Materia: Arquitecturas Web (TUDAI)

Díaz Pace

FACULTAD DE CIENCIAS EXACTAS UNICEN

Trabajo Práctico Especial: Aplicación para Monopatines Eléctricos

Descripción del problema/sistema

Una empresa quiere lanzar un negocio para permitir el alquiler de monopatines electrónicos en distintas paradas de una ciudad capital, para lo cual requiere el desarrollo de una aplicación móvil para los usuarios del servicio, y adicionalmente una aplicación Web para la gestión correspondiente. El servicio consiste en contar con una flota de monopatines eléctricos, inicialmente estacionados en diferentes paradas previamente definidas, dentro del centro de la ciudad y zonas cercanas. Los monopatines se buscan y se dejan en dichas paradas, esto es una condición básica para el funcionamiento del servicio. A continuación, se describe el funcionamiento del servicio.

Para poder utilizar el monopatín el usuario deberá crearse una cuenta en la app, asociada a una cuenta de Mercado Pago. Previamente al uso del servicio, debe haber cargado en su cuenta un monto de dinero, que se irá descontando en función del tiempo de uso del monopatín. Se puede utilizar la misma cuenta de Mercado Pago para varias cuentas del servicio.
Una cuenta podrá tener asociados varios usuarios que utilizarán los créditos cargados en la cuenta, y un usuario puede asociarse a más de una cuenta. Cada usuario tendrá un nombre y debe registrar su número de celular, email válido, nombre y apellido. La cuenta tendrá un número identificatorio y una fecha de alta.
Una vez que la cuenta tenga cargado dinero, el usuario podrá activar un monopatín para su uso, mediante un lector de código QR. En ese momento se generará un viaje asociado a la cuenta del usuario que está utilizando su app, registrando fecha y hora de inicio. El uso del monopatín es por tiempo, comienza a consumirse el crédito cuando se activa el monopatín, y esto permitirá que se encienda en ese momento. A partir de allí el usuario del servicio podrá utilizar el monopatín, y una vez que no lo requiera más deberá dejarlo en una parada previamente establecida. En este momento selecciona la opción para cortar el servicio, una vez estacionado el monopatín, finalizando el viaje. Al finalizar el viaje se va a registrar la fecha y hora de finalización y los kilómetros recorridos. Cabe aclarar que la app no debe permitir finalizar un viaje sino detecta mediante el GPS con el que cuenta el monopatín, que se encuentra en una parada permitida.
Si el usuario requirió detenerse durante no más de 15 minutos en algún punto intermedio del viaje, la app contará con una opción ***Pausar***. Con esta opción se registra una pausa asociada al viaje, para establecer que el monopatín no está siendo utilizado, aunque aún corre el gasto de créditos. De esta manera se puede establecer el uso real del monopatín, y la app permite apagarlo, pero no lo desasigna a la cuenta actual. Una vez finalizada la pausa el usuario lo puede indicar por la app, para poder encender

el monopatín. Si pasaran los 15 min automáticamente se volverá a considerar en uso el monopatín, y se comienza a cobrar un monto mayor de crédito hasta el final del viaje.
El monopatín contará con un GPS y está identificado unívocamente con ID, por lo que en todo momento se puede determinar dónde está cada monopatín. De esta manera, si un usuario necesita un monopatín podrá encontrar el más cercano a través de un mapa interactivo en la app que muestra los monopatines en la zona.
Para el mantenimiento de los monopatines se va a crear con una aplicación Web para registrar todas las acciones de mantenimiento que realizan los Encargados de Mantenimiento de monopatines. Para establecer si un monopatín requiere de mantenimiento se considera el tiempo de uso y los kilómetros recorridos, para lo que se requiere del registro de esta información, así como la generación de reportes asociados al uso de monopatines, respecto a kilómetros y tiempo de uso, incluyendo tiempo con pausas y sin pausas. Es necesario saber si un monopatín está en mantenimiento, o se encuentra habilitado para su uso, al estar en una parada definida luego de finalizado un mantenimiento.
Además, el Administrador de Monopatines es quien gestiona los monopatines y las paradas en la aplicación (por ej., agregando, quitando, actualizando datos según sea requerido), también establece los precios de tarifa normal y extras por reinicio de pausas extensas. Por otro lado, es capaz de anular cuentas cuando por algún motivo que se considere necesario.

Un primer análisis del problema reveló las siguientes funcionalidades (si bien podrían existir otras funcionalidades no detectadas, o variaciones de las planteadas):

- Registrar monopatín en mantenimiento (debe marcarse como no disponible para su uso)
- Registrar fin de mantenimiento de monopatín
- Ubicar monopatín en parada (opcional)
- Agregar monopatín
- Quitar monopatín
- Registrar parada
- Quitar parada
- Definir precio
- Definir tarifa extra para reinicio por pausa extensa
- Anular cuenta
- Generar reporte de uso de monopatines por kilómetros
- Generar reporte de uso de monopatines por tiempo con pausas
- Generar reporte de uso de monopatines por tiempo sin pausas

Todo esto lo hace el admin

El trabajo implica construir un backend de servicios REST para el problema con una arquitectura de microservicios.

1ra ENTREGA

1. Realizar un modelamiento de los distintos datos que debe guardar el sistema. Este modelamiento debe ser capturado en términos de (sub-dominios), apuntando a un diseño/implementación con microservicios. Una vez validado el modelo, generas las entidades y relaciones correspondientes, y

mapearlo a una base de datos SQL. En ciertos casos (es decir, para ciertos microservicios), puede decidirse utilizar otro tipo de base de datos (por ej., MongoDB)

2. Diseñar un backend básico de (micro-)servicios que permita realizar el ABM de las entidades (para así poblar y gestionar la(s) base(s) de datos) y dar soporte a las principales funcionalidades antes mencionadas. En este diseño, considerar que cada microservicio contará (preferentemente) con una base de datos separada.

3. Implementar los siguientes servicios/reportes:
   a. Como encargado de mantenimiento quiero poder generar un reporte de uso de monopatines por kilómetros para establecer si un monopatín requiere de mantenimiento. Este reporte debe poder configurarse para incluir (o no) los tiempos de pausa.
   b. Como administrador quiero poder anular cuentas para inhabilitar el uso momentáneo de la misma.
   c. Como administrador quiero consultar los monopatines con más de X viajes en un cierto año.
   d. Como administrador quiero consultar el total facturado en un rango de meses de cierto año.
   e. Como administrador quiero consultar la cantidad de monopatines actualmente en operación, versus la cantidad de monopatines actualmente en mantenimiento.
   f. Como administrador quiero hacer un ajuste de precios, y que a partir de cierta fecha el sistema habilite los nuevos precios.
   g. Como usuario quiero lun listado de los monopatines cercanos a mi zona, para poder encontrar un monopatín cerca de mi ubicación

---

2da ENTREGA

4. En caso que no lo haya hecho en la entrega anterior, refactorizar el diseño (monolítico) para adecuarlo a un diseño de microservicios. Esto implica
   - que se desplieguen en forma independiente (por ej., en puertos distintos),
   - que sus comunicaciones entre sí se realicen mediante servicios REST (para esto, puede considerarse el uso de RestTemplate o WebClient)
   - que se utilicen bases de datos separadas.

5. Segurizar los endpoints REST con JWT.

6. Incorporar tests de unidad e integración (Junit o Mockito). Documentar los endpoints REST con Swagger (OpenAPI).

7. Utilizar una base NoSQL (MongoDB), o bien implementar una comunicación vía protocolo gRPC entre 2 microservicios.

8. (Opcional) Desplegar la aplicación mediante contenedores (Docker) en una nube.
