

PROYECTO PYTHON: TAXI

Planteamiento

El servicio de taxis está preocupado por el auge de las empresas privadas del sector del transporte, que les quita clientela porque son más baratos y por lo general más cómodos de usar.

Han decidido trabajar en una solución, después de un análisis han determinado que esto se debe al atraso tecnológico que acarrean, se desea crear un prototipo de taxímetro digital, que pueda implementarse en toda la red de taxis. El objetivo es modernizar el sistema actual de facturación y crear un sistema que calcule las tarifas que se deben cobrar a sus clientes. Para ello han decidido confiarnos a nosotros la tarea.

La primera fase del desarrollo consistirá en un programa en Python. Ese programa debe poder ejecutarse, y al hacerlo el programa se quedará a la espera de que se le indique por input que la carrera ha comenzado. En ese momento, el programa empezará a calcular la tarifa, en función de si el taxi está en movimiento o no. Mientras el taxi esté parado, al cliente se le cobrarán 2 céntimos por segundo de carrera. En el momento en el que el taxi arranque, mientras esté en movimiento, se cobrarán 5 céntimos por segundo. Cuando se le indique al programa que la carrera ha terminado, el programa calculará el total (en Euros).

Plazos

Se os dan dos semanas para realizar la entrega y presentar la solución.

Condiciones de entrega

Para el día de la entrega, será necesario presentar:

- El repositorio en github en el que habéis trabajado
- Una demo donde presentéis el CLI que habéis desarrollado
- Una presentación para un público no técnico.
- Una presentación del código que habéis desarrollado, explicando sus puntos fuertes y sus puntos débiles, bajo vuestro punto de vista
- El enlace al tablero Kanban usado para organizaros

Tecnologías que se pueden aplicar

- Git
- Github
- Python
- Trello / Jira ...

Niveles de Entrega

Nivel Esencial:

- ☐ Un programa CLI que al iniciarse dé la bienvenida, explique su funcionamiento y se quede a la espera de instrucciones
- ☐ Cuando se le dé la orden de empezar, el programa debe empezar a calcular la tarifa contando con que el taxi empezará siempre parado
- ☐ Cuando se le indique que el taxi se pone en marcha, debe empezar a aplicar el precio del taxi en movimiento
- ☐ Se le debe poder indicar al programa que el taxi se queda quieto y que vuelve a arrancar sin acabar la carrera (el taxi puede parar en un semáforo, en un atasco, en un stop, etc)
- ☐ Al indicarle al programa que la carrera ha acabado, debe devolver el precio total en euros
- ☐ Al acabar la carrera y devolver el total, el programa debe quedarse a la espera de que se le indique que empieza una nueva carrera sin necesidad de cerrarlo y ejecutarlo de nuevo

Nivel Medio:

- ☐ El programa incorpora un sistema de logs para la trazabilidad del código.
- ☐ El programa incorpora tests unitarios.
- ☐ El programa guarda en texto plano un registro histórico de carreras pasadas que se puede consultar.
- ☐ Los precios de la carrera se pueden configurar para adaptarse a la demanda actual.

Nivel Avanzado:

- ☐ Se ha programado teniendo en cuenta un enfoque orientado a objetos (OOP)
- ☐ El programa está asegurado con contraseñas
- ☐ El programa tiene un frontend o algún otro tipo de interfaz más amigable

Nivel Experto:

- ☐ El programa guarda en una base de datos los registros de carreras pasadas
- ☐ El programa está dockerizado
- ☐ Desplegar la aplicación para que sea accesible a través de una web