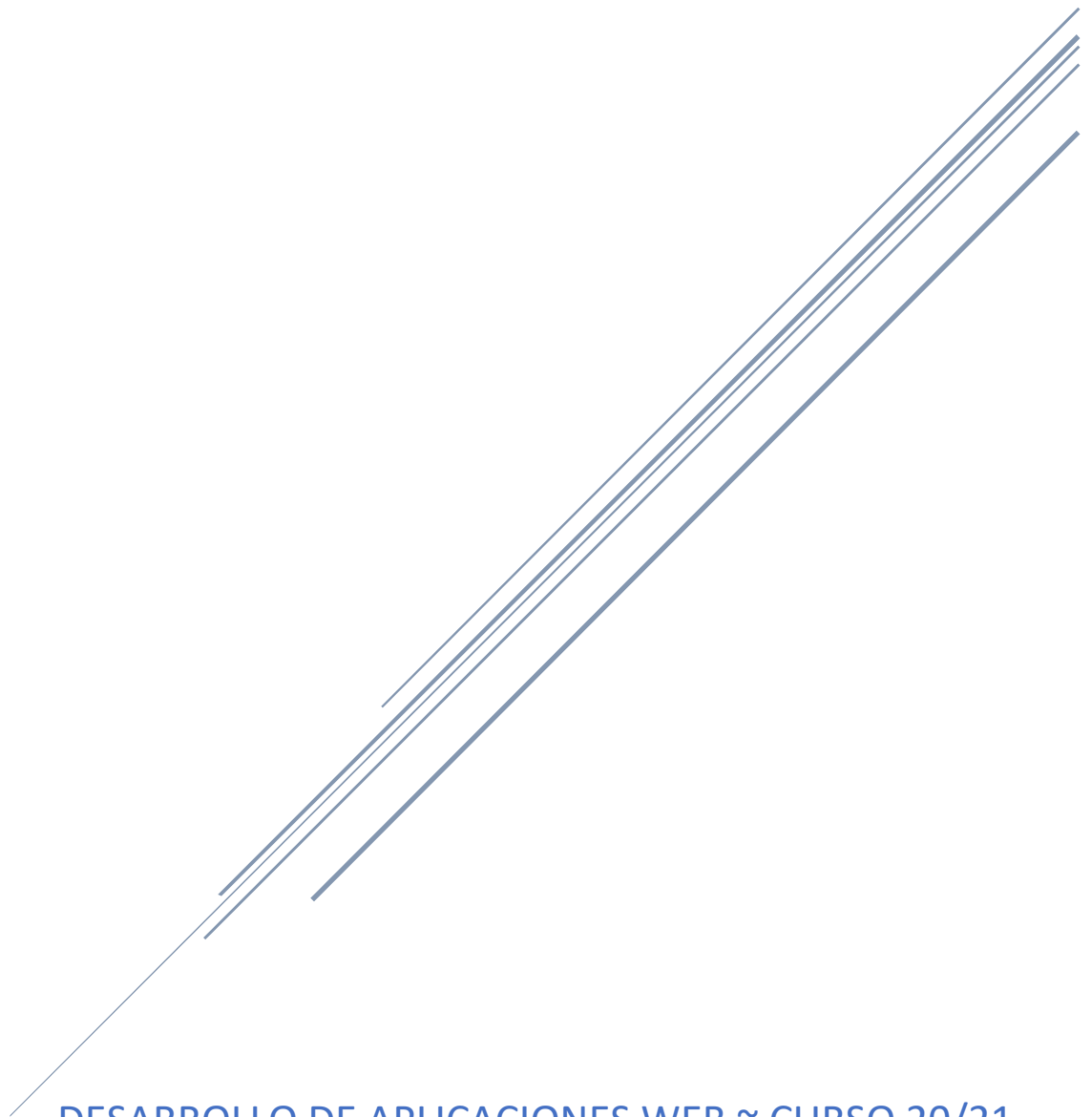


MI PROYECTO PL / SQL

BASE DE DATOS RELACIONAL VIDEOCLUB



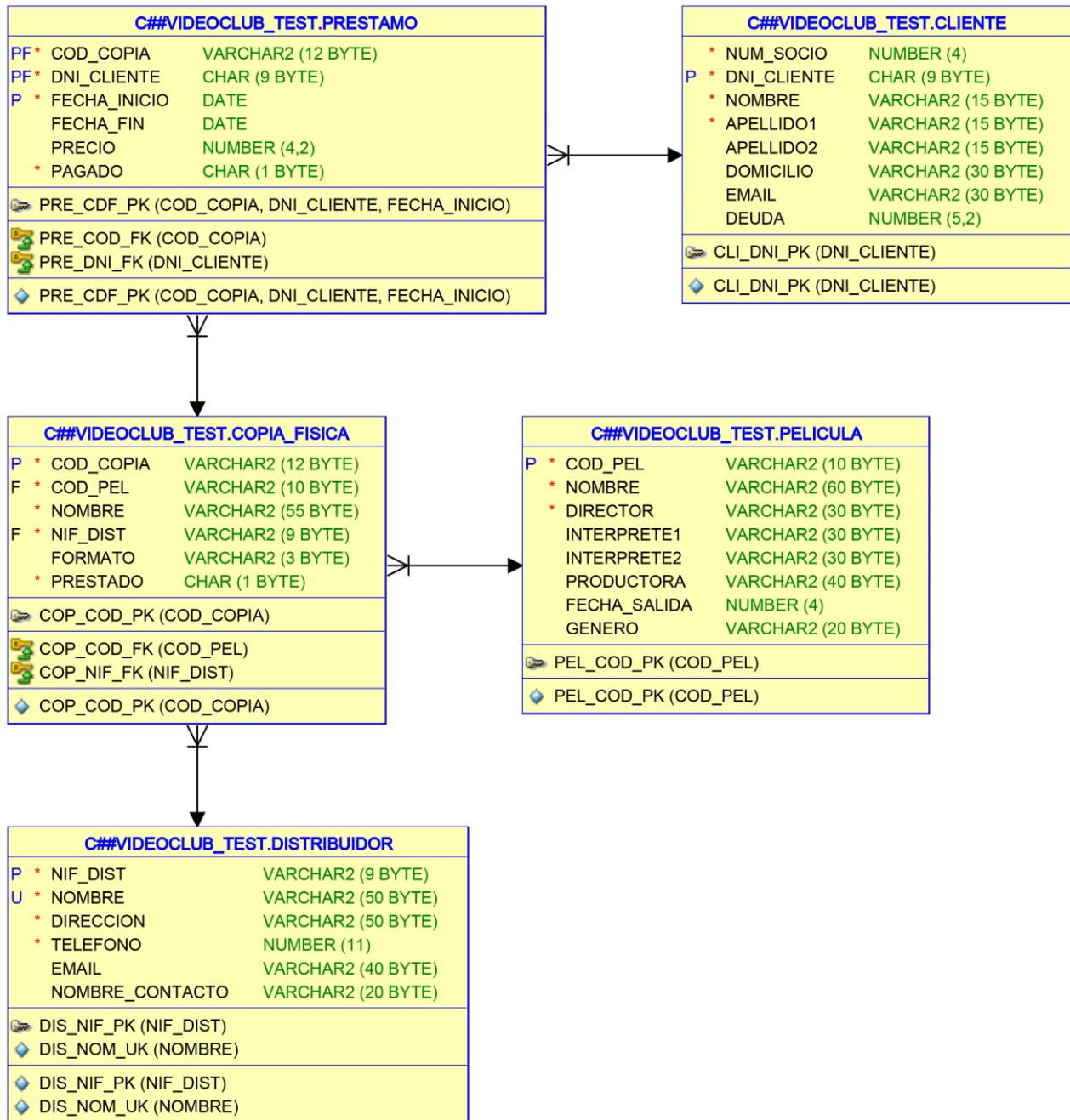
DESARROLLO DE APLICACIONES WEB ~ CURSO 20/21

ALEJANDRO SALGADO CERDEIRA

ÍNDICE

| | |
|--|---|
| 1.- MODELO RELACIONAL | 1 |
| 2.- DESCRIPCIÓN DE LA BASE DE DATOS USADA..... | 2 |
| 3.- REQUISITOS FUNCIONALES DESARROLLADOS | 2 |
| 4.- REPOSITORIO EN GITHUB..... | 5 |

1.- MODELO RELACIONAL



2.- DESCRIPCIÓN DE LA BASE DE DATOS USADA

La base de datos simula la gestión de un videoclub. Fue el proyecto entregable realizado para esta asignatura al final del primer trimestre y se decidió expandir el proyecto para completar el año lectivo sobre el mismo trabajo.

Aunque podríamos considerarla un bosquejo de lo que debería ser en la realidad, para demostrar los conocimientos de programación en PL/SQL se consideró como apta, una vez añadidas las tablas adicionales de auditoría, que no se muestran en el modelo relacional debido a su naturaleza independiente.

Como podemos ver, la estructura se basa en la parte más comercial del modelo, girando en torno a los préstamos y las copias físicas, las cuales deben ser seguidas con precisión para evitar pérdidas o robos, así como en la gestión de pagos y deudas de los clientes.

Es importante notar que para este ejercicio se han dado algunos supuestos que no serían así en un modelo final. La fecha de inicio y fin del préstamo aquí se introducen “a mano” por los empleados, cuando en la vida real se harían de manera automática con la gestión del sysdate, así como el cálculo de precios que también se han falseado.

En cuanto a los procedimientos, disparadores y funciones, se entregan los demandados por los requisitos, pero de nuevo, es un esbozo de lo que debería ser, ya que hay muchos más que se podrían implementar.

3.- REQUISITOS FUNCIONALES DESARROLLADOS

FUNCIONES:

Se desarrollan dos funciones con la idea de facilitar un posible análisis de datos de ventas:

1. Función que dado un año y un formato devuelva el porcentaje de alquileres para ese año y formato: Es decir, si queremos saber que porcentaje del total de alquileres del año 2015 corresponde a DVDs, esta función nos devuelve el dato.
2. Función que dado un año y un género cinematográfico nos devuelva el total de alquileres: De esta manera podríamos usar la función a final de año para analizar fácilmente las preferencias de nuestros clientes y así preparar los pedidos de los nuevos lanzamientos de manera más eficiente.

DISPARADORES:

Se han desarrollado los disparadores, en este ejercicio, pensando más en el control de la base de datos y su consistencia, que en la automatización de procesos:

1. Disparador de instrucción: Que impide modificar el campo Número de Socio de los clientes. Esto es así porque el número de socio se genera mediante una secuencia cuando insertamos uno nuevo. Aunque no forma parte de la clave primaria, puesto que se usa el DNI para ello, nos interesa que no se repita y que aunque se borre un cliente, ese número de socio quede desierto. De esta manera podemos saber fácilmente cuántos socios hemos tenido en la vida del videoclub. Con este disparador protegemos la integridad de este dato.
2. Disparadores de fila: Se ha creado uno por cada tabla para poder tener tablas de auditoría por cada una de ellas. La estructura de la información guardada es la siguiente:

| USUARIO | FECHA/HORA | OPCION | OLD_REG | NEW_REG |
|---------|------------|--------|---------|---------|
| | | | | |

Usuario: El usuario que ha hecho la operación.
 Fecha/Hora: Timestamp para tener hasta los milisegundos, en caso de hacer muchas modificaciones a la vez.
 Opción: Valor numérico para facilitar la búsqueda:
 1 – Insert
 2 – Update
 3 – Delete
 Old_Reg: Null si es insert.
 New_Reg: Null si es delete.

Los dos Reg se componen de una cadena en la que están concatenados todos los datos de todas las columnas de la tabla a la que se le está haciendo la auditoría. Se separan con el carácter asterisco (*) ya que no se usa en ningún otro sitio de la base de datos. De esta manera se puede separar la información con facilidad.

PROCEDIMIENTOS:

1. Procedimiento anónimo que permite probar todas las funcionalidades de la base de datos.
2. Procedimiento para insertar copias físicas de las películas: Se ha desarrollado para facilitar la tarea de los trabajadores. Se ejecuta insertando el código de la película, el formato de la copia (entre VHS, DVD y BR que son los que oferta nuestro videoclub) y el NIF del distribuidor. Comprueba que los datos proporcionados son correctos y después inserta en la base de datos la información, añadiendo un código de copia único, que será la PK y que contiene la información de la película y su formato, el nombre de la película y el estado de alquiler, por defecto 'N' (No).
3. Procedimiento para imprimir un listado de balance anual completo de alquileres: Mediante cursores anidados se revisa la actividad de cada cliente mes a mes, se calcula el total por mes y finalmente por año, con el siguiente formato:

BALANCE TOTAL DEL (AÑO):

#####

MES/AÑO: (DD/YYYY)

-- CLIENTE: Apellido, Nombre Nº Socio: (num) DNI: (dni)

- Nº Alquileres: (num) De los cuales: VHS: (num) DVD: (num) BR: (num)

- Películas sin devolver: (num)

-Listado de los códigos de las películas sin devolver:(Sólo si tiene alguna)

- Total pagado: (num) Total a deber: (num)

[Todos los clientes que han realizado alquileres ese mes]

BALANCE TOTAL FIN MES/AÑO: MES/AÑO: (DD/YYYY)

- Nº Alquileres: (num) De los cuales: VHS: (num) DVD: (num) BR: (num)

- Películas sin devolver: (num)

-Listado de los códigos de las películas sin devolver:(Sólo si tiene alguna)

- Total pagado: (num) Total a deber: (num)

#####

[Todos los meses en los que ha habido alquileres]

#####

BALANCE TOTAL FIN AÑO: (YYYY)

- Nº Alquileres: (num) De los cuales: VHS: (num) DVD: (num) BR: (num)

- Películas sin devolver: (num)

-Listado de los códigos de las películas sin devolver:(Sólo si tiene alguna)

- Total pagado: (num) Total a deber: (num)

GESTIÓN DE EXCEPCIONES

Se ha mostrado especial énfasis al control de las mismas, mediante el uso de excepciones personalizadas, control de las propias generadas por SQL y gestionando si nos interesaba o no permitir la interrupción brusca del programa. Por ello, la única manera que se contempla como interrupción del flujo de manera brusca es en el caso del disparador de instrucción, puesto que se trata de un elemento de seguridad.

4.- REPOSITORIO EN GITHUB

Enlace: [JandroSalgadoC/Bases De Datos-Proyecto PLSQL: Alumno: Alejandro Salgado - Curso: 20/21 \(github.com\)](https://github.com/JandroSalgadoC/Bases_De_Datos-Proyecto_PLSQL)

Explicación de uso:

En el directorio raíz encontrará:

- El archivo README que contiene esta misma información.
- Este archivo pdf.
- El archivo <Script_Creacion_Completa.sql > que contiene un script que genera la base de datos completa INCLUIDOS los código PL SQL de creación de funciones, procedimientos y disparadores. Esto es así porque se necesitan para la inserción posterior de datos.
- El archivo <Script_Procedimiento_Principal.sql> que, una vez cargada la base de datos, le guiará en cómo probar las funcionalidades.

Para la comodidad del análisis del código, encontrará un directorio por cada tipo de lenguaje necesario para la base de datos. Uno para DDL, otro para DML y otro para PL SQL, en este último encontrará los mismos códigos que en el script completo, pero dividido en cada requisito de este documento.

Finalmente encontrará un directorio llamado DEPRECATED en el que he guardado versiones previas del código que no afectan al ejercicio.