Escuela Superior de Informática
Universidad de Castilla-La Mancha
Examen final, 2º parcial, de Ingeniería del Software II
11 de junio de 2001

Problema 1. Una empresa de venta por correspondencia desea disponer de un programa en Java para realizar envíos masivos de correos electrónicos que simulen envíos personales. El programa utilizará una base de datos en la que se almacena información de clientes, productos, etc. El programa debe poder realizar dos tipos de envíos:

a) Envíos de ofertas de productos nuevos

b) Envíos con felicitaciones de cumpleaños

Vd. debe encargarse de desarrollar dicho programa, que debe reunir las siguientes **características**:

El usuario del programa manejará la aplicación a través de una pantalla como la mostrada en la siguiente figura.

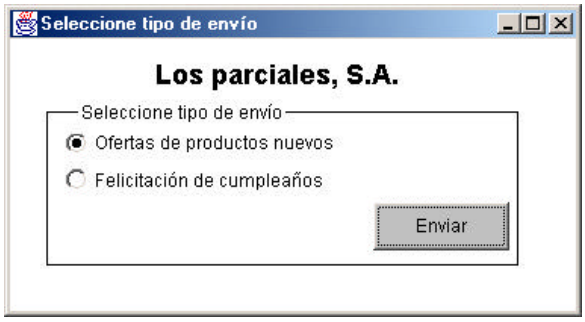

Figura 1

Los mensajes de ofertas se envían por un servidor SMTP, mientras que los de felicitaciones se envían a través de un servidor IMAP. Podemos enviar mensajes usando las clases que se muestran en la Figura 2, que tienen las siguientes características:

- No podemos manipular ninguna de las tres clases de la Figura 2, pero sí podemos crear especializaciones (clases que heredan de ellas).
- Los únicos miembros públicos en las clases de la Figura 2 son los constructores (que lo único que hacen es asignar valor a las propiedades de la clase) y el método enviar. Los demás miembros (atributos y métodos) son protegidos.
- El método *enviar* está declarado como de tipo *void* y no toma parámetros.

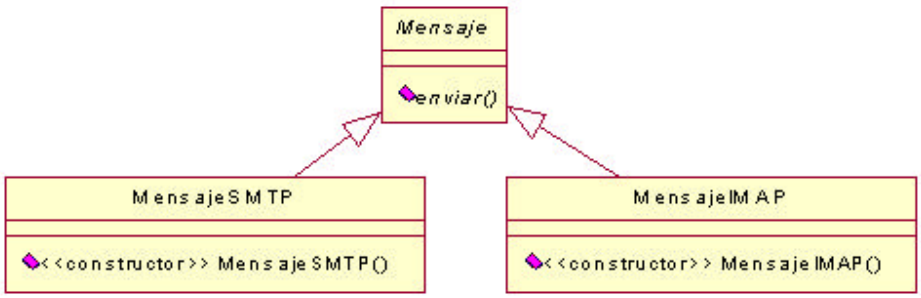

Figura 2

Una vez que en la pantalla de la figura 1 se ha elegido una opción y se ha pulsado el botón de enviar, el programa hará siempre estas operaciones:

- Carga automáticamente la lista de destinatarios en un objeto de clase ListaDestinatarios de la capa de dominio.
- Si se eligió "mensajes de oferta", el objeto de clase ListaDestinatarios carga la lista en un JFrame; si es una felicitación, la lista de destinatarios no se muestra.
- Envia el mensaje i-ésimo (SMTP o IMAP)
- Muestra el resultado del envío en la pantalla:
  - Si es de oferta y el envío es correcto, en el JFrame se muestra el destinatario en color verde, y en rojo si es incorrecto.
  - Si es de cumpleaños y el envío es correcto, el nombre del destinatario se escribe por la salida estándar (la "pantalla negra"); en caso de que el envío sea incorrecto, se saca también por la salida estándar pero entre admiraciones (p.ej.: ¡¡¡Paco Pil falló: pil@dj.net!!!).
- Guarda en un mecanismo persistente el resultado del envío:
  - Si es de oferta y se envió correctamente, se guarda en una base de datos; si no se envió correctamente, se guarda en un fichero secuencial
  - Si es de cumpleaños y se envió correctamente, se guarda en un fichero de acceso aleatorio; si se envió incorrectamente, se imprime su nombre por la impresora.

Se pide:

1) Represente mediante un diagrama de clases un diseño de alta calidad para este sistema, indicando y explicando qué patrones de diseño ha utilizado. **2 puntos.**

2) Amplíe el sistema diseñado en el apartado 1 para que envíe mensajes de cumpleaños a través de teléfono móvil. El funcionamiento será el siguiente: si el cliente tiene teléfono móvil, la felicitación de cumpleaños se le mandará como un mensaje de texto al móvil mediante una clase especializada de Mensaje y con las mismas características que MensajeSMP y MensajeIMAP. Si no tiene móvil, se le manda por correo electrónico. El sistema supondrá que los mensajes enviados al móvil llegan siempre, por lo que con estos mensajes procederemos como al enviar una felicitación de cumpleaños por correo electrónico que llega correctamente. **1,5 puntos.**

3) Basándose en el apartado anterior, represente mediante un diagrama de secuencia el funcionamiento del sistema cuando se envían mensajes de cumpleaños. **1,5 puntos.**

4) Explique (en castellano o usando diagramas UML) cómo utilizaría el patrón State (Estado) para representar el hecho de que a los clientes que tienen móvil se les felicita por móvil, y por correo electrónico a los que no lo tienen. **1 punto.**

Pregunta 2. Explique el funcionamiento de la "Old Dirty Caché". **1 punto.**