

# Parcial 2 2024 - Desarrollo Web & Mobile

## Contexto

Con el auge de las aplicaciones móviles para organizar actividades al aire libre, una agencia de viajes está interesada en desarrollar una app que permita a los usuarios explorar destinos de aventura y gestionar sus planes de viaje.

Tu tarea es crear una aplicación en React Native que funcione como asistente de viajes, permitiendo a los usuarios descubrir destinos, guardar sus favoritos, y planificar sus actividades.

## Tareas

### Vista de Destinos (30pts):

- Obtener lista de los destinos.
- Mostrar el nombre del destino y una etiqueta con la dificultad del viaje. La dificultad del viaje puede ser Fácil (representada con un tag verde), Moderada (representada con un tag amarillo), Difícil (representada con un tag violeta).
- Mostrar los destinos ordenados. Todos los destinos que sean favoritos deben aparecer primero y ordenados alfabéticamente.

### Agregar nuevos destinos y editar destinos (30pts)

Permitir que los usuarios sugieran nuevos destinos turísticos o pueda editarlos:

1. Crear un formulario que solicite:
  - a. Nombre del destino.
  - b. Descripción breve.
  - c. Nivel de dificultad, esto se debe hacer agregando 3 checkboxes (solo una debe poder ser marcada)
2. Tanto los nuevos destinos como los destinos editados deben reflejarse en la lista de destinos.
3. Esta acción debe ser creada en una nueva pantalla dentro de un nuevo tab.

### Eliminar destinos (20pts)

Permitir que los usuarios eliminen destinos que hayan agregado:

1. El usuario podrá eliminar destinos.
2. La acción debe actualizar la interfaz en tiempo real.

## Favoritos (10 pts)

Permitir a los usuarios marcar y desmarcar destinos como favoritos:

- Los favoritos deben poder ser marcados/desmarcados desde la lista de destinos.
- Los favoritos se almacenarán por parte del backend.

## Responsividad (10 pts)

Si bien estamos trabajando en aplicaciones móviles, no todas tienen el mismo tamaño y resolución. Para asegurarnos que el listado de destinos se muestre bien en todos los dispositivos, se pide que el mismo tenga un ancho máximo del 85% del tamaño del dispositivo.

-----

## Tareas opcionales

### Diferencias Android / iOS

Sería ideal que la aplicación contara con diferencias visuales según la plataforma en la cual la aplicación se encuentra corriendo.

Para ello se pide que el botón que permite agregar un nuevo destino como favorito tenga las siguientes características dependiendo de la plataforma:

- Android: Una estrella con background transparente cuando no está agregado como favorito o una estrella con background amarillo cuando no está agregado como favorito.
  - IOS: Un corazón con background transparente cuando no está agregado como favorito o un corazón con background rosado cuando no está agregado como favorito.
- 

## Notas Adicionales

Para este parcial, se utilizará la herramienta json-server. Deben primero instalarla corriendo el comando "npm i json-server -g" en la terminal. Una vez hecho esto, deben descargarse el archivo db.json provisto como recurso, navegar en la terminal hacia el directorio donde se encuentra el archivo y correr "json-server --port 8000 ./db.json --watch" para levantar el backend en el puerto 8000.

La API del backend tiene sus endpoints registrados bajo la ruta /destinos.

Las operaciones de POST, PUT y DELETE NO devuelven la información de forma actualizada, por lo que es necesario volver a hacer un GET.

### **Endpoints proporcionados por el servidor:**

1. Obtener todos los destinos:  
GET /destinations
2. Obtener un destino:  
GET /destinations/:id
3. Agregar un destino:  
POST /destinations
4. Editar un destino:  
PUT /destinations/:id
5. Eliminar un destino:  
DELETE /destinations/:id
6. Marcar o desmarcar como favorito:  
PATCH /destinations/:id

Ejemplo estructura objeto destination:

```
{  
  name: String  
  description: String  
  difficulty: String  
  isFavorite: Boolean  
}
```

Nota: Recordar que el PATCH actualiza solo el atributo enviado y el PUT actualiza todo el objeto.