

Pre proyecto

1. Resumen:

El proyecto consiste en desarrollar una aplicación web basada en Flask con MongoDB que muestra información de Pokémons obtenida de una API externa. Los usuarios podrán ver los detalles de los Pokémons, agregarlos a sus favoritos y armar su propio equipo. Se utilizará Docker Compose para orquestar los diferentes servicios y asegurar la fácil configuración y despliegue del proyecto.

2. Planteamiento del problema y justificación:

En el mundo actual, los Pokémons son una franquicia popular con una gran cantidad de seguidores. Sin embargo, puede resultar difícil acceder y organizar información detallada sobre los diferentes Pokémons. Esta aplicación web busca resolver ese problema al proporcionar a los usuarios una plataforma donde puedan explorar y administrar información de Pokémons de manera intuitiva y sencilla.

3. Aplicaciones en el marco tecnológico actual:

En el marco tecnológico actual, las aplicaciones web son ampliamente utilizadas para brindar servicios y facilitar la interacción con los usuarios. El uso de Flask como framework web y MongoDB como base de datos ofrece una combinación poderosa y flexible para desarrollar aplicaciones escalables y de alto rendimiento. Además, Docker y Docker Compose permiten la contenerización y orquestación de los diferentes componentes del proyecto, simplificando la configuración y el despliegue en diferentes entornos.

4. Objetivo:

El objetivo principal del proyecto es desarrollar una aplicación web que permita a los usuarios acceder y administrar información detallada de Pokémons. Algunos objetivos específicos incluyen:

- Obtener datos de una API externa y almacenarlos en una base de datos MongoDB.
- Mostrar una lista de Pokémons con detalles como nombre, tipo y estadísticas.
- Permitir a los usuarios ver información detallada de cada Pokémon.
- Permitir a los usuarios agregar Pokémons a su lista de favoritos.
- Permitir a los usuarios armar su equipo de Pokémons.

Proyecto:

Fases:

a. Diseño y prototipo: En esta fase se realizará el diseño de la arquitectura de la aplicación, incluyendo la estructura de la base de datos, el diseño de las vistas y rutas en Flask, y el diseño de la interfaz de usuario. También se creará un prototipo funcional para validar el concepto y la usabilidad.

b. Implementación: En esta fase se llevará a cabo la implementación de la aplicación web utilizando Flask y PyMongo para conectarse a la base de datos MongoDB. Se desarrollará el script en Python para obtener datos de la API externa y almacenarlos en la base de datos. También se configurará Docker Compose para orquestar los diferentes servicios y garantizar un despliegue sencillo y eficiente.

Proyecto

5. Fases

Diseño-Prototipo

- a. Identificación de requerimientos: Determinar los requerimientos funcionales y no funcionales del proyecto. Esto incluye la recopilación de los requisitos de la API, la definición de las funcionalidades del sitio web y los requerimientos de infraestructura.
- b. Diseño de la base de datos: Seleccionar la estructura de la base de datos para almacenar los datos de la API. Esto también puede incluir la definición de esquemas para los datos.
- c. Diseño de la arquitectura de la aplicación: Determinar cómo se construirá la aplicación y cómo se organizan los diferentes componentes, como el backend y el frontend. Esto también puede incluir la definición de la estructura de directorios y archivos.
- d. Diseño del prototipo de la interfaz de usuario: Crear un prototipo de la interfaz de usuario utilizando herramientas como Figma o Sketch. Esto permitirá visualizar cómo se verá la página web y cómo los usuarios interactúan con ella.

Implementación

- a. Configuración del entorno de desarrollo: Instalar todas las herramientas necesarias como Python, Flask, MongoDB, Docker y Docker Compose.
- b. Desarrollo del backend: Crear el backend de la aplicación utilizando Flask y MongoDB. Esto incluirá la configuración de las rutas, la creación de las funciones para obtener datos de la API, y la configuración de la conexión con la base de datos.

- c. Desarrollo del frontend: Crear el frontend de la aplicación utilizando Bootstrap 5 y HTML/CSS. Esto incluirá la creación de las páginas web, la configuración de los estilos, y la conexión con el backend para obtener los datos de la API.
- d. Pruebas y depuración: Probar la aplicación para asegurarse de que funciona correctamente y corregir cualquier error encontrado.
- e. Despliegue: Desplegar la aplicación en un entorno de producción utilizando Docker Compose.