



LAUNCHER EMPRESARIAL

Óscar Hernández, Jorge García e Iván Fernández 2DAM



Índice

¡Error! Marcador no definido.

1. Introducción

2. Objetivos

2.1. Objetivos Generales

2.2. Alcance y Especificaciones

3. Tecnologías y Herramientas

4. Arquitectura del Proyecto

4.1. Estructura de Carpetas

4.2. Flujo de Datos

5. Descripción de los Módulos

5.1. Ventas

5.2. Facturación

5.3. Inventario

5.4. Marketing

6. Implementación y Detalles Técnicos

6.1. Carga y Gestión de Datos

6.2. Generación de Gráficos con D3.js

6.3. Diseño y Estilos

7. Conclusiones y Sugerencias Futuras

1. Introducción

El proyecto Launcher Empresarial JOI surge con el objetivo de ofrecer un punto de entrada unificado para gestionar módulos críticos de un sistema empresarial, tales como Ventas, Facturación, Inventario y Marketing. Inspirado en soluciones como Odoo, se ha desarrollado una interfaz minimalista en blanco y negro que centraliza el acceso y mejora la experiencia del usuario mediante visualizaciones interactivas de datos.

2. Objetivos

2.1. Objetivos Generales

- **Interfaz Atractiva e Intuitiva:** Crear un entorno gráfico moderno que facilite la interacción y el acceso rápido a las distintas áreas.
- **Navegación Eficiente:** Diseñar un dashboard central que actúe como lanzador de módulos, utilizando menús laterales o cuadrículas de iconos.
- **Visualización Avanzada de Datos:** Implementar gráficos interactivos (barras, líneas, pastel, gauge, etc.) que permitan un análisis dinámico de las estadísticas empresariales.
- **Gestión de Usuarios y Roles:** Integrar un sistema de autenticación robusto (por ejemplo, mediante Firebase) para asegurar un control de acceso adecuado.

2.2. Alcance y Especificaciones

El proyecto contempla el desarrollo de:

- **Pantallas de Autenticación:** Formularios de login y registro, con opción de autenticación a través de cuentas de Google.
- **Dashboard Central:** Una vista principal que centralice el acceso a los módulos:

- **Ventas:** Seguimiento de tendencias y datos agregados.
- **Facturación:** Estadísticas de ingresos y seguimiento del estado de las facturas.
- **Inventario:** Gestión del stock y monitoreo de productos críticos.
- **Marketing:** Indicadores de rendimiento y análisis de inversión versus retorno.

3. Tecnologías y Herramientas

Svelte: Framework para el desarrollo de interfaces reactivas y componentes modulares.

SCSS: Preprocesador CSS que permite mantener estilos organizados y escalables.

D3.js: Biblioteca JavaScript utilizada para la creación de gráficos interactivos y visualizaciones dinámicas.

CSV: Archivo dashboard-data.csv (ubicado en public/data) que simula los datos empresariales.

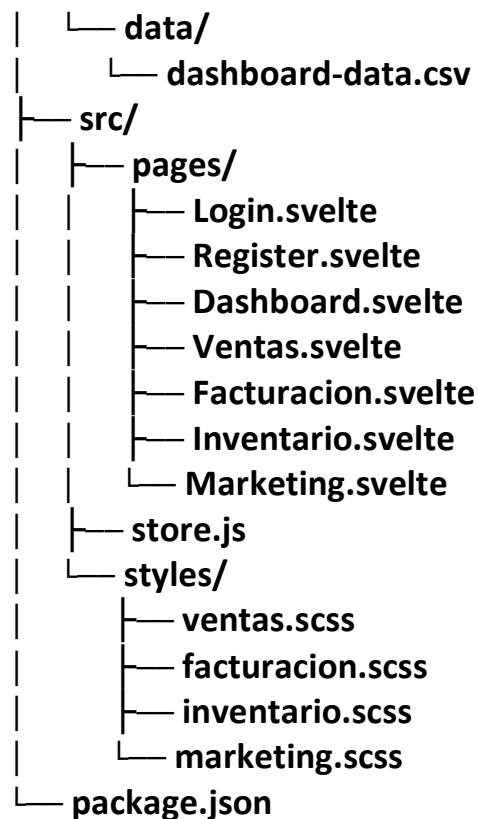
Firebase: Plataforma para la autenticación de usuarios.

4. Arquitectura del Proyecto

4.1. Estructura de Carpetas

El proyecto se organiza de la siguiente forma:

public/



Esta estructura modular permite una clara separación de responsabilidades y facilita futuras ampliaciones y mantenimientos.

4.2. Flujo de Datos

Cada módulo carga sus datos directamente desde el archivo CSV. El proceso consiste en:

- **Parseo del CSV:** Separación del contenido en filas y columnas.
- **Identificación de la Fila Correspondiente:** Cada módulo (Ventas, Facturación, Inventario y Marketing) extrae sus datos de la fila que le corresponde.
- **Conversión y Validación:** Transformación de los datos (por ejemplo, cadenas a números) y validación básica para asegurar su correcta visualización.

5. Descripción de los Módulos

5.1. Ventas

- **Gráficos:**
 - *Gráfico de Barras:* Representa la tendencia de ventas diarias con barras de esquinas redondeadas y efectos interactivos.
 - *Gráfico de Líneas:* Ilustra la evolución continua de las ventas.
- **Datos Agregados:** Se muestran el total de ventas y los ingresos acumulados.
- **Procesamiento:** En Ventas.svelte se parsea la cadena de datos de la fila "Ventas", convirtiéndola en un array numérico.

5.2. Facturación

- **Gráficos:**
 - *Gráfico de Pastel:* Representa la proporción de facturas pagadas versus pendientes.
 - *Gauge Chart:* Indica los ingresos totales en relación a un valor máximo predefinido.
- **Datos Relevantes:** Incluye ingresos totales, número de facturas y el porcentaje de facturas pagadas, extraídos de la fila “Facturación”.

5.3. Inventario

- **Gráficos:**
 - *Gráfico de Barras Horizontal:* Muestra el total de productos en stock.
 - *Donut Chart:* Ilustra el porcentaje de productos críticos en el inventario.
- **Datos Clave:** Se destacan el total de productos y la cantidad de productos críticos.

5.4. Marketing

- **Gráficos:**
 - *Gráfico de Líneas:* Visualiza la evolución de métricas clave (como inversión o alcance).
 - *Scatter Plot:* Permite analizar la relación entre variables, por ejemplo, inversión y ROI.
- **Datos Relevantes:** Se muestran indicadores como el número total de campañas y la inversión acumulada.

6. Implementación y Detalles Técnicos

6.1. Carga y Gestión de Datos

Cada componente realiza los siguientes pasos:

- **Lectura del CSV:** Utiliza `fetch()` para acceder al archivo `dashboard-data.csv`.
- **Parseo y Extracción:** Se dividen los datos en filas y se extrae la información correspondiente a cada módulo.
- **Conversión y Validación:** Los datos se transforman al formato requerido (números y arrays) y se validan para evitar errores.

6.2. Generación de Gráficos con D3.js

El uso de D3.js permite:

- **Configuración de Escalas y Ejes:** Se definen escalas (p. ej., `d3.scaleBand` para barras y `d3.scaleLinear` para líneas) que aseguran una correcta representación.
- **Dibujo de Elementos:** Se crean elementos gráficos (rectángulos, líneas, arcos) con transiciones suaves y efectos de interacción.
- **Personalización de Gráficos Especiales:** Se implementan gráficos como gauge, donut y scatter plot utilizando funciones específicas (como `d3.pie()`, `d3.arc()` y `d3.line()`).

6.3. Diseño y Estilos

- **Paleta Monocromática:** Uso predominante de blancos, negros y grises para un aspecto profesional.
- **Tipografía Moderna:** Fuentes que mejoran la legibilidad y complementan el diseño minimalista.
- **Uniformidad Visual:** Los archivos SCSS aseguran consistencia en márgenes, alineaciones y efectos visuales en cada módulo.

7. Conclusión y Sugerencias Futuras

Conclusiones:

- La estructura modular facilita el mantenimiento y la escalabilidad del sistema.
- La carga y gestión directa de datos asegura un control preciso.
- La integración de D3.js mejora la experiencia interactiva.
- El diseño minimalista refuerza la imagen profesional.

Sugerencias Futuras:

- Incorporar funcionalidad para reorganizar módulos.
- Implementar modo oscuro y opciones de personalización de temas.
- Añadir animaciones avanzadas y mejorar la gestión de errores.
- Integrar fuentes de datos reales y fortalecer la seguridad de acceso.