



CICLO 4 DESARROLLO DE APLICACIONES WEB

PROYECTO DE DESARROLLO DE APLICACIÓN WEB PARA GESTIONAR LA CADENA DE ALMACENES “LA GENÉRICA”

TABLA DE CONTENIDO

Introducción	5
Descripción del caso	5
Parte 1: Diagrama de Arquitectura y Microservicios	5
Parte 2: Especificación Funcional	7
Parte 2: Organización del Equipo Scrum	9
Parte 3: Organización de los Sprints	9
Sprint 1	9
Historias de Usuario	11
Interfaz Gráfica	12
Conjunto de Pruebas	13
Sprint 2	13
Historias de Usuario	14
Interfaz Gráfica	15
Conjunto de Pruebas	15
Sprint 3	16
Historias de Usuario	17
Interfaz Gráfica	18
Conjunto de Pruebas	18
Sprint 4	20
Historias de Usuario	20
Interfaz Gráfica	21
Conjunto de Pruebas	22
Sprint 5	23
Historias de Usuario	23
Interfaz Gráfica	23
Conjunto de Pruebas	24
Parte 4: Especificaciones Técnicas	24
Lenguajes de programación y stack tecnológico de desarrollo	24

Especificación de diseño de la base de datos no relacional.	25
Especificación de la API para conexión del Frontend con el Backend de la aplicación	26
Diagrama de Despliegue de Contenedores Docker.	34

Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
21/08/2021	1	Ing. Ricardo D. Camargo	UEB	Descripción inicial de proyecto de aplicación Web para gestionar la cadena de almacenes “La Genérica”.
29/08/2021	2	Ing. Ricardo D. Camargo	UEB	Definición del diagrama de arquitectura de microservicios para ciudades satélite y para Bogotá. Inicio de descripción funcional.
05/09/2021	3	Ing. Ricardo D. Camargo	UEB	Definición de los requerimientos funcionales por módulo, según los servicios definidos en la arquitectura general.
09/10/2021	4	Ing. Ricardo D. Camargo	UEB	Modificación de módulos, descripción de los cinco sprints completos, se agrega diagrama de despliegue de contenedores Docker.



Introducción

El propósito de este documento es dar las especificaciones para el diseño y desarrollo de la aplicación web, escrita en lenguaje Java para Web, mediante el modelo de arquitectura Single Page Application (SPA), para gestionar las transacciones comerciales de la cadena de almacenes de grandes superficies llamada “La Genérica”.

Descripción del caso

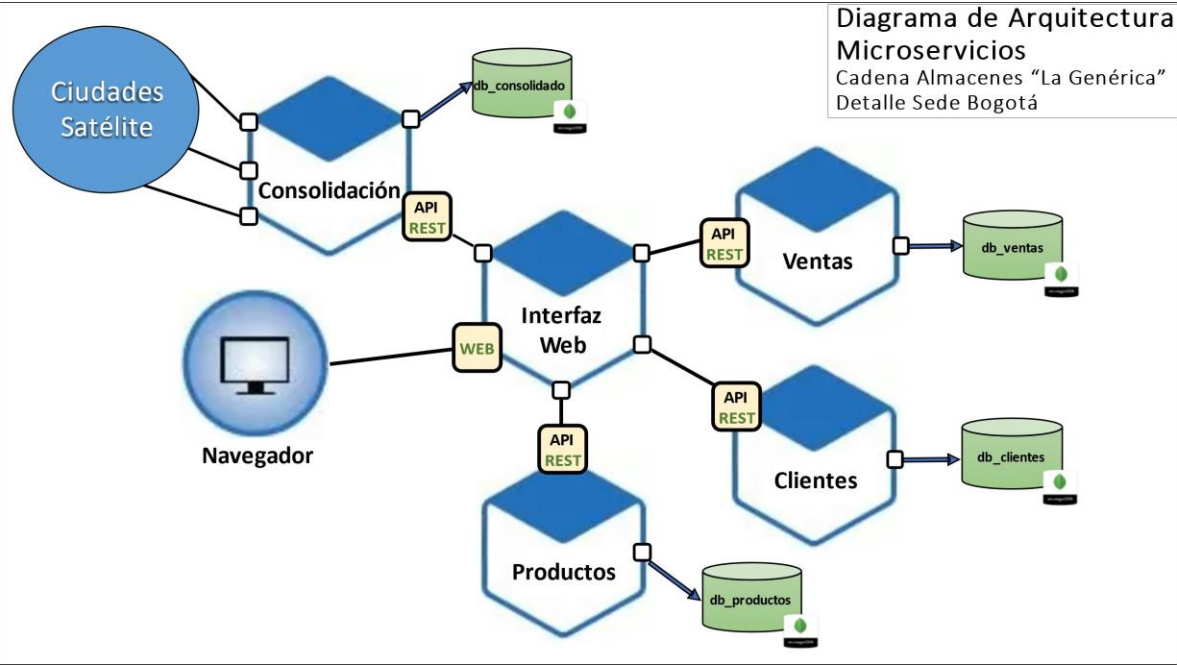
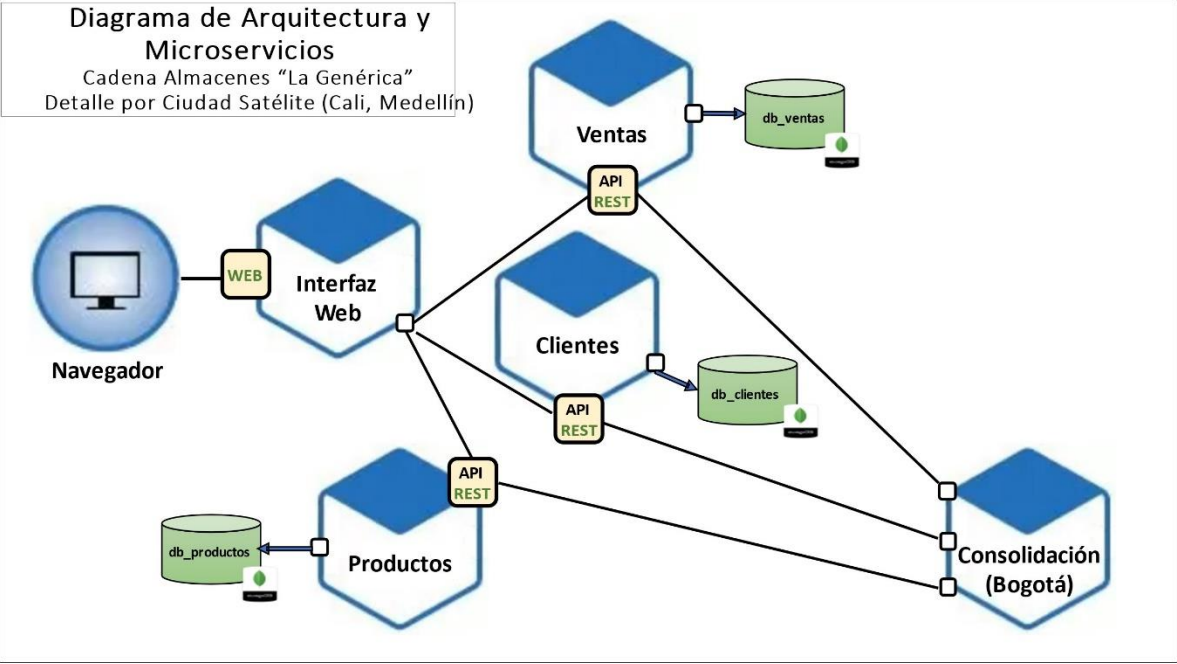
Cuando los negocios crecen, diversifican sus productos y servicios, y se expanden geográficamente, es necesario realizar consideraciones y replanteamientos de diseño para su crecimiento a nivel de los sistemas de información con los que cuentan para efectos de que, igualmente, crezcan con ellos y brinden respuestas a las nuevas realidades para mantener la competitividad.

En virtud de lo anterior, retomaremos el proyecto realizado en el ciclo 3, el cual, como sabemos era el desarrollo de software para una tienda que provee productos y servicios, que denominamos “tienda genérica”. Ahora, la tienda se convierte en la cadena de almacenes llamada “La Genérica”, la cual surgió como resultado del gran éxito que tuvo el negocio, por lo que la gerencia determinó ampliar su portafolio de productos y servicios, y expandir su operación inicial, la cual estaba en Bogotá, abriendo sucursales en Cali y Medellín.

EL EQUIPO de beneficiarios del Ciclo 4A, ha sido contratado por la cadena de almacenes para la realización el desarrollo de la aplicación mediante la implementación de los conceptos vistos en este Ciclo, y de acuerdo con las especificaciones de arquitectura, y cumpliendo con los requerimientos que en este documento se estipulan.

Parte 1: Diagrama de Arquitectura y Microservicios

La división de Tecnología de la cadena de almacenes “La Genérica”, realizó el diseño de arquitectura de la aplicación, basado en las ciudades y la especificación de servicios que van a manejar en cada ciudad, donde se aprecia Bogotá, Cali, y Medellín, cada uno con una interfaz Web (frontend), y la definición de microservicios de registro de ventas, registro de compras, registro de productos, y el microservicio de consolidación hacia Bogotá. En cada microservicio, se detalla la base de datos MongoDB que estará formando parte, y estará haciendo las operaciones de CRUD.



Parte 2: Especificación Funcional

Se identifica a continuación los requerimientos para la realización del software, consistente en los siguientes módulos y requerimientos por módulo:

1. Módulo de Gestión de Productos.

EL EQUIPO deberá desarrollar el módulo de gestión de los productos que se venden en la tienda, para lo cual se deberá cargar estos productos de un archivo plano (de texto), con la siguiente estructura de datos:

NOMBRE DEL DATO	TIPO DE DATO	LONGITUD
código_producto	BIGINT	20
nombre_producto	VARCHAR	50
nitproveedor	BIGINT	20
precio_compra	DOUBLE	
ivacompra	DOUBLE	
precio_venta	DOUBLE	

El sistema debe validar que el archivo sea un archivo separado por comas (CSV), para que sea cargado correctamente. Se anexa con el presente documento un archivo de muestra de cómo debe conformarse, y una tabla de cómo de ser el formato de cada dato

codigo_producto	nombre_producto	nit_proveedor	precio_compra	iva	precio_venta
1	Queso	123456789	12000	19	14160
2	Manzanas	987654321	18108	19	21549
3	Plátanos	456789012	29681	19	35320
4	Lechuga	321098765	29788	19	35448
5	Tomates	654321098	12739	19	15159
6	Calabaza	876543210	21315	19	25365
7	Apio	210987654	19249	19	22906
8	Pepino	543210987	10958	19	13040
9	Champiñones	765432109	11046	19	13145
10	Leche	901234567	21150	19	25169
11	Queso	345678901	26571	19	31619
12	Huevos	678901234	12445	19	14810
13	Requesón	012345678	14329	19	17052
14	Crema agria	234567890	14856	19	17679
15	Yogur	567890123	14941	19	17780

1	Melocotones	1	25505	19	30351
2	Manzanas	3	18108	19	21549
3	Plátanos	4	29681	19	35320
4	Lechuga	3	29788	19	35448
5	Tomates	1	12739	19	15159
6	Calabaza	1	21315	19	25365
7	Apio	2	19249	19	22906
8	Pepino	2	10958	19	13040
9	Champiñones	2	11046	19	13145
10	Leche	5	21150	19	25169
11	Queso	5	26571	19	31619
12	Huevos	2	12445	19	14810
13	Requesón	1	14329	19	17052
14	Crema agria	1	14856	19	17679
15	Yogur	5	14941	19	17780

16	Ternera	5	29335	19	34909
17	Salmón salvaje	5	11878	19	14135
18	Patas de cangrejo	1	29951	19	35642

Una vez hecha esta validación, el archivo será cargado desde la página Web, y se almacenará en el folder llamada db_productos, la cual tendrá, por supuesto, los mismos tipos de datos del archivo. El sistema deberá validar que el NIT del proveedor que se indica en el archivo, corresponda a un NIT que exista en la base de datos

2. Módulo de Gestión de Ventas.

EL EQUIPO deberá desarrollar el módulo de gestión de las ventas que se realicen en la tienda. El sistema, buscará los datos del cliente por cédula. Posteriormente, el sistema permitirá escribir el código del producto, y se visualizará el nombre de este en pantalla. Se digitará la cantidad a vender, y generará el valor total de venta por producto. El sistema permitirá que se realice la misma operación con otros productos hasta que se le dé la opción de totalizar. En ese momento, el sistema calculará el valor del total de IVA para tres (3) productos, según el valor de IVA definido para cada producto, y luego dará el valor total con IVA.

El sistema deberá registrar la venta con los siguientes datos: código de venta – este es un valor consecutivo, cedula del cliente, valor total de venta, valor de IVA, valor total más IVA, código de producto, cantidad, valor unitario (valor de venta), y valor total por producto en una colección llamada **db_ventas**. Finalmente, el sistema deberá mostrar un mensaje de confirmación de la transacción.

3. Módulo de Registro de Compras.

EL EQUIPO deberá desarrollar el módulo de gestión de los clientes de la tienda, para lo cual se deberán tener las opciones de crear cliente, leer clientes, actualizar datos de cliente, y borrar cliente. Estos datos se almacenarán en la colección llamada **db_clientes**, y los datos a gestionar son: cedula, nombre completo, dirección, teléfono, y correo electrónico.

4. Módulo DE Interfaz Web

EL EQUIPO deberá desarrollar el módulo que permita la interacción del sistema con el usuario, para lo cual, se deberá construir el ingreso al sistema, una vez se haya realizado la validación por nombre de usuario y contraseña. Deberá tenerse un usuario por defecto con el nombre de **admininicial**, y contraseña **admin123456** para su ingreso.

Finalmente, en este módulo se hará la integración de los módulos de productos, compras, y ventas, así como el de la consolidación de tiendas satélite (para el caso de Bogotá), para que el usuario pueda realizar todos los procesos de interacción mediante opciones de menú.

5. Módulo de Consolidación.

EL EQUIPO deberá desarrollar el módulo de consolidación de las compras y ventas de las tiendas satélite, y que sean registradas en la ciudad de Bogotá. El sistema deberá realizar las consultas de compras y ventas del día de las ciudades satélites, las cuales se registrarán en la colección llamada **db_consolidado**, con los siguientes datos: ciudad satélite y valor total ventas.

Parte 2: Organización del Equipo Scrum

EL EQUIPO se conformará por cinco (5) beneficiarios que asistan al Ciclo 3, el cual realizará el desarrollo del software objeto de este documento, mediante la aplicación del marco de trabajo Scrum, para lo cual, los miembros de este tendrán los siguientes roles y responsabilidades:

Nº	ROL	RESPONSABILIDAD
1	Product Owner	Maneja la definición funcional del producto de software completo a desarrollar, y será el punto de contacto con el cliente para validar la aceptación de cada incremento de producto con el cliente.
2	Scrum Master	Maneja, difunde y controla la ejecución correcta del marco de trabajo Scrum en todos los miembros del EQUIPO.
3	Development Team: Arquitecto	Realiza el diseño de arquitectura del software y de la base de datos.
4	Development Team: Deasrollador	Realiza la codificación del software.
5	Development Team: Control de Calidad (QA)	Realiza las pruebas al software establecidas en este documento.

Cabe mencionar que, por la naturaleza del proyecto y el número de participantes del equipo, todos ellos (as) tienen el rol de desarrolladores por defecto, de lo cual, en el momento que se estime necesario el EQUIPO podrá sumar más desarrolladores, conforme a los requerimientos y la ejecución del sprint. El Scrum Master tendrá la responsabilidad de realizar el seguimiento y control del proyecto, para lo cual, convocará las reuniones diarias de seguimiento (daily standup meeting) , en coordinación con el formador, y se utilizará la herramienta Trello, para realizar el seguimiento del desarrollo de los sprints.

Parte 3: Organización de los Sprints

Sprint 1

Microservicio de Productos.

EL EQUIPO deberá desarrollar el microservicio de gestión de los productos que se venden en la tienda, para lo cual se deberá cargar estos productos de un archivo plano (de texto), con la siguiente estructura de datos:

NOMBRE DEL DATO	TIPO DE DATO	LONGITUD
código_producto	BIGINT	20
nombre_producto	VARCHAR	50
nitproveedor	BIGINT	20
precio_compra	DOUBLE	
ivacompra	DOUBLE	
precio_venta	DOUBLE	

El sistema debe validar que el archivo sea un archivo separado por comas (CSV), para que sea cargado correctamente. Se anexa con el presente documento un archivo de muestra de cómo debe conformarse, y una tabla de cómo de ser el formato de cada dato.

codigo_producto	nombre_producto	NIT_Proveedor	precio_compra_IVA	precio_venta
1	Melocotones	1	25505	30351
2	Manzanas	3	18108	21549
3	Plátanos	4	29681	35320
4	Lechuga	3	29788	35448
5	Tomates	1	12739	15159
6	Calabaza	1	21315	25365
7	Apio	2	19249	22906
8	Pepino	2	10958	13040
9	Champiñones	2	11046	13145
10	Leche	5	21150	25169
11	Queso	5	26571	31619
12	Huevos	2	12445	14810
13	Requesón	1	14329	17052
14	Crema agria	1	14856	17679
15	Yogur	5	14941	17780
16	Ternera	5	29335	34909
17	Salmón salvaje	5	11878	14135
18	Patas de cangrejo	1	29951	35642

1	Melocotones	1	25505	19	30351
2	Manzanas	3	18108	19	21549
3	Plátanos	4	29681	19	35320
4	Lechuga	3	29788	19	35448
5	Tomates	1	12739	19	15159
6	Calabaza	1	21315	19	25365
7	Apio	2	19249	19	22906
8	Pepino	2	10958	19	13040
9	Champiñones	2	11046	19	13145
10	Leche	5	21150	19	25169
11	Queso	5	26571	19	31619
12	Huevos	2	12445	19	14810
13	Requesón	1	14329	19	17052
14	Crema agria	1	14856	19	17679
15	Yogur	5	14941	19	17780
16	Ternera	5	29335	19	34909
17	Salmón salvaje	5	11878	19	14135
18	Patas de cangrejo	1	29951	19	35642

Una vez hecha esta validación, el archivo será cargado desde la interfaz Web, y se almacenará en la colección llamada db_productos, la cual tendrá, por supuesto, los mismos tipos de datos del archivo. El sistema deberá validar que el NIT del proveedor que se indica en el archivo, corresponda a un NIT que exista en la base de datos.

Microservicio Interfaz Web (Primer Incremento de producto)

Se deberá construir el microservicio de interfaz web para realizar el login del sistema y generar la interacción con el usuario del microservicio de productos.

Para realizar la interfaz de ingreso al sistema, se hará la validación por nombre de usuario y contraseña, de lo cual el nombre de usuario correcto será **admininicial**, y la contraseña **admin123456** para su ingreso. La especificación de interfaz gráfica se encuentra detallada más adelante en este sprint.

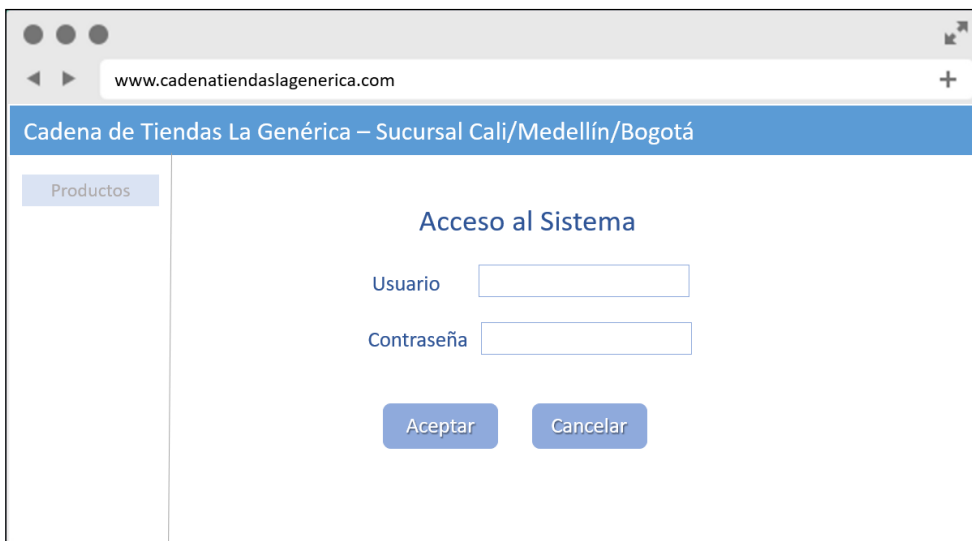
Historias de Usuario

Identificador (ID) de la historia	Rol	Característica / Funcionalidad	Razón / Resultado
HU-001	Como administrador	Necesito ingresar al sistema con el usuario inicial admininicial y la contraseña admin123456 .	Con la finalidad de validar el funcionamiento del login del sistema, la interfaz gráfica y que el usuario inicial opere correctamente.
HU-002	Como administrador	Necesito subir al sistema un archivo texto separado por comas (csv) con los datos de productos, según formato definido (1).	Con la finalidad de cargar a la tabla de productos del sistema, reemplazando los productos que hubiere previamente, y poderlos utilizar para realizar ventas.

(1) Indicado previamente en la descripción funcional

Interfaz Gráfica

Para la realización de la interfaz gráfica en términos de visualización, si bien en las siguientes gráficas se determina el texto “Cali/Medellín/Bogotá”, al momento de realizar el microservicio, deberá configurarse para que solamente aparezca una sola ciudad.



www.cadenatiendaslagenerica.com

Cadena de Tiendas La Genérica – Sucursal Cali/Medellín/Bogotá

Productos

Acceso al Sistema

Usuario

Contraseña

Aceptar Cancelar



www.cadenatiendaslagenerica.com

Cadena de Tiendas La Genérica – Sucursal Cali/Medellín/Bogotá

Productos

Nombre del Archivo Examinar

Cargar

Conjunto de Pruebas

ID Prueba	Caso de Prueba	Descripción	Acción de Entrada	Resultado Esperado
SP1-QA-1	Ingreso correcto	El usuario debe poder registrarse correctamente, si los campos de usuario y contraseña son correctos.	Escribir el nombre de usuario inicial admininicial y/o usuario creado, y la contraseña correspondiente.	Si es el usuario admininicial , o un usuario ya creado, el sistema dará ingreso al menú general.
SP1-QA-2	Ingreso incorrecto por error en el usuario y/o contraseña.	El usuario debe recibir un mensaje de error si, al consultar en la base de datos no se recupera la información, o si los datos de admininicial son errados.	Escribir el usuario y/o contraseña con errores de tipeo, o no escribir alguno de los datos solicitados.	Debe generar un mensaje de error con el error de "usuario o contraseña errados, intente de nuevo".
SP1-QA-3	Carga exitosa del archivo	El usuario realiza una carga de archivo correcta.	El usuario seleccionará un archivo al presionar el botón "Examinar", el archivo corresponde al formato CSV, y tiene la estructura de datos correcta.	Debe generar un mensaje de información "Archivo Cargado Exitosamente". Luego, debe borrar los anteriores datos que tenga la tabla Productos, e insertar los datos leídos, previa validación.
SP1-QA-4	Carga Fallida del archivo por falta de nombre de archivo	El usuario intenta realizar una carga sin archivo.	El usuario oprime el botón "Cargar" sin haber seleccionado previamente un archivo.	Debe generar un mensaje de error "Error: no se seleccionó archivopara cargar".
SP1-QA-5	Carga Fallida del archivo por errores de formato de archivo	El usuario carga un archivo inválido para ser reconocido por el sistema.	El usuario intenta cargar un archivo que tiene errores en el formato de creación de este.	Debe generar un mensaje de error "Error: formato de archivo inválido".
SP1-QA-6	Carga Fallida del archivo por errores en la validación de los datos cargados	El usuario carga un archivo con datos inválidos.	El usuario intenta cargar un archivo con errores en los tipos de datos.	Debe generar un mensaje de error "Error: datos leídos inválidos".

Sprint 2

Microservicio clientes

EL EQUIPO deberá desarrollar el microservicio de gestión de los clientes de la tienda, para lo cual se deberán tener las opciones de crear cliente, leer clientes, actualizar datos de cliente, y borrar cliente. Estos datos se almacenarán en la colección llamada **db_clientes**, y los datos a gestionar son: cedula, nombre completo, dirección, teléfono, y correo electrónico.

Microservicio interfaz web (Segundo Incremento de producto)

Se deberá realizar el segundo incremento de producto para el microservicio de interfaz web

agregando la interacción con el usuario del microservicio de productos.

Historias de Usuario

Identificador (ID) de la historia	Rol	Característica / Funcionalidad	Razón / Resultado
HU-003	Como administrador	Necesito crear nuevos clientes	Con la finalidad de registrar a los clientes que realicen compras en la tienda
HU-004	Como administrador	Necesito consultar los datos del cliente, por medio de la cédula	Con la finalidad de poder recuperar los datos de un cliente creado.
HU-005	Como administrador	Necesito actualizar los datos de los clientes que estén en el sistema	Con la finalidad de poder actualizar los datos de nombre completo, dirección, teléfono, y correo electrónico, previa consulta por cédula.
HU-006	Como administrador	Necesito borrar los datos de clientes que estén en el sistema	Con la finalidad de borrar los datos de los clientes que ya no deben tener ingreso al sistema, previa consulta por cédula.
HU-007	Como administrador	Necesito crear nuevos proveedores que entrarán al sistema	Con la finalidad de cargar los proveedores de la tienda.
HU-008	Como administrador	Necesito consultar los datos del proveedor, por medio de del NIT	Con la finalidad de poder recuperar los datos de un usuario creado.
HU-009	Como administrador	Necesito actualizar los datos de los proveedores que estén en el sistema	Con la finalidad de poder actualizar los datos de nombre de proveedor, Dirección, teléfono y ciudad, previa consulta por NIT.
HU-010	Como administrador	Necesito borrar los datos de un proveedor que estén en el sistema	Con la finalidad de borrar los datos de los proveedores que ya no deben tener ingreso al sistema, previa consulta por NIT.

Interfaz Gráfica

Para la realización de la interfaz gráfica en términos de visualización, si bien en las siguientes gráficas se determina el texto “Cali/Medellín/Bogotá”, al momento de realizar el microservicio, deberá configurarse para que solamente aparezca una sola ciudad.

www.cadenatiendaslagenerica.com

Cadena de Tiendas La Genérica – Sucursal Cali/Medellín/Bogotá

Productos

Clientes

Cédula

Nombre Completo

Dirección

Teléfono

Correo Electrónico

Consultar

Crear

Actualizar

Borrar

Conjunto de Pruebas

ID Prueba	Caso de Prueba	Descripción	Acción de Entrada	Resultado Esperado
SP2-QA-1	Creación de un nuevo cliente correcto	El usuario debe poder crear un nuevo cliente en el sistema, si todos los datos del nuevo cliente están completos.	Escribir los datos de cédula, nombre completo, dirección, teléfono, y correo electrónico del nuevo cliente, posteriormente, se deberá presionar el botón “Crear”.	Debe genera un mensaje de información “Cliente Creado”. Luego, debe insertar los datos en la base de datos y, limpiar todos los datos ya creados.
SP2-QA-2	Creación de un nuevo cliente con errores, por falta de completitud de algunos de los campos.	El usuario debe recibir un mensaje de error, si no se diligencian todos los datos del nuevo cliente.	Escribir los datos del nuevo cliente con algún faltante.	Debe genera un mensaje de error “Faltan datos del cliente”, posteriormente, debe limpiar todos los datos ya creados.
SP2-QA-3	Consulta de cliente existente.	El usuario escribe la cédula del cliente en el campo “Cédula”, y éste se encuentra en la base de datos.	Escribir la cédula del cliente a consultar.	El sistema retorna todos los demás datos de cédula, nombre completo, dirección, teléfono, y correo electrónico del cliente.

SP2-QA-4	Consulta de cliente inexistente o con errores.	El usuario debe recibir un mensaje de error.	Escribir la cédula del cliente a consultar.	Debe genera un mensaje de error "Cliente Inexistente", posteriormente, debe limpiar todos los datos ya creados.
SP2-QA-5	Actualización Correcta de datos de cliente	El usuario, actualizará los datos de nombre completo, dirección, teléfono, y correo electrónico del cliente.	Escribir la cédula del usuario a consultar para ser actualizado.	Debe genera un mensaje de información "Datos del Cliente Actualizados". Luego, debe actualizar los mismos los datos en la base de datos y, limpiar todos los datos ya creados en el formulario.
SP2-QA-6	Actualización de datos de cliente con errores por datos en blanco	El usuario, recibirá un mensaje de error de datos al poner en blanco y/u omitir datos de cédula, nombre completo, dirección, teléfono, y correo electrónico del cliente.	Escribir la cédula del cliente a consultar, y borrar u omitir algún tipo de dato de los recuperados del cliente.	Debe genera un mensaje de error "Datos faltantes", posteriormente, debe limpiar todos los datos ya creados.
SP2-QA-7	Borrado correcto de datos de cliente	El usuario, podrá borrar los datos de nombre completo, dirección, teléfono, y correo electrónico del cliente.	Escribir la cédula del cliente a consultar para ser borrado.	Debe consultar la base de datos por la existencia del cliente con la cédula indicada. Debe genera un mensaje de información "Datos del Cliente Borrados". Luego, debe borrar los mismos los datos en la base de datos y, limpiar todos los datos ya creados en el formulario.
SP2-QA-8	Borrado de datos de cliente con errores por datos en blanco	El usuario, recibirá un mensaje de error de datos al poner en blanco y/u omitir datos de cédula, nombre completo, dirección, teléfono, y correo electrónico del cliente.	Escribir la cédula del cliente a consultar, y alterar la cédula y/o borrarla.	Debe consultar la base de datos por la existencia del cliente con la cédula indicada, y debe generar un mensaje de error "Cédula Errada", posteriormente, debe limpiar todos los datos ya creados.

Sprint 3

MICROSERVICIO VENTAS.

EL EQUIPO deberá desarrollar el microservicio para la gestión de las ventas que se realicen en la tienda. El sistema, buscará los datos del cliente por cédula. Posteriormente, el sistema permitirá escribir el código del producto, y se visualizará el nombre de este en pantalla. Se digitará la cantidad a vender, y generará el valor total de venta por producto. El sistema permitirá que se realice la misma operación con otros productos hasta que se le dé la opción de totalizar. En ese momento, el sistema calculará el valor del total de IVA para tres (3) productos, según el valor de IVA definido para cada producto, y luego dará el valor total con IVA.

El sistema deberá registrar la venta con los siguientes datos: código de venta – este es un valor consecutivo, cedula del cliente, cedula del usuario, valor total de venta, valor de IVA, y valor total más IVA, de igual forma, se deberá registrar el detalle de los productos vendidos de esta venta (desde 1 hasta 3 productos) con los siguientes datos: código de producto, cantidad, valor unitario (valor de venta), y valor total, junto con el código de la venta. Todos los anteriores datos se guardarán en la colección llamada **db_ventas**. Finalmente, el sistema deberá mostrar un mensaje de confirmación de la transacción.

MICROSERVICIO INTERFAZ WEB (Tercer Incremento de producto)

Se deberá realizar el tercer incremento de producto para el microservicio de interfaz web agregando la interacción con el usuario del microservicio de ventas.

Historias de Usuario

Identificador (ID) de la historia	Rol	Característica / Funcionalidad	Razón / Resultado
HU-011	Como usuario	Necesito registrar las ventas de la tienda para un cliente.	Con la finalidad de ingresar los datos de una nueva venta de la tienda.
HU-012	Como usuario	Necesito obtener los datos del cliente mediante consulta por cédula.	Con la finalidad de cargarlos al formulario de ventas, almomento que se desee registrar una venta para este cliente.
HU-013	Como usuario	Necesito obtener los datos de los productos mediante consulta por código de producto.	Con la finalidad de cargarlos al formulario de ventas, almomento que se desee registrar una venta para este cliente.
HU-014	Como usuario	Necesito calcular el valor de venta por cada producto vendido, mediante la cantidad vendida, hasta un total de tres (3) productos.	Con la finalidad de acumular el total de venta para los valores totales de los productos.
HU-015	Como usuario	Necesito generar el valor de la venta total para el cliente, incluyendo IVA y total con IVA.	Con la finalidad de presentar al cliente los valores totales de venta e IVA al cliente.
HU-016	Como usuario	Necesito confirmar el valor de venta, dada la aceptación del cliente.	Con la finalidad registrar en la base de datos de ventas de la tienda.

Interfaz Gráfica

Para la realización de la interfaz gráfica en términos de visualización, si bien en las siguientes gráficas se determina el texto “Cali/Medellín/Bogotá”, al momento de realizar el microservicio, deberá configurarse para que solamente aparezca una sola ciudad.

Productos

Cientes

Ventas

www.cadenatiendaslagenerica.com

Cadena de Tiendas La Genérica – Sucursal Cali/Medellín/Bogotá

Cédula

Consultar

Cliente

Consec.

Cod. Producto	Nombre Producto	Cant.	Vlr. Total
<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>
<div>Total Venta</div>			<div></div>
<div>Total IVA</div>			<div></div>
<div>Total con IVA</div>			<div></div>

Confirmar

Conjunto de Pruebas

ID Prueba	Caso de Prueba	Descripción	Acción de Entrada	Resultado Esperado
SP3-QA-1	Consulta exitosa de la cédula del cliente	El usuario realiza una consulta de cédula del cliente que recupera el nombre de este de forma correcta.	El usuario digita la cédula del cliente y oprime el botón "Consultar".	Debe recuperar el nombre del cliente de la base de datos de forma correcta, desplegándolo en el espacio respectivo.
SP3-QA-2	Consulta fallida de la cédula del cliente	El usuario realiza una consulta de cédula del cliente que no se encuentra en la base de datos.	El usuario digita una cédula de cliente y oprime el botón "Consultar".	Debe generar un mensaje por pantalla que indique que la cédula no se encuentra registrada en la base de datos.
SP3-QA-3	Consulta exitosa de producto	El usuario realiza una consulta del código de producto y se recupera de forma correcta.	El usuario digita el código del producto para la venta y oprime el botón "Consultar".	Debe recuperar el nombre del producto de la base de datos de forma correcta, desplegándolo en el espacio respectivo.
SP3-QA-4	Consulta fallida de producto	El usuario realiza una consulta de código de producto que no se encuentra en la base de datos.	El usuario digita un código de producto y oprime el botón "Consultar".	Debe generar un mensaje por pantalla que indique que el código de producto no se encuentra registrada en la base de datos.

SP3-QA-5	Validación del campo cantidad de productos	El sistema realiza una validación exitosa de la cantidad de productos, donde el dato sea inferior a cero, o nulo	El usuario digita una cantidad de producto inferior a cero, o valor nulo, cuando el usuario presiona el botón "Confirmar"	Debe generar un mensaje por pantalla que indique que el valor de cantidad es incorrecto.
SP3-QA-6	Validación del campo valor total por producto	El sistema realiza una validación exitosa del valor total por cada producto de los tres permitidos.	El sistema calcula de forma correcta el valor total de cada producto (cantidad x precio de venta), cuando el usuario presiona el botón "Confirmar"	Debe mostrar el valor correcto en el campo correspondiente de valor total por producto.
SP3-QA-7	Validación del campo Total Venta	El sistema realiza una validación exitosa del total de la venta de todos los tres (3) productos	El sistema calcula de forma correcta el valor total todos los productos, cuando el usuario presiona el botón "Confirmar"	Debe mostrar el valor correcto en el campo correspondiente de valor total por producto.
SP3-QA-8	Validación del campo Total IVA	El sistema realiza una validación exitosa del total del IVA de todos los tres (3) productos	El sistema calcula de forma correcta el valor del IVA de los tres (3) productos, cuando el usuario presiona el botón "Confirmar"	Debe mostrar el valor correcto en el campo correspondiente de valor total del IVA.
SP3-QA-9	Validación del campo Total con IVA	El sistema realiza una validación exitosa del total de la venta, incluyendo el valor del IVA de todos los tres (3) productos	El sistema calcula de forma correcta el valor total de la venta, incluyendo el IVA de los tres (3) productos, cuando el usuario presiona el botón "Confirmar"	Debe mostrar el valor correcto en el campo correspondiente de valor total de la venta incluyendo el IVA.
SP3-QA-10	El consecutivo de la venta se genera exitosamente	Al momento de confirmar la venta, se genera un número consecutivo de esta.	El usuario presiona el botón "Confirmar"	Se debe imprimir un número consecutivo en el campo destinado para tal fin.

Sprint 4

MICROSERVICIO REPORTES

EL EQUIPO desarrollará el microservicio de reportes del sistema para cada una de las sedes de la cadena de tiendas, el cual deberá poder generar los siguientes como mínimo por cada sede: a) listado de clientes, y, c) total de ventas por cliente. Las consultas deberán ser por pantalla.

MICROSERVICIO INTERFAZ WEB (Cuarto Incremento de producto)

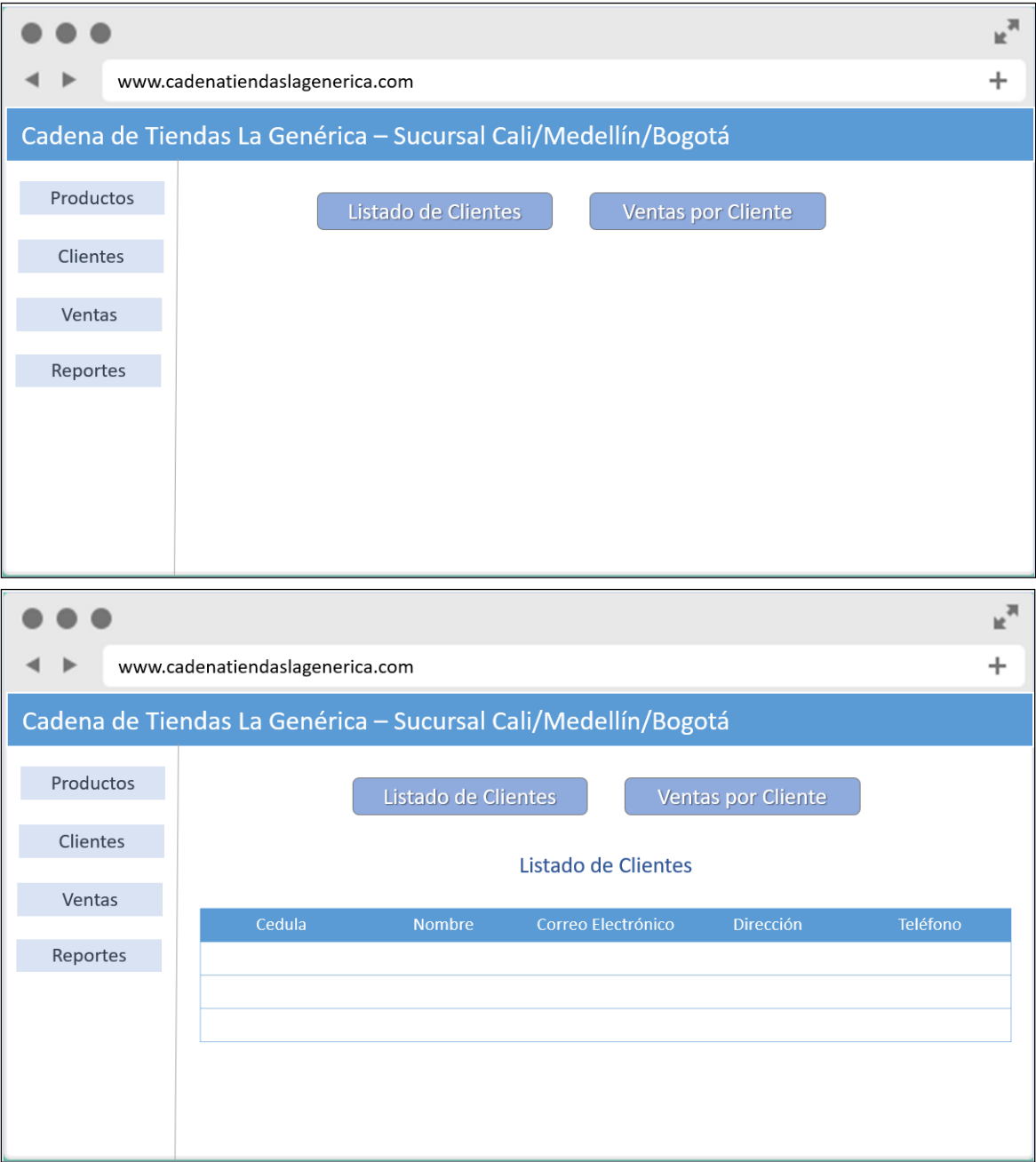
Se deberá realizar el cuarto incremento de producto para el microservicio de interfaz web agregando la interacción con el usuario del microservicio de reportes.

Historias de Usuario

Identificador (ID) de la historia	Rol	Característica / Funcionalidad	Razón / Resultado
HU-017	Como usuario	Necesito consultar el listado de clientes de la sede	Con la finalidad de verificar cuales clientes han comprado en la tienda.
HU-018	Como usuario	Necesito consultar el total de ventas por cliente de la sede	Con la finalidad de contabilizar las ventas por cliente.

Interfaz Gráfica

Para la realización de la interfaz gráfica en términos de visualización, si bien en las siguientes gráficas se determina el texto “Cali/Medellín/Bogotá”, al momento de realizar el microservicio, deberá configurarse para que solamente aparezca una sola ciudad.





Conjunto de Pruebas

ID Prueba	Caso de Prueba	Descripción	Acción de Entrada	Resultado Esperado
SP4-QA-1	Generación de listado de clientes exitosa	El usuario, al presionar el botón “listado de clientes”, puede ver el listado de clientes existente.	El usuario presiona el botón “Listado de Usuarios”	Debe llevar a una página donde se listen los datos de cedula, nombre, email, dirección y teléfono. Si no existen clientes, se deberá generar un mensaje.
SP4-QA-2	Generación de Total de Ventas por cliente exitosa	El usuario, al presionar el botón “Total ventas por Cliente”, puede ver el listado de clientes con el total de ventas y el total consolidado.	El usuario presiona el botón “Total ventas por Cliente”	Debe llevar a una página donde se listen los datos de cedula, y del cliente, y el valor total de ventas de cada cliente. Al final deberá totalizar el valor de todas las ventas desplegadas, que corresponden al total indicado por cliente. Si no existen clientes, se deberá generar un mensaje.

Sprint 5

MICROSERVICIO CONSOLIDACIÓN (solo Bogotá)

El EQUIPO desarrollará el microservicio de consolidación de las ventas de las ciudades satélite hacia Bogotá para cada una de las sedes de la cadena de tiendas, la cual realizará la consulta de las ventas que se hayan hecho hasta el momento en cada ciudad, y se realizará la sumatoria de dichas consultas, así como las de Bogotá. Se almacenará la información de los siguientes datos: fecha, ciudad, y total de Ventas en la colección db_consolidado.

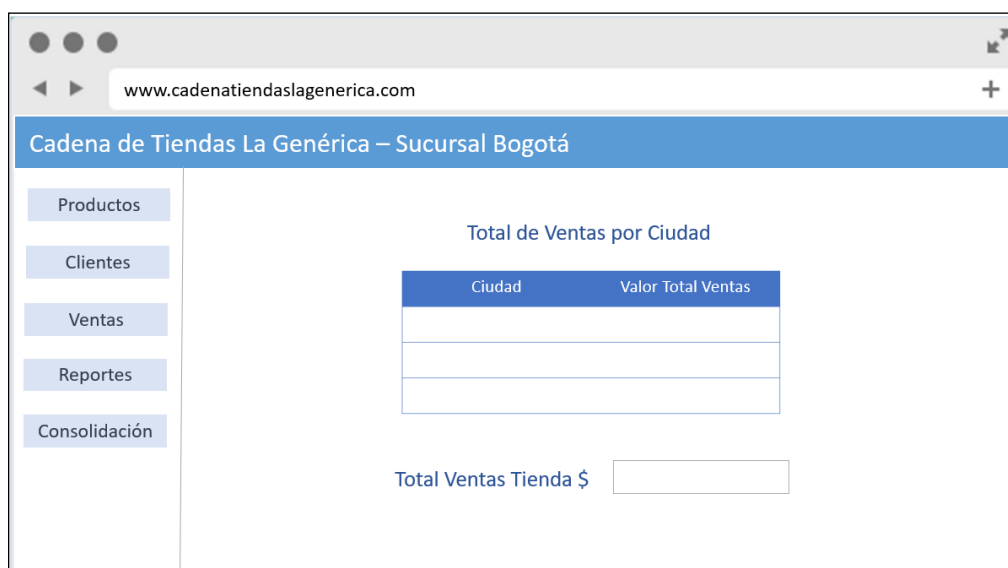
MICROSERVICIO INTERFAZ WEB (Quinto Incremento de producto)

Se deberá realizar el quinto y último incremento de producto para el microservicio de interfaz web que esté en la ciudad de Bogotá solamente, agregando la interacción con el usuario del microservicio de reportes.

Historias de Usuario

Identificador (ID) de la historia	Rol	Característica / Funcionalidad	Razón / Resultado
HU-019	Como administrador	Necesito consultar el consolidado del total de ventas por cliente de la cadena de tiendas.	Con la finalidad de verificar cuales clientes han comprado en la tienda.

Interfaz Gráfica



The screenshot shows a web browser window with the URL www.cadenatiendaslagenerica.com. The page title is "Cadena de Tiendas La Genérica – Sucursal Bogotá". On the left, there is a sidebar menu with the following items: "Productos", "Clientes", "Ventas", "Reportes", and "Consolidación". The main content area displays the title "Total de Ventas por Ciudad" above a table with two columns: "Ciudad" and "Valor Total Ventas". The table has three empty rows for data entry. Below the table, there is a label "Total Ventas Tienda \$" followed by an empty input field.

Conjunto de Pruebas

ID Prueba	Caso de Prueba	Descripción	Acción de Entrada	Resultado Esperado
SP5-QA-1	Generación de Totalde Ventas por ciudad exitosa	El usuario, al presionar el botón “Consolidación”, puede ver el listado de ciudades con el total deventas por ciudad y el gran total por toda la cadena de tiendas.	El usuario presiona el botón “Consolidación”	Debe llevar a una página nueva donde se listen los datos de ciudad, y el valor total de ventas por ciudad. Al final deberá totalizar el valor del total de ventas por toda la cadena de tiendas Si no existen ventas, se deberá generar un mensaje.

- Usar contenedores para desplegar y manejar el ciclo de vida de los componentes (Docker)

Parte 4: Especificaciones Técnicas

Lenguajes de programación y stack tecnológico de desarrollo

Los lenguajes de programación, frameworks, y herramientas tecnológicas que se emplearán parael desarrollo de este proyecto son los siguientes:

- Java EE para desarrollo Web. JDK: JavaSE 11.0.2
- Ambiente Integrado de Desarrollo (IDE) Eclipse for Enterprise Java and Web DevelopersVersión 2021-06
- Framework Spring Boot, versión 2.4.5 (integrado con el IDE)
- Gestor de dependencias Maven, versión 3.6+
- Servidor de Aplicaciones Web para Java Apache Tomcat® - Apache Tomcat 9
- Gestor de Base de datos Mongo DB Versión 5.0.2
- Cliente gestor de bases de datos MongoDB Compass 1.28.1.
- Gestor de Contenedores Docker – Desktop (Windows) v.20.10.8
- Gestor de pruebas de API – Postman. V. xx
- Repositorio Github, e integración con Eclipse.

Especificación de diseño de la base de datos no relacional.

Se presenta a continuación el diseño de la base de datos de la cadena de tiendas la genérica (CadenaLaGenerica):

Colección	Modelo	Valor de Ejemplo
db_clientes	Clientes { cedula_cliente (integer, optional), direccion_cliente (string, optional), email_cliente (string, optional), nombre_cliente (string, optional), telefono_cliente (string, optional) }	<pre>{ "cedula_cliente": 0, "direccion_cliente": "string", "email_cliente": "string", "nombre_cliente": "string", "telefono_cliente": "string" }</pre>
db_consolidad o	Consolidado { ciudad (string, optional), id (integer, optional), total_ventas (number, optional) }	<pre>{ "ciudad": "string", "id": 0, "total_ventas": 0 }</pre>
db_productos	Productos { codigo_producto (integer, optional), ivacompra (number, optional), nitproveedor (integer, optional), nombre_producto (string, optional), precio_compra (number, optional), precio_venta (number, optional) }	<pre>{ "codigo_producto": 0, "ivacompra": 0, "nitproveedor": 0, "nombre_producto": "string", "precio_compra": 0, "precio_venta": 0 }</pre>
db_ventas	Ventas { cedula_cliente (integer, optional), codigo_venta (integer, optional), detalle_venta (Array[DetalleVentas], optional), ivaventa (number, optional), total_venta (number, optional), valor_venta (number, optional) } DetalleVentas { cantidad_producto (integer, optional), codigo_producto (integer, optional), valor_total (number, optional), valor_venta (number, optional), valoriva (number, optional) }	<pre>{ "cedula_cliente": 0, "codigo_venta": 0, "detalle_venta": [{ "cantidad_producto": 0, "codigo_producto": 0, "valor_total": 0, "valor_venta": 0, "valoriva": 0 }], "ivaventa": 0, "total_venta": 0, "valor_venta": 0 }</pre>
	DetalleVentas { cantidad_producto (integer, optional), codigo_producto (integer, optional), valor_total (number, optional), valor_venta (number, optional), valoriva (number, optional) }	

Especificación de la API para conexión del Frontend con el Backend de la aplicación

1. APIs para gestión de Clientes.

Actualizar Clientes

PUT /clientes/actualizar actualizar

Parameters

Parameter	Value	Description	Parameter Type	Data Type
cliente	(required)	cliente	body	Model

Parameter content type: application/json

```
{
  "cedula_cliente": 0,
  "direccion_cliente": "string",
  "email_cliente": "string",
  "nombre_cliente": "string",
  "telefono_cliente": "string"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

Eliminar Clientes

DELETE /clientes/eliminar/{id} eliminar

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	id	path	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
204	No Content		
401	Unauthorized		
403	Forbidden		

Try it out!

Agregar Clientes

POST
/cientes/guardar
guardar

Parameters

Parameter	Value	Description	Parameter Type	Data Type
cliente	<div> (required) </div> <div> Parameter content type: application/json </div>	cliente	body	<div> Model </div> <div> Example Value </div> <div> <pre>{ "cedula_cliente": 0, "direccion_cliente": "string", "email_cliente": "string", "nombre_cliente": "string", "telefono_cliente": "string" }</pre> </div>

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

Listar Clientes

GET
/cientes/listar
listar

Response Class (Status 200)
OK

Model

Example Value

```
[
  {
    "cedula_cliente": 0,
    "direccion_cliente": "string",
    "email_cliente": "string",
    "nombre_cliente": "string",
    "telefono_cliente": "string"
  }
]
```

Response Content Type

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

2. APIs para Consolidado

Actualizar Consolidado

PUT /consolidado/actualizar actualizar

Parameters

Parameter	Value	Description	Parameter Type	Data Type
consolidado	<pre>{ "ciudad": "string", "id": 0, "total_ventas": 0 }</pre>	consolidado	body	Model Example Value

Parameter content type: application/json

```
{
  "ciudad": "string",
  "id": 0,
  "total_ventas": 0
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

Eliminar Consolidado

DELETE /detalleventas/eliminar/{id} eliminar

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	id	path	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
204	No Content		
401	Unauthorized		
403	Forbidden		

Try it out!

3. APIs para Productos

Actualizar Productos

PUT /productos/actualizar actualizar

Parameters

Parameter	Value	Description	Parameter Type	Data Type
producto	(required)	producto	body	Model

Parameter content type: application/json

```
{
  "codigo_producto": 0,
  "iva_compra": 0,
  "nit_proveedor": 0,
  "nombre_producto": "string",
  "precio_compra": 0,
  "precio_venta": 0
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

Eliminar Productos

DELETE /productos/eliminar/{id} eliminar

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	id	path	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
204	No Content		
401	Unauthorized		
403	Forbidden		

Try it out!

Guardar Productos

POST

/productos/guardar

guardar

Parameters

Parameter	Value	Description	Parameter Type	Data Type
producto	(required)	producto	body	Model
				Example Value
				<pre>{ "codigo_producto": 0, "ivacompra": 0, "nitproveedor": 0, "nombre_producto": "string", "precio_compra": 0, "precio_venta": 0 }</pre>

Parameter content type: application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

Listar Productos

GET

/productos/listar

listar

Response Class (Status 200)

OK

Model

Example Value

```
[
  {
    "codigo_producto": 0,
    "ivacompra": 0,
    "nitproveedor": 0,
    "nombre_producto": "string",
    "precio_compra": 0,
    "precio_venta": 0
  }
]
```

Response Content Type */*

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

4. APIs de Ventas

Actualizar Ventas

PUT /ventas/actualizar actualizar

Parameters

Parameter	Value	Description	Parameter Type	Data Type
venta	<pre>{ "cedula_cliente": 0, "codigo_venta": 0, "detalle_venta": [{ "cantidad_producto": 0, "codigo_producto": 0, "valor_total": 0, "valor_venta": 0, "valoriva": 0 }], "ivaventa": 0, "total_venta": 0, "valor_venta": 0 }</pre>	venta	body	<div>Model</div> <div>Example Value</div> <pre>{ "cedula_cliente": 0, "codigo_venta": 0, "detalle_venta": [{ "cantidad_producto": 0, "codigo_producto": 0, "valor_total": 0, "valor_venta": 0, "valoriva": 0 }], "ivaventa": 0, "total_venta": 0, "valor_venta": 0 }</pre>

Parameter content type: application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

Eliminar Ventas

DELETE /proveedores/eliminar/{id} eliminar

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	id	path	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
204	No Content		
401	Unauthorized		
403	Forbidden		

Try it out!

Agregar Ventas

POST /ventas/guardar guardar

Parameters

Parameter	Value	Description	Parameter Type	Data Type
venta	<pre>{ "cedula_cliente": 0, "codigo_venta": 0, "detalle_venta": [{ "cantidad_producto": 0, "codigo_producto": 0, "valor_total": 0, "valor_venta": 0, "valoriva": 0 }], "ivaventa": 0, "total_venta": 0, "valor_venta": 0 }</pre>	venta	body	Model Example Value <pre>Ventas { cedula_cliente (integer, optional), codigo_venta (integer, optional), detalle_venta (Array(DetalleVentas), optional), ivaventa (number, optional), total_venta (number, optional), valor_venta (number, optional) } DetalleVentas { cantidad_producto (integer, optional), codigo_producto (integer, optional), valor_total (number, optional), valor_venta (number, optional), valoriva (number, optional) }</pre>

Parameter content type: application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

Listar Ventas

GET /ventas/listar listar

Response Class (Status 200)

OK

Model Example Value

```
[
  {
    "cedula_cliente": 0,
    "codigo_venta": 0,
    "detalle_venta": [
      {
        "cantidad_producto": 0,
        "codigo_producto": 0,
        "valor_total": 0,
        "valor_venta": 0,
        "valoriva": 0
      }
    ],
    "ivaventa": 0,
    "total_venta": 0,
    "valor_venta": 0
  }
]
```

Response Content Type */*

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!



Diagrama de Despliegue de Contenedores Docker.

