APLICATIVO WEB GENERACION FACTURA ELECTRONICA

Presentado por

EDISON FABIAN GOMEZ CASTILLO

Profesor

Josue Nicolas Pinzon Villamil

Asignatura

PROGRAMACION III

UNIVERSIDAD PEDAGOGICA Y TECNOLOGICA DE COLOMBIA

FACULTAD DE ESTUDIOS A DISTANCIA

ESCUELA DE CIENCIAS TECNOLÓGICAS

TEGNOLOGIA EN PROGRAMACION DE SISTEMAS INFORMATICOS

CREAD: TUNJA

AÑO 2025

INFORME TÉCNICO DEL PROCESO

Proyecto: Aplicativo Web para Login y Creación de Factura Electrónica

1. **Objetivo del Proyecto**

Desarrollar un aplicativo web responsivo que permita a los usuarios autenticarse mediante un API externo y, tras un inicio de sesión exitoso, acceder a un formulario para la creación de facturas electrónicas mediante consumo de otro API.

1. **Descripción General**

El aplicativo consta de dos módulos principales:

Módulo de Login: Pantalla de autenticación donde el usuario ingresa sus credenciales. Se valida el usuario consumiendo el API de autenticación proporcionado. En caso de éxito, redirige al módulo de factura.

Módulo de Factura Electrónica: Formulario donde el usuario autenticado puede ingresar los datos mínimos requeridos para generar una factura electrónica, y enviar esta información al API correspondiente para su procesamiento.

Ambos módulos están diseñados para ser responsivos, adaptándose a distintos tamaños de pantalla para garantizar una buena experiencia en dispositivos móviles y escritorio.

1. **Tecnologías y Herramientas Utilizadas**

HTML5, CSS3, Bootstrap 5: Para estructurar y diseñar una interfaz responsiva y amigable.

JavaScript (ES6+): Para la interacción con los APIs y manejo dinámico del contenido.

Fetch API: Para realizar las solicitudes HTTP a los servicios REST externos.

API REST:

Autenticación: https://pru.clarisacloud.com:8443/seguridad/rest/api/v1/login/

Factura Electrónica: https://pru.clarisacloud.com:8443/api/factura/rest/v1/factura/nacional

1. **Metodología y Desarrollo**

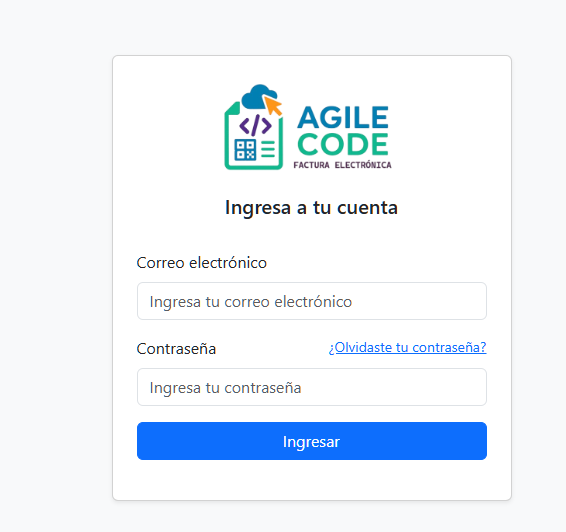
**Login**

Se creó una pantalla de login con campos para usuario y contraseña y boton iniciar sesion.

Al hacer clic en el botón de iniciar sesión, se envía una petición POST al API de autenticación con las credenciales.

Si la respuesta es exitosa, se guarda el token de acceso en localStorage para usarlo posteriormente.

Luego, se redirige al usuario a la página de creación de factura.



**Creación de Factura**

El formulario cuenta con los campos mínimos requeridos por el API para crear una factura nacional.

Inicialmente se pide seleccionar cliente con datos de prueba registrados, hay una opcion para poner consumidor final y otra opcion para registrar cliente. Cuando se elige registrar cliente de la lista desplegable se abre un modal que solicita los datos del cliente.

Los datos ingresados se validan para asegurar que no estén vacíos o mal formateados.

Al enviar el formulario, se construye un objeto JSON con los datos y se envía mediante POST al API de factura, usando el token de autenticación guardado.

Se maneja la respuesta mostrando mensajes de éxito o error para el usuario.



**Responsividad y Adaptabilidad**

Se utilizó Bootstrap 5 para asegurar que los formularios y componentes se ajusten automáticamente a distintos tamaños de pantalla.

Se probaron los módulos en dispositivos móviles y desktop para verificar el correcto comportamiento.

1. **Problemas Encontrados y Soluciones**

CORS: Al consumir APIs externas, se presentaron problemas de CORS, solucionados configurando adecuadamente el backend (en pruebas) y asegurando que el frontend usara HTTPS.

Manejo de tokens: Para mantener la sesión, se implementó almacenamiento seguro del token en localStorage y su inclusión en las cabeceras de las solicitudes API.

Validaciones: Se agregaron validaciones front-end para evitar enviar datos incompletos o erróneos, mejorando la experiencia del usuario y la calidad de la información enviada.

1. **Pruebas Realizadas**

Se probaron inicios de sesión con credenciales válidas e inválidas, verificando manejo correcto de errores.

Se ingresaron facturas con diferentes datos para comprobar que la API respondía adecuadamente.

Se verificó la correcta visualización en diferentes dispositivos y resoluciones.

1. **Conclusiones y Recomendaciones**

El desarrollo del aplicativo permitió comprender el consumo de APIs REST en un entorno real.

Se logró una interfaz responsiva y funcional para usuarios finales.

Se recomienda mejorar la gestión de sesiones y agregar opciones de recuperación de contraseña y cierre de sesión.

Para futuros desarrollos, integrar validaciones más robustas y manejo de errores detallado en el frontend.

1. **Anexos**

Código fuente del proyecto (carpeta con el proyecto).

Nota: Para este proyecto se utilizo la funcion fetch() para cargar datos desde archivos locales por tal motivo el proyecto solo puede utilizarse desde un servidor (Se puede levantar un servidor local ). Los navegadores bloquean por seguridad las solicitudes fetch() a archivos locales (file://) para evitar que scripts maliciosos accedan a archivos del usuario.