

1. Arquitectura de Software para la Aplicación Django

1.1. Componentes Principales

1. Frontend

- HTML/CSS: Estructura y estilo de las páginas.
- Bootstrap: Framework CSS para diseño responsivo y estilización.
- JavaScript: Para interactividad y manipulación dinámica de datos en el cliente.

2. Backend

- Django Framework: Framework web de Python para gestionar la lógica de negocio, modelos de datos, y controladores.
- Modelos: Definición de las estructuras de datos y lógica de base de datos.
- Vistas: Lógica para procesar solicitudes y renderizar respuestas.
- URLs: Mapeo de URLs a vistas correspondientes.

3. Base de Datos

- PostgreSQL: Sistema de gestión de bases de datos relacional utilizado para almacenar datos persistentes.

4. Autenticación y Autorización

- Django Auth: Sistema de autenticación y autorización de usuarios.
- Permisos: Control de acceso a diferentes partes de la aplicación según el rol del usuario.

5. Carga y Procesamiento de Datos

- Pandas: Librería de Python para manejar y procesar datos de archivos Excel.
- django-import-export: Librería para importar y exportar datos desde/para Excel.

6. Reporting y Visualización

- Django Templates: Para la generación de reportes y visualización de datos en formato HTML.

- Plotly/Chart.js: Librerías para crear gráficos y visualizaciones interactivas.

1.2. Estructura de Directorios y Archivos

```
UAESP/                # Directorio del proyecto
├── manage.py          # Script para gestionar el proyecto Django
├── UAESP/              # Configuración del proyecto
│   ├── __init__.py
│   ├── settings.py    # Configuración del proyecto
│   ├── urls.py        # Rutas del proyecto
│   └── wsgi.py        # Punto de entrada WSGI para despliegue
├── gestion_excel/    # Aplicación principal
│   ├── __init__.py
│   ├── admin.py      # Configuración del admin
│   ├── apps.py       # Configuración de la aplicación
│   ├── forms.py      # Formularios para datos de entrada
│   ├── models.py     # Modelos de datos
│   ├── tests.py      # Pruebas de la aplicación
│   ├── urls.py       # Rutas de la aplicación
│   └── views.py      # Vistas para manejar solicitudes
├── templates/        # Plantillas HTML
│   ├── home.html     # Página de inicio
│   ├── login.html    # Página de inicio de sesión
│   ├── upload.html   # Página para cargar archivos
│   └── data_table.html # Página para mostrar datos
```

```
| └─ report.html          # Página para generar reportes
| └─ static/             # Archivos estáticos (CSS, JS, imágenes)
|   └─ css/
|     └─ styles.css
| └─ requirements.txt    # Dependencias del proyecto
└─ db.sqlite3           # Base de datos (para desarrollo)
...

```

1.3. Flujo de Datos y Funcionalidad

1. Inicio de Sesión y Autenticación

- El usuario accede a la página de inicio (home.html).
- Desde allí, puede navegar a la página de login (login.html).
- Tras autenticarse, es redirigido a la página principal donde puede cargar archivos, ver datos y generar reportes.

2. Carga de Archivos

- El usuario accede a la página de carga (upload.html), sí y sólo sí está logueado.
- El archivo Excel es cargado y procesado usando Pandas y django-import-export.
- Los datos del archivo se almacenan en la base de datos a través de los modelos definidos en models.py.

3. Visualización de Datos

- Los datos cargados son visualizados en data_table.html.
- Se pueden aplicar filtros y editar registros según los permisos del usuario.

4. Generación de Reportes

- Los usuarios pueden generar reportes basados en los datos cargados.

- Los reportes se visualizan en report.html y pueden incluir gráficos generados con `Plotly` o `Chart.js`.

5. ****Autorización y Permisos****

- El acceso a páginas y funcionalidades está restringido según los roles de los usuarios (admin o usuario estándar).

- Solo los usuarios autenticados pueden acceder a ciertas páginas o realizar acciones específicas.