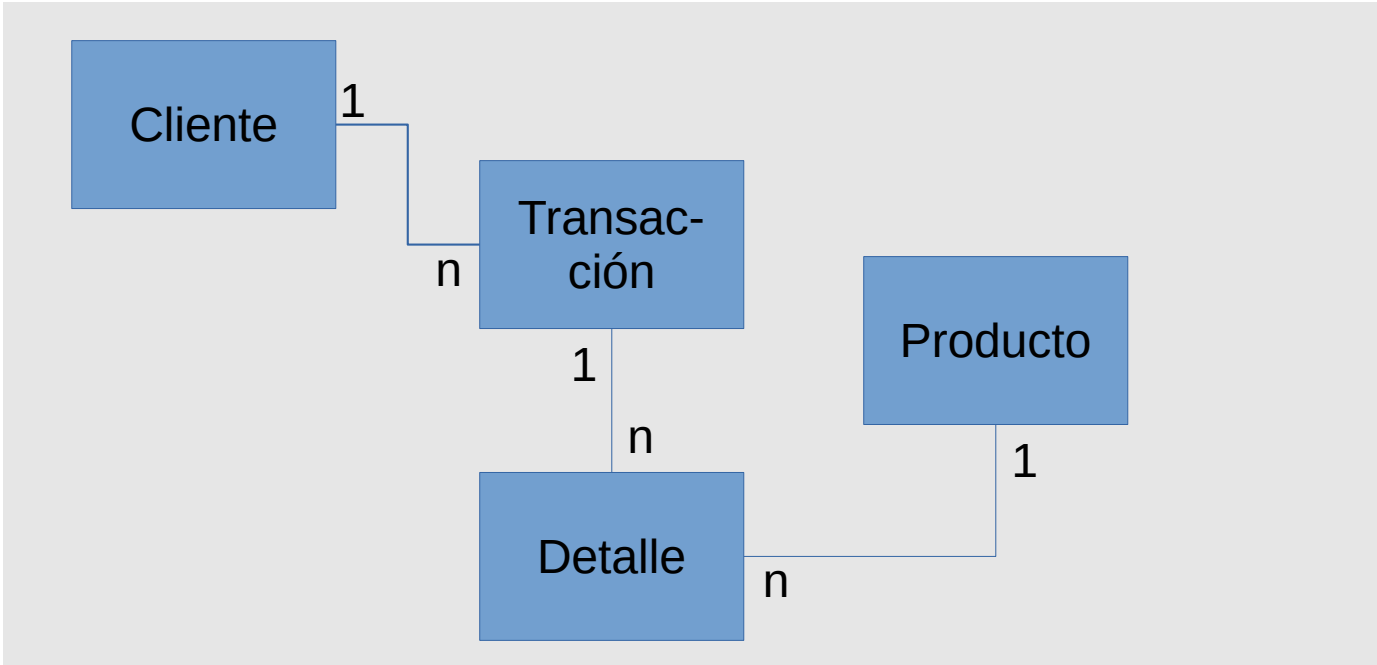


Instrucciones

Se debe armar una base de datos con el siguiente modelo:



Entidad **Cliente**

Atributo	Descripción	Restricciones
id	identificador interno del cliente	(clave primaria) número entero generado secuencialmente
nit	número de identificación tributaria del cliente	(clave candidata) texto de longitud máxima 10 caracteres, debe ser único para el cliente
nombre	nombre del cliente	texto, longitud máxima 256 caracteres
fechacreacion	fecha de creación del registro del cliente	fecha y hora (timestamp)

Entidad **Producto**

Atributo	Descripción	Restricciones
id	identificador interno del producto	(clave primaria) número entero generado secuencialmente
codigo	código identificador del producto	(clave candidata) texto de longitud máxima 15 caracteres, debe ser único para el producto
descripcion	descripción del producto	texto, longitud máxima 80 caracteres.
fechacreacion	fecha de creación del registro del producto	fecha y hora (timestamp)
precio	precio unitario del producto	monto de dos decimales y hasta 8 dígitos; corresponde al precio recomendado de una unidad del producto
saldo	saldo en existencia del producto	entero mayor o igual a cero, de hasta 5 dígitos

Entidad **Transaccion**

Atributo	Descripción	Restricciones
id	identificador interno de la transaccion	(clave primaria) número entero generado secuencialmente
fecha	fecha de la transacción	fecha y hora (timestamp)
idcliente	identificador del cliente	restricción referencial a la entidad Cliente
tipo	tipo de la transacción	texto de un carácter de longitud: V – Venta D – Devolución
idtransoriginal	identificador de la transacción original	en el caso de una transacción de venta, debe ser nulo, en el caso de una transacción de devolución, debe apuntar a la transacción de venta original

Entidad **Detalle**

Atributo	Descripción	Restricciones
id	identificador interno del detalle	(clave primaria) número entero generado secuencialmente
idtransaccion	identificador de la transacción a la que corresponde el detalle	restricción referencial a la entidad Transaccion
idproducto	identificador del producto	restricción referencial a la entidad Producto .
precio	precio unitario al que se vendió el producto	monto de dos decimales y hasta 8 dígitos; en el caso de una transacción de venta puede ser distinto al precio descrito en la entidad Producto ; en el caso de una transacción de devolución, debe ser el mismo que la transacción de venta original
cantidad	cantidad de ítems del producto	entero de hasta 5 dígitos, puede ser positivo en caso de una transacción de venta o negativo en caso de una transacción de devolución

Sobre esta base de datos se debe desarrollar un API REST que permita hacer las siguientes operaciones:

- obtener datos de un cliente (o clientes) utilizando como criterio de búsqueda un código parcial o un nombre parcial.
- crear datos de un nuevo cliente.
- modificar datos (excepto el código identificador) de un cliente.
- obtener datos de un producto (o productos) utilizando como criterio de búsqueda un código parcial o un nombre parcial.
- crear datos de un nuevo producto.
- modificar datos (excepto el código identificador) de un producto.
- registrar una transacción de venta asociada a un cliente con el detalle y cantidad de productos asociados. Las cantidades deben ser siempre positivas y no se debe repetir el mismo producto dentro de una transacción. En caso que la cantidad de un producto en la transacción sea superior al saldo del producto, la transacción total debe ser rechazada. Una vez registrada la transacción, la cantidad de cada producto se debe restar del saldo del producto, reduciéndose dicho saldo.

- en base a una transacción de venta, registrar una transacción de devolución. Las cantidades de cada producto deben ser expresadas en negativo y su valor absoluto no debe ser superior al de la venta original. Si un producto no se devuelve, la cantidad puede ser cero o puede eliminarse el registro de detalle. Si se registra un producto inexistente en la transacción original la transacción debe ser rechazada. Si la transacción original ya tenía una devolución asociada, la transacción de devolución debe ser rechazada. Una vez registrada la transacción, la cantidad de cada producto se debe restar del saldo del producto, incrementándose dicho saldo ya que la cantidad es negativa.
- obtener datos de una transacción utilizando como criterio de búsqueda su identificador.
- obtener datos de una transacción utilizando como criterio de búsqueda el identificador de la transacción original (idtransoriginal).

Se debe desarrollar una interfaz web que consuma el API REST descrito anteriormente, permitiendo a un usuario hacer las operaciones descritas.

Para propósitos del ejercicio, no es necesario tener autenticación ni autorización de usuarios. El acceso al API y a la interfaz web puede ser irrestricto. Asimismo no importa validar el orden de registro de las transacciones, se asume que los usuarios las harán en el orden correcto.

En caso de violarse alguna de las reglas de negocio, debe reportarse un error y no registrarse la operación.

Puntaje

Ítem	Puntos
Base de Datos	2
Esquema básico de la base de datos relacional creado	1
Existencia de índices y restricciones referenciales	1
Backend	13
obtener datos de clientes	1
crear datos de un nuevo cliente	1
modificar datos de un cliente	1
obtener datos de un producto	1
crear datos de un nuevo producto	1
modificar datos de un producto	1
registrar una transacción de venta	3
registrar una transacción de devolución	3
obtener datos de una transacción (por id y por idtransoriginal)	1
Frontend	13
consulta de clientes	1
crear datos de un nuevo cliente	1
modificar datos de un cliente	1
consulta de productos	1
crear datos de un nuevo producto	1
modificar datos de un producto	1
registrar una transacción de venta	3
registrar una transacción de devolución	3
consulta de transacciones	1
Git	
Código fuente en el repositorio	1
Uso de automatización en el compilado	1
Total	30

Restricciones

Tanto el backend como el frontend deben instalarse en el servidor proporcionado para el ejercicio y la prueba debe realizarse en dicho servidor. El postulante tiene la libertad de utilizar las herramientas que considere necesarias en su ambiente de desarrollo personal.

Para la persistencia debe utilizarse sólo uno de los dos motores de base de datos instalados en el servidor proporcionado para el ejercicio. El postulante puede utilizar otro motor de base de datos si lo considera necesario, pero el mismo debe ser instalado en el servidor proporcionado y se le descontará el puntaje correspondiente a Base de Datos.

Para el Backend, se puede utilizar Java/Wildfly (preferentemente), Node.js o PHP. En caso de utilizar otro lenguaje, el postulante debe efectuar la instalación en el servidor proporcionado y se le descontarán 5 puntos del puntaje correspondiente al Backend. Si el postulante lo considera necesario, puede utilizar otro application server (Tomcat, TomEE, Glassfish, Payara, etc.) pero debe considerar el tiempo de instalación y configuración.

Para el Frontend, se puede utilizar Angular (preferentemente), Vue.js, React o HTML5+Javascript puro. En caso de utilizar otro framework o lenguaje, el postulante debe efectuar la instalación en el servidor proporcionado y se le descontarán 2 puntos del puntaje correspondiente al Frontend.

Sistema Operativo

El servidor tiene el sistema operativo Linux, distribución CentOS Stream 8.

- SELinux está deshabilitado.
- El acceso a internet es irrestricto.
- Para la instalación de paquetes, está habilitado el repositorio EPEL.

Paquetes relevantes instalados sobre la instalación base:

- httpd 2.4
- mod_ssl 2.4
- nginx 1.14
- java-1.8.0-openjdk
- maven 3.5
- php 7.4
- php-fpm 7.4
- nodejs 16
- npm 8.3
- postgresql 12
- postgresql-server 12
- postgresql-jdbc 42.2
- php-pgsql 7.4
- mariadb 10.3
- mariadb-server 10.3
- mariadb-java-client 2.7
- php-mysqldb 7.4
- git 2.31

Credenciales SSH

servidor: **prueba30.lavitalicia.bo**

usuario: **root**

contraseña: **laVitalicia2023a**

Apache Httpd

Está configurado para publicar los archivos colocados en la carpeta **/var/www/html**. El acceso a PHP está configurado con FastCGI/PHP-FPM. Tiene SSL/TLS habilitado y escucha en los puertos 80 y 443 en todas las interfaces.

El servicio systemd se llama **httpd** y no se levanta automáticamente. Para levantar se debe usar:

```
# systemctl start httpd
```

Nginx

Está configurado para publicar los archivos colocados en la carpeta **/usr/share/nginx/html**. El acceso a PHP está configurado con FastCGI/PHP-FPM. Tiene SSL/TLS habilitado y escucha en los puertos 80 y 443 en todas las interfaces.

El servicio systemd se llama **nginx** y no se levanta automáticamente. Para levantar se debe usar:

```
# systemctl start nginx
```

PostgreSQL

El servicio de base de datos se levanta automáticamente y escucha en el puerto 5432 de la interfaz local (127.0.0.1) solamente. El servicio systemd se llama **postgresql**. Ya tiene creada una base de datos vacía, de nombre **prueba**.

Credenciales de administrador

usuario: **postgres**

contraseña: **postgres**

MariaDB

El servicio de base de datos, esencialmente igual a MySQL, se levanta automáticamente y escucha en el puerto 3306 de la interfaz local (127.0.0.1) solamente. El servicio systemd se llama **mariadb**. Ya tiene creada una base de datos vacía, de nombre **prueba**.

Credenciales de administrador

usuario: **root**

contraseña: **root**

Wildfly

El servidor está instalado en la carpeta **/opt/wildfly**. El servicio systemd se llama **wildfy** y no se levanta automáticamente. Para levantar se debe usar:

```
# systemctl start wildfly
```

Una vez levantado, el servidor tiene habilitado SSL/TLS y escucha en todas las interfaces en los puertos 8080 y 8443. La GUI de administración se escucha solamente en la interfaz local, puerto 9990. Para deployar paquetes por ejemplo un WAR, se debe copiar a la carpeta **/opt/wildfly/standalone/deployments**. También es posible utilizar **jboss-cli.sh**, el cual se encuentra en el path y se puede llamar directamente:

```
# jboss-cli.sh -c
```

El servidor de wildfly tiene configurados módulos con los JDBC de PostgreSQL y MariaDB. Asimismo, ya tiene creadas las datasources:

- **PostgresDS**, conecta a la base de datos **prueba** en PostgreSQL.
- **MariaDB**, conecta a la base datos **prueba** en MariaDB.