

# CINE ONLINE MEK.

Trabajo teórico de fundamentos de programación.



**Universidad de Castilla La Mancha.**

Luis Miguel B, Paulino Esteban B, Kevin Alfonso G, Efraín Enrique R.



## Contenido

ENUNCIADO.....	4
Cine online.....	4
ANALISIS DEL PROBLEMA.....	5
MAIN .....	11
Realización del programa.....	12
Codigo. ....	13
ADMINISTRADORES.....	16
Creación de un Menú.....	16
Propósito .....	16
Realización.....	16
Código .....	17
Dar de alta películas .....	18
Código .....	19
Dar de alta Críticos .....	20
Código .....	21
CRITICOS.....	23
ESQUEMA “ALTAS CRITICOS” .....	23
Justificacion del esquema .....	23
Esquematizacion del programa.....	23
Esquema apartado críticos.....	24
EXPLICACIÓN DEL CODIGO .....	25
modulolacion del código .....	25
CODIGO .....	27
USUARIOS.....	32
Proposito .....	32
Realización.....	32
menorMedia() , mayorMedia() y masDeTresE() .....	32
criticoExigente() .....	33
Código. ....	34
Clase Usuarios.java.....	34
Clase Calculos.java .....	37
Clase CalculosMenorYMayorMedia.java.....	41

JURADO .....	47
Realización del programa.....	47
Programa realizado. ....	47
Conclusiones y problemas encontrados a la hora de realizar el proyecto. ....	49
Mejoras del proyecto para futuras actualizaciones. ....	49
Bibliografía. ....	50
Software usado para el desarrollo del programa.....	50

## ENUNCIADO

### Cine online

Un cine online permite almacenar las películas de estreno y las opiniones de  $M$  críticos de cine en forma de estrellas, siendo 1 estrella la puntuación mínima y 5 la puntuación máxima. Para facilitar la tarea del jurado, se desea elaborar un programa que se encargue de gestionar toda la información, contabilizando los votos emitidos y obteniendo el ganador. Así pues, dicho programa consistirá en:

- Deberá preguntar qué tipo de rol tenemos, si administrador, crítico de cine o usuario.
- En caso de que seamos administrador:
  - o Leer del teclado el número  $N$  de películas que tiene el cine
  - o Leer del teclado los títulos de las  $N$  películas
  - o Leer del teclado el número  $M$  de críticos
  - o Leer del teclado el nombre de los  $M$  críticos
  - o Una vez que hayamos escrito el nombre de todos los críticos, crearemos un fichero por cada crítico y un único fichero que contenga el nombre de las películas. Si fuese necesario, se puede crear un fichero que contenga todos los críticos.
- En caso de que seamos críticos:
  - o El sistema nos preguntará nuestro nombre y lo buscará en el sistema, donde el administrador lo habrá escrito previamente
  - o Si el nombre se encuentra en el fichero, le mostrará el nombre de cada película y le pedirá la puntuación de cada una (del 1 al 5)
  - o Una vez hayamos terminado, guardará en el fichero correspondiente del crítico, las calificaciones de cada película junto a su nombre. Por ejemplo, el fichero Fernando.txt tendrá:

Máquina de Turing - 4

Scarface – 5

- En caso de que seamos usuarios, nos mostrará un menú que conste de las siguientes opciones:
  - o Mostrar película con menor media
  - o Mostrar película con mayor media
  - o Mostrar el crítico que menor puntuación media ha dado a todas las películas (Si hay varias, mostrar solo una de ellas)
  - o Mostrar las películas con más de 3 de media
  - o Finalizar el programa
- Al finalizar el programa, guardar en un fichero de texto los nombres de las películas y las puntuaciones media asociadas. Para ello, se considerará que en una línea aparecerá el nombre de la película, y en la siguiente línea la puntuación media.

## ANALISIS DEL PROBLEMA.

Se necesita crear un sistema gestor de películas para un cine online, con varios roles para diferentes tareas. Entre las que se encuentra:

Un **administrador** del sistema capaz de gestionar las películas que se dan de alta en el cine online, pudiéndose añadir tantas como se desee y guardarlas en una memoria no volátil.

También, dicho rol, puede dar de alta a nombres de personas, con el rol de críticos en el sistema.

**Críticos de cine**, son los encargados de calificar las películas que se han dado previamente de alta por el administrador del sistema. Su evaluación se realiza dependiendo del nombre del crítico, que se buscare que exista y posteriormente se creara una ficha de crítico con su nombre, donde se almacenaran las notas de las películas que evalúe.

Los **usuarios**, tienen un menú personal, donde pueden ver las películas en función de su nota media, tanto las mas altas como bajas, también podrán filtrar las películas por una nota mayor a 3 estrellas, una cuarta opción que poseen en su menú es, ver el crítico mas exigente a la hora de calificar las películas.

Un **jurado**, que usa el sistema para decidir quién es la película ganadora.

Por tanto, se decide realizar un menú que almacene todos estos roles y al iniciar el sistema, pregunte por el tipo de persona y cargo que tiene.

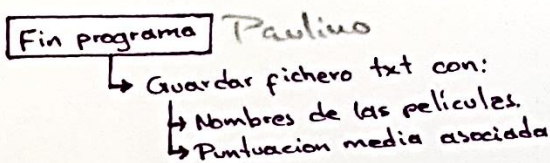
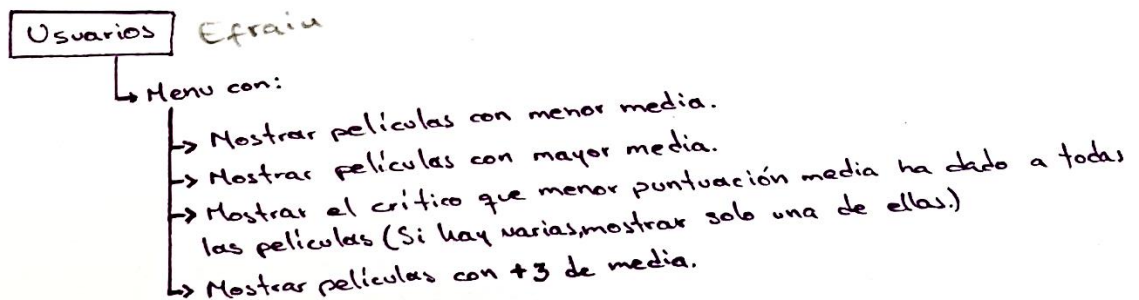
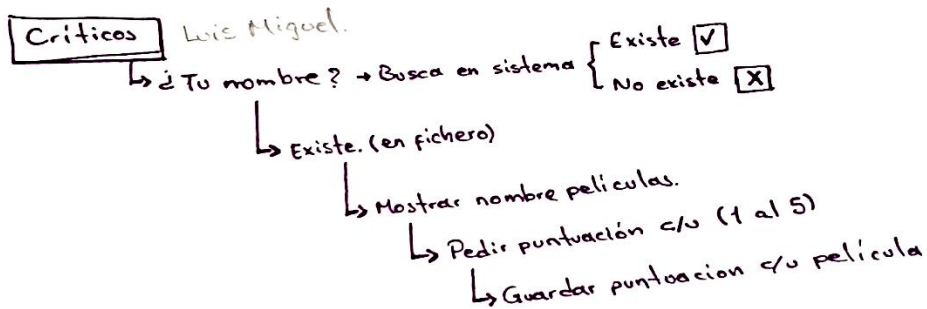
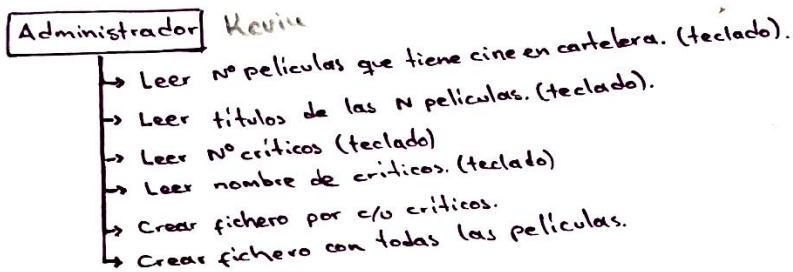
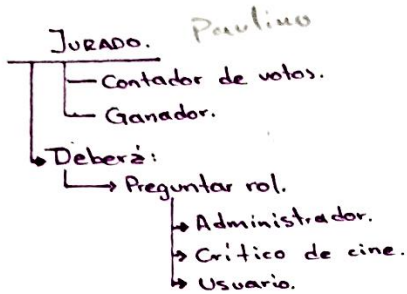
Para la programación inicial del proyecto, se siguió la siguiente división del trabajo, basándose en el rol que le corresponde a cada integrante.

## Cine online.

Almacen de películas de estreno.

Opiniones de M criticos. de cine.

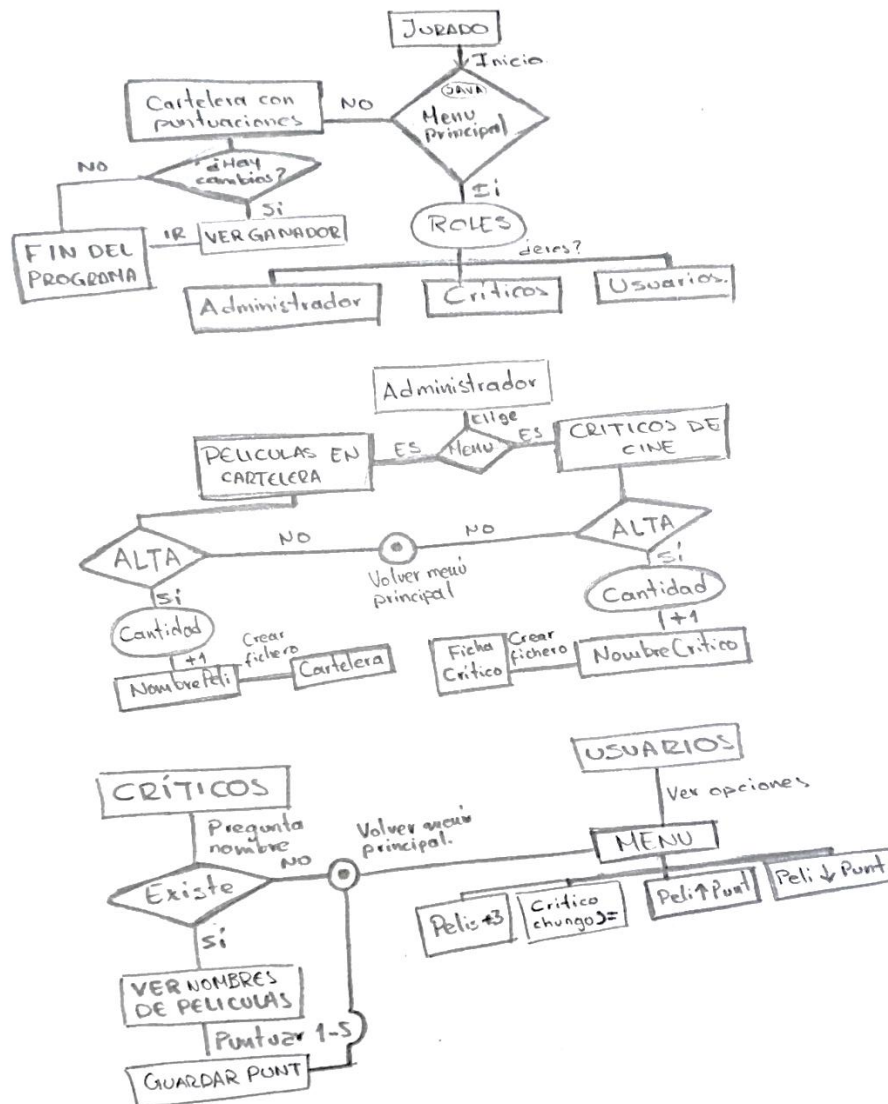
\* → \*\*\*\*\*



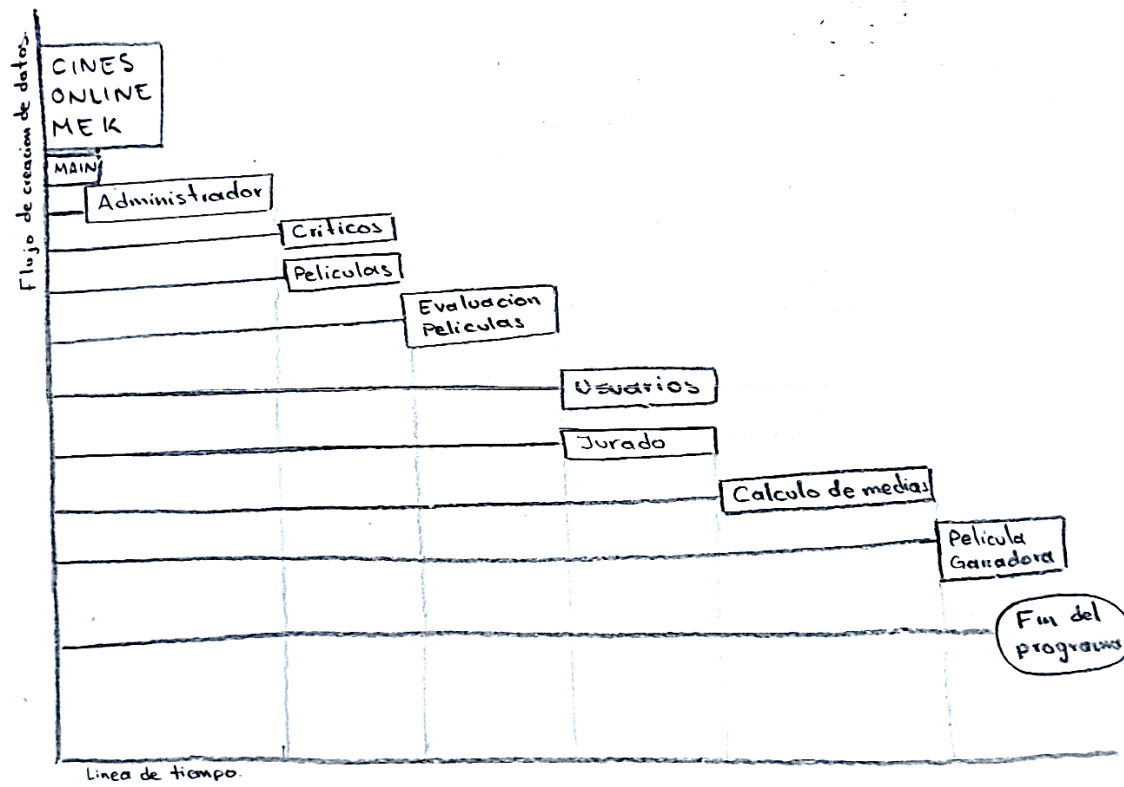
AverageScore.txt	
→ Scarface	
→	4
→ Matrix	
→	3.5



Seguidamente, se realizo un diagrama de flujo modular del programa, para poder ir viendo las diferentes secuencias que debe realizar cada rol y ver sus posibles dependencias de unos con otros.



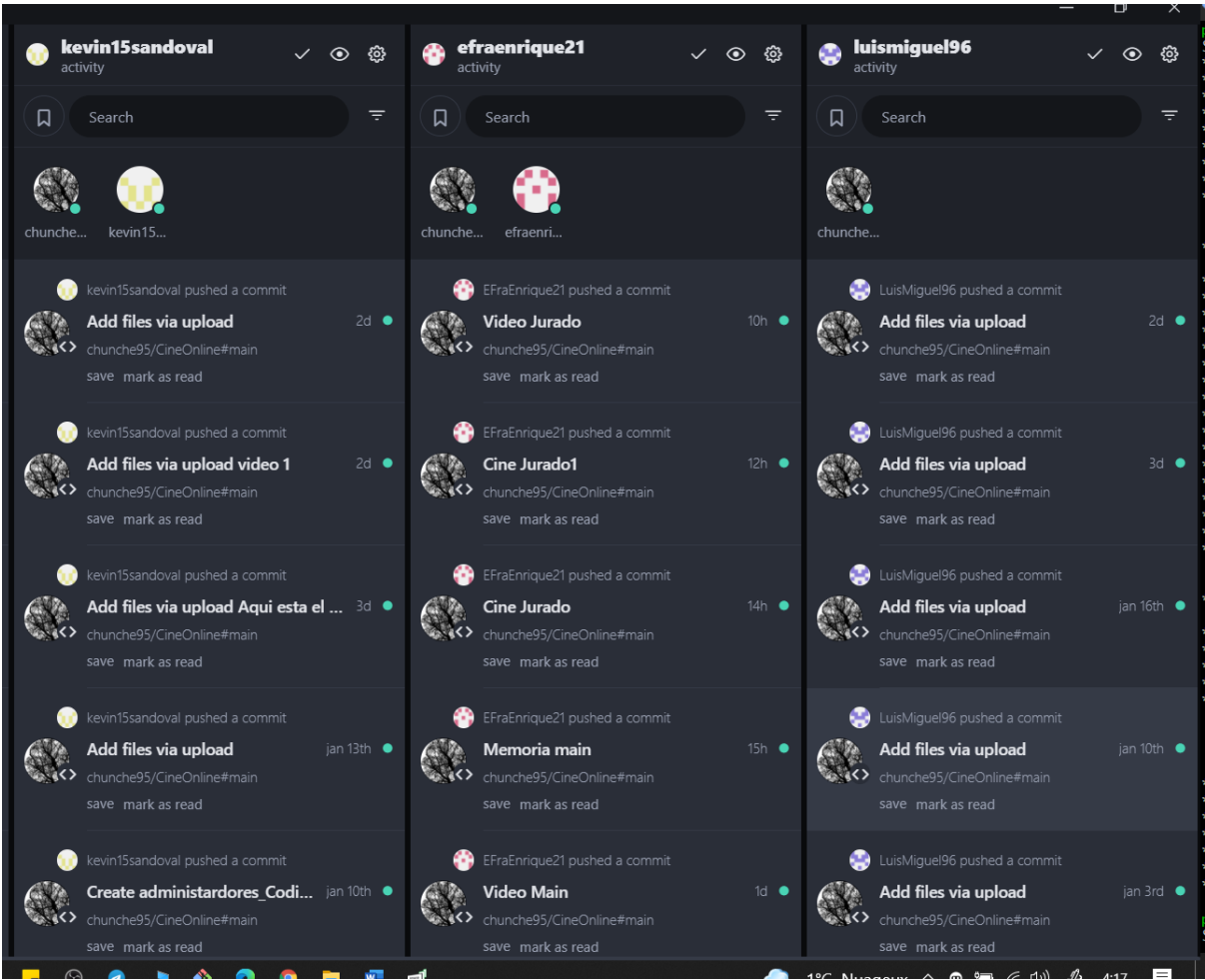
Terminando con la unificación total del proyecto y realización de las pruebas pertinentes para ver el funcionamiento global del programa.



Para llevar un control de versiones del programa, hemos realizado el programa en un repositorio de GitHub, usamos dicha herramienta para gestionar los cambios en el código fuente a lo largo del tiempo de desarrollo del proyecto.

Además de ayudarnos a tener todo el programa de manera independiente del tiempo de otro compañero. Como se puede ver a continuación, se adjuntan la trazabilidad de la terminal de Git y del gestor de tareas de DevHub.

```
MINGW64/z/githu...erroneos_para_tirar/CineOnline
$ git log
commit b3ce2eb (HEAD -> main, origin/main, origin/HEAD) Video Jurado
commit 681ebf9 Cine Jurado
commit 5f11bcd Cine Jurado
commit 88f5ccd Memoria main
commit 36653aa Video Main
commit 68f2b45 Memoria Main
commit 135f290 Delete word main.pdf
commit ce46c0f Memoria
commit 0ef0116 Merge branch 'main' of https://github.com/chunche95/CineOnline
* d62a5ca Add files via upload
commit 811c1b4 Entrega - Proyecto Cine
* fdac335 Add files via upload
commit 955d5a1 Add files via upload
commit 674a437 Add files via upload
commit 0fff37d Nueva version del Cine Online
commit cb68848 Actualizacion - Solo ZIP con codigo
commit 398ea5d Add files via upload
commit ec3fe08 Update Usuarios.java
commit 28a1867 Update Usuarios.java
commit 58e028e Practica Final. - Entrega
commit 4cd3ba9 Practica Final. - Entrega
commit 8ba178d Practica Final. - Entrega
commit c59ef03 Practica Final. - Entrega
commit 72939ac Add files via upload
commit 138c3d4 Pte de crear la lectura de fichero de peliculas y guardarlas en los ficheros de criticos
commit a91a06a Administradores - Terminado.
commit d27b8f Administradores - Terminado - Todo OK
commit 970ce91 Merge branch 'main' of https://github.com/chunche95/CineOnline
* 4e59c82 Add files via upload
commit 84273fc criticos y usuarios modificado
commit 45923d2 Create administradores_Codigo
commit 4ce3499 Add files via upload
commit 95856ec cine
commit 329ea53 Maquetación por programador - Pruebas individuales
commit 3b42c7b Voy a reorganizar ficheros
commit 50b8bae (refs/stash) On Rediseño: !!GitHub_Desktop-Rediseño>
commit abc9c2e index on Rediseño: 5a1a40b Add files via upload
commit 5a1a40b (origin/Rediseño, Rediseño) Add files via upload
commit d4959db Create administradores
commit dd7c45b Cine online
commit 33e00a1 Descripción del programa
commit ff974ed Actualización del README inicial
commit a72c478 Ficheros de proyecto de cine online UCLM
commit 662a412 Initial commit
MINGW64/z/githu...erroneos_para_tirar/CineOnline (main)
$
```



## MAIN

Un cine online permite almacenar las películas de estreno y las opciones de M críticos de cine en forma de estrellas, siendo 1 estrella la puntuación mínima y 5 la puntuación máxima. Para facilitar la tarea del jurado, se desea elaborar un programa que se encargue de gestionar toda la información, contabilizando los votos emitidos y obteniendo el ganador. Así pues, dicho programa consistirá en:

- **Deberá preguntar qué tipo de rol tenemos, si administrador, crítico de cine o usuario.**
- En caso de que seamos administrador:
  - ✓ Leer del teclado el número N de películas que tiene el cine
  - ✓ Leer del teclado los títulos de las N películas
  - ✓ Leer del teclado el número M de críticos
  - ✓ Leer del teclado el nombre de los M críticos
  - ✓ Una vez que hayamos escrito el nombre de todos los críticos, crearemos un fichero por cada crítico y un único fichero que contenga el nombre de las películas. Si fuese necesario, se puede crear un fichero que contenga todos los críticos.
- En caso de que seamos críticos:
  - ✓ El sistema nos preguntará nuestro nombre y lo buscará en el sistema, donde el administrador lo habrá escrito previamente
  - ✓ Si el nombre se encuentra en el fichero, le mostrará el nombre de cada película y le pedirá la puntuación de cada una (del 1 al 5)
  - ✓ Una vez hayamos terminado, guardará en el fichero correspondiente del crítico, las calificaciones de cada película junto a su nombre. Por ejemplo, el fichero Fernando.txt tendrá:

Máquina de Turing - 4

Scarface – 5

- **En caso de que seamos usuarios, nos mostrará un menú que conste de las siguientes opciones:**
  - ✓ Mostrar película con menor media
  - ✓ Mostrar película con mayor media
  - ✓ Mostrar el crítico que menor puntuación media ha dado a todas las películas (Si hay varias, (mostrar solo una de ellas) o Mostrar las películas con más de 3 de media
  - ✓ Finalizar el programa

- **Al finalizar el programa, guardar en un fichero de texto los nombres de las películas y las puntuaciones medias asociadas. Para ello, se considerará que en una línea aparecerá el nombre de la película, y en la siguiente línea la puntuación media**

#### **Elaboraremos un menú de opciones.**

Elaboraremos un menú de opciones que se muestre en pantalla las opciones que tenemos disponibles, en este caso debemos escoger el rol con el que deseamos entrar al cine online, porque el programa nos preguntará que rol deseamos y nos dará 4 opciones en las que deberemos de seleccionar una:

1. Jurado.
2. Administrador.
3. Critico
4. Usuario

#### **Realización del programa.**

Dentro de esta clase creamos un método menú, el cual posteriormente será llamado por el método **main**, para así mostrar lo que se encuentran en dicho método, volviendo al método menú, aquí elaboraremos un menú de opciones para que el usuario pueda elegir lo que desea realizar en el cine online, así que primero declararemos una variable String llamándola n para el nombre de usuario, utilizaremos un try catch para controlar si el usuario introduce un dato que no concuerda con lo solicitado (por ejemplo en este caso, el usuario introduce una letra en lugar de un número), cuando esto ocurra saltará un mensaje por medio de un `System.out.println` que le indicará al usuario que debe ingresar un valor numérico.

Posteriormente, el programa nos preguntará que introduzcamos por medio de un scanner la opción que deseamos realizar entre las cuales tenemos los roles de jurado, administrador, crítico o usuario y otra opción por si desea finalizar el programa.

La creación de este menú comenzará primeramente con el scanner que solicita cual rol desea realizar, después declaramos una variable que llamaremos "selección" aquí se guardará el dato que introduce el usuario y nos servirá como variable para controlar el Switch-Case que crearemos a continuación para poder determinar, redirigirá al usuario a la opción donde desea ir por ello creamos 5 casos diferentes los cuales son:

Case 1: Si el usuario ingresa esta opción nos redirigirá al método de los jurados

Case 2: En el caso dos lo que sucede es que nos redirigirá a la sección de administradores y esto lo hace llamando al método Administradores ya aquí adentro podrá realizar todos los procesos que mencionamos anteriormente

Case 3: En el caso dos lo que sucede es que nos redirigirá a la sección de críticos y esto lo hace llamando al método críticos ya aquí adentro podrá realizar todos los procesos que mencionamos anteriormente

Case 4: En el caso dos lo que sucede es que nos redirigirá a la sección de usuarios y esto lo hace llamando al método usuarios ya aquí adentro podrá realizar todos los procesos que mencionamos anteriormente

Case 0: En este caso lo que sucede es que salimos del cine online, por ello elaboramos un un System.out.println donde imprimirá “hasta pronto”, dando por finalizada esa parte del programa.

Luego hacemos un default para así poder controlar que el usuario ingrese una de las opciones disponibles ya que si no lo hace le aparecerá un mensaje por medio de un System.out.println en el que indicaremos sucedió un error inesperado y por ello que compruebe la opción seleccionada, después nos redirigirá al menú principal para volver a escoger alguna opción.

### Codigo.

```
package net.humancomputing.cineonline;

import java.io.IOException;

import static java.lang.Thread.sleep;

import java.util.InputMismatchException;

import java.util.Scanner;

/**
 * @version 0.4
 * @author Paulino Esteban Bermúdez Rodríguez | humancomputing.net
 */

public class Main {

    public static void main(String[] args) throws InterruptedException, IOException {

        menu();

    }

    public static void menu() throws IOException {

        String n = System.getProperty("user.name");

        try {

            System.out.println("+-----+");

            System.out.println("| CINE ONLINE MEK. - FUND. PROGRAMACION. |");

            System.out.println("+-----+");

            System.out.println("");

            System.out.println("Bienvenido al programa de Cines Online de estrenos.");

            System.out.println("Empecemos...");

            System.out.println("+-----+");

            System.out.println("| Identificación de usuario: |");

            System.out.println("+-----+");

            System.out.println("=> "+n);
```

```

System.out.println("Escriba el numero de su role.");

System.out.println("");

System.out.println("\t 1) Jurado.");

System.out.println("\t 2) Administrador.");

System.out.println("\t 3) Critico.");

System.out.println("\t 4) Usuario");

System.out.println("");

System.out.println("\t 0) Finalizar el programa.");

Scanner op = new Scanner(System.in);

int seleccion = op.nextInt();

switch(seleccion){

case 1:

//Jurado();

menu();

break;

case 2:

Administrador.administrador();

menu();

break;

case 3:

Critico.critico();

menu();

break;

case 4:

Usuarios.usuarios();

menu();

break;

case 0:

System.out.println(n + " hasta pronto!");

System.exit(0);

break;

default:

System.out.println("#####
#####");

System.out.println("Error inesperado!\n Compruebe la opcion seleccionada y vuelva a intentarlo de
nuevo.\n");

System.out.println("\t EY! EY! YOUUU!!!");

System.out.println("");

System.out.println(n+" debes fijarte mas en las opciones disponibles.\n Volvamos de nuevo a intentarlo...");

```

```

        System.out.println("#####");
        menu();
    }
} catch (InputMismatchException ex){
    System.out.println("#####");
    System.out.println("\t ERROR!!!");
    System.out.println("\t "+n);
    System.out.println("\t Hemos metido la pata! ");
    System.out.println("\t Debes introducir un valor numerico.");
    System.out.println("\t ");
    System.out.println("\t Volvamos a intentarlo de nuevo...");
    System.out.println("#####");
    System.out.println("\n\n");
    menu();
} catch(Exception e){
    System.out.println("#####");
    System.out.println("\t ERROR!!!");
    System.out.println("\t ");
    System.out.println("\t Nooo! ocurrio un error inesperado del sistema. Tienes más detalles a continuacion. ");
    System.out.println("");
    System.out.println("\t"+ n + " aqui tienes mas info.");
    System.out.println("\t DESCRIPCION DEL ERROR:");
    System.out.println(e);
    System.out.println("#####");
    System.out.println("\n\n");
    menu();
} finally {
    System.out.println("+-----+");
    System.out.println("| Fin del programa. |");
    System.out.println("| Gracias por evaluar CINE ONLINE MEK. |");
    System.out.println("| |");
    System.out.println("| @universidad: UCLM |");
    System.out.println("| @subject: Fundamentos de programacion I. |");
    System.out.println("| @version: 1.01 |");
}
}
}

```



## ADMINISTRADORES

Para la realización de esta parte del programa se requiere lo siguiente:

- Leer del teclado el número *N* de películas que tiene el cine
- Leer del teclado los títulos de las *N* películas
- Leer del teclado el número *M* de críticos
- Leer del teclado el nombre de los *M* críticos
- Una vez que hayamos escrito el nombre de todos los críticos, crearemos un fichero por cada crítico y un único fichero que contenga el nombre de las películas. Si fuese necesario, se puede crear un fichero que contenga todos los críticos.

### Creación de un Menú

#### Propósito

La realización del menú es para que el administrador pueda escoger qué es lo que desea hacer y además sepa cuáles son algunas de sus funciones como administrador, así que para llevarlo a cabo empezamos preguntado al administrador que desea hacer, de las cuales hay 3 opciones

1. Dar de alta una película
2. Dar de alta un crítico
3. salir

Así que para ello le pedimos que introduzca el número de la opción que quiere hacer, lo cual al introducirlo le conducirá a la opción que desea realizar.

#### Realización

La realización de este menú está compuesto por un bucle `while`, un `Switch-case` y un `try-catch`, primeramente empezamos declarando dos variables

1. un **`int`** que llamamos **`opción`**, la cual utilizaremos como parámetro para escoger cuál opción tomar, dicho parámetro que será introducido por el usuario
2. un **`boolean`** que llamamos **`menú`** y lo iniciaremos igual a **`true`**, el cual nos servirá para establecer cuándo se llevará a cabo el bucle `while` y cuando finalizara

Teniendo nuestras dos variables exportamos un `Scanner` para poder introducir datos por teclado, posteriormente abrimos el bucle `while` que se realizará siempre que `menu=true`

Declaramos las opciones que tiene al administrador por medio de un `System.out.println` y le pedimos que introduzca lo que desea realizar, para que el administrador pueda ir a la opción

*escogida utilizamos el Switch-Case con tres casos posibles, los cuales serán las tres opciones posibles*

*Hay que tener en cuenta que el parámetro de Switch es la variable opción la cual es determinada por el administrador, siguiendo con los casos serán los siguientes*

1. *Case 1: Este caso te redirige a otro método, el cual llamamos películas, donde se lleva a cabo el alta de las nuevas películas*
2. *Case 2: Este caso te redirige a otro método, el cual llamamos críticos, donde se lleva a cabo el alta de los nuevos críticos*
3. *Case 3: vuelve la variable menú en false para así salir del bucle while y dar fin a esta parte del programa*

*Dentro de este Switch se encuentra un default para así controlar el rango de las respuestas válidas de la opción, ya que cuando pongan un dato mayor que tres les enviará un mensaje indicando que no es válido dicho número y que por favor escojan una de las opciones*

*En el caso de que el administrador introduzca en lugar de un número, otro tipo de símbolo se le mandará un error, diciendo que ese dato no es válido y por favor que introduzca el número de la opción que desea realizar y esto se lleva a cabo gracias al try-catch. Es importante destacar que todo este proceso sucede en el método main.*

## Código

```
public static void administrador() throws IOException {
    menu();
} // Cierre del metodo administrador()

public static void menu() throws IOException {
    // Menu de administrador.

    System.out.println("+-----+");
    System.out.println("| BIENVENIDO AL MENU DE ADMINISTRADOR. |");
    System.out.println("+-----+");
    System.out.println("Opciones:");
    System.out.println("");
    System.out.println("1) Dar de alta peliculas en el cine.");
    System.out.println("2) Dar de alta criticos en el cine.");
    System.out.println("-----");
    System.out.println("| 0) Volver al menu principal. |");
}
```

```

System.out.println("-----");

System.out.println("Introduzca el numero de la opcion.");

// Guardamos la opcion seleccionada.

Scanner op = new Scanner(System.in);

// Casteamos el Scanner a tipo Integer

int opcion = op.nextInt();

// Verificamos que la opcion escrita sea valida.

switch(opcion){

    case 1:

        // Dirigimos a dar de alta nuevas peliculas en el cine.

        altaPeliculas();

        menu();

        break;

    case 2:

        // Dirigimos a dar de alta nuevos criticos en el cine.

        altaCriticos();

        menu();

        break;

    case 0:

        // Volvemos al menu principal del programa.

        Main.menu();

        break;

    default:

        System.out.println("Error en la opción que ha escrito, vuelva a intentarlo de nuevo.");

        // Volvemos a mostrar el menu de Administrador

        menu();

} // Cierre del switch

} // Cierre del metodo menu()

```

## Dar de alta películas

Para llevar a cabo esta tarea que corresponde a los administradores en nuestro código creamos un nuevo método en este caso nosotros llamaremos “películas”.

Primeramente, ya que lo que deseamos es que el administrador ingrese las películas por teclado introducimos un Scanner, además como estas películas tienen que guardarse en un fichero.

*La formación de un fichero es un proceso donde utilizamos la función File, la cual pusimos el nombre de "Cartelera" y le decimos cree un fichero tipo txt, posteriormente para poder escribir dentro de ese fichero exportamos las funciones PrintWriter y FileWriter, las cuales nos permitirán guardar los datos que introducimos*

*para aplicar este procedimiento utilizaremos un bucle do-while, el cual será condicionado por una variable booleana que creamos la cual llamaremos maspeliculas", dentro del do preguntaremos al administrador cuales peliculas desea agregar, estas películas posteriormente serán guardadas en un fichero*

*Cuando White finaliza entonces se cierra el fichero y manda un mensaje agradeciendo su trabajo, con esto se finaliza el método y volvemos al menú.*

## Código

```
Scanner altaP;

System.out.println("");

System.out.println("+-----+");

System.out.println("| MENU DE ALTA DE PELICULAS. |");

System.out.println("+-----+");

System.out.println("");

System.out.println("Introduzca cuantas peliculas quiere dar de alta: \n");

// Guardamos el numero de peliculas que se quieren dar de alta.

altaP = new Scanner(System.in);

// El Scanner lo pasamos a tipo Int

int nuevaP = altaP.nextInt();

/*****

 * Escritura *

 *****/

/* -----

 * Declaracion de variables.

 ----- */

int contador=0;

File HojaPapel; // variable del programa.

PrintWriter Boli; //Como un SYSO pero para ficheros.
```

```
String carteleraCine = "cartelera//cartelera.txt"; // Titulo de mi fichero - Cartelera del cine
HojaPapel = new File(carteleraCine); // Cojo el papel
Boli = new PrintWriter(new FileWriter(HojaPapel));

boolean masPeliculas = true;

/*****

* Ahora sí, vamos a comenzar a escribir dentro del fichero. *

*****/

// Creamos un bucle que nos vaya dando de alta en cartelera nuevas peliculas
do {

    System.out.println(".....");

    System.out.println("¿Nombre de la película? ");

    System.out.println(".....");

    Scanner escriboPeliculaEnFichero = new Scanner(System.in);

    // Almaceno el nombre del titulo de la pelicula en memoria

    String nombreP = escriboPeliculaEnFichero.nextLine();

    String PeliculaDeCine = "pelicula//"+nombreP+".txt";

    Boli.println(nombreP);

    contador++;

    System.out.printf("Pelicula nº: %d \n", contador);

    // Comprobacion de contador y peliculas que se deben dar de alta

    if (contador < nuevaP) {

        masPeliculas = false;

    } else {

        masPeliculas = true;

    }

} while (!masPeliculas);

// Terminamos la cartelera de peliculas del cine

Boli.close();

} // Cierre del metodo AltaPeliculas
```

## Dar de alta Críticos

Para llevar a cabo esta tarea que corresponde a los administradores en nuestro código es muy similar a dar de altas nuevas películas ya que para ello creamos un nuevo método en este caso nosotros llamaremos “críticos”.

Primeramente, ya que lo que deseamos es que el administrador ingrese los nombres de los críticos por teclado introducimos un Scanner, además estos críticos tienen que guardarse en un fichero.

La formación de un fichero es un proceso donde utilizamos la función File, la cual en este caso le pondremos el nombre de "Críticos" y le decimos cree un fichero tipo txt, posteriormente para poder escribir dentro de ese fichero exportamos las funciones PrintWriter y FileWriter, las cuales nos permitirán guardar los datos que introducimos.

Para aplicar este procedimiento utilizaremos un bucle do-while, el cual será condicionado por una variable booleana que creamos la cual llamaremos "más críticos", dentro del do preguntaremos al administrador el nombre de los críticos pero debido a que queremos que se cree un fichero por cada crítico, creamos dentro del do otro fichero que llamaremos "cada crítico" el cual creará un fichero nuevo cada vez que agreguemos un crítico y le pondrá el nombre de dicho crítico para así tener una carpeta de de todos los críticos y dentro de ella una carpeta de cada crítico.

Cuando White finaliza entonces se cierra el fichero y manda un mensaje agradeciendo su trabajo, con esto se finaliza el método y volvemos al menú.

## Código

```
public static int altaCriticos() throws IOException {

    Scanner altaC;

    System.out.println("");

    System.out.println("+-----+");

    System.out.println("| MENU DE ALTA DE CRITICOS. |");

    System.out.println("+-----+");

    System.out.println("");

    System.out.println("Introduzca cuantos criticos quiere dar de alta: \n");

    // Guardamos el numero de peliculas que se quieren dar de alta.

    altaC = new Scanner(System.in);

    // El Scanner lo pasamos a tipo Int

    int nuevoC = altaC.nextInt();

    /*****
    * Escritura *
    *****/
}
```

```

/* -----
 * Declaracion de variables.
----- */

int contador = 0;

File HojaPapel; // variable del programa.

PrintWriter Boli; // Como un SYSO pero para ficheros.

String fichaCriticos = "criticos/criticos.txt"; // Titulo de mi fichero - Ficha de los criticos del cine

HojaPapel = new File(fichaCriticos); // Cojo el papel

Boli = new PrintWriter(new FileWriter(HojaPapel));

boolean masCriticos = true;

/*****

 * Ahora sí, vamos a comenzar a escribir dentro del fichero. *

*****/

// Creamos un bucle que nos vaya dando de alta los criticos

do {

    System.out.println(".....");

    System.out.println("¿Nombre del critico? ");

    System.out.println(".....");

    Scanner escriboCriticoEnFichero = new Scanner(System.in);

    // Almaceno el nombre del titulo de la pelicula en memoria

    String nombreC = escriboCriticoEnFichero.nextLine();

    Boli.println(nombreC);

    contador++;

    System.out.printf("Critico nº: %d \n", contador);

    // Comprobacion de contador y criticos que se deben dar de alta

    if (contador < nuevoC) {

        masCriticos = false;

    } else {

        masCriticos = true;

    }

} while (!masCriticos);

// Terminamos la ficha de criticos de peliculas del cine

Boli.close();

// Retornamos el numero de criticos guardados

return nuevoC;

} // Cierre del metodo AltaCriticos

} // Class Administrador

```

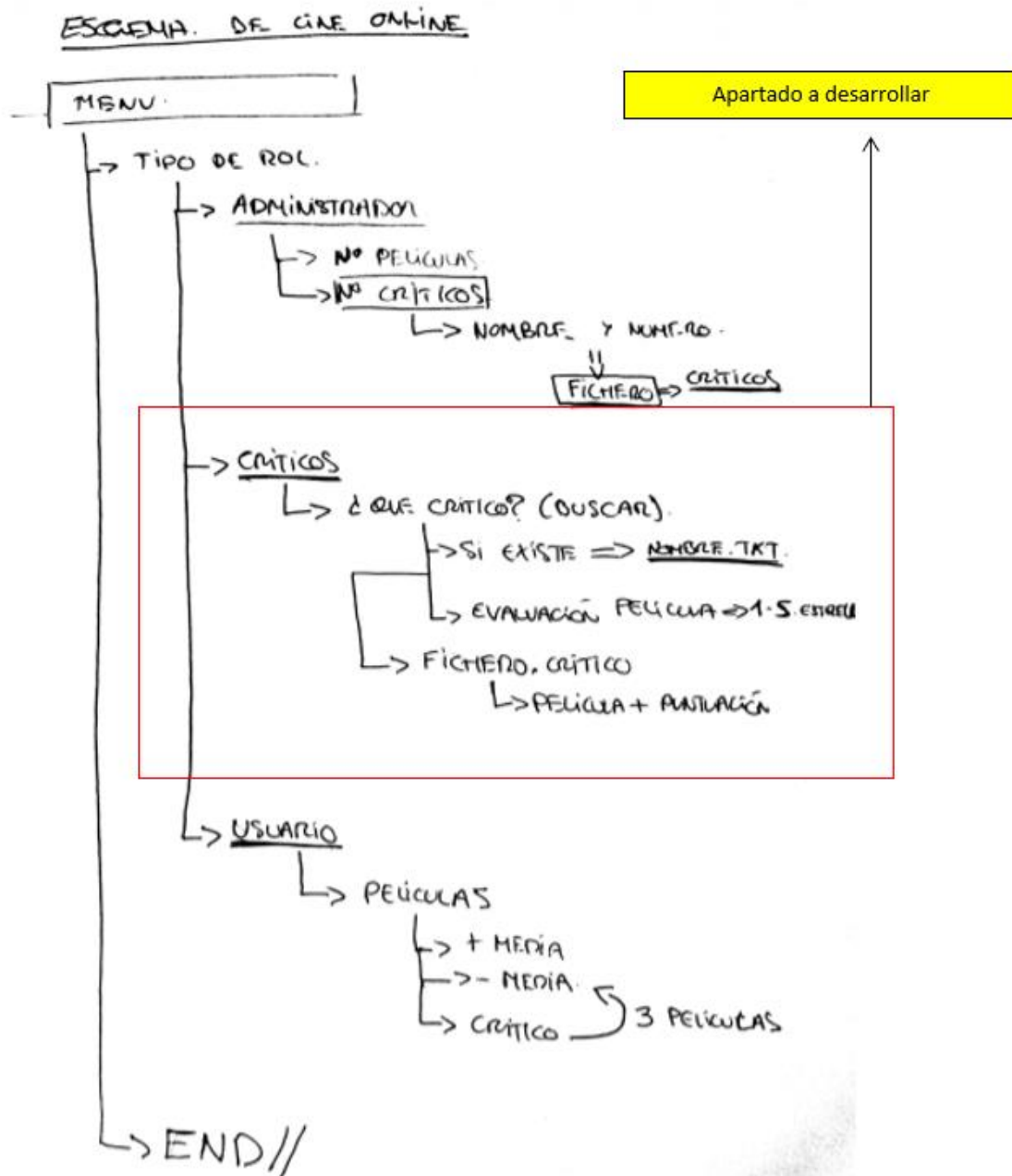
## CRITICOS.

### ESQUEMA "ALTAS CRITICOS"

#### Justificación del esquema

Antes de comenzar a construir el código, primero elaboraremos un rápido boceto de la idea del programa con el fin de optimizar la realización del código, para ellos primero se visualiza el programa total en conjunto y situar la parte del código que se va a desarrollar en este apartado

#### Esquematización del programa

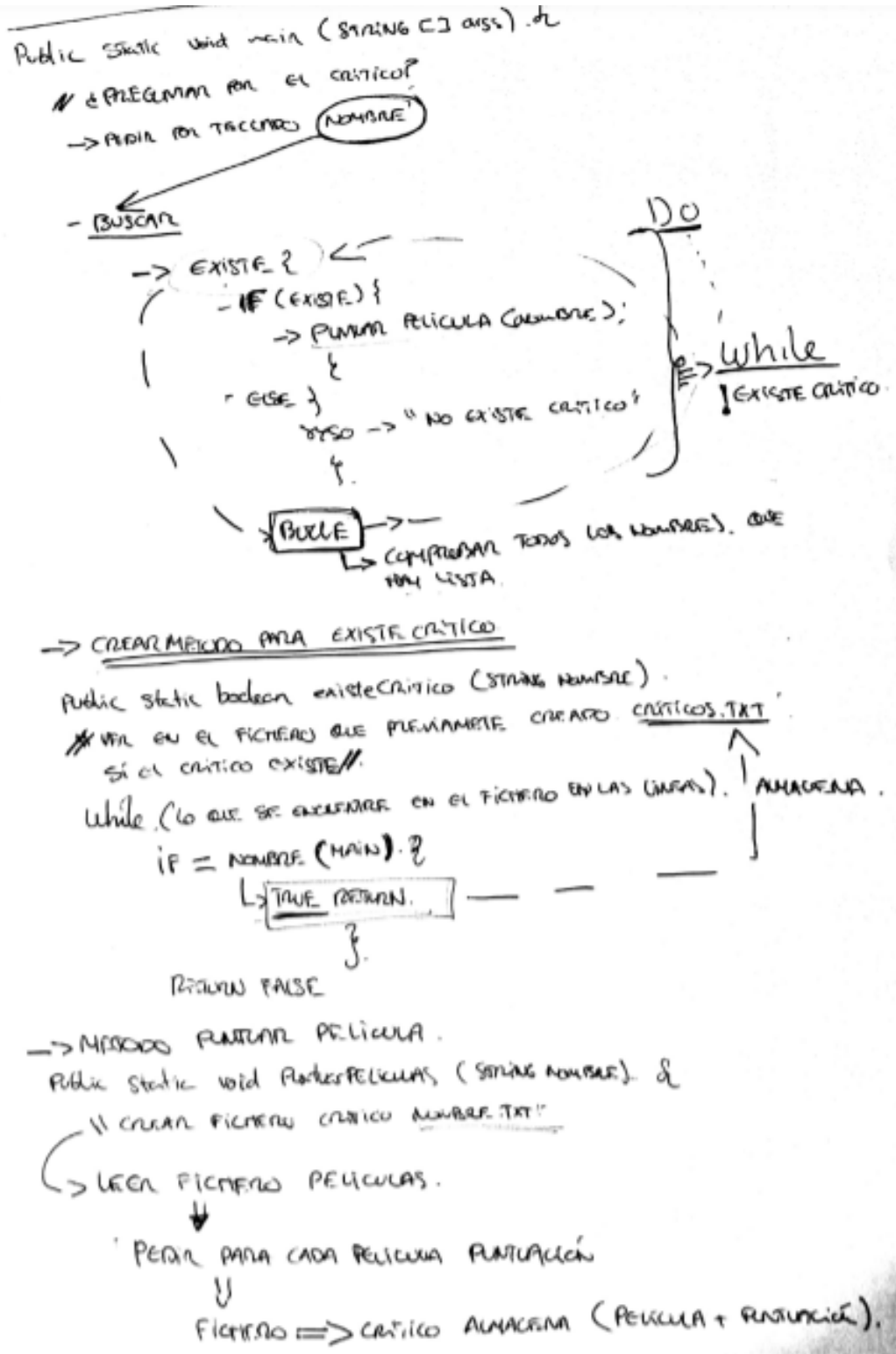




Una vez visualizado el programa en conjunto, comenzamos a la realización del apartado correspondiente a desarrollar ( zona limitada en roja en el esquema superior).

### Esquema apartado críticos

La composición del esquema se divide en 3 apartados , un apartado del programa que corresponde al main principal, y dos método que se tienen que realizar para dividir el programa y que resulte mas sencillo el objetivo del programa.



#### MAIN PRINCIPAL:

- Pedir por teclado el nombre del Critico( se puede introducir cualquier nombre) independientemente si se encuentra en el fichero de críticos.
- Buscar el nombre introducido en el fichero , y ver si existe o no llamando a un método que se creara
- Para buscar continuamente se tendrá que realizar un bucle en el cual comenzara una vez siempre hasta que encuentre el nombre del critico.
- Sí existe el critico, llamara a un método que será el de puntuarP;, sino el programa te sacara y lanzara un mensaje diciendo que el crítico no está dado de alta

#### METODO VERIFICAR SI EXISTE EL CRITICO

- Se tendrá que crear el fichero de críticos.txt ( será el fichero creado por el apartado de administrador
- Bucle while para buscar en el fichero y verificar que el nombre introducido en el main, es igual a nombre que se encuentra en el fichero de críticos entonces almacenara en un nuevo fichero el nombre.

#### METODO PUNTUACION DE PELICULAS

- Creacion de fichero correspondiente con el nombre del critico
- Leer fichero de películas
- Pedir al critico para cada película puntuación de 1 a 5 estrellas
- Almacenar cada película + puntuación en el fichero de cada critico .

### EXPLICACIÓN DEL CODIGO

#### modulacion del código

##### menu:

En el menú de criticos elaboramos un método llamado menú(); en el cual se describirá la parte correspondiente al menú de critico cuando en el main principal se quiera acceder como critico, para ello a través de de una serie de salidas de impresión indicamos que bienvenido al menú de critico y además de las opciones, donde nos encontramos dos casos: caso 0 (volvemos al menú principal del programa), caso 1:(dirigimos a dar de altas nuevas películas en el cine ) y en los demás casos que no estén contemplados el programa emitirá que la opción elegida es errónea, vuelva a intartarlo de nuevo

### **metodo buscar (c)**

Declaramos las variables Scanner=name , scanner lecturaFicha, crearemos un fichero llamado ficheroCriticos, también una variable de tipo String nombreDelFichero y a través de una variable booleana llamada existeCritico= false

Comenzamos la evaluación de las películas con varias salidas del sistema , en el cual se pedirá el nombre a través de un scanner , que será un nuevo scanner introducido al sistema.

Casterizamos el tipo String= name.nextLine.

Con un try/catch , realizamos la comprobación de que el fichero critcos.txt deba existir , por tanto nombredelfichero=criticos.txt

Seguidamente realizamos la búsqueda, apertura y lectura del fichero criticos.txt que debe existir, para ello el nombre del fichero FicheroCriticos=nuevo fichero (nombreDelFichero).

A su vez también realizamos la lecturaFicha=new Scanner (ficheroCriticos), es decir se realiza la lectura del fichero que corresponde al fichero FicheroCriticos. Para ello realizamos un bucle while( mientras todo lo que esacanea sea del fichero lecturaFicha) y aquí introducimos un condiciona en el cual comparamos mediante un tipo de string llamada nombreC.

Mediante el if buscamos al critico dentro del fichero, en el cual debe existir ese critico dentro del fichero Criticos.txt. Si encuentra el critico y coincide con el que se encuentra en el fichero, el programa lanzara un mensaje de “usuario encontrado” , además de llamada al método de puntuarP

Procedemos a cerrar los ficheros

### **metodo puntuarp.**

Declaramos las variables, mediante un scanner nora, scanner lecturaC y un printwriter llamado boli, también declaramos dos ficheros, un fichero llamado carteleraCine y fichaCritico, que corresponde con cada critico.

Variables int vuelta=0 la inicializamos a 0

Comenzamos la lectura del fichero lectura de cartelera.txt, la variable declarada anteriormente nombreCartelera= cartelera.txt.

También iniciamos la escritiura de puntuaciones correspondientes en los ficheros del crítico, atribuimos nombreCritico=nombre.txt.

A continuación realizamos la búsqueda y apertura y lectura de los ficheros, carteleraCine=new file(nombreCartelera), ficheroCritico=new file (nombreCritico)

Creamos una ficha para cada crítico.

Para poder leer los títulos de la cartelera correspondiente con el archivo CateleraCine , lo realizamos a través de un bucle while mientras lecturaC.hasNextLine me incrementas una vuelta.

Para cada película se pedirá una puntuación correspondiente de 1 a 5 estrellas , cada critico tendrá que atribuir a cada película encontrada en la carteleraCine.

El programa pedirá a cada critico que introduzca una nota de 1 a 5 estrellas, esto conlleva que la variable nota = new Scanner(System.in),entonces establecemos la relación entre la variable estrella y la variable nota.nextInt().

Entonces la nota hay que indicarle al programa que por un lado estrellas tiene que ser mayor que 0, es decir positivo y además que se encuentre en entre un rango entre 1 y 5.

Para ello establecemos un if en los cuales establecemos las opciones en las cuales el valor de la nota no se encuentra en el rango pedido y el programa mediante las salidas indicara que la película puntuada se encuentra fuera del rango, a continuación emitirá un error indicando que la nota introducida no es valida y finalmente indicara que si se quiere almacenar la nota debe volver a comenzar la evaluación.

En caso del else , nos encontramos en el estado en el cual la nota introducida es válida y por tanto se indicara un instruccion que será imprimir en el fichero el titulo de la película + las estrellas obtenidas.

Se cierra el fichero de lectura y se cierra el método de puntuar.

## CODIGO

```
public class Critico {

    public static void critico() throws IOException {

        menu();

    }

    public static void menu() throws IOException{

        // Menu de critico.

        System.out.println("+-----+");

        System.out.println(" | BIENVENIDO AL MENU DE CRITICO DE CINE. |");

        System.out.println("+-----+");

        System.out.println("Opciones:");

        System.out.println("");

        System.out.println(" 1) Evaluar peliculas en cartelera.");

        System.out.println("");

        System.out.println("-----");

        System.out.println(" | 0) Volver al menu principal. |");

        System.out.println("-----");

        System.out.println("Introduzca el numero de la opcion.");

        // Guardamos la opcion seleccionada.

        Scanner op = new Scanner(System.in);

        // Casteamos el Scanner a tipo Integer

        int opcion = op.nextInt();

        // Verificamos que la opcion escrita sea valida.
```

```

switch(opcion){
    case 1:
        // Dirigimos a dar de alta nuevas peliculas en el cine.
        buscarC();
        break;
    case 0:
        // Volvemos al menu principal del programa.
        Main.menu();
        break;
    default:
        System.out.println("\n ");
        System.out.println("Error en la opcion que ha escrito, vuelva a intentarlo de nuevo.");
        System.out.println("\n ");
        // Volvemos a mostrar el menu de Critico
        menu();
    } // Cierre del switch
} // Cierre del metodo menu()

private static void buscarC() throws IOException {
    /**
     * Declaracion de las variables.
     */
    Scanner name;
    Scanner lecturaFicha;
    File ficheroCriticos;
    String nombreDelFichero;
    boolean existeCritico = false;
    // Inicio de la evaluacion de peliculas x critico.
    System.out.println("");
    System.out.println("Iniciemos la evaluacion de peliculas.");
    System.out.println("");
    System.out.println("Introduzca su nombre: \n");
    name = new Scanner(System.in);
    // Casterizamos de Scanner a tipo String para poder compararlo despues.
    String you = name.nextLine();
    try {
        /**
         * Comprobación del nombre del critico en el fichero de 'criticos.txt'
         */

```

```

nombreDelFichero = "criticos/criticos.txt";

/**
 * Busqueda, apertura y lectura del fichero 'criticos.txt' - DEBE EXISTIR
 */
ficheroCriticos = new File(nombreDelFichero);
lecturaFicha = new Scanner(ficheroCriticos);
while (lecturaFicha.hasNextLine()){
    String nombreC = lecturaFicha.nextLine();

    // Buscamos al critico dentro del fichero - Debe estar 'CONTENIDO' dentro del txt de criticos.
    if (nombreC.contains(you)) {
        existeCritico = true;

        System.out.println("\t --- USUARIO ENCONTRADO! --- \n");

        puntuarP(you);
    }
}

lecturaFicha.close();
} catch(Exception e){
    System.out.println("\n\n\n");
    System.out.println(you);
    System.out.println("-----");
    System.out.println("| UY! Ah ocurrido un error inesperado! |");
    System.out.println("-----");
    System.out.println(e);
    System.out.println("-----");
} finally {
    System.out.println("Gracias " + you + "!");
    System.out.println("Ha terminado el programa de evaluación de peliculas.");

    // Volvemos al menu del principio.
    menu();

} // Cierre del try-catch-finally
} // Cierre del metodo puntuarPeliculas

private static void puntuarP(String yourName) throws FileNotFoundException {
    /**
     * Declaracion de las variables.
     */
    Scanner nota;
    Scanner lecturaC;

```

```

PrintWriter Boli;

File carteleraCine;

File fichaCritico;


String nombreCartelera;

String nombreCritico;

int[] mediaCritico=null;

int[] mediaUnidad=null;


int vuelta=0;

int estrellas;

try{

    // Iniciamos la lectura de 'cartelera.txt'
    nombreCartelera = "cartelera//cartelera.txt";

    // Iniciamos la escritura de puntuaciones en los ficheros del critico
    nombreCritico = "fichas//"+yourName+".txt";

    /**
     * Busqueda, apertura y lectura de los ficheros
     */

    carteleraCine= new File(nombreCartelera);

    fichaCritico = new File(nombreCritico);

    // Creamos una ficha para el critico.
    Boli = new PrintWriter(new FileWriter(fichaCritico));

    // Lectura de los titulos en cartelera.
    lecturaC = new Scanner(carteleraCine);

    while (lecturaC.hasNextLine()){

        ++vuelta;

        String tituloPeli = lecturaC.nextLine();

        System.out.println("-----");

        System.out.printf("-> Titulo: %s \n",tituloPeli);

        System.out.println("-----");

        System.out.printf("----> Escribe las puntuacion para %s (1-5): ",tituloPeli);

        System.out.println("");

        nota = new Scanner(System.in);

        estrellas = nota.nextInt();

        if ((estrellas < 1) || (estrellas > 5)) {

```

```

        System.out.println("--> FUERA DEL RANGO DE PUNTACION" + tituloPeli + " - " + estrellas );

        System.out.println("");

        System.out.println("!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!");

        System.out.println("! \t ERROR DE PUNTUCION");

        System.out.println("! \t "+ yourName+" - Introduciste una nota no valida.");

        System.out.println("! \t Las estrellas no se guardaran para este titulo.");

        System.out.println("! \t ");

        System.out.println("! \t Si quieres puntuarla, deberas volver a iniciar la evaluacion.");

        System.out.println("! \t");

        System.out.println("!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!");

        System.out.println("");

    } else {

        Boli.println(tituloPeli + " - " + estrellas);

    }

}

lecturaC.close();

Boli.close();

} catch(Exception e){

    System.out.println("-----");

    System.out.println("| UY! Ah ocurrido un error inesperado! |");

    System.out.printf( " | %s debes contactar con el admin.   |\n",yourName);

    System.out.println("-----");

    System.out.println(e);

    System.out.println("-----");

    System.out.println("");

} // Cierre del try-catch

} // Cierre del metodo puntuarP

} // Cierre de la clase Critico

```



## USUARIOS

### Proposito

En el caso de ser un usuario, el sistema debe contener ciertas tareas, teniendo la necesidad de crear un menú que conste con las opciones solicitadas.

1. Mostrar películas con menor media
2. Mostrar películas con mayor media
3. Mostrar el critico que menor puntuación media ha dado a todas las películas
4. Mostrar las películas con mas de tres estrellas de media
5. Finalizar el programa.

### Realización

Para realizar esta tarea, se realizó la creación de un menú usando *println*. Y para controlar la opción seleccionada, se utiliza una estructura de switch-case con las 5 opciones vistas anteriormente, cada *case* llama diferentes métodos dependiendo de la opción seleccionada.

Los métodos que se usan son:

menorMedia() para la opción 1.

mayorMedia() para la opción 2.

criticoExigente() para la opción 3.

masDeTresE() para la opción 4.

Para la opción 5, la salida, se envia un mensaje por pantalla despidiéndose

Main.menu() para la opción 0, retorna al menú principal de la clase main.

### menorMedia() , mayorMedia() y masDeTresE()

Llama otro método, llamado **notaMedia()**, que se ha programado en una clase java diferente, CalculosMenorYMayorMedia.java, donde se envia como argumento la opción llamada desde el menú de usuario, guardado en la variable 'op'.

Para realizar esta tarea, necesitamos saber:

- Los críticos
- El nº de películas
- Notas de c/u de las películas

Para desarrollar estos métodos:

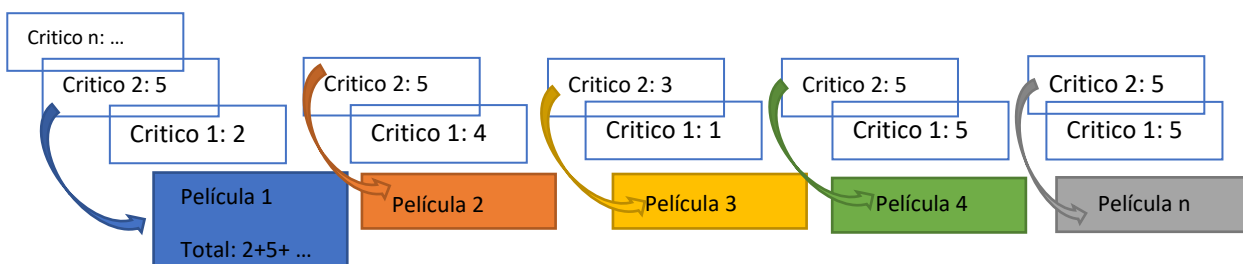
- Leeremos la ficha del critico
- Obtendremos la nota guardada en la ficha del critico
- Generaremos un fichero con todas las notas encontradas para una película, por lo que deberá coincidir el titulo.
- Calcularemos la media y usaremos cada método que se seleccione.

Para ello, se programo un método `totalPelículas()` donde nos devuelve el numero total de películas

También, tenemos un método `totalCriticos()` donde se devuelve el numero total de críticos dados de alta.

El método **notaMedia**, necesita como argumento la opción seleccionada, que se envia desde la clase Usuarios como se comento en su sección.

Se utiliza un control de excepciones mediante un try-catch, donde leeremos los críticos dados de alta y en un array igual de grande que el numero de películas, almacenaremos las calificaciones que se leen en la ficha del critico, y al cambiar de critico, se hará que se coincida la posición de la calificación y película con la evaluación del critico anterior, si hay. Es decir, buscar la siguiente estructura:



Una vez terminados los cálculos, si buscamos la nota mas baja, usaremos la estructura if-else con la `notaMinima()`, si buscamos la nota mas alta, usaremos la estructura if-else con la `notaMaxima()`.

Si lo que se solicitó es las películas de mas de tres estrellas, en la estructura del switch-case, opción 3, contenemos un if que ejecuta dicha condición y devuelve la nota media mayor de 3.

Los métodos `mediaMasBaja`, `mediaMasAlta`, `masDeTres` y `votos()`, todas necesitan como argumento la nota media calculada a lo largo del método y el titulo de la película, a excepción de este último.

Pues en estos, imprimiremos la información solicitada por el usuario, la media de las películas mas baja, la mas alta, mas de tres estrellas.

Para `votos()`, solo necesitamos el contador de líneas totales del fichero, con esto sabemos, como su nombre propiamente indica, el numero total de votos que ha tenido una película. Este método solo va a ser usado en el menú de críticos, para saber la película ganadora.

### `criticoExigente()`

De igual manera, llama el método **notaMedia()**, almacenado en la clase `Calculos.java` y muestra el nombre del critico que menor nota media da a las películas del cine

La clase cálculos almacena el método `total Películas`, que calcula en numero total de películas dadas de alta en cartelera por el administrador.

La clase cálculos almacena el método nota Media, donde se nos devolverá el nombre del critico que buscamos. Usaremos un try-catch-exception para controlar los errores.

Para realizar la búsqueda del critico, realizamos la lectura de fichero críticos, para acceder a la carpeta de fichas y saber que críticas debemos leer.

Donde volveremos a leer ficheros, pero como se comento anteriormente, solo de los críticos dados de alta. A medida que leemos la ficha, realizamos un filtro para separar los títulos de las películas de las notas medias. Y usamos un bucle for para ir calculando la media que se almacena dentro de la variable MediaTotal y si es la media mas baja, ese critico es el que necesitamos y retornamos.

## Código.

### Clase Usuarios.java

```
/**
 * • En caso de que seamos usuarios, nos mostrará un menú que conste de las siguientes
 * opciones:
 *   o Mostrar película con menor media
 *   o Mostrar película con mayor media
 *   o Mostrar el crítico que menor puntuación media ha dado a todas las películas (Si hay varias,
 *     mostrar solo una de ellas)
 *   o Mostrar las películas con más de 3 de media
 *   o Finalizar el programa
 */
package net.humancomputing.cineonline;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;

/**
 * @version 0.5
 * @autores Efrain Enrique Ramirez Mojica, Kevin Alfonso Gomez Sandoval & Paulino Esteban Bermudez Rodriguez.
 */
public class Usuarios {

    public static void usuarios() throws IOException {

        menu();

    }
}
```

```

public static void menu() throws IOException {

    // Menu de critico.

    String usuario = System.getProperty("user.name");

    System.out.println("+-----+");

    System.out.println(" | BIENVENIDO AL MENU DE USUARIOS DE CINE.   |");

    System.out.println("+-----+");

    System.out.printf(" %s ",usuario);

    System.out.println("Opciones:");

    System.out.println("");

    System.out.println(" 1) Mostrar peliculas con menor media.");

    System.out.println(" 2) Mostrar peliculas con mayor media.");

    System.out.println(" 3) Mostrar critico que menor puntuacion media da a las peliculas.");

    System.out.println(" 4) Mostrar peliculas con +3 estrellas de media.");

    System.out.println("-----");

    System.out.println(" 5) Finalizar el programa.");

    System.out.println("-----");

    System.out.println("");

    System.out.println("-----");

    System.out.println(" | 0) Volver al menu principal.   |");

    System.out.println("-----");

    System.out.println("Introduzca el numero de la opcion.");

    // Guardamos la opcion seleccionada.

    Scanner op = new Scanner(System.in);

    // Casteamos el Scanner a tipo Integer

    int opcion = op.nextInt();

    // Verificamos que la opcion escrita sea valida.

    switch(opcion){

        case 1:

            // Mostramos la menor media en las peliculas

            menorMedia();

            menu();

            break;

        case 2:

            // Mostramos la mayor media en las peliculas

            mayorMedia();

            menu();

            break;

        case 3:
    
```

[illegible]

```

        String criticoExigente = Calculos.notaMedia();

        System.out.println("");

        System.out.printf("El critico más exigente es: %s",criticoExigente);

        System.out.println("");

    }

    private static void masDeTresE() {

        int op=3;

        CalculosMenorYMayorMedia.notaMedia(op);

    }

    public static void salida() {

        System.out.println("+-----+");

        System.out.println(" | Fin del programa.          |");

        System.out.println(" | Gracias por evaluar CINE ONLINE MEK.      |");

        System.out.println(" |                                     |");

        System.out.println(" | @universidad: UCLM                      |");

        System.out.println(" | @subject: Fundamentos de programacion I.  |");

        System.out.println(" | @version: 22.01.19.beta                  |");

        System.out.println(" | @autores: Paulino Esteban Bermudez R.    |");

        System.out.println(" | Kevin Alfonso Gomez S.                  |");

        System.out.println(" |                                     |");

        System.out.println("+-----+");

    }

} // Cierre de la clase Usuarios

```

## Clase Calculos.java

```

/*
 * Clase que realiza el calculo de la media de las peliculas de un usuario/critico y muestra el más exigente.
 */

package net.humancomputing.cineonline;

import java.io.*;

import java.util.Scanner;

/**
 * @version 0.3
 * @autores: Kevin Alfonso Gomez Sandoval & Paulino Esteban Bermúdez Rodríguez | humancomputing.net
 */

```

```

public class Calculos {

    /*****
    * Datos necesario:
    *
    * - Los criticos
    *
    * - Nº de peliculas
    *
    * - Notas de c/u de las peliculas
    *
    *
    * Para desarrollar estos metodos:
    *
    * + Leere la ficha del critico
    *
    * + Obtendre el ultimo caracter de la linea leida - la nota
    *
    * + Asociare el titulo de cartelera al caracter obtenido
    *
    * + Calculamos su media y usamos en c/u de los metodos.
    *
    *****/

    // Obtenemos el numero de peliculas que tiene el cine.

    public static int totalPeliculas() throws FileNotFoundException {

        Scanner lecturaCartelera;

        File carteleraPublicada;

        String nombreDelFichero;

        String titulo;

        int total=0;

        /*
        * Peticion de datos
        */

        nombreDelFichero="cartelera/cartelera.txt";

        /**
        * Lectura del fichero
        */

        carteleraPublicada = new File(nombreDelFichero);
        lecturaCartelera = new Scanner(carteleraPublicada);

        while(lecturaCartelera.hasNextLine()){

            titulo = lecturaCartelera.nextLine();

            total++;

        }

        return total;

    }

    // Obtenemos la calificacion de cada pelicula

    public static String notaMedia() {

        String c="";
    
```

```

try {
    /**
     * 1) Leo los criticos dados de alta
     * 2) Leer cada fichero de cada critico
     * 3) Calcular la media
     */
    /*******
     * LECTURA DE FICHERO CRITICOS - datos de alta por el admin
     *****/

    // Variables necesarias
    Scanner leerAltas;
    File fichaCritico;
    String nombreDelFichero;
    int puntuacionPelicula=0;
    int contadorDeLinea=0; // Sirve para saber cuantas lineas tiene el fichero
    float media = 0;
    float mediaTotal=0;
    float menorMediaTotal = 0;
    String nombreDelCritico = "";
    /**
     * Peticion de datos
     */
    nombreDelFichero = "criticos/criticos.txt"; // Buscamos el txt en la carpeta
    fichaCritico = new File(nombreDelFichero); // Empezamos a leer el fichero con el buffer del sistema
    leerAltas = new Scanner(fichaCritico); // Iniciamos la lectura
    /**
     * LECTURA DE FICHEROS
     */
    while(leerAltas.hasNextLine()){
        nombreDelCritico = leerAltas.nextLine(); // Guardamos el nombre del critico
        /*******
         * LECTURA DE LAS FICHAS CON LAS PUNTUACIONES DE LOS CRITICOS
         *****/

        Scanner leerFicha;
        File evaluacionCritica;
        String evaluacionPeli;
        /**
         * Peticion de datos

```



```

*/
evaluacionPeli = "fichas/"+nombreDelCritico+".txt";
System.out.println("Ficha:"+evaluacionPeli);
evaluacionCritica = new File(evaluacionPeli);
leerFicha = new Scanner(evaluacionCritica);
while(leerFicha.hasNextLine()){
    String linea = leerFicha.nextLine(); // Leemos la linea de la evaluacion critica
    String divisionLinea = linea;
    String[] parts = divisionLinea.split("-");
    String tituloPelicula = parts[0];
    String puntuacionP = parts[1];
    // Casteamos de String a Integer y eliminamos el espacio
    puntuacionPelicula = Integer.valueOf(puntuacionP.trim());
    // Sacar media del critico según el numero total de peliculas publicadas.
    int totalPelisEnCartelera = totalPeliculas();
    for (int i = 0; i < totalPelisEnCartelera; i++) {
        media =puntuacionPelicula;
    }
    mediaTotal+= media;
    contadorDeLinea++;
}
mediaTotal = mediaTotal/contadorDeLinea;
// System.out.println(nombreDelCritico+" su media es de :"+mediaTotal);
// Guardamos la nota del critico
if (mediaTotal > menorMediaTotal) {
    menorMediaTotal = mediaTotal;
}
// Devolvemos el nombre del critico
c = nombreDelCritico;
// Una vez hecha la media, reiniciamos las variables iniciales.
mediaTotal=0;
media=0;
contadorDeLinea=0;
}
} catch (FileNotFoundException e1){
    System.out.println("");
    System.out.println("*****");
    System.out.println(" Fichero no encontrado...Verifique sus datos y vuelva a ejecutarlo con otros datos diferentes.");
}

```

```

        System.out.println(" ");
        System.out.println(" Error ocurrido: " + e1);
        System.out.println(" ");
    } catch (Exception e2){
        System.out.println("");
        System.out.println("Error:");
        System.out.println(e2);
    }

    return c; // RESULTADO DEL METODO - NOMBRE DEL CRITICO MAS EXIGENTE.
} // Cierre del metodo notaMedia()
} // Cierre de la clase calculos

```

## Clase CalculosMenorYMayorMedia.java

```

/**
 * Clase que contiene los metodos para calcular la menor y mayor media de las peliculas.
 */
package net.humancomputing.cineonline;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Scanner;

/**
 * @version 0.3
 * @authors: Kevin Alfonso Gomez Sandoval & Paulino Esteban Bermúdez Rodríguez | humancomputing.net
 */
public class CalculosMenorYMayorMedia {

    /**
     * Datos necesario:
     *
     * - Los criticos
     *
     * - Nº de peliculas
     *
     * - Notas de c/u de las peliculas
     *
     *
     * Para desarrollar estos metodos:
     *
     * + Leere la ficha del critico
     *
     * + Obtendre el ultimo caracter de la linea leida - la nota
     *
     * + Asociare el titulo de cartelera al caracter obtenido
     *
     * + Calculamos su media y usamos en c/u de los metodos.
     */
}

```

```
// Obtenemos el numero de peliculas que tiene el cine.

public static int totalPeliculas() throws FileNotFoundException {

    Scanner lecturaCartelera;

    File carteleraPublicada;

    String nombreDelFichero;

    String titulo;

    int total=0;

    /*
     * Peticion de datos
     */

    nombreDelFichero="cartelera/cartelera.txt";

    /**
     * Lectura del fichero
     */

    carteleraPublicada = new File(nombreDelFichero);
    lecturaCartelera = new Scanner(carteleraPublicada);

    while(lecturaCartelera.hasNextLine()){

        titulo = lecturaCartelera.nextLine();

        total++;

    }

    return total;

}

// Obtenemos el numero de peliculas que tiene el cine.

public static int totalCriticos() throws FileNotFoundException {

    Scanner lecturaCartelera;

    File carteleraPublicada;

    String nombreDelFichero;

    String titulo;

    int total2=0;

    /*
     * Peticion de datos
     */

    nombreDelFichero="criticos/criticos.txt";

    /**
     * Lectura del fichero
     */

    carteleraPublicada = new File(nombreDelFichero);
    lecturaCartelera = new Scanner(carteleraPublicada);
```

```

while(lecturaCartelera.hasNextLine()){

    titulo = lecturaCartelera.nextLine();

    total2++;

}

return total2;

}

// Obtenemos la calificacion de cada pelicula

public static void notaMedia(int opcionSeleccionada) {

    float mediaTotal=0;

    try {

        /**

        * 1) Leo los criticos dados de alta

        * 2) Leer cada fichero de cada critico

        * 3) Genero un array igual de grande que el numero de peliculas guardadas

        * 4) Hacemos un sumatorio para cada pelicula :=> posicion del array[i] + puntos

        */

        /*****

        * LECTURA DE FICHERO CRITICOS - datos de alta por el admin

        *****/

        // Variables necesarias

        Scanner leerCartelera;

        File fichaCartelera;

        String nombreDelFichero;

        int contadorDeLinea=0; // Sirve para saber cuantas lineas tiene el fichero

        float notaMinima=5;

        float notaMaxima=0;

        /**

        * Peticion de datos

        */

        nombreDelFichero = "cartelera/cartelera.txt"; // Buscamos el txt en la carpeta

        fichaCartelera = new File(nombreDelFichero);

        leerCartelera = new Scanner(fichaCartelera) ;

        while(leerCartelera.hasNextLine()){

            mediaTotal=0;

            contadorDeLinea=0;

            String tituloPeli = leerCartelera.nextLine();

            /*****

            * Leemos las notas que tiene almacenado el fichero de la pelicula *

```

```

*****/

Scanner readP;

File filePeli;

String namePeli;

/**
 * Peticion de datos
 */

namePeli = tituloPeli;

filePeli = new File(namePeli);

readP = new Scanner(filePeli);

while(readP.hasNextLine()){

    contadorDeLinea++;

    String p = readP.nextLine();

    int puntuacion = Integer.parseInt(p);

    mediaTotal += puntuacion;

}

/**
 * Calculamos la media total de la pelicula una vez terminada su lectura.
 */

    System.out.println(mediaTotal + " puntos para " + tituloPeli);

mediaTotal /= contadorDeLinea;

float array[] = new float[contadorDeLinea];

float auxiliar[] =new float[contadorDeLinea];

for (int i = 0; i < array.length; i++) {

    auxiliar[i]=mediaTotal;

    /*****

    *      MENOR      *

    *****/

    if (auxiliar[i] <= 1) {

        notaMinima=auxiliar[i];

    } else if (auxiliar[i] <= 2) {

        notaMinima=auxiliar[i];

    } else if (auxiliar[i] <= 3) {

        notaMinima=auxiliar[i];

    } else if (auxiliar[i] <= 4) {

        notaMinima=auxiliar[i];

    }

```

```

    } else if (auxiliar[i] <= 5) {

        notaMinima=auxiliar[i];

    } else {

        System.out.println("¿¿¿¿ERROR????");

        System.out.println("Algo raro has hecho! Miratelo porque lo que ha pasado era casi imposible,");

        System.out.println("pero a pasado...");

    }

    /*****

    *      MAYOR      *

    *****/

    if (auxiliar[i] >= 1) {

        notaMaxima=auxiliar[i];

    } else if (auxiliar[i] >= 2) {

        notaMaxima=auxiliar[i];

    } else if (auxiliar[i] >= 3) {

        notaMaxima=auxiliar[i];

    } else if (auxiliar[i] >= 4) {

        notaMaxima=auxiliar[i];

    } else if (auxiliar[i] >= 5) {

        notaMaxima=auxiliar[i];

    }

    else {

        System.out.println("¿¿¿¿ERROR????");

        System.out.println("Algo raro has hecho! Miratelo porque lo que ha pasado era casi imposible,");

        System.out.println("pero a pasado...");

    }

}

switch(opcionSeleccionada){

    case 1:

        // Mas baja

        mediaMasBaja(notaMinima, tituloPeli);

        break;

    case 2:

        // Mas alta

        mediaMasAlta(notaMaxima, tituloPeli);

        break;

    case 3:

        // +3

```

```

        if (mediaTotal > 3.0) {
            masDeTres(mediaTotal, tituloPeli);
        }

        break;
    case 4:
        votos(contadorDeLinea);
        mediaMasAlta(notaMaxima, tituloPeli);
        break;
    default:
        System.out.println("Error sin sentido.");
    }
} // Lectura de cartelera
} catch (FileNotFoundException e1){
    System.out.println("");
    System.out.println(" Fichero no encontrado...Verifique sus datos y vuelva a ejecutarlo con otros datos diferentes.");
    System.out.println(" Error ocurrido: " + e1);
} catch (Exception e2){
    System.out.println("");
    System.out.println("EEEEEEYYYYY!!! Hubo un error:");
    System.out.println(e2);
}
} // Cierre del metodo notaMedia()
// Sobrecargamos los metodos de calculo de medias
public static void mediaMasAlta(float mediaTotal, String titulo) {
    System.out.println("La pelicula " + titulo + " con la puntuacion mas alta es: " + mediaTotal);
}
public static void mediaMasBaja(float mediaTotal, String titulo ) {
    System.out.println("La pelicula " + titulo + " con la puntuacion mas baja es: " + mediaTotal);
}
public static void masDeTres(float mediaTotal, String titulo ) {
    System.out.println("");
    System.out.printf("La pelicula %s tiene la puntuacion: %f ",titulo,mediaTotal);
    System.out.println("");
}
public static void votos(int votos){
    System.out.println("Total de votos:" + votos);
}
} // Cierre de la clase calculos

```

## JURADO

Para poder realizar esta parte del programa vamos a necesitar, entrar en el menú de opciones que pregunte que rol tenemos, una vez dentro, ejecutamos la tarea de buscar la película ganadora.

- Jurado.

### Realización del programa.

Dentro de esta clase creamos un método jurado, el cual posteriormente será llamado por el método main para así mostrar lo que se encuentra en dicho método, empezamos declarando una variable Int que la llamaremos opción, aquí elaboraremos un menú de opciones para el usuario que dependerá de lo que desee realizar en el cine online.

Posteriormente el programa nos dirá que introduzcamos por medio de un Scanner la opción que deseamos realizar, entre las cuales tenemos los roles de administrador, crítico de cine, usuario, jurado y otra opción por si desea volver al menú principal.

La creación de este menú comenzara con el Scanner que solicita que rol desea realizar, creamos un Switch-Case para poder determinar al usuario la opción donde desea ir por ello creamos 2 casos diferentes los cuales son:

Case 1: Dirigimos a dar de alta nuevas películas en el cine.

Case 2: Volvemos al menú principal del programa.

Posteriormente hacemos un default para así poder controlar que el usuario ingrese una de las opciones disponibles ya que si no lo hace le aparecerá un mensaje por medio de un System.out.println en el que se le indicara por medio de un mensaje "Error en la opción que ha escrito, vuelva a intentarlo de nuevo", después regresara al menú principal para volver a escoger alguna opción.

### Programa realizado.

```
package net.humancomputing.cineonline;

import java.io.IOException;

import java.util.Scanner;

/**
 * @version 0.1
 * @author Paulino Esteban Bermúdez Rodríguez | humancomputing.net
 */

public class Jurado {

    public static void jurado() throws IOException {

        menu();

    }

    public static void menu() throws IOException {

        // Menu de jurado

        System.out.println("-----+");
```



```

System.out.println("| BIENVENIDO AL MENU DE JURADO. |");
System.out.println("+-----+");
System.out.println("Opciones:");
System.out.println("");
System.out.println(" 1) Pelicula ganadora de la cartelera.");
System.out.println("| 0) Volver al menu principal. |");
System.out.println("Introduzca el numero de la opcion.");
Scanner op = new Scanner(System.in);

// Casteamos el Scanner a tipo Integer
int opcion = op.nextInt();

// Verificamos que la opcion escrita sea valida.
switch(opcion){
case 1:
    // Dirigimos a dar de alta nuevas peliculas en el cine.
    ganadora();
    menu();
    break;
case 0:
    // Volvemos al menu principal del programa.
    Main.menu();
    break;
default:
    System.out.println("Error en la opción que ha escrito, vuelva a intentarlo de nuevo.");
    // Volvemos a mostrar el menu de Administrador
    menu();
} // Cierre del switch
} // Cierre del menu()

public static void ganadora() {
    CalculosMenorYMayorMedia.notaMedia(4);
}
}

```

## Conclusiones y problemas encontrados a la hora de realizar el proyecto.

El sistema, en líneas generales realiza las tareas solicitadas por el enunciado.

Sin embargo, a medida que íbamos creando nuestros códigos de manera independiente, al arbitrariedad y cierta falta de comunicación entre los integrantes, generó que mucho del código realizado se duplicara a lo largo del programa, por ello se decidió comenzar a usar GitHub como herramienta de control de versiones, así todos los integrantes tendrían la última versión publicada en el repositorio.

Aunque se realizaron diversas pruebas, la falta de tiempo para enfocar de otra forma el programa, que en su momento ya estaba bastante avanzado, impedía reestructurar la resolución de ciertos problemas, sobre todo para la clase usuarios.java, para resolverlos de una forma más sencilla y eficiente.

También, para guardar un orden y homogeneidad del sistema, todos los ficheros que se van generando en las diferentes ejecuciones del programa, se decidió que se almacenaran en diferentes carpetas accesibles, porque inicialmente, si realizábamos demasiadas pruebas consecutivas, se generaban muchos ficheros, llegando a ser incontrolable.

Por último, se decidió que cada integrante del grupo de trabajo, realizase una explicación de la parte asignada. Como cada uno de ellos tiene diferentes equipos para realizar la grabación, no existió una homogeneidad 100%, bien fuera por no tener micrófono y depender del que trae el portátil, no tener una webcam, por ejemplo para ciertos portátiles que no traen según el modelo. Aun así, por lo menos el tema del volumen del video se intentó armonizar usando software de edición, en este caso Filmora.

## Mejoras del proyecto para futuras actualizaciones.

Las mejoras podrían venir para la clase usuarios, que afectaría a su vez las clases programadas 'de las que depende', Calculos y CalculoMenorYMayorMedia, para evitar la creación de demasiados ficheros en el sistema y solucionarlo creando listas, almacenadas en arrays, porque si es un sistema con pocas películas o películas fijas siempre, caso improbable, no afectaría demasiado esta resolución, pero como en líneas generales se va cambiando de cartelera, se generarían demasiados ficheros, que con el tiempo serían incontrolables.

## Bibliografía.

- Formación recibida por el profesorado de la UCLM para la asignatura de fundamentos de programación I.
- <https://es.stackoverflow.com/>
- <https://youtube.com/>
- <https://docs.oracle.com/javase/7/docs/api/>
- <https://www.delftstack.com/es/howto/java/java-compare-integer/>
- <https://javadesdecero.es/>

## Software usado para el desarrollo del programa

- <https://meet.google.com/>
- <https://github.com/chunche95/CineOnline>
- <https://desktop.github.com/>
- <https://devhubapp.com/>
- [https://filmora.wondershare.net/es/filmora9-video-editor-es.html?gclid=CjwKCAiA866PBhAYEiwANKIneMAQuDhj4j52M7iNMgfTvE7Sa6u2PAPCj-Jo\\_5IFD1dyBQubz3kBPRoCvEcQAvD\\_BwE](https://filmora.wondershare.net/es/filmora9-video-editor-es.html?gclid=CjwKCAiA866PBhAYEiwANKIneMAQuDhj4j52M7iNMgfTvE7Sa6u2PAPCj-Jo_5IFD1dyBQubz3kBPRoCvEcQAvD_BwE)