

DOCUMENTACIÓN TÉCNICA

Sistema de Gestión ALS - Aplicación Web Flask

Autor: Diego Camarena Gutiérrez

DNI: 15970795N

Asignatura: ALS (Análisis, Lógica y Sistemas)

Fecha: Junio 2025

ENTREGA ACADÉMICA - PROYECTO ALS

Sistema de Gestión ALS - Aplicación Web Flask

****Estudiante:**** Diego Camarena Gutiérrez ****DNI:**** 15970795N ****Asignatura:**** ALS (Análisis, Lógica y Sistemas) ****Curso:**** Semestre 6 ****Fecha de entrega:**** Junio 2025

ESTRUCTURA DE LA ENTREGA

```
ALS_proyecto-camarena_gutierrez_diego-15970795N/■■■■ src/ # Código fuente completo
■■■■ app/ # Aplicación Flask principal
■■■■ models/ # Modelos de datos
■■■■ routes/ # Controladores/Rutas
■■■■ forms/ # Formularios WTF
■■■■ templates/ # Plantillas HTML
■■■■ static/ # Archivos estáticos
■■■■ services/ # Lógica de negocio
■■■■ utils/ # Utilidades
■■■■ scripts/ # Scripts de inicialización
■■■■ config.py # Configuración
■■■■ run.py # Punto de entrada
■■■■ requirements.txt # Dependencias
■■■■ README.md # Documentación del proyecto
■■■■ .env.template # Plantilla de variables de entorno
■■■■ doc/ # Documentación académica
■■■■ info.txt # Información del proyecto
■■■■ documentacion_tecnica.md # Documentación completa (PDF)
■■■■ diagrama_clases.md # Diagramas UML de clases
■■■■ diagramas_secuencia.md # Diagramas de secuencia
```

INSTRUCCIONES DE INSTALACIÓN

Requisitos Previos

- Python 3.11 o superior - pip (gestor de paquetes Python) - Navegador web moderno

Pasos de Instalación

1. ****Extraer archivos****

```
# Extraer el archivo ZIP en una carpeta# Navegar a la carpeta src/
```

2. ****Crear entorno virtual****

```
python -m venv venv
```

3. ****Activar entorno virtual****

```
# Windows:venv\Scripts\activate# Linux/Mac:source venv/bin/activate
```

4. ****Instalar dependencias****

```
pip install -r requirements.txt
```

5. ****Configurar variables de entorno****

```
# Copiar .env.template a .envcopy .env.template .env# Editar .env con tus configuraciones (opcional para testing)
```

6. ****Inicializar base de datos****

```
python scripts/init_db.py
```

7. **Ejecutar aplicación**

```
python run.py
```

8. **Acceder a la aplicación** - URL: <http://localhost:5000> - Usuario por defecto: admin - Contraseña por defecto: admin123

FUNCIONALIDADES PRINCIPALES

■ Gestión de Clientes

- Registro y edición de información de clientes - Búsqueda y filtrado - Estadísticas de pedidos por cliente - Eliminación lógica (soft delete)

■ Gestión de Productos

- Catálogo de productos textiles - Gestión de tallas y colores - Control de precios base - Posiciones de personalización

■ Gestión de Procesos

- Tipos de personalización (DTF, Sublimación, Bordado, Vinil) - Calculadora de costes automática - Configuración de precios por proceso - Gestión de tamaños y especificaciones

■ Gestión de Pedidos

- Creación de pedidos paso a paso - Múltiples items por pedido - Personalizaciones por item - Cálculo automático de totales - Estados de pedido (Pendiente, En Proceso, Completado, Entregado, Cancelado) - Gestión de fechas de entrega

■ Calculadora Avanzada

- Cálculo automático según tipo de proceso - Consideración de dimensiones para DTF/Sublimación - Tamaños específicos para bordado - Costes de setup y materiales - Márgenes de utilidad configurables

■ Sistema de Autenticación

- Login/logout seguro - Protección de rutas - Gestión de sesiones

■ Interfaz Responsiva

- Diseño moderno con Bootstrap 5 - Adaptable a dispositivos móviles - Navegación intuitiva - Formularios validados

TECNOLOGÍAS UTILIZADAS

Backend

- **Python 3.11**: Lenguaje principal - **Flask 2.3+**: Framework web - **SQLAlchemy**: ORM para base de datos - **Flask-Login**: Autenticación - **Flask-WTF**: Formularios - **SQLite**: Base de datos

Frontend

- **HTML5**: Estructura - **CSS3**: Estilos - **Bootstrap 5**: Framework CSS - **JavaScript**: Interactividad - **AJAX**: Comunicación asíncrona

ARQUITECTURA

El proyecto sigue el patrón **MVC (Modelo-Vista-Controlador)**:

- **Modelos** (`app/models/`): Gestión de datos y lógica de negocio - **Vistas** (`app/templates/`): Presentación e interfaz de usuario - **Controladores** (`app/routes/`): Lógica de aplicación y manejo de peticiones

Características Arquitectónicas:

- **Separación de responsabilidades** - **Reutilización de código** - **Escalabilidad** - **Mantenibilidad** - **Principios SOLID**

DOCUMENTACIÓN INCLUIDA

1. **info.txt**: Información completa del proyecto y estudiante 2. **documentacion_tecnica.md**: Documentación técnica detallada (convertir a PDF) 3. **diagrama_clases.md**: Diagramas UML de clases del sistema 4. **diagramas_secuencia.md**: Diagramas de secuencia de procesos principales

REPOSITORIO

URL: https://github.com/Diego-CGTZ/ALS_proyecto-CamarenaGuti-rrezDiego-15970795.git

El código fuente completo está disponible en el repositorio de GitHub para revisión y versionado.

CONTACTO

Diego Camarena Gutiérrez DNI: 15970795N Email: [Tu email académico] Universidad: [Nombre de tu universidad]

NOTAS ADICIONALES

- La aplicación incluye datos de ejemplo para facilitar las pruebas - Todos los formularios tienen validación tanto del lado cliente como servidor - Se implementa eliminación lógica para mantener integridad referencial - El sistema de cálculo de costes es completamente automático - La interfaz es completamente responsiva y accesible

****¡Gracias por revisar el proyecto!** ■**