



## 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 tres 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 seguir las buenas prácticas de nombrado (variables, paquetes, etc.) y 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.
- Solo hará entrega de la práctica un miembro del equipo a través de Moodle. La persona que entregue esta práctica lo hará igualmente con 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, siguiendo las reglas de convención del lenguaje Java.
  - 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 las 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 críticas de espectáculos, en el que los espectadores puedan darse de alta para escribir reseñas sobre espectáculos, así como interactuar con las críticas de otros espectadores indicando su utilidad mediante un sistema de *rating*. De cada espectador se debe mantener información acerca de su nombre y apellidos, nombre de usuario (*nick*) y correo electrónico (único). Por su parte, una crítica constará de un título, una puntuación otorgada al espectáculo, un texto con la reseña y las valoraciones de utilidad aportadas por otros usuarios. Un usuario no podrá valorar la utilidad de una crítica escrita por él mismo, mientras que cada usuario solo puede valorar cada reseña una única vez.

El gestor de críticas será único, independientemente del objeto que lo invoque, implementándolo para ello mediante el patrón de diseño *Singleton*. Las operaciones que ofrece el gestor de críticas en relación con los usuarios serán las siguientes: dar de alta a un usuario, dar de baja a un usuario, consultar los datos del usuario, actualizar los datos del usuario. Por otro lado, las operaciones relacionadas con las críticas serán las siguientes:

1. Crear una crítica por parte de un usuario registrado.
2. Consultar todas las críticas disponibles.
3. Borrar una crítica, si el usuario es quien la escribió.
4. Votar la utilidad de una crítica de otro usuario registrado.
5. Buscar las críticas de un usuario registrado.

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 críticas en un fichero de texto plano. Será el gestor de críticas el encargado de cargar el listado desde el fichero cuando se inicialice.
- Inicializar y acceder a las funcionalidades CRUD y de búsqueda del gestor de críticas.

Estructure adecuadamente el código para separar las clases relacionadas con los datos, las operaciones de gestión de esos datos, y la interfaz de usuario por consola.

La localización del fichero de datos se configura en un fichero de propiedades, 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 quiere implementar un gestor de espectáculos teatrales, donde cada espectáculo incluye información acerca de su título, descripción y categoría de evento (concierto, monólogo, obra de teatro, etc.). Según su modo de programación, pueden existir los siguientes tipos de espectáculos:

1. Espectáculo puntual, es aquel que solo tiene una fecha y hora de representación.
2. Espectáculo de pase múltiple, que consta de varias funciones de representación.
3. Espectáculo de temporada, que se celebra periódicamente un día de la semana entre unas fechas de inicio y fin.

Se debe programar una factoría para la creación de espectáculos según su modo de programación. Igualmente, hágase uso de interfaces cuando sea necesario.

Todos los espectáculos cuentan con un aforo de localidades, que es igual para todas las sesiones en las que se programe dicho espectáculo. La venta de localidades se realiza para una sesión (fecha y hora) concreta. No es necesario gestionar la venta de localidades más allá de mantener el número de localidades aún disponibles.

Cada espectáculo puede tener asociado una o más críticas de los espectadores que acuden a cualquiera de las sesiones de ese espectáculo. Para gestionar las críticas puede reutilizarse el gestor de críticas implementado en el ejercicio anterior, conveniente adaptado para vincular críticas y espectáculos. No es necesario controlar la asistencia del usuario que escribe la crítica al espectáculo, es decir, que exista una localidad vendida a dicho usuario para una sesión concreta.

El gestor de espectáculos debe proporcionar, al menos, las siguientes funcionalidades:

1. Dar de alta un espectáculo, incluyendo al menos una fecha de representación
2. Cancelar un espectáculo (todas las sesiones o una en particular)
3. Actualizar los datos de un espectáculo
4. Contabilizar la venta de entradas para una sesión de un espectáculo
5. Consultar las localidades disponibles para un espectáculo, dada una fecha de representación
6. Búsqueda de espectáculos por título o por categoría

7. Búsqueda de próximos espectáculos con entradas disponibles, indicando o no una categoría específica
8. Publicar una crítica para un espectáculo que ya se ha celebrado
9. Consultar las críticas de un espectáculo, dado su título
10. Eliminar críticas de un espectáculo, por parte del usuario que la creó
11. Valorar la utilidad de una crítica publicada por otro usuario

Al igual que en el ejercicio anterior, deberá implementarse un programa principal que dé acceso a las funcionalidades del gestor de espectáculos. Al comienzo, el programa debe permitir a un usuario registrarse o identificarse, si ya estuviera registrado. Se podrán reutilizar las funciones de gestión de usuarios anterior, considerando que deben existir dos tipos de usuarios:

- Usuario administrador, que puede acceder a las funciones relacionadas con la gestión de espectáculos.
- Usuario espectador, que puede crear y valorar críticas, así como ver la información de los espectáculos, pero no puede modificar sus datos ni cancelarlos.

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>