



Contexto:

Una de las habilidades más básicas de todo Frontend developer es saber presentar el contenido generado por una API en un formato atractivo y cómodo de utilizar por los usuarios. Con la amplia variedad de APIs disponibles es muy importante saber encontrar qué API necesitamos para una tarea concreta y navegar por su documentación.

Queremos introducir nuevos alimentos en nuestra dieta, y para ello hemos decidido crear una web que nos muestre la información de distintas recetas y sus ingredientes. [Spoonacular API](#) es una API que da todo tipo de información de comida: recetas, ingredientes, nutrientes, etc. Este reto consiste en crear una web que muestre distintos platos y sus detalles.



Objetivos:

1. Tener un layout de cards que contengan imágenes de Recetas (aunque no esté conectado directamente a la API). Pueden ser recetas populares, aleatorias, organizadas en categorías, etc.
2. Hacer peticiones a la API para cargar las recetas de varios platos.
3. En hacer clic en una receta, navegar hacia otra ruta donde hay más información sobre ese plato. Se puede mostrar lo que se desee, pero como mínimo se tendrían que ver los ingredientes. Debe haber algún modo de volver atrás.
4. Implementar una barra de búsqueda y/o filtros.

Mejoras que se pueden implementar



Animaciones → Se pueden implementar todas las animaciones que queráis.



Caching → Hacer caching de llamadas consultadas anteriormente



Calidad de código → Añadir al readme la calidad de código que devuelve alguna de las siguientes plataformas [Codefactor](#) o [Sonarcloud](#)

Nota

La API está limitada a solo 150 llamadas/día. En caso de estar probando y necesitar más podéis usar un mail temporales para generar otra API key.



Evaluación:

La evaluación se basará en los objetivos cumplidos, en la calidad de código y en la documentación.



En el caso de la **calidad de código** utilizamos SonarQube y softwares similares para medir errores de estilo, complejidad ciclomática, bugs, etc, a partir de los que se saca una puntuación.

Para la **documentación** lo que se quiere saber es el porqué de las tecnologías utilizadas, ver cómo se documenta el proyecto (por pequeño que sea), cómo se ha planteado la distribución de carpetas, la arquitectura, si se ha explicado cómo empezar a trabajar o contribuir en la repo, etc. Tanto los readme de una única línea como los generados por defecto de algunos frameworks no sirven.

Se recomienda utilizar el "common-readme", "standard-readme" o similares para mantener una estructura limpia y sólida: [NUWE](#) [README](#) [Templates](#).

Recursos de free hosting

1. [Vercel](#)
2. [Firebase](#)
3. [Github Pages](#)
4. [Render](#)
5. [Surge](#)
6. [Netlify](#)



Submission:

Subir la solución a la plataforma a través de un repositorio Github.