

Guía de seguimiento para el proyecto dual.

1. Trabajo a través de API REST. Investiga la tecnología asociada al sistema RestAPI, desarrollando los distintos tipos de mensajes para el envío y recepción de datos.

<https://aws.amazon.com/es/what-is/restful-api/>
<https://www.ibm.com/mx-es/think/topics/rest-apis>
<https://dev.to/dennysjmarquez/todo-lo-que-necesitas-saber-sobre-api-rest-glosario-de-terminos-esenciales-y-mas-29pc>

Nota: consulta el sistema de intercambio de información a través de JSON.

<https://www.arsys.es/blog/formato-json-que-es-y-para-que-sirve>

A través de estos links iniciales u otros que encuentres interesantes, elabora un informe con las ideas principales de esta tecnología, añade una pequeña lista de recursos, software, enlaces...

2. Consumo de API REST de Países del Mundo En la URL <https://restcountries.com/> se encuentra disponible una API REST pública que permite obtener información sobre todos los países del mundo, como su nombre, capital, población, continente, entre otros datos relevantes.

Práctica el paso de mensajes sobre esta api usando el software [Postman](#) (web o escritorio). Prepara un pequeño documento dónde se explique de forma detallada cómo realizar estas llamadas.

3. Objetivo Implementar dos clientes web que consuman esta API y permitan realizar diferentes tipos de consultas.

Requisitos Cliente 1: Vanilla JavaScript (Puro). Crear una aplicación web utilizando HTML, CSS y JavaScript puro (sin frameworks). La aplicación debe permitir: Consultar todos los países (consulta global). Consultar países filtrados por continente. Consultar un país específico por su nombre.

Investiga a través de estos enlaces el uso de Fetch para la consulta de APIRest a través de Vanilla Javascript.

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API
https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch
<https://www.freecodecamp.org/espanol/news/javascript-fetch-api-para-principiantes/>
<https://www.digitalocean.com/community/tutorials/how-to-use-the-javascript-fetch-api-to-get-data>
<https://es.javascript.info/fetch-api>

Una vez conseguidos los objetivos iniciales, extender el filtro para poder seleccionar por población (mínimo y máximo) y por idioma (de todos los posibles).

Requisitos cliente 2: React.js. Crear una aplicación con React.js. Incluir componentes adecuados para organizar la lógica y la presentación. La aplicación debe permitir las mismas funcionalidades que el cliente en Vanilla JavaScript.

<https://es.react.dev/>

<https://develohero.io/blog/react-tutorial>

<https://es.react.dev/learn/tutorial-tic-tac-toe>

<https://react.dev/learn>

Detalles adicionales.

- El diseño debe estar enfocado en la funcionalidad. Mostrar los resultados en una tabla, lista o tarjetas (a elección del desarrollador). Se recomienda usar fetch() en el cliente Vanilla JS y axios o fetch en React.

-Incluir manejo de errores en caso de que la API no responda o no se encuentren resultados.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/try...catch>

https://medium.com/@armando_amador/how-to-make-http-requests-using-fetch-api-and-promises-b0ca7370a444

4. Implementación de una API REST para registro de consultas.

Objetivo. Diseñar e implementar una API REST que registre en una base de datos las consultas realizadas a la API de países (<https://restcountries.com/>), con el fin de generar estadísticas de uso. Esta API debe permitir guardar, consultar y analizar la información de las búsquedas realizadas por los usuarios desde los clientes desarrollados en la Parte 1.

Se debe implementar una API REST utilizando alguno de los siguientes entornos:

PHP con un framework como Laravel.

<https://kinsta.com/blog/laravel-api/>

<https://dev.to/rohitsondigala/creating-a-restful-api-in-laravel-a-comprehensive-guide-27oi>

<https://cvallejo.medium.com/c%C3%B3mo-crear-una-api-en-laravel-10-367fea542f88>

Python con Flask.

<https://flask.palletsprojects.com/en/stable/>

<https://auth0.com/blog/developing-restful-apis-with-python-and-flask/>

Prepara una API Rest como prueba (en una de las dos tecnologías) y haz consultas a cada uno de los endpoints implementados a través de Postman. Recuerda que siempre debe usar JSON como sistema de intercambio.

Funcionalidades mínimas de la API.

POST /consultas Guardar una consulta realizada por un usuario. Se deben registrar los siguientes datos: Tipo de consulta (global, por continente, por país) Parámetro usado (si aplica, por ejemplo: "Europa" o "Argentina") Fecha y hora IP del usuario

GET /consultas Obtener todas las consultas registradas (modo administrador o para análisis).

GET /estadisticas Retornar datos agregados, como: Total de consultas por tipo País o continente más consultado Número de consultas por día

Recomendaciones técnicas.

- Utilizar una base de datos relacional (MySQL/MariaDB, PostgreSQL).
- Normalizar la estructura de la tabla de consultas. Implementar control de errores y validación de datos recibidos.
- Implementar una interfaz web sencilla para visualizar las estadísticas.

Extra (opcional).

- Implementar autenticación básica para proteger el acceso a los datos de análisis.