

Front-end

1. Estructura de directorios.

La estructura del proyecto frontend de FacturaApp se ha organizado en directorios lógicos que separan claramente los recursos estáticos, las páginas HTML y los módulos funcionales. Esta organización facilita la mantenibilidad y escalabilidad del proyecto a medida que crecen sus funcionalidades.

```
FacturaApp/
├── assets/
│   ├── css/
│   │   ├── sb-admin-2.css
│   │   └── style.css
│   ├── img/
│   │   ├── esquema-base-datos.png
│   │   ├── favicon.png
│   │   ├── forgot.jpg
│   │   ├── login-img.png
│   │   ├── loginimage.jpg
│   │   └── logo.png
│   └── js/
│       ├── actions_reports.js
│       ├── article_actions.js
│       ├── company_actions.js
│       ├── customer_actions.js
│       ├── dashboard_vars.js
│       ├── factura_actions.js
│       ├── global.js
│       ├── menu_actions.js
│       ├── menu_functions.js
│       ├── utils.js
│       └── vista_facturas.js
├── index.html
├── forgot-password/
│   └── index.html
├── register/
│   └── index.html
├── pages/
│   └── report.html
├── favicon.ico
├── dashboard/
│   └── index.html
```

← funciones JS necesarias para la logica del negocio

← Página principal (inicio de sesión)

← Carpeta para recuperación de contraseña

← Página de con instrucciones para recuperar la cuenta

← Carpeta de registro de nuevos usuarios

← Página de registro nuevo usuario(empresa)

← Página de reportes solo para admin (para mi)

← Icono del navegador

← Módulo visual principal de usuario

← Página en la que esta toda la lógica del negocio (creación de artículos, clientes, facturas, envió emails, creación de PDF)

Justificación de la estructura

Esta organización cumple con buenas prácticas de desarrollo web, ya que:

- Separa claramente lógica (JS), estilo (CSS) y contenido (HTML).
- Agrupa funcionalidades por módulo, lo que permite escalabilidad.
- Facilita la navegación y mantenimiento del código por parte de nuevos desarrolladores

2. Planificar las tareas de programación front-end

Para el desarrollo del frontend de **FacturaApp**, he diseñado una planificación clara y estructurada que abarca tanto la construcción de las interfaces HTML/CSS como la programación funcional en JavaScript. El objetivo ha sido implementar una interfaz de usuario completa y funcional que permita crear, consultar y gestionar todos los datos empresariales clave antes de su envío al backend desarrollado en Django.

Fases de planificación de tareas

Diseño de la arquitectura de pantallas

Se decidió que la aplicación contaría con 4 pantallas principales, cada una con una finalidad específica:

Ruta	Función principal
/index.html	Pantalla de inicio de sesión del usuario.
/register/index.html	Formulario para el registro de nuevos usuarios.
/forgot-password/index.html	Página de ayuda para recuperar el acceso a la cuenta.
/dashboard/index.html	Página principal tras login, núcleo del área de gestión.

Maquetación HTML y estilo CSS

- Se crearon las páginas base con estructura semántica HTML5.
- Se definieron estilos consistentes desde assets/css/style.css.
- Se aplicó un diseño limpio, responsive y compatible con escritorio y dispositivos móviles.
- Se organizó todo dentro de carpetas funcionales: assets/css, assets/img, assets/js.

Implementación de lógica en JavaScript (JS)

En la vista **dashboard**, se concentró la lógica del negocio. Desde ahí, mediante JavaScript puro, se programaron funcionalidades para:

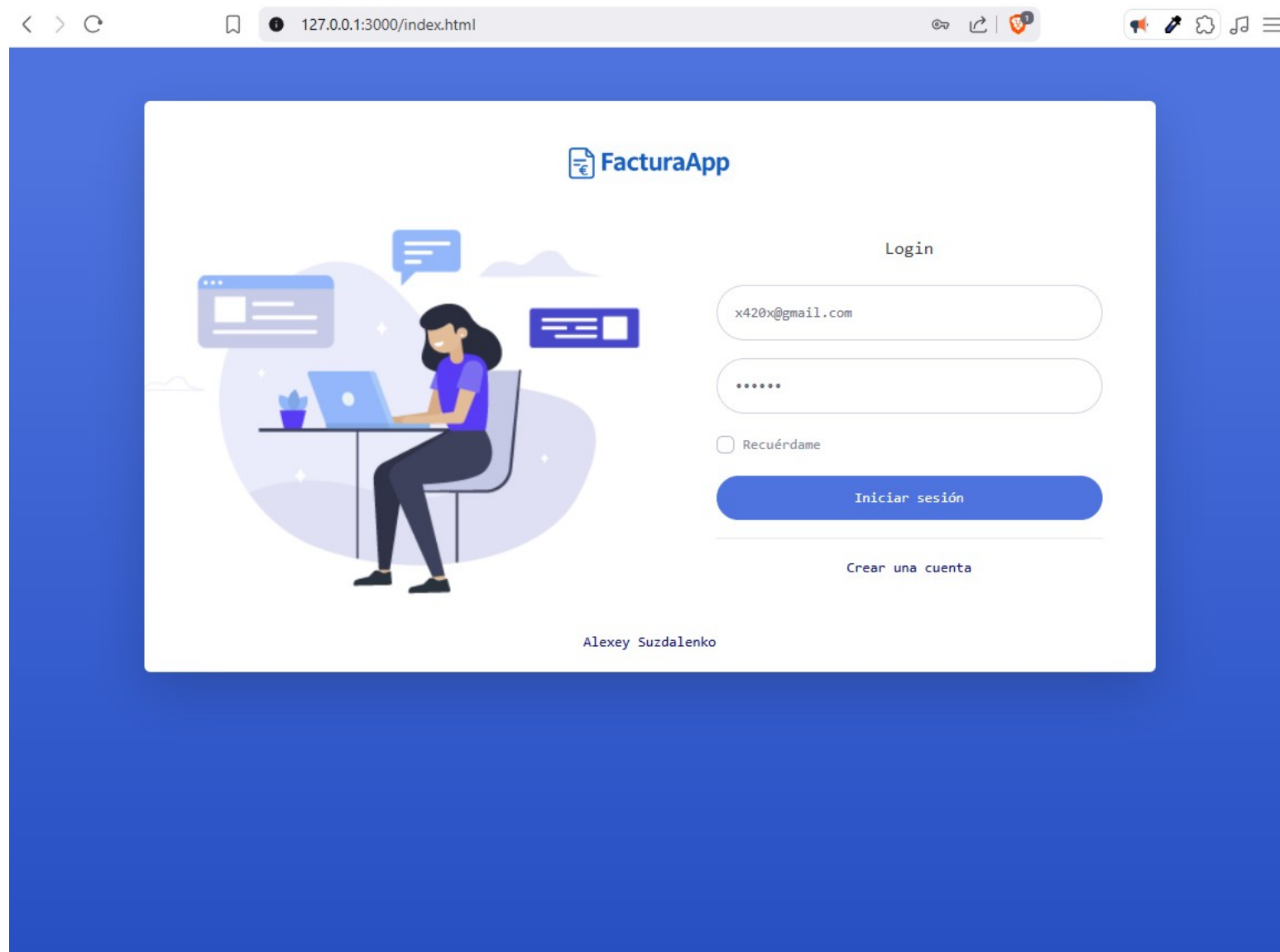
- Mostrar dinámicamente formularios según el módulo activo (empresa, artículos, clientes, facturas...).
- Capturar datos introducidos por el usuario en tiempo real.
- Calcular totales en líneas de factura (precio x cantidad, IVA, subtotal, etc.).
- Generar estructuras de datos listas para ser enviadas al backend Django (en JSON).
- Gestionar los formularios de creación, modificación y visualización de:
 - Datos de la empresa
 - Artículos (productos o servicios)
 - Clientes
 - Facturas y líneas de factura

Preparación para la comunicación con el backend

- Se estructuraron las llamadas a funciones fetch en JS para enviar/recibir datos con el backend (Django).
- Cada módulo JS maneja sus propias operaciones (company_actions.js, factura_actions.js, etc.), organizadas dentro de assets/js/.
- Se establecieron puntos clave para implementar los endpoints API en Django una vez esté listo el backend.

La planificación front-end ha seguido una lógica modular, separando visualización, comportamiento y lógica de negocio. Esto ha permitido construir una interfaz profesional y extensible, lista para conectarse a un backend en producción. Cada pantalla se ha implementado de forma progresiva, completando primero la estructura y luego la funcionalidad asociada en JavaScript.

3. Implementación y maquetación al menos de la página home completamente funcional.



La página de inicio de sesión (index.html) de FacturaApp ha sido diseñada y desarrollada con una estructura profesional, funcional y visualmente atractiva. Esta representa la puerta de entrada a la aplicación y su diseño responde tanto a criterios estéticos como funcionales.

Diseño y maquetación

- La interfaz se ha construido utilizando **HTML5 semántico**, asegurando accesibilidad y correcta estructura.
- El diseño visual está basado en **CSS3 personalizado** dentro del archivo style.css, respetando la paleta de colores corporativa de FacturaApp.
- Se ha dividido el layout en dos columnas principales:
 - **Columna izquierda:** ilustración representativa con estilo moderno, reforzando el contexto tecnológico y profesional del proyecto.
 - **Columna derecha:** formulario de login y elementos interactivos.

Elementos implementados

- **Logo corporativo** en la parte superior, con icono e identidad visual de la marca.
- **Formulario de inicio de sesión** con campos para:
 - Correo electrónico
 - Contraseña
 - Casilla "Recuérdame"

- **Botón de inicio de sesión** con estilo interactivo y accesible.
- Enlace de acceso directo a la creación de cuenta (register) para usuarios nuevos.
- Pie de página con firma de autor: “Alexey Suzdalenko”.

Funcionalidad

- El formulario está preparado para enviar datos al backend mediante POST para iniciar sesión.
- Se han aplicado **validaciones de entrada básicas con JavaScript** (campos obligatorios).
- El diseño es **responsive**, adaptándose correctamente a pantallas móviles y de escritorio.

El resultado es una **página de inicio funcional, intuitiva y coherente con la identidad de FacturaApp**, que cumple con su propósito principal: permitir el acceso seguro al área privada de la aplicación. La estructura está lista para conectarse a la lógica de autenticación en el backend Django.

4. Despliegue de la aplicación en un servidor web local.

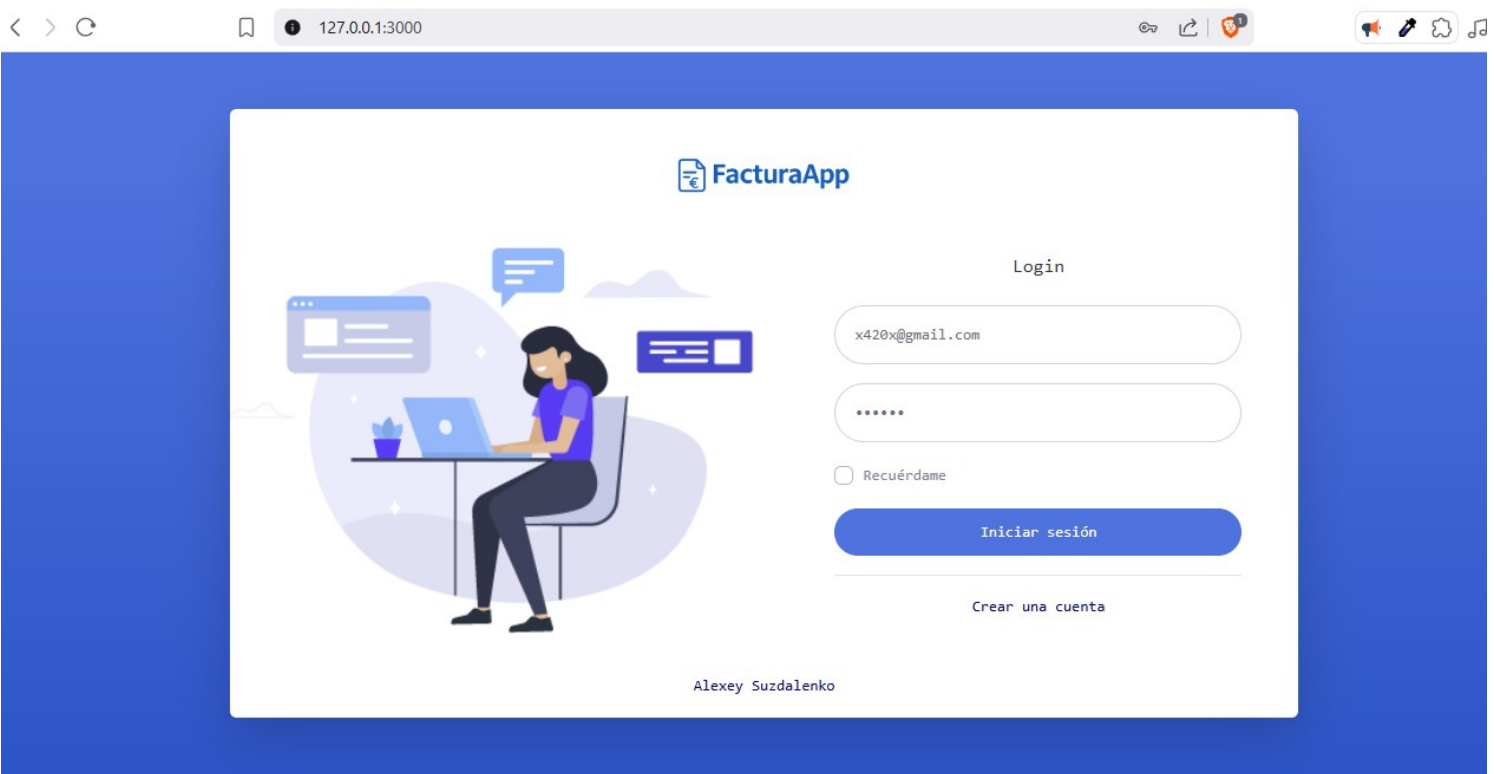
Para verificar y probar el correcto funcionamiento del frontend de **FacturaApp**, se realizó el despliegue local de la aplicación en un entorno de desarrollo personal. Este paso es fundamental para comprobar visualmente el resultado del código HTML, CSS y JavaScript, y garantizar su correcto comportamiento antes de subirlo a producción.

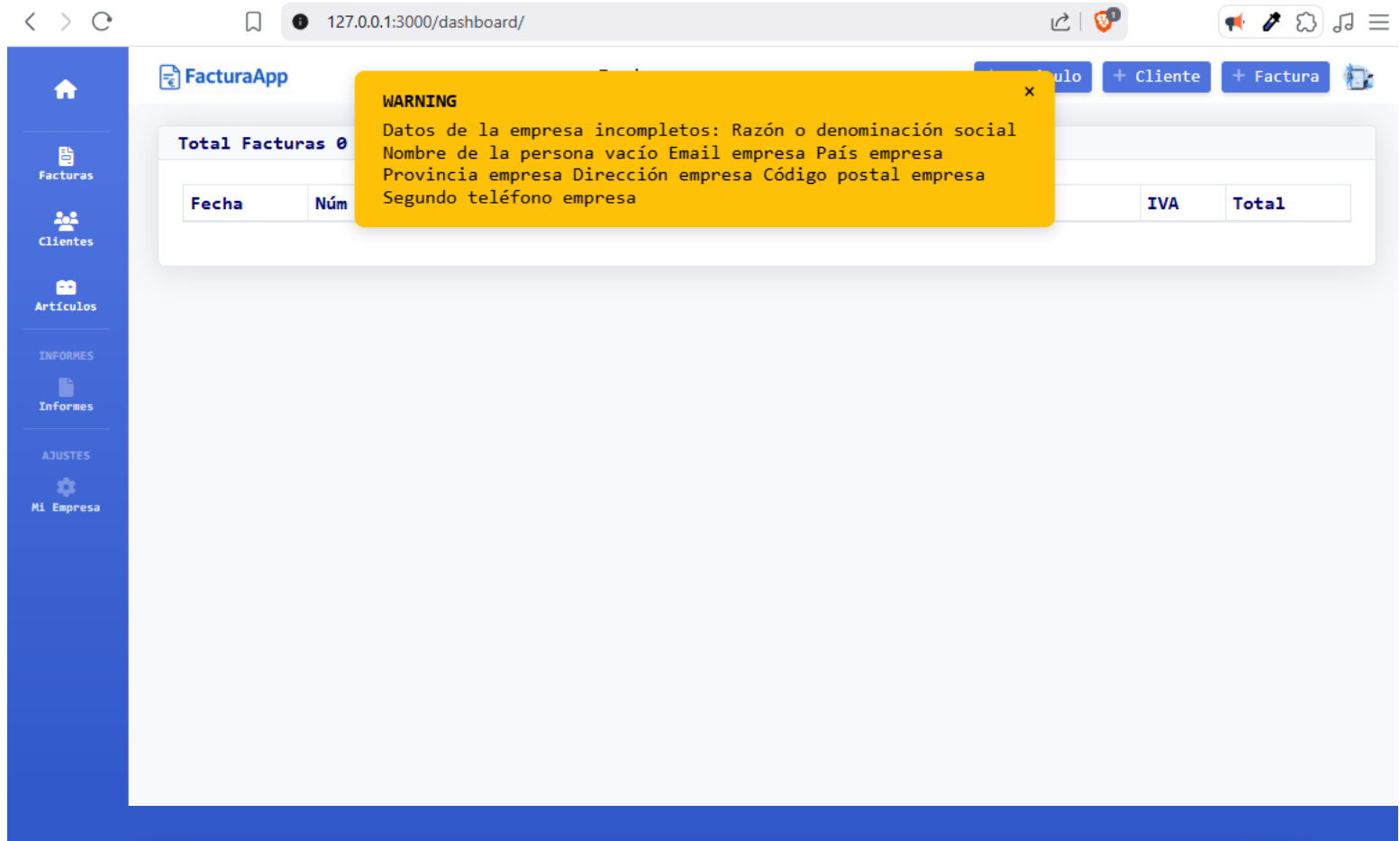
Método de despliegue utilizado

Para el despliegue local, utilicé el entorno de desarrollo **Visual Studio Code** junto con la extensión **Live Server**, que permite ejecutar un servidor local con recarga automática de páginas.

Pasos realizados:

1. Se abrió el proyecto en Visual Studio Code.
2. Se instaló la extensión “Live Server”.
3. Se hizo clic derecho sobre el archivo index.html y se seleccionó “Open with Live Server”.





4. El navegador abrió la dirección `http://127.0.0.1:3000/`, mostrando la página de inicio (`index.html`)

Verificaciones realizadas

- Visualización de todos los elementos gráficos, formularios y botones.
- Comprobación de la funcionalidad JavaScript (formularios, validaciones, interacción).
- Confirmación de la estructura responsive del diseño en distintas resoluciones.
- Pruebas de enlaces entre pantallas (navegación entre login, registro y recuperación de contraseña).

El despliegue local ha sido fundamental para desarrollar y depurar la interfaz de **FacturaApp** de forma efectiva. Gracias a esta prueba en entorno local, se garantizó la calidad visual y funcional de las páginas antes de subir la aplicación a un servidor remoto.

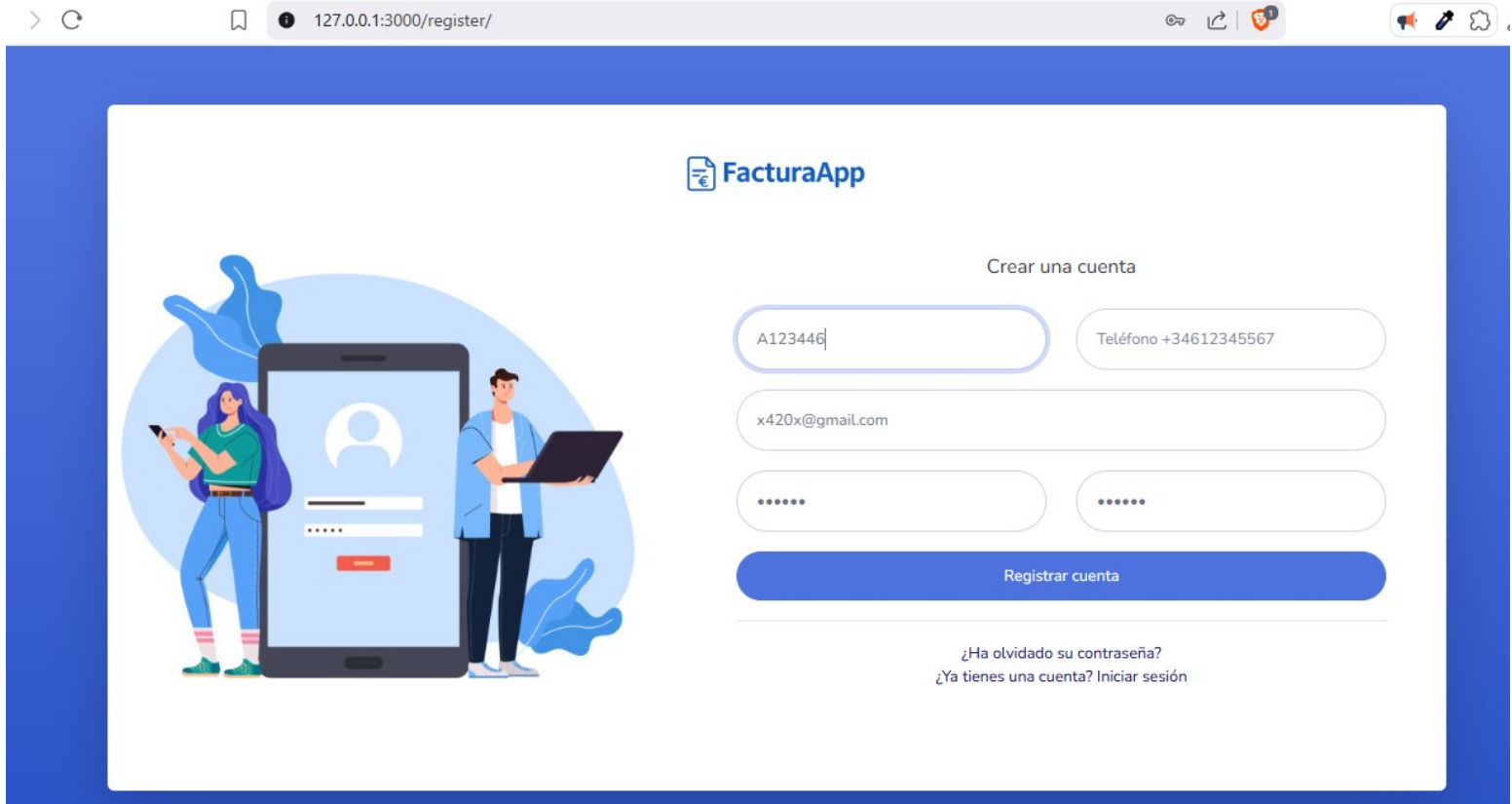
Front-end 2

1. Implementar el resto de las secciones front-end

Tras desarrollar la página principal de inicio de sesión (`index.html`), se implementaron el resto de las secciones necesarias para el correcto funcionamiento del frontend de FacturaApp. Estas secciones forman la estructura visual completa que permite al usuario interactuar con las distintas funcionalidades de la aplicación.

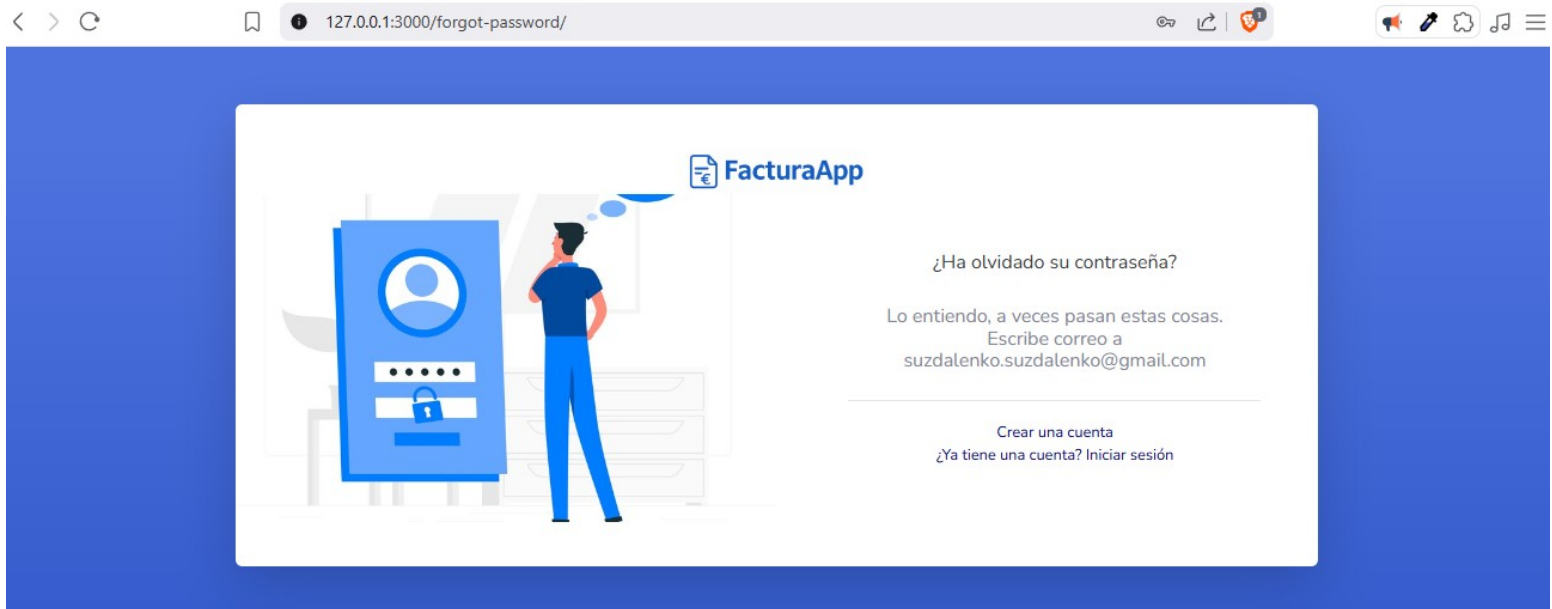
`/register/index.html`

- Página para el **registro de nuevos usuarios**.
- Contiene formulario con campos como nombre, email, contraseña, confirmación, etc.
- Estilizada con la misma identidad visual que el resto del sitio.
- Validación de datos con JavaScript antes de enviar al backend.



/forgot-password/index.html

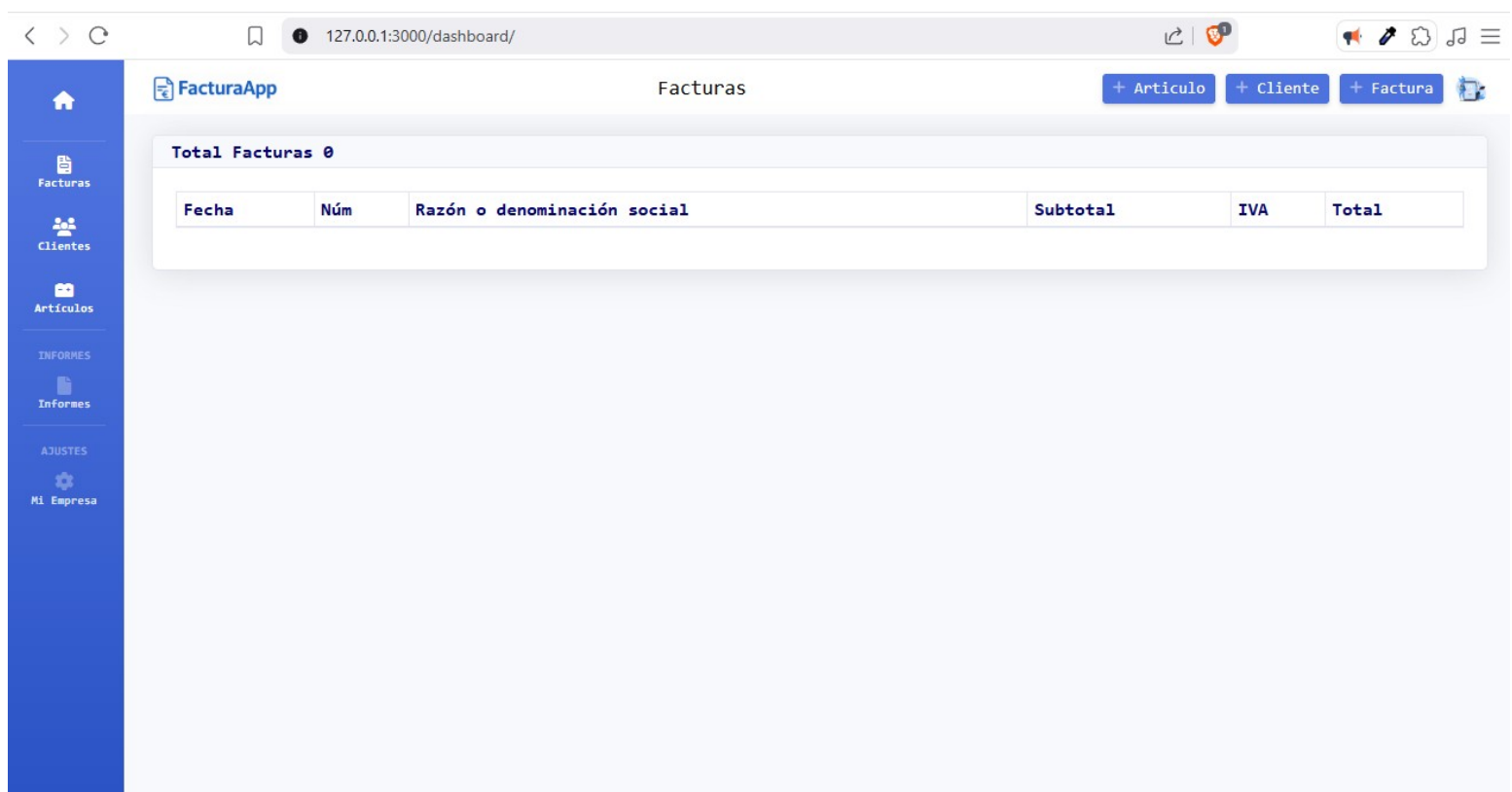
- Página para **recuperación de cuenta**.
- Muestra instrucciones al usuario para restablecer la contraseña.
- Diseño coherente con el resto de la aplicación.



/dashboard/index.html

- Página principal tras el inicio de sesión.
- Contiene el **núcleo del negocio**: gestión de clientes, artículos, facturas, y datos de empresa.
- Integra múltiples secciones mostradas dinámicamente mediante JavaScript:
 - Formulario de empresa
 - Módulo de clientes
 - Módulo de artículos

- Módulo de facturas
- Reportes y estadísticas
- Utiliza scripts modulares (assets/js/) para manejar cada bloque de funcionalidades.
- Estructura responsive adaptable a diferentes resoluciones.



Formulario de la propia empresa



Vista de creación de la factura:

127.0.0.1:3000/dashboard/#CreateInvoice

FacturaApp

Crear Factura

+ Artículo

+ Cliente

+ Factura

Tipo Factura

Tipo
Ordinaria/Rectificativa/Abono
Factura Ordinaria

Datos cliente y tipo de factura

Número cliente
NIF CIF cliente
Razón o denominación social
Nombre cliente

Datos vehículo

Matrícula
Marca/Modelo/Kilómetros

Líneas de factura

Código Art.	Descripción Art.	Cantidad	Precio	% Dto.	Importe	IVA
		1		0		21 %
+ Línea						
	Mano de obra		40,00			21 %
						Subtotal: 0
						IVA: 0
						Total: 0

Observaciones

Crear Factura

Organización y modularidad

- Todo el frontend se basa en HTML, CSS y JavaScript puros.
- Se usa una **estructura modular de scripts** en assets/js/, separando acciones por entidad (clientes, facturas, etc.).
- Se mantiene la coherencia visual con el uso de estilos comunes desde style.css.

Gracias a la implementación completa de todas las secciones del frontend, el usuario puede navegar por la aplicación, gestionar su empresa y sus documentos sin necesidad de recargar ni salir del entorno visual. Todo esto ha sido desarrollado en código limpio, funcional y fácilmente conectable con el backend de Django.

2. Desplegar la aplicación en un servidor web remoto, con acceso desde internet.

Una vez completado el desarrollo local del frontend de **FacturaApp**, se procedió al despliegue de la aplicación en un servidor remoto con acceso público a través de internet. Para ello se utilizó el servicio de alojamiento estático **Firebase Hosting** de Google.

Plataforma de despliegue: Firebase Hosting

Firebase fue la plataforma elegida por su facilidad de uso, gratuidad en proyectos pequeños y velocidad de implementación. Esta solución permite alojar proyectos web estáticos compuestos por HTML, CSS, JavaScript e imágenes.

Ventajas de Firebase Hosting:

- Acceso rápido y seguro mediante HTTPS.
- Fácil integración con herramientas de desarrollo.
- Configuración y despliegue en pocos minutos.
- URL personalizada sin coste adicional.

Pasos realizados para el despliegue

1. Se instaló la CLI de Firebase:

```
npm install -g firebase-tools
```

2. Se inició sesión con una cuenta de Google:

```
firebase login
```

3. Se inicializó el proyecto en la raíz del frontend:

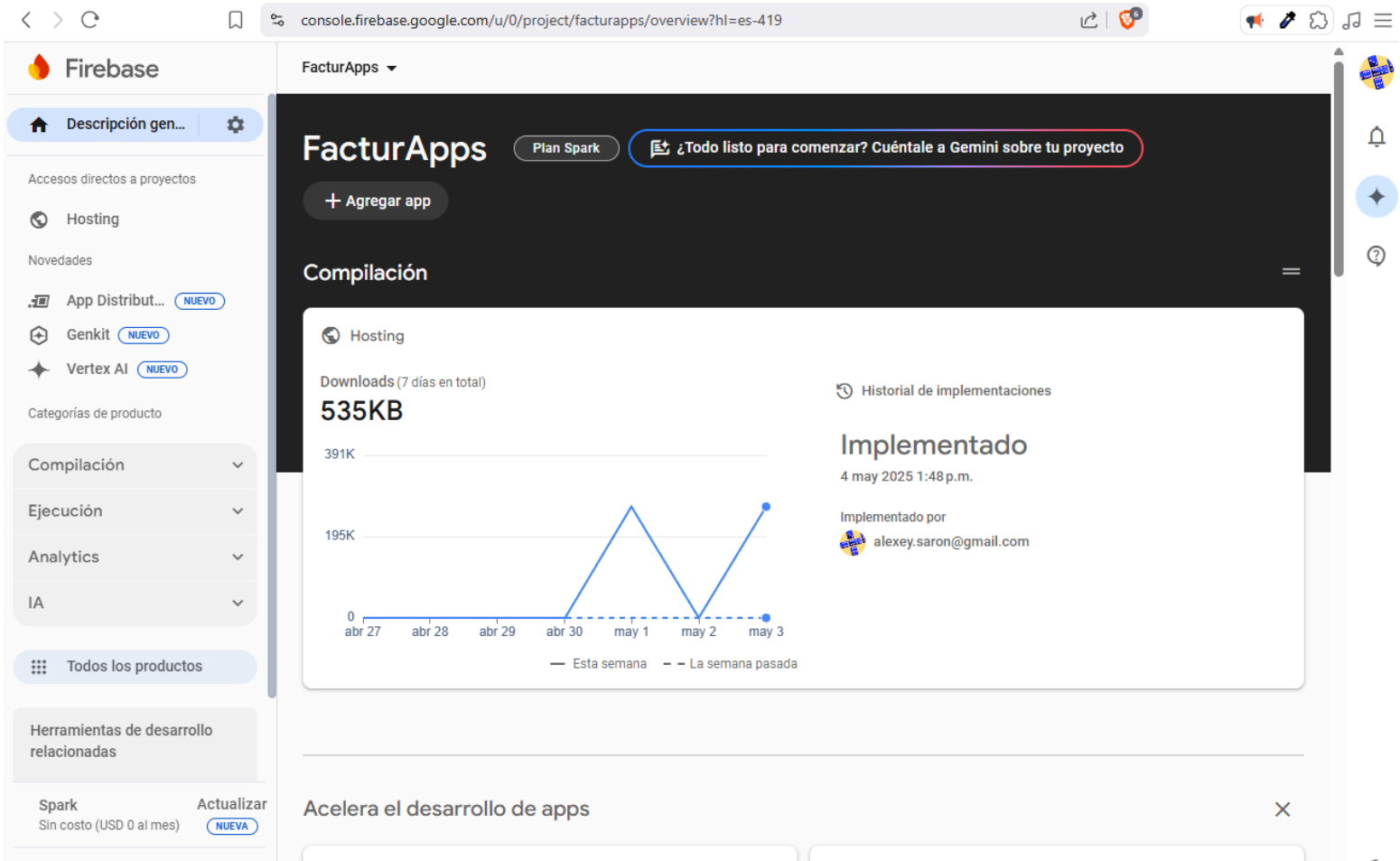
```
firebase init
```

- Se seleccionó "Hosting"
- Se configuró la carpeta pública como `.` o `front-html`

4. Se construyó el contenido y se desplegó:

```
firebase deploy
```

Proyecto FacturaApp en firebase:



Resultado

El proyecto quedó publicado en la siguiente URL accesible desde cualquier navegador:

<https://facturapps.web.app/>

Esto permite probar la interfaz en un entorno real, mostrarla públicamente y facilitar el acceso a usuarios o clientes.

