

ACOMPañAMIENTO POR TEMÁTICA DE PI

El objetivo de este documento es darte una guía sobre cómo estructurar el desarrollo de tu Proyecto Individual. ¡De esta forma podrás medir tus avances diarios!

Si algunos días completas las tareas definidas en este documento, te sugerimos que sigas avanzando con las del siguiente día. Podrás ahorrar tiempo, corregir bugs, darle estilos a tu app, crear testing, etc.

IMPORTANTE

Revisa este [CALENDARIO](#) de actividades durante el desarrollo del PI.

DIA 1

Enviar mensaje de presentación y actividades a realizar.

- ☐ Instalar dependencias en Back-End y Front-End.
- ☐ Crear la base de datos (por consola o líneas de comando).
- ☐ Levantar el entorno completo (y sin errores).
- ☐ Analizar estructura de datos de la API.
 - [JSON Formatter](#) te permitirá ver la información de una forma más ordenada.
- ☐ Diagrama de entidad/relación de la base de datos.
- ☐ Subir repositorio a GitHub.

Documentación que te puede ayudar este primer día.

- [Como subir un proyecto local a GitHub](#)
-

DIA 2

- ☐ Generar los modelos de la base de datos según los campos especificados en el readme.
 - Recuerda usar un tipo de dato UUID para tus elementos (excepto si tu temática es "Countries").
- ☐ Definir la relaciones entre modelos de tipo N a N.
- ☐ Visualizar la generación de la tabla intermedia en la base de datos.
- ☐ Tener en cuenta nombres de propiedades en la API.

Documentación que te puede ayudar este segundo día.

- [Types de datos, aquí podrán ver que es UUID y demás valores](#)

- [Relaciones de una base de datos](#)

DIA 3

- ☐ Ruta GET para cargar información de tipos/géneros/etc (según temática).
- ☐ Utilizar algún método que evite la carga de duplicados.
- ☐ Visualizar carga correcta de tipos en la base de datos.

Documentación que te puede ayudar este tercer día.

- [Instalar Postman](#)
- [Instalar Insomnia](#)
- [Como poner Thunder Client](#)

DIA 4

- ☐ Definir la ruta GET para buscar información de la API.
 - Conviene que ambos tengan la misma estructura de datos, para que en el Front-End se muestren de igual manera, ya sea que vengan de la BD o de la API.
- ☐ Modularizar el Back-End, separando rutas y controladores.
- ☐ Definir ruta para obtener un elemento por **id**.
 - Debes obtener toda la información necesaria para mostrar en el componente **Detail** de dicho elemento.
 - El tipo de dato del **id** te indicará si debes buscar en la API o en tu base de datos.

Documentación que te puede ayudar este cuarto día.

- [Guia basica Express](#)
- [¿Cuándo utilizar body/query/params?](#)
- [¿Cuándo utilizar body/query/params? \(parte 2\)](#)

DIA 5

- ☐ Definir ruta POST para crear un elemento en la base de datos.
 - Recibe información por body.
 - Tener en cuenta que el Front-End deberá enviar información en un formato adecuado.
 - Debes obtener toda la información necesaria para almacenar en la BD, junto con los tipos asociados a dicho elemento. Primero almacenas el

nuevo elemento. Paso seguido, a dicho elemento almacenado, le añades los tipos.

- ☐ Buscar elementos por nombre.
 - Utiliza alguna extensión o herramienta que te permita hacer llamados tipo POST (ej:Thunder Client, Postman, Insomnia, etc)
- ☐ Asegúrate de que el nuevo elemento creado se almacenó en tu BD, y que su ID está asociado a los tipos asociados (mira la tabla intermedia)

Documentación que te puede ayudar este quinto día.

- [Guia basica Express](#)
- [¿Cuándo utilizar body/query/params?](#)
- [¿Cuándo utilizar body/query/params? \(parte 2\)](#)

DIA 6

- ☐ Componentes: Landing Page, Home, Cards, Card y Detail.
 - Al menos para poder configurar las rutas.
- ☐ React Router Dom
 - Routing en componentes del Front-End.

Documentación que te puede ayudar este sexto día.

- [Documentación de React \(Antiguo\)](#)
- [Documentación de React \(Actual\)](#)

DIA 7

- ☐ Configuración de Redux (store (Provider), Reducer, actions)
- ☐ Completar componentes Card, Cards y Detail de cada elemento.
- ☐ Comenzar a dar estilos básicos.

Documentación que te puede ayudar este séptimo día.

- [Redux Toolkit](#)

DIA 8

- ☐ Genera el paginado del sitio, para mostrar la cantidad de elementos solicitados por página.
- ☐ Coloca un searchbar para buscar elementos por nombre
 - Debes informar al usuario, si realiza la búsqueda de un elemento que no existe

Documentación que te puede ayudar este octavo día.

- Conceptos básicos de [Flexbox](#)
- [Froggy Flexbox](#)

DIA 9

- ☐ Formulario de creación de nuevo elemento.
 - Tener en cuenta la información que necesita la ruta del back.
- ☐ El formulario debe tener todos los datos que almacena ese modelo en la base de datos. Además, debe permitir agregar varios tipos asociados a dicho elemento.
- ☐ Validaciones de campos con JavaScript.
 - Mediante estados locales de React.
 - Utiliza formularios reactivos (que los errores aparezcan a medida que los campos se completan de forma incorrecta y que desaparezcan al llenarlos correctamente).
 - El botón del formulario debe estar deshabilitado, hasta que todos los campos requeridos estén completos y sin errores.
 - Coloca los tipos disponibles en la base de datos para que el usuario pueda elegir varios (1 o más).
 - Dale al usuario, la posibilidad de eliminar algún tipo seleccionado (si se confundió al agregarlo), antes de que pueda crear el nuevo elemento.
 - Muestra al usuario un mensaje que informe que el elemento fue creado correctamente y en caso de haber un error (que informe que no se ha creado).

DIA 10

- ☐ Genera las funciones para ordenamiento y filtrado de elementos.
- ☐ Testear la aplicación y anótalos para solucionarlos.
- ☐ Refactor y corrección de bugs.
- ☐ Dale a la página un estilo homogéneo. Usa paleta de colores, botones con el mismo estilo en todo el sitio.
 - El filtrado te devolverá menos cantidad del total de elementos.
 - Si filtras por un tipo y aplicas ordenamiento, el ordenamiento se debe aplicar solo a los elementos filtrados, no a todos los elementos
 - Coloca un botón para limpiar los filtros, para volver a mostrar la totalidad de elementos

DIA 11

- ☐ Puedes hacer testing.
- ☐ Si en el día 10 encontraste bugs, en el día de hoy debes de arreglarlos.
- ☐ Entender y practicar el flujo de mi código
- ☐ Puedes agregar algo extra: mínimos detalles que crees pueden sumar valor a lo que tienes. Ej: Borrado de un elemento de mi BD, likes a elementos o al sitio, etc.
- ☐ Recuerda repasar un poco de teoría. Piensa, ¿Qué problema resuelven?: (useEffect, useState, el reducer, store, actions, la relación en la base de datos, las rutas, etc).

DIA 12

Limpia la aplicación de bugs

- ☐ Sigue practicando para la presentación.
- ☐ Empolija tus estilos
- ☐ Retira todos los console.log