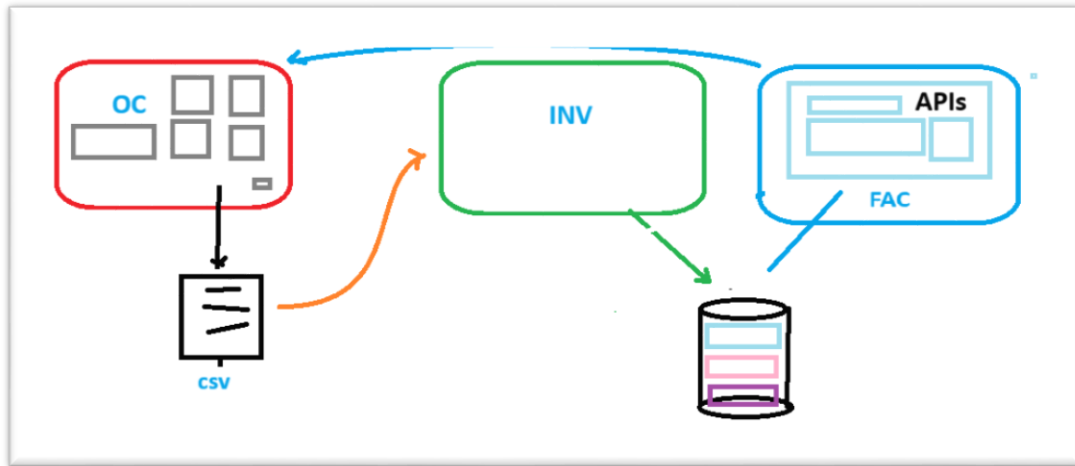


Nombre: Leonardo Salazar



1. Generamos un Sistema de Orden de Compra, simula consumiendo una API publica y completa la información de la orden de compra llenando la cantidad y el número de orden

```

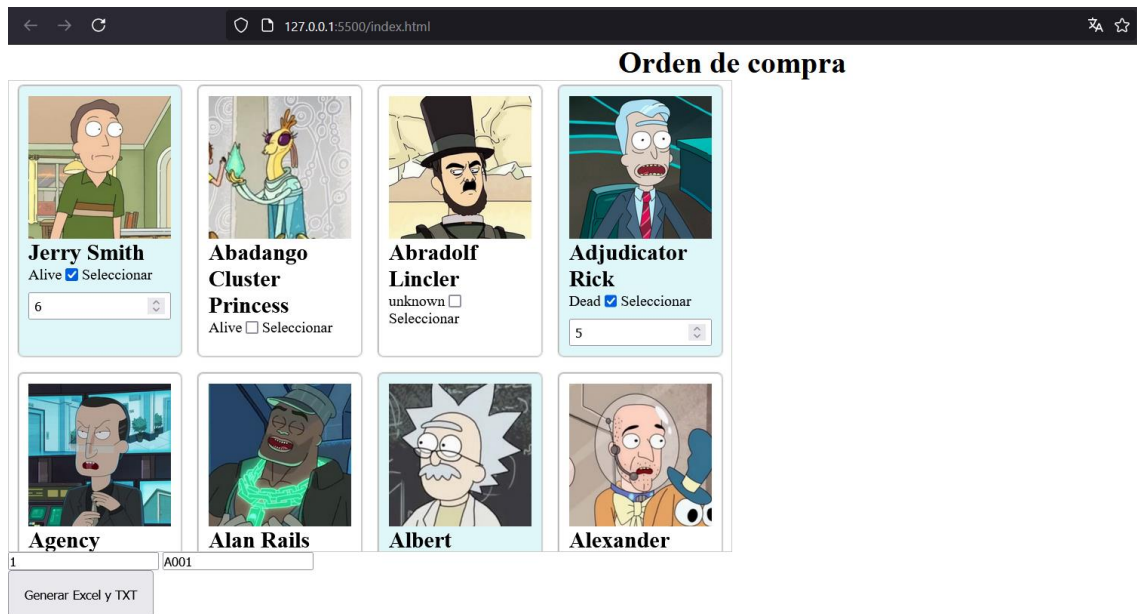
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Personajes</title>
7   <script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.min.js"></script>
8   <script defer src="index.js"></script>
9 </head>
10 <body>
11   <h1>Personajes</h1>
12   <main></main>
13   <button id="generateExcel">Generar Excel y TXT</button>
14
15 </body>
16 </html>
17
18
19
20

```

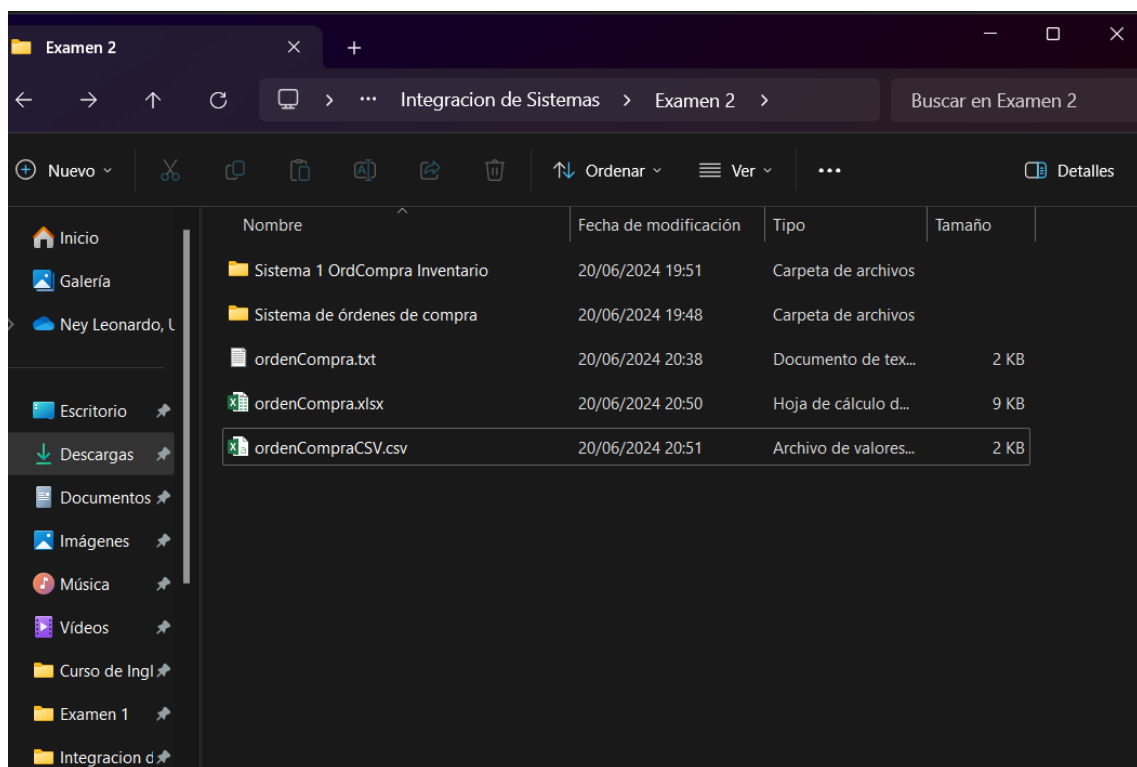
```

1 function getCharacters(done) {
2   const results = fetch("https://rickandmortyapi.com/api/character");
3
4   results
5     .then(response => response.json())
6     .then(data => {
7       done(data);
8     });
9 }
10
11 getCharacters(data => {
12   data.results.forEach(personaje => {
13     const article = document.createElement("div");
14     <div class="image-container">
15       
16     </div>
17     <h2>${personaje.name}</h2>
18     <span>${personaje.status}</span>
19   </div>
20   );
21   const main = document.querySelector("main");
22   main.appendChild(article);
23 });
24
25 function generateID(index) {
26   const number = String(index).padStart(3, '0');
27   return `0${number}`;
28 }
29
30 function saveToTxt(characters) {
31   const headers = "ID, Image, Name, Status";
32   const txtContent = characters.map(c => `${c.ID}, ${c.image}, ${c.name}, ${c.status}`).join('\n');
33   const fullContent = `${headers}\n${txtContent}`;
34   const blob = new Blob([fullContent], { type: 'text/plain' });
35 }
36
37

```



2. Luego de realizar la Orden de Compra se genera el archivo de Excel

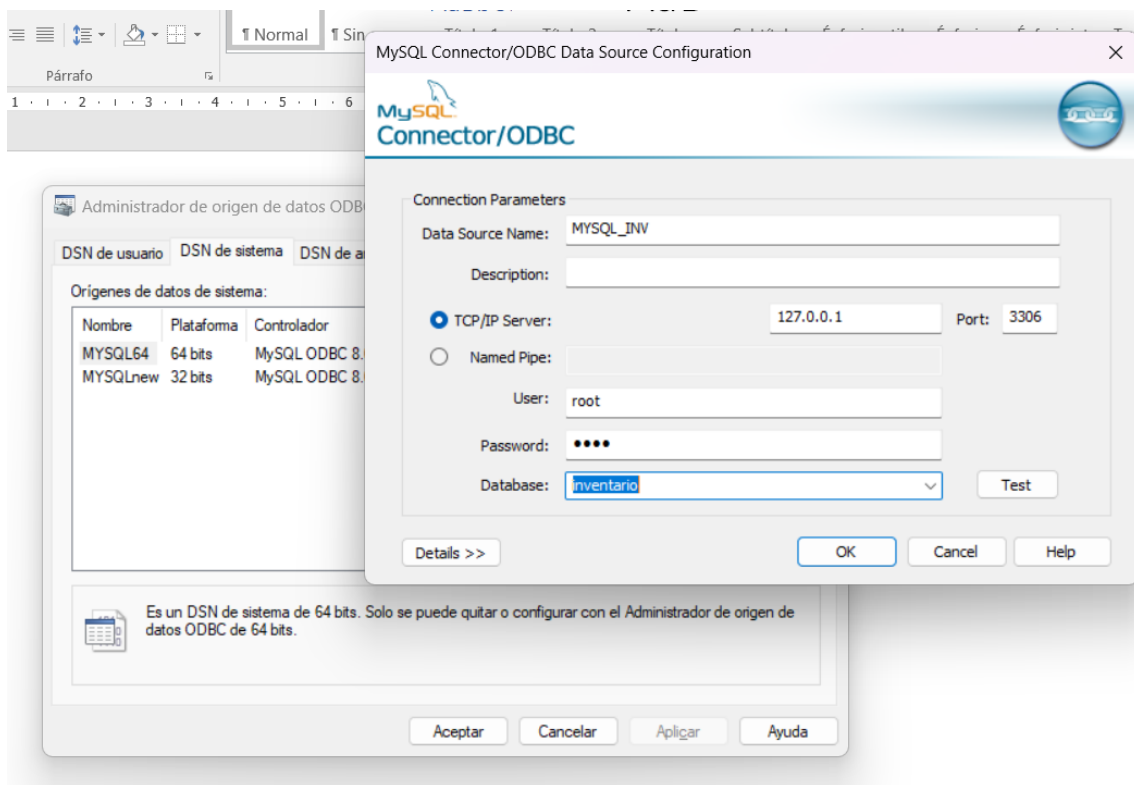


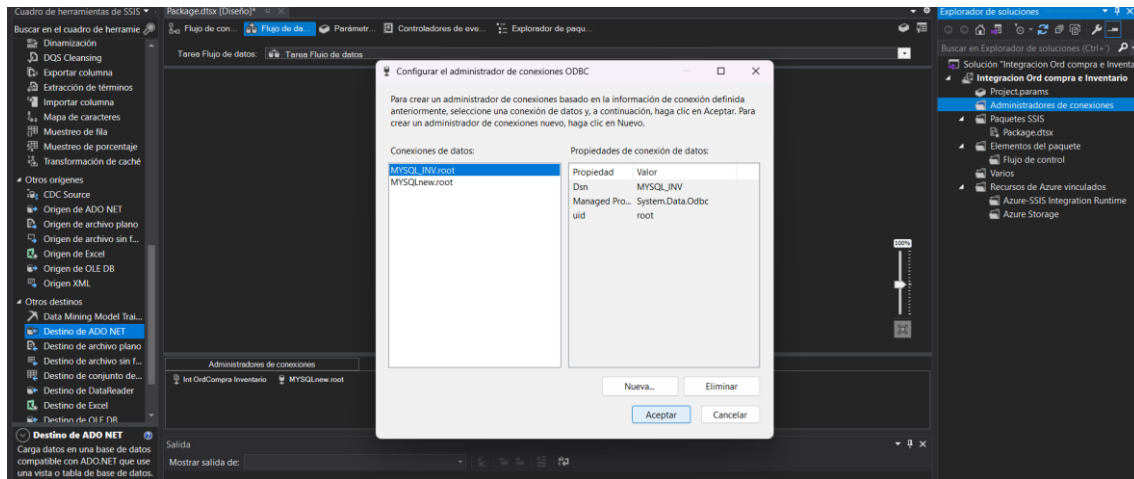
ordenCompra.csv - Excel

idCliente	numOrden	idProducto	imagen	nombre	estado	cantidad								
1	A001	5	https://rickan	Jerry Smith	Alive	6								
1	A001	8	https://rickan	Adjudicator Ri	Dead	5								
1	A001	11	https://rickan	Albert Einstein	Dead	2								

ordenCompra

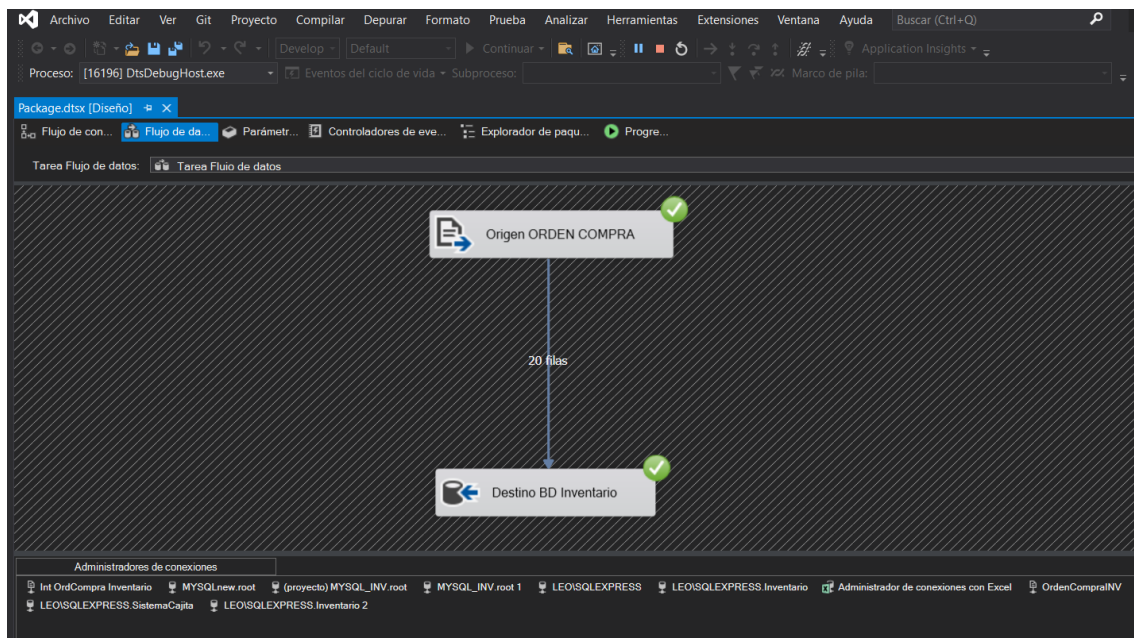
3. Integramos el archivo CSV en el Sistema de Inventario, el sistema de inventario carga la información de Orden Compra a la base de datos InventarioCompartido

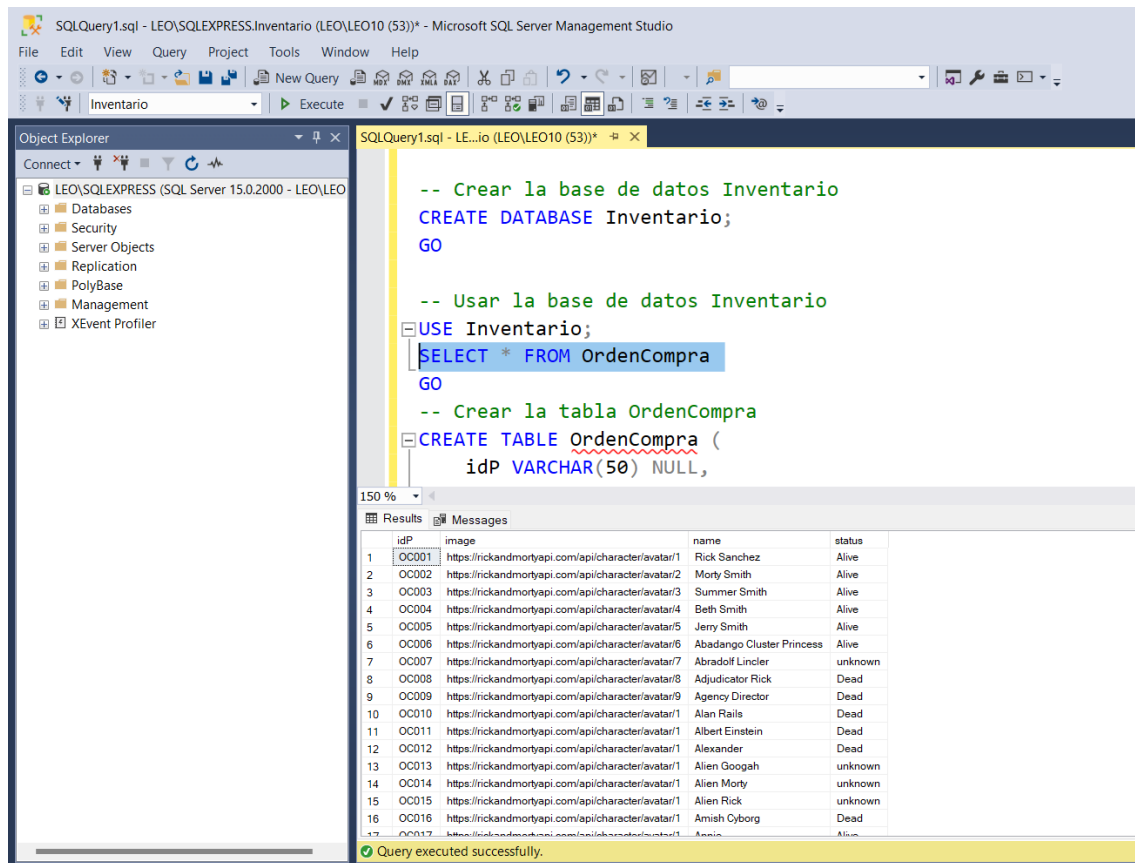




Integramos el CSV generado por el Sistema de Orden de compra a la base de datos del sistema Inventario. Mediante el uso de un query “Procedimiento almacenado”

```
-- Crear el procedimiento almacenado para actualizar el stock
DELIMITER //
CREATE PROCEDURE ActualizarStock()
BEGIN
    UPDATE Producto p
    INNER JOIN ProductoTemporal pt ON p.idProducto = pt.idProducto
    SET p.stock = pt.stock;
END //
DELIMITER ;
```





4. Base de datos compartida la creé en Mysql “InventarioCompartido” la Cual contiene las tablas Cliente, Producto, OrdenCompra, Factura

```
CREATE DATABASE InventarioCompartido;

-- Usar la base de datos InventarioCompartido
USE InventarioCompartido;

-- Crear la tabla Cliente
CREATE TABLE Cliente (
    idCliente INT PRIMARY KEY AUTO_INCREMENT,
    nombreCliente VARCHAR(100) NOT NULL,
    correo VARCHAR(50) NOT NULL,
    telefono CHAR(10) NOT NULL
);

-- Crear la tabla Producto
CREATE TABLE Producto (
    idProducto INT PRIMARY KEY AUTO_INCREMENT,
    nombreProducto VARCHAR(100) NOT NULL,
    precio DECIMAL(10, 2) NOT NULL,
    stock INT NOT NULL
);

-- Crear la tabla OrdenCompra
CREATE TABLE OrdenCompra (
    idOrden INT PRIMARY KEY AUTO_INCREMENT,
    idCliente INT NOT NULL,
    numOrden VARCHAR(50) NOT NULL UNIQUE,
    idProducto INT NOT NULL,
    imagen TEXT NULL,
```

```

        nombre VARCHAR(100) NULL,
        estado VARCHAR(50) NULL,
        cantidad INT NOT NULL,
        FOREIGN KEY (idCliente) REFERENCES Cliente(idCliente),
        FOREIGN KEY (idProducto) REFERENCES Producto(idProducto)
    );

-- Crear la tabla Factura
CREATE TABLE Factura (
    idFactura INT PRIMARY KEY AUTO_INCREMENT,
    numOrden VARCHAR(50) NOT NULL,
    fechaFactura DATE NOT NULL,
    totalFactura DECIMAL(10, 2) NOT NULL
);

-- Ingreso registros de prueba

USE InventarioCompartido;
-- Insertar registros en la tabla Cliente
INSERT INTO Cliente (nombreCliente, correo, telefono) VALUES
('Juan Perez', 'juan.perez@example.com', '0987654321'),
('Maria Gomez', 'maria.gomez@example.com', '0987654322'),
('Carlos Ramirez', 'carlos.ramirez@example.com', '0987654323'),
('Ana Lopez', 'ana.lopez@example.com', '0987654324'),
('Luis Martinez', 'luis.martinez@example.com', '0987654325');

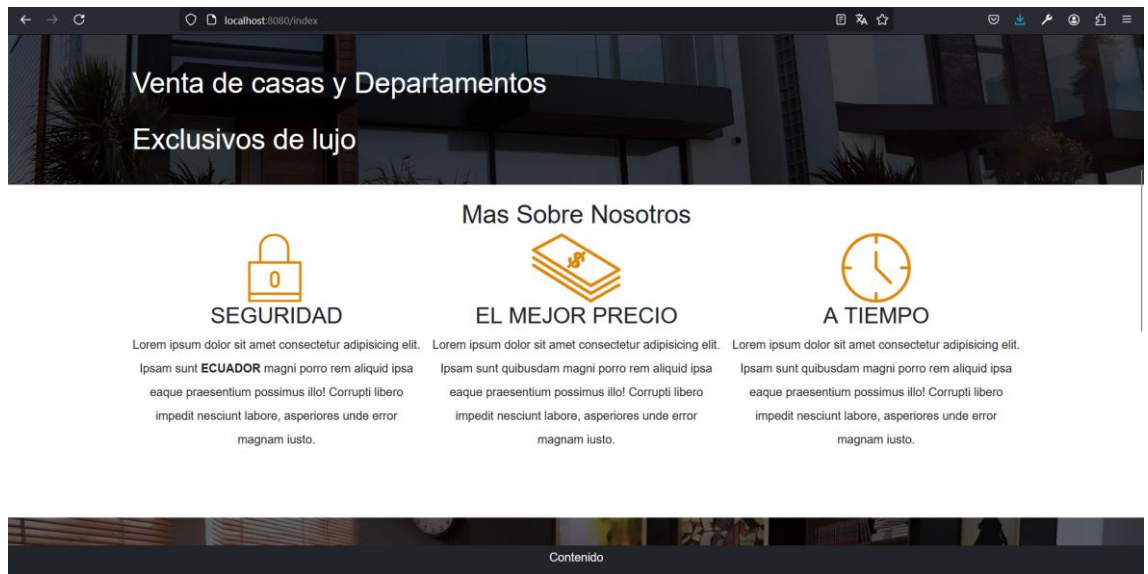
-- Insertar registros en la tabla Producto
INSERT INTO Producto (nombreProducto, precio, stock) VALUES
('Producto A', 10.50, 100),
('Producto B', 20.75, 150),
('Producto C', 15.00, 200),
('Producto D', 25.00, 50),
('Producto E', 30.00, 75);

-- Insertar registros en la tabla OrdenCompra
INSERT INTO OrdenCompra (idCliente, numOrden, idProducto, imagen, nombre, estado,
cantidad) VALUES
(1, 'OC001', 1, 'https://example.com/image1.jpg', 'Producto A', 'Enviado', 2),
(2, 'OC002', 2, 'https://example.com/image2.jpg', 'Producto B', 'Procesando', 3),
(3, 'OC003', 3, 'https://example.com/image3.jpg', 'Producto C', 'Entregado', 1),
(4, 'OC004', 4, 'https://example.com/image4.jpg', 'Producto D', 'Enviado', 5),
(5, 'OC005', 5, 'https://example.com/image5.jpg', 'Producto E', 'Procesando', 4);

USE InventarioCompartido;
-- Insertar registros en la tabla Factura
INSERT INTO Factura (numOrden, fechaFactura, totalFactura) VALUES
('OC001', '2024-06-01', 21.00),
('OC002', '2024-06-02', 62.25),
('OC003', '2024-06-03', 15.00),
('OC004', '2024-06-04', 125.00),
('OC005', '2024-06-05', 120.00);

```

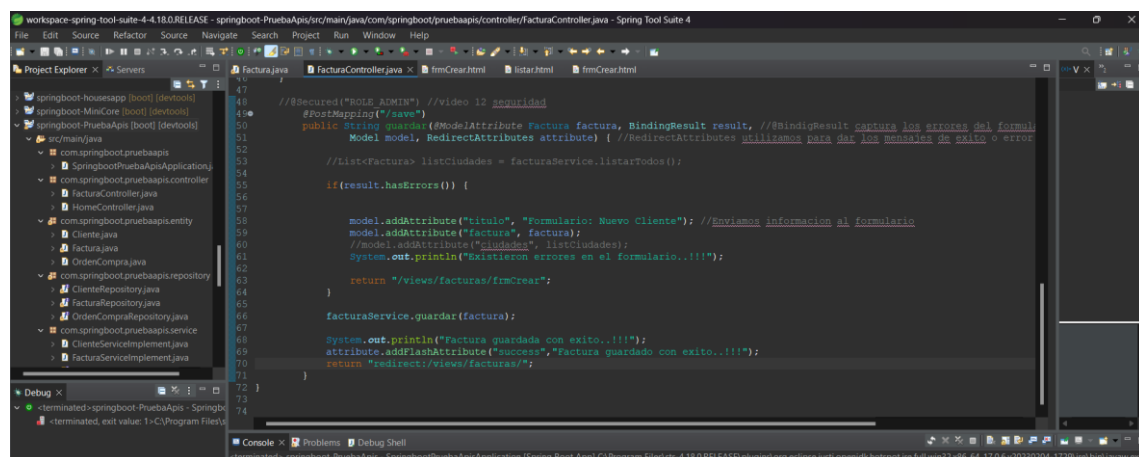
5. El sistema de Facturacion nos brinda una API POST para obtener la información de OrdenCompra y Generar la factura.



La API recibe la información los parámetros de la orden de compra, en si el precio total de la factura y los datos de la fecha y numero de Orden.

Se realizo 2 APIs una para listar todas las facturas y otra para guardar la factura en la base de datos.

```
Factura.java | FacturaController.java | frmCrear.html | listar.html | frmCrear.html
19 @RequestMapping("/views/facturas")
20 public class FacturaController {
21
22     @Autowired
23     private IFacturaService facturaService;
24
25     @GetMapping("/")
26     public String listarFacturas(Model model) {
27         List<Factura> listadoFactura = facturaService.listarTodos();
28         for (Factura factura : listadoFactura) {
29             System.out.println(factura);
30         }
31         model.addAttribute("titulo", "Lista de Facturas");
32         model.addAttribute("facturas", listadoFactura);
33         return "/views/facturas/listar";
34     }
35 }
```



Podemos visualizar la información de todas las tablas

Tenemos la lista de los Clientes

Mini COREInicioClientesInmueblesEstadísticaAPIs

Lista de Clientes

Busca "Nuevo Cliente" Para Crear Nuevo Registro

Id	Nombres	Apellidos	Teléfono	Email	Ciudad	Editar	Eliminar	Inmuebles
1	Jose Marco	Guachamin Lasso	0897644343	Jose34@gmail.com	QUITO	Editar	Eliminar	Inmuebles
3	Dario Juan	Galindez Burbano	0987655443	Dario99@gmail.com	CUENCA	Editar	Eliminar	Inmuebles
4	Wily Fernando	Vaca Sosa	0978966775	Wily11@gmail.com	GYE	Editar	Eliminar	Inmuebles
5	Lorenzo Carlos	Solis Minda	0986655663	Lorenzo87@gmail.com	QUITO	Editar	Eliminar	Inmuebles
6	Hugo Eduardo	Zambrano Lopez	0897866774	hugo@gmail.com	IBARRA	Editar	Eliminar	Inmuebles
7	Juan	Pérez	123456789	juan@example.com	QUITO	Editar	Eliminar	Inmuebles
8	María	Gómez	987654321	maria@example.com	QUITO	Editar	Eliminar	Inmuebles
9	Carlos	López	555555555	carlos@example.com	QUITO	Editar	Eliminar	Inmuebles
10	Ana	García	999999999	ana@example.com	QUITO	Editar	Eliminar	Inmuebles

Podemos obtener la lista de Facturas y sus datos

FACTURACIÓNInicioOTRO

Lista de Facturas

idFactura	numOrden	fechaFactura	totalFactura
1	OC001	2024-06-01 00:00:00.0	21.00
2	OC002	2024-06-02 00:00:00.0	62.25
3	OC003	2024-06-03 00:00:00.0	15.00
4	OC004	2024-06-04 00:00:00.0	125.00
5	OC005	2024-06-05 00:00:00.0	120.00

Visualizamos también la tabla de Producto

Id	cliente nombre	Area	Tipo	Sector	Dormitorios	Baños	Precio	Descripcion	Editar	Eliminar
1	Hugo Eduardo	80	Apartamento	El Batán	2	1	150000.0	Acogedor apartamento en venta	Editar	Eliminar
2	Dario Juan	120	Casa	Iñaquito	4	3	350000.0	Amplia casa con jardín	Editar	Eliminar
3	Wily Fernando	90	Apartamento	Nayon	1	1	120000.0	Lindo apartamento cerca del centro	Editar	Eliminar
4	Jose Marco	150	Casa	El Inca	5	4	450000.0	Espaciosa casa con piscina	Editar	Eliminar
5	Dario Juan	70	Apartamento	Sangolqui	2	1	180000.0	Moderno apartamento con vistas	Editar	Eliminar
6	Jose Marco	110	Casa	San Carlos	4	3	320000.0	Casa en esquina con patio	Editar	Eliminar
7	Wily Fernando	95	Apartamento	Prensa	2	2	140000.0	Apartamento luminoso en zona tranquila	Editar	Eliminar
8	Dario Juan	130	Casa	Cotacollao	6	4	550000.0	Amplia casa con acabados de lujo	Editar	Eliminar
9	Jose Marco	75	Apartamento	Carcelén	1	1	100000.0	Apartamento ideal para parejas	Editar	Eliminar
10	Lorenzo Carlos	85	Casa	Recreo	3	2	280000.0	Casa remodelada con patio trasero	Editar	Eliminar
11	Dario Juan	95	Apartamento	Quitumbe	2	1	160000.0	Apartamento moderno con excelente ubicación	Editar	Eliminar
12	Wily Fernando	120	Casa	La Mariscal	4	3	380000.0	Casa colonial en el centro histórico	Editar	Eliminar
13	Jose Marco	70	Apartamento	El Batán	1	1	130000.0	Apartamento acogedor con balcón	Editar	Eliminar
14	Wily Fernando	140	Casa	Iñaquito	5	4	480000.0	Casa con amplio jardín y zona de BBQ	Editar	Eliminar
15	Dario Juan	105	Apartamento	Nayon	3	2	200000.0	Apartamento con hermosa vista panorámica	Editar	Eliminar
16	Wily Fernando	90	Casa	El Inca	3	2	320000.0	Casa con diseño moderno y acabados de primera	Editar	Eliminar