

Prueba Técnica – Full Stack

La siguiente es una prueba para evaluar a los postulantes a Full Stack.

INTRODUCCIÓN

Este proyecto contiene una serie de requerimientos de un Caso Práctico, que busca evaluar las capacidades técnicas del candidato con respecto a las principales funciones y responsabilidades que se requieren dentro del área de Desarrollo de Tecnología de *BlueMedical*

¿Qué se busca evaluar?

Principalmente los siguientes aspectos:

- Creatividad para resolver los requerimientos,
- Calidad del código entregado (estructura y buenas prácticas),
- Utilizar patrones de diseño.
- Explicación de la solución.
- Eficiencia de los algoritmos entregados,
- Familiaridad con Frameworks y plataformas de desarrollo.

IMPORTANTE

1. Duración de la prueba: 72 hrs. (inician al confirmar de recibido)
2. Se requiere de una cuenta de GitHub para realizar este ejercicio.
3. Crear un `respositorio público` en github y enviar el link por correo.
4. Por requerimientos del puesto debes utilizar las siguientes tecnologías como base, PHP, MySQL, Nginx.
5. Se requiere utilizar el framework Laravel para Backend.
6. Utilizar Vue u otro framework o lenguaje de preferencia para Frontend.
7. Compartir screenshots de la aplicación funcionando. .
8. Compartir screenshots del diagrama de base de datos.

Deseable

Utilizar arquitectura de microservicios.

EJERCICIOS

Ejercicio

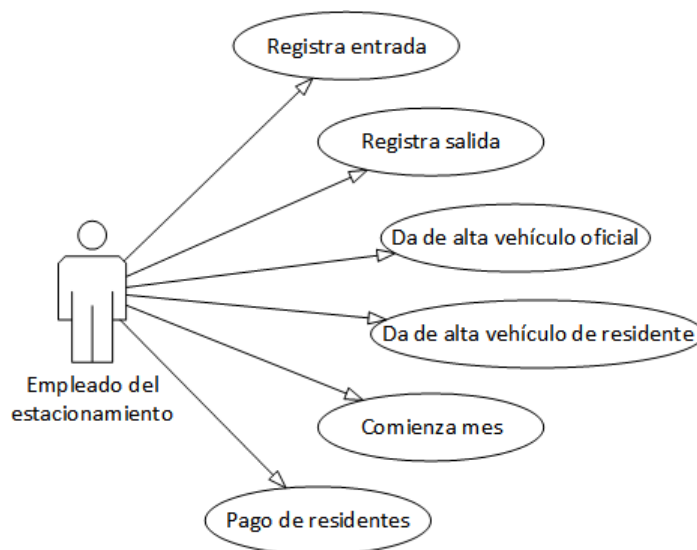
Se desea administrar el acceso de vehículos a un estacionamiento de pago. El estacionamiento no se encuentra automatizado, por lo que existe un empleado encargado de registrar las entradas y salidas de vehículos.

Los vehículos se identifican por su número de placa. Cuando un vehículo entra en el estacionamiento el empleado registra su entrada y al salir registra su salida y, en algunos casos, cobra el importe correspondiente por el tiempo de estacionamiento.

El importe cobrado depende del tipo de vehículo:

- Los vehículos oficiales no pagan, pero se registran sus estancias para llevar el control. (Una estancia consiste en una hora de entrada y una de salida)
- Los residentes pagan a final de mes a razón de \$0.05 el minuto. La aplicación irá acumulando el tiempo (en minutos) que han permanecido estacionados.
- Los no residentes pagan a la salida del estacionamiento a razón de \$0.5 por minuto. Se prevé que en el futuro puedan incluirse nuevos tipos de vehículos, por lo que la aplicación desarrollada deberá ser fácilmente extensible en ese aspecto.

Casos de uso



A continuación se describen los casos de uso. No se entra en detalles de la interacción entre el empleado y la aplicación (punto 1 de cada caso de uso), puesto que no va a ser tarea de este ejercicio desarrollar esa parte.

Caso de uso "Registra entrada"

1. El empleado elige la opción "registrar entrada" e introduce el número de placa del coche que entra.
2. La aplicación apunta la hora de entrada del vehículo.

Caso de uso "Registra salida"

1. El empleado elige la opción "registrar salida" e introduce el número de placa del coche que sale.
2. La aplicación realiza las acciones correspondientes al tipo de vehículo:
 - Oficial: asocia la estancia (hora de entrada y hora de salida) con el vehículo
 - Residente: suma la duración de la estancia al tiempo total acumulado
 - No residente: obtiene el importe a pagar

Caso de uso "Da de alta vehículo oficial"

1. El empleado elige la opción "dar de alta vehículo oficial" e introduce su número de placa.
2. La aplicación añade el vehículo a la lista de vehículos oficiales

Caso de uso "Da de alta vehículo de residente"

1. El empleado elige la opción "dar de alta vehículo de residente" e introduce su número de placa.
2. La aplicación añade el vehículo a la lista de vehículos de residentes.

Caso de uso "Comienza mes"

1. El empleado elige la opción "comienza mes".
2. La aplicación elimina las estancias registradas en los coches oficiales y pone a cero el tiempo estacionado por los vehículos de residentes.

Caso de uso "Pagos de residentes"

1. El empleado elige la opción "genera informe de pagos de residentes" e introduce el nombre del archivo en el que quiere generar el informe.
2. La aplicación genera un archivo que detalla el tiempo estacionado y el dinero a pagar por cada uno de los vehículos de residentes. El formato del archivo será el mostrado a continuación:

Núm. placa	Tiempo estacionado (min.)	Cantidad a pagar
S1234A	20134	1006.70
4567ABC	4896	244.80
...

La aplicación contará con un programa principal basado en un menú que permitirá al empleado interactuar con la aplicación (dicho programa principal no forma parte de este ejercicio).

Persistencia de datos

La información de cada una de las estancias de los vehículos será almacenada en una base de datos. Debido a que el manejador de base de datos puede ser modificado en cualquier momento, se utilizará un ORM de tu preferencia y debes utilizar MySQL como base de datos principal.

Puntos que se deben desarrollar

- Diagrama de clases y diagrama de secuencia de las partes encargadas de la aplicación.
- Aplicación para gestionar las estancias de los vehículos. Deberá incluir:
 - Código de las clases que permitan gestionar los vehículos con sus datos asociados (estancias, tiempo, etc.), las listas de vehículos registrados como oficiales y residentes, etc.
 - Mapeo de las clases para poder almacenar la información en la base de datos.
 - Clases para gestionar la persistencia de datos, incluida la configuración de conexión a la base de datos.
- Utilizar estándares REST de manera correcta entre microservicios
- Incluir una forma de autenticación entre microservicios