**Programa CRUD MySQL**

Alfonso López Delgado

1ºDAM

He comenzado el proyecto en GIT creando un repositorio para guardar el progreso que voy desarrollando y he utilizado el entorno de eclipse para escribir el código, pruebas Junit y respectiva documentación JavaDoc.

Para el diagrama de Gantt he utilizado la aplicación de CANVA en el cual apuntaba las horas que realizaba en cada momento del proyecto.

**EXPLICACIÓN DEL PROGRAMA**

He estructurado el programa en tres paquetes:

* ***Paquete utilidades*** 🡪 Que contendrá métodos de interacción con el usuario, validaciones y representaciones gráficas:
  + ***Validaciones*** 🡪 Esta clase se encargará de la interacción con el usuario, pidiendo textos y datos numéricos por consola y validaciones de los datos introducidos.
  + ***Utilidades*** 🡪 Contendrá la representación gráfica del menú, el menú de elección y la creación de los objetos.
* ***Paquete datos*** 🡪 Se encargará de realizar las conexiones con MySQL para el almacenamiento de los datos, contiene la clase objeto y los métodos para el tratamiento de los datos con la base de datos.
  + ***Pelicula*** 🡪 Es la clase objeto que contiene el constructor, getter, setter y toString.
  + ***Sql*** 🡪 En esta clase se realizará la mayoría de procedimientos que tengan que ver con la base de datos y el tratamiento de sus datos.
* ***Paquete control*** 🡪 Contiene la clase con la que se va a ejecutar el programa.

Para este programa he reutilizado algunos métodos que desarrollé en la anterior práctica 2 para ahorrar tiempo, además de utilizar la estructura de conexión SQL que proporcionaste en las tutorías.

**EXPLICACIÓN DEL CÓDIGO**

**PAQUETE UTILIDADES**

* 1. **Utilidades 🡪** Es la clase donde confluyen la mayoría de los elementos creados en otras clases del programa.
     + **Método caracteristicasPeli(texto) 🡪** Método el cual generará un objeto de la clase Películas y se introducirán los valores correspondientes, reutilizando los métodos para introducir datos por teclado de la clase “Validacion”.
     + **Método menú() 🡪** Se encarga de mostrar el menú gráfico que se lanzará por consola y de implementar la mecánica del menú para poder interactuar con las opciones creadas, mediante un switch.
       - **Case 1:** Llama al método “CrearPeli”, el cual recibirá un objeto creado, en la clase “caracteristicasPeli();” y lo añadirá al objeto “cine” donde se encuentra el ArrayList.
       - **Case 2:** Llama al método “ModificarPeli”, el cual recibirá la posición numérica que le proporcionará el método “BuscarPeli()” del dato introducido por teclado por el usuario y lo modificará.
       - **Case 3:** Llama al método “EliminarPeli”, el cual recibirá la posición numérica que le proporcionará el método “BuscarPeli()” del dato introducido por teclado por el usuario y lo eliminará.
       - **Case 4:** Llama al método “MensajeBuscar” y mostrará por pantalla una de las opciones predefinidas.
       - **Default:** Mostrará un texto de salida

Por último se controlaran las excepciones correspondientes.