



PRÁCTICA 1: INICIACIÓN A JAVA

En esta práctica nos iniciaremos en el desarrollo con el lenguaje Java, incorporando además patrones de diseño para acostumbrarnos como programadores a seguir las mejores prácticas (*best practices*) en el desarrollo de software, distintivas de los buenos profesionales.

En un equipo de 2 personas, que se mantendrá invariable durante el presente curso académico, haremos uso de los elementos habituales del lenguaje, además de E/S de ficheros y de consola, desarrollo de ficheros de propiedades y patrones de diseño.

Tenga en cuenta los siguientes **aspectos normativos** referentes a esta práctica:

- Junto con el código fuente, se entregará un informe en formato PDF (máximo 4 páginas) que deberá explicar y fundamentar las decisiones de diseño e implementación que ha adoptado el equipo, así como la relación de fuentes consultadas y un análisis de las dificultades encontradas.
- El código debe estar completamente documentado haciendo uso de Javadoc, tanto a nivel de clase como de métodos.
<https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>
- La propuesta de ejercicios tiene margen de decisión por parte del equipo. Dado que se espera que estas decisiones sean distintas por cada equipo de prácticas, deberán estar justificadas pertinentemente en el informe.
- Sólo hará entrega de la práctica un miembro del equipo a través de Moodle. La persona que entregue esta práctica lo hará igualmente de las demás.
- Se hará entrega a través de la tarea Moodle dispuesta para ello de un archivo ZIP que encapsule el proyecto Eclipse (preferiblemente) o fuentes Java de la práctica.
 - Se debe tener especial cuidado en el nombrado de paquetes y clases.
 - El archivo PDF correspondiente al informe debe estar incluido en el raíz del directorio del proyecto.
 - Para cada ejercicio, se deberá entregar el ejecutable pertinente (JAR, WAR, etc.), incluido en el raíz del ZIP de proyecto, así como un README.txt con las instrucciones pertinentes para su ejecución.
 - El archivo ZIP que se entrega debe ser nombrado de la siguiente forma:
GM<gm>_<login>.zip donde <gm> es el número de grupo mediano y <login> es el *nick* de la UCO de la persona del equipo que hace la entrega de esta y el resto de prácticas.

El incumplimiento de las normas de entrega (contenido, nombrado, etc.) implicará la consideración de práctica no entregada. La fecha de entrega se dispondrá en la tarea Moodle correspondiente. Igualmente, el retraso en la entrega supondrá la no calificación en la práctica.

IMPORTANTE: No se admitirá entrega alguna fuera de plazo o a través de un medio distinto a la tarea de Moodle (p.ej. envío por email).

EJERCICIO 1

Se quiere implementar un **gestor de contactos**, en el que cada Contacto mantiene información acerca de su nombre, apellidos, fecha de nacimiento, email (único) y un conjunto de intereses que identifican al usuario. Estos intereses son asignados mediante sistema de *tagging*.

El gestor de usuarios será único, independientemente del objeto que lo invoque, implementándolo para ello mediante el patrón de diseño *Singleton*. Las operaciones que ofrece el gestor de contactos serán las siguientes: dar de alta contacto, dar de baja contacto, consultar datos de contacto, actualizar datos de contacto, buscar contacto por email, buscar por {nombre, apellidos}, buscar por interés, buscar por edad.

Para probar la funcionalidad del gestor, se desarrollará un programa Java con interfaz texto (E/S por consola) que ofrezca las siguientes opciones:

- Leer/escribir datos del gestor de contactos en fichero de datos. Será el gestor de contactos el encargado de cargar el listado desde el fichero cuando se inicialice.
- Inicializar y acceder a las funcionalidades CRUE y de búsqueda del gestor de contactos.

La localización del fichero de datos se configura en un fichero de propiedades texto, a las que el programa Java accederá, haciendo uso de la clase *java.util.Properties*.

<https://docs.oracle.com/javase/8/docs/api/java/util/Properties.html>

Tutorial: <https://www.baeldung.com/java-properties>

Para la implementación de la entrada de datos por consola se puede utilizar la clase *java.util.Scanner*.

<https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html>

EJERCICIO 2

Se desea implementar un tablón de anuncios. Para ello, un Anuncio estará compuesto de un identificador, título, usuario propietario, (0+) usuarios destinatarios y el cuerpo del anuncio. El Tablón se encargará de mostrar todos los anuncios según el criterio de ordenación indicado por el usuario: fecha de publicación o usuario propietario.

En el tablón se podrán generar 4 tipos de anuncios:

- Anuncio general, que puede ser visto por todos los usuarios.
- Anuncio temático, que deberá tener asociado uno o más temas de interés (se utilizarán los temas propuestos previamente en el gestor de contactos, y que deberán ser configurables a partir de un fichero de propiedades).
- Anuncio individualizado, que se remite específicamente a uno o varios usuarios destinatarios.
- Anuncio *flash*, que se muestra en el tablón restringido entre una fecha/hora de inicio y una fecha/hora final.

Se debe programar una factoría para la generación de los tipos de anuncio. Igualmente, hágase uso de interfaces cuando sea necesario. Para la gestión de usuarios, reutilice el gestor de contactos del Ejercicio 1.

En general, un gestor de anuncios, que guarda todos los anuncios publicados en el sistema. Los anuncios van pasando por distintas fases: editado → en-espera → publicado → archivado. El estado "editado" se corresponde los con anuncios que están creados pero siguen en edición; "en-espera" con aquellos que han sido publicados pero que la fecha de inicio no se ha alcanzado aún; "publicados" si aparecen en el tablón; "archivados" si han sido eliminados por el usuario propietario o han caducado.

Algunas de las funcionalidades básicas del tablón son:

- Editar anuncio / guardar anuncio.
- Publicar anuncio.
- Archivar anuncio.
- Buscar por fecha.
- Buscar por tema de interés (alguno de los temas de interés del usuario coincide con los temas de interés asociados al anuncio).
- Buscar por usuario propietario.
- Buscar por usuario destinatario.

Se pide implementar un tablón de anuncios con, al menos, las siguientes funcionalidades:

- Identificarse como un usuario.
 - El usuario podrá gestionarse con el programa del Ejercicio 1.
 - Se debe comprobar que el usuario existe en el gestor.

- Se podrá registrar con distintos usuarios en momentos diferentes.
- Inscribirse como usuario a temas de interés.
- Crear un nuevo anuncio de cualquiera de sus correspondientes tipos.
- Mostrar el tablón de anuncios del usuario.

Recuerde que ambos programas deben entregarse con su respectivo fichero JAR, incluyendo además un fichero texto README.txt en el que se explique los parámetros requeridos para la ejecución del programa.

Tutorial: <https://www.baeldung.com/java-create-jar>