

MANUAL TÉCNICO:

Para arrancar el proyecto se necesitan los siguientes paquetes:

NPM v 10.8.3

NODE v 22.9.0

Navegador Brave, o FireFox

Especificando las siguientes clases importantes que sirven para el funcionamiento correcto de proyecto:

MODELOS:

Cajas: Esta clase representa la información sobre una caja de una sucursal. Contiene el código de la caja, el nombre y código del usuario asociado, el nombre y código de la sucursal donde se encuentra la caja. Los getters y setters permiten obtener y modificar estos atributos.

Cliente: La clase Cliente modela la información básica de un cliente, incluyendo su identificación, nombre, teléfono y NIT. También incluye un atributo **total_gastado** que sirve para reportes. Si el NIT está vacío, se asigna el valor "CF". Tiene métodos para obtener y modificar cada uno de estos atributos.

Compra: Modela la compra realizada en una sucursal. Contiene el código de la compra, la sucursal, la fecha de la compra, el total de la compra y el usuario que la realizó. Los valores iniciales se asignan como vacíos o NaN (no numérico).

Login: Representa el proceso de inicio de sesión de un usuario, con un identificador (ID), contraseña y el rol del usuario. Los métodos permiten obtener y modificar el ID de usuario, la contraseña y el rol asignado.

Producto: Modela un producto en el inventario. Tiene atributos como código, nombre, precio unitario de venta, sucursal, cantidad en bodega, pasillo en bodega, pasillo en estantería y cantidad en estantería. Se pueden obtener y modificar todos estos valores a través de getters y setters.

ProdutosCompra: Representa los productos involucrados en una compra, incluyendo el ID de la compra, ID del producto, cantidad, precio unitario y el subtotal. El constructor inicializa estos valores y también permite añadir nuevos productos a la compra.

ProductosSucursales: Esta clase modela la relación entre productos y sucursales, especificando cuántos productos hay en estantería y bodega, y sus respectivos pasillos en cada sucursal. Sirve para rastrear la distribución de productos en diferentes lugares de almacenamiento dentro de la sucursal.

ProdcutosVenta: Similar a **ProdcutosCompra**, esta clase se utiliza para modelar los productos en una venta, con el ID de la venta, ID del producto, cantidad vendida, precio unitario y el subtotal.

SolicitudesModificacionCliente: Modela una solicitud de modificación de datos de un cliente, incluyendo atributos como el ID de la solicitud, usuario que la realiza, sucursal, cliente, descripción del cambio, si ha sido aprobado y si está en proceso de edición. Los métodos permiten obtener y modificar estos atributos.

SolicitudesTarjetas: Representa una solicitud de tarjeta de descuento, con atributos como código de la solicitud, estado, cliente asociado, motivo y descripción. Los métodos permiten obtener y modificar esta información.

Sucursal: Modela una sucursal de una tienda. Contiene la identificación de la sucursal, el nombre, la dirección y la cantidad de dinero en caja. Los métodos permiten obtener y modificar estos valores.

TarjetasDescuento: Representa una tarjeta de descuento asignada a un cliente, con atributos como código, tipo de tarjeta, cliente asociado, nombre del cliente, total de puntos acumulados, puntos por cada gasto y la última fecha de modificación. Los métodos permiten obtener y modificar estos valores.

Usuario: Representa un usuario en el sistema, con atributos como identificación, nombre, contraseña, rol, sucursal asignada y si está activo o no. El constructor inicializa estos valores y puede ser utilizado para crear y gestionar la información de los usuarios.

Venta: Modela una venta en la tienda, con atributos como el código de la venta, usuario, sucursal, cliente, NIT del cliente, total sin descuento, total con descuento, fecha, caja y puntos de descuento. Los métodos permiten obtener y modificar cada uno de estos atributos.

SERVICIOS:

1. **CajasService:**

- Maneja operaciones relacionadas con las cajas, como obtener cajas asignadas a un usuario, usuarios no asignados a cajas, o asignar una nueva caja.
- Utiliza parámetros **rol** y **idU** o **idS** para determinar el rol del usuario y la sucursal/usuario con la que interactúa.

2. **ClientesService:**

- Gestiona clientes y solicitudes de modificación de sus datos.
- Permite obtener, añadir y modificar clientes, así como consultar y confirmar solicitudes de modificación de datos de clientes.

3. **ComprasService:**

- Controla las compras, permitiendo obtener compras por sucursal, crear nuevas compras y cancelarlas.

4. **LoginService:**

- Maneja el proceso de inicio de sesión, enviando credenciales al backend y gestionando la sesión del usuario.

5. **ProductosService:**

- Se encarga de las operaciones relacionadas con los productos, como obtener productos de una sucursal, crear productos para compras y ventas, y mover productos entre diferentes ubicaciones (bodega y estantería).

6. **ReportesService:**

- Facilita la obtención de diferentes reportes, como los descuentos realizados, las ventas más grandes, los productos más vendidos, etc.

7. **SolicitudesTarjetasService:**

- Permite manejar las solicitudes de tarjetas de descuento, incluyendo la creación y modificación de dichas solicitudes.

8. **SucursalesService:**

- Gestiona las sucursales, permitiendo obtener una lista de sucursales o información de una sucursal específica.

9. **TarjetasService:**

- Permite la creación de nuevas tarjetas de descuento para clientes y obtener información de las tarjetas existentes.

10. **UsuariosService:**

- Gestiona los usuarios del sistema, permitiendo obtener la lista de usuarios, añadir nuevos usuarios, actualizarlos o eliminarlos.

11. **VentasService:**

- Maneja las operaciones de ventas, permitiendo realizar nuevas ventas, cancelarlas, y registrar los pagos con y sin descuentos.

Consideraciones Generales:

- Cada clase está decorada con `@Injectable` para ser inyectada en componentes o servicios que la necesiten.
- Todas las interacciones con el backend se realizan mediante el servicio `HttpClient` de Angular.