**Plataforma Taller de Vehiculos**

La empresa LR Pelaez ofrece servicios de reparación, mantenimiento, entre otros. Últimamente, la empresa LR Pelaez se ha encontrado a sí misma con el problema de que no tiene suficiente espacio para que sus clientes puedan parquear sus vehículos. Esto se debe a que no tienen una forma de medir en tiempo real la capacidad de sus instalaciones para dar cabida a vehículos. Por este motivo la empresa LR Pelaez lo ha contactado a usted y a su equipo para el desarrollo de una plataforma de agendamiento de citas para los diferentes servicios ofrecidos.

Los tipos de usuarios que debe manejar el sistema son administrador, usuario de planta y usuario mecánico.

El usuario de planta debe ser creado por el administrador y debe estar en la capacidad de:

* Configurar los servicios ofrecidos por el taller (revisión de frenos, pastillas, discos, suspensión, amortiguadores, cambio de aceite, alineación, y rotación de llantas). Para cada uno de los servicios se debe indicar: descripción, costo y duración.
* Modificar el estado de un servicio (Disponible, no disponible).
* Asignar un servicio a un usuario mecánico.
* Ver agenda de citas por día de cada servicio.
* Programar cita para un servicio indicando la placa del vehículo a revisar.
* Cancelar una cita previamente solicitada (Con mínimo 24 horas de antelación).
* Generar los siguientes reportes:
  + Servicio más solicitado.
  + Servicio menos solicitado.
  + Listado de mecánicos y sus asignaciones por día.
  + Servicios completados.

Un usuario mecánico debe estar en la capacidad de:

* Asignar el estado de completado a un servicio.
* Ver listado de sus asignaciones.
* Añadir comentarios sobre el estado del vehículo de un servicio.
* Cambiar estado (En reparación, reparado)

Un administrador se encarga de gestionar los usuarios de planta y mecánicos y además ejerce control total de la plataforma.

***Notas***:

* ***Los datos suministrados por los usuarios deben cumplir la política de privacidad de datos vigente, es decir las contraseñas de los usuarios deben almacenarse de forma cifrada y la conexión al servidor debe realizarse de forma segura.***
* ***La base de datos debe ser no relacional.***
* ***Se debe utilizar una arquitectura desacoplada, es decir, una API back end y un front end que consuma dicha API.***
* ***La API back end debe estar desarrollada en Express.js***
* ***El front end debe estar desarrollado en React.js***
* ***Se debe utilizar Bootstrap como librería CSS para manejar los estilos de su aplicación.***