

Documentación del Proyecto: Calorie Tracker (Monitor de Balance Nutricional Diario)

Proyecto para la materia de Programación de Aplicaciones Web

Objetivo Principal: Demostrar la capacidad de construir una aplicación web funcional y escalable utilizando un stack de desarrollo moderno.

I. Descripción del Proyecto: Conceptualización y Alcance

El proyecto **Calorie Tracker** es una aplicación web diseñada para asistir a los usuarios en el monitoreo y análisis de su ingesta nutricional diaria. Más allá de ser un simple contador de calorías, la aplicación funciona como un **Centro Nutricional** que guía al usuario a través de cuatro pilares fundamentales del seguimiento dietético: la personalización de la meta calórica, el registro eficiente de alimentos, el seguimiento de hábitos saludables y el análisis en tiempo real de la composición de macronutrientes.

Contexto y Relevancia de la Aplicación

En el contexto actual, la adherencia a objetivos de salud y fitness es a menudo obstaculizada por la falta de *feedback* inmediato y la complejidad de las herramientas de registro. Esta aplicación aborda directamente este problema al ofrecer una interfaz limpia, dinámica y con actualizaciones instantáneas, transformando el tedioso proceso de conteo calórico en una experiencia clara y motivadora.

Decisiones Clave de Tecnología:

Componente	Tecnología Seleccionada	Justificación Práctica
Frontend Core	React & TypeScript (TSX)	Elegido por su capacidad para manejar estados complejos de manera predecible (React Hooks) y por la seguridad en el desarrollo que ofrece TypeScript, minimizando errores de tipado en la lógica nutricional.

Estilización	Tailwind CSS	Permite un desarrollo rápido de la interfaz, asegurando un diseño totalmente responsivo y facilitando la implementación de estilos dinámicos como el Modo Oscuro/Claro.
Persistencia de Datos	localStorage (Aislado)	Para fines de demostración, se simula una base de datos al guardar los perfiles y logs de manera asíncrona, con una lógica de aislamiento por ID de usuario para mantener la integridad de los datos.

II. Objetivo y Flujo de Usuario

El objetivo del proyecto se centra en la **funcionalidad práctica** y la **claridad de la información**, evitando la sobrecarga de datos.

A. Objetivo de la Aplicación

- Cálculo Personalizado:** Determinar la meta calórica diaria del usuario con una alta precisión utilizando una fórmula validada científicamente.
- Monitoreo Instantáneo:** Proveer *feedback* visual inmediato (barras de progreso y estados) que muestre el grado de adherencia a la meta calórica en tiempo real.
- Análisis Profundo:** Descomponer el consumo de calorías en su composición macro-nutricional (Proteínas, Carbohidratos, Grasas) para permitir al usuario evaluar la calidad y el balance de su dieta.

B. Flujo Detallado del Usuario (La Práctica)

El diseño del flujo se planificó para ser lineal y lógico:

- Ruta /registro (Onboarding):**
 - Si el usuario no tiene un perfil en el navegador, se le dirige aquí.
 - Se utiliza un formulario simple para capturar los datos antropométricos y el nivel de actividad.
 - Al enviar, la aplicación ejecuta la lógica de la **Fórmula Mifflin-St Jeor** para calcular la TMB y el Gasto Energético Total (GET), asignando una meta calórica diaria (`dailyGoal`).
- Ruta /dashboard (Operación Diaria):**
 - Registro:** El usuario utiliza la barra de búsqueda para encontrar y registrar alimentos. Se utiliza una modal para ingresar la cantidad consumida, asegurando que el cálculo sea preciso.
 - Visualización:** Las métricas clave (calorías consumidas, calorías restantes, progreso) se actualizan inmediatamente al guardar un nuevo registro. También se implementó un módulo de seguimiento de Hábitos que se guarda de forma independiente.

3. Ruta /historial (Análisis Nutricional):

- Esta página, renombrada de 'Historial' a '**Centro Nutricional**', fue diseñada para un análisis detallado del día **actual**.
- Muestra el **Macro Split Analyzer**, un gráfico de barras que compara la distribución porcentual de los macronutrientes consumidos (P/C/G) contra un objetivo ideal (40/35/25), permitiendo al usuario tomar decisiones dietéticas informadas.

III. Instrucciones para Correr la Aplicación

La aplicación fue construida bajo un entorno de desarrollo estándar de React.

A. Pre-requisitos del Entorno

Es indispensable tener instalado un entorno de ejecución de JavaScript, como **Node.js** (versión 18 o superior), junto con su administrador de paquetes asociado (**npm**).

B. Procedimiento de Instalación

Abra la terminal en el directorio raíz del proyecto y ejecute los siguientes comandos:

1. Instalar Dependencias:

```
npm install
```

Este comando descarga e instala todas las librerías necesarias definidas en el archivo `package.json` (incluyendo React, TypeScript y Tailwind CSS).

2. Ejecutar el Servidor de Desarrollo:

```
npm run dev
```

3. Esto inicia el servidor local (generalmente en `http://localhost:5173`) y abre la aplicación en el navegador, con **recarga en caliente** (Hot Reloading) para la demostración.

C. Pruebas de Funcionalidad

1. **Simulación de Nuevo Usuario:** La primera carga lo llevará a `/registro`. Complete el perfil para generar su meta calórica.
2. **Prueba de Aislamiento:** Los datos se guardan con una clave única (`dailyLog_[ID_USUARIO]`), demostrando la capacidad de manejar múltiples usuarios si se integrara un sistema de autenticación real.

3. **Prueba del Centro Nutricional:** Registre varios alimentos con diferente composición (ej. una proteína y un carbohidrato) y luego navegue al **Centro Nutricional** para observar cómo se actualiza el porcentaje de distribución de macros.

IV. Explicación de Fórmulas y Lógica Generada

La robustez del proyecto reside en la fidelidad de sus cálculos y la eficiencia de su lógica.

A. Lógica del Cálculo Calórico (TMB y GET)

La lógica se encuentra encapsulada en el *custom hook* `useTmbCalculator.ts` y la fórmula utilizada es la de **Mifflin-St Jeor**, seleccionada por su alta precisión clínica.

Fórmulas Implementadas:

Sexo	Fórmula de TMB
Hombres	$TMB = (10 * P) + (6.25 * A) - (5 * E) + 5$
Mujeres	$TMB = (10 * P) + (6.25 * A) - (5 * E) - 161$

Donde P es el Peso en kg, A es la Altura en cm, y E es la Edad en años.

El cálculo final del Gasto Energético Total (GET), que define la meta calórica diaria (`dailyGoal`), se obtiene multiplicando la TMB por el **Factor de Actividad (AF)**. Esta lógica demuestra la integración exitosa de la teoría nutricional en la funcionalidad del *software*.

B. Lógica del Análisis Macro-Nutricional

El **Centro Nutricional** basa su análisis en la **Regla de Conversión 4-4-9**, estándar en nutrición para convertir gramos de macronutrientes en sus respectivas calorías.

1. **Acumulación de Gramos:** El sistema itera sobre todas las entradas de `dailyLog` para sumar los gramos totales de Proteínas, Carbohidratos y Grasas.
2. **Cálculo Calórico por Macro:** Se aplica la regla de conversión:
 - Calorías de Proteínas = Gramos * 4
 - Calorías de Carbohidratos = Gramos * 4
 - Calorías de Grasas = Gramos * 9
3. **Determinación del Porcentaje:** Se calcula el porcentaje que representa cada macronutriente del total de calorías aportadas por los macros, contrastándolo con la meta ideal establecida, ofreciendo una métrica de **Balance Dietético**.

V. Archivos de requerimientos

Los requerimientos para el desarrollo de este proyecto están disponibles en el siguiente enlace. Dicho enlace detalla cada uno de los requerimientos, incluyendo una descripción de lo solicitado en cada punto.

<https://gamma.app/docs/Proyecto-Contador-de-Calorias-Web-App-zsibvzzdv1gjkwj?authuse=r=0&mode=doc>

VI. Rúbrica de evaluación

Criterio	Descripción	Ponderación	Nivel excelente (100%)	Nivel satisfactorio(70 %)	Nivel básico (40%)
1. Registro y autenticación	Implementa correctamente el registro de usuario y persistencia en localStorage o indexedDB.	10 pts.	Registro funcional y persistente; permite ingresar con datos guardados.	Registro funciona parcialmente, sin persistencia total.	Registro simulado o incompleto.
2. Cálculo de requerimiento calórico diario (TMB / TDEE)	Aplica la fórmula según peso, altura, edad, sexo y actividad física.	15 pts	Fórmula correcta y resultados coherentes; interfaz clara para ingresar datos.	Cálculo funcional pero con errores leves o sin validaciones.	No realiza el cálculo correctamente o está incompleto.
3. Registro de alimentos consumidos	Permite agregar alimentos desde lista predefinida con valor calórico.	15 pts	Funcional, intuitivo y actualiza calorías automáticamente.	Funciona parcialmente o requiere recargar para ver cambios.	No suma o no registra correctamente.

4. Visualización del progreso diario (gráfico o barra)	Muestra gráficamente si el usuario está dentro o fuera de su meta.	10 pts	Barra/círculo dinámico y coherente con los datos.	Gráfico básico o estático.	Solo muestra texto, sin visualización visual.
5. Diseño de interfaz (UI/UX)	Apariencia general, coherencia visual y usabilidad.	10 pts	Diseño profesional, responsive y coherente con tema salud/bienestar.	Diseño aceptable pero sin consistencia visual o con detalles de usabilidad.	Diseño pobre o no responsive.
6. Navegación y estructura de componentes	Correcta implementación de rutas y componentes en React.	10 pts	Uso adecuado de react-router-dom, componentes bien estructurados.	Navegación funciona parcialmente o código desorganizado.	Navegación limitada a una sola vista o sin modularidad.
7. Buenas prácticas en TypeScript y React	Tipado, uso de hooks, limpieza del código y organización.	15 pts	Usa interfaces/types, useState, useEffect, y componentes bien comentados.	Usa TS pero con tipado incompleto o mala organización.	Código sin tipado o desordenado.
8. Funcionalidades extra	Implementa una o más de las mejoras opcionales.	5 pts	Implementa al menos una mejora funcional (modo oscuro, mensajes, PDF, etc.).	Intenta agregar una mejora pero no funcional al 100%.	No incluye ninguna funcionalidad extra.

9. Interacción con el visitante (Feria)	Capacidad del alumno para explicar la app, su funcionamiento y utilidad.	5 pts	Explica con claridad, demuestra dominio técnico y buena actitud con el público.	Explica con dificultad o con apoyo del equipo.	No puede explicar el funcionamiento básico.
10. Presentación del stand	Organización visual, materiales de apoyo y presentación general.	5 pts	Mesa bien presentada, materiales listos (cartel, laptop, app funcional).	Presentación aceptable pero con desorden o sin materiales extra.	Stand desorganizado o sin preparación visual.
Total		100 pts			

VII. MOCKUPS DE PANTALLAS

<https://balsamiq.cloud/sttzrqd/pd3cqee>

¡Crea tu Perfil Calórico!

Nombre:

Edad (años): 25 | Peso (kg): 70

Altura (cm): 175 | Género: Hombre

Nivel de Actividad: Moderado (ejercicio 3-5 días/semana)

Este factor ajusta tu Gasto Energético Total.

Calcular Meta y Continuar

¿Ya tienes una cuenta? [Inicia sesión aquí](#)

Calorie Tracker

Calorías Restantes (Meta: 2595 kcal)

2595 kcal

Faltan  Progreso: 0% del objetivo completado.

💡 Sigue adelante! Te faltan pocas calorías para completar tu objetivo diario.

⚠️ ¡Aún te falta energía! Te faltan "2595 kcal". Puedes optar por:
1 taza de Arroz Cocido (1/2 taza) y 1 plátano tabasco (1/2 pieza).

Distribución de Macronutrientes (Gramos)

Añade alimentos para ver tu distribución de macros.

Totales Consumidos:
Proteínas: 0g

Mis Hábitos Diarios

Aún no has seleccionado hábitos. Haz clic en "Gestional" para empezar.

Registro Diario

Buscar alimento (ej: man)
-- Filtrar por Categoría --

Comidas Registradas Hoy (0)

Aún no has registrado ningún alimento.

Análisis Nutricional del Día

Fecha: 2025-11-23 | Consumo Total: 475 kcal

Distribución de Macronutrientes (vs. Meta Ideal)

Macronutriente	Consumo Real (%)	Meta Ideal (%)
Proteínas	39% (47g)	Meta Ideal: 35%
Carbohidratos	45% (55g)	Meta Ideal: 40%
Gorduras	16% (9g)	Meta Ideal: 25%

↓ Déficit respecto a Ideal

Nota: La distribución ideal utilizada para este análisis es 40% Carbohidratos, 35% Proteínas, 25% Gorduras (Regla 4-4-9).

VIII. DIAGRAMA DE FLUJO



