

PROYECTO INTERMODULAR

SPRINT 5 DISEÑO MODELADO (BACKEND)

Actividad 3 – Diagramas de Secuencia

1. Búsqueda de alimento en la BD y obtener datos nutricionales

Incluye la comunicación con la API externa (opcional)¿?

CÓDIGO

```
@startuml
actor Usuario
participant "FrontEnd" as FE
participant "BackEnd" as BE
participant "API Externa de Alimentos" as API

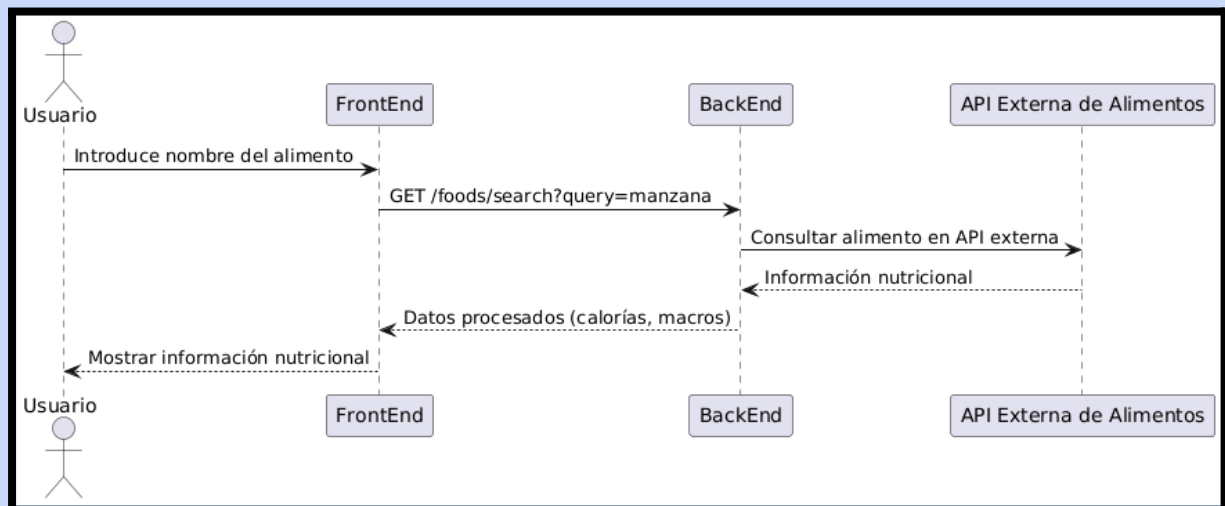
Usuario -> FE: Introduce nombre del alimento
FE -> BE: GET /foods/search?query=manzana

BE -> API: Consultar alimento en API externa
API --> BE: Información nutricional

BE --> FE: Datos procesados (calorías, macros)
FE --> Usuario: Mostrar información nutricional

@enduml
```

IMAGEN



2. Añadir alimento a una comida

Aquí se muestra cómo se registra la comida en el Backend y se actualizan las calorías totales en la BD.

CÓDIGO

```
@startuml
actor Usuario
participant "FrontEnd" as FE
participant "BackEnd" as BE
database "Base de Datos" as DB

Usuario -> FE: Selecciona alimento y cantidad

FE -> BE: POST /meals/addFood {alimentoId, cantidad}

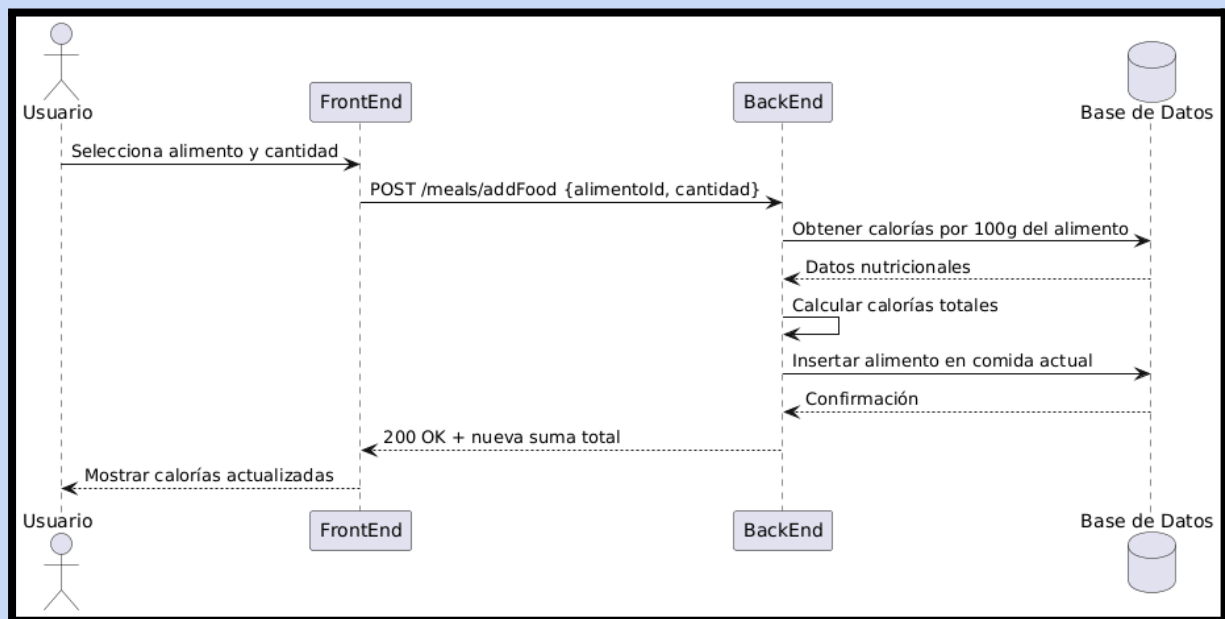
BE -> DB: Obtener calorías por 100g del alimento
DB --> BE: Datos nutricionales

BE -> BE: Calcular calorías totales
BE -> DB: Insertar alimento en comida actual
DB --> BE: Confirmación

BE --> FE: 200 OK + nueva suma total
FE --> Usuario: Mostrar calorías actualizadas

@enduml
```

IMAGEN



Actividad 4 – Diagrama de Componentes

Aquí se define la arquitectura lógica con los diagramas de Backend más uno global.

1. Diagrama de componentes (Backend)

Este diagrama muestra la API, los módulos internos, la capa de negocio y la base de datos.

CÓDIGO

```
@startuml
title Arquitectura de Componentes - Backend

package "Backend (Node/Express o Spring o similar)" {

    component "API REST" as API

    component "Módulo de Usuarios" as UserModule
    component "Módulo de Alimentos" as FoodModule
    component "Módulo de Recetas" as RecipeModule
    component "Módulo Social" as SocialModule
    component "Capa de Seguridad / JWT" as Auth
    component "Capa de Acceso a Datos" as DAO

    API --> UserModule
    API --> FoodModule
    API --> RecipeModule
    API --> SocialModule
    API --> Auth
}
```

```

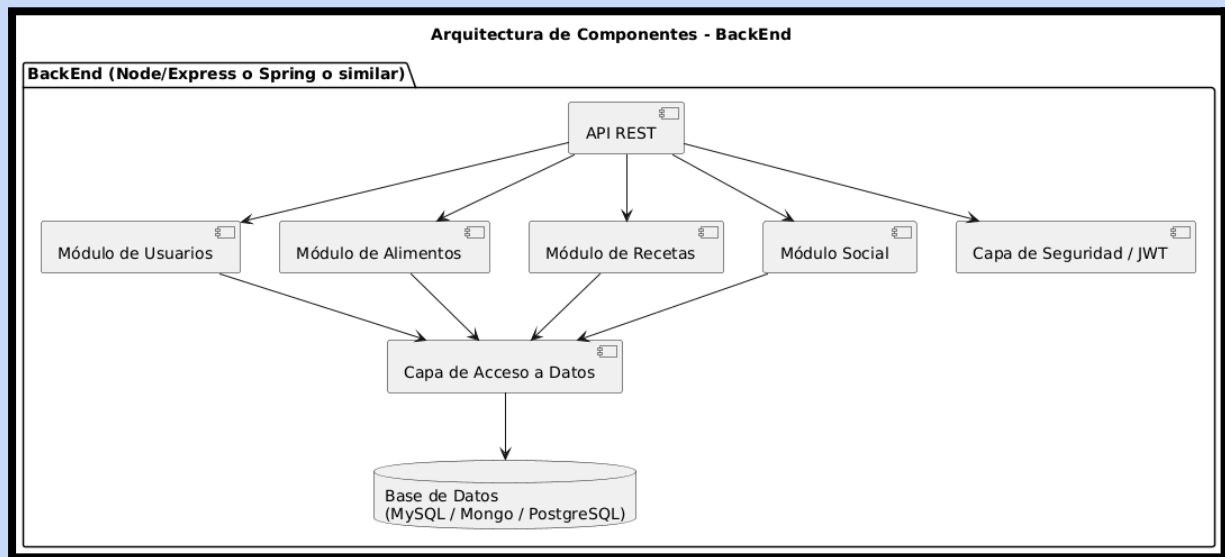
UserModule --> DAO
FoodModule --> DAO
RecipeModule --> DAO
SocialModule --> DAO

database "Base de Datos\n(MySQL / Mongo / PostgreSQL)" as DB
DAO --> DB
}

@enduml

```

IMAGEN



2. Diagrama global Frontend + Backend

Unión de ambos diagramas para ver la estructura final.

CÓDIGO

```

@startuml
title Componentes Globales - FrontEnd & BackEnd

package "FrontEnd (SPA)" {
    component "Pantallas y Módulos" as FE_Modulos
    component "API Client" as FE_API
}

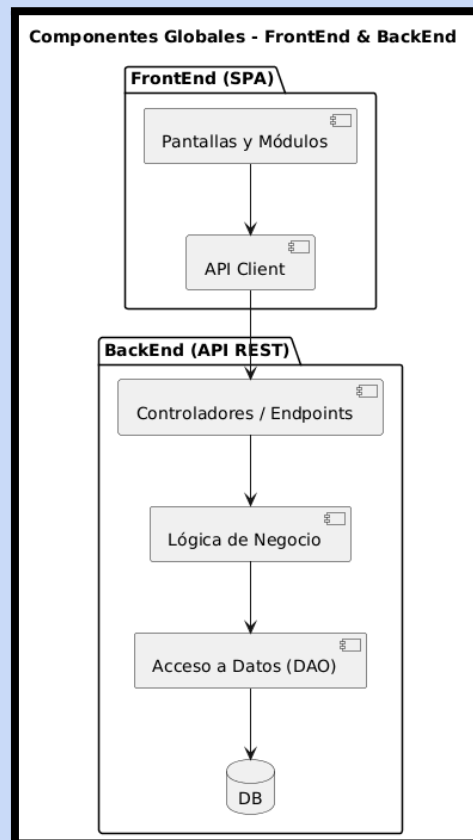
package "BackEnd (API REST)" {
    component "Controladores / Endpoints" as BE_API
    component "Lógica de Negocio" as BE_Service
    component "Acceso a Datos (DAO)" as BE_DAO
    database "DB" as DB
}

```

```
FE_Modulos --> FE_API
FE_API --> BE_API
BE_API --> BE_Service
BE_Service --> BE_DAO
BE_DAO --> DB
```

@endum1

IMAGEN



Actividad 5 – Modelo de Datos de Intercambio en JSON

Se debe definir los formatos de datos JSON que intercambia el sistema.

1. JSON de Receta Guardada / Receta social

CÓDIGO

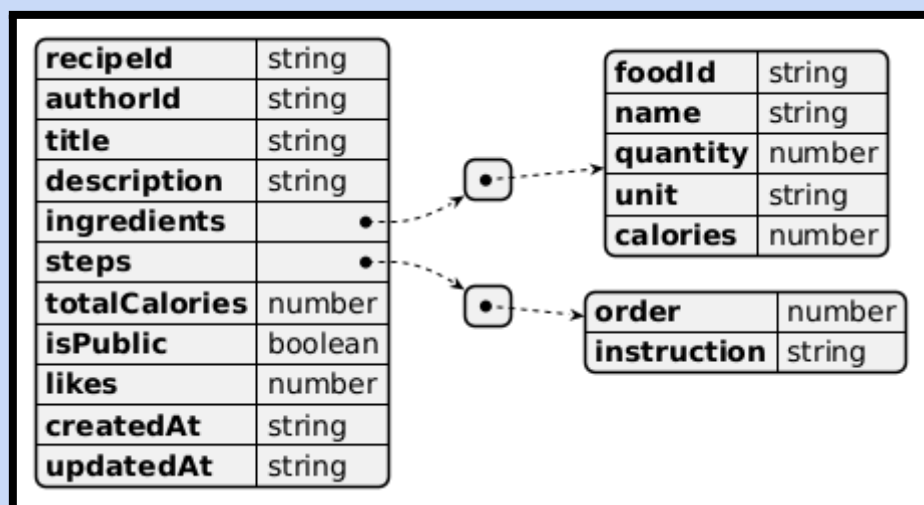
```
@startjson
{
  "recipeId": "string",
  "authorId": "string",
  "title": "string",
  "description": "string",
```

```

"ingredients": [
  {
    "foodId": "string",
    "name": "string",
    "quantity": "number",
    "unit": "string",
    "calories": "number"
  }
],
"steps": [
  {
    "order": "number",
    "instruction": "string"
  }
],
"totalCalories": "number",
"isPublic": "boolean",
"likes": "number",
"createdAt": "string",
"updatedAt": "string"
}
@endjson

```

RECETA



Actividad 6 – Diagrama ER (Entidad–Relación)

Se debe definir el modelo de datos relacional para la aplicación, no sería el modelo final para el proyecto pero me servirá para ir sentando las bases para cuando se empiece a hacer.

CÓDIGO

```
@startuml
title Modelo de Datos (IE Diagram) - Calorie Tracker

entity "Usuario" as user {
    * user_id : UUID
    --
    nombre : string
    email : string
    password_hash : string
    fecha_creacion : datetime
}

entity "Alimento" as food {
    * food_id : UUID
    --
    nombre : string
    calorias_por_100g : number
    unidad_base : string
}

entity "Comida" as meal {
    * meal_id : UUID
    --
    user_id : UUID
    fecha : date
    tipo : string // desayuno, almuerzo, cena, snack
    total_calorias : number
}

entity "ItemComida" as meal_item {
    * meal_item_id : UUID
    --
    meal_id : UUID
    food_id : UUID
    cantidad : number
    unidad : string
    calorias_totales : number
}

entity "Receta" as recipe {
    * recipe_id : UUID
    --
    author_id : UUID
}
```

```

    titulo : string
    descripcion : string
    total_calorias : number
    es_publica : boolean
    fecha_creacion : datetime
}

entity "IngredienteReceta" as recipe_ing {
    * ingredient_id : UUID
    --
    recipe_id : UUID
    food_id : UUID
    cantidad : number
    unidad : string
    calorias_totales : number
}

entity "LikeReceta" as recipe_like {
    * like_id : UUID
    --
    recipe_id : UUID
    user_id : UUID
}

' Relaciones
user ||--o{ meal : "1:N"
meal ||--o{ meal_item : "1:N"
food ||--o{ meal_item : "1:N"

user ||--o{ recipe : "1:N"
recipe ||--o{ recipe_ing : "1:N"
food ||--o{ recipe_ing : "1:N"

user ||--o{ recipe_like : "1:N"
recipe ||--o{ recipe_like : "1:N"

@enduml

```


IMAGEN

