

Etapa 1

Bases de Datos y Programación Web.

Tema: Pedidos Minimarket

Integrantes: Angel Araya, Brenda Araya, Carlos Escobedo.

Introducción y contexto general:

Una marca de minimarket nos ha solicitado desarrollar un sistema para que se pueda agilizar el proceso de compra de productos, y de esta manera tener menos congestión de gente en el local, actualizarse al sistema de ventas en la pandemia, y de esta manera evitar posibles contagios de C19.

La marca necesita que el sistema logre añadir productos al stock para así poder mostrarlo al cliente, además que este tenga un carrito de compras .

Solución:

Para esta problemática se tiene como solución un sistema de inventario en el cual el administrador podrá añadir productos, eliminar productos, editarlos, agregar descuentos, modificar precios. También el sistema tendrá una vista del inventario en modo cliente, donde este visualizará los productos, podrá filtrarlos por tipo (lácteos, carnes, bebidas, alcohol, etc) y tendrá una barra de búsqueda para buscar un producto específico. Además se tendrá un carrito de compras donde el cliente podrá añadir productos o quitarlos, elegir el método de pago (débito y crédito), se mostrará el precio de cada producto y el total de todos estos. Finalizando todo el proceso luego del pago, se le entregará un comprobante al cliente (boletera) para así poder hacer el retiro de sus productos en la sucursal.

Web:

Requerimientos funcionales:

- El sistema debe almacenar datos básicos del cliente (nombre, teléfono, rut, dirección, correo electrónico).
- El sistema debe almacenar los productos comprados, las boletas, y la información de la compra.
- El sistema debe almacenar la dirección del minimarket.
- El sistema debe exigir que el usuario esté registrado para poder comprar.
- El sistema debe permitir el registro de usuarios.
- El sistema debe permitir agregar productos al inventario.
- El sistema debe permitir eliminar productos del inventario.
- El sistema debe permitir modificar precios de productos.
- El sistema debe permitir modificar la cantidad de productos disponibles.
- El sistema debe permitir agregar descuento a productos.
- El sistema debe tener una vista de administrador.
- El sistema debe tener una vista de cliente.
- El sistema debe tener un buscador de productos.
- El sistema debe mantener guardado el carrito de compras con los productos añadidos.
- El sistema debe permitir el modificar el nombre e imagen de los productos.

Requerimientos no funcionales:

- El sistema debe ser seguro para su uso, pues tiene que resguardar los datos personales y de compra del cliente.
- El sistema debe de soportar el uso de muchas personas simultáneamente (al menos 1000) ya que es una página pública y todos pueden entrar a verla sin necesidad de estar ingresado.

Diagrama de secuencia de la solución:

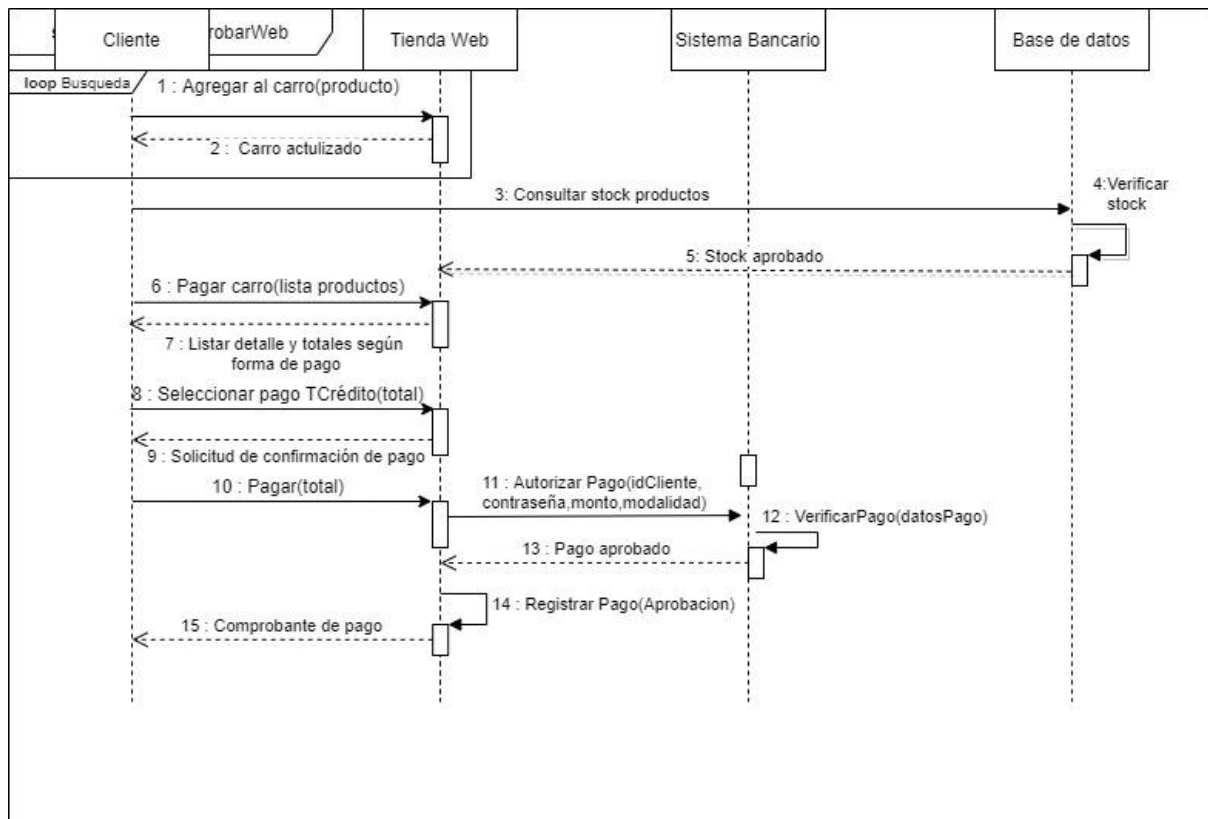
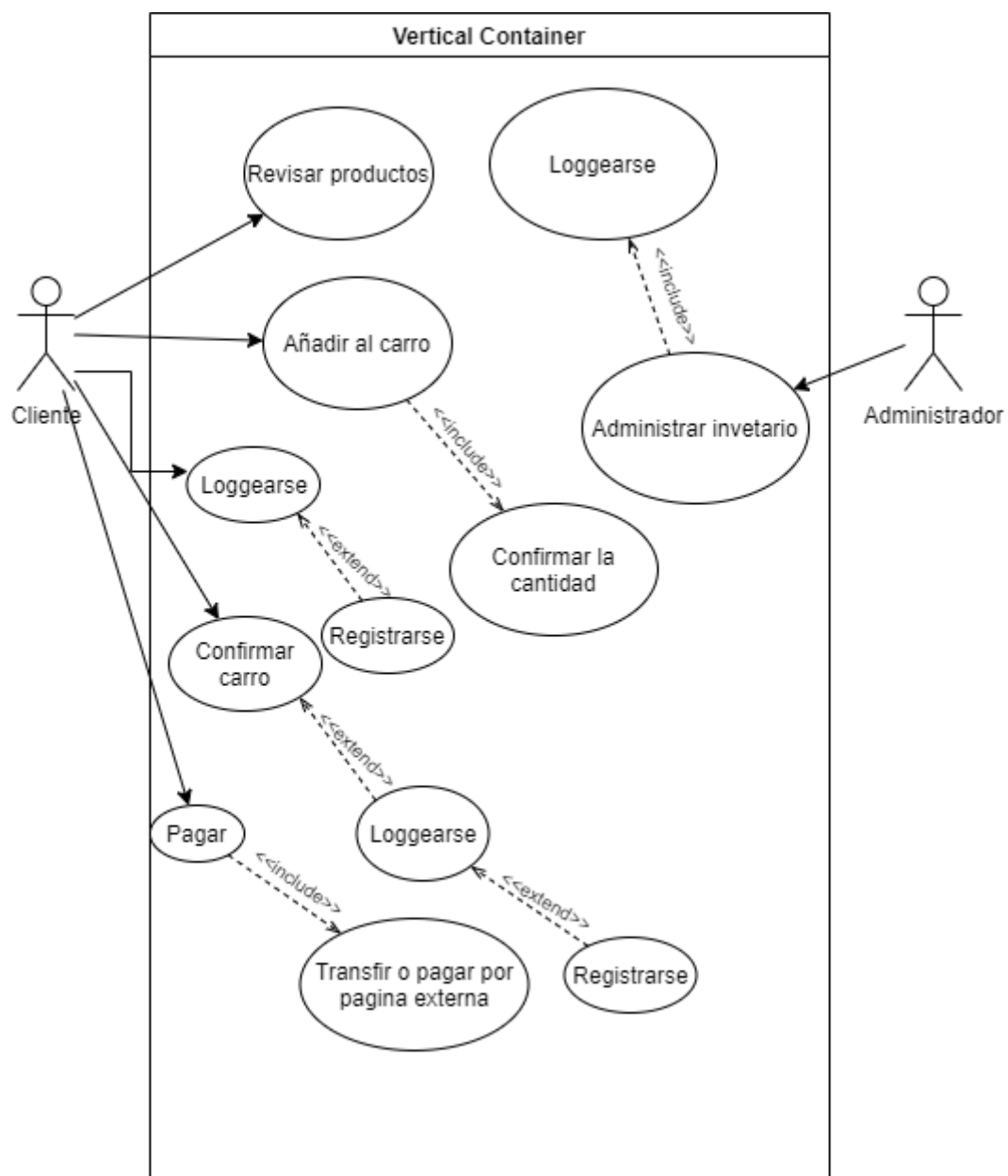
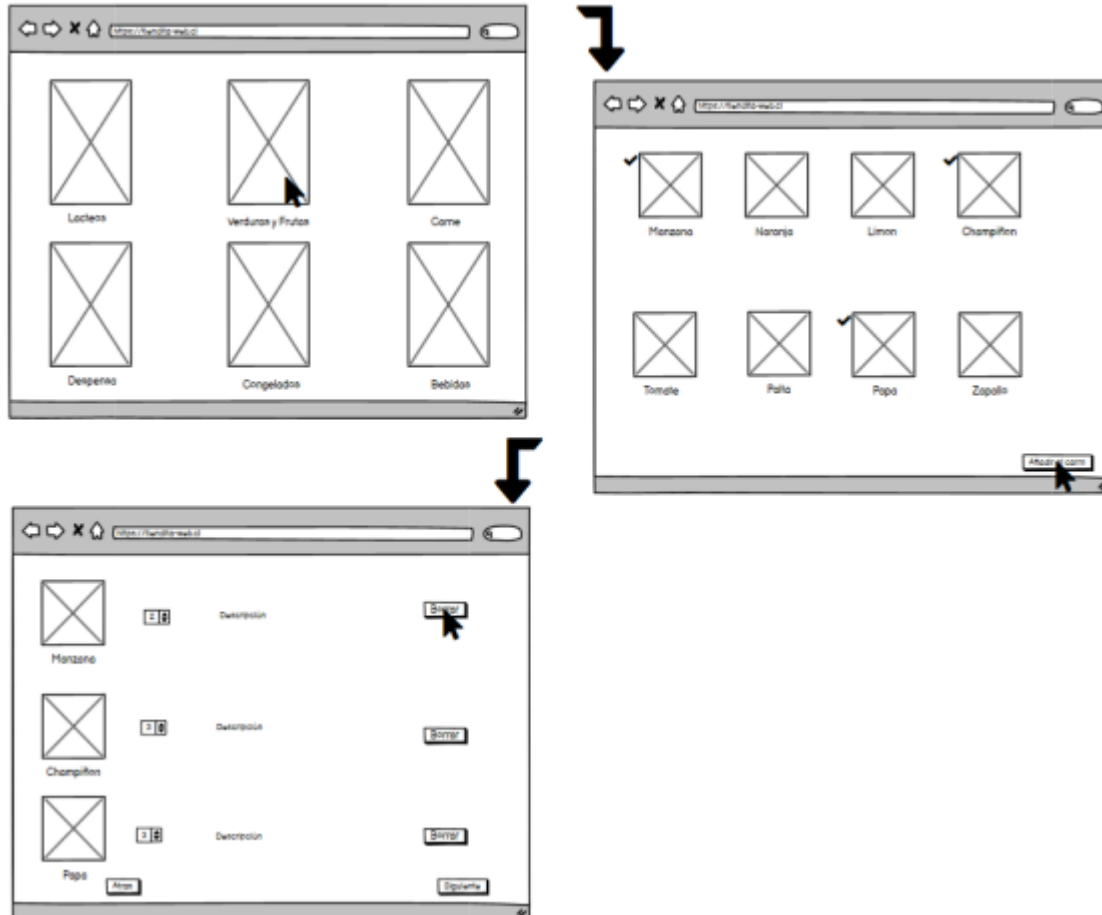


Diagrama de caso de uso:

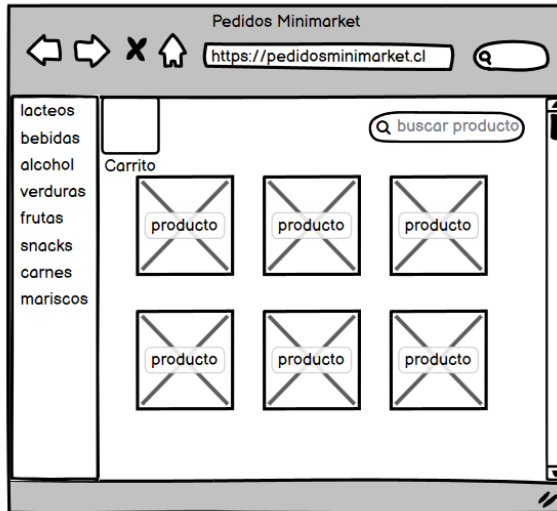


Mockups de pantallas (etapa inicial):

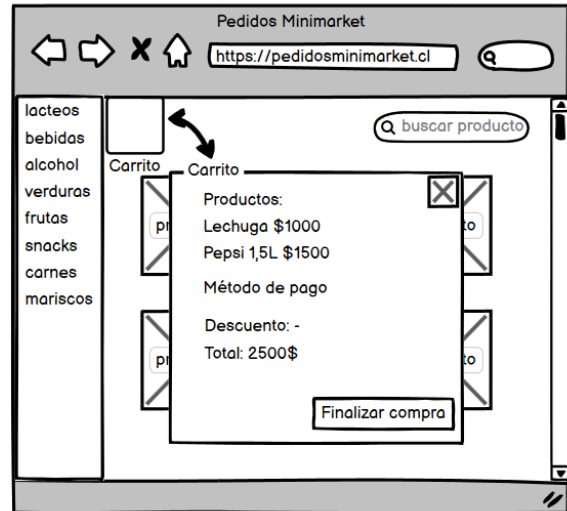


Mockups de pantalla (etapa final):

Pantalla front:



(figura 1)

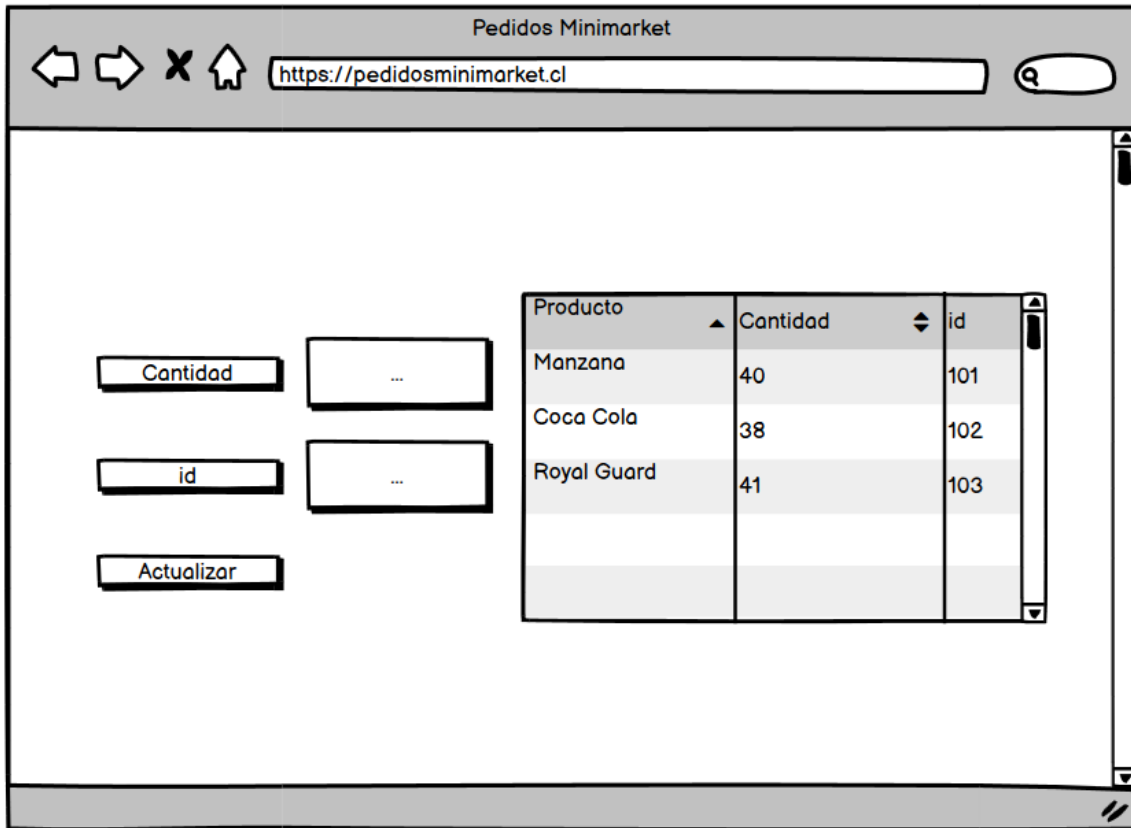


(figura 2)

- En la figura número 1, se tiene la vista del usuario al buscar y ver productos.
- En la figura número 2, se puede visualizar el carrito de compras al dar click.

En la etapa final se añadió una barra de búsqueda para simplificar el filtro de productos y de esta manera ser más cómoda para el cliente permitiéndole ahorrar tiempo.

Pantalla back:



The screenshot shows a web browser window titled "Pedidos Minimarket". The address bar displays "https://pedidosminimarket.cl". The main content area contains a form on the left and a table on the right.

Form:

- A label "Cantidad" next to a text input field containing "...".
- A label "id" next to a text input field containing "...".
- An "Actualizar" button.

Table:

Producto	Cantidad	id
Manzana	40	101
Coca Cola	38	102
Royal Guard	41	103

(figura 3)

- En la figura 3 se visualiza desde la vista de administrador el inventario para añadir stock.

Diccionario de Datos:

ADMINISTRADOR					
Nombre Atributo	PK/FK	Tipo de dato	NULL/ NOT NULL	Valores por defecto	Comentario
usuario	-	varchar(30)	NOT NULL	-	Nombre de la cuenta del administrador
contraseña	-	varchar(30)	NOT NULL	-	Contraseña de la cuenta del administrador
rut_admin	PK	varchar(12)	NOT NULL	-	RUT del administrador de la cuenta del administrador
nombre_admin	-	varchar(40)	NOT NULL	-	Nombre de admin de la cuenta del administrador

CLIENTE					
Nombre Atributo	PK/FK	Tipo de dato	NULL/ NOT NULL	Valores por defecto	Comentario
nombre	-	varchar(30)	NOT NULL	-	Nombre de la cuenta del cliente
rut	PK	varchar(12)	NOT NULL	-	RUT del cliente
email	-	varchar(30)	NOT NULL	-	EMAIL del cliente
telefono	-	int(9)	NOT NULL	-	Telefono del cliente
contraseña	-	varchar(30)	NOT NULL	-	Contraseña del cliente
direccion	-	varchar(30)	NOT NULL	-	Dirección del cliente

CARRITO

Nombre Atributo	PK/FK	Tipo de dato	NULL/ NOT NULL	Valores por defecto	Comentario
id	PK	int(11)	NOT NULL	-	Contador de productos
rut_cliente	-	varchar(9)	NOT NULL	-	Método de pago
cantidad	-	int(11)	NOT NULL	-	id del carrito de compras
id_producto	FK	int(11)	NOT NULL	-	id del producto

PRODUCTO

Nombre Atributo	PK/FK	Tipo de dato	NULL/ NOT NULL	Valores por defecto	Comentario
nombre_producto	-	varchar(30)	NOT NULL	-	Nombre del producto
id_producto	PK	int(11)	NOT NULL	-	id del producto
precio	-	int(9)	NOT NULL	-	Precio del producto
descuento_producto	-	int(2)	NOT NULL	-	Descuento aplicado al producto
id_inventario	FK	int(11)	NOT NULL	-	id del inventario
tipo	-	varchar(30)	NOT NULL	-	Tipo de producto (carnes, lacteos, bebidas, etc)

BOLETA

Nombre Atributo	PK/FK	Tipo de dato	NULL/ NOT NULL	Valores por defecto	Comentario
rut_cliente	FK	varchar(12)	NOT NULL	-	RUT del cliente
id_metodoPago	FK	int(11)	NOT NULL	-	Método con el que se va a pagar (visa, paypal, débito, etc)
id_boleta	PK	int(11)	NOT NULL	-	id de la boleta
total	-	int(11)	NOT NULL	-	Precio total de los productos

BOLETA_DETALLE

Nombre Atributo	PK/FK	Tipo de dato	NULL/ NOT NULL	Valores por defecto	Comentario
id_detalle	PK	int(11)	NOT NULL	-	RUT del cliente
id_producto	FK	int(11)	NOT NULL	-	Método con el que se va a pagar (visa, paypal, débito, etc)
id_boleta	FK	int(11)	NOT NULL	-	id de la boleta
cantidad	-	int(11)	NOT NULL	-	Cantidad de productos en el carrito
valor	-	int(11)	NOT NULL	-	Precio de los productos
fecha	-	date	NOT NULL	-	Fecha de emisión de la boleta

INVENTARIO

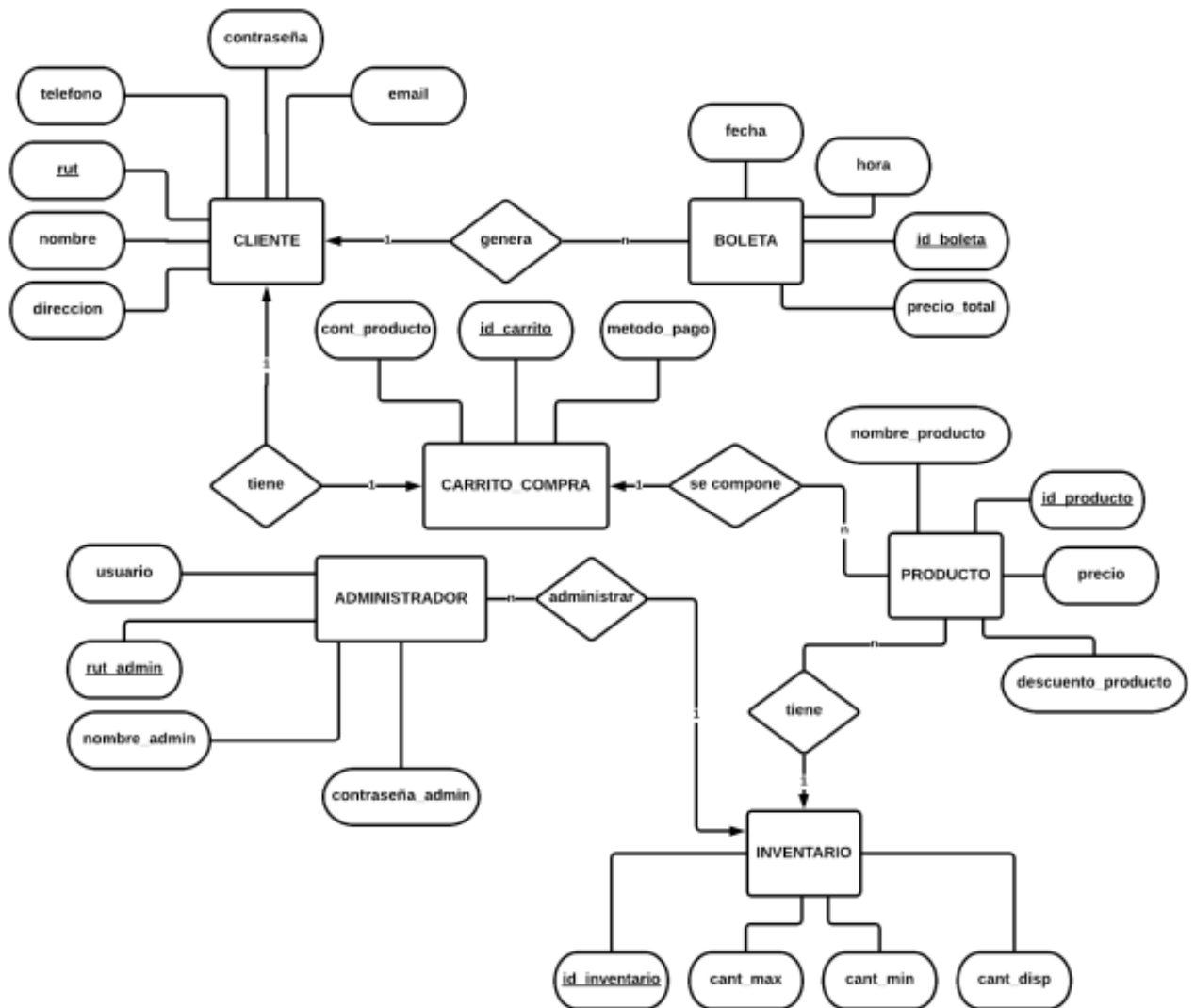
Nombre Atributo	PK/FK	Tipo de dato	NULL/ NOT NULL	Valores por defecto	Comentario
cantidad	-	int(2)	NOT NULL	0	Cantidad disponible de x producto
id_inventario	PK	int(11)	NOT NULL	-	id del inventario
rut_admin	FK	varchar(12)	NOT NULL	-	RUT del administrador

METODO_PAGO

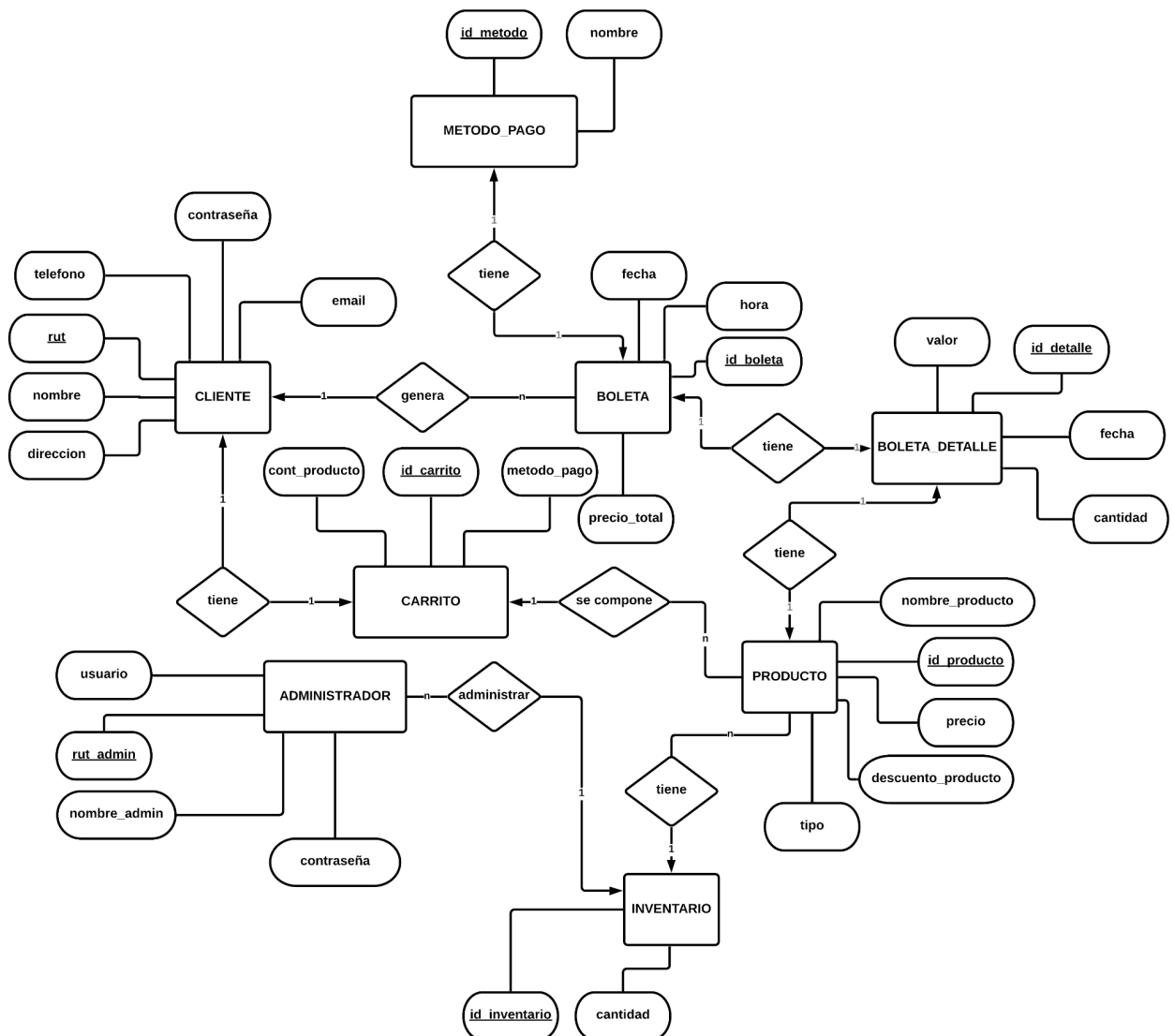
Nombre Atributo	PK/FK	Tipo de dato	NULL/ NOT NULL	Valores por defecto	Comentario
id_metodo	PK	int(11)	NOT NULL	-	id del metodo de pago
nombre	-	int(30)	NOT NULL	-	Nombre del tipo de pago

BD:

Modelo Entidad-Relación (etapa inicial):

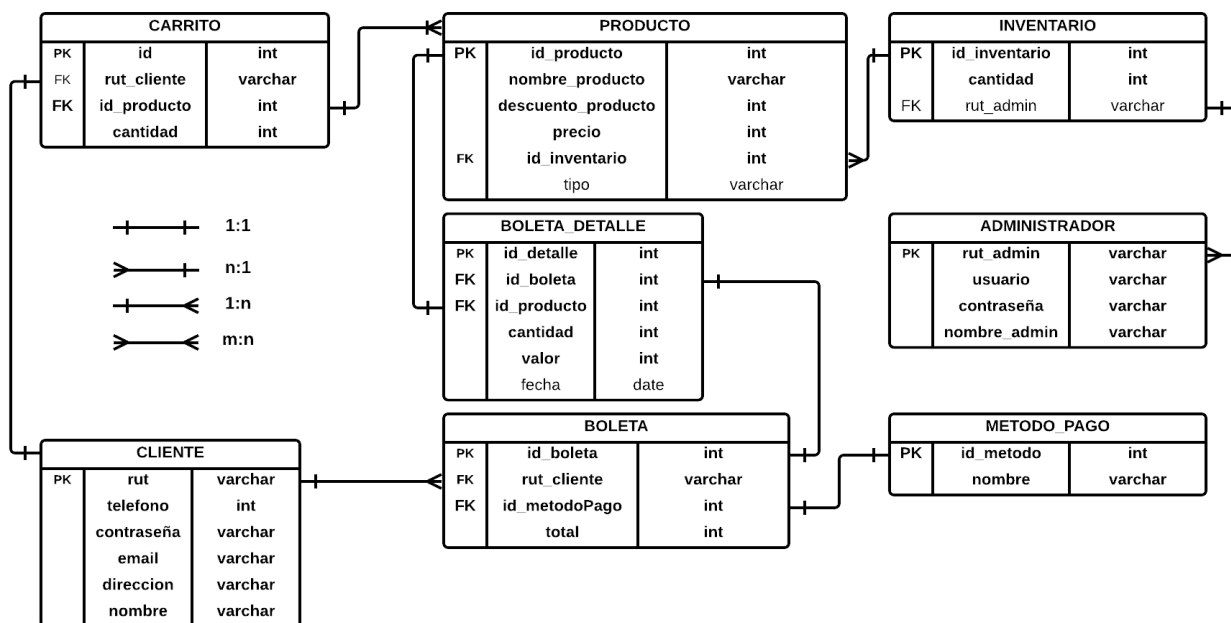


Modelo Entidad-Relación (etapa final):



El modelo entidad relación que teníamos en la etapa inicial no cumplía con las necesidades del problema presentado, pues las relaciones no eran coherentes, habían demasiados atributos innecesarios, entidades innecesarias y la ausencia de las realmente importantes. Era un modelo relacional basado en un problema que no entendimos del todo, por ende estaba erróneo.

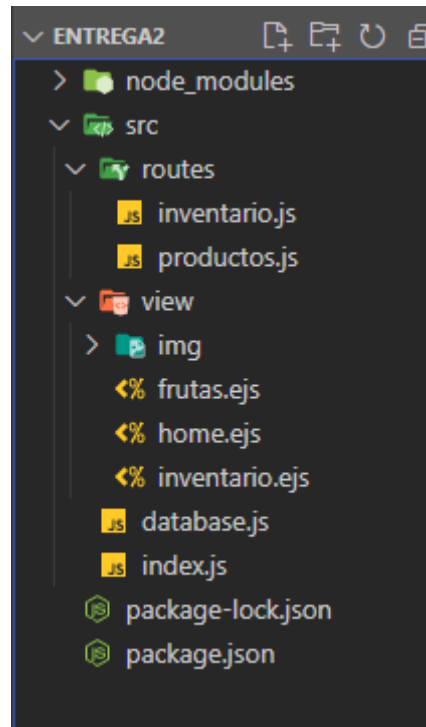
Modelo Relacional:



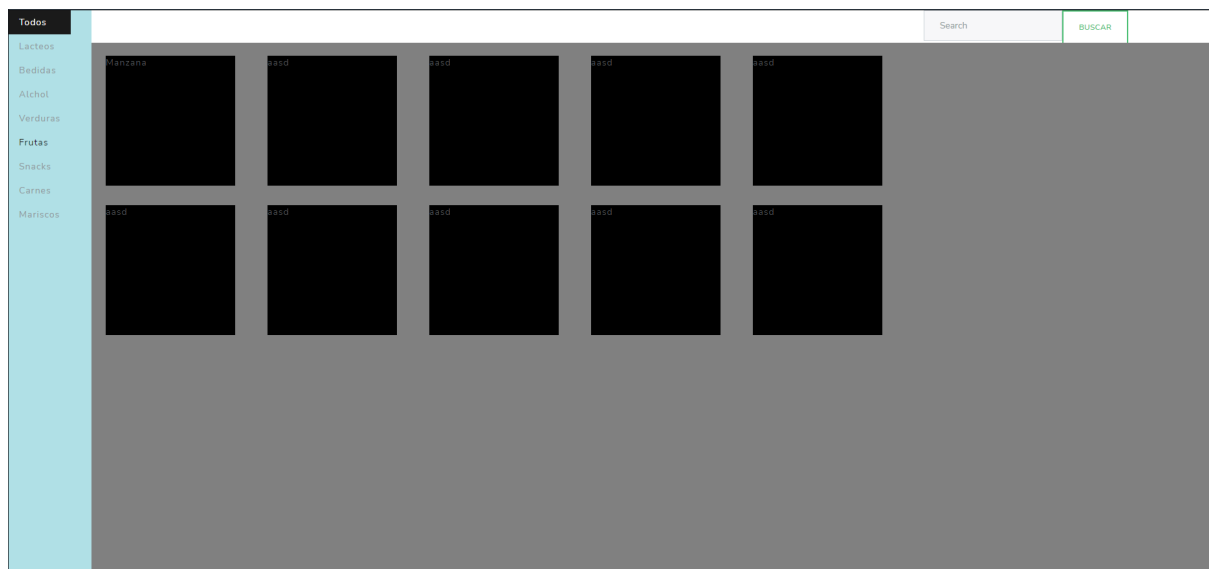
Restricciones de integridad referencial y supuestos:

- El sistema se usará en una sola sucursal.
- El sistema solo permitirá un carrito de compra por usuario.

Backend (API rest):



Pantallas Front (inicial):



Pantalla front general de todos los productos.



Vista solamente del tipo frutas.

Pantalla Back(inicial):

FORMULARIO

ID	Nombre	Cantidad	
1	pera	13	<div>Cantidad para sumar=</div> <input type="text"/> <div>AGREGAR</div>
2	manzana	7	<div>Cantidad para sumar=</div> <input type="text"/> <div>AGREGAR</div>
3	platano	7	<div>Cantidad para sumar=</div> <input type="text"/> <div>AGREGAR</div>
4	papaya	2	<div>Cantidad para sumar=</div> <input type="text"/> <div>AGREGAR</div>
5	Coca cola	69	<div>Cantidad para sumar=</div> <input type="text"/> <div>AGREGAR</div>

Pantallas Front (Final):

[illegible]

ACCEDER

INICIO

Search

SEARCH

ACCEDER

INICIAR SESION

INICIO

Search

SEARCH

REGISTRAR

Rut

Telefono

Contrasena

Email

Direccion

Nombre

ENVIAR

Todos

Lacteos

Bedidas

Alchol

Verduras

Frutas

Snacks

Carnes

Mariscos

Search

BUSCAR

pera

manzana

platano

papaya

Todos

Lacteos

Bedidas

Alchol

Verduras

Frutas

Snacks

Carnes

Mariscos

Search

BUSCAR

ACCEDER

REGISTRARSE

Pantalla Back(Final):

FORMULARIO

Nombre

Descuento

Precio

Tipo

ID inventario

ENVIAR

ID	Nombre	Cantidad	
1	pera	14	<div>Añadir al inventario =</div> <div></div> <div>AGREGAR</div>
2	manzana	7	<div>Añadir al inventario =</div> <div></div> <div>AGREGAR</div>
3	platano	7	<div>Añadir al inventario =</div> <div></div> <div>AGREGAR</div>
4	papaya	2	<div>Añadir al inventario =</div> <div></div> <div>AGREGAR</div>
5	Coca cola	69	<div>Añadir al inventario =</div> <div></div> <div>AGREGAR</div>

Estructura BD no relacional:

Local

5 DBS10 COLLECTIONS

☆ FAVORITE

HOSTlocalhost:27017

CLUSTERStandalone

EDITIONMongoDB 5.0.4 Community

Filter your data

admin

carrito_compra

administrador

boleto

boleto_detalle

carrito

cliente

inventario

metodo_pago

producto

















config

Collections

CREATE COLLECTION

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
administrador	2	69.5 B	139.0 B	1	36.9 KB	
boleto	2	109.0 B	218.0 B	1	36.9 KB	
boleto_detalle	2	137.0 B	274.0 B	1	20.5 KB	
carrito	2	103.0 B	206.0 B	1	20.5 KB	
cliente	1	153.0 B	153.0 B	1	20.5 KB	
inventario	5	91.0 B	455.0 B	1	36.9 KB	
metodo_pago	2	61.0 B	122.0 B	1	20.5 KB	
producto	5	154.8 B	774.0 B	1	36.9 KB	

Dump BD no relacional:

	administrador.bson	06-12-2021 7:52	Archivo BSON	1 KB
	administrador.metadata.json	06-12-2021 7:52	JSON File	1 KB
	boleta.bson	06-12-2021 7:52	Archivo BSON	1 KB
	boleta.metadata.json	06-12-2021 7:52	JSON File	1 KB
	boleta_detalle.bson	06-12-2021 7:52	Archivo BSON	1 KB
	boleta_detalle.metadata.json	06-12-2021 7:52	JSON File	1 KB
	carrito.bson	06-12-2021 7:52	Archivo BSON	1 KB
	carrito.metadata.json	06-12-2021 7:52	JSON File	1 KB
	cliente.bson	06-12-2021 7:52	Archivo BSON	1 KB
	cliente.metadata.json	06-12-2021 7:52	JSON File	1 KB
	inventario.bson	06-12-2021 7:52	Archivo BSON	1 KB
	inventario.metadata.json	06-12-2021 7:52	JSON File	1 KB
	metodo_pago.bson	06-12-2021 7:52	Archivo BSON	1 KB
	metodo_pago.metadata.json	06-12-2021 7:52	JSON File	1 KB
	producto.bson	06-12-2021 7:52	Archivo BSON	1 KB
	producto.metadata.json	06-12-2021 7:52	JSON File	1 KB

Consultas:

Consultas de la entrega 1 en el txt.

- Consulta para mostrar tablas, nombres del producto, id del inventario y cantidad (sin utilización web).**
 SELECT inventario.cantidad, producto.nombre_producto FROM inventario
 inner JOIN producto on producto.id_inventario=inventario.id_inventario
- Consulta para actualización de cantidad de productos (sin utilización web).**
 update inventario set cantidad = 3+cantidad where id_inventario=1
- Insertar nuevos datos a la tabla de productos (en formulario)**
 INSERT into producto SET ?
- Consulta que inserta datos en la tabla boleta.**
 INSERT INTO boleta (fecha,total,rut_cliente,id_metodoPago) SELECT
 '2021-09-15',SUM(producto.precio * carrito.cantidad),'22.222.222-2',2 FROM
 producto INNER JOIN carrito ON (producto.id_producto= carrito.id_producto)
- Consulta que inserta datos dentro de la tabla boleta_detalle.**
 INSERT INTO boleta_detalle (valor,cantidad,id_boleta,id_producto) SELECT
 producto.precio,carrito.cantidad,3,producto.id_producto FROM producto INNER
 JOIN carrito ON (producto.id_producto= carrito.id_producto)

Conclusión:

Como conclusión podemos comentar que se aprendió bastante sobre la creación de base de datos y la utilización de herramientas tales como xampp, mysql, mongoDB, nodejs, bootstratp, etc. Sin embargo no pudimos alcanzar la meta esperada esto debido a que si bien trabajabamos en el proyecto tampoco resultó ser tan sencillo, aun así podemos decir que aprendimos lo básico y esencial para poder llevar a cabo algún proyecto personal y entender la información que se puede encontrar en internet.