

# Memoria de la practica 1

Jaime Lorenzo Sánchez

Rocío Del Viejo Cupido

José Ángel Expósito Fernández

10 de octubre de 2021

# Índice general

<b>1. Introducción</b>	<b>3</b>
<b>2. Estructura del proyecto</b>	<b>4</b>
2.1. Contenido del directorio ficheros . . . . .	4
2.2. Contenido del directorio src/es/uco/pw . . . . .	5
2.2.1. Directorio comun/menu . . . . .	5
2.2.2. Directorio ejercicio1/critica . . . . .	5
2.2.3. Directorio ejercicio1/espectador . . . . .	6
2.2.4. Directorio ejercicio2/espectáculos . . . . .	6
<b>3. Funcionamiento del programa</b>	<b>8</b>
3.1. Inicio y acceso al sistema . . . . .	8
3.2. Acceso al gestor de críticas y espectáculos . . . . .	9
3.2.1. Gestor de críticas . . . . .	9
Crear una crítica . . . . .	9
Consultar todas las críticas disponibles . . . . .	9
Borrar una crítica del usuario . . . . .	9
Valorar la utilidad de una crítica de otro usuario . . . . .	9
Buscar las críticas de otro usuario registrado . . . . .	10
3.2.2. Acceso al perfil del usuario . . . . .	10
3.2.3. Gestor de espectáculos . . . . .	10
Dar de alta un espectáculo . . . . .	10
Cancelar un espectáculo (todas las sesiones o una en particular) . . . . .	10
Actualizar los datos de un espectáculo . . . . .	11

Contabilizar la venta de entradas para una sesión . . . . .	11
Consultar las localidades disponibles para un espectáculo . . . . .	11
Búsqueda de espectáculos por título o categoría . . . . .	11
Búsqueda de próximos espectáculos con entradas disponibles . . . . .	11
Consultar las críticas de un espectáculo . . . . .	11
Eliminar críticas de un espectáculo . . . . .	11
Valorar la utilidad de una crítica de otro usuario . . . . .	11

#### 4. Bibliografía y fuentes 12

# Capítulo 1

## Introducción

Uno de los aspectos más importantes como programadores es seguir las mejoras prácticas en el desarrollo. Para ello, se ha planteado resolver el siguiente problema mediante patrones de diseño:

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 interaccionar con las críticas de otros espectadores indicando su utilidad mediante un sistema de rating. Además, se desea implementar un gestor de espectáculos que emplee dicho gestor de críticas, y permite la creación de cualquier espectáculo en función del tipo, el cual dependerá de el número y fecha de las sesiones en las que tenga lugar su representación (puntual, de pase múltiple y de temporada).

Para ello, se ha implementado un programa mediante el lenguaje de programación Java y los patrones de diseño Singleton y Factory.

Antes de comenzar a realizar el programa, es importante estructurar la información que vamos a utilizar (ficheros, clases, etc) de la mejor forma posible.

Una vez estructurado correctamente los archivos del programa, debemos de comprobar el patrón de diseño más eficaz para la función que deseemos realizar. En este caso emplearemos el patrón de diseño Singleton para la creación del gestor de críticas y del gestor de usuarios, y utilizaremos el patrón de diseño Factory para la creación del gestor de espectáculos.

# Capítulo 2

## Estructura del proyecto

Al realizar la estructura del programa, se han creado una serie de directorios que almacenarán los archivos utilizados por nuestro programa.

Dichos directorios son los siguientes:

1. ficheros: Directorio que almacena la información de las críticas, espectáculos, etc creados por el usuario; así como el fichero de propiedades del programa.
2. src/es/uco/pw: Directorio principal de nuestro proyecto.

### 2.1. Contenido del directorio ficheros

El directorio ficheros contiene los siguientes archivos:

1. criticas.txt: Fichero de texto que almacena la información de las distintas críticas registradas en el sistema.
2. espectadores.txt: Fichero de texto que almacena la información de los espectadores que se han registrado en el sistema.
3. propiedades.properties: Fichero de propiedades que almacena la ubicación de los ficheros de texto, los cuáles serán necesarios para obtener la información ya registrada en el sistema.

## 2.2. Contenido del directorio `src/es/uco/pw`

El directorio `src/es/uco/pw` contiene los directorios y archivos necesarios para la correcta ejecución de nuestro programa.

Este directorio almacena los siguientes directorios:

1. `comun/menu`: Directorio que almacena los distintos menus a mostrar al usuario durante la ejecución del programa.
2. `ejercicio1`: Directorio que contiene los archivos, clases, etc necesarios para la creación del gestor de críticas.
3. `ejercicio2`: Directorio que contiene los archivos, clases, etc necesarios para la creación del gestor de espectáculos.

Además, se almacena en este directorio la clase principal del programa (`ProgramaPrincipal.java`), la cuál es utilizada para la ejecución del programa.

También se encuentran aquí el archivo ejecutable `.jar` y el fichero de texto plano `README.txt`.

### 2.2.1. Directorio `comun/menu`

Este directorio contiene los siguientes ficheros java:

1. `Menus.java`: Clase java que contiene la información necesaria para generar los distintos menus que se le mostrarán al usuario durante la ejecución del programa. Contiene las opciones del menú de acceso (registro o inicio de sesión), el del gestor de críticas y el del gestor de espectáculos.

### 2.2.2. Directorio `ejercicio1/critica`

El directorio `ejercicio1/critica` contiene las clases java necesarias para la creación y gestión de un gestor de críticas.

Dichas clases son las siguientes:

1. Critica.java: Clase que contiene las funcionalidades necesarias para la creación de una crítica.
2. EvaluacionUtilidadCritica.java: Clase que contiene las funcionalidades necesarias para realizar la evaluación de la utilidad de una crítica.
3. GestorCriticas.java: Clase que contiene las funcionales necesarias para la creación y utilización de un gestor de críticas, diseñado utilizando el patrón de diseño Singleton.

### **2.2.3. Directorio ejercicio1/espectador**

Directorio que contiene las clases java necesarias para la creación y gestión de un gestor de espectadores.

Dichas clases son las siguientes:

1. Espectador.java: Clase que contiene las funcionalidades necesarias para la creación de un espectador.
2. GestorEspectadores.java: Clase que contiene las funcionalidades necesarias para la creación de un gestor de espectadores, diseñado utilizando el patrón de diseño Singleton.

### **2.2.4. Directorio ejercicio2/espectáculos**

1. CategoriaEspectaculo.java: Enumeración de las distintas categorías de un espectáculo: concierto, monólogo y obra de teatro.
2. EspectaculoMultiple.java: Clase que contiene las funcionalidades necesarias para la creación de un espectáculo de pase múltiple.
3. EspectaculoPuntual.java: Clase que contiene las funcionalidades necesarias para la creación de un espectáculo puntual.
4. EspectaculoTemporada.java: Clase que contiene las funcionalidades necesarias para la creación de un espectáculo de temporada.

5. `FactoriaEspectaculos.java`: Clase que sirve como plantilla para crear cualquier tipo de espectáculo.
6. `GestorEspectaculos.java`: Clase que contiene las funcionalidades necesarias para dar de alta espectáculos, cancelarlos, modificarlos, mostrarlos y buscarlos.
7. `InterfazEspectaculo.java`: Clase que almacena las declaraciones de las funciones comunes de un espectáculo.
8. `SesionEspectaculo.java`: Clase que contiene las funcionalidades necesarias para gestionar la fecha y hora en la que se realiza la sesión de un espectáculo.



# Capítulo 3

## Funcionamiento del programa

### 3.1. Inicio y acceso al sistema

Una vez el usuario ha iniciado el programa, se cargarán en los gestores (espectáculos, críticas y espectadores) la información registrada en el sistema ( dicha información se obtendrán utilizando un fichero de propiedades que almacena la ubicación de los ficheros de texto con los datos ya registrados en el sistema).

Una vez obtenidos los datos ya registrados en el sistema ( en caso de error, se notificará al usuario), se procederá a mostrar al usuario un menú con las siguientes funcionalidades:

1. Registrar espectador: En caso de que el usuario no tenga creada una cuenta, se procederá a crear una cuenta. Para ello, se le pedirá al usuario sus datos personales necesarios para crear la cuenta, además de realizarse las siguientes comprobaciones:
  - a)* Nombre del usuario unico: Se comprobará si el nombre de usuario del usuario (nick) ya está registrado o no en el sistema. En caso afirmativo, se notificará al usuario un error y no se creará la cuenta del usuario; en caso contrario, se continuará con la siguiente comprobación.
  - b)* Correo en formato correcto: Se comprobará si el correo introducido por el usuario está en un formato de correo válido ( xxx@xxx.es/com).

2. Iniciar sesión: Se solicitará al usuario introducir su nombre de usuario (nick). En caso de que este se encuentre, se procederá a mostrarse el menú de las posibles opciones que tiene, el gestor de críticas del ejercicio 1 y el gestor de espectáculos del ejercicio 2. En caso de no encontrarse este, se avisará al usuario de que el nick no se encuentra registrado en el sistema.

## **3.2. Acceso al gestor de críticas y espectáculos**

Si el usuario se ha identificado correctamente en el sistema se mostrará el siguiente menú:

### **3.2.1. Gestor de críticas**

Desde esta opción se accederá al gestor de críticas, desde el cual podrá realizar a través de otro menú las siguientes opciones:

#### **Crear una crítica**

Primero se introducirá su título para comprobar si ya existe. Si ya está registrada no se creará, en caso contrario se pedirá una puntuación (número entero entre 0 y 10) y una reseña. Con todos estos datos junto con el nombre del usuario la crítica se guardará en una lista con las demás.

#### **Consultar todas las críticas disponibles**

Se mostrarán los datos de cada una de las críticas incluidas en la lista.

#### **Borrar una crítica del usuario**

Se pedirá el título de la crítica a eliminar y se buscará en la lista de críticas. Si se encuentra y se comprueba que el usuario es el autor de la misma, se procederá a su eliminación.

#### **Valorar la utilidad de una crítica de otro usuario**

Se podrán dar dos casos: que la crítica aún no tenga ninguna valoración o que sí la tenga. En el primer caso simplemente se añadirá la nueva valoración, en el segundo se comprobará si el usuario ya la había valorado antes, y en tal caso ya no podrá evaluarla.

### **Buscar las críticas de otro usuario registrado**

Recibirá como parámetro el nick del usuario que busquemos, recorrerá la lista que contiene todas las críticas y se mostrará las que pertenezcan a ese usuario.

### **3.2.2. Acceso al perfil del usuario**

Aquí se le mostrará al usuario sus datos personales, tales como su nombre, apellidos, nick y correo electrónico con el que se ha registrado.

### **3.2.3. Gestor de espectáculos**

Desde esta opción se accederá al gestor de espectáculos que permitirá dar de alta, modificar y mostrar espectáculos. Según su modo de programación, existirán tres tipos de espectáculos: puntual (una representación), de pase múltiple (varias representaciones) y de temporada (se celebra periódicamente un día de la semana entre una fechas de inicio y fin). Para crear y modificar adecuadamente los espectáculos según su modo de programación se ha hecho uso del patrón de diseño Factory. Hay que tener en cuenta que existen dos tipos de usuarios: administrador y espectador, y habrá ciertas funcionalidades que no pueden realizar ambos. Así, quedará reservada a los administradores la posibilidad de dar de alta los espectáculos, modificarlos y cancelarlos. Por último, se navegará entre las distintas opciones del gestor a través del siguiente menú:

#### **Dar de alta un espectáculo**

Se solicitarán los distintos valores al usuario para registrar un nuevo espectáculo, que se guardará en la lista correspondiente a su tipo.

#### **Cancelar un espectáculo (todas las sesiones o una en particular)**

Se solicitará el título del espectáculo y se comprobará que este sea un espectáculo existente con anterioridad, en caso de que exista, este será eliminado totalmente o sólo una de sus sesiones.

### **Actualizar los datos de un espectáculo**

Se solicitará el título del espectáculo para actualizar, se buscará y se solicitarán todos los nuevos campos que se desee actualizar.

### **Contabilizar la venta de entradas para una sesión**

Se indicará el número máximo de entradas que existen para una sesión (estén disponibles o no).

### **Consultar las localidades disponibles para un espectáculo**

Dada una fecha de representación, se buscará la sesión específica del espectáculo y se dará el número de entradas que quedan disponibles para esta.

### **Búsqueda de espectáculos por título o categoría**

Se pedirá el campo por el que se desea buscar y se buscará el espectáculo, mostrando sus datos por pantalla en que caso de encontrarse.

### **Búsqueda de próximos espectáculos con entradas disponibles**

Se buscarán cuáles son los próximos espectáculos a los que se puede asistir, además se podrá especificar la categoría del espectáculo al que se desea asistir.

### **Consultar las críticas de un espectáculo**

Tras solicitar su título, se buscarán las posibles críticas asociadas al espectáculo y se mostrarán.

### **Eliminar críticas de un espectáculo**

Si el usuario registrado es el creador de una crítica, este podrá decidir eliminarla.

### **Valorar la utilidad de una crítica de otro usuario**

Al igual que en el gestor de críticas del primer ejercicio, se podrá valorar por un usuario que no haya valorado la crítica de manera previa.

# Capítulo 4

## Bibliografía y fuentes

Enunciado de la practica:

[https://moodle.uco.es/m2122/pluginfile.php/118332/mod\\_resource/content/4/ProgWeb\\_2122\\_EnunciadoPractica1.pdf](https://moodle.uco.es/m2122/pluginfile.php/118332/mod_resource/content/4/ProgWeb_2122_EnunciadoPractica1.pdf)

Manual de LaTeX:

<https://manualdelatex.com/tutoriales>

Wiki con tutoriales de Overleaf:

<https://es.overleaf.com/learn/latex/Tutorials>