



## **Sistema B2B**

### **Manual Técnico**

Proyecto interdisciplinario de Análisis, Diseño y Fabricación de Sistemas y Sistemas de Bases de Datos 2.

DYLAN GABRIEL RODAS SAMAYOA

GABRIEL ESTUARDO LEMUS PACHECO

## ÍNDICE

1	DESCRIPCIÓN DEL MANUAL	3
1.1	Objeto	3
1.2	Alcance	3
1.3	Funcionalidad	3
2	MAPA DEL SISTEMA	¡Error! Marcador no definido.
2.1	Navegación del Sistema de Ventas	¡Error! Marcador no definido.
2.2	Navegación del Sistema de Fábrica	¡Error! Marcador no definido.
3	DESCRIPCIÓN DEL SISTEMA	4
3.1	Sistema de Ventas	5
3.1.1	Inicio	¡Error! Marcador no definido.
3.1.2	Registro	¡Error! Marcador no definido.
3.1.3	Inicio de Sesión	¡Error! Marcador no definido.
3.1.4	Catálogo de Ventas	¡Error! Marcador no definido.
3.1.5	Ventas	¡Error! Marcador no definido.
3.1.6	Clientes	¡Error! Marcador no definido.
3.1.7	Catálogo de Dispositivos Electrónicos	¡Error! Marcador no definido.
3.1.8	Órdenes	¡Error! Marcador no definido.
3.1.9	Gestión de Órdenes	¡Error! Marcador no definido.
3.1.10	Perfil	¡Error! Marcador no definido.
3.1.11	Página No Encontrada	¡Error! Marcador no definido.
3.2	Sistema de Fábricas	¡Error! Marcador no definido.
3.2.1	Inicio	¡Error! Marcador no definido.
3.2.2	Registro	¡Error! Marcador no definido.
3.2.3	Inicio de Sesión	¡Error! Marcador no definido.
3.2.4	Catálogo de Dispositivos Electrónicos	¡Error! Marcador no definido.
3.2.5	Catálogo de Clientes	¡Error! Marcador no definido.
3.2.6	Órdenes	¡Error! Marcador no definido.
3.2.7	Gestión de Órdenes	¡Error! Marcador no definido.
3.2.8	Reportes de Facturación	¡Error! Marcador no definido.
3.2.9	Perfil	¡Error! Marcador no definido.
3.2.10	Página No Encontrada	¡Error! Marcador no definido.
3.2.11	Búsqueda de dispositivos	¡Error! Marcador no definido.
4	FAQ	¡Error! Marcador no definido.
5	GLOSARIO	¡Error! Marcador no definido.
6	BIBLIOGRAFÍA Y REFERENCIAS	¡Error! Marcador no definido.

## **1 DESCRIPCIÓN DEL MANUAL**

### **1.1 Objeto**

El presente documento pretende mostrar el funcionamiento técnico de la aplicación Sistema B2B para la compra y venta de dispositivos electrónicos.

### **1.2 Alcance**

Se pretende mostrar de una manera técnica y profunda el funcionamiento de la aplicación Sistema B2B para la compra y venta de dispositivos electrónicos, para todo aquel interesado pueda obtener el conocimiento mediante la lectura de este manual.

### **1.3 Funcionalidad**

Dada la necesidad de cubrir la forma de entablar las interacciones entre distintas compañías para establecer cadenas de suministro, el siguiente sistema se implementa para diversas empresas de dos distintas lógicas de negocio en un ecosistema B2B, siendo el primer tipo un sistema para fabricantes de dispositivos electrónicos de una marca específica que recibe pedidos y envía los pedidos a distintas distribuidoras y vendedoras de dispositivos electrónicos que lo soliciten. Y el segundo, un sistema para distribuidoras y ventas de dispositivos electrónicos que venden electrónicos de varias marcas a clientes que llegan a sus instalaciones y permite hacer sus pedidos hacia las fábricas.

## **2 DESCRIPCIÓN DEL SISTEMA**

Este sistema permitirá a los fabricantes de dispositivos recibir pedidos y gestionar sus envíos a los distribuidores. Por su parte, los distribuidores podrán solicitar pedidos a las fábricas y gestionar sus ventas. Por parte de los fabricantes, podrán detallar la información de sus dispositivos, incluyendo datos como el tipo de dispositivo electrónico, categoría, modelo, entre otros. Los detalles que los fabricantes agregan sobre sus productos les permitirán a los distribuidores realizar consultas sobre los productos de manera más específica, para satisfacer necesidades puntuales que puedan tener sus clientes.

El sistema de ventas estará disponible para 3 tipos de clientes: clientes individuales, grandes clientes y distribuidores mayoristas. Los clientes individuales tendrán acceso a realizar compras de productos. Por su parte, los grandes clientes y distribuidores mayoristas pagan una suscripción anual que les dará un descuento del 5% y 15%, respectivamente. Adicionalmente, los distribuidores mayoristas podrán realizar compras a crédito y realizar pedidos a futuro para los productos que no se encuentren en existencia.

## 2.1 Webservices

### 2.1.1 Nueva marca

**Definición:** realizar el registro de una nueva fábrica como una marca dentro del sistema de ventas. Cuando una nueva fábrica se registre dentro del sistema de fábricas, se hace el llamado a este web service para que se lo comunique al backend de ventas y lo registre como una nueva marca.

**Parámetros:**

- **newBrand:** nombre de la nueva marca en el sistema de ventas.

**JSON Modelo:**

```
{
  nombre: "222555GB4", // Nombre de la fábrica que será registrada como marca
}
```

### 2.1.2 Nuevo cliente

**Definición:** registrar a un vendedor del sistema de ventas como un cliente dentro del sistema de fábricas.

**Parámetros:**

- **seller:** nombre del vendedor en el sistema de ventas que será registrado como un cliente en el sistema de fábricas.
- **email:** correo electrónico del vendedor.

**JSON Modelo:**

```
{
  name: { type: String, required: true },
  email: { type: String, required: true },
  shippingTimes: [
    {
      factoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Factory" },
      shippingTime: { type: Number, required: true }
    }
  ],
}
```

### 2.1.3 Pagar orden

**Definición:** registrar los dispositivos pagados de una orden que realizó un cliente del sistema de fábricas, dentro de su catálogo en el sistema de ventas. De esta forma, los dispositivos registrados del sistema de fábricas son obtenidos para el sistema de ventas, de forma de que aparezcan dentro del catálogo de los vendedores y que puedan ser visualizados por los clientes.

**Parámetros:**

- **order:** orden de dispositivos por pagar y que serán registrados en el sistema de ventas.
- **clientName:** nombre del cliente en el sistema de fábricas y vendedor en el de ventas.

**JSON Modelo:**

```
{
  nombre: "222555GB4", // Nombre de la fábrica que será registrada como marca
  _id: { type: Number, required: true },
  clientId: { type: mongoose.Schema.Types.ObjectId, ref: "Client" },
  completed: { type: Boolean, required: true },
  maxDeliveryDate: { type: Date, required: false },
  deliveredDate: { type: Date, required: false },
  canceled: { type: Boolean, required: true },
  completelyPayed: { type: Boolean, required: false },
  isClientOrder: { type: Boolean, required: false },
  devices: [
    {
      factoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Factory" },
      deviceId: { type: mongoose.Schema.Types.ObjectId, ref: "Device" },
      quantity: { type: Number, required: true },
      price: { type: Number, required: true },
      estimatedDeliveryDate: { type: Date, required: false },
      delivered: { type: Boolean, required: true },
      payed: { type: Boolean, required: true },
      deliveredDate: { type: Date, required: false },
      canBeDisplayed: { type: Boolean, required: true },
      displayed: { type: Boolean, required: true },
      lastReported: { type: Date, required: false },
    },
  ],
}
```

**2.1.4 Nueva orden**

**Definición:** generar una nueva orden cuando un cliente desee adquirir nuevos dispositivos para agregar a su catálogo.

**Parámetros:**

- **order:** conjunto de dispositivos por agregar.

**JSON Modelo:**

```
{
  name: { type: String, required: true },
  email: { type: String, required: true },
  shippingTimes: [
    {
      factoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Factory" },
      shippingTime: { type: Number, required: true }
    },
  ],
}
```

**2.1.5 Nueva orden de cliente**

**Definición:** generar una nueva orden cuando un cliente, desde el sistema de ventas, desee adquirir nuevos dispositivos, hacia el sistema de fábricas.

**Parámetros:**

- **order:** conjunto de dispositivos por agregar.

**JSON Modelo:**

```
{
  name: { type: String, required: true },
  email: { type: String, required: true },
  shippingTimes: [
    {
      factoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Factory" },
      shippingTime: { type: Number, required: true }
    }
  ],
}
```

### 2.1.6 Pagar orden

**Definición:** realizar el registro de una nueva fábrica como una marca dentro del sistema de ventas. Cuando una nueva fábrica se registre dentro del sistema de fábricas, se hace el llamado a este web service para que se lo comuniqué al backend de ventas y lo registre como una nueva marca.

**Parámetros:**

- **newBrand:** nombre de la nueva marca en el sistema de ventas.

**JSON Modelo:**

```
{
  nombre: "222555GB4", // Nombre de la fábrica que será registrada como marca
}
```

### 2.1.7 Modificar orden

**Definición:** registrar a un vendedor del sistema de ventas como un cliente dentro del sistema de fábricas.

**Parámetros:**

- **seller:** nombre del vendedor en el sistema de ventas que será registrado como un cliente en el sistema de fábricas.
- **email:** correo electrónico del vendedor.

**JSON Modelo:**

```
{
  name: { type: String, required: true },
  email: { type: String, required: true },
  shippingTimes: [
    {
```

```
        factoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Factory" },
        shippingTime: { type: Number, required: true }
      },
    ],
  }
}
```

### 2.1.8 Pagar orden modificada

**Definición:** registrar los dispositivos pagados de una orden que realizó un cliente del sistema de fábricas, dentro de su catálogo en el sistema de ventas. De esta forma, los dispositivos registrados del sistema de fábricas son obtenidos para el sistema de ventas, de forma de que aparezcan dentro del catálogo de los vendedores y que puedan ser visualizados por los clientes.

**Parámetros:**

- **order:** orden de dispositivos por pagar y que serán registrados en el sistema de ventas.
- **clientName:** nombre del cliente en el sistema de fábricas y vendedor en el de ventas.

**JSON Modelo:**

```
{
  nombre: "222555GB4", // Nombre de la fábrica que será registrada como marca
  _id: { type: Number, required: true },
  clientId: { type: mongoose.Schema.Types.ObjectId, ref: "Client" },
  completed: { type: Boolean, required: true },
  maxDeliveryDate: { type: Date, required: false },
  deliveredDate: { type: Date, required: false },
  canceled: { type: Boolean, required: true },
  completelyPayed: { type: Boolean, required: false },
  isClientOrder: { type: Boolean, required: false },
  devices: [
    {
      factoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Factory" },
      deviceId: { type: mongoose.Schema.Types.ObjectId, ref: "Device" },
      quantity: { type: Number, required: true },
      price: { type: Number, required: true },
      estimatedDeliveryDate: { type: Date, required: false },
      delivered: { type: Boolean, required: true },
      payed: { type: Boolean, required: true },
      deliveredDate: { type: Date, required: false },
      canBeDisplayed: { type: Boolean, required: true },
      displayed: { type: Boolean, required: true },
      lastReported: { type: Date, required: false },
    },
  ],
}
```

### 2.1.9 Pagar orden de cliente

**Definición:** generar una nueva orden cuando un cliente desee adquirir nuevos dispositivos para agregar a su catálogo.

**Parámetros:**

- **order:** conjunto de dispositivos por agregar.



**JSON Modelo:**

```
{
  name: { type: String, required: true },
  email: { type: String, required: true },
  shippingTimes: [
    {
      factoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Factory" },
      shippingTime: { type: Number, required: true }
    }
  ],
}
```

**2.1.10 Actualizar orden de cliente**

**Definición:** generar una nueva orden cuando un cliente, desde el sistema de ventas, desee adquirir nuevos dispositivos, hacia el sistema de fábricas.

**Parámetros:**

- **order:** conjunto de dispositivos por agregar.

**JSON Modelo:**

```
{
  name: { type: String, required: true },
  email: { type: String, required: true },
  shippingTimes: [
    {
      factoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Factory" },
      shippingTime: { type: Number, required: true }
    }
  ],
}
```

**2.1.11 Nueva marca**

**Definición:** realizar el registro de una nueva fábrica como una marca dentro del sistema de ventas. Cuando una nueva fábrica se registre dentro del sistema de fábricas, se hace el llamado a este web service para que se lo comunique al backend de ventas y lo registre como una nueva marca.

**Parámetros:**

- **newBrand:** nombre de la nueva marca en el sistema de ventas.

**JSON Modelo:**

```
{
  nombre: "222555GB4", // Nombre de la fábrica que será registrada como marca
}
```

**2.1.12 Nuevo cliente**

**Definición:** registrar a un vendedor del sistema de ventas como un cliente dentro del sistema

de fábricas.

**Parámetros:**

- ***seller:*** nombre del vendedor en el sistema de ventas que será registrado como un cliente en el sistema de fábricas.
- ***email:*** correo electrónico del vendedor.

**JSON Modelo:**

```
{
  name: { type: String, required: true },
  email: { type: String, required: true },
  shippingTimes: [
    {
      factoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Factory" },
      shippingTime: { type: Number, required: true }
    }
  ],
}
```

### 2.1.13 Pagar orden

**Definición:** registrar los dispositivos pagados de una orden que realizó un cliente del sistema de fábricas, dentro de su catálogo en el sistema de ventas. De esta forma, los dispositivos registrados del sistema de fábricas son obtenidos para el sistema de ventas, de forma de que aparezcan dentro del catálogo de los vendedores y que puedan ser visualizados por los clientes.

**Parámetros:**

- ***order:*** orden de dispositivos por pagar y que serán registrados en el sistema de ventas.
- ***clientName:*** nombre del cliente en el sistema de fábricas y vendedor en el de ventas.

**JSON Modelo:**

```
{
  nombre: "222555GB4", // Nombre de la fábrica que será registrada como marca
  _id: { type: Number, required: true },
  clientId: { type: mongoose.Schema.Types.ObjectId, ref: "Client" },
  completed: { type: Boolean, required: true },
  maxDeliveryDate: { type: Date, required: false },
  deliveredDate: { type: Date, required: false },
  canceled: { type: Boolean, required: true },
  completelyPaid: { type: Boolean, required: false },
  isClientOrder: { type: Boolean, required: false },
  devices: [
    {
      factoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Factory" },
      deviceId: { type: mongoose.Schema.Types.ObjectId, ref: "Device" },
      quantity: { type: Number, required: true },
      price: { type: Number, required: true },
      estimatedDeliveryDate: { type: Date, required: false },
      delivered: { type: Boolean, required: true },
      paid: { type: Boolean, required: true },
      deliveredDate: { type: Date, required: false },
      canBeDisplayed: { type: Boolean, required: true },
      displayed: { type: Boolean, required: true },
    }
  ]
}
```

```
      lastReported: { type: Date, required: false },
    },
  ],
}
```

### 2.1.14 Correo de vendedor

**Definición:** generar una nueva orden cuando un cliente desee adquirir nuevos dispositivos para agregar a su catálogo.

**Parámetros:**

- **order:** conjunto de dispositivos por agregar.

**JSON Modelo:**

```
{
  name: { type: String, required: true },
  email: { type: String, required: true },
  shippingTimes: [
    {
      factoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Factory" },
      shippingTime: { type: Number, required: true }
    }
  ],
}
```

### 2.1.15 Dispositivos de fábricas

**Definición:** generar una nueva orden cuando un cliente, desde el sistema de ventas, desee adquirir nuevos dispositivos, hacia el sistema de fábricas.

**Parámetros:**

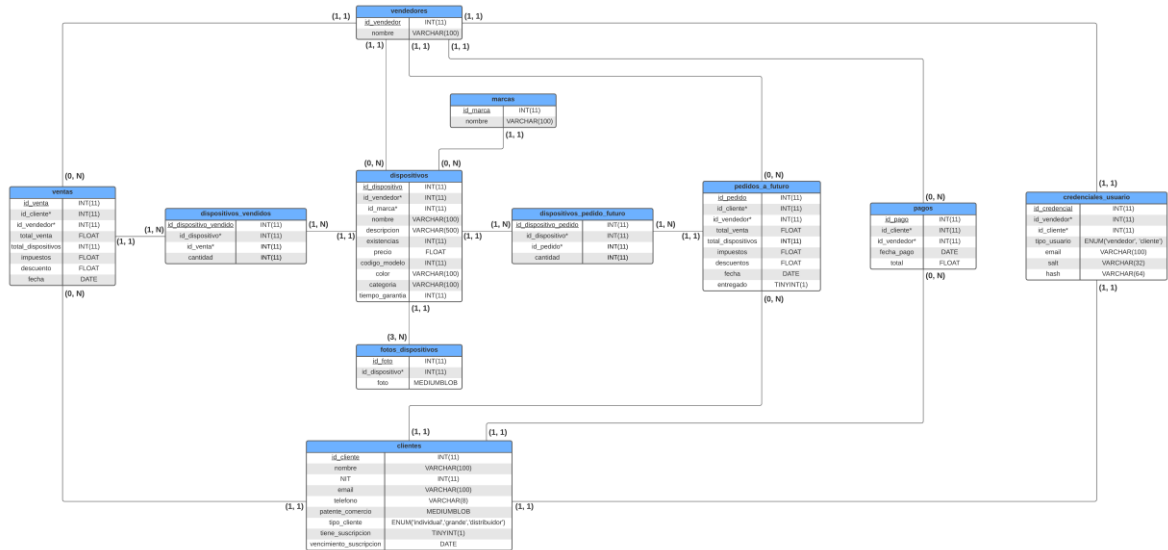
- **order:** conjunto de dispositivos por agregar.

**JSON Modelo:**

```
{
  name: { type: String, required: true },
  email: { type: String, required: true },
  shippingTimes: [
    {
      factoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Factory" },
      shippingTime: { type: Number, required: true }
    }
  ],
}
```

## 2.2 Arquitectura del Sistema

### 2.2.1 Diagrama Entidad-Relación



### 2.2.2 Tablas del sistema de ventas

<i>Dispositivos</i>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
id_dispositivo	int	Llave foránea del dispositivo.
id_vendedor	int	Llave foránea que identifica a cuál vendedor pertenece el dispositivo.
id_marca	int	Llave foránea de la marca del dispositivo.
nombre	varchar(100)	Nombre del dispositivo.
descripcion	varchar(500)	Descripción del dispositivo.
existencias	int	Cantidad del dispositivo en existencia en bodega.
precio	float	Precio de lista del dispositivo.
codigo_modelo	int	Código del modelo del dispositivo.
color	varchar(100)	Color del dispositivo.
categoria	varchar(100)	Categoría del dispositivo.
tiempo_garantia	int	Años/meses/días de garantía.

Tabla con información de los dispositivos

<i>Fotografías dispositivos</i>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
id_fotografia	int	Llave primaria de la fotografía.
id_dispositivo	int	Llave foránea que identifica al dispositivo asociado con la fotografía.
foto	medblob	Fotografía del dispositivo.

Tabla con las fotografías de los dispositivos.

<i><b>Marcas</b></i>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
id_marca	int	Llave primaria de la marca.
nombre	varchar(100)	Nombre de la marca.

Tabla del catálogo de marcas.

<i><b>Vendedores</b></i>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
id_vendedor	int	Llave primaria.
nombre	varchar(100)	Nombre de la tienda/vendedor.

Tabla con los datos de los vendedores.

<i><b>Clientes</b></i>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
id_cliente	int	Llave primaria.
nombre	varchar(100)	Nombre del cliente.
nit	int	NIT del gran cliente o distribuidor mayorista. (Nullable).
Email	Varchar(100)	Email del gran cliente o distribuidor mayorista. Puede ser distinto al email de las credenciales. (Nullable).
telefono	varchar(8)	Teléfono del gran cliente o distribuidor mayorista. (Nullable).
patente_de_comercio	medblob	Patente de comercio (imagen) del gran cliente o distribuidor mayorista. (Nullable).
tipo_cliente	enum	Tipo del cliente (individual, gran cliente o distribuidor mayorista).
tiene_suscripcion	boolean	Indica si tiene o no suscripción.
vencimiento_suscripcion	date	Fecha del vencimiento de la suscripción. (Nullable).

Tabla con los datos de los clientes.

<i><b>Ventas</b></i>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
id_venta	int	Llave primaria
id_cliente	int	Llave foránea del cliente que compró los dispositivos.
id_vendedor	int	Llave foránea del vendedor.
total_venta	float	Total facturado al cliente.
total_dispositivos	int	Total de dispositivos vendidos.
impuestos	float	Total de impuestos.
descuentos	float	Descuento aplicados a la venta.
fecha	date	Fecha en que se realizó la venta.

Tabla con la información de las ventas realizadas.

<i>Dispositivos_vendidos</i>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
id_dispositivo_vendido	int	Llave primaria.
id_venta	int	Llave foránea de la venta.
id_dispositivo	int	Llave foránea del dispositivo.
cantidad	Int	Cantidad de dispositivos vendidos.

Tabla con los datos de los dispositivos adquiridos en una venta.

<i>Pedidos_a_futuro</i>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
id_pedido	int	Llave primaria del pedido.
id_cliente	int	Llave foránea del distribuidor que adquirirá los dispositivos.
id_vendedor	int	Llave foránea del vendedor.
total_venta	float	Total facturado al cliente.
total_dispositivos	int	Total de dispositivos vendidos.
impuestos	float	Total de impuestos.
descuentos	float	Descuento aplicados a la venta.
fecha	date	Fecha en que se realizó el pedido.
entrega	boolean	Indica si el pedido ya ha sido entregado.

Tabla con la información de los pedidos a futuro de los distribuidores mayoristas.

<i>Dispositivos_pedido_futuro</i>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
id_disp_pedido	int	Llave primaria.
id_dispositivo	int	Llave foránea del dispositivo.
id_pedido	int	Llave foránea del pedido a futuro.
cantidad	int	Cantidad del dispositivo adquirido en el pedido a futuro.

Tabla con la información de los dispositivos que serán adquiridos en un pedido futuro.

<i>Pagos</i>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
id_pago	int	Llave primaria del pago
id_cliente	int	Llave foránea del cliente.
id_vendedor	int	Llave foránea del vendedor.
fecha_pago	date	Fecha en que se realizó el pago.
total	float	Total pagado.

Tabla con la información de los pagos.

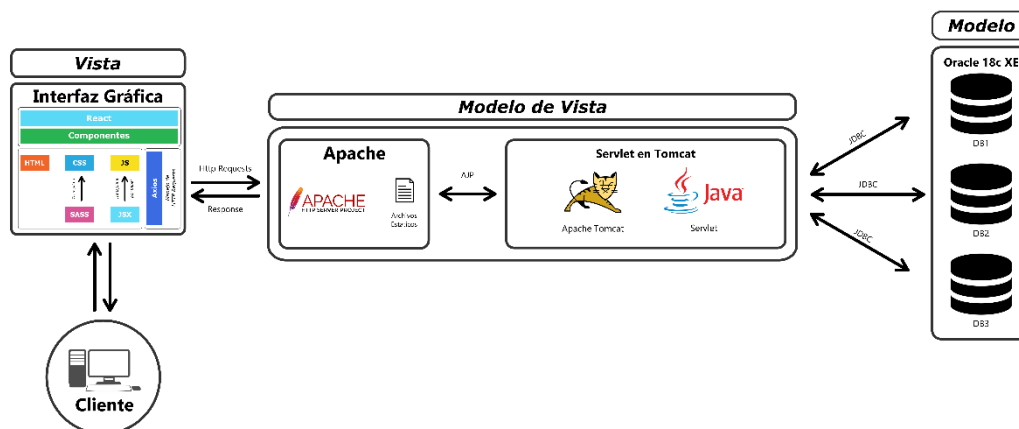
<i>Credenciales_usuarios</i>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
id_usuario	int	Llave primaria del usuario.
id_vendedor	int	Llave foránea del vendedor (null si el usuario es cliente).
id_cliente	int	Llave foránea del cliente (null si el usuario es vendedor).
tipo_usuario	boolean/enum	Tipo del usuario (vendedor o cliente). Dato para poder dirigirlo a la página correspondiente.
email	varchar(100)	Email del usuario utilizado para el login.
salt	varchar(32)	Sal criptográfica.
hash	varchar(64)	Digest resultado de hashear la concatenación de contraseña y sal.

Tabla con la información de los usuarios.

### 2.2.3 Stack de tecnologías utilizados

- Front-end: **React**
- Sistema de fábrica (MERN stack):
  - **MongoDB**: como base de datos no relacional de documentos.
  - **Express.js**: como framework web de Node.js.
  - **Node.js**: como entorno del backend.
- Sistema de ventas:
  - **Oracle Database 18c XE**: como base de datos relacional.
  - **Apache HTTP Server**: como servidor web de archivos estáticos para la entrega de archivos de JSP, actuando como proxy de las peticiones del servlet desplegado en Tomcat.
  - **Tomcat**: como servidor de aplicaciones.
  - **Servlet**: pequeño programa de Java servido por Tomcat. Estará configurado para recibir y responder los requests HTTP, a la vez que gestiona las operaciones CRUD en la base de datos.
  - Protocolos/conectores de comunicación:
    - **AJP**: (Apache JServ Protocol). protocolo que permite el envío de solicitudes desde un servidor web, como Apache, a un servidor de aplicaciones, como Tomcat.
    - **JDBC**: (Java Database Connectivity) es un estándar de Java que proporciona la interfaz para conectarse desde Java a bases de datos relaciones, como Oracle 18c XE.

## 2.2.4 Diagrama de arquitectura del sistema de ventas



*Arquitectura del Sistema de Ventas*