

Trabajo Práctico Integrador

Backend de Aplicaciones 2023

Estamos trabajando en la implementación de un sistema de alquiler de bicicletas para una cierta ciudad. De acuerdo a nuestro modelo, el sistema opera bajo los siguientes supuestos:

- Llamamos *cliente* a quién está registrado en el sistema para alquilar bicicletas. Únicamente ellos pueden alquilar bicicletas.
- Cada bicicleta que se alquila es retirada de una estación y devuelta en una estación distinta.
- Asumimos que siempre hay una bicicleta disponible en cada estación y que toda estación tiene lugar disponible para una devolución.
- El precio del alquiler se calcula en el momento de la devolución de la bicicleta y este precio se calcula bajo las siguientes reglas:
 - Hay un costo fijo por realizar el alquiler y un costo por hora completa (Cuenta como hora completa a partir del minuto 31, antes de eso se tarifa fraccionado por minuto). Existe una tabla en la base de datos que se provee que indica cuáles son estos costos por cada día de la semana.
 - Se cobra un monto adicional por cada KM que separe la estación de retiro de la estación de devolución. La base de datos provista contiene el precio adicional por KM. El cálculo de esta distancia se explica en las aclaraciones finales
 - Para los días promocionales configurados en el sistema, se aplica un porcentaje de descuento sobre el monto total del alquiler. Para que este descuento se aplique, se considera que el retiro de la bicicleta se hizo en el día promocional. La base de datos provista contiene los días que se consideran de descuento y el descuento aplicable.
- Al momento de la devolución, el cliente decide en qué moneda se le va a mostrar el importe adeudado. Por defecto el monto se muestra en Pesos Argentinos, pero puede expresarse en cualquier otra moneda soportada por el sistema.

Se solicita desarrollar un backend, compuesto por microservicios, que exponga un API REST (con representación en JSON) para soportar, mínimamente, las siguientes funcionalidades:

1. Consultar el listado de todas las estaciones disponibles en la ciudad
2. Consultar los datos de la estación más cercana a una ubicación provista por el cliente.
3. Iniciar el alquiler de una bicicleta desde una estación dada
4. Finalizar un alquiler en curso, informando los datos del mismo y el costo expresado en la moneda que el cliente desee. La moneda puede ser elegida en el momento de

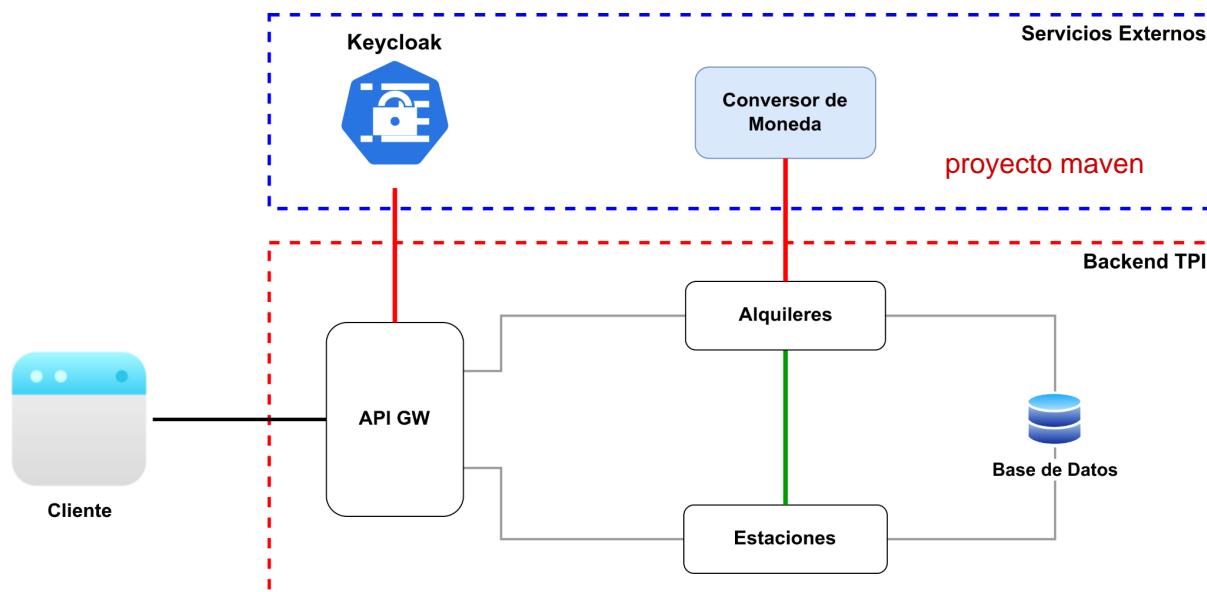
finalizar el alquiler y, en caso de no hacerlo, el monto se expresa en pesos argentinos.

5. **Agregar una nueva estación al sistema**
6. Obtener un listado de los alquileres realizados aplicando, por lo menos, un filtro

Tener en cuenta que:

- ❖ El backend debe presentar un único punto de entrada (debe exponer todos los endpoints en el mismo puerto), para lo que deberá implementar un API Gateway
- ❖ Con este enunciado se entrega una base de datos como base para utilizar en el mismo
- ❖ **Las llamadas a los distintos endpoints únicamente deben ser permitidas para clientes autenticados.** Para esto, se consideran únicamente dos roles:
 - *Administrador*
 - Puede agregar nuevas estaciones
 - Puede obtener listados sobre los alquileres realizados
 - *Cliente*
 - Puede realizar consultas sobre las estaciones, realizar alquileres y devoluciones.
- ❖ Para el desarrollo de este trabajo práctico, será necesario un trabajo de investigación básico sobre manejo de Fecha/Hora en Java. **La recomendación es investigar el uso de las clases del paquete `java.time`**
- ❖ Se considerará, en la evaluación, **el uso correcto de los códigos de respuesta HTTP**
- ❖ **Todos los endpoints que se generen deben estar documentados utilizando Swagger**
- ❖ Para calcular la distancia entre dos estaciones, se considerará simplemente la distancia euclídea entre ambos puntos y cada grado corresponderá a 110000 m. Se aclara que este cálculo no es correcto, pero es suficiente para los fines de este Trabajo Práctico.
- ❖ Este trabajo se acompaña por un foro, cuya finalidad exclusiva es evacuar dudas sobre este trabajo práctico. Las respuestas brindadas en el foro por los docentes de la cátedra tienen tanta validez como lo enunciado aquí.

Una posible (pero no obligatoria) organización de los servicios se muestra en la siguiente imagen:



Al respecto de la base de datos provista, la misma contiene 3 tablas como se detalla a continuación:

- **Estaciones:** Almacena las distintas estaciones disponibles en el sistema. Sus columnas son:
 - *ID*: Identificador de la estación
 - *NOMBRE*: Nombre de la estación
 - *FECHA_HORA_CREACION*: Fecha y Hora en la que se creó la estación en el sistema
 - *LATITUD*: Latitud donde se encuentra ubicada la estación
 - *LONGITUD*: Longitud donde se encuentra ubicada la estación
- **Tarifas:** Configura las tarifas a tener en cuenta para calcular el monto del alquiler. Sus columnas son:
 - *ID*: Identificador de la tarifa
 - *TIPO_TARIFA*: 1 - Tarifa Normal, 2 - Tarifa de Descuento
 - *DEFINICIÓN*: 'S' - Definida por el día de la semana, 'C' definida por día, mes y año.
 - *DIA_SEMANA*: Si la tarifa se define por día de la semana, esta columna indica el día, comenzando por el Lunes (Valor: 1) y terminando en Domingo (Valor: 7)
 - *DIA_MES*: Si la tarifa se define por día, mes y año, esta columna indica el día del mes.
 - *MES*: Si la tarifa se define por día, mes y año, esta columna indica el mes.
 - *ANIO*: Si la tarifa se define por día, mes y año, esta columna indica el año.
 - *MONTO_FIJO_ALQUILER*: El monto fijo a cobrar por iniciar el alquiler
 - *MONTO_MINUTO_FRACCION*: El monto a cobrar por cada minuto extra cuando se tarifa fraccionado (antes del minuto 31)
 - *MONTO_HORA*: El monto a cobrar por cada hora completa de alquiler
 - *MONTO_KM*: El monto a cobrar por cada KM que separa la estación de retiro y la de devolución

- **Alquileres:** Almacena los distintos alquileres realizados en el sistema. Sus columnas son:
 - *ID*: Identificador del alquiler
 - *ID_CLIENTE*: Identificador del cliente que realizó el alquiler
 - *ESTADO*: 1-Iniciado, 2-Finalizado
 - *ESTACION_RETIRO*: ID de la estación donde se retiró la bicicleta
 - *ESTACION_DEVOLUCION*: ID de la estación donde se devolvió la bicicleta
 - *FECHA_HORA_RETIRO*: Fecha y Hora del retiro de la bicicleta
 - *FECHA_HORA_DEVOLUCION*: Fecha y Hora en la que se devolvió la bicicleta
 - *MONTO*: Monto cobrado por el alquiler
 - *ID_TARIFA*: ID de la tarifa que se utilizó para calcular el monto del alquiler