



Base de datos para videoclub

- Luis Alberto De La Cruz Guerrero
- Miguel Angel Ramirez Cruz
- Oscar Elí Melchor Hernandez
- Cesar Eduardo Gonzales Noriega
- Diego Alejandro macías Espejel



Descripción de proyecto



Descripción



Modelo de negocio

El video club Oztral requiere un aplicación web (como YouTube) para poder vender o alquilar las películas de su catálogo.

Para esto se ha decidido diseñar una base de datos que pueda satisfacer sus necesidades de acuerdo a su modelo de trabajo

Modelo de negocio

De acuerdo a usuarios

- Los clientes pueden alquilar o comprar una película.
- El precio es distinto para el alquiler o venta.
- Solo pueden hacer una compra a la vez.
- Los usuarios pueden puntuar las películas.
- La forma de pago de los usuarios son tarjetas de credito o debito (pueden tener varias).
- Los usuarios cuentan con un registro.

De acuerdo a las películas

- Las películas cuentan con información básica (actores, director, calificación, clasificación, categoría, etc).
- Poseen dos precios, alquiler y venta.
- Los actores y directores cuentan con información básica (nombre, biografía, edad, etc).

De acuerdo a administración

- El administrador puede alterar valores de las películas, actores, directores.
- Todas las acciones del administrador son guardadas en una bitácora.

Fase de diseño

01

Diseño conceptual

Se crea un diagrama Entidad-Relación.

02

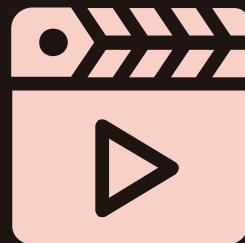
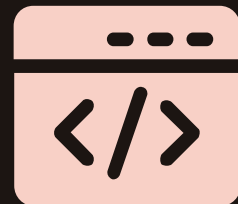
Diseño lógico

Se analizan las propiedades de cada entidad.

03

Diseño físico

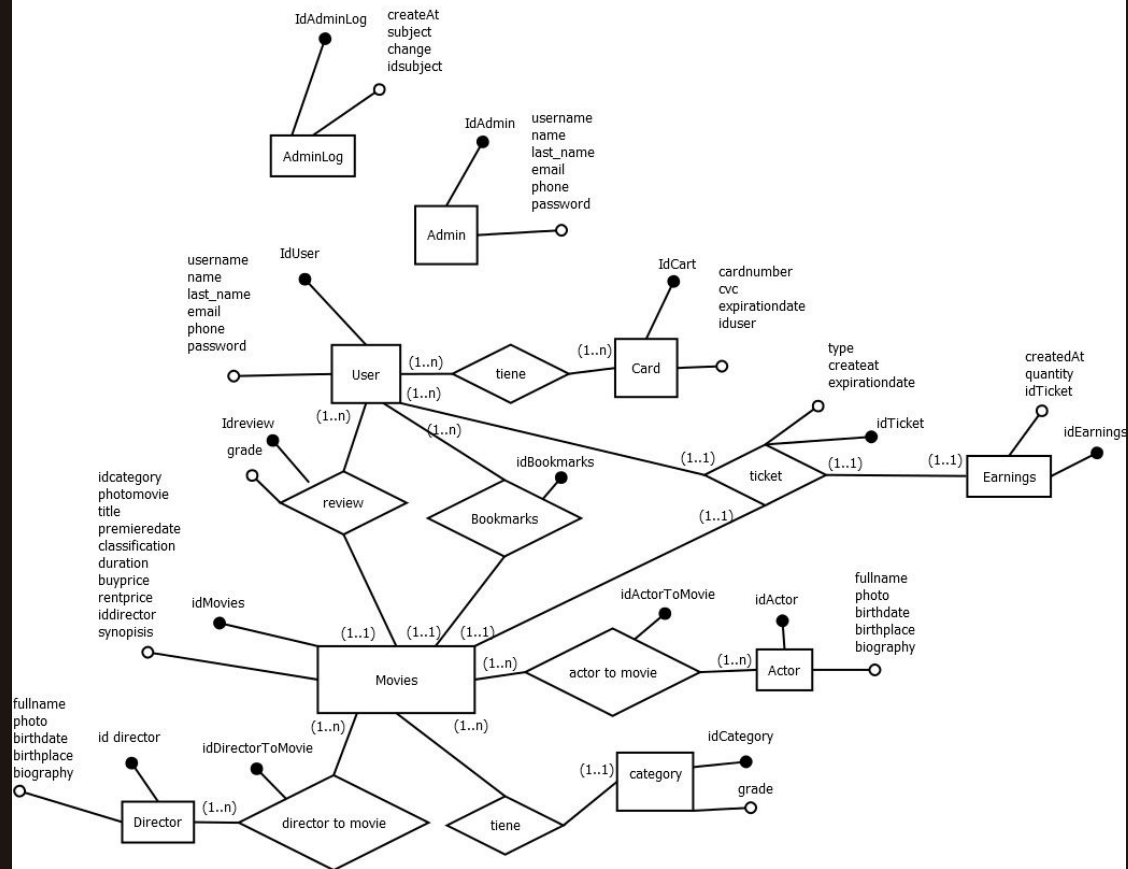
Código SQL para la creación de la base de datos.





01

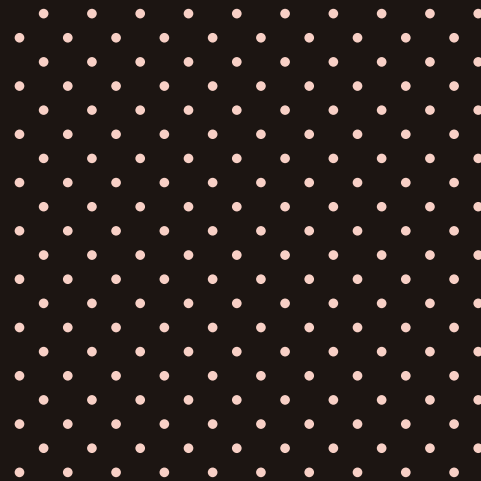
Diseño conceptual





02

Diseño lógico



Diseño Lógico

USER

(id, username, name, lastname, email, phone, password)

ADMIN

(id, username, name, lastname, email, phone, password)

- id es Int es la clave primaria, no puede ser nula y es autoincremental.
- username es Varchar, contiene el nombre del Usuario, no puede ser nulo y debe ser único.
- name es Varchar, contiene el nombre verdadero del Usuario, no puede ser nulo.
- lastname es Varchar, contiene el apellido verdadero del Usuario, no puede ser nulo.
- email es Varchar, contiene el correo del Usuario, no puede ser nulo y debe ser único.
- phone es Varchar, contiene el teléfono del Usuario, no puede ser nulo y es único.
- password es Varchar, contiene la contraseña del Usuario, no puede ser nulo.

Diseño Lógico

ACTOR

(id, fullname, photo, birthdate, birthplace, biography)

DIRECTOR

(id, fullname, photo, birthdate, birthplace, biography)

- id almacena el id de cada fila, es de tipo int, es auto incrementable y no es nulo.
- fullname es tipo varchar, y no es nulo.
- photo es tipo Varchar y no es nulo.
- birthdate es tipo Date y no puede ser nulo.
- birthplace es de tipo Varchar y no puede ser nulo.
- biography es de tipo Varchar y no puede ser nulo.



03

Diseño físico



Diseño Físico

```
CREATE TABLE `User` (  
  `id` INTEGER NOT NULL AUTO_INCREMENT,  
  `username` VARCHAR(50) NOT NULL,  
  `name` VARCHAR(50) NOT NULL,  
  `lastname` VARCHAR(50) NOT NULL,  
  `email` VARCHAR(50) NOT NULL,  
  `phone` VARCHAR(50) NOT NULL,  
  `password` VARCHAR(50) NOT NULL,  
  UNIQUE INDEX  
  `User_username_key`(`username`),  
  UNIQUE INDEX `User_email_key`(`email`),  
  UNIQUE INDEX `User_phone_key`(`phone`),  
  PRIMARY KEY (`id`)  
) DEFAULT CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;
```

```
ALTER TABLE `Card` ADD CONSTRAINT  
`Card_userId_fkey` FOREIGN KEY (`userId`)  
REFERENCES `User`(`id`) ON DELETE RESTRICT ON  
UPDATE CASCADE;
```

```
ALTER TABLE `Ticket` ADD CONSTRAINT  
`Ticket_userId_fkey` FOREIGN KEY (`userId`)  
REFERENCES `User`(`id`) ON DELETE RESTRICT ON  
UPDATE CASCADE;
```

```
ALTER TABLE `Review` ADD CONSTRAINT  
`Review_userId_fkey` FOREIGN KEY (`userId`)  
REFERENCES `User`(`id`) ON DELETE RESTRICT ON  
UPDATE CASCADE;
```

Fase de implementación

01

Inserts a la base de datos

Agregando contenido a todas las tablas dentro del sistema.

02

Creación de usuarios

Cada uno con sus respectivos permisos.

03

Vistas del sistema

Se generan métodos que nos permiten mostrar datos.

04

Procesos almacenados

Creamos instrucciones que realizan tareas en específico.

05

Triggers para el sistema

Se disparan reglas asociadas a una tabla.

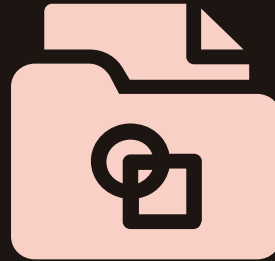
06

Backup de la base de datos

Respaldo para asegurar la integridad de la base de datos.

INSERTS

```
INSERT INTO Category (name) VALUES ('Acción');  
INSERT INTO Category (name) VALUES ('Ciencia Ficción');  
INSERT INTO Category (name) VALUES ('Comedia');  
INSERT INTO Category (name) VALUES ('Drama');  
INSERT INTO Category (name) VALUES ('Musical');  
INSERT INTO Category (name) VALUES ('Romance');  
INSERT INTO Category (name) VALUES ('Superhéroes');  
INSERT INTO Category (name) VALUES ('Terror');
```



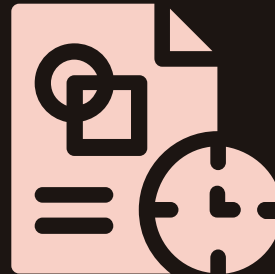
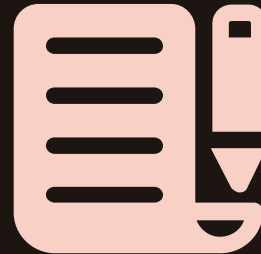
USUARIOS Y PERMISOS

```
GRANT SELECT, INSERT, DELETE ON cineclub .  
Card TO 'client'@'localhost';  
GRANT SELECT, INSERT, UPDATE, DELETE ON  
cineclub . User TO 'client'@'localhost';  
GRANT SELECT, INSERT, DELETE ON cineclub .  
Ticket TO 'client'@'localhost';  
GRANT SELECT, INSERT ON cineclub . Review  
TO 'client'@'localhost';  
GRANT SELECT, INSERT, DELETE ON cineclub .  
Bookmark TO 'client'@'localhost';  
FLUSH PRIVILEGES;
```



VISTAS

```
CREATE TABLE `User` (  
  `id` INTEGER NOT NULL AUTO_INCREMENT,  
  `username` VARCHAR(50) NOT NULL,  
  `name` VARCHAR(50) NOT NULL,  
  `lastname` VARCHAR(50) NOT NULL,  
  `email` VARCHAR(50) NOT NULL,  
  `phone` VARCHAR(50) NOT NULL,  
  `password` VARCHAR(50) NOT NULL,  
  UNIQUE INDEX `User_username_key`(`username`),  
  UNIQUE INDEX `User_email_key`(`email`),  
  UNIQUE INDEX `User_phone_key`(`phone`),  
  PRIMARY KEY (`id`)  
) DEFAULT CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;
```



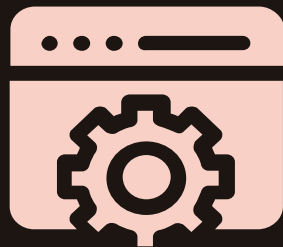
PROCEDIMIENTOS ALMACENADOS

```
CREATE PROCEDURE getMyMovies(IN USER_ID int)
BEGIN
SELECT * FROM Ticket
JOIN Movie M on M.id = Ticket.movieId
WHERE userId = USER_ID AND NOT(ISNULL(movieId));
END;
```



TRIGGERS

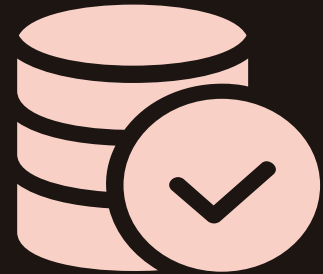
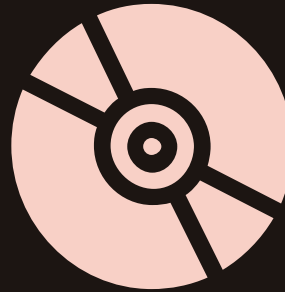
```
CREATE TRIGGER adminLogAddedActor
BEFORE INSERT
ON _ActorToMovie FOR EACH ROW
BEGIN
INSERT INTO AdminLog (subject, `change`, subjectId)
VALUES ('Movie', 'Actor added to Movie.', new.b);
INSERT INTO AdminLog (subject, `change`, subjectId)
VALUES ('Actor', 'Actor added to Movie.', new.a);
END;
```



BACKUP

```
mysqldump -u [username] -p [database_name]  
[tbl_name ...] > [backup_name].sql
```

```
mysqldump -u root -p cineclub > backup.sql
```



Gracias

Do you have any questions?

addyouremail@freepik.com

+91 620 421 838

yourcompany.com



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution