

Proyecto I – TecFlix

Instituto Tecnológico de Costa Rica
Área de Ingeniería en Computadores
Algoritmos y Estructuras de Datos II (CE 2103)
Primer Semestre 2019
Valor 25%



Objetivo General

- Implementar una aplicación que permita paginar miles de resultados.

Objetivos Específicos

- Desarrollar una aplicación en el lenguaje de programación C++
- Investigar acerca de programación orientada a objetos en C++.
- Aplicar **patrones de diseño** en la solución de un problema.
- Aplicar conceptos de paginación en C++.

Descripción del Problema

TecFlix es una aplicación para visualizar metadata y trailers de películas o series de televisión. Es una aplicación de escritorio que consume un dataset de películas. Dicho dataset es provisto como un archivo csv similar a https://drive.google.com/open?id=1nfyB08_-ZYRS8gqmmFfQ6JZ8eGLc6W7K) y la aplicación muestra todas las películas listadas en el dataset junto con información sobre estas. La aplicación se conecta a IMDB (sigue un URL provisto en el dataset, descarga la página y extrae las partes relevantes) y muestra el poster y el trailer.

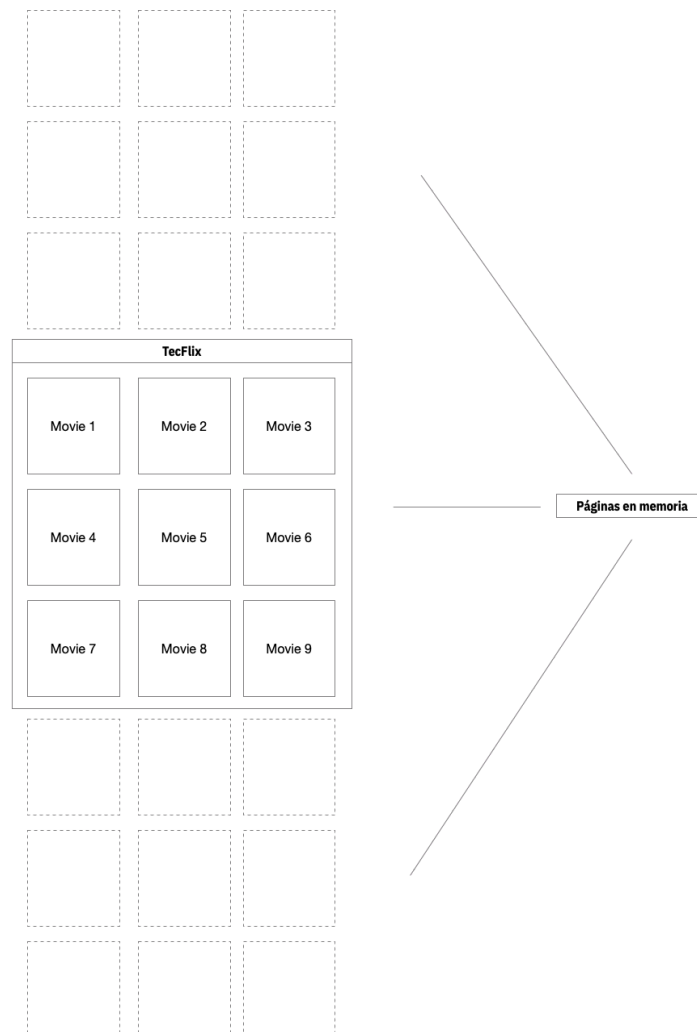


Dado que el dataset puede crecer considerablemente, TecFlix es escalable y para mantener una experiencia de usuario óptima y evitar sobrecargar la memoria del cliente, los resultados se pagan de forma que solo 3 páginas se mantengan en memoria (y en la interfaz gráfica) en cualquier momento: la página anterior, la página actual y la página siguiente.

La aplicación tiene dos modos en el que muestra el listado de películas: paginación tradicional y scroll infinito. Para ambos modos, las siguientes reglas aplican:

1. Cada tarjeta para cada película tiene un tamaño fijo dependiendo del nivel de zoom (se puede hacer zoom en la interfaz gráfica).
2. Considerando el tamaño de la ventana, se calcula dinámicamente la cantidad de tarjetas que puede ver el usuario sin hacer scroll, es decir el tamaño de cada página dependen del tamaño de la ventana y del tamaño de las tarjetas.

En el modo de scroll infinito, se monitorea el evento de scroll. Cuando el usuario hace scroll hacia arriba o hacia abajo, se cargan nuevas páginas y se descargan las páginas no visibles.



En el modo de paginación tradicional, se muestra un paginador con números que permite al usuario avanzar entre páginas haciendo click en los números de página. Al saltar por ejemplo a la página 20, se trae a memoria la 19 y la 21.

La aplicación tiene un modo de no paginación, en el que se carga todo el dataset. Debe mostrarse, el uso de memoria (dentro de la aplicación) y que se muestre claramente cómo se reduce al usar paginación. Se debe calcular y mostrar cuanta memoria usa cada página (incluyendo los elementos de UI que se utilicen).

Documentación requerida

1. Internamente, el código se debe documentar utilizando DoxyGen y se debe generar el HTML de la

documentación.

2. Dado que el código se deberá mantener en GitHub, la documentación externa se hará en el Wiki de GitHub. El Wiki deberá incluir:
 - a. Breve descripción del problema
 - b. **Planificación y administración del proyecto:** se utilizará la parte de project management de GitHub para la administración de proyecto. Debe incluir:
 - Lista de features e historias de usuario identificados de la especificación
 - Distribución de historias de usuario por criticalidad
 - Plan de iteraciones que agrupen cada bloque de historias de usuario de forma que se vea un desarrollo incremental
 - Descomposición de cada user story en tareas.
 - Asignación de tareas a cada miembro del equipo.
 - c. Diagrama de clases en formato JPEG o PNG
 - d. Descripción de las estructuras de datos desarrolladas.
 - e. Descripción detallada de los algoritmos desarrollados.
 - f. Problemas encontrados en forma de bugs de *github*: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.

Aspectos operativos y evaluación:

1. **Fecha de entrega: De acuerdo al cronograma del curso**
2. El proyecto tiene un valor de 25% de la nota del curso.
3. El trabajo es **individual**.
4. Es obligatorio utilizar un GitHub.
5. Es obligatorio integrar toda la solución.
6. El código tendrá un valor total de 85%, la documentación 15%.
7. De las notas mencionadas en el punto anterior se calculará la Nota Final del Proyecto.
8. Se evaluará que la documentación sea coherente, acorde a la dificultad/tamaño del proyecto y el trabajo realizado, se recomienda que realicen la documentación conforme se implementa el código.
9. La nota de la documentación es proporcional a la completitud del proyecto.
10. La documentación se revisará según el día de entrega en el cronograma.
11. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
12. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la cita revisión oficial
13. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
 - a. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
 - b. Si no se utiliza un manejador de código se obtiene una nota de 0.
 - c. Si la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
 - d. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
 - e. El código debe ser desarrollado en C++ (Linux), en caso contrario se obtendrá una nota de 0.
 - f. Si no se siguen las reglas del formato de email se obtendrá una nota de 0.
 - g. La nota de la documentación debe ser acorde a la completitud del proyecto.
14. La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto. El único requerimiento que se consultará durante la defensa del proyecto es el diagrama de clases, documentación interna y la documentación en el manejador de código.

15. Cada estudiante tendrá como máximo 15 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
16. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
17. Cada estudiante es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberán avisar al menos 2 días antes de la revisión a el profesor para coordinar el préstamo de estos.
18. Durante la revisión únicamente podrán participar el estudiante, asistentes, otros profesores y el coordinador del área.