# Examen práctico 60% de la nota total

Netflix lo ha contratado para que desarrolle el módulo de ordenamiento de su plataforma,

para poder recibir la aprobación, usted deberá hacer un prototipo de cómo funcionará el

sistema de ordenamiento.

Se debe mostrar en el lienzo de fondo negro los ítems de netflix, recuerde que hay tipo

serie y tipo movie. Cada ítem tendrá nombre,año de estreno, rating (mostrar con %) y si es serie o movie y finalmente tendrá una imagen la cual es el mismo nombre escrito en el txt, pero se encuentra con minúscula y sin espacios (revisar imágenes) para cargar las imágenes se debe utilizar una repetitiva y el api de String.

1. Cargar la información del txt y pintar en el lienzo según las indicaciones dadas.
2. Se desea que el orden natural de los ítem de netflix sea el rating.
3. El programa debe guardar un archivo .txt con la lista de items ordenados por cada atributo (un archivo con los ítems ordenados por nombre, otro archivo ordenados por rating y así por cada atributo). El nombre del ítem debe quedar en mayúscula.
4. El programa debe tener una interfaz gráfica la cual muestre la información de los txt, La cual debe ordenarse según la interacción que ustedes escojan (presionando teclas, botones, etc.) es libre de cada desarrollador, sin embargo, debe tener instrucciones de cómo organizar la lista por cada atributo.

**Bonus (0.5 en la parte práctica)**

5. Al seleccionar un ítem de la lista y arrastrarlo hacia otro, estos dos ítems

cambiarán respectivamente de posición.

**Rúbrica**

1. **Análisis y diseño de software (30%)**

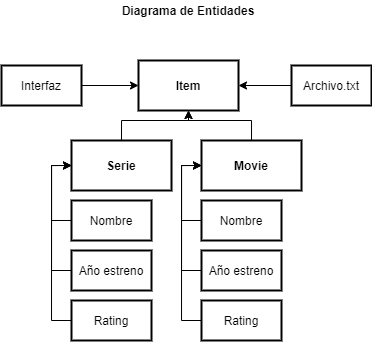
a. Se sugiere sacar requerimientos funcionales (no tiene nota)

b. Diagrama de clases (tener en cuenta arquitectura MVC).

1. **Implementación del código (70%)**
2. Correcta implementación del diagrama de clases.10%
3. Correcto uso de herencia 10%
4. Correcta separación de paquetes y clases según su responsabilidad. 10%
5. Correcta utilización de la lista adecuada según los requerimientos. 10%
6. Correcta carga de información 10%
7. Correcto guardado de información 20%
8. Correcto manejo de cadenas de texto 10%
9. Correcta implementación del ordenamiento de la lista de mascotas de manera natural y parcial. 20%
   * 1. Por nombre
     2. Rating
     3. Año
     4. Tipo de ítem

**Entidades:**

1. **Items.**
   1. Serie
      1. Nombre.
      2. Año de estreno.
      3. Rating.
   2. Movie
      1. Nombre.
      2. Año de estreno.
      3. Rating.
2. **Archivo.TXT.**
3. **Interfaz.**

****

**RF1:** Cargar archivo.txt con los items

|  |  |
| --- | --- |
| RF1 | |
| **Descripción:** | Se debe cargar un archivo.txt que contiene el nombre de los ítems. |
| **Entradas:** | Serie y movie:  Nombre, año de estreno y Rating |
| **Salidas:** | El archivo debe de estar cargado. |
| **PreCondiciones:** | El archivo debe de existir. |
| **Post-Condiciones:** | El archivo.txt debe poder ser llamado y editado. |

**RF2:** Se debe crear una clase item

|  |  |
| --- | --- |
| RF2 | |
| **Descripción:** | Se debe crear una clase ítem que contenga y cargue los ítem del archivo.txt |
| **Entradas:** | Nombre de los ítems e información que contiene el archivo.txt. |
| **Salidas:** | Clases ítem con el archivo.txt cargado en ella. |
| **PreCondiciones:** | El archivo.txt debe estar creado y cargado. |
| **Post-Condiciones:** | Se debe de leer el archivo.txt y poder usar. |

**RF3:** Ordenar items

|  |  |
| --- | --- |
| RF3 | |
| **Descripción:** | Se debe poder ordenar los ítem sin ningún inconveniente con un método en la clase ítem. |
| **Entradas:** | Archivo.txt con Serie y Movie con su respectiva fecha, nombre y rating. |
| **Salidas:** | Se podrá usar los ítems para pintar elementos en la pantalla. |
| **PreCondiciones:** | Debe existir la clase ítems y el archivo.txt con los datos. |
| **Post-Condiciones:** | Se debe poder usar los datos y modificarlos. |

**RF4:** Guardar archivo.txt

|  |  |
| --- | --- |
| RF4 | |
| **Descripción:** | El programa debe guardar un archivo .txt con la lista de items ordenados por cada atributo y cada atributo debe guardar su archivo.txt, a su vez el nombre del ítem debe estar en mayúsculas. |
| **Entradas:** | Lista ordenada con respecto al archivo.txt. |
| **Salidas:** | Otra versión del archivo.txt organizado según la lista |
| **PreCondiciones:** | Debe existir la lista y está ya debe de estar ordenada. |
| **Post-Condiciones:** | Debe de crearse un archivo.txt modificado por cada item |

**RF5:** Crear clase interfaz grafica

|  |  |
| --- | --- |
| RF5 | |
| **Descripción:** | Se debe de crear una clase en la cual se introducirán varios elementos gráficos que se usaran en el programa. |
| **Entradas:** | Elementos gráficos. (PImage). |
| **Salidas:** | Los elementos gráficos deben de estar inicializados. |
| **PreCondiciones:** | La clase ítem y el archivo.txt deben de existir. |
| **Post-Condiciones:** | Se podrán usar los elementos gráficos, llamarlos y modificarlos. |

**RF6:** Crear interfaz gráfica.

|  |  |
| --- | --- |
| RF6 | |
| **Descripción:** | El programa debe de tener una interfaz gráfica para poder pintar los ítems del archivo.txt. |
| **Entradas:** | Elementos gráficos e imágenes |
| **Salidas:** | Se debe de haber pintado los ítems descritos del archivo.txt |
| **PreCondiciones:** | La clase interfaz gráfica, item y el archivo.txt deben de existir. |
| **Post-Condiciones:** | El programa deberá mostrar los elementos gráficos descritos por el archivo.txt y se deben de ver la interfaz con estos en la pantalla. |