

## Backend test - C#

Una veterinaria requiere ofrecer a sus clientes una plataforma para el control general de sus mascotas.

La plataforma debe contener un módulo para inicio de sesión y registro de clientes. El cliente deberá registrarse por medio de su cédula y una contraseña alfa numerica. Una vez que el cliente se haya registrado deberá dirigirse a su perfil para completar su información personal (Apellido, dirección, fecha de nacimiento, email, nombre, sexo, teléfono, tienda y una foto opcional).

En el perfil, el cliente dispondrá de su información personal como el nombre completo, foto, edad, sexo, email y número de teléfono. Así mismo tendrá acceso a dos botones uno para actualizar el perfil y el segundo para dirigirse a la página principal.

El cliente podrá interactuar en la plataforma por medio de una barra de navegación desde el cual podrá acceder al perfil, registro de mascotas y tareas. Adicionalmente en la página principal se debe desplegar el listado de las mascotas registradas con la siguiente información (Nombre, Apodo, costo total, fecha de próxima tarea, Número de tareas pendientes del día, porcentaje de tareas realizadas y raza de la mascota). Es requerido agregar un filtro que permita realizar una búsqueda por nombre de mascota o número de tareas pendientes. Además, se le permitirá al cliente ordenar por cualquier columna (Nombre, Apodo, fecha de próxima tarea, Número de tareas pendientes del día, Número de tareas del día y raza). Para finalizar se requiere mostrar el costo total de las tareas del día de todas las mascotas del cliente, ubicar antes del listado.

El módulo de mascota será un CRUD, los campos a incluir serán (Apodo, nombre, fecha de nacimiento, foto, sexo y raza). Este módulo dispondrá de dos servicios, uno para registrar una mascota de un cliente y otro para obtener las mascotas de dicho cliente (Retornar máximo 5 mascotas por petición, agregar tareas pendientes del día por mascota).

Para finalizar se agregará un CRUD para que el cliente pueda gestionar las tareas de una mascota, los campos a incluir serán (Mascota, tipo de tarea, nombre de tarea, fecha y hora).

Tipo de tarea	Nombre de Tarea	Costo
Campo	Salida al parque	1000\$
Campo	Entrenamiento	5000\$
Campo	Vacunas	10000\$
Casa	Inspección general	

Casa	Alimentación	12500\$
------	--------------	---------

#### Administrador:

El administrador de la plataforma tendrá la facultad de acceder y manipular toda la información de los clientes. Además se le agregará un módulo para el registro de sus tiendas.

#### Tiendas:

Las tiendas deben estar relacionadas por ciudades (Barranquilla, Bogotá, Cali, Cartagena y Medellín). Los clientes en el módulo de su perfil podrán seleccionar la tienda de su preferencia filtrando por ciudad. Para finalizar el administrador debe tener la ubicación de las tiendas en un mapa, además debe tener acceso a un listado con la siguiente información (Cédula del cliente, Nombre del Cliente, tienda, Apodo de mascota y número de tareas asignadas), incluir filtro por tienda.

#### Consideraciones:

- El cliente no podrá agregar más de 5 tareas a una mascota por día.
- Tanto las rutas de los endpoints como los parámetros a enviar para realizar las peticiones deberán ser entregados.
- Todos los campos de las tareas serán requeridos
- Restringir cualquier acceso de los clientes a las rutas de administrador.
- Tener en cuenta los requerimientos del perfil para el desarrollo del código.

#### Bonus point:

Realizar un endpoint que retorne una lista en relación al tipo de tarea. Campos a retornar (Tipo de tarea, Nombre de Tarea, nombre completo del cliente, edad, nombre de mascota, raza, fecha de tarea). Agregar paginación (10 máximo por página) y un filtro por nombre y tipo de tarea.

#### Criterios de evaluación

Aplicación del patrón MVC

Validaciones

Pruebas unitarias

#### Presentación

Código de la aplicación en Github o Bitbucket

\*\* Si se despliega la aplicación en un servicio de su preferencia se debe incluir la url de la aplicación.