

# Documentación del Sistema BusIA

## Descripción General

BusIA es un sistema de gestión de pasajes y empresas de buses desarrollado con Django. El sistema permite la gestión de usuarios, empresas de transporte, y la venta de pasajes.

## Estructura del Proyecto

### Aplicaciones Principales

- 1 **usuarios**: Gestión de usuarios y autenticación
- 2 **pasajes**: Sistema de venta y gestión de pasajes
- 3 **empresas**: Administración de empresas de transporte
- 4 **bus\_empresas**: Gestión de buses y relaciones con empresas
- 5 **landing**: Página principal y presentación del sistema

### Componentes Principales

#### 1. Sistema de Usuarios

- Registro de usuarios
- Inicio de sesión
- Gestión de perfiles

#### 2. Sistema de Pasajes

- Búsqueda de viajes
- Reserva de pasajes
- Gestión de tickets

#### 3. Sistema de Empresas

- Registro de empresas
- Gestión de rutas
- Administración de flota

## Tecnologías Utilizadas

- Django Framework
- SQLite (Base de datos)
- HTML/CSS/JavaScript (Frontend)
- Bootstrap (Framework CSS)

## Estructura de Archivos

```
busIA/  
... busIA/           # Configuración principal del proyecto  
... bus_empresas/    # Gestión de buses  
... empresas/        # Gestión de empresas
```

```
... landing/           # Página principal
... pasajes/           # Sistema de pasajes
... usuarios/          # Sistema de usuarios
... static/            # Archivos estáticos
... media/             # Archivos subidos por usuarios
... templates/         # Plantillas HTML
```

## Funcionalidades Detalladas

### Módulo de Usuarios

- Registro de nuevos usuarios
- Autenticación
- Gestión de perfiles
- Historial de viajes

### Módulo de Pasajes

- Búsqueda de rutas
- Reserva de asientos
- Gestión de pagos
- Emisión de tickets

### Módulo de Empresas

- Registro de empresas
- Gestión de rutas
- Administración de horarios
- Control de flota

## Guía de Instalación

- 1 Clonar el repositorio
- 2 Crear un entorno virtual
- 3 Instalar dependencias: `pip install -r requirements.txt`
- 4 Realizar migraciones: `python manage.py migrate`
- 5 Crear superusuario: `python manage.py createsuperuser`
- 6 Iniciar servidor: `python manage.py runserver`

## Mantenimiento y Soporte

### Base de Datos

- Respaldos periódicos recomendados
- Limpieza de registros antiguos
- Optimización de consultas

### Seguridad

- Autenticación de usuarios
- Protección de rutas
- Validación de formularios
- Encriptación de datos sensibles

## Contacto y Soporte

Para soporte técnico o consultas, contactar a:

- Correo: [correo de soporte]
- Teléfono:[número de soporte]

---

*Documentación generada el 29 de enero de 2025*