

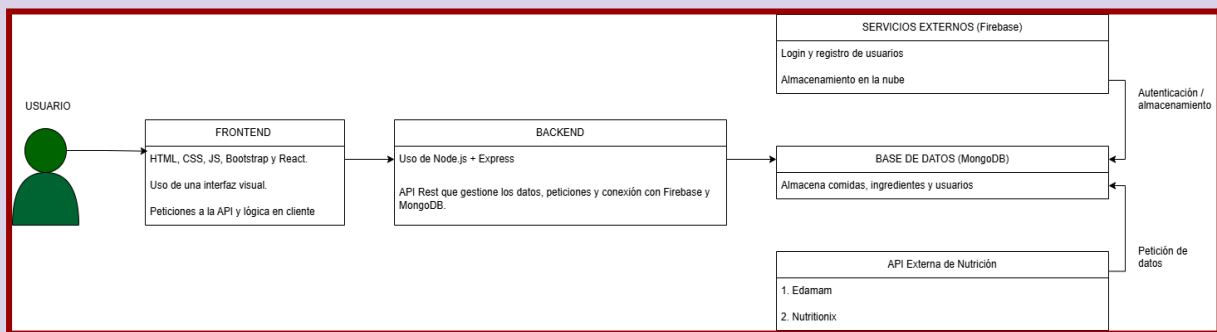
# Diseño de la arquitectura y preparación tecnológica

Proyecto escogido: **Calculadora de Calorías Personalizada**

## Objetivo del sprint

Definir cómo estará estructurado el proyecto a nivel técnico (arquitectura general) y preparar los entornos base de las tecnologías elegidas para comenzar el desarrollo en los siguientes sprints.

## Diseño esquemático del sistema



## Descripción del esquema

- Frontend ⇒ Backend:**  
El usuario interactúa desde la interfaz (HTML, React y Bootstrap). Cuando añade o consulta comidas, el navegador envía peticiones HTTP (fetch) al backend en formato JSON.
- Backend ⇔ Base de datos:**  
El servidor gestiona las operaciones CRUD (crear, leer, actualizar, eliminar) sobre las colecciones de usuarios, comidas e ingredientes en MongoDB.
- Backend ⇔ Firebase:**  
Se usa para autenticación (login/registro mediante Firebase Auth) y posible almacenamiento en la nube (Firebase Storage).
- Backend ⇒ API de nutrición:**  
El servidor realiza peticiones a la API externa (por ejemplo, Nutritionix) para obtener la información nutricional de ingredientes no registrados.
- Backend ⇒ Frontend:**  
Devuelve las respuestas en JSON que React usa para renderizar la información en la interfaz.

## Justificación de las tecnologías elegidas

COMPONENTE	TECNOLOGÍA/S	JUSTIFICACIÓN
FRONTEND	HTML5/CSS3, JavaScript, React.	HTML y CSS forman la base de la estructura y estilo. Bootstrap acelera la maquetación responsive y React permite crear componentes dinámicos e interactivos.
BACKEND	Node.js + Express.js	Facilita la creación de APIs REST con buen rendimiento. Ideal para integrarse con MongoDB y servicios externos como Firebase.
BASE DE DATOS	MongoDB	Base NoSQL flexible, adecuada para almacenar información de comidas, ingredientes y usuarios sin una estructura rígida.
SERVICIOS EXTERNOS	Firebase(Auth y Storage)	Simplifica el sistema de autenticación y el almacenamiento remoto, sin necesidad de configurar un servidor adicional.
API EXTERNA	Nutritionix / Edamam	Permite obtener datos nutricionales actualizados, evitando mantener una base propia de calorías.
CONTROL DE VERSIONES	GitHub	Facilita la gestión de versiones y la colaboración entre frontend y backend.
ENTORNO DE DESARROLLO	VScode + npm + Postman	Entorno de desarrollo moderno, ligero y compatible con Node.js y React.

## Creación de repositorios correspondientes para el proyecto

- <https://github.com/victormanuel-98/VMRC-PI-FRONT>  
Contendrá la interfaz de usuario desarrollada con **HTML**, **CSS**, **JavaScript** y **React**.
- <https://github.com/victormanuel-98/VMRC-PI-BACK>  
Incluirá la lógica del servidor implementada con **Node.js**, conexión a **MongoDB** y posible integración con **Firebase** para autenticación o almacenamiento.

## Planificación en Jira con tareas técnicas especializadas

enlace a Jira:

<https://adaits-team.atlassian.net/jira/software/projects/GDCS2AYD/boards/2/timeline>