

Tarea 4.1: Desarrollo y Despliegue de una Aplicación Web con Flask

RA5: Genera documentos web utilizando lenguajes de guiones de servidor. *Criterios de evaluación: a, c, d, e, f, g, h*

RA6: Genera documentos web con acceso a bases de datos utilizando lenguajes de guiones de servidor. *Criterios de evaluación: a, b, c, d, e, f, g*

Descripción

En esta tarea, trabajaréis en grupos de **4 personas** para desarrollar y desplegar una aplicación web utilizando **Flask** como framework de desarrollo. Cada grupo desarrollará una aplicación con una funcionalidad específica, que será asignada por el profesor.

El proyecto debe incluir:

Desarrollo de la aplicación en **Python** utilizando **Flask**.

Conexión con una base de datos **MySQL** utilizando **PyMySQL**.

Implementación de un sistema de **login de usuarios**, almacenando las contraseñas en formato **hash**.

Uso de **Trello** para la planificación y organización del trabajo en equipo.

Publicación del código fuente en **GitHub**.

Despliegue en producción de dos formas:

1. **Contenedor Docker**, que deberá publicarse en **Docker Hub**.
2. **Plataforma PythonAnywhere**.

Requisitos técnicos

1. Desarrollo en Python y Flask

- La aplicación debe seguir una estructura ordenada de proyecto Flask.
- Debe incluir rutas, controladores y plantillas en **HTML + Jinja2**.
- Implementación de un sistema de autenticación con **registro y login**.
- Uso de **werkzeug.security** para **hashear y verificar contraseñas**.

2. Base de datos MySQL

- Debe contener al menos:
 - Una tabla para la gestión de **usuarios** con los siguientes campos:
 - id (clave primaria, autoincremental).
 - username (nombre de usuario, único).
 - email (correo electrónico, único).
 - password_hash (contraseña almacenada como hash).
 - Otras tablas relacionadas con los datos de la aplicación asignada.
- Uso del conector **PyMySQL** para interactuar con la base de datos.
- Archivo de configuración (config.py) para gestionar las credenciales de la base de datos.

3. Gestión del proyecto en GitHub y Trello

- Cada grupo debe crear un **repositorio en GitHub** para su proyecto.
- Uso de **Trello** para organizar el trabajo, con listas y tarjetas asignadas a los miembros del equipo.

4. Despliegue de la aplicación

- **Docker:**
 - Crear un **Dockerfile** para contenerizar la aplicación.
 - Publicar la imagen en **Docker Hub**.
- **PythonAnywhere:**
 - Configurar el entorno en la plataforma.
 - Subir y ejecutar la aplicación.

5. Documentación del proyecto

- Un archivo README.md con:
 - Explicación del proyecto y su funcionalidad.
 - Instrucciones de instalación y ejecución.
 - Pasos para desplegar en **Docker** y **PythonAnywhere**.

Uso de Trello:

1. Definir responsabilidades desde el inicio

- Cada grupo debe crear una lista en Trello llamada **"Roles y Responsabilidades"** donde indiquen:
 - Quién se encarga del desarrollo en Flask.
 - Quién se encarga de la base de datos y la integración con MySQL.
 - Quién se encarga del login y la seguridad.
 - Quién se encarga del despliegue en **Docker** y **PythonAnywhere**.
 - Quién se encarga de la documentación (README.md de **GitHub**).

2. Crear un tablero bien estructurado

Cada equipo debe estructurar su Trello con listas como:

- **Tareas por hacer.**
- **En proceso.**
- **En revisión / Testing.**
- **Completadas.**

Cada tarea en Trello debe tener:

- **Descripción clara** de lo que hay que hacer.
- **Etiqueta con el área de trabajo** (Ej. "Backend", "Base de datos", "Autenticación", "Despliegue").

-
- **Fecha límite** para evitar retrasos.
 - **Checklist de subtareas**, si es algo complejo.
 - **Asignado a una persona específica** (para que cada uno tenga claro su rol).
-

3. Usar comentarios y actualizaciones

Cada alumno debe:

Escribir en los comentarios de su tarjeta cómo avanza.

Adjuntar capturas de pantalla o enlaces (ej. commit en GitHub) para demostrar progreso.

Avisar si necesita ayuda o tiene bloqueos.

Entrega de la tarea (plazo máximo, viernes 7 de marzo):

Cada grupo deberá entregar en el espacio habilitado al efecto en Moodle Centros:

- Enlace al código fuente en un **repositorio de GitHub**. Cualquier documentación irá en el archivo **README.md**.

- Enlace al tablero de **Trello** con la planificación del equipo (desde el inicio de la tarea).

- Enlace a la imagen de **Docker Hub**.

- Enlace a la aplicación desplegada en **PythonAnywhere**.