

# Prueba Técnica React Native

## Contexto

Se dispone de un servicio web que devuelve un árbol de destinos. La url del servicio es la siguiente:

<https://api-int.stay-app.com/v1/establishments/destinations?apikey=4ef1c773b26fea71a6638c948dab3bb1>

Este servicio funciona mediante el método GET, no tiene autenticación, y recibe un parámetro apikey que identifica la aplicación a la que pertenece el árbol de destinos.

El servicio devuelve un JSON con tres campos:

- Un campo success que contiene un flag indicando si la consulta ha terminado satisfactoriamente.
- Un campo errors que contiene un array de errores, si los ha habido.  
Ejemplo:  
["AUTH\_FAILED\_API\_KEY\_NOT\_VALID"]
- Un campo data que contiene el array de nodos del árbol de destinos.

Los campos de un nodo son los siguientes:

| Campo                            | Tipo de dato | Observaciones  |
|----------------------------------|--------------|--|
| id                               | String       | ID del nodo  |
| childs                           | Array        | Array de subnodos  |
| destinationData.translatableName | Diccionario  | Nombre del destino   |
| destinationData.photographs      | Array        | Array de IDs de imágenes   |
| destinationData.coordinates      | Diccionario  | Coordenadas del destino  |
| fatherDestination                | String       | ID del nodo padre  |
| isTop                            | Booleano     | Flag que indica si es un destino destacado                       |
| isFinalNode                      | Booleano     | Flag que indica si es un nodo final                              |
| numEstablishments                | Entero       | Número de establecimientos asociados a ese nodo o a sus subnodos |

## Tarea a realizar

Se solicita una aplicación que consuma dicho endpoint y muestre los nodos en un listado.

Cada celda mostrará al menos los siguientes datos:

- Nombre del destino.
- En caso de ser un destino destacado, un Indicativo que lo identifique como tal.

Se podrá visualizar nodos anidados ya sea directamente en el listado o mediante navegación, celdas desplegadas o cualquier otra solución elegida por el candidato.

Como punto adicional (no obligatorio), al seleccionar un nodo final se mostrará una nueva pantalla, muy sencilla, que muestre el nombre del destino.

## Requisitos

- Se deberá usar React Native y TypeScript.
- Deberá funcionar en dispositivos Android e iOS.
- En STAY usamos una arquitectura hexagonal, pero se podrá estructurar el código como se quiera, no hay límite en el número de clases a usar ni restricciones a la hora de aplicar los patrones de diseño que se crean necesarios.
- Se podrán utilizar las librerías y/o frameworks que se consideren.
- El diseño de la interfaz queda a elección del desarrollador siempre que cumpla con los requisitos mínimos antes mencionados.
- Se deberá adjuntar un resumen en un README con una breve descripción de las decisiones tomadas y la forma de probar el desarrollo.

## ¿Qué se evaluará?

- Estilo de código: cómo se estructura el código, naming, formato, patrones y/o principios de diseño que se utilizan.
- Arquitectura de la solución. no somos amigos de la sobreingeniería, pero valoraremos el uso de una arquitectura hexagonal o similar.

## ¿Cómo enviar la prueba?

- Puedes alojar la prueba en un repositorio público de Github o Bitbucket y compartírnos el enlace.
- Por supuesto, los derechos de tu código, son tuyos y de nadie más.

