde se mostrará los correspondientes formularios para la modificación del documento y

podrán realizarse las acciones oportunas.

- 4) Crear desde el panel de administración de contenidos la etiqueta de la nueva acción, para poder incluirla en la máquina de estados.

Acciones iniciales al instalar el sistema:

<i>Acciones</i>
Create
Read
Modify
View
Finish
Correct
Publish
Recover
Reject
Erase

Tabla 4.2 Acciones sobre los contenidos

- **Create:** Creación de un nuevo contenido.
- **Read:** Visualización de un contenido que se encuentra publicado.
- **Modify:** Modificación de un contenido.
- **View:** Visualización de un contenido que aún no ha sido publicado.
- **Finish:** Se da por terminado un contenido, dando paso a su posterior revisión.
- **Correct:** Realizar correcciones sobre un contenido.
- **Publish:** Publicación de un contenido.
- **Reject:** Rechazar un contenido.
- **Recover:** Recuperar un contenido que en otro momento fue rechazado.
- **Erase:** Eliminar un contenido de la base de contenidos.

4.5.3.- Estados

Cualquier tipo de contenido dentro de Babel, siempre se encuentra en un estado

determinado.

Estados iniciales al instalar el sistema:

<i>Estados</i>
(Vacio)
New
Finished
Corrected
Published
Rejected

Tabla 4.3 Estados en los que se puede encontrar un contenido

- **Estado vacío**: Estado inicial.
- **New**: Nuevo contenido creado.
- **Finished**: Contenido concluido y listo para pasar por las etapas de revisión.
- **Corrected**: Contenido revisado y corregido.
- **Published**: Contenido publicado.
- **Rejected**: Contenido rechazado.

4.5.4.- Transiciones de estados.

Una transición es una tupla formada por el **estado** en el que se encuentra el contenido, la **acción** que se realiza sobre él, y el **nuevo estado** al que cambia. No es posible definir más de una transición con el mismo estado inicial y la misma acción, ya que provocaría inconsistencias en el comportamiento del sistema.

Veamos un ejemplo:

- Nuevo – Terminar – Terminado.
- Nuevo – Terminar – Publicado.

Si un contenido se encuentra en estado “Nuevo” y se ejecuta la acción “Terminar”, en este caso, el sistema no sabría determinar si debe permutar al estado “Terminado” o

“Publicado”.

Es posible la creación , actualización y eliminación de transiciones de la máquina de estados.

Transiciones de estados definidas inicialmente:

<i>Estado inicial</i>	<i>Acción</i>	<i>Estado siguiente</i>
	Create	New
New	Modify	New
New	View	New
New	Finish	Finished
Finished	View	Finished
Finished	Correct	Corrected
Finished	Reject	Rejected
Finished	Publish	Published
Corrected	View	Corrected
Corrected	Reject	Rejected
Corrected	Publish	Published
Corrected	Correct	Corrected
Published	Read	Published
Published	Modify	New
Rejected	View	Rejected
Rejected	Recover	New
Rejected	Erase	

Tabla 4.4 Transiciones de estado

4.5.5.- Permisos sobre los contenidos

Con los permisos se determina cuales de las acciones definidas en la máquina de estados puede realizar un determinado rol. Un permiso esta definido por:

- **Rol**
- **Acción** que puede realizar.

- **Radio de acción**, que determina el ámbito de la acción. Puede tomar dos posibles valores:

- Global: Puede realizar la acción sobre cualquier contenido.
- Personal: Puede realizar la acción únicamente sobre los contenidos en los que aparece como autor.

Permisos definidos inicialmente:

- **Anonymous**: Únicamente puede leer los contenidos publicados.
- **Editor**: Crear, visualizar y modificar sus contenidos y leer los que se encuentren publicados.
- **Reviewer**: Puede crear y modificar sus propios contenidos. De forma global puede modificar, finalizar, corregir, rechazar, recuperar, previsualizar, eliminar y publicar.
- **Manager**: Los mismos privilegios que el revisor, pero además puede modificar la máquina de estados.
- **Administrator**: No tiene permisos sobre los contenidos. Este tipo de usuario se dedica únicamente a tareas de administración del sistema.

4.5.6.- Máquina de estados inicial

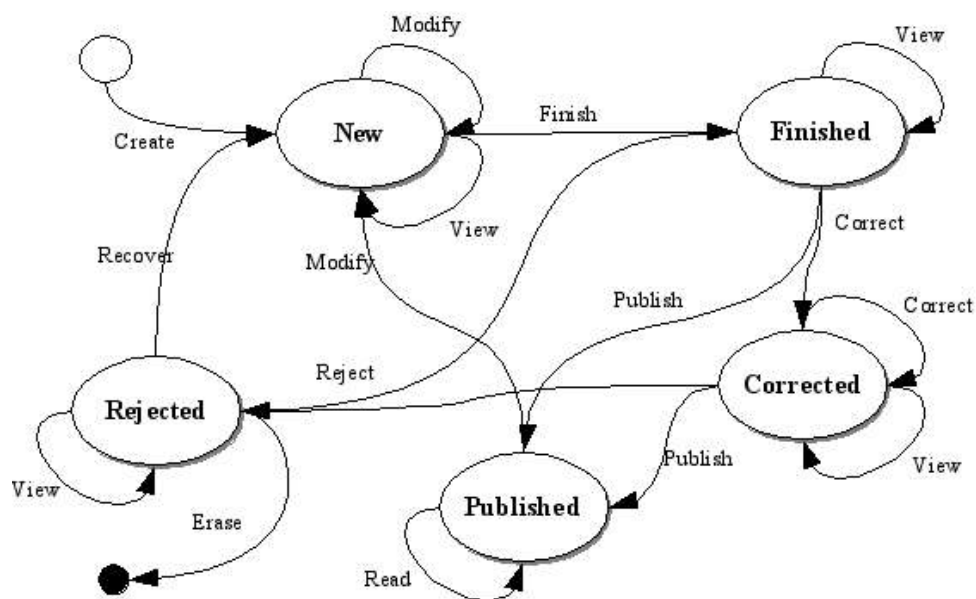


Figura 4.4 Máquina de estados inicial del sistema

En la figura siguiente se refleja la máquina de estados inicial del sistema, y que resume los elementos que la forman citados en los puntos anteriores.

- Un número alto de estados implica:
 - ✓ Mayor control sobre la información.
 - x Proceso lento hasta la publicación.
 - x Mayor número de flujos y carga de trabajo.

- Un número bajo de estados implica:
 - ✓ Publicación rápida.
 - ✓ Poca carga de trabajo.
 - x Escaso control sobre la información.

En función del lugar en el que se implante el sistema y cuales sean las necesidades y preferencias, se construirá una máquina de estados más o menos compleja.

En este caso se ha optado por una solución intermedia, para tener control sobre la información y no tardar demasiado tiempo en la publicación de los contenidos.

4.6.- MÓDULOS

En Babel los contenidos están regulados y controlados a través de la máquina de estados, el resto de elementos están divididos en diferentes **módulos**. Existen inicialmente dos módulos en Babel:

- Administración del sistema.
- Administración de la parte de contenidos.

Existen cuatro permisos diferentes sobre los módulos:

- Visualización.
- Creación.
- Actualización.

- Eliminación.

Los permisos pueden ser definidos a **nivel de rol** (qué permisos tiene un rol sobre un determinado usuario) o a **nivel de un usuario** (permisos de forma específica para cada usuario).

Cuando un usuario se encuentra en un determinado módulo y desea realizar una operación:

- 1) El sistema comprueba primero si tiene permiso específico de usuario.
- 2) Si no tiene permiso el usuario, mira si lo tiene su rol. Si en alguno de los dos casos posee permiso, el usuario podrá realizar la acción dentro del módulo.

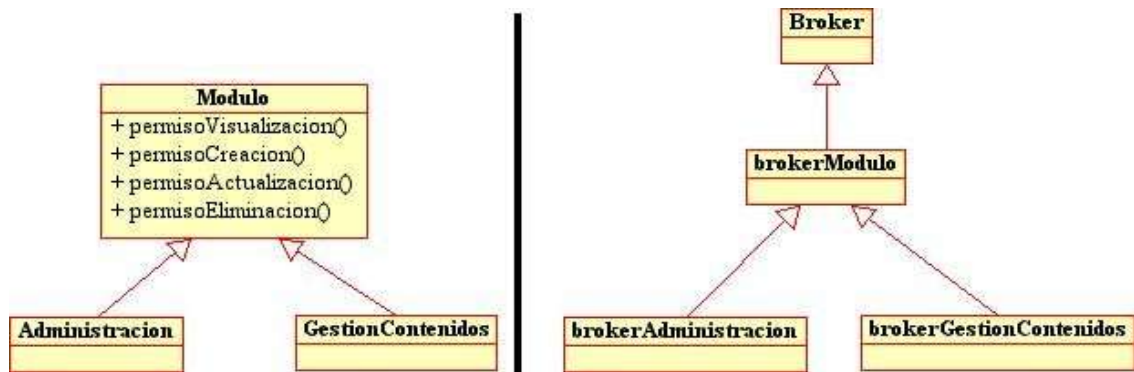


Figura 4.5 Diagrama de clases de los módulos

La seguridad de los módulos está implementada en la clase padre “Modulo” donde se determinan los permisos. En las subclases, específicas de cada módulo se definen las acciones de cada módulo en concreto.

Tal como está realizado el diseño, la creación de un nuevo módulo es una tarea sencilla. Se crearía una nueva subclase de la clase “Modulo” donde se implementarían las acciones dentro de ese módulo, y la correspondiente clase “broker” para el acceso y manipulación de los datos persistentes.

Los permisos quedan almacenados de forma persistente en la base de datos con una estructura de tablas sencilla:

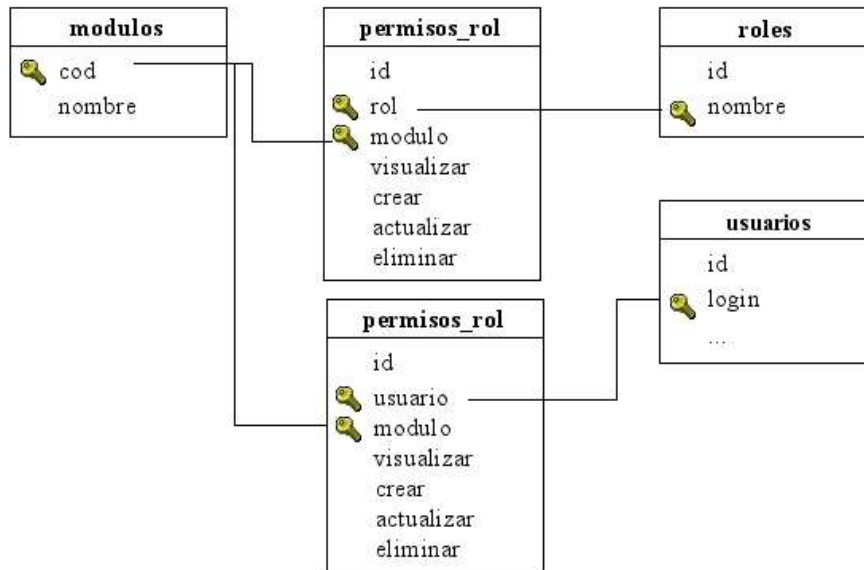


Figura 4.6 Diseño de tablas para los módulos del sistema

- **módulos:** Se almacenan los códigos y nombres de los módulos del sistema.
- **permisos_rol:** Permisos de visualización, creación, actualización y eliminación de un rol sobre un determinado módulo.
- **permisos_usuario:** Permisos de usuario sobre un módulo.

4.6.1.- Módulo de administración

Inicialmente el módulo de administración es únicamente accesible por un usuario cuyo rol sea el de “Administrator”. Pero es posible añadir permisos sobre roles y usuarios de forma individual con respecto a este módulo.

El módulo de administración permite tareas administrativas del sistema:

- **Buscador:** Permite actualizar los índices utilizados para las búsquedas.
- **Gestión de Usuarios:** Administración de usuarios, pudiendo modificar los datos de su cuenta o el rol que posee.
- **Diccionarios:** Desde esta sección es posible asociar un archivo que contenga

términos y definiciones, para ampliar la base de términos del sistema.

- **Gestión de idiomas:** Activación y desactivación de idiomas en el sistema.
- **Definición de niveles de idioma:** Se definen los diferentes niveles de cada idioma especificando los puntos necesarios para alcanzar cada uno de ellos.
- **Roles de usuario.** Definición de los roles del sistema.
- **Módulos del sistema:** Nombre de los módulos instalados en el sistema.
- **Permisos sobre módulos:** Gestión de permisos de los roles y los usuarios sobre los módulos.
- **Variables del sistema:** Las variables sirven como parámetros de configuración. Es posible crear nuevas y editar o eliminar las existentes. Algunos ejemplos de variables del sistema son: contenidos por página que deseamos mostrar, tamaño máximo de un archivo subido al servidor por el usuario, estado final de la máquina de estados, máximo número de preguntas en un ejercicio ...

4.6.2.- Módulo de gestión de contenidos.

El módulo de gestión de contenidos permite configurar la máquina de estados y otros aspectos relacionados con los contenidos. Únicamente es accesible inicialmente por el rol “Manager”, pero al igual que el módulo de administración es posible dar permisos para que otros roles o usuarios tengan acceso.

Las tareas posibles desde el módulo de gestión de contenidos son:

- Creación de categorías. Los contenidos se encuentran clasificados de acuerdo a una serie de categorías. Cada idioma tiene las suyas propias.
- Gestión de acciones posibles sobre los contenidos.

- Gestión de estados.

- Gestión de las transiciones de estados.

- Permisos sobre los contenidos (asociación de los roles y las acciones definidas).

- Ampliación de la fuente de palabras para el juego del ahorcado.

4.7.- CONTENIDOS

Los contenidos constituyen una de las partes más importantes del sistema. Son junto con la parte de comunicación, esenciales para el usuario en el aprendizaje de idiomas. Inicialmente el sistema dispone de cuatro tipos de contenidos diferentes:

- Documentos teóricos.
- Ejercicios prácticos.
- Exámenes, formados por un conjunto de ejercicios prácticos.
- Traducciones y correcciones sobre traducciones.

A la hora de realizar el diseño y la implementación de los contenidos se han tenido en cuenta una serie de factores importantes:

- 1) Evitar almacenar información redundante.
- 2) Facilidad para la creación de nuevos tipos de contenidos.
- 3) Poseer algún patrón o interfaz que indique que operaciones debe implementar siempre cualquier tipo de contenido y que sirva de guía en su creación.

Para solventar el primer punto, se analizó cuales eran los campos o atributos comunes en todos los tipos de contenidos que se iban a implantar inicialmente en el sistema y agruparlos en una única clase.

En cuanto al punto tercero, se podría utilizar la misma clase en la que se hacía la

agrupación de atributos comunes para especificar qué operaciones debe implementar cualquier tipo de contenido. Por tanto, una posible solución sería la creación de una clase abstracta “Contenido” con los atributos y funciones comunes, así como los métodos abstractos que debe implementar cada una de las subclases correspondientes a cada tipo de contenido.

Esta división permite incluir de forma sencilla nuevos tipos de contenidos y solucionar el punto dos, citado anteriormente.

Podemos verlo de forma resumida en el siguiente diagrama de clases, donde se muestra los atributos de la clase contenido (comunes a todos los tipos de contenido) y los métodos abstractos que deben implementar (este diagrama no incluye el resto de operaciones de la clase contenido así como los atributos y métodos del resto de clases).

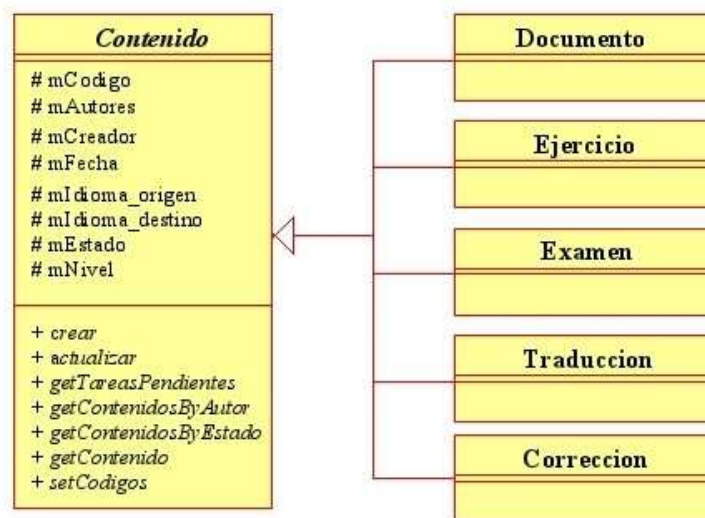


Figura 4.7 Diagrama de clases sobre contenidos (abreviado)

El diseño de tablas en la base de datos es muy similar:

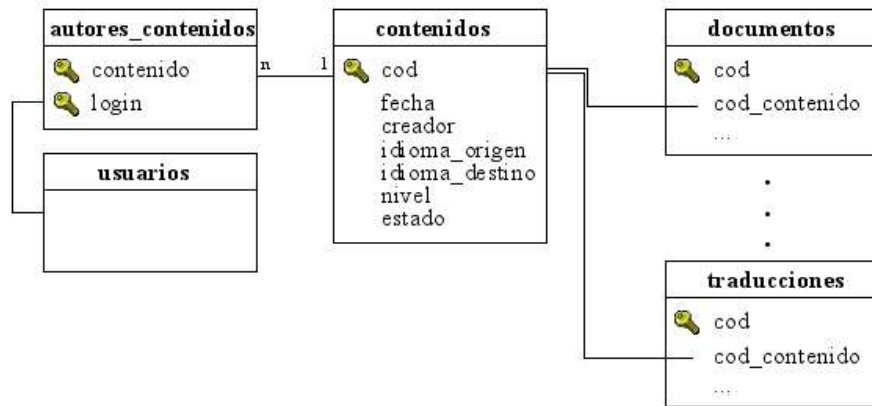


Figura 4.8 Diseño de tablas para los contenidos

Métodos abstractos:

- **crear:** Creación del contenido.
- **actualización:** Modificación del contenido.
- **getTareasPendientes:** Obtiene cuales son las tareas pendientes de un usuario relacionadas con el tipo de contenido.
- **getContenidosByAutor:** Obtiene todos los contenidos del autor.
- **getContenidosByEstado:** Obtiene los contenidos según el estado en el que se encuentren.
- **getContenido:** Obtiene todos los atributos del tipo de contenido.
- **setCodigos:** Completa el objeto con los códigos del contenido tomados de la base de datos, necesarios para recuperar la información y realizar operaciones que produzcan cambios persistentes.

De cara a la usabilidad es un objetivo primordial que el usuario pueda crear los contenidos de forma sencilla, y a la vez poder crear texto enriquecido para dotar a los documentos de una aspecto más atractivo.

Por tanto, para la creación y modificación de contenidos se ha incluido en el sistema editores WYSIWYG.



Figura 4.9 Editor WYSIWYG

El editor utilizado ha sido SPAW. Editor WYSIWYG implementado en lenguaje PHP y JavaScript. La barra de herramientas que presenta es totalmente configurable, ya que disponemos de su código fuente.

Para Babel, se ha confeccionado una barra de herramientas con las siguientes utilidades:

- Deshacer.
- Rehacer.
- Negrita.
- Cursiva.
- Subrayado.
- Posibilidad de incluir barras o líneas de separación.
- Enumeración.
- Listas.
- Tabulaciones.
- Alineaciones: a la izquierda, centrado y a la derecha.

4.7.1.- Acciones posibles sobre un contenido

Siempre que el usuario se encuentre en el contexto de un determinado contenido se muestra cuales son las acciones que se pueden realizar en ese instante sobre el contenido. Estas dependen de:

- El estado en el que se encuentre el contenido.
- El rol del usuario.
- Las acciones que pueda realizar el usuario con el rol que posee.

La función **accionesPosibles(\$cod,\$login,\$rol,\$estado)** implementada en la clase

“**Contenido**” es la encargada de determinar qué acciones puede realizar el usuario.

La forma de obtenerlas es la siguiente:

accionesPosibles (\$cod,\$login,\$rol,\$estado)

1. Cotejamos el estado del contenido con el estado definido en las transiciones de estado pertenecientes a la máquina de estados: **estado** – acción – estado siguiente, obteniendo las acciones posibles sobre el contenido.
2. De las acciones que se pueden realizar sobre el contenido, nos quedamos únicamente con aquellas que pueda realizar el usuario con el rol que posee (definidas en la máquina de estados).
3. **PARA CADA** acción **DE** acciones
 SI es_autor
 acción permitida.
 SI NO es_autor
 SI radio_accion = GLOBAL
 acción permitida.

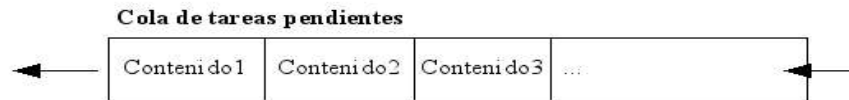
Recordemos que cuando definíamos en la máquina de estados las acciones que podía realizar un rol, definíamos también el radio de acción, que podía ser GLOBAL (sobre cualquier contenido) o PERSONAL (únicamente sobre los contenidos en que es autor).

4.7.2.- Tareas pendientes

Las acciones realizadas por los usuarios sobre los contenidos generan flujos de trabajo que se traducen como tareas pendientes para cada usuario.

Podemos entender por tarea pendiente un contenido sobre el cual es posible realizar

una acción. Las tareas pendientes se encuentran en una cola FIFO, la primera en entrar es la primera en ser mostrada para ser atendida. Cuando se realiza la acción pendiente sobre el contenido pasa de nuevo al final de la cola, siempre y cuando todavía sea posible realizar alguna acción. A la hora de atender las tareas pendientes no tiene por que respetarse el orden impuesto en la cola.



Cada clase correspondiente a cada tipo de contenido debe implementar el método abstracto **getTareasPendientes(\$login,\$rol)**. Existe una cola de tareas pendientes diferente para cada tipo de contenido.

La forma de obtenerlas es la siguiente:

1. Obtenemos los contenidos ordenados por fecha de actualización (fecha de cuando se realizó por última vez una acción) y el estado en el que se encuentran (en la clase **Documento** se obtendrán los documentos, en **Ejercicio** los ejercicios prácticos ...).
2. Se considerará tarea pendiente para el usuario:
 - Aquellos contenidos sobre los cuales el usuario pueda realizar alguna acción. (véase el punto anterior *acciones posibles sobre un contenido*).
 - Además estas acciones tienen que provocar un cambio de estado, no son válidas aquellas acciones definidas en una transición de estado donde el estado inicial y el estado siguiente es el mismo. Por ejemplo: View – Finished – View. Si para el estado Finished sólo existiera esta transición, cuando los contenidos se encontraran en este estado no se producirían tareas pendientes.
 - El estado del contenido debe ser distinto al estado final (Publicado).

Cada usuario tendrá mayor o menor carga de trabajo de acuerdo a la configuración de la máquina de estados. Con la configuración inicial, la carga de trabajo de los revisores es mucho más alta que la de los editores. Un número mayor de privilegios implica mayor responsabilidad y tareas por cumplir.

4.7.3.- Bloqueo de contenidos

Un contenido puede pertenecer a varios usuarios, por tanto, se puede dar el caso en el que varios usuarios accedan en un mismo instante de tiempo al mismo contenido. El sistema no permite el trabajo concurrente sobre un mismo contenido, la forma que tiene de actuar es bloqueando el recurso cuando un usuario va a realizar una acción que pueda modificar el contenido, evitando de esta forma, posibles sobreescrituras.

La forma de implementarlo ha sido mediante la utilización de **semáforos**:

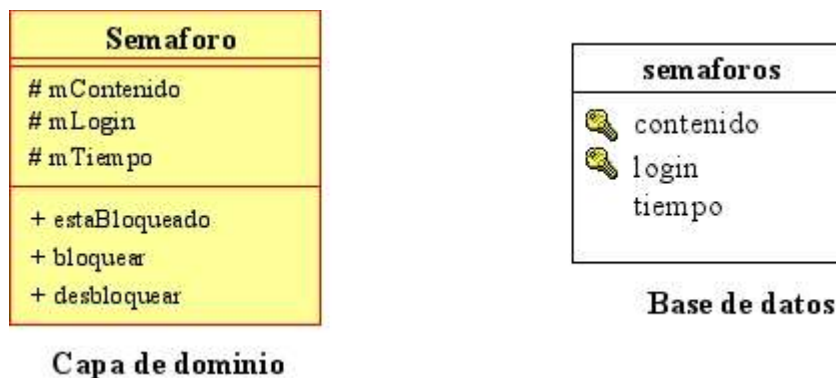


Figura 4.10 Bloqueo de contenidos mediante la implementación de semáforos

La clase **Semaforo** de la capa de dominio, tiene como atributos el código del contenido que se encuentra bloqueado, el login del usuario que ha bloqueado el recurso y el instante de tiempo en el que se bloqueó. Posee implementadas las operaciones **estaBloqueado** que informa si un determinado recurso se encuentra bloqueado, el método **bloquear** para bloquear el recurso y **desbloquear** para desbloquearlo.

Los recursos bloqueados son almacenados en una tabla de la base de datos llamada **semaforos**. Cuando el recurso es liberado se elimina de la tabla.

El atributo **tiempo** es fundamental. Existe un tiempo máximo en el que un recurso puede encontrarse bloqueado, el valor de este tiempo puede ser definido desde la administración.

Se debe dejar un tiempo prudencial para la modificación de contenidos, actualmente el valor máximo de tiempo coincide con el tiempo de vida de la sesión en el navegador, de tal forma que si caduca la sesión, el usuario no puede operar dentro del sistema y el recurso queda liberado.

Los casos en los que se puede alcanzar el límite máximo de tiempo son:

- El usuario se encuentra en el sistema y ha superado el límite de tiempo.
- El usuario abandonó el sistema de forma incorrecta. Por ejemplo, cerrando la ventana del navegador o por un fallo producido por el suministro de energía en el dispositivo con el que lo está visualizando.

4.7.4.- Documentos teóricos

Un documento teórico consta de los siguientes atributos:

- Título.
- Subtítulo.
- Descripción.
- Contenido.
- Categoría.

Existe la posibilidad de poder asociar ficheros a un documento, pudiendo adjuntar imágenes o cualquier otro documento que el usuario tenga almacenado en su máquina. El sistema permite la generación del **documento** en formato **PDF** y su posterior descarga. Por tanto los documentos podrán ser accedidos desde el sistema o bien de forma local en la máquina del usuario siempre y cuando lo haya obtenido con anterioridad en formato PDF.

Para facilitar el acceso a los documentos se han incluido una serie de filtros, que permiten al usuario listar los documentos de acuerdo a una serie de criterios:

- Seleccionar todos los documentos de una categoría.
- Ordenar según la fecha.
- Ordenar según el título.
- Ordenar según el nivel del idioma.
- Es posible elegir entre una ordenación ascendente o descendente.

El diseño está formado por una clase **Documento** que hereda de la clase abstracta **Contenido**, donde se encuentran los atributos comunes a todos los tipos de contenidos y los métodos abstractos que está obligada a implementar.

A parte de los métodos abstractos, la clase **Documento** posee los suyos propios:

- **maxSize**: Obtiene el tamaño máximo permitido para los ficheros asociados a un documento y que son subidos al servidor.
- **crearArchivo**: Cuando un usuario asocia un fichero a un documento, se guarda en el sistema con el siguiente formato para evitar sobreescrituras: Código del contenido seguido de la fecha y el nombre del archivo con su correspondiente extensión. Por ejemplo: 42_2005-11-15_deportes.doc
- **getArchivo**: Obtiene un archivo asociado a un documento.
- **eliminarArchivo**: Se encarga de eliminar físicamente el archivo del sistema junto con la asociación al documento al que pertenece.
- **numDocumentos**: Número de documentos del sistema.

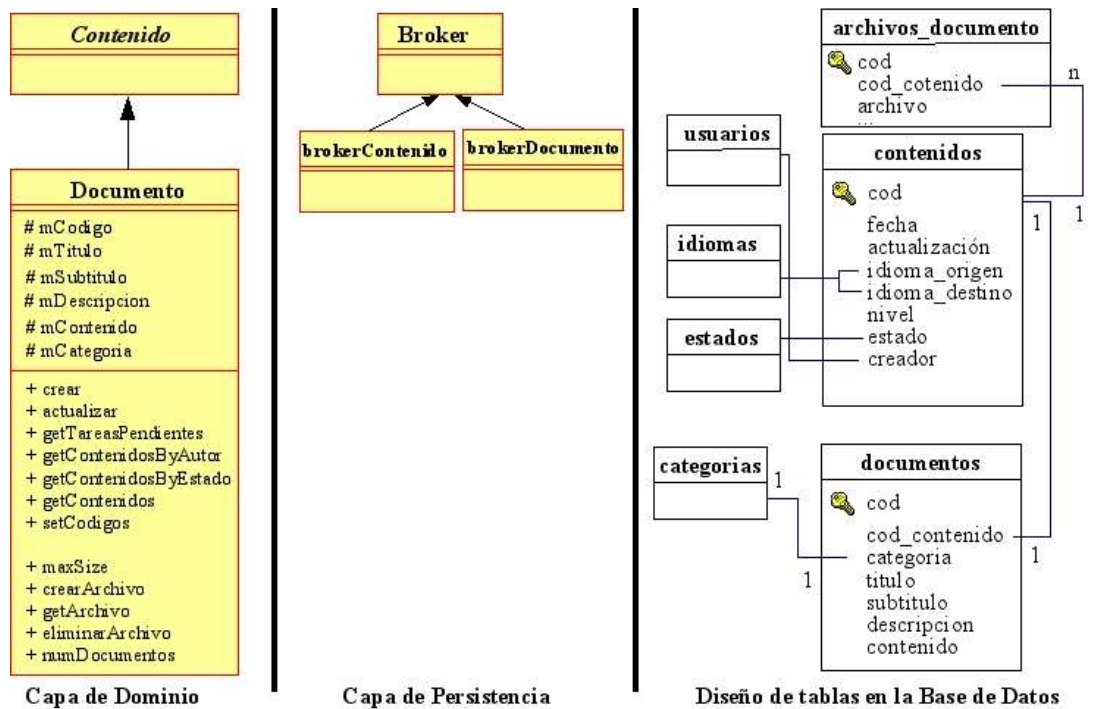


Figura 4.11 Diseño de documentos teóricos

La información de los documentos queda almacenada de forma persistente en la base de datos. Todas las operaciones de los documentos relacionadas con la base de datos se encuentran implementadas en la clase **brokerDocumento** que hereda de la clase **Broker**.

A continuación se pasará a describir brevemente la estructura de tablas en la base de datos, necesaria para almacenar los documentos:

- **contenidos:** Tabla donde se almacenan los datos comunes a cualquier tipo de contenido.
 - Fecha de creación.
 - Actualización: Fecha en la que se realizó por ultima vez una acción sobre el documento.
 - Idioma origen e Idioma destino: Los contenidos poseen un idioma origen y un idioma destino, que indican el idioma del autor y el idioma que se desea aprender. Por ejemplo, un usuario cuyo idioma nativo sea el alemán y desea que con ese documento se puedan aprender algunos

conceptos en inglés. Puede realizar un documento mixto en el que mezcle inglés y comentarios en alemán.

- Nivel de dificultad al que pertenece el contenido.
- Estado en el que se encuentra el contenido.
- Creador del contenido.

- **documentos:** Tabla donde se almacena los datos específicos de un documento: título, subtítulo, descripción, contenido y categoría. La tabla documentos y contenidos están relacionadas a través del código del contenido que es único.
- **archivos_documentos:** Archivos asociados a un documento.

4.7.5.- Traducciones

Otro de los tipos de contenidos existentes en el sistema, son las traducciones libres. Las traducciones constan de tres campos:

- Título de la traducción.
- Texto a traducir.
- Texto traducido.

Una vez publicadas, el resto de usuarios pueden realizar correcciones, especificando las partes incorrectas de la traducción junto con la solución o posibles mejoras que se podrían realizar.

Se ha creado la clase **Traduccion**, que hereda e implementa los métodos abstractos de la clase **Contenido**. Posee dos métodos aparte de los abstractos:

- **getCodigoByPadre:** Obtiene el código de la traducción a partir del código del contenido.
- **numTraducciones:** Número de traducciones creadas en el sistema.

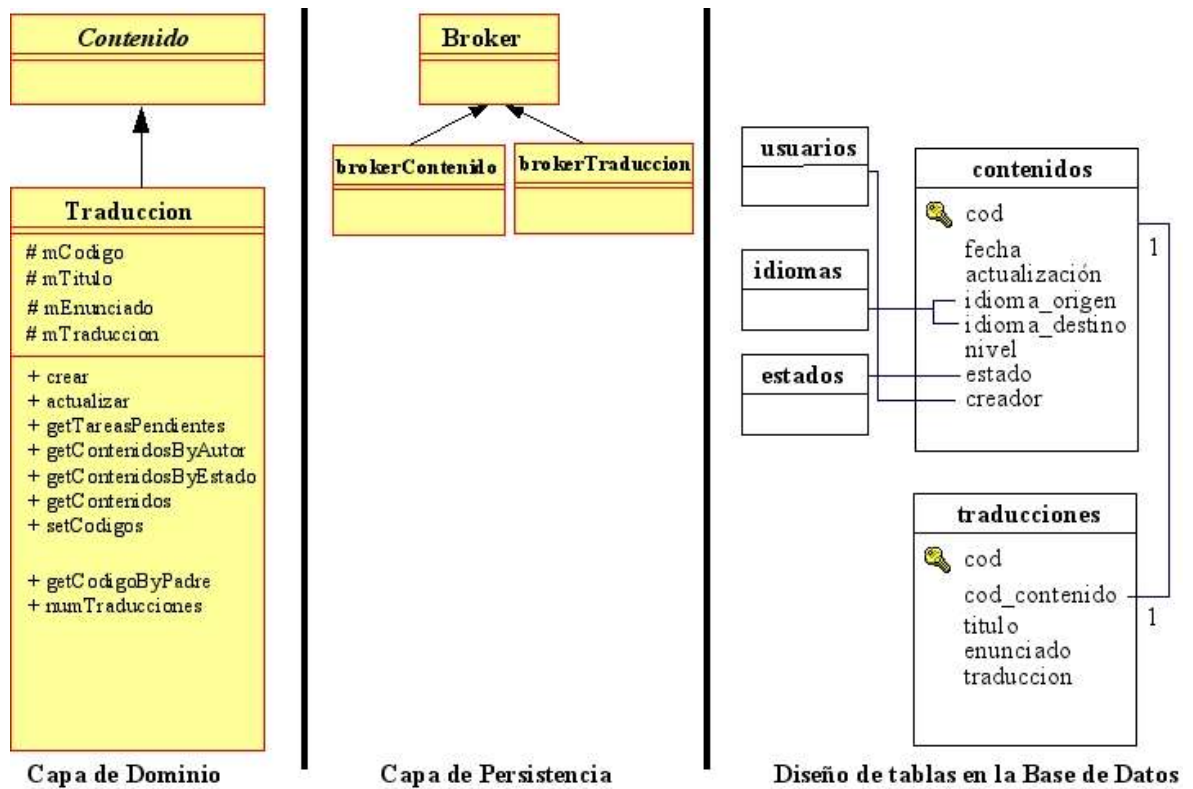


Figura 4.12 Diseño traducciones

Las tablas necesarias para almacenar de forma persistente las traducciones son:

- **Contenidos:** (véase 4.7.4).
- **Traducciones:** Donde se almacena el título, enunciado y traducción. Está relacionada con la tabla contenidos a través del campo cod_contenido.

4.7.6.- Correcciones sobre las traducciones

Las correcciones están altamente relacionadas con las traducciones. Cuando las traducciones son publicadas, el resto de usuarios puede realizar correcciones o mejoras, quedando siempre patente el fragmento de texto original y la modificación sobre éste.

Los atributos que forman una corrección son:

- Código del contenido original.

- Texto original sobre el que se desea realizar la corrección.
- Corrección o modificación realizada.

En la siguiente figura vemos un ejemplo de una corrección realizada sobre parte del texto de una traducción.



Figura 4.13 Ejemplo de corrección

El texto enmarcado en el recuadro, pertenece al texto original. Debajo se muestra la propuesta del usuario que ha realizado la corrección.

Se ha creado una clase **Correccion** que hereda de la clase **Contenido**.

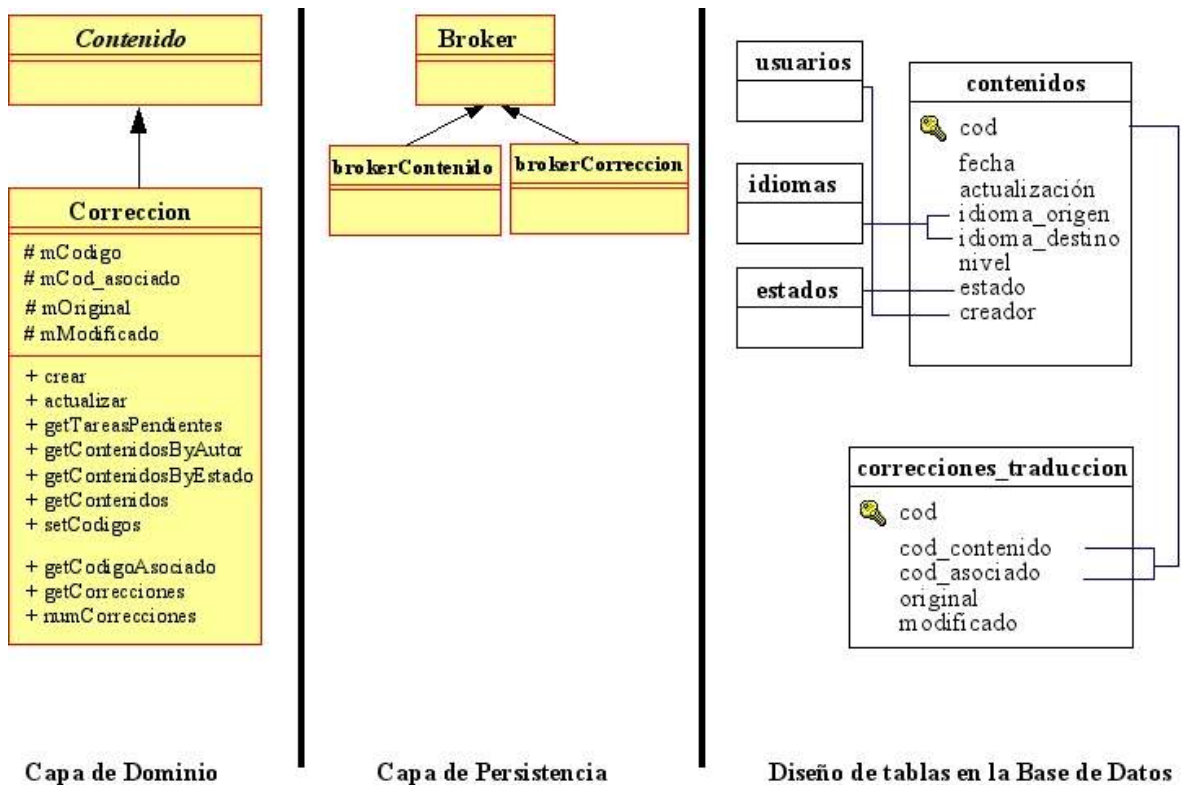


Figura 4.14 Diseño de correcciones

Posee tres métodos propios:

- **getCodigoAsociado:** Obtiene el código de la traducción sobre la que está realizada la corrección.
- **getCorrecciones:** Obtiene todas las correcciones realizadas sobre una determinada traducción.
- **numCorrecciones:** Número de correcciones de una traducción.

Las correcciones son almacenadas en la base de datos en la tabla **correcciones_traducion:**

- código: Código de la corrección.
- cod_contenido: Código que le corresponde en la tabla de contenidos.
- cod_asociado: Código de la traducción sobre la que está hecha la corrección (el código que tiene la traducción en la tabla de contenidos).
- original: Texto original de la traducción sobre el que se va a realizar una corrección.
- modificado: Corrección realizada.

4.7.7.- Ejercicios prácticos

Los usuarios del sistema pueden crear diferentes tipos de ejercicios prácticos. Una vez publicados, pueden ser resueltos por el resto de usuarios y el sistema se encargará de corregirlo automáticamente.

Existen dos tipos de ejercicios prácticos:

- Selección de la respuesta o respuestas correctas: Se formula una pregunta y se especifican las respuestas, seleccionando posteriormente cuál o cuáles son las verdaderas. También existe la posibilidad de que todas sean falsas.

- Completar espacios en blanco. Sobre un texto aquellas palabras que se escriban entre corchetes, aparecerán como espacios en blanco que deberán ser completados. Por ejemplo: Marte es el planeta [rojo] , se mostrará Marte es el planeta ____ . Siendo la respuesta la palabra “rojo”.

Los ejercicios se encuentran implementados en la clase **Ejercicio** de la capa de dominio. Esta clase es hija de la clase abstracta **Contenido** de donde hereda los atributos comunes a todos los tipos de contenidos, los métodos comunes y los métodos abstractos que debe implementar.

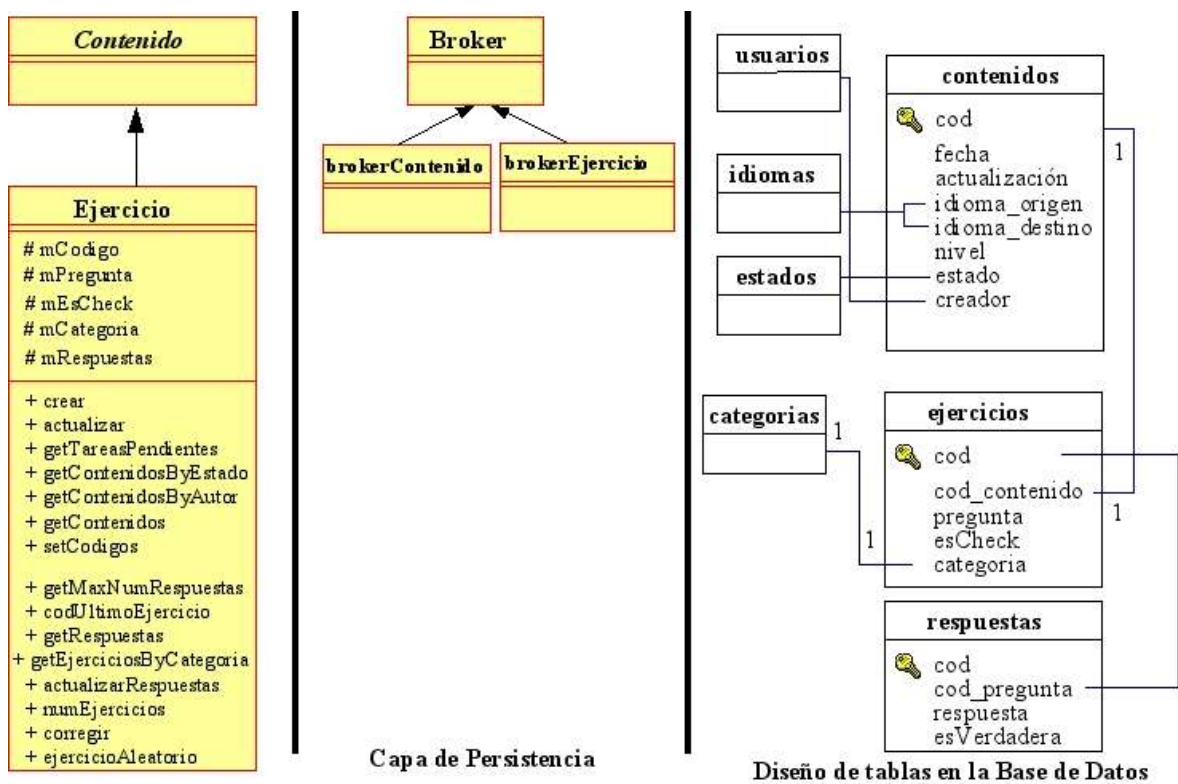


Figura 4.15 Diseño de ejercicios prácticos

La clase Ejercicio presenta cinco atributos:

- mCodigo: Código del ejercicio.
- mPregunta: Si el ejercicio es del tipo selección de una o varias respuestas correctas, sirve para especificar la pregunta, y si es del tipo completar espacios

en blancos, se incluye el texto con aquellas palabras entre corchetes donde se desee que aparezcan los espacios en blanco.

- **mEsCheck**: Si es verdadero, el ejercicio es del tipo selección de respuesta o respuestas correctas, si por el contrario es falso, es del tipo completar espacios en blanco.
- **mCategoria**: Categoría a la que pertenece el ejercicio.
- **mRespuestas**: Código de las respuestas que pertenecen al ejercicio.

Posee ocho métodos propios:

- **getMaxNumRespuestas**: Obtiene el número máximo de respuestas permitidas para una pregunta. El número máximo de preguntas puede ser modificado desde la administración.
- **codUltimoEjercicio**: Código del último ejercicio creado por un autor.
- **getRespuestas**: Obtiene las respuestas de un ejercicio.
- **getEjerciciosByCategoria**: Ejercicios que pertenecen a una determinada categoría.
- **actualizarRespuestas**: Actualización de las respuestas de un ejercicio.
- **numEjercicios**: Número de ejercicios creados en el sistema.
- **corregir**: Corrección del ejercicio. Devuelve verdadero si está correctamente resuelto o falso en caso contrario.
- **ejercicioAleatorio**: Toma un ejercicio de forma aleatoria de la base de contenidos.

Los ejercicios son almacenados en la base de datos de forma persistente en dos tablas. La tabla **ejercicios** donde se almacenan los ejercicios del tipo *completar espacios en*

blanco y la pregunta de los ejercicios del tipo *selección de la respuesta o respuestas correctas*. Para estos últimos existe la tabla **respuestas**, donde se almacenan las posibles respuestas indicando si es verdadera o falsa.

4.7.7.1.- Corrección de ejercicios.

La corrección de los ejercicios se realiza a través del método **corregir** implementado en la clase **Ejercicio**. Los parámetros de este método son dos arrays. Un primer array con las respuestas correctas, y un segundo array con las respuestas del usuario.

En el caso de que el ejercicio sea del tipo en el que hay que seleccionar una o varias respuestas, los arrays estarán formados por secuencias de ceros y unos. Por ejemplo, una pregunta con tres respuestas siendo la segunda la única verdadera, se generaría un array con los valores: 0 | 1 | 0.

Cuando los ejercicios son del tipo completar espacios en blanco, los arrays estarán formados por las palabras que se deben incluir en los espacios en blanco (aquellas que se habían especificado entre corchetes), en el caso del array de respuestas, y por las palabras con las que el usuario ha completado los espacios en el array de respuestas del usuario. Por ejemplo : [Marte] es el planeta [rojo] produciría el array de soluciones [Marte | rojo].

A la hora de crear los arrays, todas las letras de las palabras son transformadas a minúsculas, de tal forma que si la respuesta es “Marte” y el usuario escribe “marte”, seguiría siendo correcta.

El procedimiento de corrección para los dos casos es el mismo: Comprobar que el array de respuestas y el de soluciones dadas por el usuario sean idénticos. Si los arrays son iguales el método devolverá verdadero (ejercicio correcto), en caso contrario devolverá falso.

4.7.8.- Exámenes

Los exámenes son una agrupación de ejercicios prácticos. Cualquier usuario registrado en el sistema puede crear exámenes. Una de las grandes ventajas es la posibilidad de reutilización de la base de contenidos. Cuando un usuario se dispone a crear un examen no sólo puede seleccionar su propios ejercicios prácticos, si no que además puede elegir ejercicios creados por otros usuarios que ya han sido publicados con anterioridad.

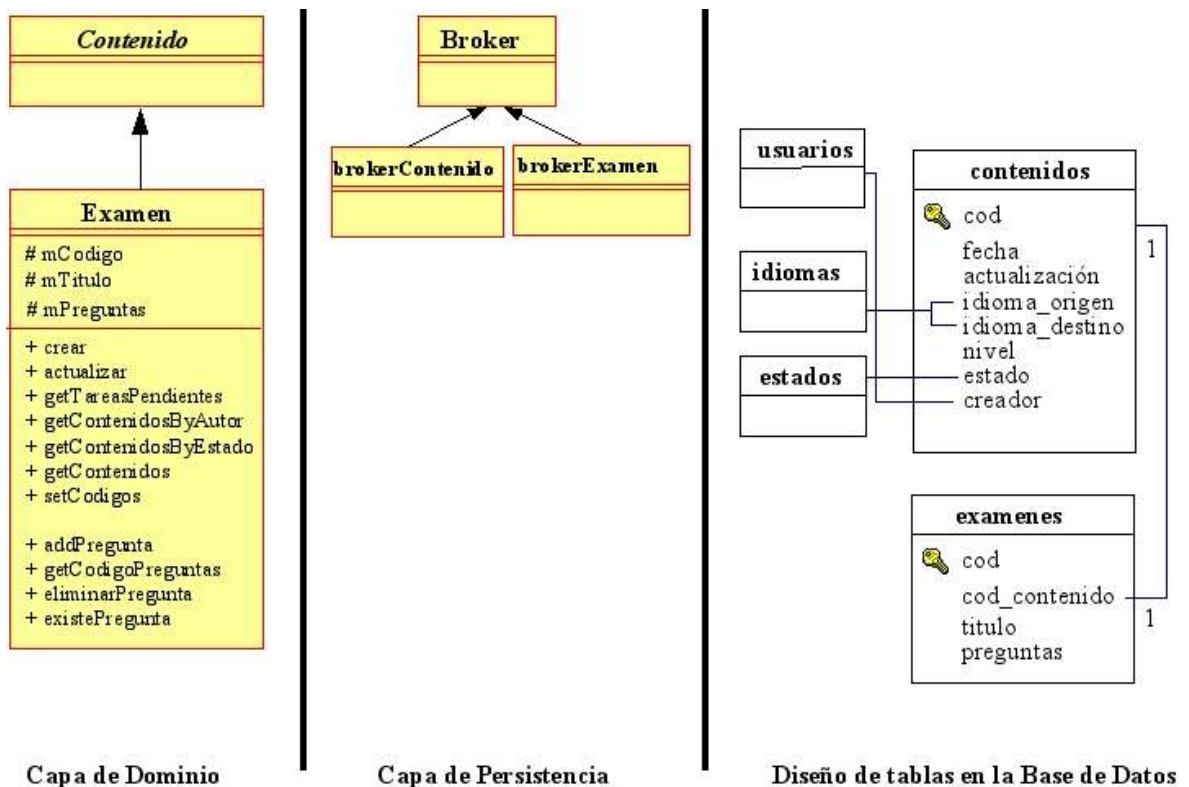


Figura 4.16 Diseño de exámenes

La implementación de los exámenes se ha realizado en una clase llamada **Examen**. con los atributos:

- **mCodigo:** Código del examen.
- **mTitulo:** Título del examen.
- **mPreguntas:** Listado de preguntas que forman parte del examen.

La clase además tiene cuatro métodos propios sin contar los métodos abstractos que debe implementar de la clase **Contenido**:

- **addPregunta**: Añade una nueva pregunta al examen.
- **getCodigoPreguntas**: Obtiene el código de las preguntas que forman parte del examen.
- **eliminarPregunta**: Elimina una pregunta del examen.
- **existePregunta**: Comprueba si una pregunta ya existe en un examen.

Las operaciones relacionadas con la base de datos están implementadas en la clase **brokerExamen**. En la base de datos existe la tabla **examenes** donde se guardan los exámenes creados junto con su título y los códigos de las preguntas que lo forman.

4.7.8.1.- Corrección de exámenes

La corrección de exámenes es un proceso sencillo una vez implementado la corrección de un ejercicio. Se utilizan dos variables donde se van acumulando los ejercicios correctos y los incorrectos. Para cada uno de los ejercicios que componen el examen se utiliza el método **corregir** de la clase **Ejercicio** que nos informa si el ejercicio ha sido resuelto correctamente. En función del resultado se incrementará la variable destinada a ejercicios correctos o la destinada a ejercicios incorrectos.

Finalizada la corrección se muestran todos los ejercicios con su solución y el número total de ejercicios correctos e incorrectos.

4.8.- BUSCADOR DE CONTENIDOS

El buscador de contenidos permite al usuario poder acceder a la información almacenada en la base de contenidos a partir de un conjunto de palabras claves. Permite realizar búsquedas sobre todo tipo de contenidos o sobre algún tipo en concreto. Es posible realizar búsquedas específicas sobre:

- Documentos.
- Traducciones.
- Ejercicios prácticos.
- Exámenes.

Es un factor importante que el tiempo de búsqueda sea bajo. Inicialmente se pensó en la posibilidad de buscar directamente sobre los campos de la base de datos con consultas, mediante la sentencia SELECT:

- ✓ Las búsquedas siempre están actualizadas.
- x Mal rendimiento cuando la base de contenidos alcanza un volumen considerable.

Por tanto esta solución no es viable, debido al rendimiento que ofrecerá con una base de contenidos amplia.

La solución que se ha tomado, es la de realizar un procesamiento de los contenidos almacenados para extraer índices y almacenarlos en una estructura de datos:

- x Las búsquedas no están actualizadas al instante
- ✓ Velocidad alta en la búsqueda de contenidos.

El procesamiento de los contenidos consiste en eliminar las etiquetas HTML que estos posean, símbolos o caracteres especiales y dejar únicamente las palabras. Cada una de las palabras será almacenada en una tabla de índices con los códigos de los contenidos donde aparece la palabra.

Cada tipo de contenidos tiene su propia tabla de índices. Una tabla para los documentos, otra para los ejercicios ...

Veamos un ejemplo:

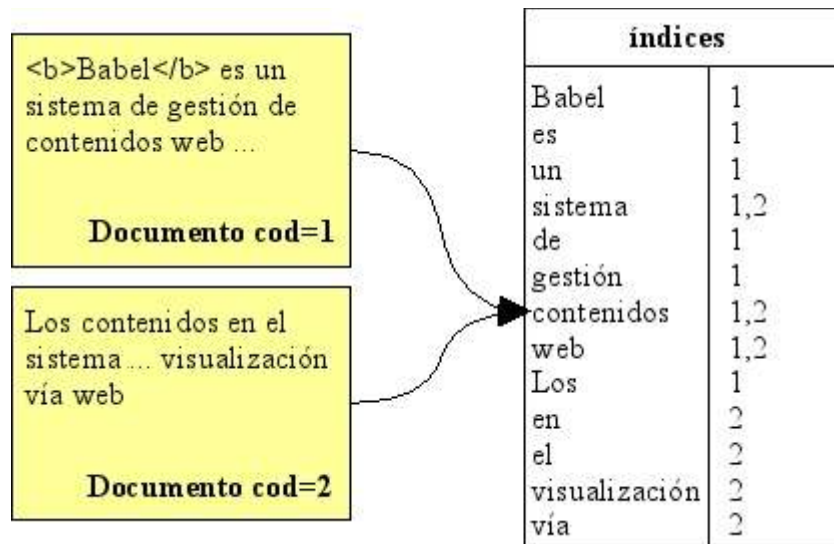


Figura 4.17 Creación de índices para el buscador

Para la generación de índices se analizan las palabras de todos los campos de un tipo de contenido.

Pertenece al resultado de la búsqueda aquellos contenidos en los que:

- Alguno de sus campos contengan cualquiera de las palabras especificadas en la búsqueda.
- El autor que está realizando la búsqueda pueda realizar alguna acción sobre el contenido. Puede que exista en la base de datos algún contenido con las palabras buscadas, pero no es mostrado si el usuario no puede realizar alguna acción.

Por cada uno de los contenidos encontrados se muestra:

- El título del contenido.
- El creador.
- Los fragmentos de texto pertenecientes al contenido donde se encuentran las palabras claves, que serán mostradas de forma destacada. Para acelerar el proceso únicamente se busca la primera aparición de cada palabra clave y sólo sobre los contenidos de la página actual (los contenidos encontrados se muestran paginados).
- El par de idiomas al que pertenece.
- La fecha en la que se actualizó por última vez.

- El tipo de contenido del que se trata.

4.8.1.- Creación de índices para los documentos

El procedimiento seguido para la generación de índices en los documentos es el siguiente:

```
PARA CADA documento
    //la función limpiar elimina las etiquetas HTML y
caracteres especiales
    //y símbolos de puntuación.
    LIMPIAR(titulo)
    LIMPIAR(subtitulo)
    LIMPIAR(descripción)
    LIMPIAR(contenido)

    OBTENER ARRAY DE PALABRAS título
    OBTENER ARRAY DE PALABRAS subtitulo
    OBTENER ARRAY DE PALABRAS descripción
    OBTENER ARRAY DE PALABRAS contenido

    PARA CADA palabra de CADA array
        SI palabra EXISTE
            ACTUALIZAR índices
        SI NO
            CREAR palabra
            INSERTAR índice
        FIN_SI
    FIN_PARA
FIN_PARA
```

4.8.2.- Creación de índices para las traducciones

El procedimiento para la generación de índices de las traducciones es muy similar al de los documentos. La única diferencia se encuentra en los campos.

```

PARA CADA traducción
    LIMPIAR(titulo)
    LIMPIAR(enunciado)
    LIMPIAR(traduccion)

OBTENER ARRAY DE PALABRAS título
OBTENER ARRAY DE PALABRAS enunciado
OBTENER ARRAY DE PALABRAS traducción

PARA CADA palabra de CADA array
    SI palabra EXISTE
        ACTUALIZAR índices
    SI NO
        CREAR palabra
        INSERTAR índice
    FIN_SI
FIN_PARA
FIN_PARA

```

4.8.3.- Creación de índices para los ejercicios prácticos

En el caso de los ejercicios prácticos podemos tener preguntas y respuestas. La búsqueda debe realizarse sobre los atributos de las preguntas y sobre los de las respuestas. El procedimiento es el siguiente.

```

PARA CADA ejercicio
    LIMPIAR(pregunta)
    OBTENER ARRAY DE PALABRAS pregunta
    PARA CADA palabra de la pregunta
        SI palabra EXISTE
            ACTUALIZAR indices
        SI NO
            CREAR palabra
            INSERTAR índice
    FIN_SI

```

```

FIN_PARA
  PARA CADA respuesta de la pregunta
    LIMPIAR(respuesta)
    OBTENER ARRAY DE PALABRAS respuesta
    PARA CADA palabra de la respuesta
      SI palabra EXISTE
        ACTUALIZAR indices
      SI NO
        CREAR palabra
        INSERTAR índice
    FIN_SI
  FIN_PARA
FIN_PARA
FIN_PARA

```

4.8.4.- Creación de índices para los exámenes

Recordemos que un examen no es más que un conjunto de ejercicios prácticos, por tanto para la generación de índices es posible la reutilización del procedimiento que generaba los índices para los ejercicios.

```

PARA CADA examen
  LIMPIAR(titulo)
  OBTENER ARRAY DE PALABRAS título
  PARA CADA palabra del titulo
    SI palabra EXISTE
      ACTUALIZAR índices
    SI NO
      CREAR palabra
      INSERTAR índice
  FIN_SI
FIN_PARA
PARA CADA ejercicio del examen
  CREAR índices del ejercicio.
//estos índices son almacenados en la tabla de índices de los
//exámenes y no en la de los ejercicios.
FIN_PARA

```

4.9.- DICCIONARIOS

El diccionario es una herramienta con la que el usuario puede obtener el significado de las palabras. Babel es un sistema en el que existen diferentes idiomas activos, por lo que surge la necesidad de tener diccionarios para diferentes lenguas.

Otra complejidad añadida es la combinación de estos idiomas (español-inglés, español-francés, francés-alemán ...) donde cada uno necesita su propio diccionario. Un mayor número de idiomas activos supone un mayor número de combinaciones posibles.

Por tanto, las necesidades que surgen a la hora de realizar el diseño y la implementación de diccionarios en Babel son las siguientes.

- Facilidad para la creación de nuevos diccionarios.
- Posibilidad de añadir nuevos términos.
- Diccionarios con un amplio número de términos para que la mayoría de búsquedas realizadas muestren un resultado.
- Búsqueda rápida.

Existen en la red un número amplio de diccionarios de carácter libre en formato de texto plano, con extensiones como .txt o .dict. Estos diccionarios son utilizados por numerosas herramientas para la búsqueda de términos.

Con la utilización de estos tipos de diccionarios se solventa el problema de tener un amplio número de términos, y en diferentes idiomas con diferentes combinaciones.

Inicialmente se tomó la decisión de utilizar el fichero del diccionario directamente, es decir, se asociaba un fichero a un par de idiomas. Cuando se realizaba la búsqueda se abría el fichero y se recorría de forma secuencial buscando coincidencias. Pero los resultados en cuanto a rendimiento no eran demasiados buenos, teniendo en cuenta que estos diccionarios poseen entre 20.000 y 25.000 términos.

Las ventajas y desventajas de esta solución son las siguientes:

- ✓ Facilidad para añadir nuevos diccionarios.
- ✓ Número amplio de términos.
- ✗ Sólo es posible asociar un diccionario a un par de idiomas.
- ✗ Búsquedas lenta, dependiendo el tiempo de la posición en la que se encuentre el término en el fichero.

La estructura de estos tipos de diccionarios es sencilla. En cada una de las líneas del fichero se define el término y su definición, separadas por un determinado carácter. Suele ser “:” o un carácter de tabulación, por lo que extraer los términos es una tarea sencilla. Almacenar estos términos en la base de datos permitirá al sistema realizar búsquedas mucho mas rápidas al no tener que hacerlo de forma secuencial.

La **solución** que se tomó, fue crear una tabla en la base de datos con los campos: palabra1, palabra2, idioma1, idioma2.

Desde el panel de administración es posible especificar el carácter utilizado para la separación, y el fichero sobre el cuál se realizará la extracción de términos y definiciones que serán almacenados en la base de datos. Esto supone:

- ✓ Facilidad para añadir nuevos diccionarios.
- ✓ Número amplio de términos.
- ✓ Es posible añadir nuevos términos a un par de idiomas, ya que se puede especificar un número indeterminado de ficheros. Esto permitiría añadir diccionarios especializados en determinadas áreas. Por ejemplo diccionarios especializados en informática, medicina ...
- ✓ Búsquedas rápidas.

Las búsquedas se realizan a través de una sentencia SELECT buscando las coincidencias con LIKE.

```
SELECT palabra1,palabra2 FROM diccionarios WHERE (palabra1
LIKE '%$palabra%' OR palabra2 LIKE '%$palabra%') AND
((idioma1='$idioma1' AND idioma2='$idioma2') OR
```



```
(idioma1='$idioma2' AND idioma2='$idioma1'))
```

Si el usuario se encuentra en el idioma Español-Inglés se realiza la búsqueda sobre términos Español-Inglés o Inglés-Español almacenados en la tabla diccionarios.

El resultado de la búsqueda son dos arrays: Un array de coincidencia exacta de la palabra y otro con palabras similares. Por cada uno de los resultados obtenidos en la sentencia SELECT se analiza si la coincidencia es total, pasando al array de palabras exactas o si es diferente pasando al de coincidencias.

Veamos un ejemplo en la siguiente figura en el que se ha buscado el término “woman”:

Resultado

woman : mujer

Palabras similares encontradas

businesswoman : mujer de negocios

chairwoman : presidenta

charwoman : asistenta

Englishwoman : inglesa

forewoman : encargada

horsewoman : amazona

kinswoman : parienta

policewoman : mujer policía

saleswoman : vendedora

sportswoman : mujer deportista

washerwoman : lavandera

womanhood : condición femenina

womanish : femenino

womanly : femenino

4.10.- COMUNICACIÓN ENTRE LOS USUARIOS

En un sistema en el que participan diferentes usuarios y donde pueden coincidir en el tiempo, es necesario la comunicación entre ellos. En Babel existen diferentes modos de comunicación:

- **Mensajería interna:** Cada usuario tiene su propio buzón de mensajes. Cualquier usuario registrado en el sistema puede enviar un mensaje privado a otro usuario.
- **Chat:** Existen diferentes salas de chat separadas de acuerdo al idioma nativo del usuario y al idioma que desea aprender. Así, la sala Español-Inglés, estará formada por personas cuyo idioma nativo sea el castellano y desean practicar inglés.
- **Comunicación por voz:** Desde el chat es posible la comunicación por voz entre los usuarios, pudiendo practicar conversación hablada.

4.10.1.- Mensajería interna

Cada usuario registrado en el sistema dispone de un buzón de mensajes privados. Una vez identificado el usuario, el sistema le avisará del número de mensajes nuevos de los que dispone.

El usuario a través de un lista ordenada alfabéticamente de usuarios puede enviar mensajes que llegarán al instante.

Los mensajes son almacenados en la tabla “**mensajes**” de la base de datos con los siguientes campos:

- **origen:** Usuario que envía el mensaje.
- **destino:** Usuario destino del mensaje.
- **mensaje:** Cuerpo del mensaje.
- **leído:** Booleano que nos informa si el mensaje ya ha sido leído con anterioridad.
- **fecha:** Fecha en la que se envió el mensaje.

Cada vez que se envía un nuevo mensaje, se crea un registro en la tabla.

4.10.2.- Chat

El chat implementado permite la comunicación entre usuarios de diferentes nacionalidades en tiempo real. Existen diferentes salas, una por cada combinación de idiomas posible.

Los elementos que forman el chat son los siguientes:

- **Lista de usuarios** conectados a la sala. Para cada usuario se muestra su idioma nativo representado mediante una bandera, el login o nick que posee en el sistema y si tiene cuenta de usuario Skype se mostrará un icono con el que se podrá llamar al usuario.
- **Panel de mensajes**, donde se muestran los mensajes escritos por los usuarios conectados.
- **Barra de selección del color** de la fuente de texto.
- Caja de texto para la **escritura de mensajes**.
- Caja de texto para realizar **correcciones**. Es posible seleccionar parte del texto escrito por otro usuario y realizar una corrección, que posteriormente se mostrará de forma destacada. Cualquier usuario podrá corregir errores cometidos por otros usuarios en conversación escrita.



Figura 4.19 Ejemplo de corrección realizada en el chat

- Los usuarios pueden consultar los **diccionarios** mientras mantienen conversación con el resto de integrantes de la sala.

Uno de los objetivos que se pretendía conseguir, es que el chat pudiera utilizarse en diferentes navegadores sin la necesidad de la instalación de algún plugin adicional. Por ello ha sido implementado con PHP y la generación de documentos HTML.

Tanto la lista de usuarios como los mensajes de la sala están contenidos en un IFRAME. Con la utilización de IFRAMES es posible incluir páginas dentro de otras páginas. Ambas se van actualizando de acuerdo a un tiempo establecido, por ejemplo, la lista de usuarios del canal cada diez segundos y los mensajes cada cinco. Estos valores son configurables desde la administración del sistema.

Los usuarios online y los mensajes de cada sala son almacenados en la base de datos. En el iframe correspondiente a los usuarios de la sala se cargará el documento php, encargado de acceder a la base de datos y obtener los usuarios online de la sala, y en el iframe de los mensajes, el que se encarga de recuperar los mensajes de la sala.

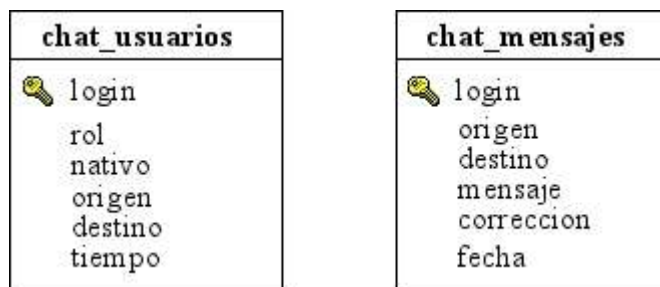


Figura 4.20 Estructura de tablas de la base de datos para el Chat

La tabla **chat_usuarios** almacena los usuarios online de las salas. Los campos que la forman son los siguientes:

- **login**: Login del usuario.
- **rol**: Rol del usuario.
- **nativo**: Idioma nativo del usuario.

- **origen**: Idioma origen de la sala.
- **destino**: Idioma destino de la sala.
- **tiempo**: Cuando se actualiza la lista de usuarios, cada usuario que se encuentra en la sala actualiza este tiempo. Existe entre los parámetros de configuración del sistema, un tiempo máximo de inactividad en las salas de chat. Si un usuario ha salido de la sala sin pulsar el botón de desconexión, cuando supere este tiempo de inactividad será eliminado de la lista.

La tabla **chat_mensajes** es la utilizada para almacenar los mensajes enviados por los usuarios. Un mensaje está formado por el **login** del usuario que lo envía, idiomas **origen** y **destino** para determinar en que sala debe ser mostrado el mensaje, **corrección** (si no se ha realizado ninguna corrección este campo estará vacío) y la **fecha** en la que se envió el mensaje.

Puede darse el caso de que el número de usuarios en las diferentes salas sea bastante elevado, por lo que el número de mensajes almacenados crezca rápidamente perjudicando al rendimiento. En la configuración del sistema es posible determinar el número de mensajes máximo por sala que se puede almacenar en la base de datos. Cuando se ha superado el doble de este valor se eliminan los mensajes más antiguos de la base de datos. Se realiza la operación cuando llega al **doble** por el siguiente motivo: Supongamos que hemos elegido en administración que se muestren 30 mensajes en la sala y a partir de ahí se borren. Cuando llegue a 31 mensajes, si no fuera el doble, se borraría uno, quedando 30 de nuevo, por tanto, a partir de esta cifra cada nuevo mensaje supondría una operación en la base de datos.

El control de la barra de scroll de la lista de mensajes supuso en un principio un problema. Recordemos que los mensajes se listaban en un documento. php que a su vez se cargaba en un IFRAME de la página principal. Cada vez que los mensajes eran actualizados la barra de scroll volvía a su posición inicial, esto era un impedimento para que el usuario pudiera ver los últimos mensajes escritos, o si se encontraba en la situación en la que hubiera movido la barra de scroll para leer un mensaje anterior.

Con el uso de JavaScript es posible determinar la posición de la barra de scroll de un iframe. Si un usuario ha escrito un mensaje o no ha movido de forma manual la barra

de scroll, esta siempre se mostrará bajada para poder ver los últimos mensajes. Si el usuario la ha movido hacia arriba para poder leer mensajes anteriores, se guarda la posición de la barra para que tras la actualización respete la posición anterior.

El sistema proporciona a través del chat, una buen medio para la comunicación entre usuarios de diferentes nacionalidades en tiempo real, pudiendo practicar conversación escrita y depurando errores con la corrección y colaboración de los participantes.

4.10.3.- Comunicación por voz

Skype es una aplicación que permite la comunicación por voz ip, con posibilidad de añadir plugins para videoconferencia. La calidad que ofrece es excelente, independientemente de la distancia a la que se encuentren los participantes de la conversación. La comunicación se puede establecer entre dos o más usuarios al mismo tiempo. Actualmente es utilizada por más de cinco millones de personas.

Debido a la calidad que ofrece indiferentemente de la distancia, es un medio excelente para que los usuarios conectados al chat de Babel puedan comunicarse por voz en tiempo real y practicar de esta forma conversación hablada.

Los usuarios desde su cuenta personal, tienen la posibilidad de incluir su nombre de usuario Skype. Aquellos usuarios que tengan cuenta en esta aplicación se mostrará un icono en el listado de usuarios del chat para que puedan llamarles.

Este icono será un enlace al url: `callto://nombre de usuario Skype`.



Figura 4.21 Ejemplo de usuario con posibilidad de hacer llamada con Skype

El protocolo `callto://` utilizado en el navegador, llama a un usuario de Skype siempre y cuando tenga la herramienta instalada y en funcionamiento.

4.11.- JUEGOS

Una forma amena de aprender, es hacerlo jugando. Para ello se han implementado dos juegos: “El ahorcado” y “El juego de las preguntas”

4.11.1.- El Ahorcado

El juego del Ahorcado consiste en intentar adivinar una palabra a partir de las letras que la forman.

Para tener un amplio número de palabras por idioma, se ha habilitado en la parte de administración de contenidos una interfaz, donde es posible especificar un texto y extraer todas las palabras de éste.

Las palabras son almacenadas en la tabla **ahorcado** de la base de datos, con los campos **palabra** e **idioma**.

Dependiendo del idioma que desee aprender el usuario, se tomará de forma aleatoria una de las palabras pertenecientes a ese idioma.

En el interfaz del juego se muestran tantas casillas como letras tenga la palabra y una caja de texto donde el usuario podrá realizar un nuevo intento escribiendo una letra.

Si la letra especificada por el usuario se encuentra en la palabra perteneciente a la solución, se mostrará en aquellas posiciones donde aparezca. Si por el contrario no existe se irá completando el dibujo clásico del ahorcado.

Una vez finalizado el juego, el sistema da la opción de poder buscar la palabra en el diccionario.

El juego y la obtención de la palabra ha sido implementado con PHP. Una vez cargado el documento en el navegador, la comprobación de las letras, visualización de letras acertadas, y resultado final del juego se realiza en el lado del cliente, con JavaScript.

4.11.2.- Juego de las preguntas

En el juego de las preguntas se reutilizan los ejercicios prácticos de la base de contenidos, tomando cada vez uno de ellos de forma aleatoria. El usuario podrá intentar resolverlo y el sistema realizará la corrección de forma automática. Una vez resuelto el ejercicio, es posible continuar tomando de nuevo de forma aleatoria otro ejercicio.

Los ejercicios que se seleccionan para el juego dependerán del par de idiomas que esté seleccionado. Si el usuario se encuentra en Alemán-Francés, los ejercicios mostrados en el juego, serán exclusivamente los creados para ese par de idiomas.

4.12.- SÍNTESIS DE VOZ

Un aspecto fundamental en el aprendizaje de idiomas, es conocer la pronunciación de las palabras. Para dotar al sistema de esta posibilidad, se ha utilizado la aplicación de carácter libre **Festival** desarrollada por la universidad de Edimburgo, que realiza la síntesis de voz a partir de un determinado texto.

Es necesario que **Festival** se encuentre instalado en el servidor y posea permisos para la ejecución por parte de cualquier usuario. La función **exec()** de PHP permite ejecutar comandos en el servidor.

Cuando un usuario desea escuchar la pronunciación de una determinada palabra, se crea en el servidor un archivo con formato WAV cuyo nombre coincide con el de su login. Se ejecuta de forma externa el comando `text2wave` (comando de festival para la generación de archivos wav con la síntesis realizada) y se devuelve al usuario el fichero de sonido.

```
$fp = fopen($usuario, "w");  
fwrite($fp, $palabra_voz);  
fclose($fp);  
exec("text2wave ".$usuario." -o
```


".\$usuario.".wav");

Si el usuario tiene instalado en el navegador alguna aplicación para la reproducción multimedia como QuickTime o RealPlayer se reproducirá el sonido automáticamente, sino se mostrará una ventana de descarga del archivo, para que el usuario pueda almacenarlo en su disco y poder reproducirlo con otra aplicación.

4.13.- USABILIDAD WEB

En el capítulo “*Antecedentes, estado de la cuestión*” se estudió la importancia de la usabilidad en los sistemas y algunos aspectos claves para intentar conseguirla.

Muchos de esos aspectos han sido tenidos en cuenta para que el sistema tenga un mayor grado de usabilidad.

- Saber de que trata el sitio a simple vista. Para ello se ha diseñado un logo con la frase “Free Languages Scholl” que se visualiza en todas las páginas. Además en la página principal del sistema se describe las posibilidades del usuario en el sistema.
- Utilización de **migas de pan**, para que el usuario se sienta localizado dentro del sistema en cualquier momento. Las migas de pan han sido implementadas mediante una **pila**. El sistema va apilando aquellos sitios por los que pasa. Cuando el usuario vuelve hacia uno de los sitios ya visitados extrae los elementos de la cima de la pila, hasta llegar al que ha sido elegido.
- Buscador de contenidos: Incluir un buscador en la aplicación proporciona una forma sencilla y rápida de acceder a la información.
- Utilización de pestañas en los contenidos. Los contenidos en el apartado de “Mis contenidos” y “Tareas pendientes” se encuentran clasificados de acuerdo al tipo de contenido y separados por pestañas.

- Diferenciación clara de las diferentes zonas del sitio.



Figura 4.22 Zonas del sitio web

- Se ha evitado la sobrecarga de información. Se ha prestado especial cuidado en mostrar la información más importante y necesaria.
- El sistema oculta aquella información que no es necesaria:
 - En el contexto de un determinado contenido, únicamente se muestran las acciones que puede realizar el usuario en ese momento, y no todas las acciones que puede realizar de acuerdo a su rol.
 - Tareas Pendientes: Se muestran únicamente aquellos contenidos sobre los que el usuario puede realizar una determinada acción.
 - Buscador de Contenidos. Al resultado de la búsqueda sólo pertenecen aquellos contenidos en los que se encuentra alguna de las palabras claves buscadas y además el usuario puede realizar alguna acción.

- Las imágenes utilizadas como botones, sobre las cuales es posible la interacción del usuario, poseen ayuda contextual. Cuando el usuario sitúa el puntero del ratón sobre la imagen aparece un texto aclarativo.

- Se ha evitado la inclusión de imágenes de mediano o gran tamaño, para evitar ruido visual y permitir la accesibilidad desde dispositivos de menor tamaño como puede ser un PDA,

- Finalmente podemos decir que cuando un usuario visita Babel puede responder a las preguntas planteadas por Steve Krug para la usabilidad de páginas web:
 - ◆ ¿Qué sitio es este?
 - El logo mostrado en todas las páginas y la descripción del sistema en la página principal y en las secciones de ayuda, facilitan al usuario la tarea de entender los fines a los que está orientado el sistema.

 - ◆ ¿ En qué página estoy? ¿Dónde estoy en el esquema de las cosas?
 - ◆ La utilización de migas de pan permite al usuario saber por donde se ha movido en el sistema y donde se encuentra actualmente.

 - ◆ ¿ Cuáles son las principales secciones del sitio ?
 - ◆ Las principales secciones del sitio quedan bien resaltadas en las pestañas centrales con los menús desplegables y el menú de la derecha con las diferentes herramientas.

 - ◆ ¿ Cómo busco algo ?
 - ◆ En el menú de la derecha se muestra siempre de forma permanente un buscador para la búsqueda de contenidos.

4.14.- MÉTODOS DE APRENDIZAJE

La mayoría de métodos de enseñanza de idiomas estudiados (véase 3.6), pueden ser puestos en práctica en Babel en gran medida.

El método gramática-traducción se centraba principalmente en el estudio correcto del lenguaje estudiando sus normas, dejando en un segundo plano la práctica del habla. La posibilidad de poder crear documentos teóricos, permite a los autores crear documentos con las diferentes reglas gramaticales de cada idioma, junto con los ejercicios prácticos y traducciones que ayudarán al usuario a aprender a escribir correctamente el idioma.

El método directo por el contrario se centraba en aprender a hablar correctamente un idioma sin entrar en detalle de las normas del mismo. La posibilidad de comunicación a través del chat por escrito y mediante voz, permite a los usuarios poner en práctica este método de aprendizaje.

Los métodos audio-oral y audio-visuales aplicaban las nuevas tecnologías. Consiste en el aprendizaje de un idioma con la repetición por parte del usuario de sonidos y asimilación de imágenes. En Babel se puede poner en práctica debido a que es posible la asociación de ficheros de vídeo y audio a los documentos.

En cuanto al método nocional-funcional recordemos que:

“Esas unidades del lenguaje se ordenan adecuadamente de acuerdo con las necesidades de comunicación de aquellos que aprenden un idioma. (En este sentido, se debe también tener en cuenta el aprendizaje de un idioma con finalidades especiales y concretas)”.

Los contenidos están ordenados de acuerdo a un nivel de idioma y a una serie de categorías que permite que el usuario pueda aprender de acuerdo a sus necesidades y el nivel del idioma que posea.

5.- RESULTADOS

5.1.- RESULTADOS OBTENIDOS

El resultado es el diseño y desarrollo de una herramienta de gestión de contenidos web, que sirve como apoyo para el aprendizaje y la práctica de idiomas.

Se han cumplido los objetivos que se plantearon inicialmente. El sistema posee una máquina de estados con la que es posible controlar la información almacenada en la base de contenidos y cuenta con herramientas de comunicación, chat, síntesis de voz, diccionarios y juegos que sirven de apoyo en el proceso de aprendizaje del usuario.

El sistema puede ser utilizado en diferentes máquinas con distintos tamaños y distintos sistema operativos.

Ha sido desarrollado con el lenguaje orientado a objetos PHP5 y bases de datos MySQL. Se encuentra alojado en un servidor Apache.

Además se ha realizado la integración de diferentes tecnologías como la utilización de Festival para la síntesis de voz, comunicación por voz entre usuarios con la aplicación Skype, Editor WYSIWYG Spaw implementado en PHP y la utilización de las librerías PDFlib para la generación de documentos PDF a partir de código HTML.

5.2.- POSIBLES COSTES DE IMPLANTACIÓN DEL SISTEMA

El estudio realizado, ha servido para analizar los posibles costes y ganancias que se obtendrían en el caso de poner el sistema en explotación. También ha influido en la decisión de las tecnologías utilizadas para el desarrollo.

5.2.1.- Costes anuales de hosting

A continuación se mostrarán dos tablas: una con plataforma GNU/Linux y otra con plataforma Windows, junto con las tecnologías que proporcionan para el desarrollo de web dinámicas y su coste anual.

	<i>Plataforma GNU/Linux</i>								
Fuente	Apache	Tomcat	PHP	JSP	MySQL	PostgreSQL	Espacio	Precio/año	
arsys.es	✓		✓		✓		400 MB	192,00 €	
abserver.es	✓		✓		✓		300 MB	200,00 €	
ferca.com	✓	✓	✓	✓	✓		100 MB	325,00 €	
claranet.es	✓		✓		✓		200 MB	447,75 €	
piensasolutions.com	✓		✓		✓		300 MB	95,40 €	
acens.com	✓		✓		✓		225	480,00 €	
oxxus.net	✓	✓	✓	✓	✓	✓	500	360,00 €	
javaservlethosting.com	✓	✓	✓	✓	✓		150	320,00 €	
visionwebhosting.com	✓	✓	✓	✓	✓		1,2 GB	156,00 €	
discountasp.net	✓		✓		✓			necesario contactar	
generalhosting.com									
bizhostnet.com									
activehost.com									
evidalia.com	✓		✓		✓		400 MB	168,00 €	
hosting.com	✓		✓		✓		1 GB	420,00 €	
jatol.com									
interadventure.com	✓	✓	✓	✓	✓	✓	1,5 GB	180,00 €	
mochahost.com	✓	✓	✓	✓	✓		1 GB	120,00 €	
weblinet.com	✓	✓	✓	✓	✓	✓	500 MB	420,00 €	
winwebhosting.com	✓		✓		✓		500 MB	95,00 €	
mypagehost.com	✓	✓	✓	✓	✓	✓	2 GB	79,00 €	
javapipe.com	✓	✓	✓	✓	✓	✓	500 MB	360,00 €	
dhapcenter.es	✓		✓		✓	✓	300MB	44,00 €	

Tabla 5.1 Costes de Hosting en plataforma GNU/Linux

Fuente	Plataforma Windows												Espacio	Precio
	Win. 2003 Server	ODBC	ISS	MSQL	PHP	Tomcat / JSP	ASP. NET	ASP	MySQL	Access	Postgre			
arsys.es	✓							✓		✓		600 MB	348 €	
abserver.es														
ferca.com	✓	✓	✓	✓	✓	✓	✓	✓	✓			450 MB	395 €	
claranet.es														
piensasolutions														
acens.com	✓	✓		✓			✓	✓		✓		225 MB	480 €	
oxxus.net														
javaservlethosting														
visionwebhosting														
discountasp.net		✓	✓	✓	✓		✓	✓	✓	✓				
generalhosting	✓				✓		✓	✓	✓	✓			200 €	
bizhostnet.com	✓				✓		✓	✓	✓	✓		20 GB	100 €	
activehost.com	✓				✓		✓	✓	✓	✓		200 MB	300 €	
evidalia.com														
hosting.com	✓			✓	✓		✓	✓	✓	✓		1 GB	420 €	
jatol.com														
interadventure		✓	✓	✓	✓	✓	✓	✓	✓	✓		1 GB	480 €	
mochahost.com														
weblinet.com														
winwebhosting	✓			✓	✓	✓	✓	✓	✓	✓		500 MB	130 €	
mypagehost.com														
javapipe.com														
dhapcenter.es	✓				✓		✓	✓	✓	✓	✓	600 MB	44 €	

Tabla 5.2 Costes de Hosting en plataforma Windows

De estas dos tablas es posible extraer las siguientes conclusiones:

- La mayoría de servidores proporcionan soporte para PHP y MySQL tanto en plataformas GNU/Linux como Windows.
- El número de servidores para alojar aplicaciones realizadas en Java y JSP no es muy numeroso.
- Los lugares de hosting donde es posible elegir entre servidores GNU/Linux y Windows, los precios suelen ser más bajos en el caso de servidores GNU/Linux, no pudiendo alojar aplicaciones desarrolladas en ASP o ASP.NET.

- Media aritmética de los costes en servidores GNU/Linux: 247,89 €
- Media aritmética de los costes en servidores Windows: 289,7 €

5.2.2.- Costes de alquiler de un servidor

Otra de las posibilidades que existen a la hora de poner el sistema en explotación es la de alquilar un servidor, pudiendo tener de esta forma más libertad en las tecnologías utilizadas.

A continuación se muestra una tabla donde se indican algunos sitios de la red en los que es posible el alquiler de servidores, las características del mismo, y su coste si se trata de una plataforma GNU/Linux o Windows.

<i>Fuente</i>	<i>Servidor</i>	<i>GNU/Linux</i>	<i>Windows</i>
centrodedatos.com	P4. 3 Ghz, 1024 MB de memoria RAM y 160 GB de espacio	1.428,00 € /año	1.428,00 € / año
serveisweb.com	P4. 2,8 Ghz, 2 memorias RAM de 512 MB, 80 GB de espacio	653 € /año	763 € /año
ram.es	P4 2,8 Ghz, 512 RAM, 120 GB de espacio	588 € /año	1188 € / año
ferca.com	Intel Celeron 2,4 Ghz, 512 RAM, 80 GB de espacio	1296 € /mes	1621 € / año
arsys.es	P4. 2,8 Ghz, 512 RAM, 80 GB de espacio.	1.188 € /año	1.188 € / año
evidaliahost.com	Pentium Celeron 2 Ghz 1 GB de RAM, 160 GB de espacio.	2340 € / año	no posee máquinas con Windows.

<i>Fuente</i>	<i>Servidor</i>	<i>GNU/Linux</i>	<i>Windows</i>
simbionet.com	P3. 733 Mhz, 256 RAM, 10 GB de espacio	420 € /año	no posee máquinas con Windows.
acens.com	P4. 3,2 Ghz 512 de RAM, 80GB de espacio.	3.000 € / año	3.588 € / año
sync.es	P4. 2,8 Ghz, 512 de RAM, 80 GB de espacio.	2.148 € /año	2.688 € / año

Las conclusiones que podemos obtener de este punto son las siguientes.

- La más destacable y que es posible ver a simple vista, es que el coste de alquilar un servidor con GNU/ Linux es menos costoso que uno con tecnología de Microsoft.
- El coste de desarrollar la aplicación en PHP o JSP funcionando es un servidor GNU/Linux es menor que si se desarrolla en ASP o ASP.NET alojándola en un servidor con Windows.
- Media aritmética de los costes para servidores GNU/Linux: 1451,22 €
- Media aritmética de los costes para servidores GNU/Linux: 1780,57 €
- El coste de alquiler de un servidor es elevado sobre todo si tenemos en cuenta que sólo es necesario alojar una aplicación. Para poner el sistema en explotación quizás sería conveniente un servicio de hosting en lugar de un servidor dedicado.

5.2.3.- Coste de adquisición de un servidor

Se puede tener en cuenta la posibilidad de comprar un servidor para alojar la aplicación. Al precio del servidor habría que sumar el coste de la línea y el de energía eléctrica.

A continuación se muestra un listado de precios de servidores con una serie de características que creemos que son suficientes para el funcionamiento eficiente de la

aplicación.

<i>Modelo</i>	<i>Procesador</i>	<i>Memoria RAM</i>	<i>Espacio</i>	<i>Precio</i>
Power Edge SC430	P4 2,8 Ghz	512 MB	80 GB	359 euros
HP Proliant ML350 G3 XCOM	P3 2,8 Ghz	512 MB	80 GB	789 euros
Xseries 206	p4 3,2 Ghz	256 MB	60 GB	600 euros
Tyan Computer. Copr Transport GX28-W Dual	AMD Opteron 2,8 Ghz	512 MB	160 GB	879,55 euros
HP Proliant ML110G2 Tower Server	2,8 Ghz Celeron	256 MB	80 GB	496,44 euros
HP Compact Proliant DL 380	P3- 1 Ghz	128 MB	60 GB	224 euros
IBM NetFinity 5500 M20	P3 500 Mhz	256 MB	60 GB	602,95 euros
Dell PowerEdge 4400	P3 953 Mhz	512 MB	80 GB	467,50 euros.
IBM X Series X235 Xeon	P4 2,4 Ghz	256 MB	80 GB	767 euros
Monarch Computer Empro series 1U Rackmount Server	AMD Opteron 1,4 Ghz	512 MB	80 GB	1153 euros

5.2.4.- Decisiones de las tecnologías utilizadas

Para desarrollar el sistema, era necesario un lenguaje de programación que permitiera la construcción de sitios web dinámicos. El diseño que se quería realizar era un diseño multicapa, pudiendo separar de forma sencilla las capas de presentación, dominio y persistencia.

Entre los lenguajes posibles se encontraban PHP4, PHP5, ASP, ASP.NET y JSP. ASP quedaba descartado debido a que la separación en capas no era una tarea sencilla.

PHP4 posee una orientación a objetos muy básica por lo que también quedaba descartado.

A la hora de elegir entre JSP, PHP5 y ASP.NET influyó el análisis de costes realizado. Los costes de implementar la aplicación en ASP.NET son bastante elevados en cuanto a costes de hosting y licencias, por tanto la elección se realizaría entre PHP5 y JSP.

El coste de implementar la aplicación en alguno de estos dos lenguajes es nulo. PHP5 proporciona grandes ventajas sobre PHP4, ya que al ser altamente orientado a objetos permite la separación clara en diferentes capas. PHP5 es de carácter libre, la disponibilidad del código fuente permite la optimización del código, ya que se ha ido mejorando por parte de los usuarios, obteniendo de esta forma un alto rendimiento. El sistema desarrollado en este proyecto fin de carrera, es un sistema web. PHP dispone de un número elevado de funciones que facilitan y simplifican el desarrollo en este tipo de aplicaciones.

A estas razones debemos sumar que poseía un nivel alto de conocimiento del lenguaje, y me encontraba familiarizado con algunas librerías externas que se han utilizado en el proyecto, como por ejemplo, la generación de documentos PDF o editores WYSIWYG para la creación de texto enriquecido. Estos son los motivos por lo que se ha implementado la aplicación en el lenguaje PHP5.

En cuanto a la base de datos utilizada (MySQL) , según se ha podido observar la mayoría de servidores proporcionan soporte para bases de datos MySQL. PHP permite una forma de comunicación sencilla y eficiente entre las aplicaciones y bases de datos MySQL.

La elección del servidor ha sido Apache. Debido a que es un servidor que no supone costes ,es multiplataforma y proporciona un alto rendimiento.

Si el sistema se pusiera en explotación la decisión de las tecnologías utilizadas supondría un bajo coste, ya que tan sólo sería necesario costes de hosting, pudiendo obtener así unos mayores beneficios.

5.2.5.- Mecanismos de financiación del sistema

Se ha pensado en diferentes formas de costear el alojamiento del sistema en un servidor y obtener posibles beneficios.

- **Publicidad.** Incluir un sistema en el que una imagen en la cabecera y otras de menor tamaño en el lateral derecho vayan rotando cada vez que se cargue una nueva página. Debido a que el sistema está orientado al aprendizaje de idiomas, academias y agencias de viajes podrían estar interesadas.

Del análisis de costes de hosting para servidores GNU/Linux, con soporte para PHP, y MySQL obteníamos que la de menor coste era de 44 € al año, la media de 247,89 € y la de mayor coste 480 €.

Veamos en cada uno de los casos a partir de que momento se obtendrían beneficios, teniendo en cuenta una imagen publicitaria en la cabecera y tres más pequeñas en la parte de la derecha.

- 44 € al año.
 - Imagen de la cabecera: 20 €/año por anunciante.
 - Imagen del lateral: 8 €/año.
 - A partir de una imagen en la cabecera y tres laterales se podría comenzar a obtener beneficios.
- 247,89 al año:

Si establecemos los precios en la misma proporción que la situación anterior:

 - Imagen de la cabecera: 112,67 €/año.
 - Imagen lateral: 45 €/año.
- 480 € al año.
 - Imagen de la cabecera: 218,18 €/año.
 - Imagen lateral: 87,27 €/año.

- **Modificaciones a medida.** Si alguna organización estuviera interesada en la implantación del sistema, podría solicitar modificaciones a medida, adaptando las herramientas a sus necesidades. El coste de cada modificación vendría dado por: *Horas de trabajo empleadas x el coste por hora.*
- **Venta del código fuente.** La totalidad del código fuente podría ser vendido por un valor comprendido entre 5000 y 8000 euros.
- **Compra por contenidos.** Se podría ampliar la funcionalidad del sistema en un futuro, para poder crear contenidos por cuya adquisición sea necesario realizar un previo pago según la cantidad que se indique.

6.- CONCLUSIONES Y PROPUESTAS

6.1.- CONCLUSIONES

En la mayoría de países se destaca la importancia de aprender uno o dos idiomas además del que aprendemos en el lugar en el que nacemos, por las ventajas que proporciona tanto a nivel personal como profesional.

Unos años atrás tan sólo era posible el aprendizaje de idiomas en centros especializados para tal fin, mediante libros o viajando al lugar donde se hablaba la lengua que deseabas aprender. La aparición de nuevas tecnologías e Internet como medio de difusión y comunicación, abre un amplio abanico de posibilidades en el proceso de aprendizaje de idiomas.

La mayoría de sistemas dedicados al aprendizaje de idiomas que existen en la red suelen contar con una serie de contenidos fijos, o bien son actualizados por los administradores del sitio cada cierto tiempo, no pudiendo contribuir los usuarios a la ampliación de la base de contenidos. Con la creación de Babel, se proporciona una herramienta de apoyo en el aprendizaje de idiomas, donde los usuarios pueden compartir sus conocimientos y participar de forma colaborativa en el aprendizaje de idiomas del resto de usuarios.

Una de las grandes ventajas que proporciona es su base de contenidos compartida. Usuarios de diferentes nacionalidades podrán crear documentos, ejercicios prácticos, traducciones ... que servirán de ayuda a otras personas.

El hecho de que los usuarios puedan contribuir con sus conocimientos, provocará que la base de contenidos alcance un tamaño considerable en poco tiempo, pudiendo llegar a convertirse en una gran enciclopedia para el aprendizaje de idiomas.

La máquina de estados que regula el comportamiento en el proceso de adquisición y

tratamiento de la información, permite que el sistema se adapte a las necesidades y preferencias del lugar donde sea implantado.

El sistema además está apoyado por herramientas de comunicación, diccionarios, síntesis de voz, buscador de contenidos y juegos que hacen de él un sistema más completo.

El diseño modular que se ha realizado, permitirá de forma sencilla añadir más tipos de contenidos y nuevos módulos en el futuro.

6.2.- PROPUESTAS Y LÍNEAS FUTURAS

Son varias las mejoras que se podrían realizar en el sistema. Con el diseño realizado, estas mejoras no supondrían grandes cambios en el código fuente. A continuación pasaremos a analizar algunas de ellas.

- Creación del tipo de contenido Curso. Al igual que en los exámenes es posible la agrupación de diferentes ejercicios prácticos, sería interesante tener la posibilidad de poder crear cursos. Estos estarían formados por una agrupación de los distintos tipos de contenidos, pudiendo contener un curso diferentes documentos teóricos, ejercicios y exámenes.
- Ampliar las modalidades de ejercicios prácticos. Por ejemplo, con ejercicios en los que se relacionan palabras con imágenes.
- En el tipo de ejercicios en el que hay que completar espacios en blanco, incluir la posibilidad de definir pistas, de tal forma que el usuario que está realizando el ejercicio pueda obtener algún tipo de ayuda extra.
- Creación de nuevos juegos como pueden ser: Sopa de letras, crucigramas y text twist en el que se muestran un número de palabras aleatorias con las que el usuario debe formar el mayor número de palabras posibles.
- Módulo de Viajes. En un sistema frecuentado por usuarios de diferentes países,

sería interesante incluir un módulo de viajes, donde los usuarios puedan exponer sus experiencias vividas en otros países. En este módulo también podría barajarse la posibilidad de que agencias de viajes estuvieran interesadas en la publicación de ofertas.

- Módulo de costumbres y culturas. En este módulo, los usuarios de diferentes países podrían escribir acerca de las costumbres típicas de su país y temas relacionados con su cultura. Esto permitiría al usuario aprender otras culturas aparte del idioma.
- Representar gráficamente la máquina de estados, facilitaría al usuario ver la configuración actual a simple vista. Existe en PHP la librería GD library a través de la cual es posible la generación dinámica de imágenes.

Javier Alonso Albusac Jiménez

7.- BIBLIOGRAFÍA

- [AQU82]: Aquilino Sánchez Pérez, "*La Enseñanza de Idiomas. Principios, problemas y métodos*", 1982, HORA, S.A. Barcelona
- [CAS99]: Ester del Castillo, Material docente de la asignatura de Autómatas y Lenguajes Formales, 1999
- [COL03]: Colado, "*C.Diseño y desarrollo de aplicaciones web multidispositivo*", 2003, Germinus XXL
- [COU01]: Adrián Coutin Domínguez, "*Arquitectura de información de sitios web*", 2001, Anaya Multimedia
- [CSS03]: Owen Briggs, "*CASCADING STYLE SHEETS*", 2003, Anaya Multimedia
- [DOM04]: Eduardo Domínguez, Mario Piattini, Material docente de la asignatura Ingeniería del Software I, 2004,
- [DUB05]: Paul Dubois, "*MYSQL*", 2005, Anaya Multimedia
- [LAC99]: Carmen Lacave, Juan Giralt, Material docente de la asignatura Estructura de Datos, 1999
- [GAL04]: Juan Diego Gutiérrez Gallardo, "*Desarrollo web con PHP 5 y MySQL*", 2004, Anaya Multimedia
- [GRO00]: Andree Growney, Joe Burns, "*JAVASCRIPT*", 2000, Pearson Educación
- [KOC00]: N. Koch, "*Comparing Development Methods for Web Applications*", 2000, Anaya Multimedia
- [KRU03]: Steve Krug, "*No me hagas pensar*", 2003, Prentice Hall
- [POL04] Macario Polo Usaola, Material docente de la asignatura Ingeniería del Software II, 2004,
- [NIE02]: Jacob Nielsen, "*Usabilidad. Jacob Nielsen 50 sitios web*", 2002, Prentice Hall
- [NIE_02]: Jakob Nielsen, "*Usabilidad. Diseño de sitios web*", 2002, Prentice Hall
- [PRI04]: M.Prieto, Material docente de la asignatura Sistemas Colaborativos, 2004
- [RIC98]: Jack C. Richards, Theodore S. Rodgers, "*Enfoques y métodos en la enseñanza de idiomas*", 1998, CAMBRIDGE
- [RUI00]: Francisco Ruiz, M.Ortega, Material docente de la asignatura Bases de Datos, 2000

- [RUI01]: Julian Ruiz, Material docente de la asignatura Programación Concurrente, 2001
- [SKL05]: David Sklar, "*Introducción a PHP5*", 2005, Anaya Multimedia
- [SUH03]: Suh, P, "*Content Management Systems*", 2003, Glasshaus
- [THO03]: Laura Thomson, Luke Welling, "*Desarrollo web con PHP y MySQL*", 2003, Anaya Multimedia
- [TYP04]: Typo 3, Typo3 Documentation, 2004
- [ZOP04]: Zope, Libro de Zope, 2004

Referencias Web

- [APA]: Servidor Apache, <http://www.apache.org>
- [ECM]: Portal de gestión documental España, <http://www.ecm-spain.com>
- [INT]: Introducción a zope + apache + bases de datos relacionales, <http://aditel.org/jornadas/02/ponencias/zope>
- [MLG]: Metodos de enseñanza de idiomas y tipos de ejercicios prácticos, www.cepmalaga.com/revistas/algarabia/pdf/005_expl.pdf
- [SQL]: Página oficial de MySQL, <http://www.mysql.com>
- [PAG]: Paginación de contenidos con PHP, <http://jpinedo.webcindario.com/scripts/paginator/>
- [PDF]: Generación de documentos PDF, <http://www.pdflib.com/>
- [PHP]: Página oficial de PHP, <http://www.php.net>
- [SPA]: Editor WYSIWYG Spaw, <http://www.solmetra.com/>
- [SKY]: Pagina oficial de la aplicación Skype, que permite la comunicación por voz ip, <http://www.skype.com>
- [TYP]: Página oficial de Typo3, <http://www.typo3.org>
- [VILL]: Razones para implantar gestión de contenidos en la empresa, http://www.alzado.org/articulo.php?id_art=130
- [WIK]: Enciclopedia libre Wikipedia, <http://es.wikipedia.org>
- [ZOP]: Página oficial de Zope, <http://www.zope.org>

8.- ANEXOS

8.1.- ANEXO A – MANUAL DEL USUARIO

En este anexo se describirá brevemente la funcionalidad del sistema mediante un manual de usuario.

Un usuario anónimo únicamente puede leer los contenidos almacenados en el sistema y utilizar los diccionarios. Es importante registrarse en el sistema, para obtener una cuenta de usuario y poder realizar un mayor número de actividades.

8.1.1.- Cuenta de usuario



Figura 8.1 Creación de una nueva cuenta de usuario

Para crear una nueva cuenta de usuario, nos dirigimos a la parte superior derecha para hacer clic sobre la palabra “Login” o en el panel de la cuenta de usuario donde pone “Haz login ahora”.

De este modo accedemos al panel de identificación del sistema, donde es posible

crear una nueva cuenta de usuario, activar la cuenta u obtener una nueva contraseña en el caso de que la hayamos olvidado.

Para crear una nueva cuenta de usuario realiza clic sobre “*Si aún no tiene cuenta, puede registrarse ahora*”.

A continuación, se deberá rellenar los datos de usuario necesarios para la creación de una nueva cuenta. Una vez finalizado se enviará un correo electrónico al usuario notificándole su registro y el código de activación que debe utilizar para activar la cuenta.

Para activar la cuenta de usuario, nos dirigimos al mismo lugar en el que creamos la cuenta, y pulsamos sobre “*Activación de la cuenta de usuario*”, donde deberemos introducir el login con el que nos hemos registrado y el código de activación que hemos recibido en el correo electrónico.

Cuando un usuario se identifica en el sistema el aspecto del panel de la cuenta de usuario varía:



Figura 8.2 Cuenta de usuario

Se muestra el login del usuario junto con el avatar asociado, el rol que posee en el sistema, el acceso al buzón de mensajes y número de mensajes nuevos, se da la posibilidad de cambiar los datos de la cuenta de usuario, acceso a “Mis contenidos” y “Tareas pendientes”.

Mis contenidos

Se muestra los datos del usuario junto con los puntos y el nivel que posee en cada idioma. El usuario puede interactuar con todos los contenidos en los que es autor. Cada tipo de contenido está separado en su pestaña correspondiente, realizando clic sobre la pestaña podremos acceder a los contenidos de ese tipo.

Para cada uno de los contenidos se muestra el título, la fecha de cuando se realizó la última actualización, el par de idiomas al que pertenece, el estado en el que se encuentra, la posibilidad de añadir o eliminar autores del contenido y si el documento se encuentra bloqueado por alguno de los autores (cualquier acción que pueda modificar el contenido bloquea el recurso).

The screenshot displays the 'Mis contenidos' (My contents) interface. At the top, there is a header with the text '» Mis contenidos' and a small icon. Below this, a section titled '» Datos de Usuario' (User Data) is shown, containing the following information:

- Login:** revisor
- Rol:** Reviewer
- Nombre completo:**
- Fecha de creación:** 2005-10-23
- Nivel y puntos:** 110 Puntos (Nivel 1)

Below the user data, there are several tabs for content types: 'Documentos teóricos', 'Ejercicios prácticos', 'Exámenes', 'Traducciones', and 'Correcciones'. The 'Documentos teóricos' tab is currently selected and highlighted in yellow.

At the bottom, a table lists the content items. The table has the following columns: 'Título', 'Actualizado', 'Idioma', 'Estado', 'Autores', and 'Bloqueo'. The first row shows a content item with the following details:

Título	Actualizado	Idioma	Estado	Autores	Bloqueo
1 Reglas de construcci...	2005-11-21 16:22:31	ES ES	New	+ -	🔒

Figura 8.3 Mis contenidos

Tareas pendientes.

En las tareas pendientes se muestran aquellos contenidos del sistema que están pendientes de realizar alguna acción. Dependiendo del rol de usuario, el número de tareas pendientes será mayor o menor. Un rol con mayor responsabilidad tendrá un mayor número de tareas pendientes.

Las tareas pendientes están ordenadas de forma ascendente respecto de la fecha de actualización, de tal forma que los contenidos que se actualizaron anteriormente se muestran primero para que sean atendidos antes. Para atender las tareas pendientes, no es necesario respetar este orden, puede ser resuelta cualquiera de ellas.

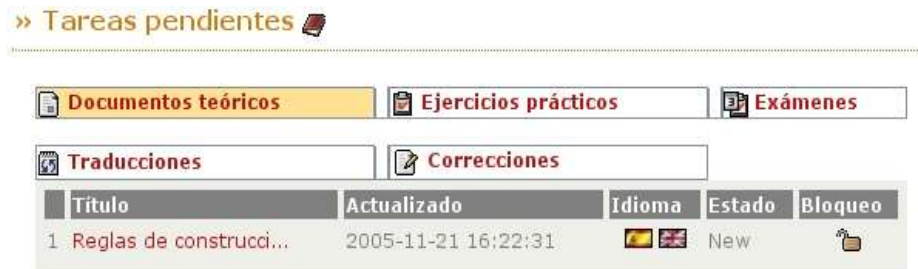


Figura 8.4 Tareas pendientes

Mensajes privados

Si pulsamos sobre el enlace “Mensajes” de la cuenta de usuario, accedemos al buzón del usuario, donde se listan los mensajes privados. Al lado de cada uno de los mensajes, aparece un icono que representa un sobre cerrado si no ha sido leído con anterioridad o abierto si ya ha sido leído. Para leer un mensaje basta con realizar clic sobre el login del autor.

Para enviar un nuevo mensaje, hacemos clic sobre el icono del sobre que existe en el panel de la cuenta de usuario. Elegimos el destino en el listado de usuarios ordenados alfabéticamente, escribimos el mensaje y pulsamos el botón de enviar.

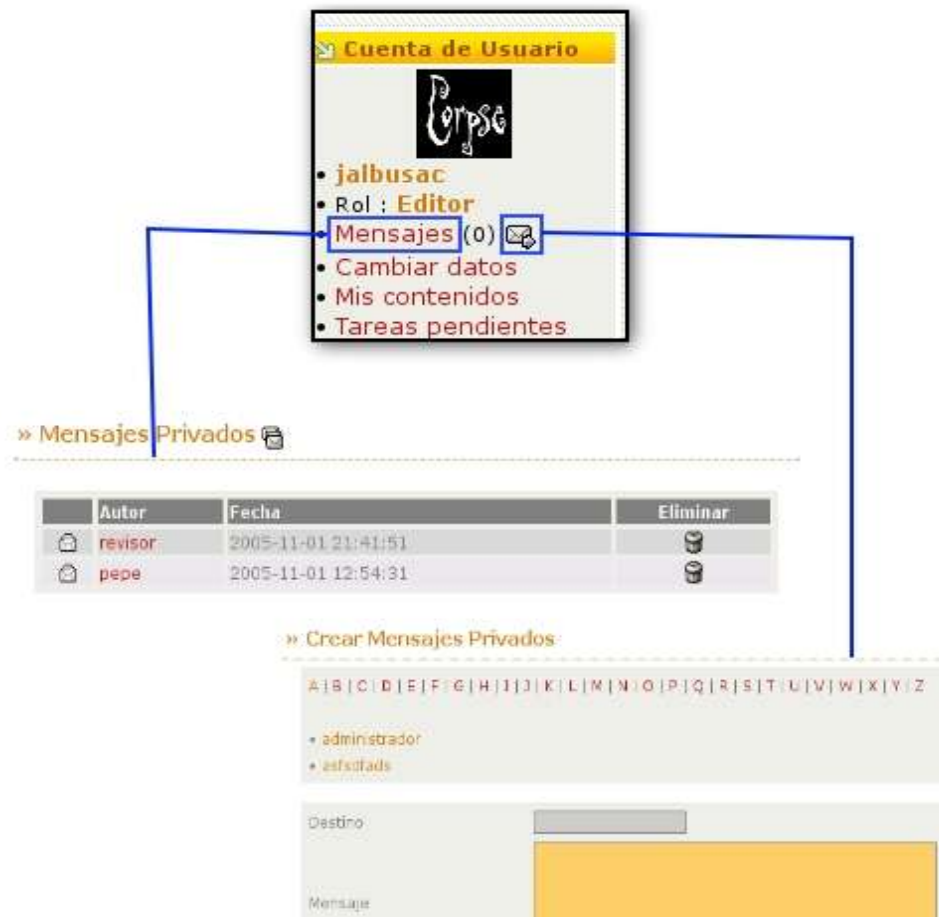


Figura 8.5 Buzón de mensajes

8.1.2.- Herramientas

- Buscador de contenidos.
- Diccionarios y pronunciación de palabras.
- Juegos: Ahorcado y juegos de preguntas.
- Chat.

El usuario dispone de un **buscador de contenidos** con el que puede acceder a la información a partir de un conjunto de palabras claves.



Figura 8.6 Buscador de contenidos

Se puede limitar la búsqueda a un tipo de contenido específico con la utilización del filtro.

Con la utilización de los **diccionarios** el usuario puede buscar el significado de las palabras. Los diccionarios pueden ser utilizados por usuarios anónimos y usuarios registrados a los que además se le da la posibilidad de escuchar la pronunciación de la palabra pulsando sobre el icono que representa un altavoz.



Figura 8.7 Diccionarios

Los diccionarios pueden ser accedidos desde el menú de la derecha, en la creación de contenidos y desde el chat.

El sistema dispone de dos **juegos** con los que el usuario puede aprender jugando. Son accesibles desde la página principal.

- El Ahorcado. El usuario debe intentar adivinar una palabra del idioma que desea aprender. Se muestra una caja de texto por cada una de las letras que forman la palabra, y otra caja para probar una nueva letra en cada turno.

» Juegos Ahorcado



Si el dibujo se completa antes de que hayamos adivinado la palabra perdemos.

- Juego de las preguntas. Se muestra un ejercicio aleatoriamente de la base de contenidos, que el usuario deberá intentar resolver.

El sistema dispone de un chat para la comunicación en tiempo real entre los usuarios. Esta dividido en diferentes salas, tantas como combinaciones de idiomas existen. Por ejemplo la sala Español – Inglés estará formada por usuarios que hablen castellano y quieran aprender inglés.

El chat consta de :

- Lista de usuarios conectados a la sala.
- Panel donde se muestran los mensajes.
- Diccionarios y barra de colores para la selección del color del texto.
- Caja de texto para escribir los mensajes.
- Área donde copiaremos el texto que deseemos corregir.



Figura 8.8 Chat

Para realizar una corrección, seleccionamos el texto que deseamos corregir y pulsamos el botón copiar (también es posible seleccionarlo y arrastrarlo al área de correcciones), una vez copiado escribimos en la caja destinada a los mensajes la modificación y pulsamos enviar.

La comunicación por voz entre usuarios es posible a través de la aplicación Skype [SKY] que permite conversaciones mediante voz ip a una gran calidad independientemente del lugar en el que se encuentren los usuarios. Los usuarios que posean cuenta con Skype pueden especificarla en los datos de la cuenta de usuario para que cuando se conecten al chat, aparezca al lado de su login un icono verde con un teléfono. Para poder llamar al usuario y establecer conversación es necesario que la aplicación Skype esté funcionando y pulsar sobre el icono verde.

8.1.3.- Selección de idiomas

Para seleccionar el idioma de la interfaz, el usuario debe dirigirse a la parte superior de la página donde se encuentran las banderas que representan los distintos idiomas en los que la interfaz está disponible. Pulsando sobre cada una de ellas el idioma de la interfaz varía.

Los contenidos del sistema, así como los juegos y el chat están ordenados de acuerdo a un par de idiomas: Idioma origen e idioma destino. El idioma origen se corresponde al idioma que domina el usuario y el destino el que desea aprender.

Es posible seleccionarlo en el menú despegable de idiomas activos.



Si por ejemplo seleccionamos Spanish-English:

- Los contenidos que se visualizarán serán los que pertenezcan a este par de idiomas.
- La sala de Chat a la que podremos acceder será la de Español-Inglés.
- Los diccionarios sobre los que se realizarán la búsqueda son diccionarios Inglés-Español o Español-Inglés.
- El juego del ahorcado tomará aleatoriamente palabras en inglés.
- El juego de preguntas tomará ejercicios prácticos pertenecientes a Español-Inglés.

El idioma de la interfaz y el par de idiomas seleccionado para los contenidos se encuentran en las pestañas de la parte superior.



8.1.4.- Administración del sistema

Inicialmente el único usuario que puede realizar tareas de administración del sistema es el que posea el rol de “**Administrator**”.

Un administrador se dedica exclusivamente a tareas de administración. Cuando este se identifica en el panel de la cuenta de usuario en lugar de mostrar “Mis contenidos”, “Tareas pendientes” ... aparece un enlace a administración.

» Panel de Administración



Figura 8.9 Panel de administración

Desde el panel de administración se pueden realizar la siguientes actividades.

- Buscador. Desde esta sección es posible actualizar los índices que utiliza el buscador para realizar las búsquedas.
- Gestión de usuarios. Modificación de las cuentas de usuario. Desde aquí es posible modificar el rol de cada usuario.
- Diccionarios. Ampliación de términos en los diccionarios. Se especifica un diccionario en formato de texto plano y el sistema extraerá los términos para ampliar la base de contenidos.
- Gestión de idiomas. Activación y desactivación de los idiomas activos en el sistema.

- Definición de los niveles de idioma. Se especifican los niveles de un idioma y los puntos necesarios para llegar a alcanzarlo.
- Roles de usuario. Eliminar, actualizar y crear nuevos roles de usuario.
- Permisos sobre módulos. Definición de permisos para los roles y usuarios de forma individual sobre los módulos del sistema.
- Variables del sistema. Parámetros de configuración del sistema, es posible eliminar o actualizar los existentes y crear nuevos. Algunos parámetros configurables son: Rol por defecto al crear un usuario, máximo número de contenidos mostrados por página, tiempo máximo de bloqueo

8.1.5.- Administración de contenidos

Inicialmente sólo pueden realizar administración de contenidos los que posean el rol “**Manager**”. Desde la administración de contenidos es posible configurar la máquina de estados y por tanto el comportamiento del sistema en el tratamiento y control sobre la información

Para acceder a la administración de contenidos desplegamos el menú de contenidos y pulsamos “**Administración**”.



Una vez dentro tenemos varias posibilidades.

» Gestión de contenidos.



Figura 8.10 Administración de contenidos

- **Categorías de contenidos:** Creación de diferentes categorías para la clasificación de contenidos. Cada idioma tiene sus propias categorías.
- **Acciones.** Creación, actualización y eliminación de acciones que se pueden realizar sobre un contenido.
- **Estados.** Creación, actualización y eliminación de estados en los que se puede encontrar un contenido.
- **Transiciones de estados.** Creación, actualización y eliminación de transiciones de la máquina de estados.
- **Permisos sobre los contenidos.** Definición de las acciones que puede realizar un determinado rol sobre los contenidos.
- **Ahorcado.** Ampliación del número de palabras para el juego del ahorcado. Se muestra un área de texto donde el administrador podrá especificar cualquier texto. El sistema extraerá automáticamente las palabras.

8.1.6.- Contenidos

Los contenidos son accesibles desde la cuenta de usuario en “Mis contenidos” y “Tareas pendientes” y desde el menú desplegable “Contenidos” situado en la parte superior.



Cuando realicemos clic sobre alguno de los tipos de contenidos, se mostrará el listado de contenidos de ese tipo que se **encuentren** en estado **publicado**, así como las acciones que se pueden realizar asociadas al estado inicial de la máquina de estados: La única acción posible con la máquina de estados que viene por defecto en el sistema es crear un nuevo contenido.

Para entrar en el contexto de un determinado contenido pulsamos sobre su título. Se mostrará la ficha del contenido y cuales son las acciones posibles que se pueden realizar sobre el contenido.

Las acciones posibles definidas inicialmente en el sistema son:

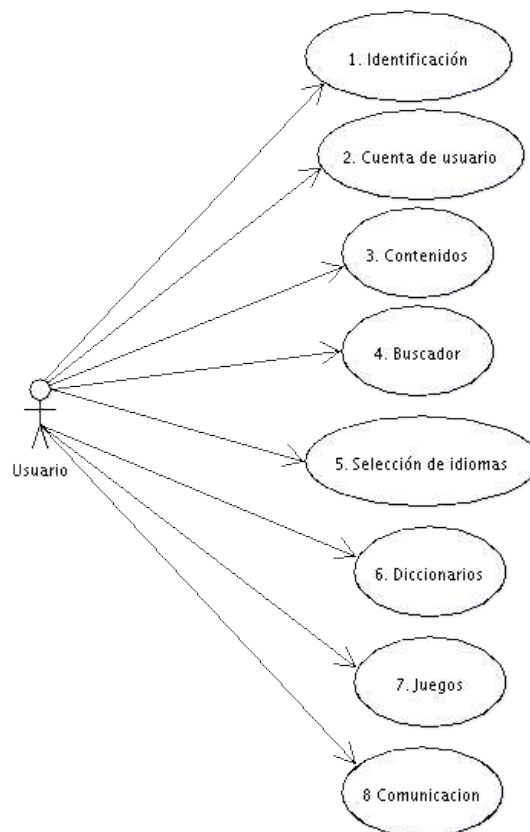
- **Modify.** Nos llevará a una página con el contenido actual y podrá ser modificado.
- **Finish.** El autor da por terminado el documento para que pase por las etapas de revisión.
- **Correct.** Corrección del contenido y elección del nivel de complejidad.
- **View.** Previsualización del contenido antes de que sea publicado.
- **Publish.** Publicación del contenido.
- **Reject.** Rechazo del contenido.
- **Recover.** Recuperación de un contenido rechazado.

- **Read.** Lectura de un contenido publicado.
- **Erase.** Eliminación de un contenido rechazado.

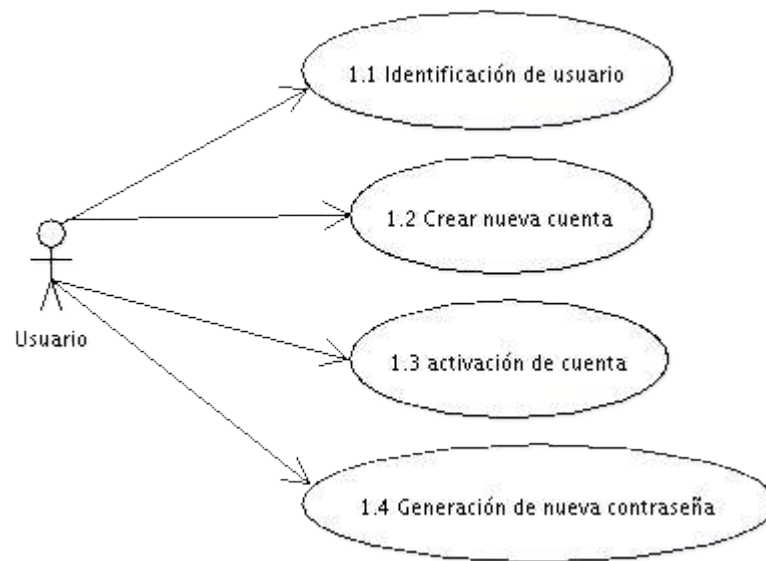
La mayoría de estas acciones o bien realizan un cambio de estado del contenido directamente, o nos lleva a una página con un formulario formado por diferentes campos que el usuario deberá completar o modificar.

8.2.- ANEXO B – DIAGRAMAS DE CASOS DE USO

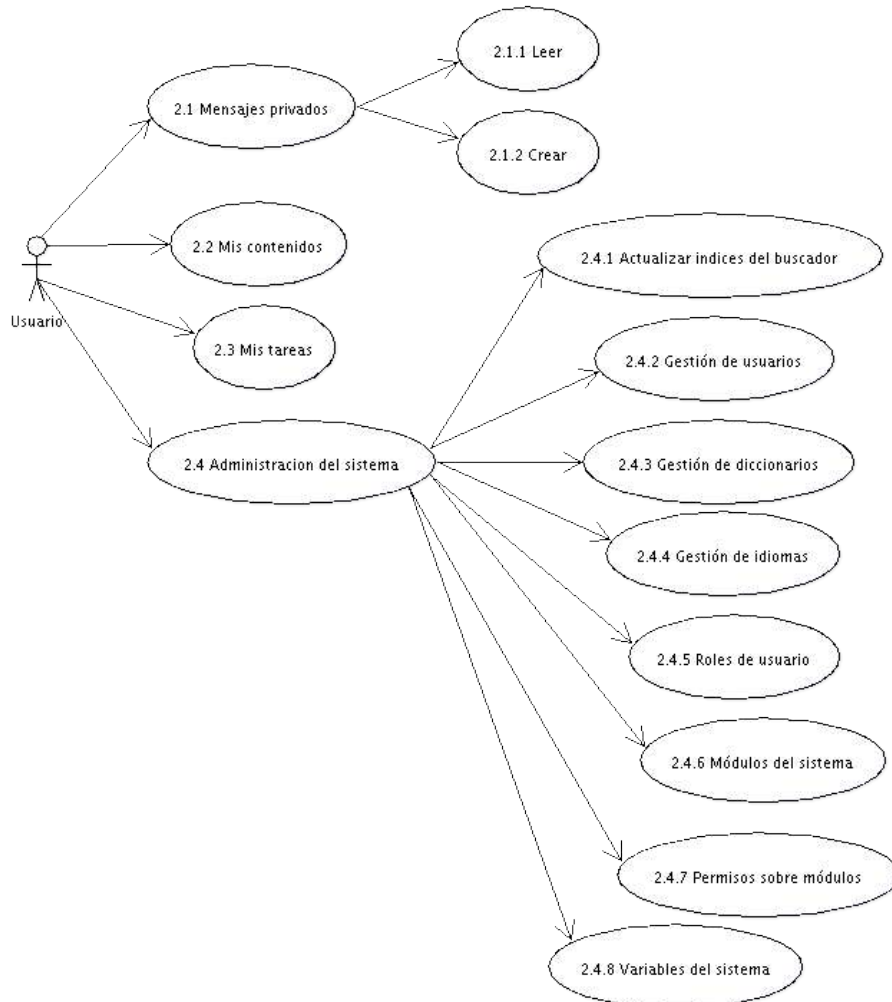
Los diagramas de casos de uso describen [DOM04] la funcionalidad del sistema. Permiten ver de una forma sencilla las posibles interacciones del usuario con el sistema. Para la realización de los siguientes diagramas no se ha tenido en cuenta el rol del usuario, ya que los permisos y posibles acciones que pueda realizar cada tipo de rol pueden ser modificadas desde la administración del sistema.



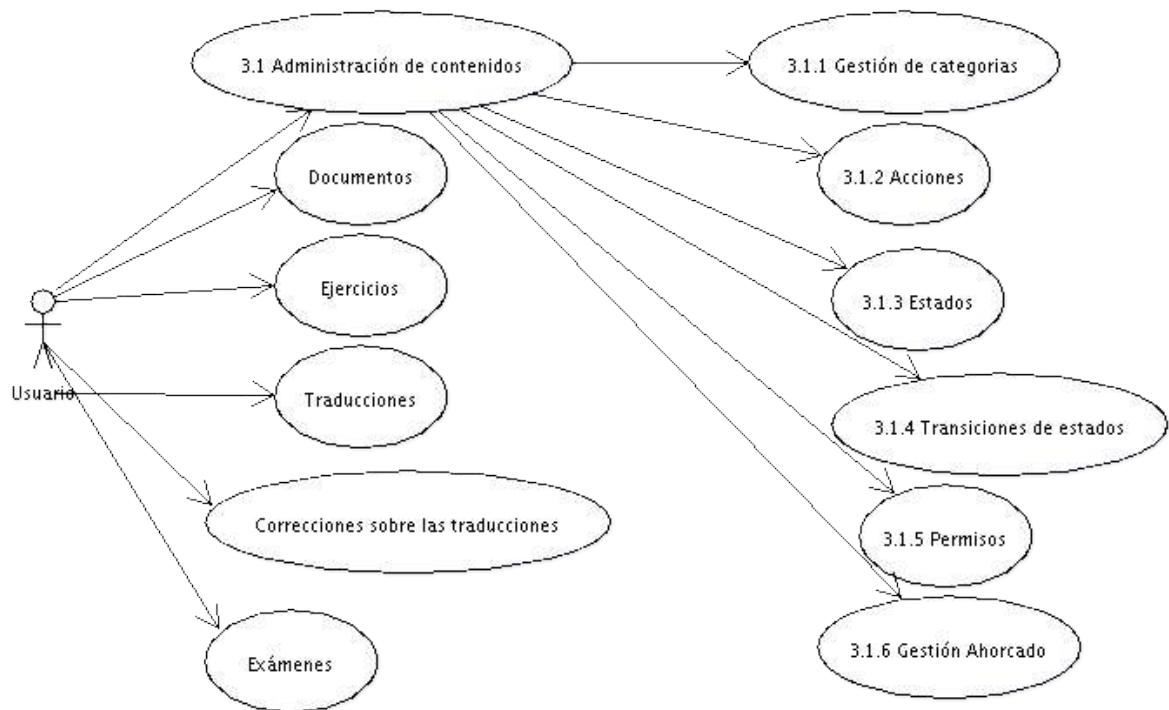
1.- Identificación



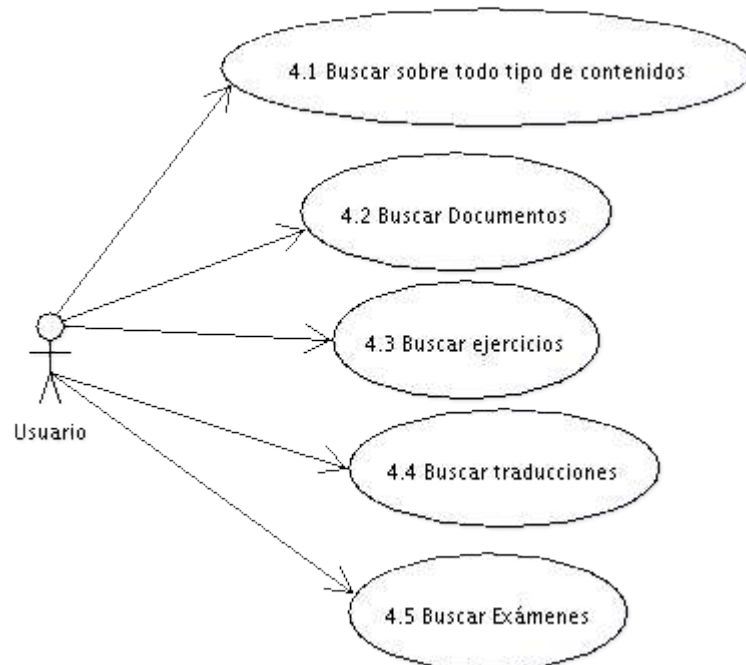
2.- Cuenta de usuario



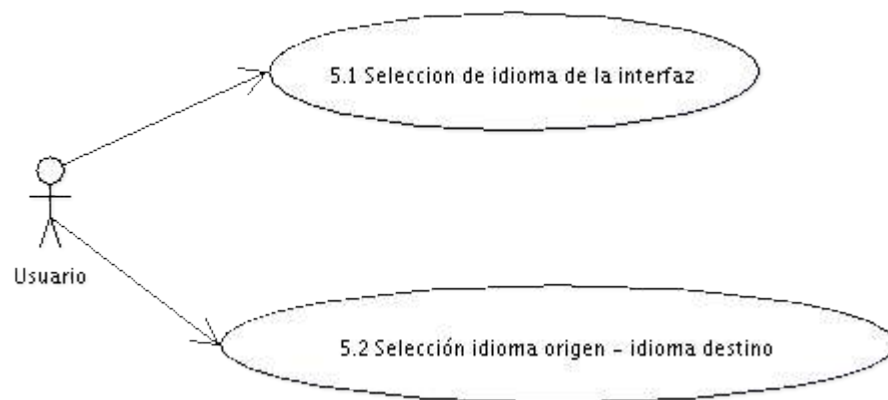
3.- Contenidos



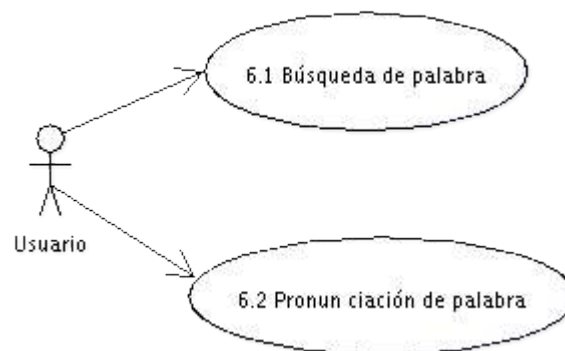
4.- Buscador



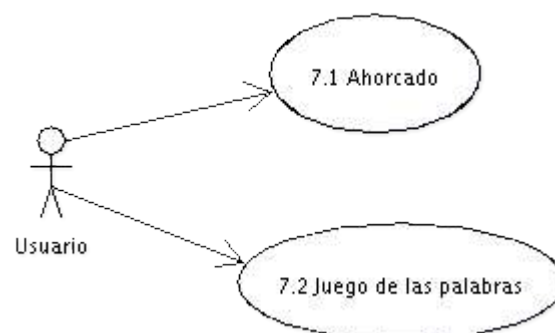
5.- Selección de idioma



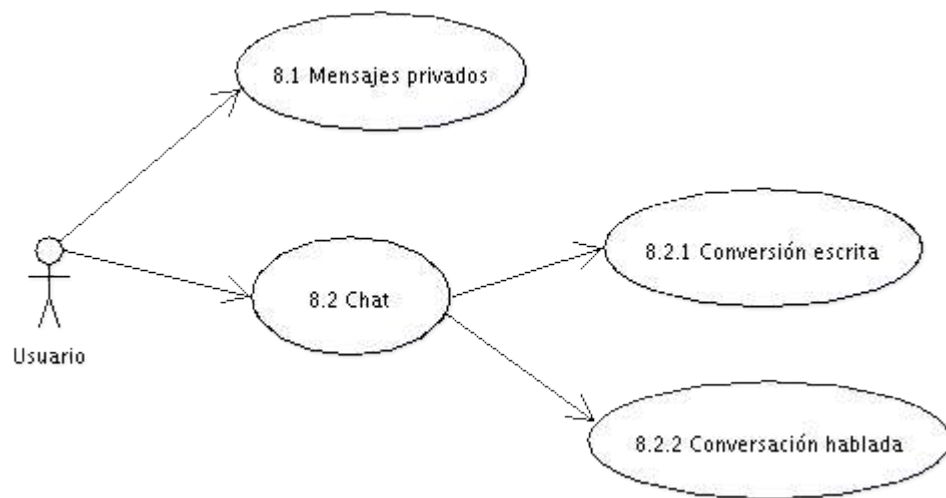
6.- Diccionarios



7.- Juegos



8.- Comunicación

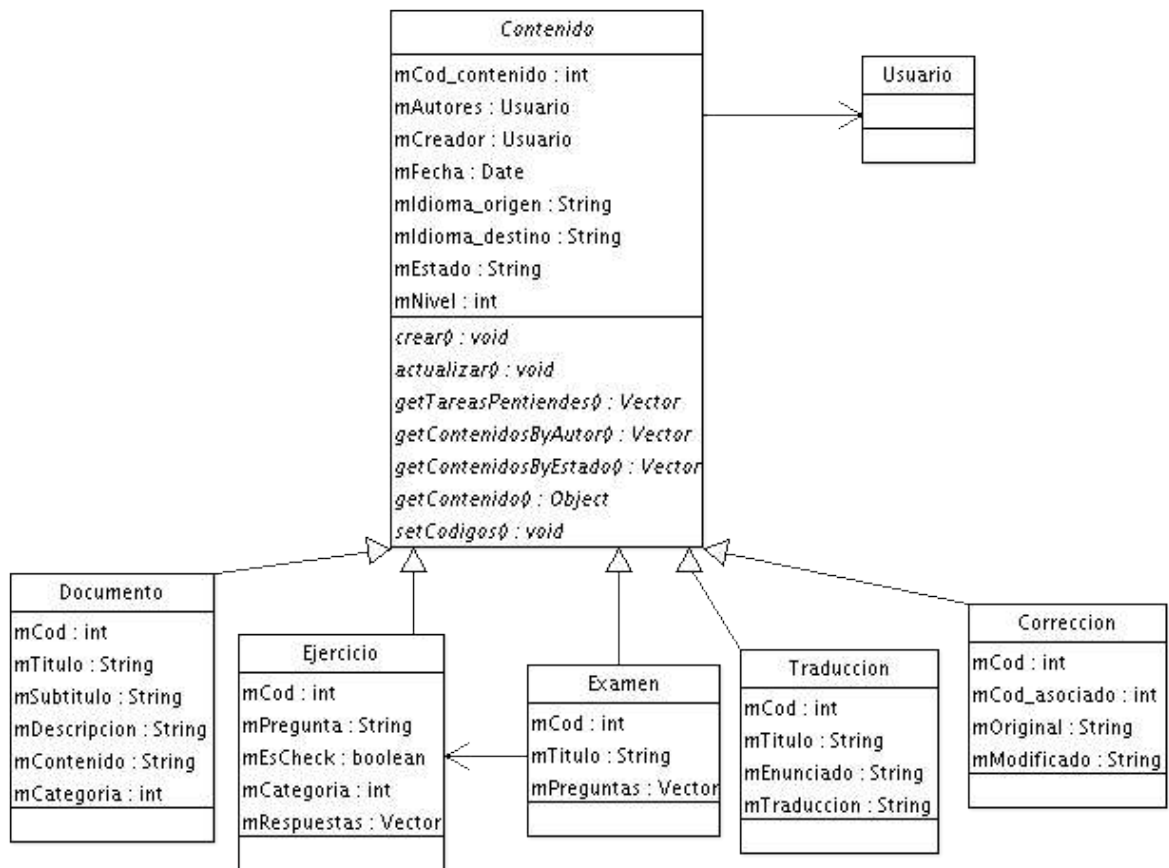


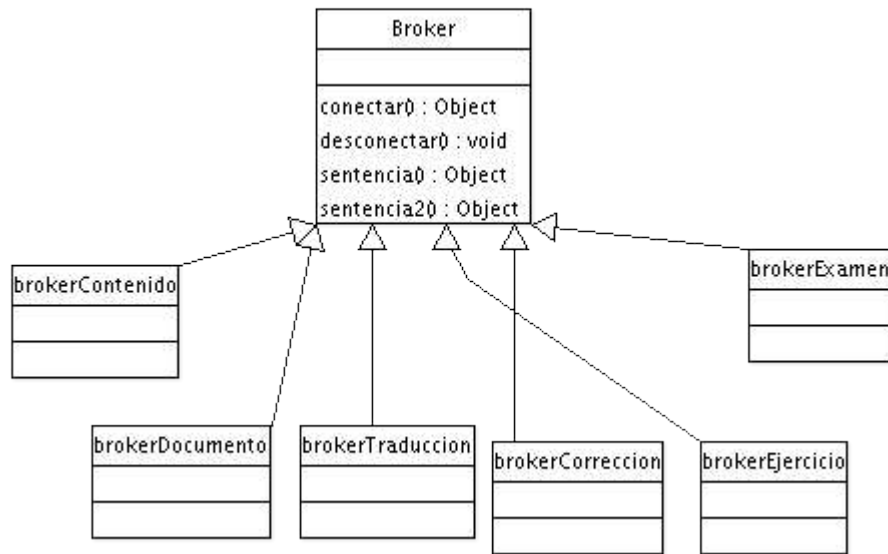
8.3.- ANEXO C - DIAGRAMA DE CLASES

A continuación se mostrarán los diagramas de las clases más importantes del sistema junto con su estructura jerárquica.

Contenidos

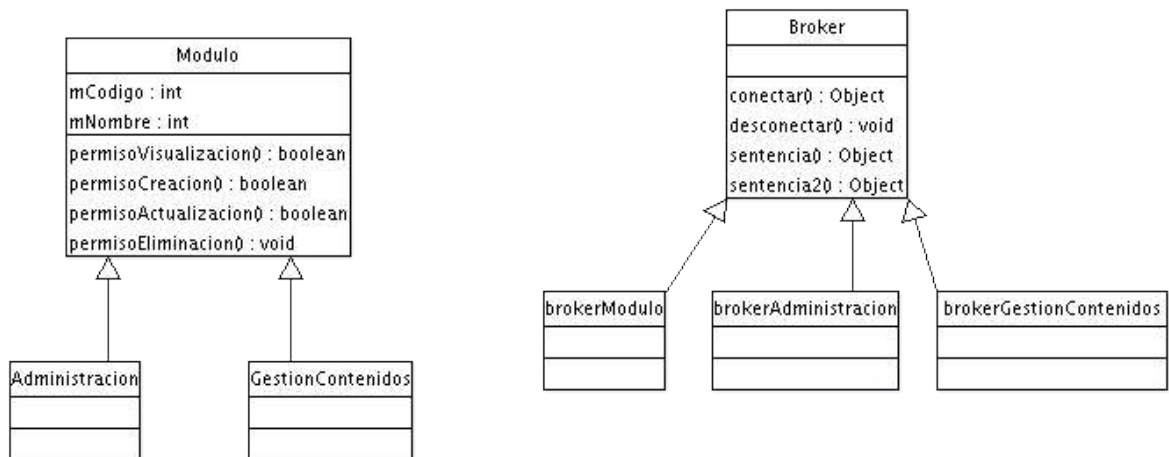
Algunas de las clases referentes a los contenidos poseen más de quince métodos, con el fin de simplificar el diseño y resaltar la estructura jerárquica, únicamente se muestran los métodos abstractos y los atributos de la clase.



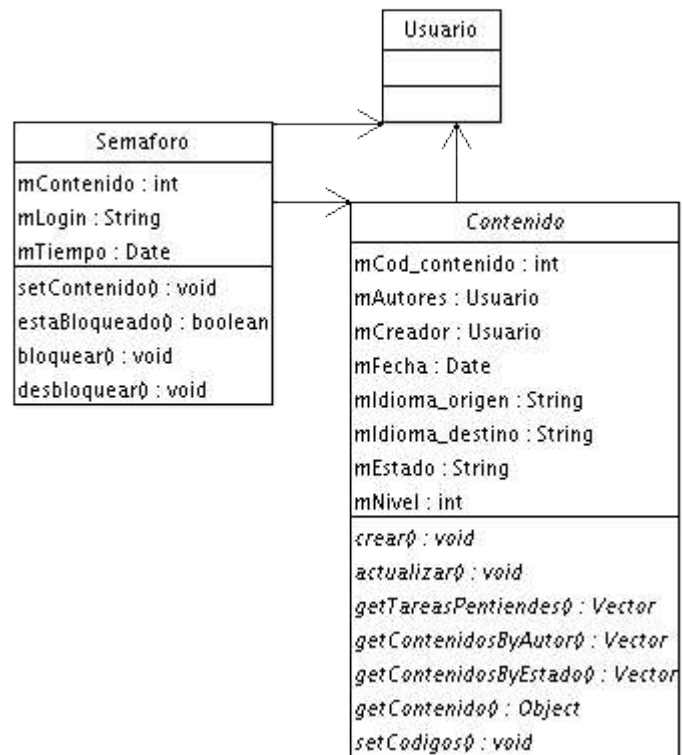


Módulos

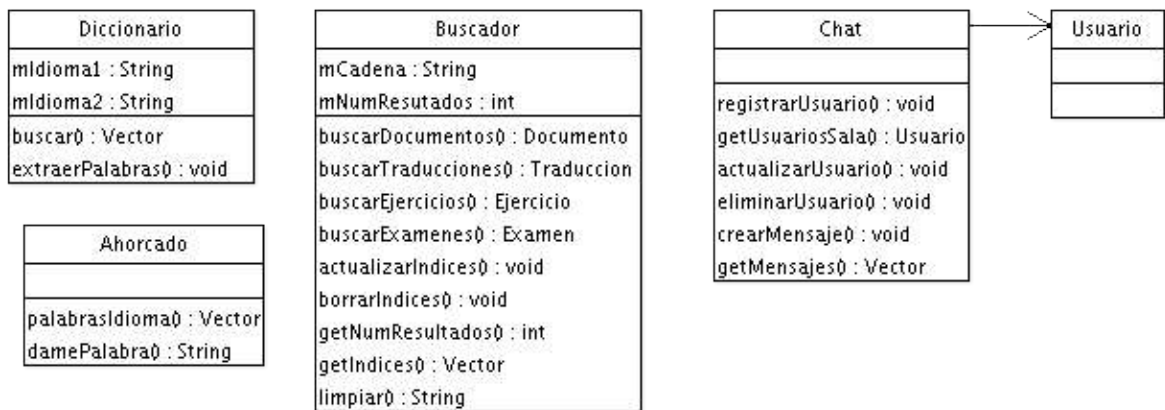
Por el mismo motivo que en los contenidos se ha simplificado el diagrama de clases para los módulos.



Semáforos



Otras clases



8.4.- ANEXO D- ESTRUCTURA DEL CÓDIGO FUENTE

El proyecto está formado por **más de 17.000 líneas de código**. Puede encontrarse en el CD adjunto, en el directorio **pfc**. A continuación se pasará a describir la estructura de directorios y el contenido de cada uno de ellos.

■ **pfc**

- **Dominio:** Capa de dominio donde se encuentra implementada la funcionalidad del sistema.
 - **Clases:** Clases de la capa de dominio.
 - **Scripts:** Scripts utilizados para la generación de índices en el buscador de contenidos.

- **Estilo:** En ese directorio se almacenan las diferentes hojas de estilo utilizadas.
 - **pc.css:** Hojas de estilo para ordenadores de sobremesa.
 - **pda.css:** Hojas de estilo utilizado para dispositivos PDA.

- **Idiomas:** Directorio donde se encuentran los archivos de idiomas para la interfaz de usuario. Uno por cada idioma activo en el sistema.

- **Persistencia:** Capa de persistencia del sistema. Se encuentra la clase broker con las funciones básicas para la manipulación de la base de datos, y una clase broker para cada una de las clases de la capa de dominio.

- **Presentación:** Elementos de la capa de presentación del sistema.
 - **archivos:** Directorio donde se almacenan los ficheros asociados a los documentos.
 - **avatars:** Cada uno de los usuarios posee una imagen asociada a su cuenta (avatar), es en este directorio donde se almacenan las imágenes.
 - **diccionarios:** Diccionarios de texto plano subidos al servidor y utilizados para la extracción de términos y definiciones.
 - **html_to_pdf:** Librerías utilizadas para la generación de documentos en formato PDF a partir de contenido HTML.

- **imagenes:** En este directorio se almacenan las imágenes utilizadas en la capa de presentación.
- **js:** Script de JavaScript para realizar efectos sobre las capas de la interfaz.
- **sonidos:** Archivos con formato WAV generados para la pronunciación de las palabras por parte de los usuarios.
- **spaw:** Librerías del editor WYSIWYG Spaw.

8.5.- ANEXO E – INSTALACIÓN Y CD ADJUNTO.

Es posible la utilización del sistema desde <http://www.freebabel.com> si no se desea instalarlo de forma local.

A continuación se describirán los pasos necesarios para la instalación del sistema tanto en plataforma Windows, como GNU/Linux en distribuciones basadas en Debian.

Plataforma Windows:

En el CD adjunto en el directorio **software**, ejecutar el archivo **wamp5_1.4.6a**. Instalará en el sistema el servidor Apache con PHP5 y MySQL.

Dentro del directorio que hayamos elegido para la instalación de wamp5 se habrá creado un directorio llamado **www**. Copiar en este directorio la carpeta **pfc** del CD adjunto. Para acceder al sistema escribir en el navegador la dirección:
<http://localhost/pfc/presentacion>.

Por último queda la creación de la base de datos. En el directorio elegido para la instalación de wamp5, debe existir un directorio **mysql** destinado a albergar las bases de datos, copiar el directorio **babel** que se encuentra dentro de **bd** en el CD adjunto. Esto creará la base de datos de Babel con contenidos iniciales. Si se desea crear la base de datos en blanco, se deben ejecutar las sentencias SQL que se encuentran en /sql/bd.sql dentro del CD adjunto.

La configuración por defecto del sistema para el acceso a la base de datos MySQL, es login: **root** y sin contraseña, si se han elegido otros valores diferentes debe editarse el archivo /pfc/persistencia/config.php especificando los nuevos valores.

Plataforma Linux:

Añadir en el archivo `/etc/apt/sources.list` el repositorio:

deb <http://people.debian.org/~dexter> php5 hoary

Una vez añadido la nueva fuente actualizamos con `apt-get update`. Cuando finalice la actualización ejecutamos en orden:

```
$ apt-get install apache2
$ apt-get install mysql-server
$ apt-get install php5
$ apt-get install libapache2-mod-php5
$ apt-get install php5-gd
$ apt-get install php5-dom
$ apt-get install php5-pgsql
$ apt-get install php5-mysql
```

PHP5 por defecto tiene soporte para MySQL pero puede que no este activado inicialmente. El problema viene por conflictos entre licencias, la nueva versión de PHP está desarrollada bajo BSD mientras que MySQL continúa con GPL .

Para activarlo existen dos opciones:

- 1) Si se han bajado los fuentes de PHP se debe compilar con la opción `mysql` (`--with-mysql`).
- 2) Si se ha instalado con la herramienta `apt`, editar el archivo `/etc/php5/apache2/php.ini` y quitar el comentario de la línea `;extension=mysql.so`

Tablas de tipo InnoDB.

Si deseamos crear tablas de tipo InnoDB en MySQL, debemos editar el archivo `/etc/mysql/my.cnf` y comentar la línea `: skip-innodb`.

Las tablas de tipo InnoDB en MySQL, nos permiten crear claves foráneas con las que podemos relacionar tablas y actualizar y eliminar registros en cascada.

Register Globals

Para poder crear sesiones en PHP, o bien enviar mediante el método POST valores con formularios es necesario dar el valor "on" a la variable register_globals.

editar el archivo /etc/php5/apache2/php.ini ---> register_globals = on

Copiar el directorio **pfc** del CD adjunto, en el directorio /var/www. Para acceder al sistema escribe en el navegador la dirección <http://localhost/pfc/presentacion>.

Para la creación de la base de datos del sistema, existen dos opciones: copiar el directorio **babel** que se encuentra dentro de **bd** en el directorio adjunto (/bd/babel) en nuestro sistema dentro de /var/lib/mysql/. Si deseamos crear la base de datos únicamente con la estructura de tablas podemos ejecutar las sentencias SQL que se encuentran en /sql/bd.sql del CD adjunto.

Por defecto, el nombre de usuario para acceder a la base de datos es **root** y sin password, si se especifica un nombre de usuario o contraseña diferente es necesario modificarlos en el archivo config.php ubicado en /var/www/pfc/persistencia/config.php

CD Adjunto