

Documentación Técnica del Sistema Compost-IoT

1. Selección del lenguaje de programación y sistema gestor de base de datos

Para el desarrollo del sistema Compost-IoT, se seleccionaron tecnologías ampliamente utilizadas y adecuadas para proyectos web modulares, escalables y de rápido desarrollo:

Lenguaje de programación: Python

Python es un lenguaje interpretado, multiplataforma y de alto nivel que permite una rápida prototipación. Cuenta con una amplia comunidad y un ecosistema rico en librerías para desarrollo web, análisis de datos y conectividad con hardware IoT (por ejemplo, con ESP32).

Framework web: Django

Django es un framework de alto nivel para el desarrollo web con Python. Proporciona seguridad, un sistema de autenticación robusto, ORM integrado y una estructura modular ideal para aplicaciones con múltiples funcionalidades como autenticación, manejo de usuarios y visualización de datos en tiempo real.

Sistema gestor de base de datos: SQLite

Para el entorno de desarrollo y pruebas iniciales, se eligió SQLite por ser una base de datos liviana, que no requiere instalación ni configuración adicional. Está totalmente integrada con Django y permite realizar migraciones y consultas fácilmente. Posteriormente, el sistema puede migrarse a PostgreSQL para entornos de producción o escenarios de mayor escala, dado que Django también tiene soporte completo para ello.

Posteriormente, el sistema puede migrarse a **PostgreSQL** para entornos de producción o escenarios de mayor escala, dado que Django también tiene soporte completo para ello.

Lenguajes de desarrollo web (Front-End): HTML, CSS y JavaScript

- **HTML5** se utiliza para definir la estructura semántica del contenido mostrado al usuario.
- **CSS3** permite el diseño visual de las interfaces, incluyendo la disposición de elementos, colores y estilos responsivos para distintos dispositivos.
- **JavaScript** se utiliza para proporcionar interactividad dinámica en la interfaz del sistema, como la carga de imágenes, validación de formularios y mejora de la experiencia de usuario.

Estas tecnologías complementan el backend en Django para brindar una solución web completa, moderna y funcional tanto en navegadores como en dispositivos móviles o de escritorio.

2. Codificación de los procesos identificados en el sistema propuesto

El sistema Compost-IoT fue desarrollado dividiendo sus funciones en partes más pequeñas y comprensibles, lo que facilita su organización y mantenimiento. Se utilizaron herramientas que permiten que el sistema sea claro, ordenado y fácil de actualizar, especialmente gracias al uso del framework Django.

Los principales procesos que se programaron en el sistema fueron los siguientes:

Autenticación de usuarios

- Permite que los usuarios se registren, inicien sesión y cierren sesión de manera segura.
- Se crearon pantallas y formularios que interactúan con el sistema de control de acceso de Django, el cual se encarga de verificar la identidad del usuario.

Gestión de unidades de compostaje

- Permite a los usuarios crear, ver y eliminar sus propios contenedores de compostaje.
- Se diseñaron formularios fáciles de usar, que solo están disponibles para personas que hayan iniciado sesión. Cada unidad queda registrada con el nombre del usuario dueño.

Captura de datos de sensores

- Guarda automáticamente las lecturas de sensores (por ahora simuladas) sobre temperatura, humedad, pH y oxígeno.
- Se crearon estructuras en la base de datos que almacenan esta información. También se agregó una función que genera datos de ejemplo para hacer pruebas sin sensores reales.

Visualización de estadísticas

- Muestra gráficamente las mediciones promedio de cada unidad de compostaje, facilitando el análisis por parte del usuario.
- Se tomaron los datos almacenados y se procesaron para generar gráficos interactivos que se muestran en la página web.

Panel de control (Menú)

- Ofrece una vista general del sistema donde el usuario puede acceder rápidamente a estadísticas, unidades registradas y datos de prueba.
- Se diseñó una pantalla central que organiza de forma clara toda la información del usuario, accesible solo después de iniciar sesión.

Todos estos componentes fueron desarrollados paso a paso, aplicando buenas prácticas como guardar versiones del código, separar la parte visual de la lógica del sistema, y hacer pruebas antes de usarlo en un entorno real.

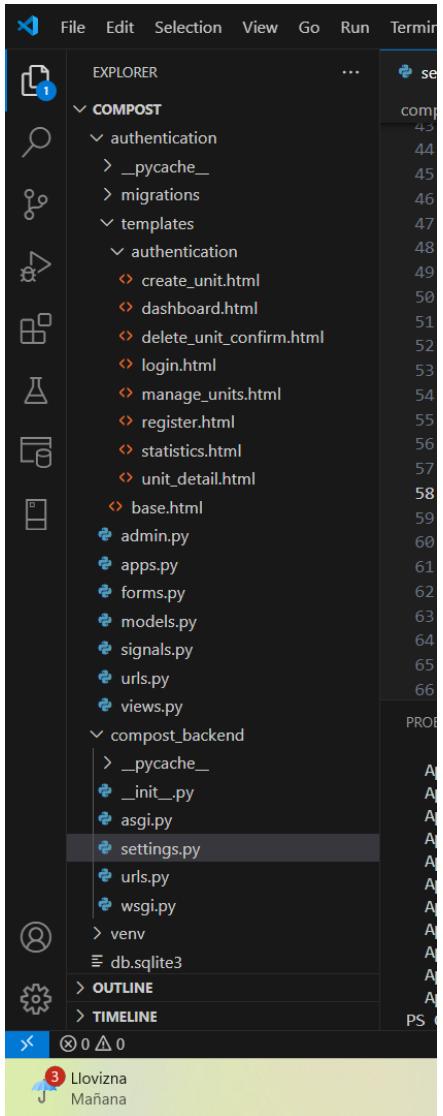
3. Elaboración de la documentación técnica de la construcción del sistema

La documentación técnica del sistema Compost-IoT ha sido elaborada para facilitar su comprensión, mantenimiento y escalabilidad. Esta documentación cubre la estructura del sistema, los componentes desarrollados, instrucciones de instalación y uso, así como la arquitectura general.

Estructura del proyecto

El sistema se encuentra estructurado según las convenciones de Django. La jerarquía de archivos incluye:

```
compost/
    ├── compost_backend/          # Proyecto Django principal
    |   ├── settings.py          # Configuración global del sistema
    |   ├── urls.py              # Enrutamiento principal
    |   └── wsgi.py               # Interfaz para servidores web
    ├── authentication/          # Aplicación encargada de login, registro,
    |   └── dashboard             y gestión
    |       ├── models.py         # Modelos de Usuario, Unidades y Lecturas
    |       ├── views.py          # Lógica de negocio (controladores)
    |       ├── urls.py          # Enrutamiento interno
    |       └── templates/        # Plantillas HTML
    |           └── static/       # Archivos CSS, JS e imágenes
    ├── db.sqlite3                # Base de datos por defecto (SQLite)
    └── manage.py                 # Herramienta de administración del proyecto
```



Componentes principales

Modelos: UserProfile, CompostUnit, SensorReading

Vistas: login, registro, dashboard, estadísticas, CRUD de unidades, generación de datos de pruebas

Plantillas HTML con etiquetas de Django

Archivos estáticos: CSS, JS, imágenes

Requisitos técnicos:

Lenguaje: Python 3.13

Framework: Django 5.2.1

Base de datos: SQLite

Frontend: HTML5, CSS3, JavaScript (Chart.js)

Entorno virtual: venv

Instalación y configuración

Se incluye una guía de instalación con los pasos para configurar el entorno virtual, instalar dependencias y ejecutar el servidor local.