

Documento Descripción del Proyecto

Etapa III: Biblioteca de Papers Online

Integrantes: Manuel Barrera
Mauricio Moragra
Silvio Vera

Profesora: Eliana Providei
Manuel García

Fecha: 06 de diciembre de 2020

Índice

Introducción	4
Descripción problema/Solución	5
Requerimientos	5
Funcionales	5
No Funcionales	6
Diagrama de casos de uso	7
Diagrama de secuencia	8
Agregar paper	8
Mockups Iniciales	9
Vista Principal	9
Vista Papers	10
Mockups Finales	11
Vista Principal	11
Vista Papers	12
Modelo entidad relación Etapa 1	13
Modelo entidad relación Final	14
Modelo relacional	15
Supuestos	15
Diccionario de datos	16
administrador	16
cliente	16
paper	16
orden	17
ord_pap	17
adm_pap	17
Consultas etapa 1	18
2 alter	18
3 select (con join)	18
2 update	19
3 insert	19
2 delete	19
drop	19
Consultas etapa 2	19
Subconsultas	19

Operadores	21
Función	22
Group by	22
Schema	23
Consultas etapa 3	24
Conclusión	27

Introducción

A lo largo de las tres entregas se ha ido modelando la solución generada para el problema planteado, se ha ido avanzando paso a paso a través del diseño de una base de datos junto con una página en la cual en un principio solo estaba para mostrar información para el cliente y administrador. Evolucionando con la segunda entrega se agregó funcionalidad para que el administrador pudiera administrar todo lo relacionado con los papers y base de datos, logrando este poder ingresar, modificar o eliminar papers de una manera sencilla, además de poder visualizar las ventas de los diferentes papers y poder ver las ganancias generadas por papers, además de tener un inicio de sesión para el administrador logrando proteger los datos que el usuario no puede visualizar y manipular. Pero para esta última entrega la página diseñada es completamente funcional, solucionando el problema de seguridad que se surge al crear un inicio de sesión para el cliente y agregando un carrito de compras donde con el que el cliente puede interactuar, en el cual se puede agregar un producto, eliminar 1 o vaciar el carrito de compras, además de mejorar el diseño de la página para la parte del cliente haciéndola más agradable para el uso de esta sección.

Descripción problema/Solución

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
 - Ambas partes cuentan con un respectivo login en el cual podrán iniciar sesión, el cliente no podrá acceder a la parte del administrador y el administrador no podrá acceder a las sección del cliente.
- El cliente podrá realizar las siguientes acciones:
 - Comprar papers.
 - El cliente una vez inicie sesión podrá ver en catálogo un botón para poder comprar papers además de ver un carrito en el cual se añadirán.
 - Reservar papers.
 - Al momento de comprar un paper el estado de este estará como no entregado, una vez este se le envíe al cliente se cambiara de estado a Entregado.
- El Administrador podrá realizar las siguientes acciones:
 - Agregar papers.
 - El administrador podrá ver con una sesión de administrador iniciada un formulario en el cual podrá completar para añadir un paper
 - Quitar papers.
 - En esa vista el administrador podrá visualizar un botón que dice 'eliminar' el cual al presionar eliminar el papel que contenga dicho botón
 - Revisar número de ventas y reservas.

- Podrá ver en la barra de navegación un botón que dice 'Vendido' y 'Ganancias' el cual se podrá visualizar la parte de cuanto se ha vendido un paper y cuanta ganancia ha generado.
- Modificar papers
 - Podrá ver en la barra de navegación un botón que dice 'modificar' el cual redireccionará a un formulario que al completarlo podrá modificar dicho paper indicando el Id del paper.
- 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

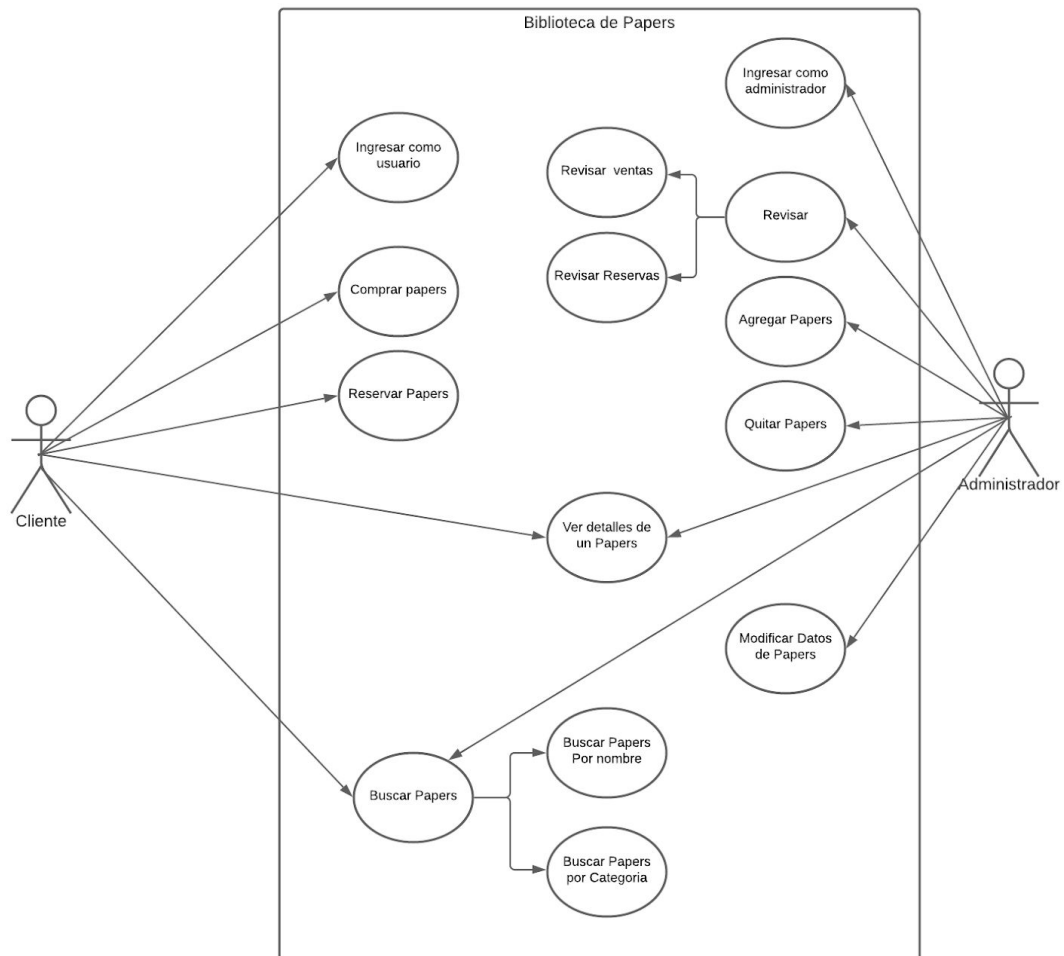


Figura 1: En el siguiente diagrama se representa la forma en que actúa el administrador y el cliente dentro de nuestra app web.

Diagrama de secuencia

Agregar paper

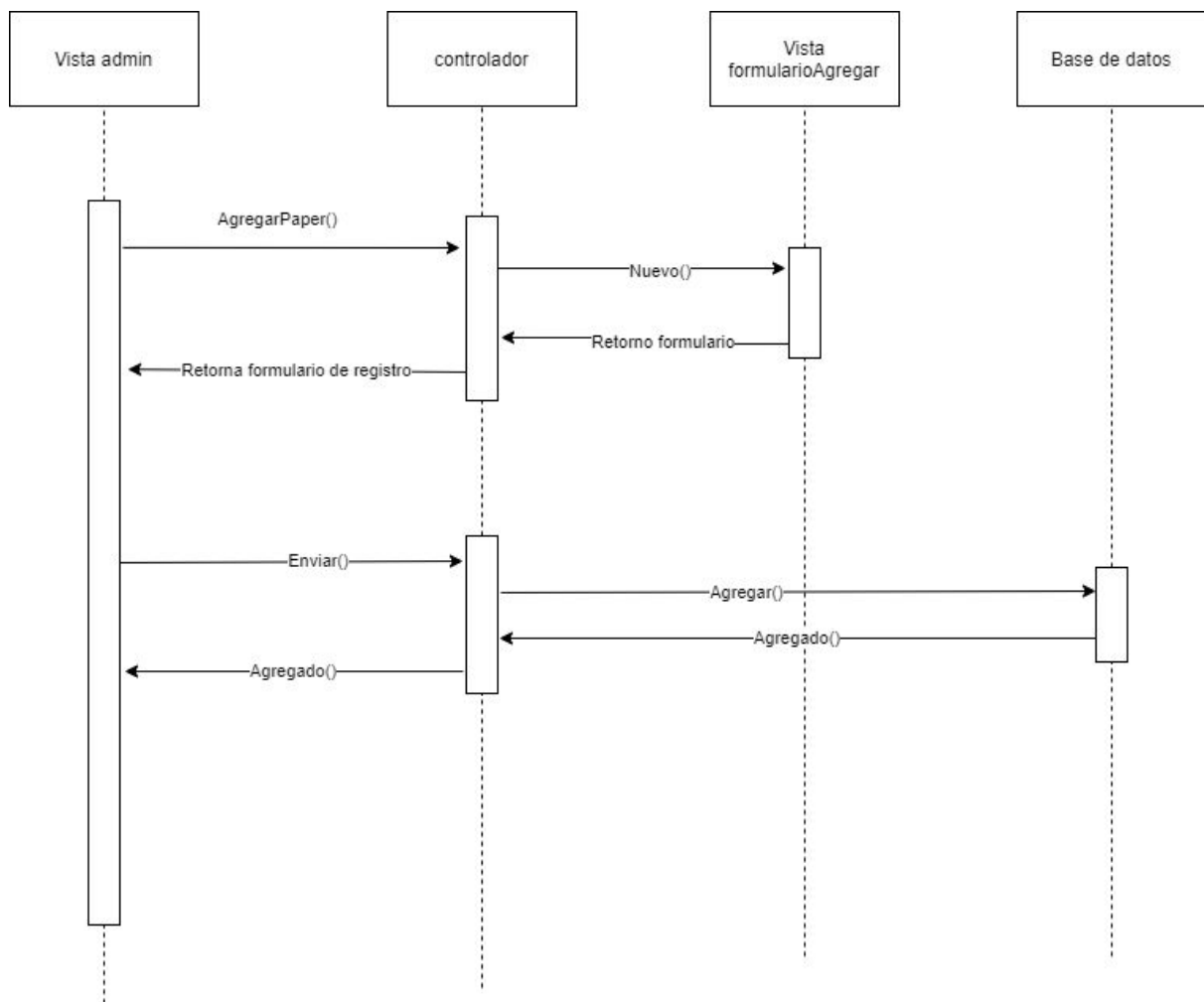


Figura 2: Transacciones entre la vista de administrador y vista de formulario donde se ingresaran los datos, donde el controlador realizara las transacciones entre ambas vistas, y luego envía los datos al SGBD.

Mockups Iniciales

Vista Principal

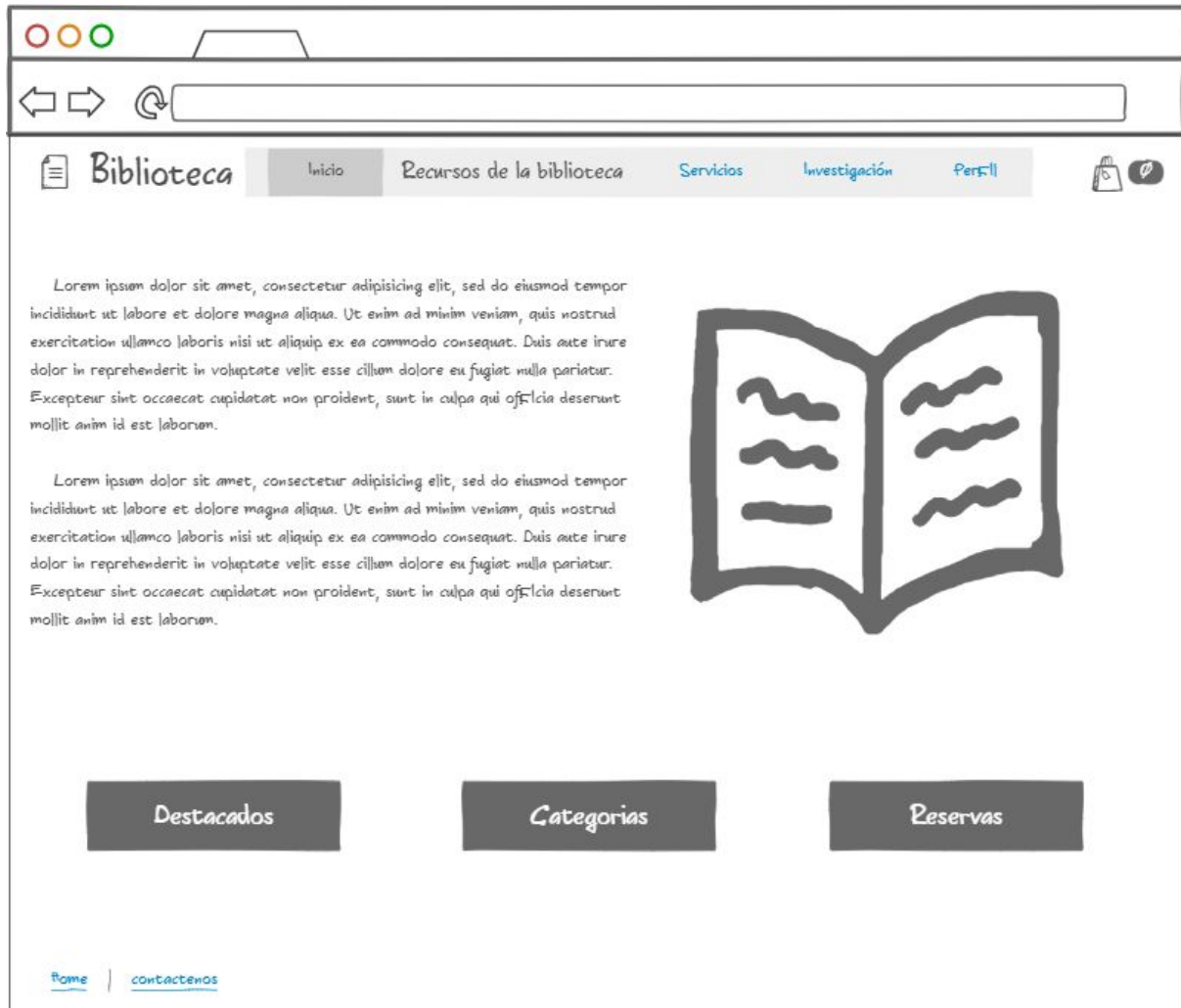


Figura 3: La vista actual es la que se le muestra al usuario al entrar a la página, en esta te puedes mover entre diferentes vistas.

Vista Papers

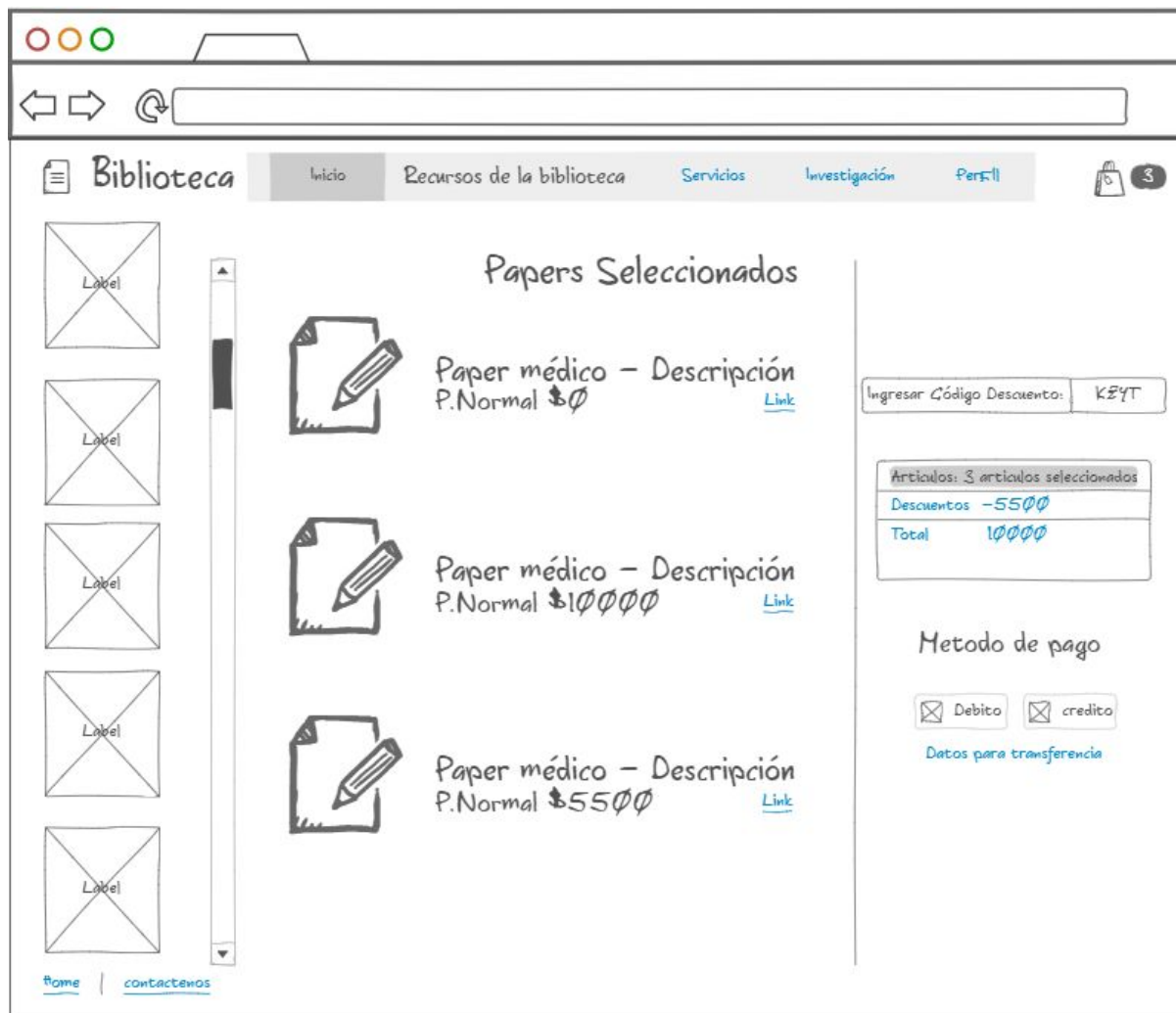
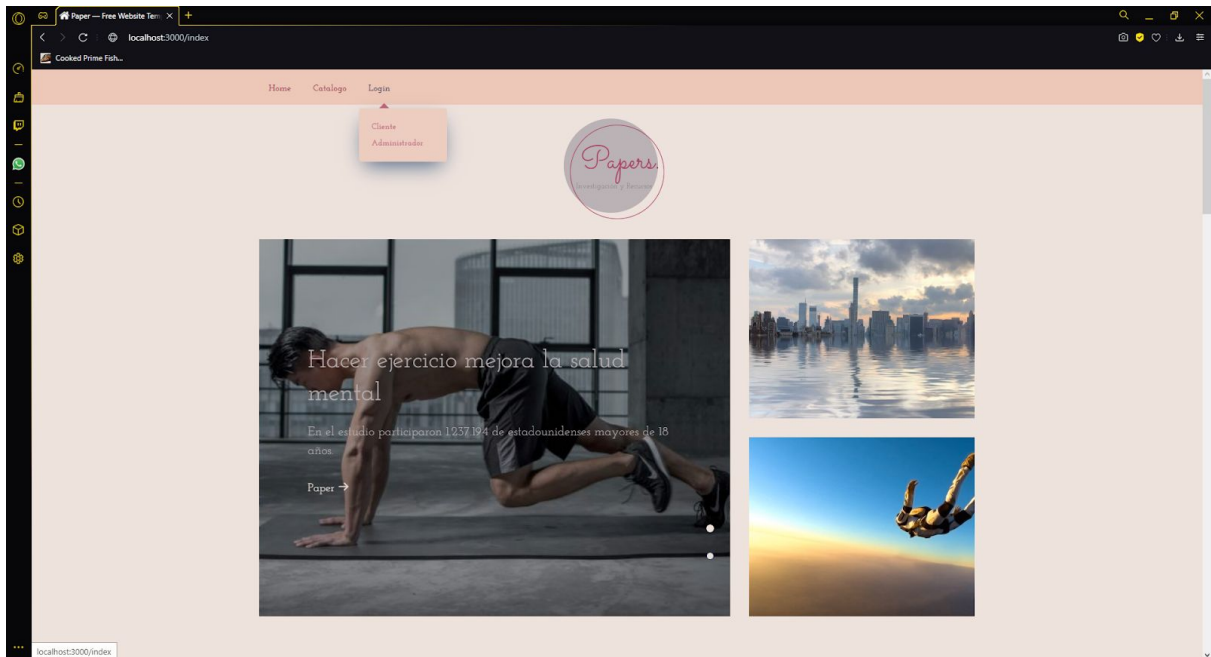


Figura 4: Corresponde a la vista del usuario al seleccionar los papers, es el carrito de venta y muestra el monto total a pagar por los papers, al monto total se le puede descontar con un código de descuento.

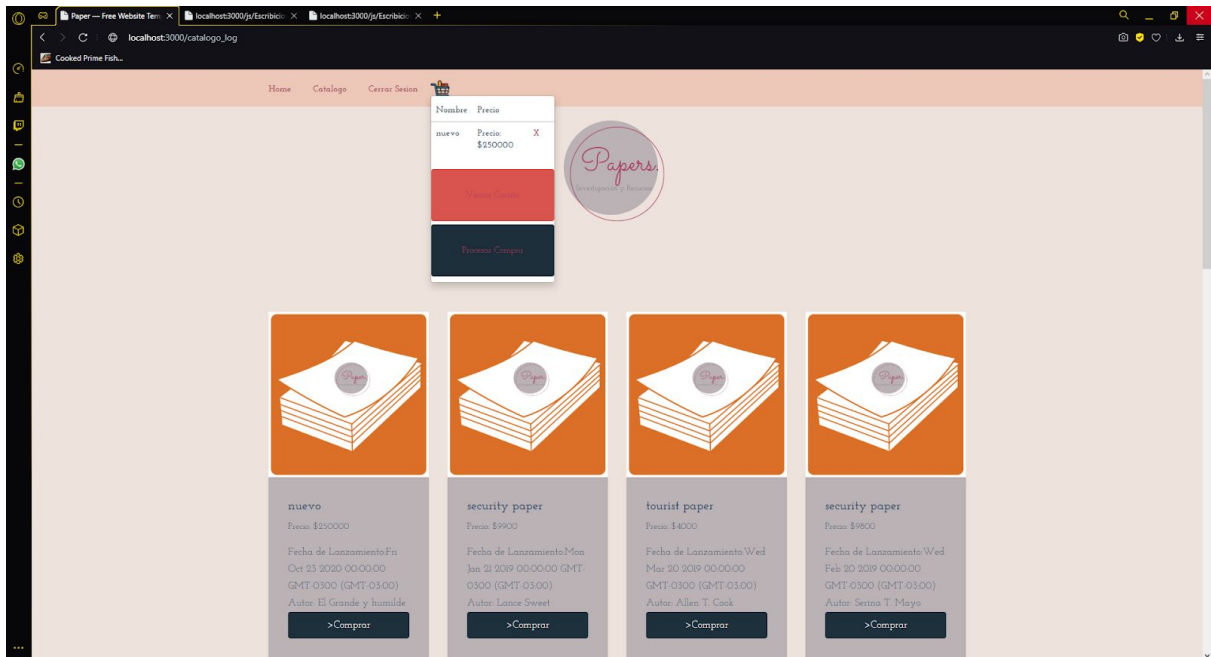
Mockups Finales

Vista Principal



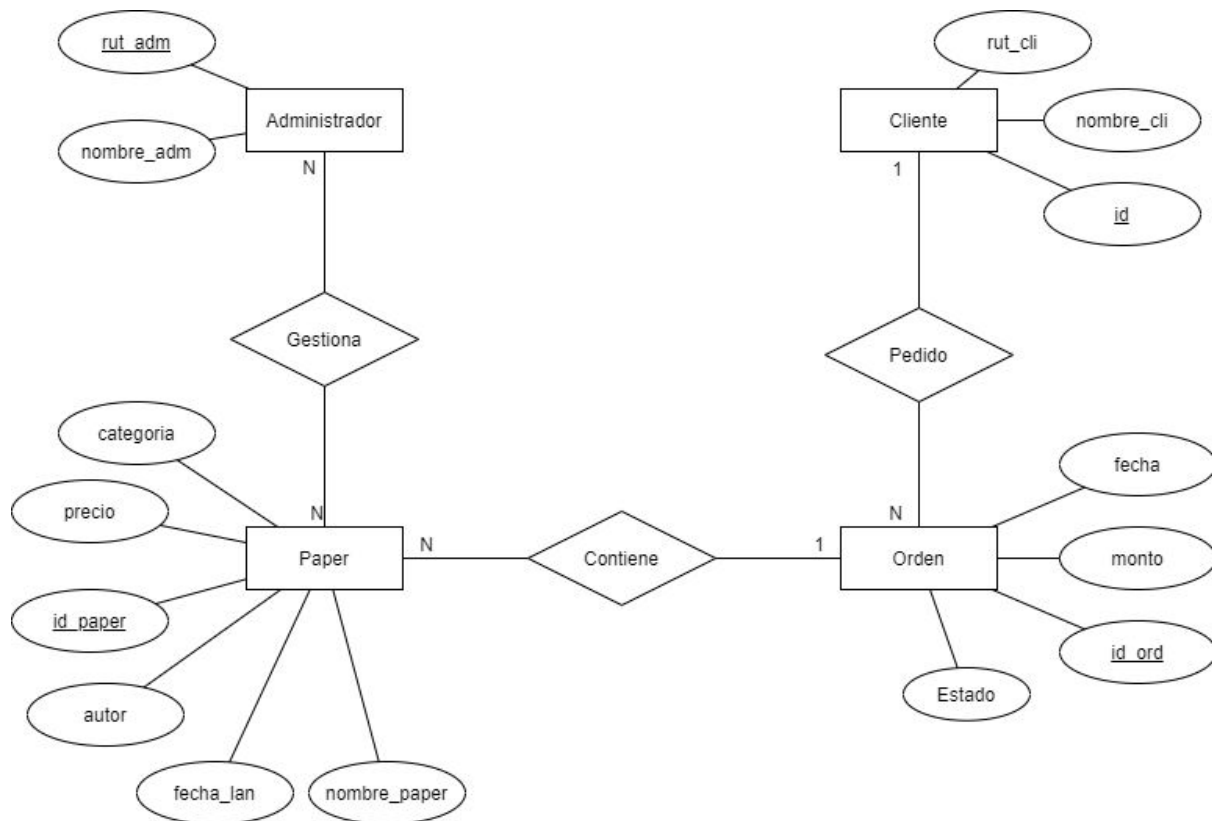
La vista principal no cambió a grandes rasgos, exceptuando algunas secciones como la parte de categorías, destacados y reservas ya que estas ideas fueron descartadas por tema de estar limitados por el tiempo de entrega dejando un catálogo donde se pueden visualizar todos lo papers un inicio de sesión tanto para cliente y administrador.

Vista Papers



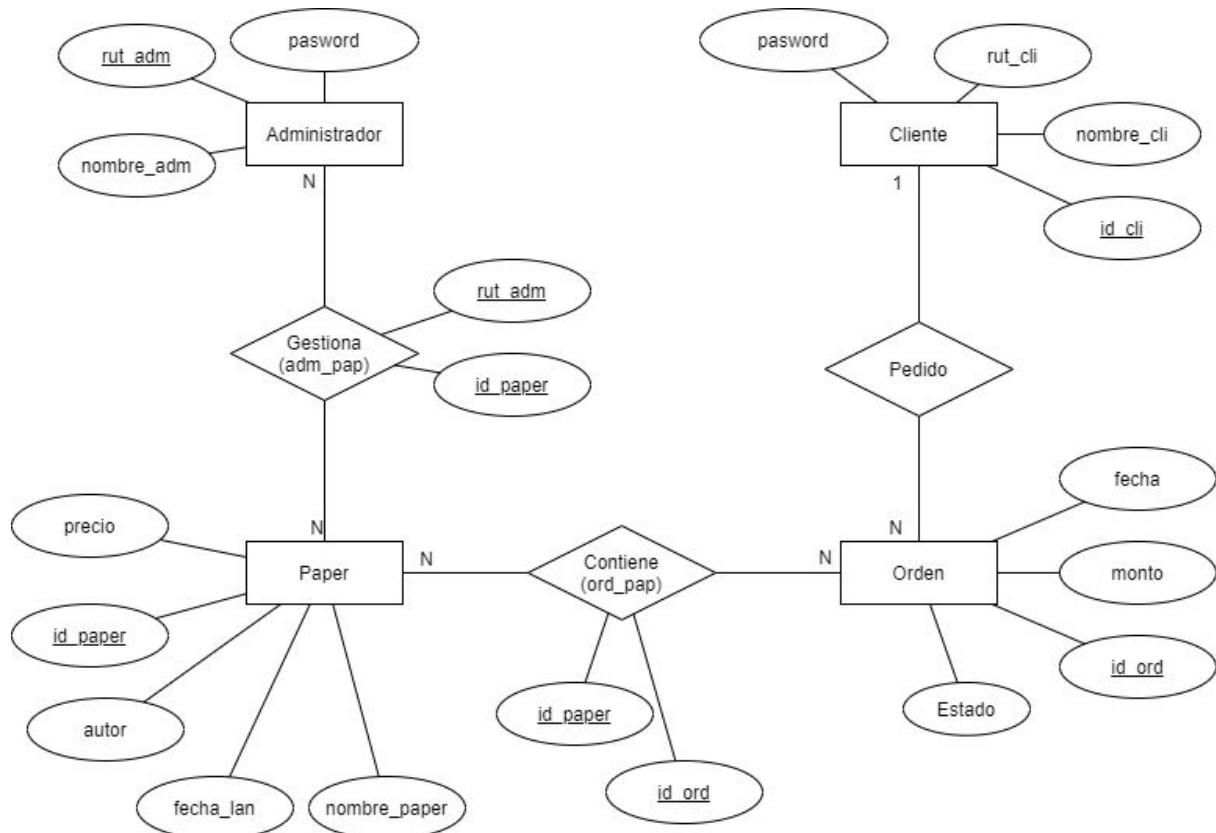
En cuanto a la vista del cliente de paper para comprarlos se elimino la seccion de la izquierda que se planteó en un inicio ya que no se tenía nada que agregar en esa sección, en cuanto a cómo visualizar los paper se tomó la decisión de mostrar los paper por columnas de 4 papers ya que se considera que queda más cómodo para el cliente. En cuanto al carrito se colocó en la sección de la barra de navegación para que se centre en visualizar los paper.

Modelo entidad relación Etapa 1



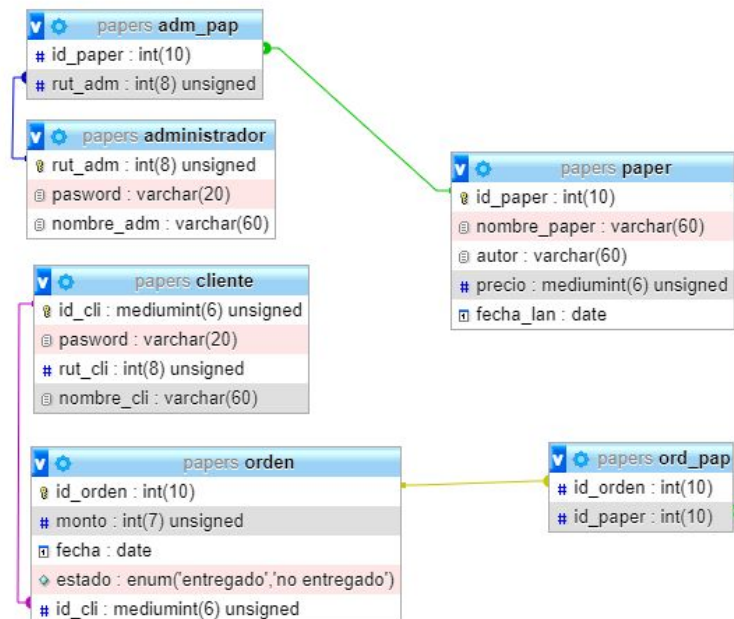
Modelo entidad relacion etapa N°1

Modelo entidad relación Final



Modelo entidad relación etapa N°1: tras diversas correcciones el modelo final quedó con dos tablas intermedias entre administrador y papel ya que es una relación N:N, además de adaptar el modelo ya que entre piper y orden se consideró para esta última entrega una relación de N:N. Para ambos usuarios(cliente y administrador) en un principio no se considero una distinción de usuarios por lo tanto se agrego un atributo para ambos denominado 'password' con la cual cada usuario podrá iniciar sesion y se eliminó el atributo 'categoría' de paper ya que en un principio se tenía pensado separa los paper por categoría, esta idea se desechó ya que se iban a repetir demasiados datos

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.

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
password		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
password		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	Int(10) auto_increment	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	Null	No

orden

Nombre atributo	PK/ FK	Tipo de dato	Null/ Not Null	Valores por defecto
id_orden	PK	Int (10) auto_increment	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

ord_pap

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

adm_pap

Nombre atributo	PK/ FK	Tipo de dato	Null/ Not Null	Valores por defecto
id_paper	FK	Int(10)	Not null	No
rut_adm	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 paper.id_paper, paper.nombre_paper, paper.autor, paper.precio, orden.id_orden,  
orden.fecha  
from paper join orden join ord_pap  
on paper.id_paper = ord_pap.id_paper and orden.id_orden=ord_pap.id_orden;
```

```
π paper.id_paper, paper.nombre_paper, paper.autor, paper.precio, orden.id_orden  
orden.fecha(((paper) ⋈ paper.id_paper = orden.id_paper (orden)) ⋈ orden.id_orden =  
ord_pap.id_orden (ord_pap))
```

- orden de un cliente

```
select * from cliente join orden where cliente.id_cli = orden.id_cli;
```

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

- papers de un cliente

```
select cliente.id_cli, cliente.rut_cli, cliente.nombre_cli, paper.id_paper, paper.nombre_paper,  
paper.autor, paper.precio  
from cliente inner join orden inner join paper inner join ord_pap  
on cliente.id_cli=orden.id_cli and paper.id_paper = ord_pap.id_paper and  
orden.id_orden=ord_pap.id_orden;
```

```
π cliente.id_cli, cliente.rut_cli, cliente.nombre_cli, paper.id_paper, paper.nombre_paper,  
paper.autor, paper.precio((((cliente) ⋈ cliente.id_cli = orden.id_cli (orden))⋈ paper.id_paper  
= ord_pap.id_paper (paper))) ⋈ orden.id_orden=ord_pap.id_orden (ord_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";

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) values(243581324, 10000,"no entregada", "20/04/01", 123465);

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;

Consultas etapa 2

Subconsultas

- La siguiente consulta muestra el promedio de ganancia al sumar todas las ordenes hechas por el usuario

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

- La siguiente consulta muestra las ganancias totales de un solo autor.

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

π paper.nombre_paper, paper.autor, paper.precio, orden.fecha (((paper) \bowtie paper.id_paper
=ord_pap.id_paper(orden) \bowtie orden.id_orden = ord_pap.id_orden, σ autor = (π autor σ
autor = 'Allen T. Cook'(paper))(ord_pap)))

- La siguiente consulta muestro todas las compras realizadas por usuarios y su número de orden, finalmente se saca el total de todas las ganancias sumando todos los montos de las compras.

```
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 inner join paper inner join ord_pap
on paper.id_paper =ord_pap.id_paper and orden.id_orden = ord_pap.id_orden
where paper.precio between 3000 and 6000;
```

```
 $\pi$  orden.id_orden , orden.fecha, paper.precio(((orden)  $\bowtie$  paper.id_paper =ord_pap.id_paper
(paper))  $\bowtie$  orden.id_orden = ord_pap.id_orden  $\sigma$  paper.precio > 3000  $\wedge$  paper.precio <
6000 (paper))
```

- La siguiente consulta selecciona los papers con una orden y que tengan como id_orden mayor a 537044333

```
select orden.id_orden , orden.fecha, paper.precio
from orden inner join paper inner join ord_pap
on paper.id_paper =ord_pap.id_paper and orden.id_orden = ord_pap.id_orden
where orden.id_orden in(select id_orden from orden where id_orden > 537044333);
```

```
 $\pi$  orden.id_orden , orden.fecha, paper.precio(((orden)  $\bowtie$  paper.id_paper =ord_pap.id_paper
(paper))  $\bowtie$  orden.id_orden = ord_pap.id_orden  $\sigma$  orden.id_orden in( $\pi$  id_orden  $\sigma$  id_orden >
537044333(orden))(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 inner join paper inner join ord_pap
on paper.id_paper =ord_pap.id_paper and orden.id_orden = ord_pap.id_orden
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 orden inner join paper inner join ord_pap
on paper.id_paper =ord_pap.id_paper and orden.id_orden = ord_pap.id_orden
group by paper.id_paper;
```

$$\pi \text{ paper.nombre_paper, paper.autor}(((\text{paper}) \bowtie \text{paper.id_paper} = \text{ord_pap.id_paper} \\ (\text{ord_pap})) \bowtie \text{ord_pap.id_orden} = \text{orden.id_orden} (\text{orden})) \text{ total} \leftarrow G_{count(\text{paper.id_paper})} \\ (\text{paper.id_paper})$$

- Mostrar la ganancia en dinero por paper

```
select paper.nombre_paper, paper.autor, sum(paper.precio)as total
from orden inner join paper inner join ord_pap
on paper.id_paper =ord_pap.id_paper and orden.id_orden = ord_pap.id_orden
group by paper.id_paper;
```

$$\pi \text{ paper.nombre_paper, paper.autor}(((\text{paper}) \bowtie \text{paper.id_paper} = \text{ord_pap.id_paper} \\ (\text{ord_pap})) \bowtie \text{ord_pap.id_orden} = \text{orden.id_orden} (\text{orden})) \text{ total} \leftarrow G_{sum(\text{paper.precio})} \\ (\text{paper.id_paper})$$

- Mostrar la cantidad de papers que ha comprado un cliente

```
select cliente.nombre_cli, count(paper.nombre_paper) as total
from cliente inner join orden inner join paper inner join ord_pap
on cliente.id_cli=orden.id_cli and paper.id_paper=ord_pap.id_paper and orden.id_orden =
ord_pap.id_orden
group by cliente.id_cli;
```

$$\pi \text{ cliente.nombre_cli } (((\text{cliente}) \bowtie \text{cliente.id_cli} = \text{orden.id_cli} (\text{orden})) \bowtie \text{orden.id_orden} = \text{ord_pap.id_orden} (\text{ord_pap}) \bowtie \text{ord_pap.id_paper} = \text{paper.id_paper} (\text{paper}))$$

$$\text{total} \leftarrow G_{\text{count}(\text{paper.nombre_paper})} (\text{cliente.id_cli})$$

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'
```

Consultas etapa 3

Simbología:

Subconsultas

Operadores

Funciones

Group by

Schema:(3)

- Obtener el paper de menor precio con una orden hecha por el usuario (Mostrar el paper comprado con el menor precio)

```
select paper.id_paper, paper.nombre_paper, paper.autor, paper.precio, orden.id_orden,
orden.fecha
from paper join orden join ord_pap
on paper.id_paper = ord_pap.id_paper and orden.id_orden=ord_pap.id_orden
where precio = (select min(precio) from paper);
```

```
 $\pi$  paper.id_paper, paper.nombre_paper, paper.autor, paper.precio, orden.id_orden,
orden.fecha (((paper)  $\bowtie$  paper.id_paper = ord_pap.id_paper (orden))  $\bowtie$ 
orden.id_orden=ord_pap.id_orden  $\sigma$  precio = ( $\pi$  min(precio) (paper))(ord_pap))
```

- Contar cuántas órdenes existen de distintos papers

```
select paper.id_paper, paper.nombre_paper, paper.autor, paper.precio, orden.id_orden,
orden.fecha, count(ord_pap.id_orden) As total_de_ordenes
from paper join orden join ord_pap
on paper.id_paper = ord_pap.id_paper and orden.id_orden=ord_pap.id_orden;
```

```
 $\pi$  paper.id_paper, paper.nombre_paper, paper.autor, paper.precio, orden.id_orden,
orden.fecha, count(ord_pap.id_orden) (((paper)  $\bowtie$  paper.id_paper = ord_pap.id_paper
(orden))  $\bowtie$  orden.id_orden=ord_pap.id_orden (ord_pap))
```

- mostrar las compra de los clientes entre un rango de fechas determinadas

```
select orden.id_orden, orden.fecha , cliente.id_cli, cliente.rut_cli, cliente.nombre_cli,
paper.id_paper, paper.nombre_paper, paper.autor, paper.precio
from cliente inner join orden inner join paper inner join ord_pap
on cliente.id_cli=orden.id_cli and paper.id_paper = ord_pap.id_paper and
orden.id_orden=ord_pap.id_orden and orden.fecha between "2020-10-01" and "2020-11-30"
```

π

orden.id_orden,orden.fecha,cliente.id_cli,cliente.rut_cli,cliente.nombre_cli,paper.id_paper,or

den.paper,nombre_paper, paper.autor, paper.precio (((cliente) \bowtie cliente.id_cli = orden.id_cli
(orden)) \bowtie paper.id_paper=ord_pap.id_paper (paper) \bowtie
orden.id_orden=ord_pap.id_orden(ord_pap) σ orden.fecha > 2020-10-01 \wedge orden.fecha <
2020-11-30 (orden)))

- mostrar quienes obtuvieron el paper de mayor precio

```
select cliente.nombre_cli, paper.id_paper, paper.nombre_paper, paper.autor, paper.precio,
orden.id_orden, orden.fecha
from paper join orden join ord_pap join cliente
on paper.id_paper = ord_pap.id_paper and orden.id_orden=ord_pap.id_orden and
orden.id_cli= cliente.id_cli
where precio = (select max(precio) from paper);
```

π paper.id_paper, paper.nombre_paper, paper.autor, paper.precio, orden.id_orden,
orden.fecha (((paper) \bowtie paper.id_paper = ord_pap.id_paper (orden)) \bowtie
orden.id_orden=ord_pap.id_orden(ord_pap)) \bowtie orden.id_cli= cliente.id_cli σ precio = (π
max(precio) (paper))(cliente)

- mostrar el conteo de cuántos papers de cada uno se han vendido en un rango de fechas

```
select paper.nombre_paper, paper.autor, count(paper.id_paper)as total
from orden inner join paper inner join ord_pap
on paper.id_paper =ord_pap.id_paper and orden.id_orden = ord_pap.id_orden and
orden.fecha between "2020-11-01" and "2020-11-30"
group by paper.id_paper;
```

π paper.nombre_paper, paper.autor(((paper) \bowtie paper.id_paper = ord_pap.id_paper
(ord_pap)) \bowtie ord_pap.id_orden=orden.id_orden (orden) σ orden.fecha > 2020-11-01 \wedge
orden.fecha < 2020-11-30 (orden)) total $\leftarrow G_{count(paper.id_paper)}(paper.id_paper)$

- mostrar las ganancias que ha generado cada autor

```
select paper.autor, sum(precio) AS Total_ganacia_autor
from orden inner join paper inner join ord_pap
on paper.id_paper =ord_pap.id_paper and orden.id_orden = ord_pap.id_orden
group by paper.autor;
```

π paper.autor(((paper) \bowtie paper.id_paper = ord_pap.id_paper (ord_pap)) \bowtie
ord_pap.id_orden=orden.id_orden (orden)) Total_ganacia_autor $\leftarrow G_{sum(paper.precio)}$
(paper.autor)

- mostrar los ingresos por fecha

```
select orden.fecha, sum(paper.precio) as total
from orden inner join paper inner join ord_pap
on paper.id_paper = ord_pap.id_paper and orden.id_orden = ord_pap.id_orden
group by orden.fecha;

 $\pi$  orden.fecha(((paper)  $\bowtie$  paper.id_paper = ord_pap.id_paper (ord_pap))  $\bowtie$ 
ord_pap.id_orden = orden.id_orden (orden)) total  $\leftarrow G_{sum(paper.precio)}$ 
(orden.fecha)
```

- mostrar la cantidad de papers vendidos por día , agrupados por fecha

```
select orden.fecha, count(orden.fecha) as total
from orden inner join paper inner join ord_pap
on paper.id_paper = ord_pap.id_paper and orden.id_orden = ord_pap.id_orden
group by orden.fecha;
```

```
 $\pi$  orden.fecha(((paper)  $\bowtie$  paper.id_paper = ord_pap.id_paper (ord_pap))  $\bowtie$ 
ord_pap.id_orden = orden.id_orden (orden)) total  $\leftarrow G_{count(orden.fecha)}$ 
(orden.fecha)
```

- En la siguiente consulta obtenemos las órdenes de los clientes ordenadas por el monto menor a 4000

```
select orden.id_orden, orden.fecha, orden.monto, cliente.id_cli
from orden inner join cliente
on cliente.id_cli = orden.id_cli
where id_orden in(select id_orden from orden where orden.monto < 4000);
```

```
 $\pi$  orden.id_orden, orden.fecha, orden.monto, cliente.id_cli ((orden)  $\bowtie$ 
cliente.id_cli = orden.id_cli  $\sigma$  id_orden in( $\pi$  id_orden  $\sigma$  orden.monto < 4000(orden))(cliente))
```

- mostrar en qué fecha se tuvo menos ingresos

```
select orden.fecha, sum(paper.precio) as total
from orden inner join paper inner join ord_pap
on paper.id_paper = ord_pap.id_paper and orden.id_orden = ord_pap.id_orden
where precio = (select min(precio) from paper)
group by orden.fecha;
```

```
 $\pi$  orden.fecha (((orden)  $\bowtie$  paper.id_paper = ord_pap.id_paper (paper))  $\bowtie$  orden.id_orden =
ord_pap.id_orden  $\sigma$  precio = ( $\pi$  min(precio) (paper))(ord_pap)) total  $\leftarrow G_{sum(paper.precio)}$ 
(orden.fecha)
```

Conclusión

A medida que se avanzaba en el proyecto el equipo de trabajo experimentó distintas dificultades tanto en el diseño como en el planteamiento de la solución por ende se tuvo que modificar tanto la base de datos como la página web y sus componentes, limitados por el tiempo no se pudo implementar algunos de los requerimientos funcionales que un primera instancia se plantearon como poder ver el detalle del paper, también se descartaron algunas ideas como el ocultar paper ya que esta funcionalidad no se considero muy esencial para el funcionamiento, se logró que al agregar compras al carrito se guardan en la localStorage de la página, lamentablemente no se logró recuperar esos datos para escribirlos en un txt.

En cuanto al trabajo del equipo, se trabajó eficientemente en ambas partes, por parte del diseño tanto como en la funcionalidad de la página.

En cuanto al trabajo realizado, el equipo está conforme y contento con la solución presentada ya que en el grupo si bien en algunos ramos anteriores se trabajó con desarrollo web ninguno tenía un gran dominio en este ámbito. Por lo tanto, este trabajo se considera una gran experiencia en el desarrollo web ya que en este se realizó una distinción de usuario con seguridad, y logrando distintas funcionalidades para ambos usuarios.