

PRUEBA TECNICA

COLOMBIA COMPRA EFICIENTE

**REALIZADO POR:
CARLOS MARIO DURAN CAMELO**

21 de abril de 2022

Tabla de contenido

1.	BASE DE DATOS	3
1.1.	Descripción de Tablas.....	3
1.2.	Entidad Relación.....	3
2.	API DESARROLLADO EN SPRING BOOT	4
2.1.	Spring Tools 4 Eclipse	4
2.2.	Spring Boot 2.6.6.....	5
2.3.	Estructura de las API	5
2.4.	Conexión a BASE DE DATOS	7
2.5.	Realización de pruebas de servicios mediante “POSTMAN”	8
3.	FRONT END.....	9
3.1.	Estructura del Desarrollo	10
4.	MANUAL DEL PROGRAMA	11
4.1.	Inicio de Usuario.....	11
4.2.	Listado de concepto	12
4.3.	Formulario de eliminación y edición.....	12

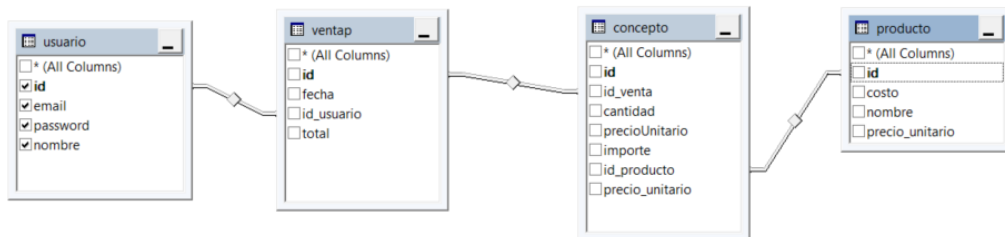
1. BASE DE DATOS

Se realizó la entidad relación en SQL Server 2012 en donde se tiene las siguientes tablas:

1.1. Descripción de Tablas

- Usuario: Es donde se almacena la información del usuario para ingresar al sistema.
- Producto: Se almacena la información del producto
- Ventap: Encontraremos la información principal de la venta.
- Concepto: Se almacena el detalle de la venta.

1.2. Entidad Relación

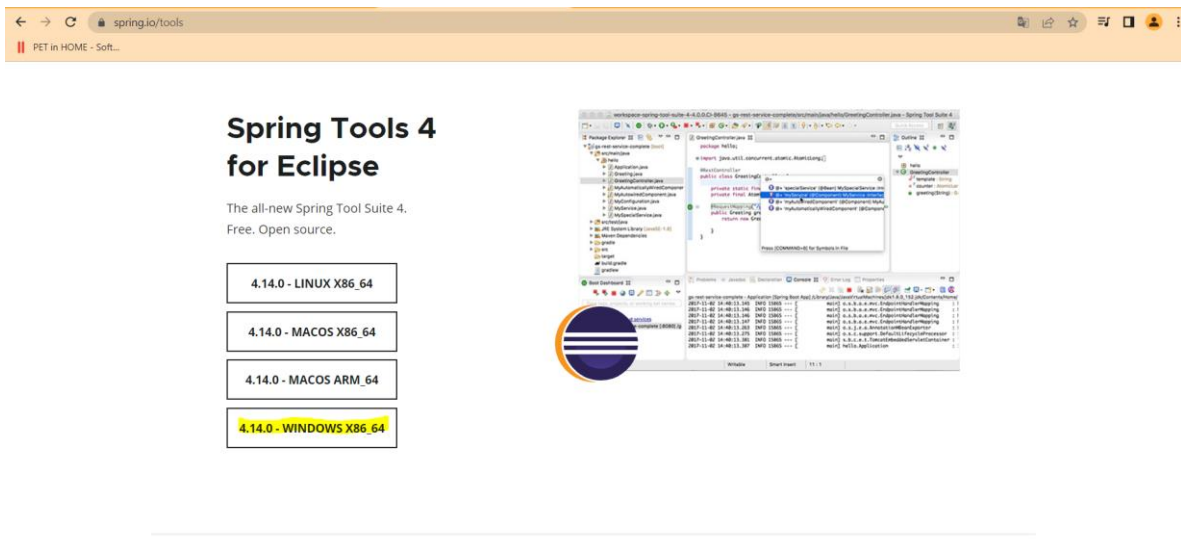


2. API DESARROLLADO EN SPRING BOOT

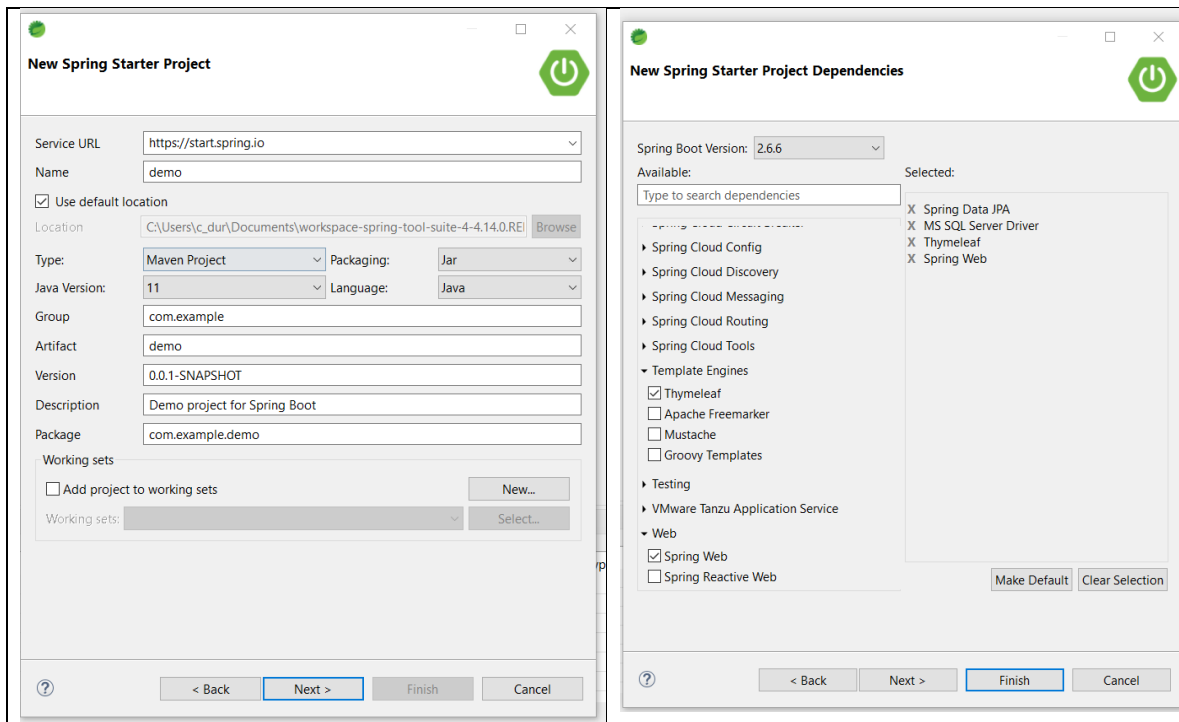
La versión que se utilizó utilizada para el desarrollo del api fueron los siguientes:

2.1. Spring Tools 4 Eclipse

Se encuentra la descarga en la siguiente dirección: <https://spring.io/tools>



2.2. Spring Boot 2.6.6



La versión de java instalada es: Java 8

Nombre de instalador: jdk-8u201-windows-x64

2.3. Estructura de las API

Se realizó la creación de cuatro "Package" que son:

- Com.init.products: Donde se inicializa la aplicación y configurando temas como permisos SSL, esto con el fin de evitar inconvenientes para el consumo del mismo.
- Com.init.products.dao: en esta encontramos cuatro "interfaces" en donde se crea los constructores de cada servicio.
 - ConceptoDAO.java
 - ProductsDAO.java
 - SaleDAO.java
 - UserDAO.java

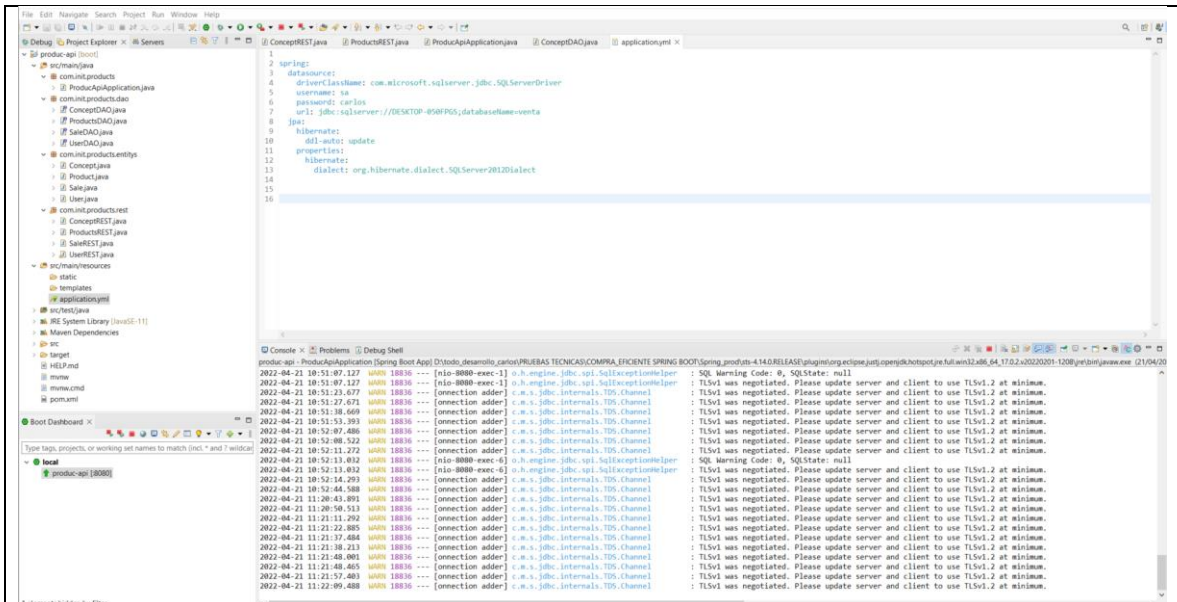
Nota: para cada una de las tablas creadas en base de datos se realizó una interface.

- Com.init.products.entitys: En esta encontraremos las entidades correspondientes a cada tabla las cuales son:
 - ConceptoDAO.java
 - ProductsDAO.java
 - SaleDAO.java
 - UserDAO.java
- Com.init.products.rest: podemos encontrar en este paquete todos los servicios expuestos y el cual se realizó para cada una de las tablas creadas en base de datos.
 - ConceptoREST.java
 - ProductsREST.java
 - SaleDAO.REST
 - UserDAO.REST

2.4. Conexión a BASE DE DATOS

La cadena de conexión se encuentra en: src/main/resources

```
spring:
  datasource:
    driverClassName: com.microsoft.sqlserver.jdbc.SQLServerDriver
    username: sa
    password: carlos
    url: jdbc:sqlserver://DESKTOP-050FPGS;databaseName=venta
jpa:
  hibernate:
    ddl-auto: update
  properties:
    hibernate:
      dialect: org.hibernate.dialect.SQLServer2012Dialect
```

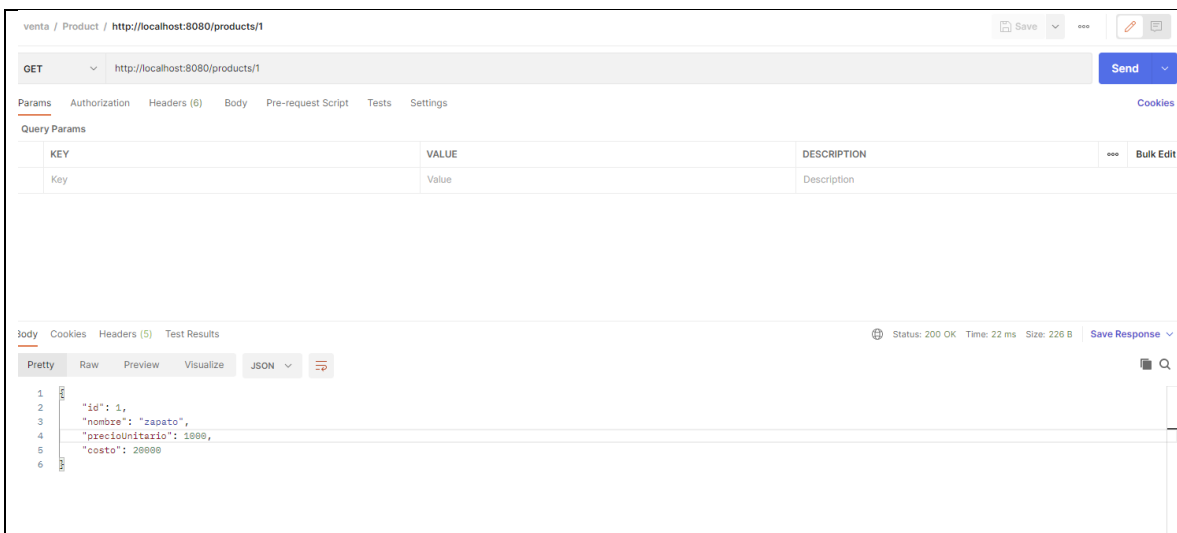
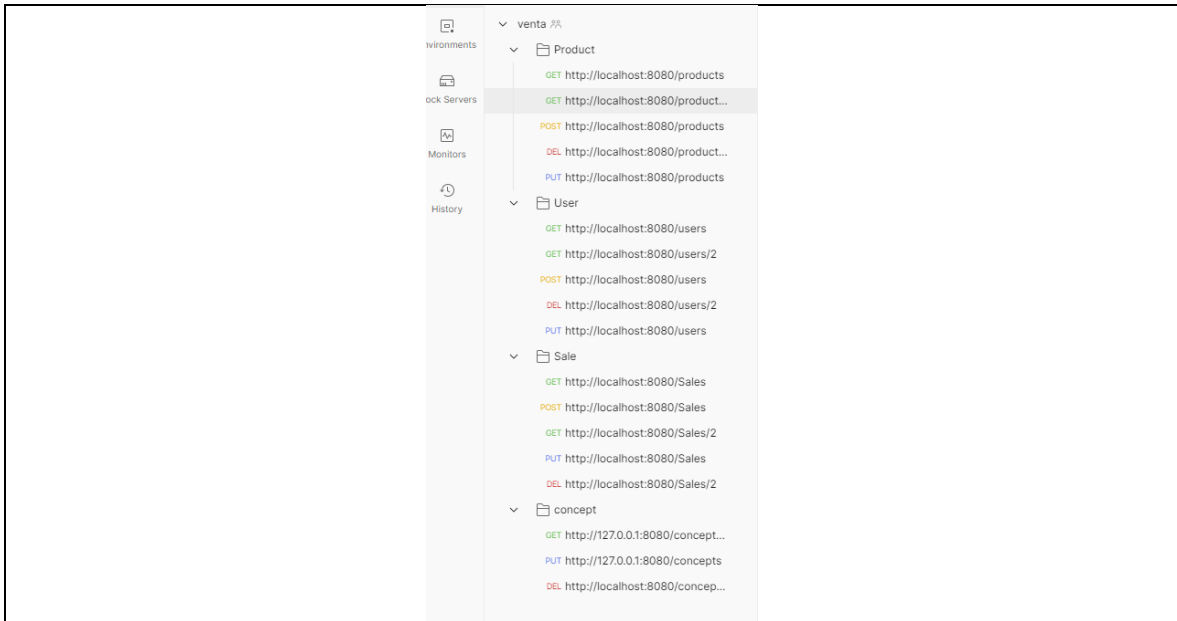


NOTA: Para cada una de las tablas se realizó la creación del CRUD, en donde podemos encontrar el consumo de los servicios por (GET, POST, PUT Y DELETE)

2.5. Realización de pruebas de servicios mediante “POSTMAN”

En la colección llamada venta, encontraremos 4 carpetas el cual fue nombrada como:

- **Product:** Servicios correspondientes a los productos.
- **User:** Servicios correspondientes a la tabla de usuario.
- **Sale:** Servicios correspondientes a la tabla de venta.
- **Concept:** Servicios correspondientes a la tabla de Concepto.

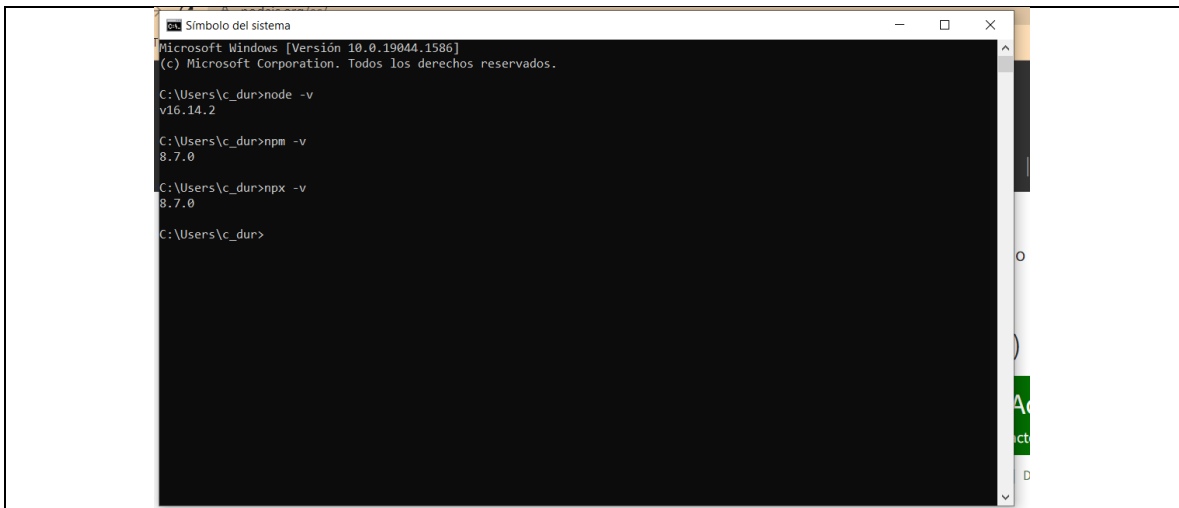


3. FRONT END

Se realizó el desarrollo del Front-End en React js

Aplicación y versiones instaladas:

- **Node.js:** Versión 16.14.2
- **Npm:** Versión 8.7.0
- **Npx:** Versión 8.7.0
- **Visual Studio Code:** Versión 1.66.2



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.1586]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\c_dur>node -v
v16.14.2

C:\Users\c_dur>npm -v
8.7.0

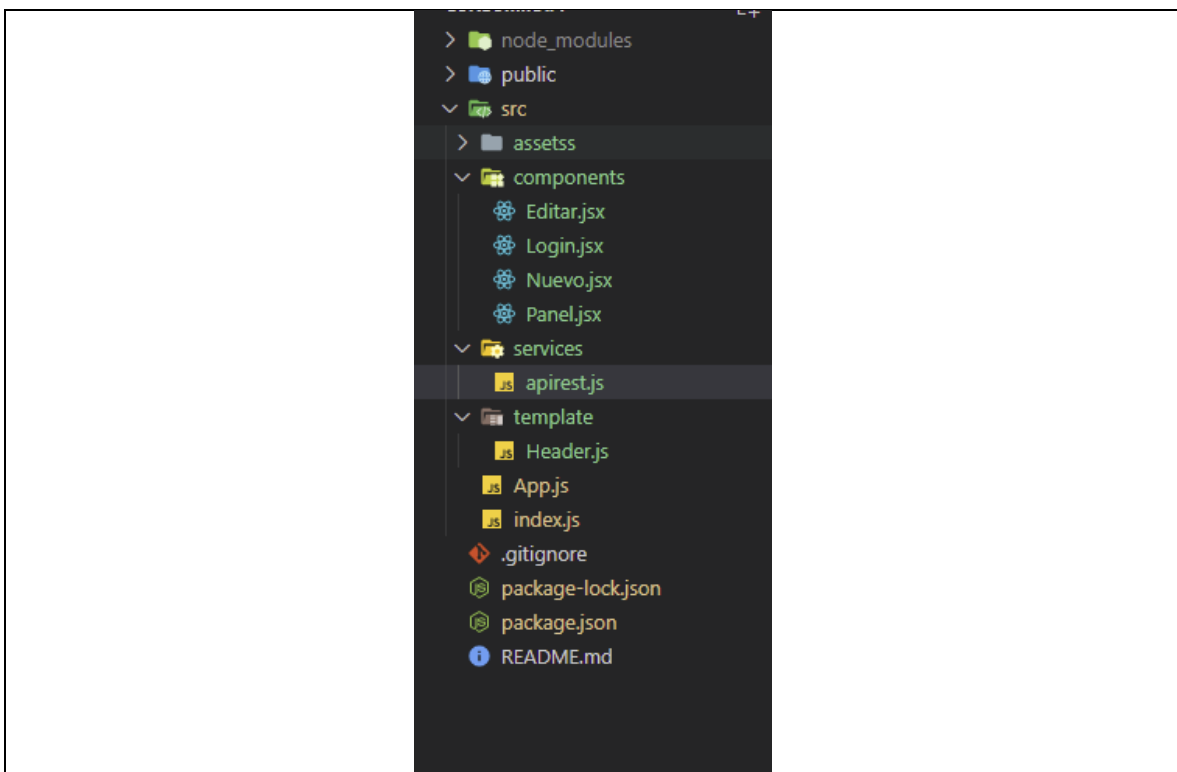
C:\Users\c_dur>npx -v
8.7.0

C:\Users\c_dur>
```

3.1. Estructura del Desarrollo

La estructura del desarrollo esta compuesto de la siguiente manera:

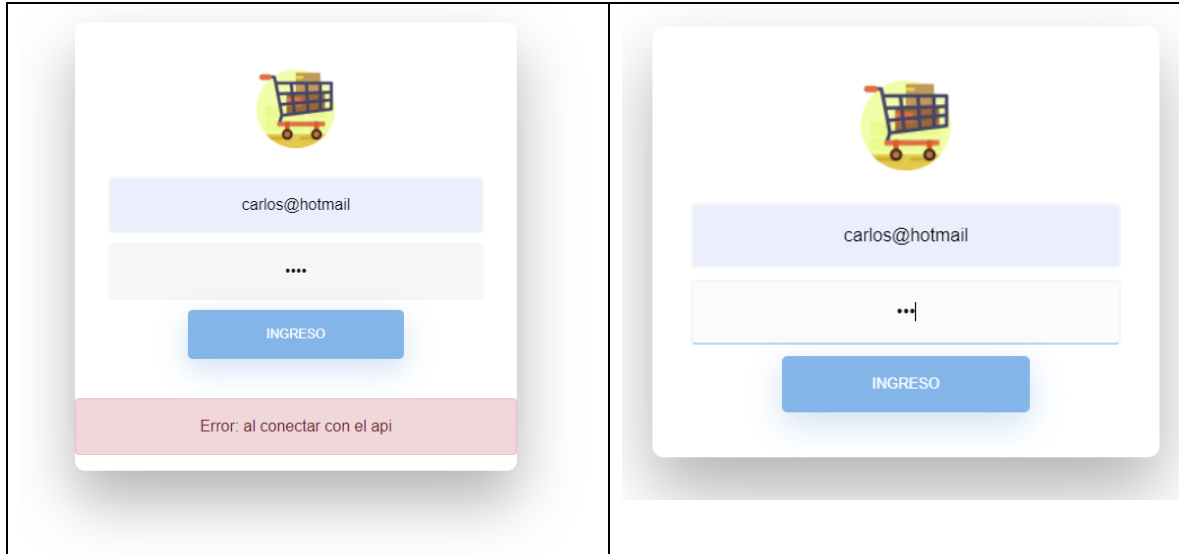
- **Components:** en esta encontraremos los componentes “.jsx” para la editar, nuevo, login y panel.
Nota: Para el componente nuevo, no tuve el tiempo suficiente para realizar el desarrollo.
- **Services/apirest.js:** en esta archivo javascript encontramos la ruta para el consumo del servicio.
- **Template/Header:** en esta ruta encontramos el archivo javascript en donde esta creado el header.



4. MANUAL DEL PROGRAMA

Consta de 3 módulos y el cual se describe a continuación:

4.1. Inicio de Usuario



En esta valida si el usuario se encuentra en base de datos para dar acceso al módulo de lo contrario arroja un mensaje.

4.2. Listado de concepto

Consumir Api				
PRODUCTOS				
Id Venta	Cantidad	Precio Unitario	Id Producto	Producto
1	1	10000	1	zapato
<button>salir</button>				

Nos lista la cantidad de datos que tenemos en la tabla concepto, la darle clic sobre el registro nos lleva a otro modulo.

4.3. Formulario de eliminación y edición.

Consumir Api	
Editar Concepto	
TOTAL EN VENTA: 10000	
ID VENTA: 1	
Precio Unitario	<input type="text" value="Precio Unitario"/>
Cantidad	<input type="text" value="1"/>
Valor de Importacion	<input type="text" value="0"/>
Producto	<input type="text" value="1"/>
<button>Editar</button>	<button>Eliminar</button> <button>salir</button>

En este formulario permite editar información y de igual forma eliminarla.