UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

FACULTAD DE PRODUCCION Y SERVICIOS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

Curso: Programación Web 2

PROYECTO FINAL

Docente: Rosa Yuliana Gabriela Paccotacya Yanque

Grupo 1

Integrantes:

- Chávez Chambi, Marco David
- Choque Quispe Eduardo Jhosue
- Choquehuanca Bedoya Brayan Denilson

Arequipa - Perú

2024

SISTEMA DE ADMINISTRACIÓN DE SUPERMERCADO

1. GitHub Backend y Frontend

https://github.com/marcoch018/PWEB2_PROJECT_TEORIA/tree/main

2. Tecnologías Usadas

Backend: Django

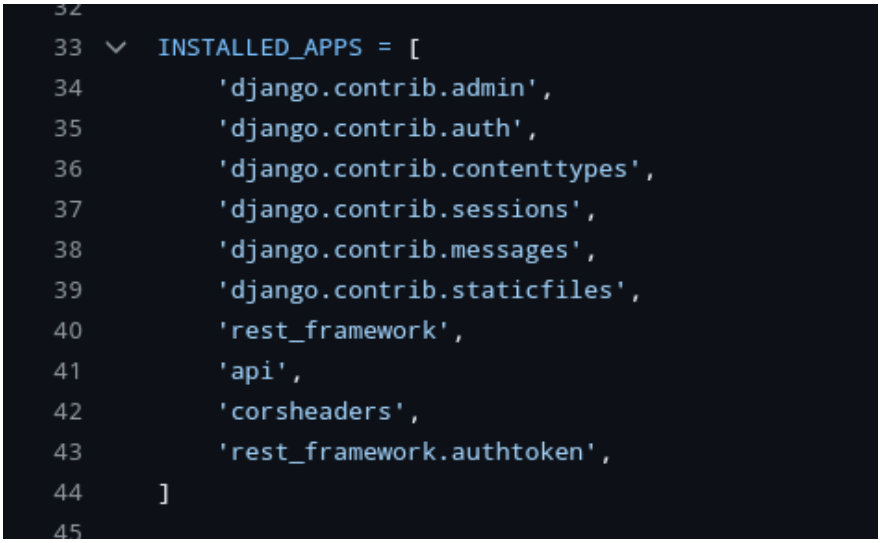
Frontend: React

Css

2. Desarrollo del Proyecto

a. Backend

1. ***Se creo la app api y se configuró settings.py***



```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'api',
    'corsheaders',
    'rest_framework.authtoken',
]
```

```python
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'corsheaders.middleware.CorsMiddleware',
]
```

```python
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'supermercado_db',
        'USER': 'root',
        'PASSWORD': '1234',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```

2. *Se crearon los modelos a utilizar en la base de datos*

```python
from django.db import models

class Usuario(models.Model):
    ROLES = [
        ('administrador', 'Administrador'),
        ('empleado', 'Empleado'),
    ]
    nombre = models.CharField(max_length=100)
    email = models.EmailField(unique=True)
    contraseña = models.CharField(max_length=255)
    rol = models.CharField(max_length=15, choices=ROLES)

    def __str__(self):
        return self.nombre

class Producto(models.Model):
    nombre = models.CharField(max_length=100)
    descripcion = models.TextField(blank=True, null=True)
    precio = models.DecimalField(max_digits=10, decimal_places=2)
    stock = models.IntegerField()
    categoria = models.CharField(max_length=50, blank=True, null=True)

    def __str__(self):
        return self.nombre

class Venta(models.Model):
    usuario = models.ForeignKey(Usuario, on_delete=models.CASCADE)
    fecha = models.DateTimeField(auto_now_add=True)
    total = models.DecimalField(max_digits=10, decimal_places=2)

class DetalleVenta(models.Model):
    venta = models.ForeignKey(Venta, on_delete=models.CASCADE)
    producto = models.ForeignKey(Producto, on_delete=models.CASCADE)
    cantidad = models.IntegerField()
    subtotal = models.DecimalField(max_digits=10, decimal_places=2)

```

3. *Se crean los serializers, los cuales transformarán los modelos a json*

```python
from rest_framework import serializers
from .models import Usuario, Producto, Venta, DetalleVenta

class UsuarioSerializer(serializers.ModelSerializer):
    class Meta:
        model = Usuario
        fields = '__all__'

class ProductoSerializer(serializers.ModelSerializer):
    class Meta:
        model = Producto
        fields = '__all__'

class VentaSerializer(serializers.ModelSerializer):
    class Meta:
        model = Venta
        fields = '__all__'

class DetalleVentaSerializer(serializers.ModelSerializer):
    class Meta:
        model = DetalleVenta
        fields = '__all__'

```

4. *Se crean las vistas para cada modelo con ModelViewSet, donde se encuentran todos los objetos de cada modelo y su serializer.*

   Al heredar de ModelViewSet, cada uno de estos ViewSets proporciona automáticamente las acciones CRUD básicas: Listar,Crear, Leer, Actualizar y Eliminar.

```python
from django.shortcuts import render
from rest_framework import viewsets
from .models import Usuario, Producto, Venta, DetalleVenta
from .serializers import UsuarioSerializer, ProductoSerializer, VentaSerializer, DetalleVentaSerializer

class UsuarioViewSet(viewsets.ModelViewSet):
    queryset = Usuario.objects.all()
    serializer_class = UsuarioSerializer

class ProductoViewSet(viewsets.ModelViewSet):
    queryset = Producto.objects.all()
    serializer_class = ProductoSerializer

class VentaViewSet(viewsets.ModelViewSet):
    queryset = Venta.objects.all()
    serializer_class = VentaSerializer

class DetalleVentaViewSet(viewsets.ModelViewSet):
    queryset = DetalleVenta.objects.all()
    serializer_class = DetalleVentaSerializer

```

5. *Se crean las urls con ayuda de un router.*

   Cada register asocia una URL con un ViewSet específico.

Esto generará automáticamente las rutas estándar como /usuarios/, /usuarios/{id}/, etc., para realizar operaciones sobre los usuarios.

```python
from django.urls import path, include
from rest_framework.routers import DefaultRouter
from .views import UsuarioViewSet, ProductoViewSet, VentaViewSet, DetalleVentaViewSet

router = DefaultRouter()
router.register(r'usuarios', UsuarioViewSet)
router.register(r'productos', ProductoViewSet)
router.register(r'ventas', VentaViewSet)
router.register(r'detalle_ventas', DetalleVentaViewSet)

urlpatterns = [
    path('', include(router.urls)),
]
```

b. *Frontend*

1. *Api*

Esto permite realizar solicitudes HTTP para obtener datos sobre productos, usuarios y ventas desde el servidor.

```javascript
// src/api/api.js
import axios from 'axios';

const API_URL = 'http://localhost:8000/api';

// Configuracion base para  solicitudes HTTP
const api = axios.create({
    baseURL: API_URL,
    headers: { 'Content-Type': 'application/json',
    },
});

export default api;

export const fetchProductos = async () => {
    const response = await axios.get(`${API_URL}/productos/`);
    return response.data;
};

export const fetchUsuarios = async () => {
    const response = await axios.get(`${API_URL}/usuarios/`);
    return response.data;
};

export const fetchVentas = async () => {
    const response = await axios.get(`${API_URL}/ventas/`);
    return response.data;
};

// Nuevas funciones a futuro para solicitudes
```

2. *App*

Se utiliza React Router para gestionar la navegación entre diferentes páginas o componentes.

La aplicación permite a los usuarios iniciar sesión, registrarse y acceder a diferentes secciones dependiendo de su rol (usuario, empleado o administrador).

Utiliza componentes para mostrar listas de productos, usuarios y ventas, así como formularios para crear nuevos productos y usuarios.

```jsx
// src/App.js

import React, { useState } from 'react';
import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';
import Productos from './components/Productos';  // Cambiado a Productos
import ProductoList from './components/ProductoList';
import UsuariosList from './components/UsuariosList';
import VentasList from './components/VentasList';
import DetalleVentasList from './components/DetalleVentasList';
import CrearProducto from './components/CrearProducto';
import CrearUsuario from './components/CrearUsuario';
import Registro from './components/Registro';
import Login from './components/Login';
import './styles/App.css';

const HomePage = () => (
  <div className="home-page">
    <header className="App-header">
      <h1>Supermercado Plaza Vea</h1>
      <nav className="navbar">
        <Link to="/login">Iniciar Sesión</Link>
        <Link to="/registro">Registrarse</Link>
      </nav>
    </header>
    <div className="welcome-text">
      <h2>Bienvenido a Supermercado Plaza Vea</h2>
      <p>Por favor, inicie sesión o regístrese para continuar.</p>
    </div>
  </div>
);

function App() {
  const [rol, setRol] = useState(null);

  return (
    <Router>
      <div className="App">
        {rol && <div className="user-info">Hola, {rol}</div>}
        <Routes>
          <Route path="/" element={<HomePage />} />
          <Route path="/login" element={<Login setRol={setRol} />} />
          <Route path="/registro" element={<Registro />} />
          {rol === 'usuario' && (
            <Route path="/productos" element={<Productos />} />
          )}
          {rol === 'empleado' && (
            <>
              <Route path="/productos" element={<ProductoList />} />
              <Route path="/crear-producto" element={<CrearProducto />} />
              <Route path="/detalle-ventas" element={<DetalleVentasList />} />
            </>
          )}
          {rol === 'administrador' && (
            <>
              <Route path="/productos" element={<ProductoList />} />
              <Route path="/usuarios" element={<UsuariosList />} />
              <Route path="/ventas" element={<VentasList />} />
              <Route path="/detalle-ventas" element={<DetalleVentasList />} />
              <Route path="/crear-producto" element={<CrearProducto />} />
              <Route path="/crear-usuario" element={<CrearUsuario />} />
            </>
          )}
        </Routes>
        <footer className="footer">
          © 2024 Supermercado Plaza Vea
        </footer>
      </div>
    </Router>
  );
}

export default App;
```

3. *Componentes*

Aquí se muestran los componentes con los que se forma cada ruta dentro de la página principal

Name	Last commit message	Last commit date
..		
CrearProducto.js	token de verificación	2 hours ago
CrearUsuario.js	Creación de token para el form CrearUsuario.js	47 minutes ago
DetalleVentasList.js	Conexion back - front	yesterday
Login.js	token de verificación	2 hours ago
Navbar.js	Creación de token para el form CrearUsuario.js	47 minutes ago
ProductoList.js	token de verificación	2 hours ago
Productos.js	token de verificación	2 hours ago
Registro.js	creación nuevo rol usuario para registro.js del front	2 hours ago
Usuarios.js	Conexion back - front	yesterday
UsuariosList.js	Conexion back - front	yesterday
Ventas.js	Conexion back - front	yesterday
VentasList.js	Conexion back - front	yesterday

Login se muestra para saber si el rol del usuario

```jsx
import React, { useState } from 'react';
import axios from 'axios';

const Login = ({ setRol }) => {
    const [email, setEmail] = useState('');
    const [password, setPassword] = useState('');  // Asegúrate de usar "password" aquí

    const handleSubmit = async (event) => {
        event.preventDefault();
        try {
            const response = await axios.post('http://localhost:8000/api/login/', { email, password });  // Envía "pass
            const { token, rol } = response.data;
            localStorage.setItem('token', token);
            setRol(rol);
        } catch (error) {
            console.error('Error al iniciar sesión:', error);
        }
    };

    return (
        <form onSubmit={handleSubmit}>
            <h2>Login</h2>
            <input type="email" value={email} onChange={(e) => setEmail(e.target.value)} placeholder="Email" required /
            <input type="password" value={password} onChange={(e) => setPassword(e.target.value)} placeholder="Contrase
            <button type="submit">Iniciar Sesión</button>
        </form>
    );
};

export default Login;
```

Además, dependiendo del rol del user se mostrarán diferentes navbar

```jsx
// src/components/Navbar.js
import React from 'react';
import { Link } from 'react-router-dom';

const Navbar = ({ rol }) => {
    return (
        <nav>
            <ul>
                {rol === 'usuario' && (
                    <li><Link to="/productos">Productos</Link></li>
                )}
                {rol === 'empleado' || rol === 'administrador' ? (
                    <>
                        <li><Link to="/productos">Productos</Link></li>
                        <li><Link to="/producto-list">ProductoList</Link></li>
                        <li><Link to="/ventas">Ventas</Link></li>
                        <li><Link to="/usuarios-list">UsuariosList</Link></li>
                    </>
                ) : null}
                {rol === 'administrador' ? (
                    <li><Link to="/crear-usuario">CreaUsuario</Link></li>
                ) : null}
            </ul>
        </nav>
    );
};

export default Navbar;
```

Igualmente, dependiendo del rol se mostrará o no ProductoList, dónde se puede eliminar o crear productos.

```jsx
// src/components/ProductoList.js

import React, { useState, useEffect } from 'react';
import { useNavigate } from 'react-router-dom';
import { getProductos, deleteProducto } from '../services/productosService';
import '../styles/ProductoList.css';

const ProductoList = () => {
  const [productos, setProductos] = useState([]);
  const navigate = useNavigate();

  useEffect(() => {
    cargarProductos();
  }, []);

  const cargarProductos = async () => {
    try {
      const data = await getProductos();
      setProductos(data);
    } catch (error) {
      console.error('Error al cargar productos', error);
    }
  };

  const eliminarProducto = async (id) => {
    try {
      await deleteProducto(id);
      cargarProductos(); // Recargar lista después de eliminar
    } catch (error) {
      console.error('Error al eliminar producto', error);
    }
  };

  return (
    <div className="producto-list">
      <h2>Lista de productos</h2>
      <button onClick={() => navigate('/crear-producto')}>Agregar Producto</button>
      <ul>
        {productos.map((producto) => (
          <li key={producto.id}>
            {producto.nombre} - ${producto.precio}
            <button onClick={() => eliminarProducto(producto.id)}>Eliminar</button>
          </li>
        ))}
      </ul>
    </div>
  );
};

export default ProductoList;
```

4. ***Style***

Estos son los archivos css que se usan en cada componente

5. Conclusiones

La combinación de Django como backend y React como frontend ofrece una solución poderosa y eficiente para el desarrollo de aplicaciones web modernas.

Django proporciona un marco robusto y escalable que facilita la creación de APIs RESTful y la gestión de bases de datos, mientras que React permite construir interfaces de usuario interactivas y dinámicas.

6. Autoevaluación

	Chávez Marco	**Choque Eduardo**	**Choquehuanca Brayan**
Chávez Marco	**-**	**16**	**20**
Choque Eduardo	**20**	**-**	**20**
Choquehuanca Brayan	**20**	**16**	**-**