

DOCUMENTACIÓN TÉCNICA

Sistema de Gestión ALS - Aplicación Web Flask

Autor: Diego Camarena Gutiérrez

DNI: 15970795N

Asignatura: ALS (Análisis, Lógica y Sistemas)

Fecha: Junio 2025

DOCUMENTACIÓN TÉCNICA

Sistema de Gestión ALS - Aplicación Web Flask

Autor: Diego Camarena Gutiérrez **DNI:** 15970795N **Asignatura:** ALS (Análisis, Lógica y Sistemas) **Fecha:** Junio 2025

1. INTRODUCCIÓN

1.1 Descripción del Proyecto

El Sistema de Gestión ALS es una aplicación web desarrollada en Flask que permite la gestión integral de:

- **Clientes**: Registro y administración de información de clientes
- **Productos**: Catálogo de productos con especificaciones técnicas
- **Procesos**: Definición de procesos de fabricación
- **Pedidos**: Gestión completa del flujo de pedidos

1.2 Objetivos

- Digitalizar la gestión de procesos empresariales
- Proporcionar una interfaz intuitiva para usuarios
- Automatizar cálculos de costes y tiempos
- Centralizar la información en una base de datos

1.3 Alcance

Sistema web completo con funcionalidades CRUD para todas las entidades, sistema de autenticación y panel de administración.

2. ARQUITECTURA DEL SISTEMA

2.1 Arquitectura General

La aplicación sigue el patrón **MVC (Modelo-Vista-Controlador)**:

```

+-----+ +-----+ +-----+ +-----+ +-----+
VISTA  + CONTROLADOR + MODELO  + (Templates) + (Routes) +
(Modelos) + HTML/CSS/JS + Flask Routes + SQLAlchemy
+-----+ +-----+ +-----+ +-----+ +-----+

```

2.2 Tecnologías Utilizadas

Backend - **Python 3.11**: Lenguaje de programación principal - **Flask 2.3+**: Framework web ligero - **SQLAlchemy**: ORM para base de datos - **Flask-Login**: Gestión de sesiones de usuario - **Flask-WTF**: Manejo de formularios - **Flask-Migrate**: Migración de base de datos

Frontend - **HTML5**: Estructura de páginas - **CSS3**: Estilos y diseño - **Bootstrap 5**: Framework CSS responsivo - **JavaScript**: Interactividad del cliente

4. FUNCIONALIDADES IMPLEMENTADAS

4.1 Sistema de Autenticación

- Login y logout de usuarios - Gestión de sesiones con Flask-Login - Protección de rutas mediante decoradores

4.2 Gestión de Clientes

- **Crear**: Formulario de registro de nuevos clientes - **Leer**: Lista paginada de clientes activos - **Actualizar**: Edición de información de clientes - **Eliminar**: Eliminación lógica (is_active = False)

4.3 Gestión de Productos

- Catálogo completo de productos - Categorización de productos - Control de precios base - Gestión de estado activo/inactivo

4.4 Gestión de Procesos

- Definición de procesos por producto - Cálculo de costes (mano de obra + materiales) - Estimación de tiempos - Vinculación producto-proceso

4.5 Gestión de Pedidos

- Creación de pedidos con validación - Estados: pendiente, en_proceso, completado, cancelado - Cálculo automático de precios totales - Seguimiento de fechas de entrega

4.6 Calculadora de Costes

- Cálculo dinámico basado en: - Precio base del producto - Costes de proceso - Cantidad del pedido - Márgenes de beneficio

5. INTERFAZ DE USUARIO

5.1 Diseño Responsivo

La aplicación utiliza Bootstrap 5 para garantizar: - Adaptabilidad a diferentes tamaños de pantalla - Interfaz moderna y profesional - Componentes UI consistentes

5.2 Navegación

- **Dashboard**: Panel principal con resumen - **Menú lateral**: Navegación por módulos - **Breadcrumbs**: Orientación del usuario - **Botones de acción**: Operaciones CRUD claramente identificadas

5.3 Formularios

- Validación del lado cliente y servidor - Mensajes de error descriptivos - Campos requeridos claramente marcados - Autocompletado en campos relacionados

6. SEGURIDAD

6.1 Medidas Implementadas

- ****CSRF Protection****: Tokens CSRF en todos los formularios - ****SQL Injection****: Uso de SQLAlchemy ORM - ****XSS Protection****: Escape automático en templates - ****Sesiones Seguras****: Configuración de cookies seguras

6.2 Validación de Datos

- Validación del lado servidor con WTForms - Sanitización de entrada de datos - Validación de tipos de datos - Longitud máxima de campos

7. TESTING Y CALIDAD

7.1 Testing

- Scripts de testing para funcionalidades CRUD - Verificación de integridad de datos - Testing de formularios y validaciones

7.2 Calidad del Código

- Estructura modular y organizada - Separación de responsabilidades - Comentarios y documentación - Seguimiento de estándares PEP 8

8. INSTALACIÓN Y DESPLIEGUE

8.1 Requisitos del Sistema

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

8.2 Instalación Local

```
# 1. Clonar repositorio git clone https://github.com/Diego-CGTZ/ALS_proyecto-CamarenaGuti-rrezDiego-15970795.git# 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 cp .env.template .env# 6. Inicializar base de datos python scripts/init_db.py# 7. Ejecutar aplicación python run.py
```

8.3 Acceso a la Aplicación

- URL: http://localhost:5000 - Usuario por defecto: admin - Contraseña por defecto: admin123

9. CONCLUSIONES

9.1 Objetivos Cumplidos

✓ Sistema completo de gestión empresarial ✓ Interfaz intuitiva y responsiva ✓ Funcionalidades CRUD para todas las entidades ✓ Sistema de autenticación robusto ✓ Cálculos automáticos de

costes ✓ Estructura de código mantenible

9.2 Tecnologías Dominadas

- Desarrollo web con Flask - Modelado de datos con SQLAlchemy - Frontend responsivo con Bootstrap - Control de versiones con Git - Testing y debugging

9.3 Posibles Mejoras Futuras

- Implementación de API REST - Sistema de reportes avanzado - Integración con sistemas de pago - Notificaciones en tiempo real - Dashboard con gráficos interactivos - Soporte para múltiples idiomas

10. ANEXOS

10.1 Estructura de Archivos

```
src/■■■■ app/■■■■ __init__.py■■■■ models/■■■■ __init__.py■■■■ ■■■■
cliente.py■■■■ ■■■■ producto.py■■■■ ■■■■ proceso.py■■■■ ■■■■ pedido.py■■■■ ■■■■
user.py■■■■ routes/■■■■ ■■■■ __init__.py■■■■ ■■■■ auth.py■■■■ ■■■■ main.py■■■■ ■■■■
■■■■ clientes.py■■■■ ■■■■ productos.py■■■■ ■■■■ procesos.py■■■■ ■■■■ pedidos.py■■■■
■■■■ forms/■■■■ ■■■■ __init__.py■■■■ ■■■■ auth_forms.py■■■■ ■■■■ cliente_forms.py■■■■
■■■■ ■■■■ producto_forms.py■■■■ ■■■■ proceso_forms.py■■■■ ■■■■ pedido_forms.py■■■■ ■■■■
templates/■■■■ ■■■■ base.html■■■■ ■■■■ auth/■■■■ ■■■■ clientes/■■■■ ■■■■ productos/■■■■
■■■■ ■■■■ procesos/■■■■ ■■■■ pedidos/■■■■ ■■■■ static/■■■■ ■■■■ css/■■■■ ■■■■ js/■■■■ ■■■■
images/■■■■ ■■■■ services/■■■■ ■■■■ __init__.py■■■■ ■■■■ cliente_service.py■■■■ ■■■■
producto_service.py■■■■ ■■■■ proceso_service.py■■■■ ■■■■ pedido_service.py■■■■ ■■■■
utils/■■■■ ■■■■ __init__.py■■■■ ■■■■ decorators.py■■■■ ■■■■ helpers.py■■■■ ■■■■
validators.py■■■■ ■■■■ scripts/■■■■ ■■■■ init_db.py■■■■ ■■■■ backup_manager.py■■■■ ■■■■
dev_utils.py■■■■ ■■■■ config.py■■■■ ■■■■ run.py■■■■ ■■■■ requirements.txt■■■■ ■■■■ .env.template
```

10.2 Dependencias del Proyecto

```
Flask==2.3.3Flask-SQLAlchemy==3.0.5Flask-Login==0.6.3Flask-WTF==1.1.1Flask-Migrat
e==4.0.5WTFForms==3.0.1Werkzeug==2.3.7python-dotenv==1.0.0email-validator==2.0.0
```

****Fin de la Documentación Técnica**** *Sistema de Gestión ALS - Diego Camarena Gutiérrez - 15970795N*