



UNIVERSIDAD DE CÓRDOBA
ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA

INGENIERÍA INFORMÁTICA
TERCER CURSO. PRIMER CUATRIMESTRE

PROGRAMACIÓN WEB.

Práctica 1: Iniciación a Java.

Juan Pedro Muñoz Jimenez
p52mujij@uco.es

Ángel Sevilla Molina
i42semoa@uco.es

Curso académico 2020-2021
Córdoba, 16 de octubre de 2020

Índice

1. Fichero de propiedades y de datos	1
2. Agenda de contactos	1
2.1. Clase Contacto	1
2.2. Clase GestorContactos	2
2.3. Clase GestorMainProgram	2
3. Tablón de anuncios	3
3.1. Clases de anuncios	3
4. Clase GestorAnuncios	4
5. Clase AnunciosMainProgram	4

1. Fichero de propiedades y de datos

El proyecto contiene un fichero de propiedades con las rutas de los ficheros de datos, localizados en el directorio *data*, así como la información de los temas de interés que sigue un diseño de etiquetas.

Para los datos de los usuarios solo se ha utilizado un fichero. En la primera línea se guarda el primer identificador disponible, el cual nos sirve para identificar a partir de que id podemos registrar nuevos usuarios sin crear conflictos. Al guardar el este identificador en el fichero de datos, siempre lo tendremos a salvo al no depender de los usuarios registrados. Por ejemplo, si borramos los últimos usuarios, el identificador disponible no cambiaría en consecuencia y no se crearían conflictos con los anuncios ya creados.

Para los ficheros de datos se ha optado por tener un fichero de datos único para cada tipo de anuncio. De esta manera solo se recorre una vez la lista de anuncios a la hora de cargar/guardar. Además, cada tipo de anuncio mantiene información diferente, y por lo tanto requiere un formato distinto para poder ser almacenado. Al tener distintos ficheros se logra evitar problemas de comprobaciones innecesarias o de cast.

2. Agenda de contactos

El primer ejercicio a realizar es una agenda de contactos. Las clases que implementan la agenda se encuentran en el paquete *es.uco.pw.contactos*, y son tres: *Contacto*, *GestorContactos* y *GestorMainProgram*.

2.1. Clase Contacto

La clase *Contacto* representa la información de un contacto de la agenda. Solo dispone de métodos *get* y *set*, así como un constructor. Cada contacto se identifica o por un código numérico, o por su correo electrónico, ambos únicos para dicho usuario. Pero esa naturaleza de unicidad no se comprueba en la propia clase, sino en el gestor.

Lo más destacable de esta clase es el manejo de los temas de interés del contacto, a través de la definición de un vector de booleanos, donde el valor *i*-ésimo determina si el contacto tiene interés por el tema *i*-ésimo.

2.2. Clase GestorContactos

La clase *GestorContactos* sigue el patrón de diseño *singleton*, y su objetivo es permitir el manejo de la agenda. Mantiene una lista con todos los contactos existentes, facilitando operaciones de memoria dinámica a costa de perder eficiencia en consulta. Dada la naturaleza del identificador numérico de cada contacto, y el que la lista se mantiene siempre ordenada de forma ascendente, se pueden utilizar algoritmos de búsqueda más eficientes con otras estructuras de datos. En esta implementación se hace uso de búsquedas lineales.

2.3. Clase GestorMainProgram

La clase *GestorMainProgram* contiene la aplicación principal que permite la gestión de los contactos gracias a la clase *GestorContactos*. El programa funciona mediante un menú con opciones accesible por línea de comandos.

Para mantener actualizada la información el programa carga los datos de los contactos al inicio de su ejecución. Al mismo tiempo, se guardan todos los cambios realizados al salir del programa. Sin embargo estos cambios solo se registran si se sale de manera exitosa, haciendo uso de la opción del menú, lo que significa que un fallo durante la ejecución de la aplicación provocaría la pérdida de las modificaciones realizadas en ella.

Como aspecto negativo, algunas funcionalidades como dar de baja a un usuario suponen un problema en término de rendimiento debido a operaciones redundantes. En este ejemplo, se le solicita al usuario que introduzca el correo electrónico del contacto a borrar de la agenda. Una vez introducido se busca al usuario para verificar la operación, búsqueda de orden de complejidad $O(n)$. Si el usuario confirma la baja, entonces se invoca al método *eliminarContacto*, que internamente realiza otra búsqueda y lo elimina definitivamente de la lista.

Este mismo problema puede encontrarse en la modificación de la información de un contacto. Lo que realmente realiza el programa solicitar toda la información de nuevo, y modificar de golpe los campos del objeto seleccionado por medio del correo electrónico. Se ha tomado esta decisión debido a las limitaciones de la línea de comandos, así como para evitar sucesivas búsquedas para actualizar cada uno de los campos. El método seleccionado solo implica dos búsquedas: una de comprobación del usuario y otra para modificar sus datos.

3. Tablón de anuncios

El segundo ejercicio es un tablón de anuncios que utiliza como base la agenda de contactos. Las clases relativas a los anuncios se encuentran en el paquete *es.uco.pw.anuncios*, y son *IAnuncio*, *AnuncioGeneral*, *AnuncioTematico*, *AnuncioIndividualizado*, *AnuncioFlash*, *GestorAnuncio* y *AnunciosMainProgram*.

3.1. Clases de anuncios

Para el diseño de anuncios se ha optado por la definición de una clase interfaz, *IAnuncio*, que defina los métodos comunes y necesarios para el resto de anuncios. Estos son básicamente los get y set de atributos como el identificador, título, cuerpo, fecha de publicación, usuario propietario o la fase en la que se encuentra. También operaciones para publicar el anuncio o archivarlo.

- La clase *AnuncioGeneral* representa aquel anuncio que puede ser visto por todos los usuarios, y por tanto no requiere de métodos específicos aparte de su constructor.
- La clase *AnuncioTematico* representa aquel anuncio asociado a un tema de interés. Para la gestión de los temas de interés se ha seguido el mismo diseño que el de la agenda de contactos, utilizando un vector de booleanos.
- La clase *AnuncioIndividualizado* representa aquel anuncio que se ha enviado a un conjunto de usuarios específicos, controlado por medio de una lista con identificadores de dichos usuarios.
- La clase *AnuncioFlash* representa aquel anuncio que solo está activo por una franja temporal especificada. Como todos los anuncios ya tienen su fecha de publicación que indica el comienzo del anuncio, por lo que lo único específico es la fecha de fin. La comprobación de si dicha fecha de fin se ha alcanzado, y por tanto que el anuncio deba ser archivado, se realiza en la carga de los datos de los anuncios, es decir, al iniciar la aplicación.

Cada anuncio pasa por una serie de fases, que son las siguientes:

- La fase 0 o *editado* representa a aquel anuncio que se encuentra en proceso de edición. Solo los anuncios que se encuentran en esta fase pueden ser modificados. Un anuncio se encuentra en esta fase al crearse, y no está visible en el tablón. Cuando el usuario indique una fecha de publicación, entonces pasará a la fase 1.
- La fase 1 o *en-espera* representa a aquel anuncio cuya fecha de publicación se ha indicado pero todavía no está visible en el tablón porque no se ha alcanzado. Una vez esta fecha se alcance, pasa a la fase 2.
- La fase 2 o *publicado* representa a aquel anuncio cuya fecha de publicación se ha alcanzado y está visible en el tablón de anuncios. Si su fecha de fin se alcanza (si es un anuncio flash) o el usuario decide archivarla, entonces pasará a la fase 3.
- La fase 3 o *archivado* representa a aquellos anuncios cuya información se mantienen pero ya no están disponibles.

4. Clase GestorAnuncios

La clase *GestorAnuncios* sigue el mismo estilo que *GestorContactos*, con un patrón de diseño *singleton* así como las ventajas y desventajas del uso de la lista. Las mayores dificultades se encuentran en las operaciones de búsqueda y ordenación, que trabajan con listas ajenas a las de la propia agenda de anuncios para poder aplicar distintos filtros al mismo tiempo.

5. Clase AnunciosMainProgram

La clase *AnunciosMainProgram* contiene la aplicación principal que permite la gestión de los anuncios gracias a la clase *GestorAnuncio*. Una de las mayores dificultades se encuentra en la función de editar anuncios, que solo permite modificar la información común de los anuncios, como el título o el cuerpo, pero no los campos específicos, como los temas de interés en el caso de los anuncios temáticos.

Se ha incorporado una opción que permite al usuario ver, además del tablón de anuncios, todos aquellos anuncios de los que él sea usuario propietario, incluidas aquellas que se encuentren en otra fase que no sea *publicado*.