EASYCAB

Práctica 1

Adrián Muñoz Girona – 50592313H
David Pérez Sampedro – 48774775D
Curso 2024-2025

1. Informe de Componentes de Software

1. EC_Central:

EC_Central es el componente principal que gestiona la lógica de control de todo el sistema de taxis autónomos. Este módulo se conecta con los taxis y recibe solicitudes de clientes para asignarles un servicio.

Su objetivo es leer el mapa de la ciudad desde un archivo de configuración y gestionar la base de datos de taxis disponibles.

Cada vez que recibe una solicitud de un cliente (a través de EC_Customer), verifica si hay u taxi disponible para asignarlo a la solicitud. Si el servicio puede ser proporcionado envía una respuesta de aceptación (“OK”) al cliente y envía instrucciones al taxi correspondiente para que recoja al cliente. Si no hay taxis disponibles, responde con una denegación (“KO”).

A lo largo del recorrido de cada taxi, EC_Central recibe actualizaciones constantes sobre la posición del taxi y actualiza el estado del mapa de la ciudad.

EC_Central también tiene la capacidad de tomar decisiones en caso de incidentes (como cambiar el destino del taxi, hacer que regrese a la base, detener el taxi, o reanudar su movimiento).

Interacción

EC_Central se comunica con EC_DE para controlar los taxis y con EC_Customer para recibir solicitudes de clientes. También mantiene el estado del mapa y distribuye su situación actualizada a cada componente relevante.

2. EC_DE (Digital Engine – Motor Digital del Taxi)

EC_DE es el cerebro de cada taxi. Al iniciar, se autentica con EC_Central para confirmar que está listo para recibir servicios.

Cuando se le asigna un servicio, EC_DE recibe el destino y comienza a mover el taxi hacia esa ubicación en el mapa, moviéndose paso a paso y enviando su posición actualizada a EC_Central.

EC_DE responde a incidentes detectados por los sensores (EC_S). Si los sensores envían una señal de incidencia (“KO”), EC_DE detiene al taxi hasta recibir una señal de reanudación (“OK”).

Cambia el estado del taxi a “END” cuando llega a su destino, y espera nuevas órdenes para continuar o regresar a la base.

Interacción

EC_CE se comunica constantemente con EC_Central para recibir instrucciones y enviar actualizaciones de su posición. También se conecta a EC_S para recibir el estado de los sensores que monitorean su recorrido.

3. EC_S (Sensores del Taxi)

EC_S simula los sensores a bordo del taxi que informan sobre cualquier obstáculo o incidencia en el camino.

Envía una señal de estado cada segundo al EC_DE correspondiente: “OK” si el camino está despejado o “KO” si se detecta algún obstáculo o problema.

La aplicación permite al usuario provocar manualmente una incidencia en tiempo de ejecución pulsando una tecla, lo que envía una señal “KO” para detener el taxi hasta que se reanude el estado “OK”.

Interacción

EC_S se conecta directamente con EC_DE para enviarle los datos de estado en tiempo real, ayudando al taxi a tomar decisiones de seguridad en su trayecto.

4. EC_Customer (Aplicación del Cliente)

EC_Customer es la aplicación usada por los clientes para solicitar un servicio de taxi. Lee de un archivo una lista de solicitudes de servicio, con el destino de cada viaje.

Al iniciar, envía su primera solicitud a EC_Central y espera una respuesta: “OK” si el servicio fue aceptado y se le ha asignado un taxi, o “KO” si no hay taxis disponibles en ese momento.

Una vez que el servicio concluye (éxito o fracaso), EC_Customer espera 4 segundos y pasa a solicitar el siguiente servicio de la lista.

Interacción

EC_Customer se conecta a EC_Central para enviar sus solicitudes de servicio y recibir actualizaciones sobre la aceptación o rechazo del servicio.

2. Guía de Despliegue de la Aplicación EasyCab

Requisitos previos.

Antes de iniciar el despliegue, hay que tener:
1. Mínimo 3 PCs o máquinas virtuales (en local o en la nube).

2. Instalación de Docker (opcional) para ejecutar los componentes en contenedores si deseas un despliegue aislado.

3. Dependencias, en nuestro caso:

- Python
- Kafka
- SQLite3

4. Configuración de red. Los PCs deben poder comunicarse entre sí a través de los puertos especificados en la configuración.

Paso 1. Configuración de EC_Central (Central de Control).

1. Configurar la Base de Datos:

- Crear una BD de datos SQLite para almacenar el estado de los taxis.
- Insertar datos iniciales de los taxis, con sus IDs y estados de inicio. Esto puede hacerse mediante un script SQL o desde el propio código.

2. Configurar el Mapa de la Ciudad:

- Preparar el archivo de configuración del mapa en el formato especificado.

```
<ID_LOCALIZACION><COORDENADA_X><COORDENADA_Y>
```

- Guarda este archivo en el directorio donde EC_Central pueda acceder a él.

3. Iniciar EC_Central

- En el PC o contenedor designado para EC_Central, ejecutar el programa con los parámetros necesarios.

```
python3 EC_Central <PUERTO_ESCUCHA><IP_BROKER><PUERTO_BROKER>
```

- Asegurarse de que EC_Central queda a la espera de solicitudes en el puerto especificado.

Paso 2. Configuración de EC_DE(Motor Digital de cada Taxi).

1. Iniciar EC_DE:

- En los PCs designados para los taxis, ejecutaR EC_DE para cada taxi con los siguientes parámetros:

```
python3 EC_DE.py <IP_Central><PUERTO_CENTRAL><IP_BROKER>
<PUERTO_BROKER><IP_EC_S><PUERTO_EC_S><ID_TAXI>
```

- Cada taxi debe conectarse a EC_Central y autenticarse con su ID para ser reconocido en el sistema.

2. Verifica la Conexión:

- Confirmar que EC_DE está conectado correctamente con EC_Central y listo para recibir solicitudes de servicio.

Paso 3. Configuración de EC_S (Sensores de cada Taxi).

1. Inicia EC_S:

- Ejecutar el programa EC_S en el mismo PC o en un dispositivo separado según la configuración deseada:

```
python3 EC_S.py <IP_EC_DE><PUERTO_EC_DE>
```

- Este componente comenzará a enviar señales de estado (OK/KO) a EC_DE cada segundo.

2. Prueba de Funcionamiento de los Sensores:

- Activa manualmente una incidencia (tecla ENTER) para verificar que EC_DE responde correctamente al recibir un mensaje KO desde EC_S.

Paso 4. Configuración de EC_Customer (Aplicación de Cliente).

1. Cargar del Archivo de Solicitudes de Servicio:

- Preparar un archivo de solicitudes de servicio en el formato especificado.
  (<ID_LOCALIZACION>)
- Guardar este archivo en el directorio de EC_Customer

2. Iniciar EC_Customer:

- Ejecutar la aplicación cliente en el PC designado para EC_Customer:

```
python3 EC_Customer.py <IP_BROKER><PUERTO_BROKER>
<ID_CLIENTE>
```

- La aplicación cliente comenzará a leer del archivo y enviará una solicitud de servicio a EC_Central.

Paso 5. Pruebas de Conectividad y Funcionalidad

1. Prueba de Solicitudes y Asignación de Taxis:

- Desde EC_Customer, realizar una solicitud de servicio y verifica que EC_Central asigna un taxi disponible y envía la confirmación.

2. Simulación de Movimiento:

- Observar el movimiento del taxi en el mapa y confirmar que EC_Central está actualizando correctamente la posición en tiempo real.

3. Prueba de Incidencias:

- Activa manualmente una incidencia desde EC_S y confirma que EC_DE detiene el taxi, informando a EC_Central sobre la situación.

4. Validación de Comandos Manuales en EC_Central:

- Usar los comandos STOP, RESUME, DESTINATION, RETURN desde EC_Central y verificar que los taxis respondan de acuerdo con cada comando.

CAPTURAS DE PANTALLA