

700 - laboratorios integradores

April 27, 2023

1 Laboratorios integradores

1.1 Escenario 1: app de delivery

1.1.1 Descripción del proyecto

El equipo de desarrollo ha sido contratado por una empresa de delivery de comida para desarrollar un sistema de seguimiento de la ubicación de sus repartidores. El sistema debe permitir a los clientes de la empresa rastrear la ubicación de su pedido y estimar el tiempo de llegada. El sistema debe estar basado en una API RESTful que permita a los clientes acceder a la información del pedido en tiempo real.

El equipo de desarrollo debe implementar el sistema utilizando Python y Flask, y se recomienda el uso de SQLite como base de datos para almacenar la información del pedido y la ubicación de los repartidores. También se debe usar Docker para crear un contenedor para el proyecto y Gunicorn para el servidor de producción.

1.1.2 Tareas

1. Investigación de Docker Compose Antes de comenzar con el desarrollo del proyecto, el equipo de desarrollo debe investigar cómo utilizar Docker Compose para administrar múltiples contenedores de Docker. Deben investigar cómo configurar un archivo de docker-compose.yml para que pueda ejecutar el proyecto de Flask, la base de datos SQLite y cualquier otro servicio necesario para el proyecto.
2. Diseño de la base de datos El equipo de desarrollo debe diseñar la estructura de la base de datos que se utilizará para almacenar la información del pedido y la ubicación de los repartidores. Se recomienda el uso de SQLite como base de datos para el proyecto debido a su simplicidad y facilidad de uso.
3. Implementación de la API RESTful El equipo de desarrollo debe implementar la API RESTful utilizando Flask. La API debe tener las siguientes rutas:

/pedidos: Devuelve una lista de todos los pedidos en la base de datos. /pedidos/: Devuelve los detalles de un pedido específico. /repartidores: Devuelve una lista de todos los repartidores en la base de datos. /repartidores/: Devuelve los detalles de un repartidor específico. /ubicacion/: Devuelve la ubicación actual de un repartidor específico.

4. Implementación del sistema de seguimiento de ubicación El equipo de desarrollo debe implementar un sistema que permita a los repartidores actualizar su ubicación en tiempo real y almacenarla en la base de datos. Se recomienda el uso de generadores para simular el movimiento de los repartidores en lugar de usar datos GPS reales.

5. Pruebas y depuración El equipo de desarrollo debe realizar pruebas exhaustivas del sistema para garantizar que funcione correctamente y que la API sea accesible para los clientes. Deben solucionar todos los errores y problemas identificados durante las pruebas.
6. Despliegue Finalmente, el equipo de desarrollo debe preparar el proyecto para su implementación en producción. Deben crear un contenedor Docker para el proyecto y usar Gunicorn para servir la aplicación Flask en un entorno de producción.

1.1.3 Entregables

El equipo de desarrollo debe entregar los siguientes elementos al finalizar el proyecto:

1. Código fuente completo del proyecto.
2. Archivo de docker-compose.yml configurado para ejecutar el proyecto.
3. Documentación detallada del proyecto que incluya una descripción de la base de datos, la estructura de la API, las pruebas realizadas y los problemas identificados y solucionados.
4. Archivo README que incluya instrucciones necesarias para ejecutar el proyecto y cualquier otra información relevante para su uso y mantenimiento.

1.2 Escenario 2: Red social

1.2.1 Descripción del proyecto

El equipo de desarrollo ha sido contratado por una nueva red social para implementar un sistema de registro de usuarios y publicación de contenido. El sistema debe permitir a los usuarios crear una cuenta, iniciar sesión y publicar mensajes en su perfil. Los usuarios también deben poder seguir a otros usuarios y ver sus publicaciones en una línea de tiempo. El sistema debe estar basado en una API RESTful que permita a los clientes acceder a la información del usuario y de sus publicaciones.

El equipo de desarrollo debe implementar el sistema utilizando Python y Flask, y se recomienda el uso de PostgreSQL como base de datos para almacenar la información del usuario y de las publicaciones. También se debe usar Docker para crear un contenedor para el proyecto y Gunicorn para el servidor de producción.

1.2.2 Tareas

1. Investigación de Docker Compose Antes de comenzar con el desarrollo del proyecto, el equipo de desarrollo debe investigar cómo utilizar Docker Compose para administrar múltiples contenedores de Docker. Deben investigar cómo configurar un archivo de docker-compose.yml para que pueda ejecutar el proyecto de Flask, la base de datos PostgreSQL y cualquier otro servicio necesario para el proyecto.
2. Diseño de la base de datos El equipo de desarrollo debe diseñar la estructura de la base de datos que se utilizará para almacenar la información del usuario y de las publicaciones. Se recomienda el uso de PostgreSQL como base de datos para el proyecto debido a su escalabilidad y capacidad de manejar grandes volúmenes de datos.
3. Implementación de la API RESTful El equipo de desarrollo debe implementar la API RESTful utilizando Flask. La API debe tener las siguientes rutas:

/usuarios: Devuelve una lista de todos los usuarios registrados en la base de datos. /usuarios/: Devuelve los detalles de un usuario específico. /publicaciones: Devuelve una lista de todas las

publicaciones en la base de datos. `/publicaciones/`: Devuelve los detalles de una publicación específica. `/timeline/`: Devuelve la línea de tiempo de un usuario específico, que incluye sus propias publicaciones y las de los usuarios que sigue.

4. Implementación del sistema de registro y autenticación El equipo de desarrollo debe implementar un sistema que permita a los usuarios crear una cuenta, iniciar sesión y cerrar sesión. Las contraseñas deben estar cifradas y almacenadas de forma segura en la base de datos.
5. Implementación del sistema de publicación de contenido El equipo de desarrollo debe implementar un sistema que permita a los usuarios crear y publicar contenido en su perfil. Los usuarios deben poder editar y eliminar sus propias publicaciones.
6. Implementación del sistema de seguimiento de usuarios El equipo de desarrollo debe implementar un sistema que permita a los usuarios seguir a otros usuarios y ver sus publicaciones en una línea de tiempo. Los usuarios deben poder dejar de seguir a otros usuarios en cualquier momento.

1.2.3 Pruebas y depuración

El equipo de desarrollo debe realizar pruebas exhaustivas del sistema para garantizar que funcione correctamente y que la API sea accesible para los clientes. Deben solucionar todos los errores y problemas identificados durante las pruebas.

1.2.4 Despliegue

Finalmente, el equipo de desarrollo debe preparar el proyecto para su implementación en producción. Deben crear un contenedor Docker para el proyecto y usar Gunicorn para servir la aplicación Flask en un entorno de producción.

1.2.5 Entregables

El equipo de desarrollo debe entregar los siguientes elementos al finalizar el proyecto:

1. Código fuente completo del proyecto.
2. Archivo de `docker-compose.yml` configurado para ejecutar el proyecto.
3. Documentación detallada del proyecto que incluya una descripción de la base de datos, la estructura de la API, las pruebas realizadas y los problemas identificados y solucionados.
4. Archivo README que incluya instrucciones necesarias para ejecutar el proyecto y cualquier otra información relevante para su uso y mantenimiento.