Documento Descripción del Proyecto
Etapa II: Biblioteca de Papers Online

Integrantes: Manuel Barrera
Mauricio Moragra
Silvio Vera

Profesora: Eliana Providel
Manuel García

Fecha: 10 de noviembre de 2020

1

Índice

Descripción problema **3**

Requerimientos **3**
- Funcionales 3
- No Funcionales 3

Diagrama de casos de uso **4**

Diagrama de secuencia **5**
- Agregar paper 5

Mockups **6**
- Vista Principal 6
- Vista Papers 7

Modelo entidad relación **8**

Modelo relacional **9**

Supuestos **9**

Diccionario de datos **10**
- administrador 10
- cliente 10
- paper 10
- orden 11
- adm_pap 11

Consultas etapa 1 **12**
- 2 alter 12
- 3 select (con join) 12
- 2 update 12
- 3 insert 13
- 2 delete 13
- drop 13

Consultas etapa 2 **14**
- Subconsultas 14
- Operadores 14
- Función 15
- Group by 15
- Schema 16

Descripción problema

Como equipo de trabajo se nos presenta la problemática en la cual se requiere crear una app web que registre datos en una base de datos en la cual se tenga como objetivo el facilitar la compra y reserva de papers por parte del público que visite su sitio web, teniendo en cuenta que se debe controlar cada venta de los papers tanto el número de ventas como reservas por papers específicos pudiendo guardar datos relevantes respecto a los clientes para posteriormente poder hacer toma de decisiones respecto las compras y reservas de estos.

Además adicional a lo anterior mencionado se pide una funcionalidad donde el administrador pueda modificar (ordenar, agregar o borrar) los papers ingresados o por ingresar así pudiendo actualizar sus ventas constantemente para alcanzar un mayor número de clientes.

Requerimientos

❖ Funcionales

- ➢ Distinción de usuarios entre cliente y administrador.
- ➢ El cliente podrá realizar las siguientes acciones:
  - ■ Comprar papers.
  - ■ Reservar papers.
- ➢ El Administrador podrá realizar las siguientes acciones:
  - ■ Agregar papers.
  - ■ Quitar papers.
  - ■ Revisar número de ventas y reservas.
  - ■ Modificar papers
- ➢ Buscar papers por nombre.
- ➢ Listar papers por categoría.
- ➢ Ver detalles de un papers.

❖ No Funcionales

- ➢ Lograr mantenibilidad sin afectar gran parte del sistema, se logra manejando bien.
- ➢ Tiene que ofrecer seguridad a sus clientes en cuanto a sus datos.

Diagrama de casos de uso

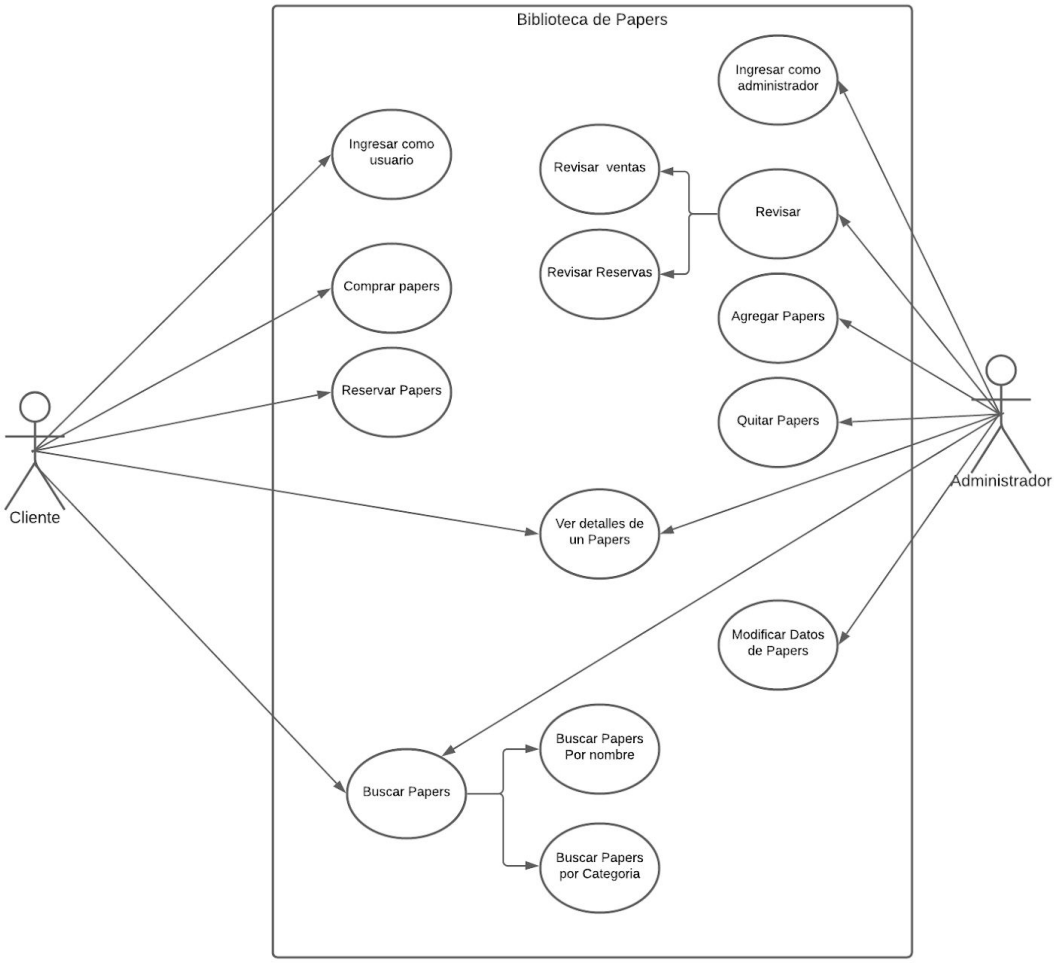


Figura 1: En el siguiente diagrama se representa la forma en que actúa el administrador y el cliente dentro de nuestra app web.

Diagrama de secuencia

Agregar paper

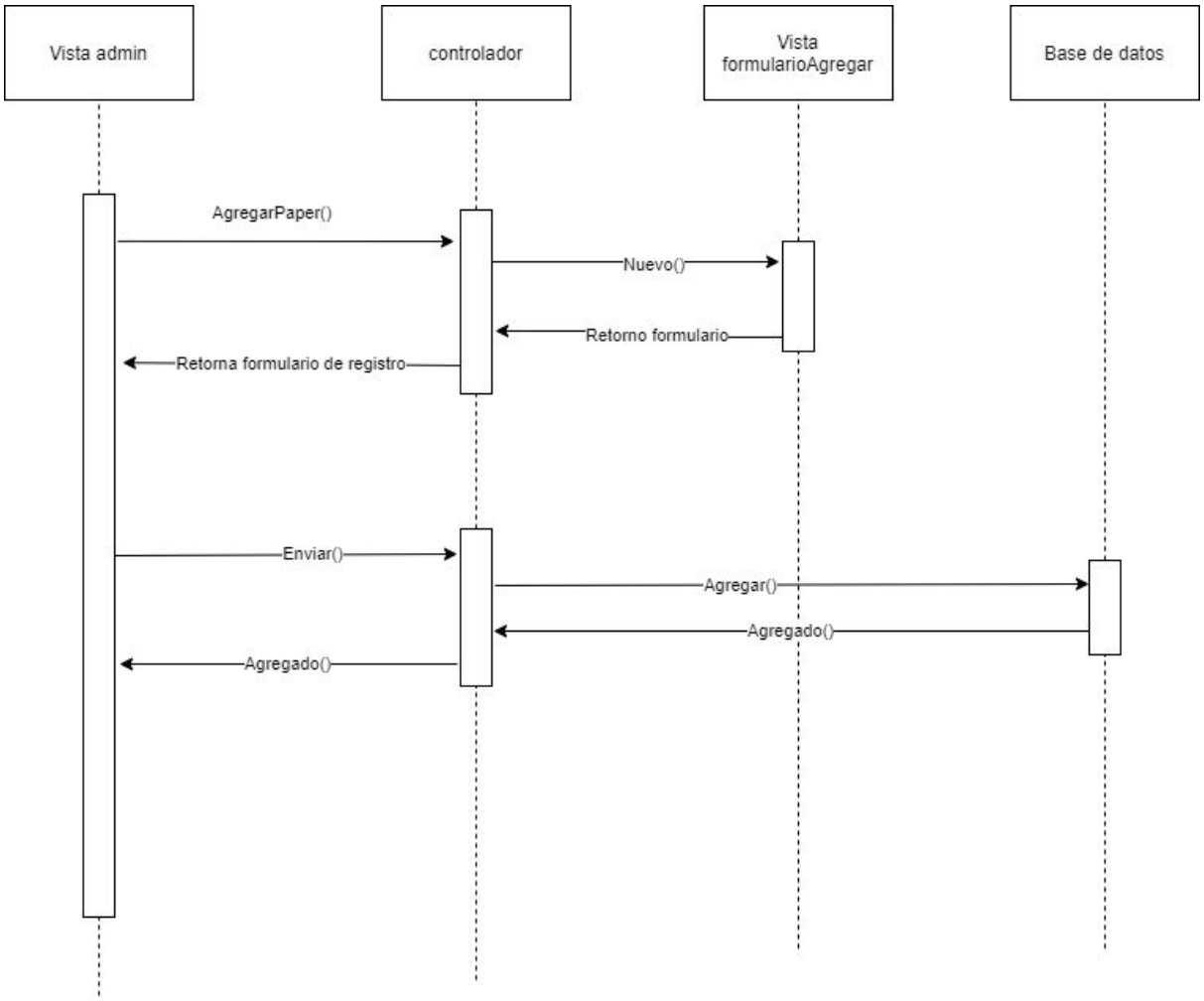


Figura 2: Transacciones entre la vista de administrador y vista de formulario donde se ingresaran los datos, donde el controlador realizara las transacciones entre ambas vistas, y luego envía los datos al SGBD.

Mockups

Vista Principal

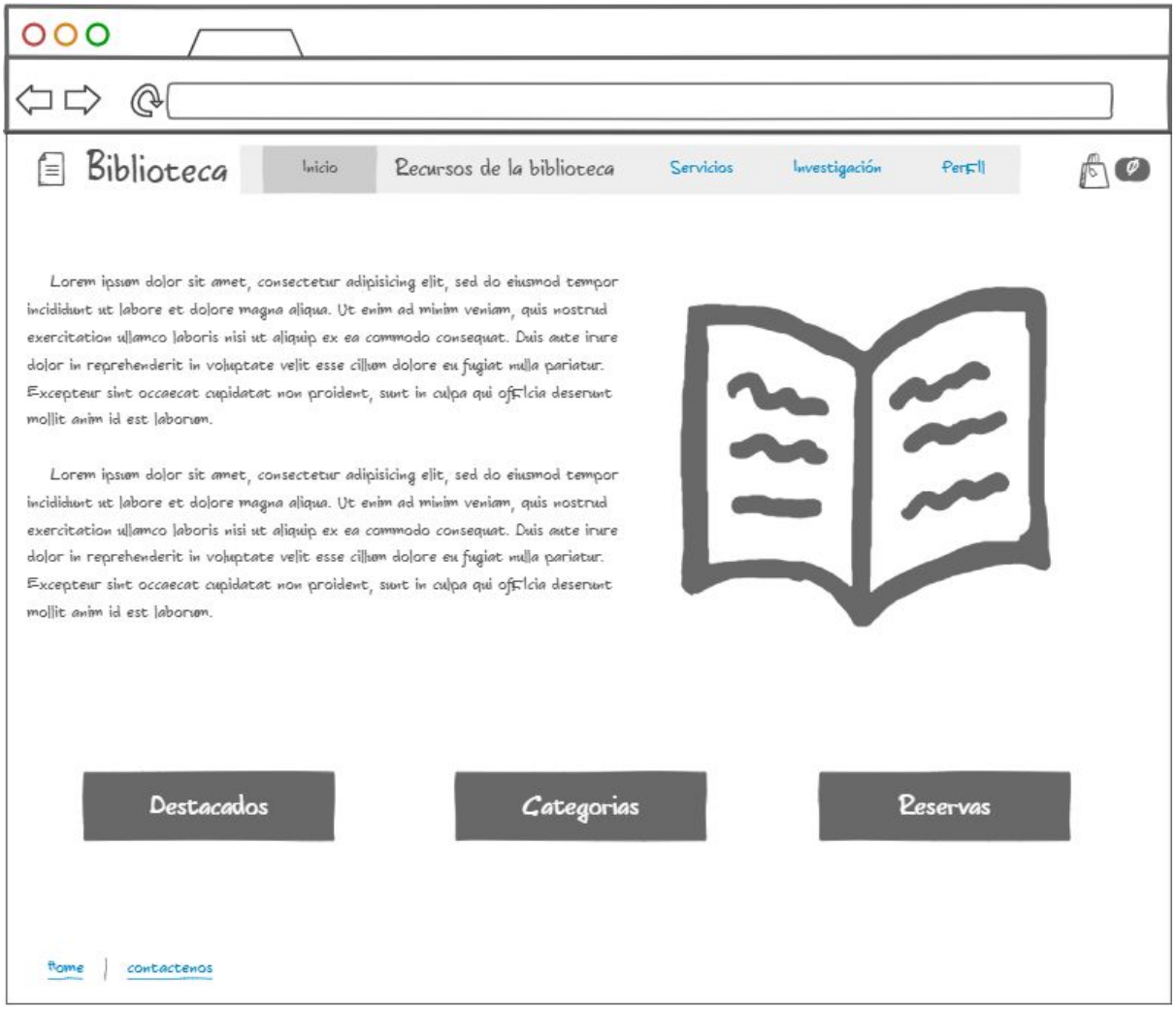


Figura 3: La vista actual es la que se le muestra al usuario al entrar a la página, en esta te puedes mover entre diferentes vistas.

Universidad de Valparaíso CHILE

Vista Papers

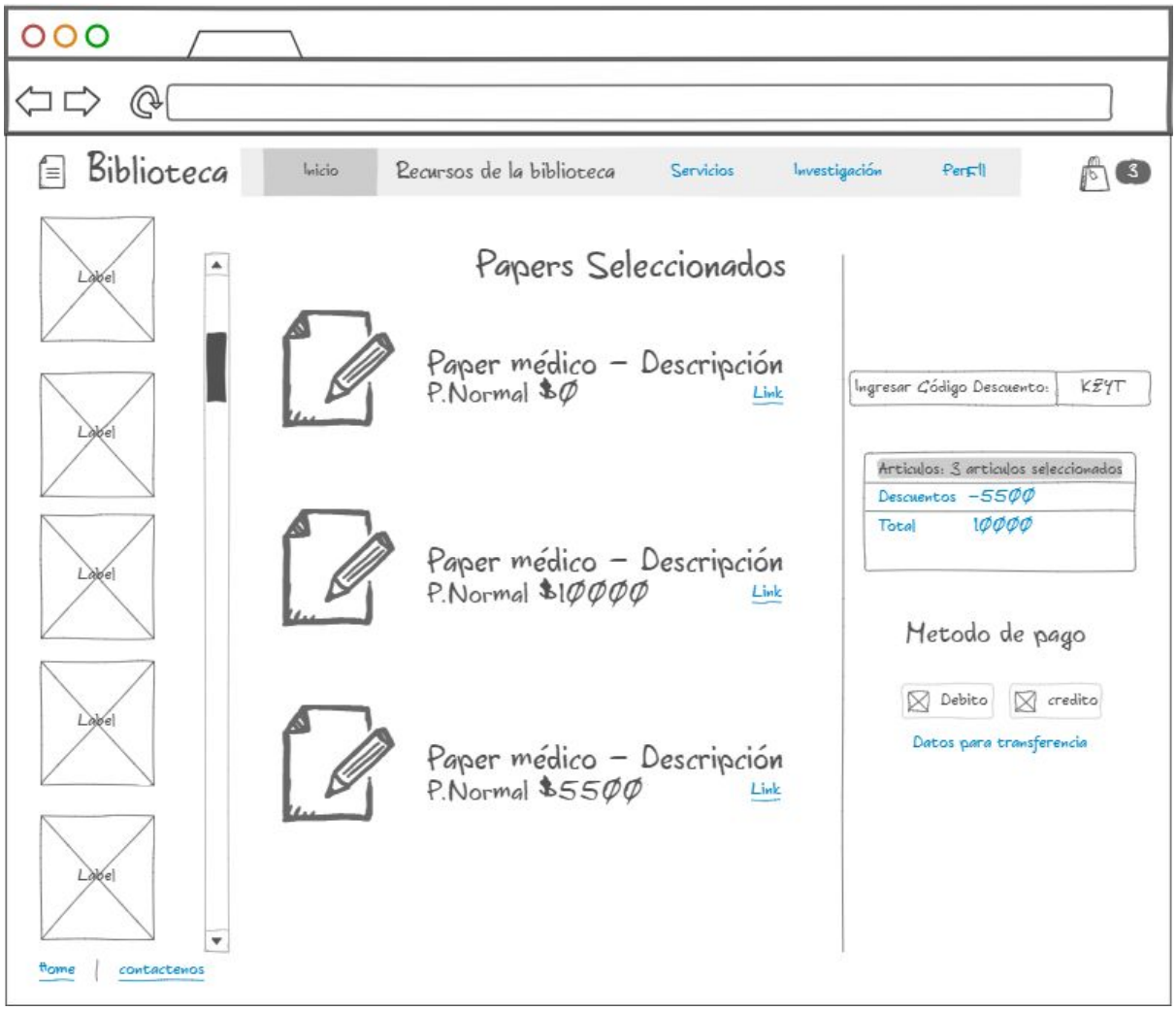
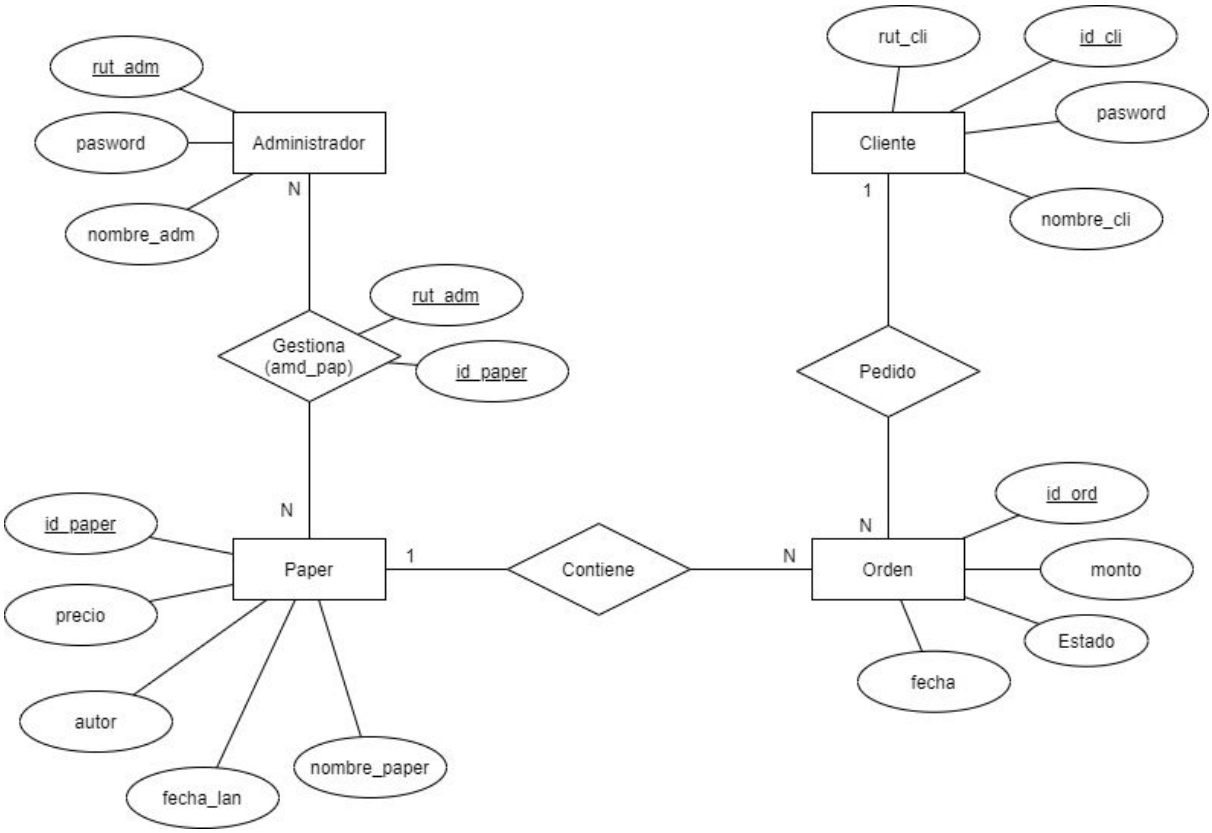
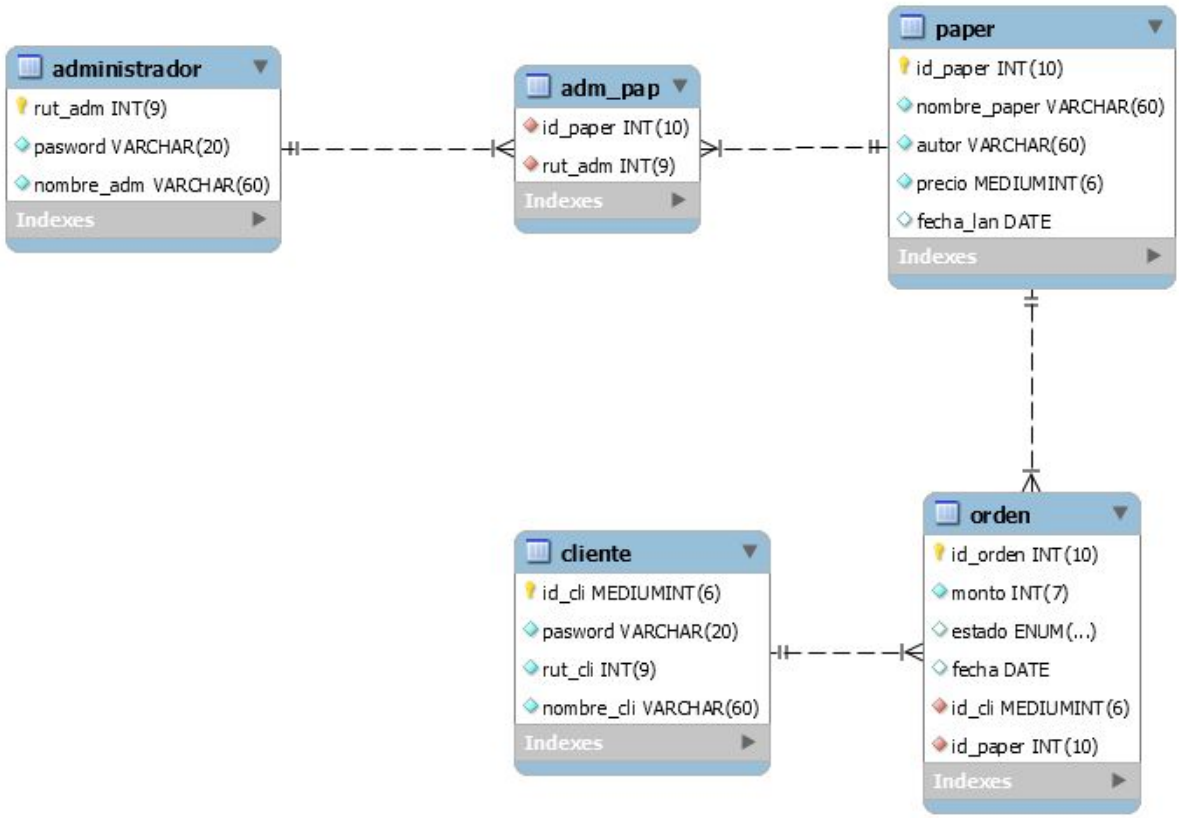


Figura 4: Corresponde a la vista del usuario al seleccionar los papers, es el carrito de venta y muestra el monto total a pagar por los papers, al monto total se le puede descontar con un código de descuento.

Modelo entidad relación



Modelo relacional



Supuestos

- ❏ Todas las transacciones son realizadas a través de un método seguro.
- ❏ Los clientes no tendrán un rut asociado a dos o más id.
- ❏ Todas las compras realizadas tienen un único método el cual es de forma online.
- ❏ No existirán más de 20 administradores.
- ❏ El cliente puede realizar un número indefinido de órdenes.
- ❏ No existe reembolso por la compra de un paper.
- ❏ Las reservas se deben cancelar como máximo 1 día antes de su lanzamiento.

Universidad de Valparaíso CHILE

Diccionario de datos

administrador

Nombre atributo	PK/ FK	Tipo de dato	Null/ Not Null	Valores por defecto
rut_adm	PK	Unsigned Int(9)	Not null	No
pasword		Varchar(20)	Not null	No
nombre_adm		Varchar(60)	Not null	No

cliente

Nombre atributo	PK/ FK	Tipo de dato	Null/ Not Null	Valores por defecto
id_cli	PK	unsigned mediumint(6)	Not Null	No
pasword		Varchar(20)	Not null	No
rut_cli		unsigned Int(9)	Not null	No
nombre_cli		Varchar(60)	Not null	No

paper

Nombre atributo	PK/ FK	Tipo de dato	Null/ Not Null	Valores por defecto
id_paper	PK	unsigned Int(10)	Not null	No
nombre_paper		Varchar(60)	Not null	No
autor		Varchar(60)	Not null	No
precio		unsigned mediumint(6)	Not null	No
fecha_lan		Date	Not null	No

orden

Nombre atributo	PK/ FK	Tipo de dato	Null/ Not Null	Valores por defecto
id_orden	PK	Int (10) unsigned	Not null	No
monto		Int (7) unsigned	Not null	No
estado		ENUM('entregada','no entregada')	Null	no entregada
fecha		Date	Null	No
id_cli	FK	mediumint(6) unsigned	Not Null	No
id_paper	FK	unsigned Int(10)	Not null	No

adm_pap

Nombre atributo	PK/ FK	Tipo de dato	Null/ Not Null	Valores por defecto
id_paper	PK/ FK	Int(10) unsigned	Not null	No
rut_adm	PK/ FK	Int(9) unsigned	Not null	No

Consultas etapa 1

2 alter

alter table cliente add fechadenacimiento date;
alter table cliente drop fechadenacimiento;

3 select (con join)

papers en una orden
select * from paper join orden where paper.id_paper = orden.id_paper;

π id_paper,nombre_paper,autor,precio,fecha_lan,id_orden,monto,fecha,id_cli ((paper) ⋈ paper.id_paper = orden.id_paper (orden))

orden de un cliente
select * from cliente join orden where cliente.id_cli = orden.id_cli;

π cliente.id_cli,cliente.rut_cli,cliente.nombre_cli,orden.id_orden,orden.monto,orden.estado,orden.fecha ((cliente) ⋈ cliente.id_cli = orden.id_cli (orden))

papers de un cliente

select * from cliente join orden join paper where cliente.id_cli=orden.id_cli and orden.id_paper=paper.id_paper;
π cliente.id_cli,cliente.rut_cli,cliente.nombre_cli,orden.id_orden,orden.monto,orden.estado,orden.fecha,paper.id_paper,paper.nombre_paper,paper.autor,paper.precio,paper.fecha_lan (((cliente) ⋈ cliente.id_cli = orden.id_cli (orden))⋈ orden.id_paper=paper.id_paper (paper))

2 update

update cliente set nombre_cli = “Claudio Yang” where nombre_cli = “Kevyn Bradshaw”;
update administrador set nombre_adm = “Andres Gonzalez” where nombre_adm =”Amir Phelps“;

Universidad de Valparaíso CHILE

3 insert

insert into cliente(id_cli, rut_cli, nombre_cli) values (123465, 12345678-9, “Rodrigo Maturana”);

insert into orden (id_orden, monto, estado, fecha, id_cli, id_paper) values(243581324, 10000,"no entregada", “20/04/01”, 123465,45471119);

insert into paper(id_paper, nombre_paper, autor, precio, fecha_lan) values (98039045, “investigación cosmética”, “alis torres”, 10000, “20/04/01”);

2 delete

delete from cliente where nombre_cli = “Fritz Walls”;
delete from administrador where nombre_adm = “Beck Morin”;

drop

drop table cliente;

Universidad de Valparaíso CHILE

Consultas etapa 2

Subconsultas

- monto medio de las órdenes donde la id_cli sea igual en ambas tablas

```
SELECT orden.id_orden, orden.monto, cliente.nombre_cli, (SELECT AVG(monto) FROM
orden)
from orden inner join cliente
on orden.id_cli = cliente.id_cli;
```

- selecciona el paper solo de un autor.

```
SELECT paper.nombre_paper, paper.autor,paper.precio, orden.fecha
from paper inner join orden
on paper.id_paper = orden.id_paper
where autor = (select autor from paper where autor = 'Allen T. Cook');
```

π paper.nombre_paper, paper.autor, paper.precio, orden.fecha((paper) ⋈ paper.id_paper = orden.id_paper σ autor = (π autor σ autor = 'Allen T. Cook'(paper))(orden))

- suma total de todas las órdenes realizadas por clientes.

```
select cliente.rut_cli, cliente.nombre_cli, orden.id_orden, orden.fecha, orden.monto, (select
SUM(precio)
from paper) AS total_monto_ordenes
from cliente join orden
on cliente.id_cli = orden.id_cli;
```

Operadores

- selecciona los papers a los que se hicieron una orden y que estén entre 3000 y 6000 pesos

```
select orden.id_orden , orden.fecha, paper.precio
from orden join paper
on orden.id_paper = paper.id_paper
where paper.precio between 3000 and 6000;
```

 orden.id_orden , orden.fecha, paper.precio((orden) ⋈ orden.id_paper = paper.id_paper σ paper.precio > 3000 ∧ paper.precio < 6000 (paper))

- selecciona los papers a los que se hicieron una orden y que no estén entre 3000 y 6000 pesos

```
select orden.id_orden , orden.fecha, paper.precio
from orden join paper
on orden.id_paper = paper.id_paper
where paper.precio not between 3000 and 6000;
```

 orden.id_orden , orden.fecha, paper.precio((orden) ⋈ orden.id_paper = paper.id_paper σ paper.precio < 3000 ∧ paper.precio > 6000 (paper))

Función

- Mostrar la ganancia total de un autor determinado

```
select orden.id_orden, paper.nombre_paper, paper.autor, paper.precio, sum(precio) AS
Total_ganacia_autor
from orden join paper
on orden.id_paper = paper.id_paper
where paper.autor = 'Cadman C. Vazquez';
```

Group by

- mostrar el conteo de cuántos papers de cada uno se han vendido

```
select paper.nombre_paper, paper.autor, count(paper.id_paper)as total
from cliente join orden join paper
where cliente.id_cli=orden.id_cli and orden.id_paper=paper.id_paper
group by paper.id_paper;
```

π paper.nombre_paper, paper.autor(((cliente) ⋈ cliente.id_cli = orden.id_cli (orden))⋈ orden.id_paper=paper.id_paper (paper)) total ← $G_{count(paper.id\_paper)}$ (paper.id_paper)

- Mostrar la ganancia en dinero por paper

```
select paper.nombre_paper, paper.autor, sum(paper.precio)as total
from cliente join orden join paper
where cliente.id_cli=orden.id_cli and orden.id_paper=paper.id_paper
group by  paper.id_paper;
```

π paper.nombre_paper, paper.autor(((cliente) ⋈ cliente.id_cli = orden.id_cli (orden))⋈ orden.id_paper=paper.id_paper (paper)) total ← $G_{sum(paper.precio)}$ (paper.id_paper)

- Mostrar la cantidad de papers que ha comprado un cliente

```
select cliente.nombre_cli, count(paper.nombre_paper)as total
```

from cliente join orden join paper
where cliente.id_cli=orden.id_cli and orden.id_paper=paper.id_paper
group by cliente.id_cli;

π cliente.nombre_cli(((cliente) ⋈ cliente.id_cli = orden.id_cli (orden))⋈ orden.id_paper=paper.id_paper (paper)) total ← $G_{count(paper.nombre\_paper)}$ (cliente.id_cli)

Schema

- Mostrar usuarios que tienen permiso de insertar , borrar o crear datos

select Host, Db, User, insert_priv, delete_priv, create_priv
from mysql.db
where Db='papers' and insert_priv='Y' and create_priv='Y' and delete_priv='Y'

- Mostrar los permisos de los usuarios en la base de datos papers

select *
from mysql.db
where Db='papers'