



Solicitante:

I.T.S. – Instituto Tecnológico Superior Arias - Balparda .

Nombre de Fantasía del Proyecto: BINDEV .

Grupo de Clase: 3° IF Turno: Nocturno .

Materia: Programación Web.

**Nombre de los Integrantes del Grupo: Alvarez Nicolas, Armand-ugón
Ignacio, Estefan German, Rivera Fabricio.**

Fecha de entrega: 12/ 09 /2022

Instituto Tecnológico Superior Arias Balparda
Gral. Flores 3591 esq. Bvar. José Batlle y Ordoñez - Montevideo

Objetivo

Definir y especificar el funcionamiento del sistema y exponer los usuarios de prueba para la utilización del mismo.

Alcance

Este documento está realizado con el fin de dejar una documentación estructurada para futuras personas que quieran hacer uso de la misma.

Índice

1. Modelo cliente - servidor	4
2. Método de autenticación	4
3. Usuarios	5
3.1 Clientes	5
3.1.1 Tipos de clientes:	5
3.2 Empleados	6

1. Modelo cliente - servidor

Para la realización de este sistema, se aplicó el patrón de diseño MVC (Modelo Vista Controlador) el cual consta de una vista que sería el cliente(Front-end) el cual solicita información hacia el servidor a través de una API Rest. La api se encarga de tomar la solicitud del cliente y procesarla dentro del servidor, primero que nada el controlador se encarga de recibir la solicitud del cliente, corroborar la información que sea válida y realizar los procesos necesarios. El modelo interactúa con la base de datos y el controlador para realizar las consultas que sean necesarias a la BD, para al final devolverle el resultado al cliente. La comunicación cliente servidor se realiza en formato JSON bidireccionalmente.

2. Método de autenticación

Para mayor seguridad de la información del usuario y el servidor se implementa la autenticación mediante token el cual el flujo funciona de la siguiente manera:
El servidor recibe las credenciales del cliente, corrobora su validez, y se genera un Token con una clave secreta alojada en el servidor, dentro de este Token se coloca un payload(contenido) el cual es el ID de usuario y se le fija una validez de 24hs. Una vez creado, se le manda al cliente en la respuesta y se fija en el navegador con una cookie(En nuestro caso en el local storage). De esta manera para cada solicitud que vaya a hacer un cliente se valida que ese token que no se haya expirado ni modificado en su forma natural con una función proveída de la librería y se decodifica para extraer su identidad(Id de usuario almacenada dentro). El token se envía en los headers de la request con la clave access-token y el valor del token.

3. Usuarios

3.1 Clientes

Para los usuarios de tipo cliente está disponible el alta y la modificación, aun la baja no se a implementado pero la idea es simplemente cambiar uno de los atributos dentro del modelo de usuario en la BD la cual es “state” por defecto cada vez que se genera un registro de usuario se inserta este valor en 1, para la baja es simplemente cambiar ese atributo en 0, luego el sistema contempla que cada vez que se vaya a logear un usuario ese valor este en 1. De lo contrario se le negara el acceso.

3.1.1 Tipos de clientes:

Empresa:

Para los clientes de tipo empresa se les solicitará dos datos adicionales: nrut y razón social.

Independiente

Se les solicitará los mismos datos del cliente de tipo empresa exceptuando el número de rut y la razón social.

Usuario: test@test.com

Contraseña: 123456

NOTA: La implementación para los usuarios de tipo empresa aun no esta realizada en el CLIENTE, pero si en la API. El cuerpo de la request del cliente para un **cliente** normal es la siguiente:

```
{  
  "email": "juan@gmail.com",  
  "name": "Juan",  
  "surname": "Perez",  
  "password": "123456",  
  "type": "NORMAL"  
}
```

Empresa:

```
{  
  "email": "juancito@gmail.com",  
  "name": "Juan",  
  "surname": "Perez",  
  "password": "123456",  
  "type": "COMPANY",  
}
```

```
"nRut": "12312312",  
"company": "Empresa SA"  
}
```

Este cuerpo es enviado a su respectivo endpoint a través de un POST.

Para registrar un usuario simplemente llenamos un formulario con las credenciales necesarias y listo. El servidor se encarga de validar la autenticidad de los datos y que no exista previamente. Lo mismo para iniciar sesión.

Para la modificación, una vez logueado. en la parte derecha superior vamos al panel de usuario y completamos los datos que vayamos a modificar y hacemos un submit.

3.2 Empleados

La interfaz para los empleados aún no está implementada a nivel cliente. Pero se puede probar el alta y el login a través de POSTMAN o algún cliente para hacer peticiones HTTP.

Para los empleados se manejan 3 roles distintos, JEFE(Podrá dar de alta empleados nuevos), COMPRADOR, VENDEDOR.

La aplicación contempla la autoridad que tiene cada rol sobre el sistema, un ejemplo: El usuario JEFE es el único que podrá dar de alta nuevos empleados en el sistema, en cambio los otros roles NO.

JEFE:

USUARIO: 50219376

CONTRASEÑA: 123456

VENDEDOR:

USUARIO: 50219375

CONTRASEÑA: 123456

COMPRADOR:

USUARIO: 50219374

CONTRASEÑA: 123456

4. Documentación de la API

Link:

<https://documenter.getpostman.com/view/17337246/VUqmwKHZ>

7. Vocabulario y Simbología

8. Bibliografía

NULL

HOJA TESTIGO

MATERIA: Programacion Web

Nombre del Profesor: Leonardo Carámbula

Nota Final

--