

Ingeniería en Computación

IMDB

Profesora: Adriana Álvarez

1. Motivación

A las personas siempre les ha gustado el entretenimiento por medios visuales tales como películas, series, documentales, entre otros.

Cada persona tiene sus preferencias en cuanto a género o categoría.

Les gusta ir al cine a ver una película en pantalla gigante o verlas desde la comodidad del hogar.

También, se vuelve una afición comentar sobre ellas e incluso ponerles un ranking.

El mundo de las películas no sólo nos entretiene mostrando diversos mundos, dramas, tragedias, comedias, también transporta al que las ve a esos mundos que siempre dejan un mensaje a interpretar en cada individuo.

2. El problema

Se quiere crear un sistema que consolide las películas, series, documentales y cualquier otra producción artística relacionada en un solo repositorio.

Este debe ser accesible a todos los usuarios, pero también debe tener personalización de ellos, en el sentido que se pueden crear listas de favoritos, *rankings*, *reviews*, comentarios, compras de *streaming*, entre otros.

El sistema debe cumplir con todos los requerimientos indicados más adelante en este documento.

3. Arquitectura

Estructure su aplicación como un Sistema N Capas. Investigue más detalles sobre esta arquitectura, de modo que cuente con al menos los componentes básicos ilustrados en la Figura 1. A continuación se da una descripción general de cada una de las capas:

- *User Interface (UI)*: capa de presentación, encargada de mostrar las vistas de interfaz gráfica mediante la cual los usuarios interactúan con la aplicación. Esta capa no toma decisiones ni procesa lógica de negocio, solamente se encarga de mostrar o capturar información.
- *Business Layer (BL)*: capa de lógica de negocio, encargada de modelar las reglas de negocio de la aplicación, toma las decisiones, cálculos, procesos, etc.
- *Data Access (DA)*: capa de acceso a datos, abstrae la comunicación del sistema con la base de datos, no debe confundirse con la base de datos como tal.

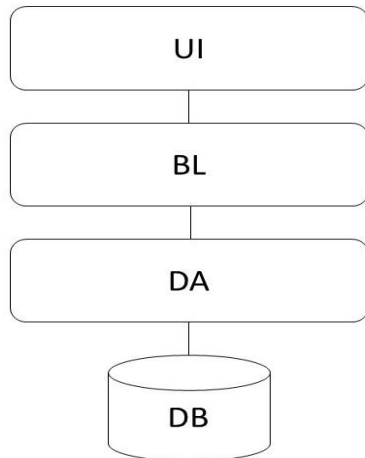


Figura 1 – Arquitectura N-Capas

Es probable que durante el desarrollo del sistema requiera de la implementación de capas adicionales. Durante la revisión se deberá justificar las decisiones hechas sobre el diseño de la aplicación. Adicional a esto se espera una implementación elegante, con un código fuente que siga las normas básicas establecidas en el libro Clean Code (Martin, 2008).

4. Requerimientos

a. Registro

El sistema debe permitir lo siguiente:

- Administradores para las operaciones de seguridad y administración.
- Registro de personas que ingresan al sistema con al menos nombre, apellidos, cédula o su equivalente en pasaporte, fecha de nacimiento, edad, foto, nacionalidad, comunidad a la que pertenecen, correo electrónico y teléfono.
- Cada persona que se conecta al sistema debe hacerlo por medio de un usuario y clave. La clave

debe almacenarse encriptada haciendo uso de un algoritmo de encriptación tal como RSA, SHA, entre otros.

- Deben existir usuarios Administradores para registrar la información de catálogos.
- Todos los datos pueden ser editados considerando los permisos de seguridad.
- Los administradores pueden agregar las películas con al menos la siguiente información:
 - Título
 - Año de estreno
 - Duración
 - Categorías (acción, ficción, romántica, ciencia ficción, drama, terror, entre otras)
 - Director
 - Escritores
 - Actores
 - Fotos de las personas
 - Fotos de la película
 - Review
 - Sinopsis
 - Trailer
 - Plataformas en las que se puede encontrar la película
 - Precio de *streaming*
- El sistema también debe funcionar para series, con las siguientes características adicionales:
 - Temporada
 - Número de temporada con total de temporadas. Por ejemplo: 1/4
 - Episodio
 - Número de episodios con el total de episodios. Por ejemplo: 1/9
 - Documentales

- d. Otros tipos de producciones cinematográficas.
- h) El sistema debe estar diseñado para que sirva para una película, serie, documental u otro.
- i) Las películas también se pueden comprar para *streaming*.
- j) Los usuarios pueden agregar y eliminar películas o series de su carrito de compras sin necesidad de comprarlos o puede comprarlos.
- k) También puede adquirir los productos y elegir un método de pago para sus compras, cada usuario debe tener acceso a su historial de compras.
- l) Podrán dejar *reviews* en cada una de las películas o series que han adquirido con su respectiva calificación en estrellas y un comentario.
- m) Los actores, directores y otros que considere deben contener la siguiente información:
 - a. Nombre completo
 - b. Fecha de nacimiento
 - c. Lugar de nacimiento
 - d. Nacionalidad
 - e. Una breve biografía
 - f. Padres
 - g. Hijos
 - h. Pareja
 - i. Hermanos
 - j. Altura
 - k. Dato de trivia
 - l. Listado de todas las películas, series, otros en lo que haya participado. Cada uno de estos con un *link* al producto.

b. Publicación de *rate* o *reviews*

- a) Se debe visualizar la cantidad de estrellas dadas. De forma

numérica y visual en estrellas y se debe resaltar las estrellas obtenidas del total de 5 estrellas.

- b) También se deben ver los comentarios y los *reviews*.

c. Favoritos

El sistema debe permitir a los usuarios optar por los productos favoritos como un *wish list* y desde ahí ir al detalle del producto por medio de links.

d. Auditoría y bitácora

El sistema debe administrar campos de auditoría para los registros de datos en las tablas de la base de datos y para la última actualización de un registro.

Además, debe tener una bitácora con los cambios de precio de un producto y del dueño del producto.

Utilice *triggers* para cumplir con esta funcionalidad donde se requiera.

Adicionalmente, en este módulo debe incluir una consulta con la bitácora de cambios de precio de los productos.

e. Control y Usabilidad

El sistema permite a los administradores las operaciones de agregar, editar, eliminar o inactivar registros principalmente de los catálogos.

Todos los catálogos deben registrarse, editarse y consultarse por medio del sistema. Se considera no aceptable el registro de datos por medio de la base de datos.

f. Módulo de Consultas

Este módulo les permitirá a los usuarios visualizar información precisa e inmediata.

Todas las consultas deben contener filtros de los datos que se solicitan.

Todas las consultas deben permitir combinación de datos en los filtros y filtros en blanco.

Cada consulta se debe hacer con un solo *query*.

Las consultas mínimas que debe ofrecer el sistema son:

Consultas para administradores:

- a) Top N (valor parametrizable) de los productos que más ventas han hecho con cantidad de ventas realizadas.

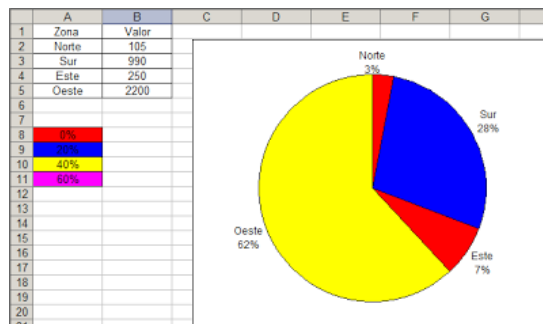
Para usuarios:

- a) Debe existir una pantalla para búsquedas de cualquier producto.
- b) Historial de compras realizadas para 3 meses, 6 meses y 1 año y todas las compras.
- c) Listado de productos vistos recientemente con *link* al detalle de los productos.
- b) Top N (valor parametrizable) de las 250 películas más populares. Listado de productos con sus *reviews*. Que contenga el nombre del producto, categoría y otras características básicas. Ordenado por cualquiera de los campos. Filtros: cualquiera de los campos.

g. Módulo de Estadísticas

Este módulo les permitirá a los administradores tener información precisa e inmediata.

Cada una de las estadísticas deben mostrarse con datos numéricos, porcentajes y gráficos. Por ejemplo:



Todas las estadísticas deben tener los datos, el porcentaje, la cantidad y además el gráfico.

Las estadísticas mínimas que debe ofrecer el sistema son:

- a) Total y porcentaje de productos por clasificación.
- b) Total de usuarios por género y rango de edad.

Todas las estadísticas deben permitir combinación de datos en los filtros.

h. Jobs

El sistema debe generar un reporte al día con el top 10 (parámetro) de productos con mejor calificación según su promedio.

Este reporte debe generarse como un Job de base de datos que corre todos los días a las 12 a.m. de lunes a domingo.

5 Otras consideraciones

1. El sistema debe presentarse en la defensa con datos reales y bien clasificados.
2. Los catálogos deben tener los datos previamente registrados para la defensa del proyecto. Al menos 50 productos, 10 usuarios, 20 compras y con interacciones entre los usuarios y toda la información de catálogos.
3. No se permiten datos basura.
4. Todos los campos que se ingresen en el sistema deben tener su respectiva salida (consulta, estadística, reporte, otro que considere).
5. Las consultas, estadísticas, entre otros *sql* deben ser altamente eficientes. Se evalúa rendimiento y eficiencia.
6. Todas las consultas deben estar programadas de forma eficiente haciendo uso de *joins* donde se requiera. No es aceptable hacer consultas individuales cuando pueden realizarse eficientemente con *joins*.
7. Debe utilizar Oracle como motor de base de datos. No se acepta otro motor de base de datos.
8. Usted debe identificar donde aplica crear tablas catálogo para que el sistema sea flexible. Se calificará diseño.
9. El sistema debe cumplir con los atributos de calidad de usabilidad, flexibilidad, fácil modificación, eficiencia, seguridad, entre otros.
10. La aplicación debe ser gráfica.
11. La aplicación debe ser de escritorio.
12. Se evaluará el diseño y la correcta identificación de los parámetros.
13. La aplicación debe tener un nombre y logo creado por el equipo. No puede ser un logo copiado de internet o de otras personas y debe ser original. Puede solicitar ayuda de un estudiante de diseño. Debe indicar la fuente.
14. Debe justificar sus decisiones de diseño en el modelo conceptual o lógico.
15. El código debe trabajarse desde un programa de control de versiones y se deberá demostrar en la defensa la cantidad de *commits* realizados por cada miembro del equipo. Se recomienda por ejemplo *Gitlab*, *Github*.
16. El código debe ser en inglés.
17. El modelo conceptual debe estar en inglés.
18. El equipo puede estar conformado por grupos de un número máximo según le indique la profesora.
19. La entrega oficial se debe hacer en un zip en el *tecDigital* antes de la fecha y hora ahí indicada. No se aceptarán trabajos posteriores a esa fecha y hora. En caso de problemas de conexión, se debe aplicar el plan B de la presentación de Admin.
20. La entrega debe contener los archivos fuente y el ejecutable de la aplicación (si aplica). No se acepta un *link* a un repositorio.
21. La documentación debe entregarse en *pdf*, *latex*, *doc*. No se acepta un *link* al documento.
22. Cualquier sospecha de copia anulará el trabajo y se procederá con el trámite administrativo.

23. El mejor proyecto tendrá +1 punto extra.

6 Documentación

La siguiente documentación debe ser entregada:

1. Diagrama de entidad – relación (modelo conceptual). Debe ser en inglés.
2. Manual de usuario con la descripción del propósito y uso del sistema que incluya imágenes significativas de la aplicación. Si utiliza un *wiki* para la documentación, esta debe accederse desde la aplicación y el *wiki* completo debe ser entregado en el *tecDigital*.
3. El manual de usuario y las etiquetas del sistema pueden ser en español o en inglés.
4. Diccionario de datos.
5. Documento de matriz de casos de prueba (en Excel) con al menos 50

casos de prueba con los cuales se probará el programa. Debe contener un id como enumerador del caso de prueba, la descripción del caso de prueba, el resultado esperado y el estado (Certificado, Con Error, Pendiente). Además de una tabla pivote con la cantidad de casos de prueba en cada estado y el total.

6. El código debe estar documentado internamente según *Clean Code*. Los procedimientos, funciones o paquetes importantes deben ir documentados con descripción, el autor principal, fecha de creación.

7 Referencias

Puede utilizar el sistema IMDB como referencia:

<https://www.imdb.com/>