

GRUPO 10

# FRAMEWORK

DISEÑO AVANZADO DE APLICACIONES WEB

# SVELTE

RAFAELA KARA - CRISTOBAL MUÑOZ - FLAVIO TARSETTI

# CalorieFit

## Ingreso de datos:

- Edad
- Género
- Estatura
- Peso
- Nivel actividad física
- Objetivo

- Cantidad de comidas al día
- Dieta alimentaria

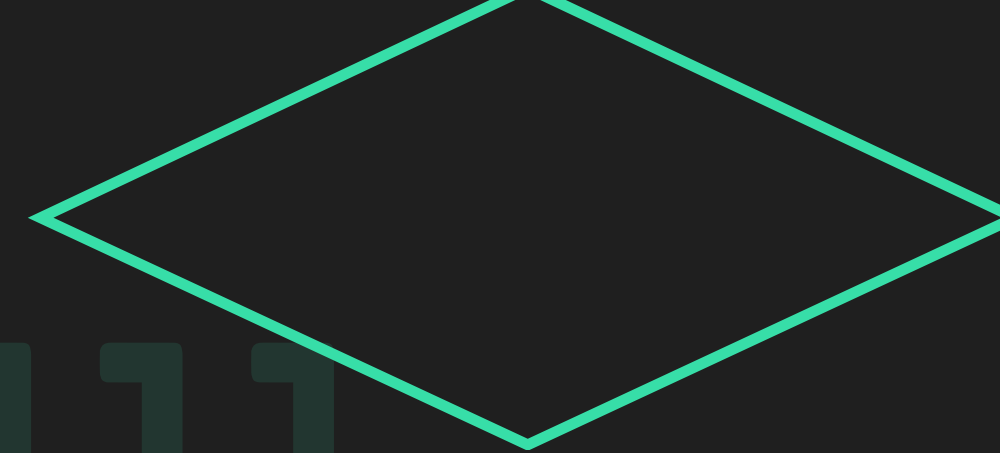
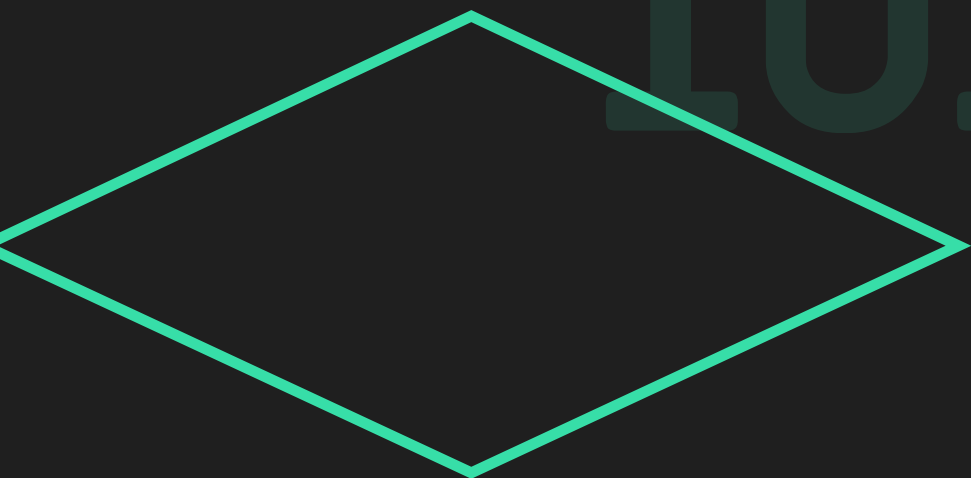
Escoger  
recetas

Crear menú



101001100010111  
000111010111001  
011010101001010  
110110110101010  
101010101001010

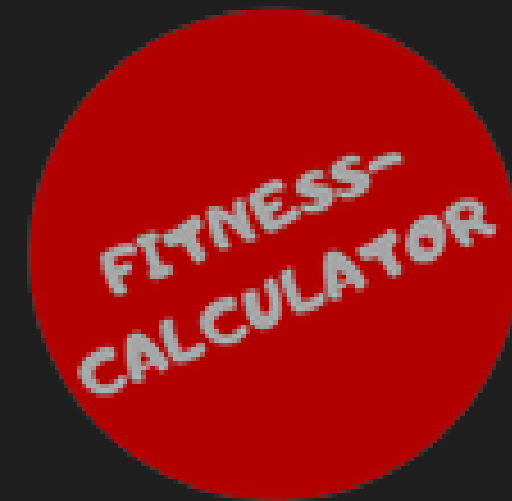
**DEMO**



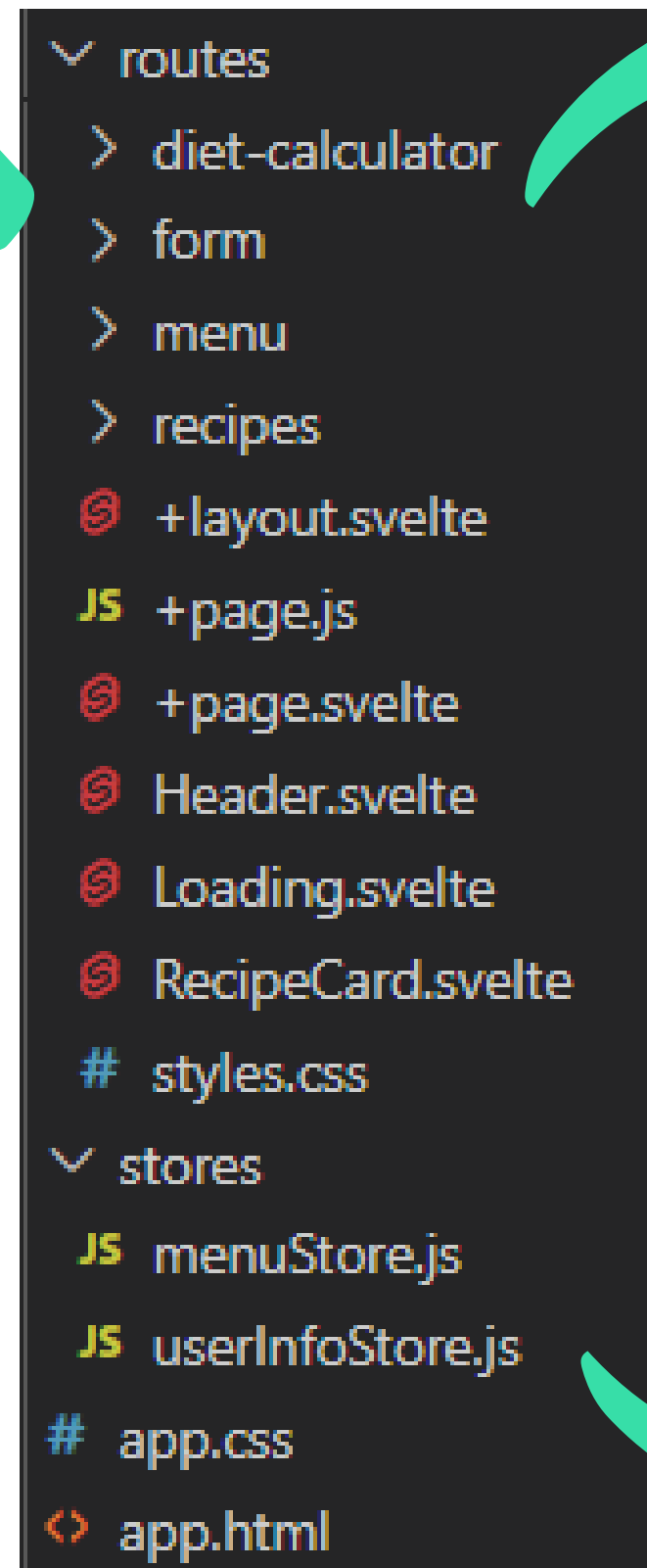
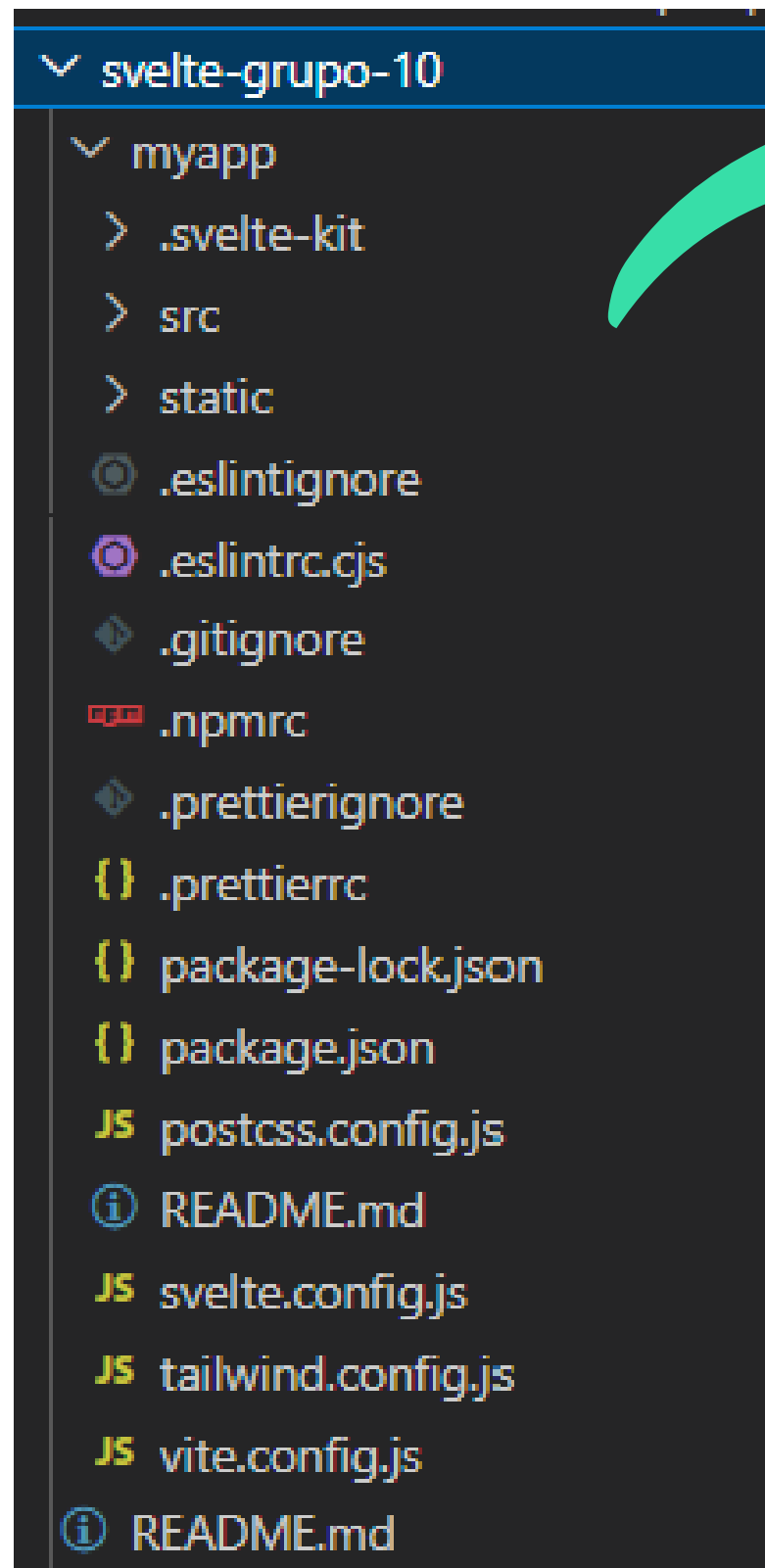


# CARACTERÍSTICAS IMPLEMENTADAS

- ✓ No es un "to do list"
- ✓ Más de una componente
- ✓ Manejo no trivial del estado
- ✓ Responsive
- ✓ Utilización de API externa  
Fitness Calculator → Calorías  
Edamam → Recetas



# ESTRUCTURA ARCHIVOS



## 5 VISTAS

- Home
- Form
- Diet Calculator
- Recipes
- Menu

## 7 COMPONENTES

- Header
- Loading
- UserForm
- DailyCalories
- DietForm
- SelectRecipes
- RecipeCard

## 2 STORES

- menuStore
- userInfoStore

# STORES

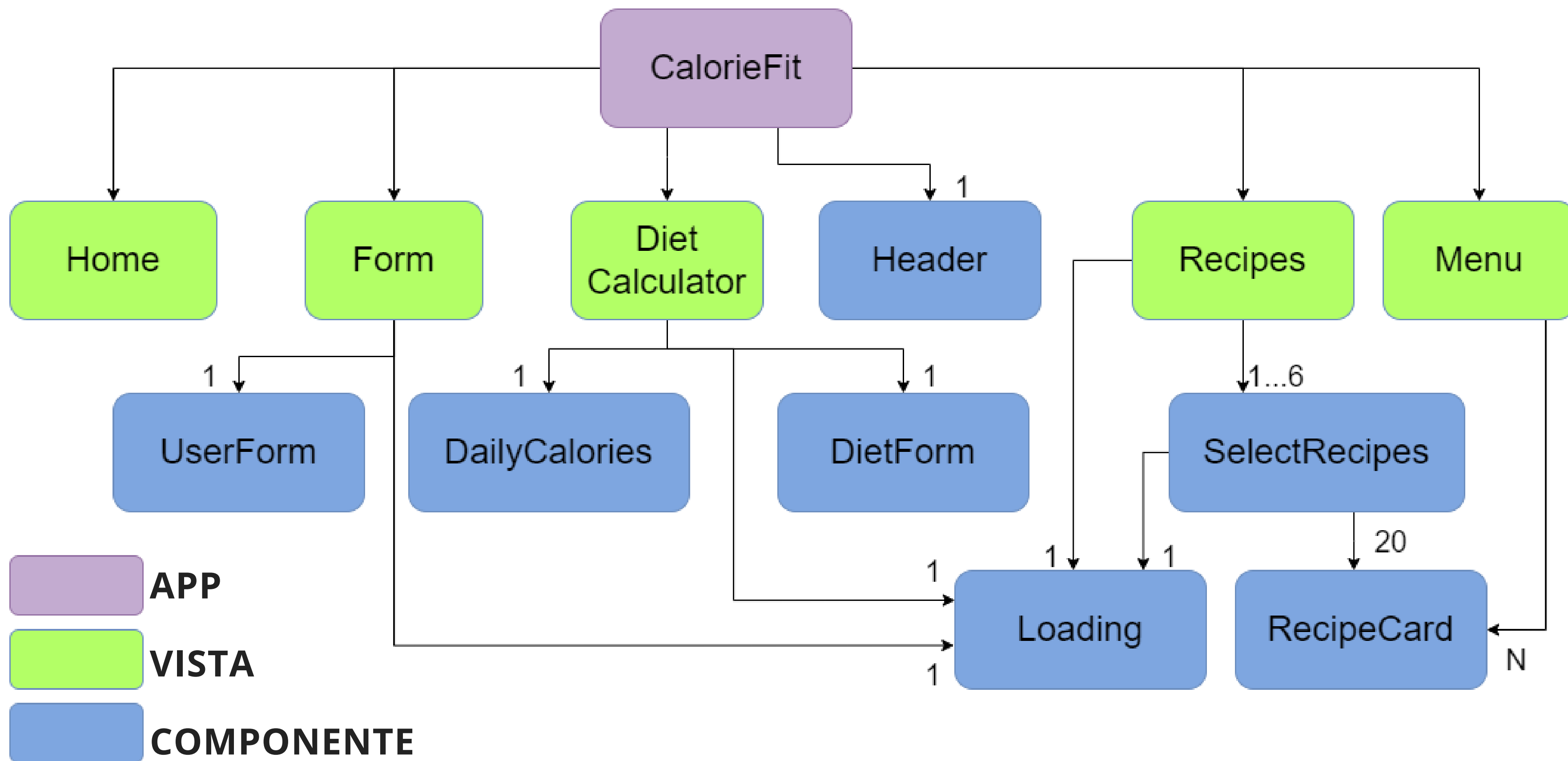
INFORMACIÓN  
DEL USUARIO

INFORMACIÓN DE  
RECETAS Y MENU

```
1 import { writable } from 'svelte/store';
2
3 export const calories = writable(null);
4 export const caloriesLeft = writable(null);
5 export const typeOfMeal = writable([]);
6 export const diets = writable([]);
7 export const mealSelection = writable(null);
```

```
1 import { writable } from 'svelte/store';
2
3 export const likedRecipes = writable([]);
4 export const recipes = writable([
5     { mealType: 'breakfast', recipes: []},
6     { mealType: 'brunch', recipes: []},
7     { mealType: 'lunch', recipes: []},
8     { mealType: 'snack', recipes: []},
9     { mealType: 'teatime', recipes: []},
10    { mealType: 'dinner', recipes: []}
11]);
12
13 export function updateRecipes(type, newRecipes) {
14     recipes.update(items => {
15         return items.map(item => {
16             if (item.mealType === type) {
17                 return { ...item, recipes: newRecipes };
18             }
19             return item;
20         });
21     });
22 }
```

Setter para  
agregar recetas  
aun tipo de comida



# VISTA HOME

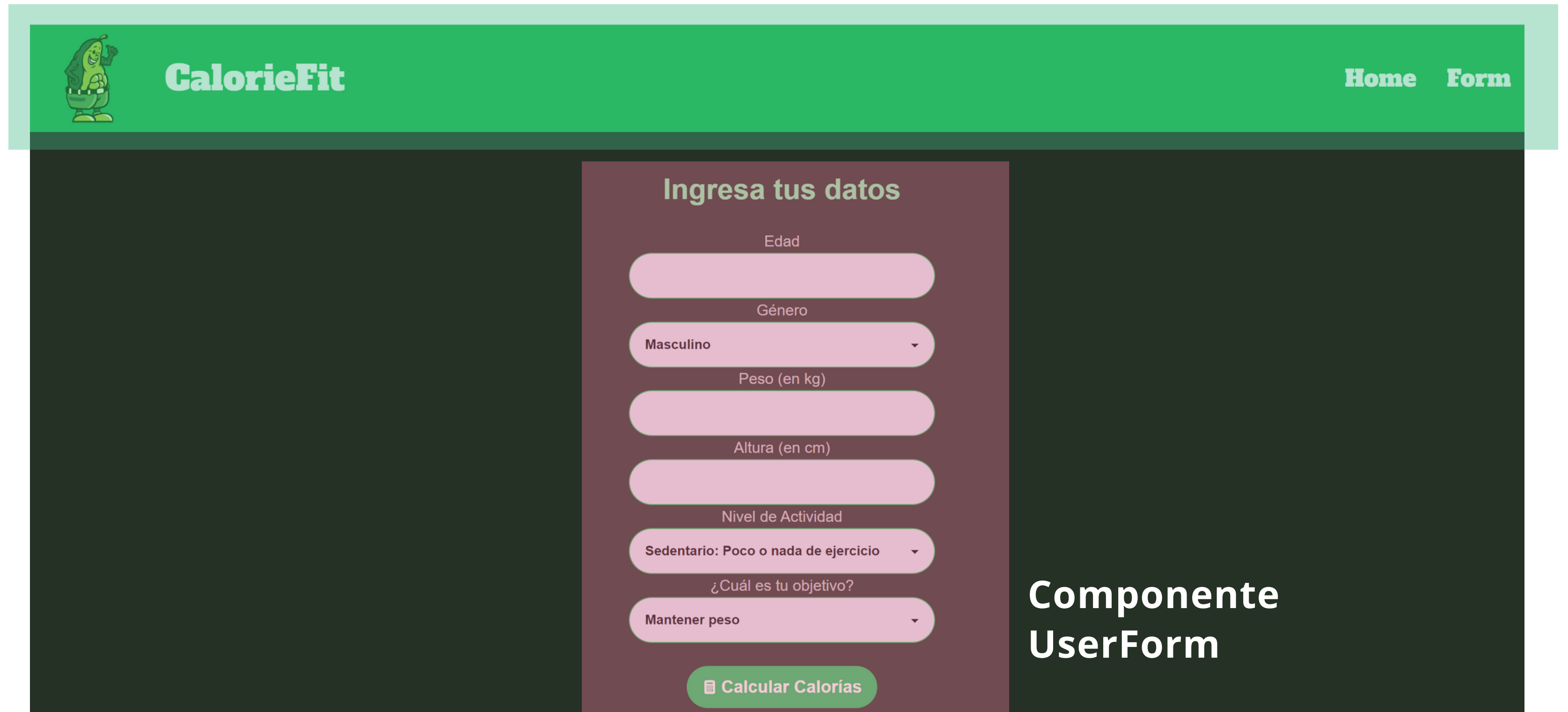
## Componente Header






# VISTA FORM

## Componente Header



The mockup shows a web interface for 'CalorieFit'. The header is green with a cartoon banana character on the left and navigation links 'Home' and 'Form' on the right. The main content area has a dark green background. A central maroon box contains the form titled 'Ingresa tus datos'. The form includes input fields for 'Edad', 'Peso (en kg)', and 'Altura (en cm)'. It also has dropdown menus for 'Género' (set to 'Masculino'), 'Nivel de Actividad' (set to 'Sedentario: Poco o nada de ejercicio'), and '¿Cuál es tu objetivo?' (set to 'Mantener peso'). A green button with a calendar icon and the text 'Calcular Calorías' is at the bottom of the form.

 **CalorieFit**

Home Form

### Ingresa tus datos

Edad

Género

Masculino

Peso (en kg)


Altura (en cm)

Nivel de Actividad

Sedentario: Poco o nada de ejercicio

¿Cuál es tu objetivo?

Mantener peso

 **Calcular Calorías**

## Componente UserForm

# VISTA DIET CALCULATOR

## Componente Header



CalorieFit

Home

Form

## Componente DailyCalories

Consumo diario de  
calorías recomendado

2103 kcal

← Volver

## Ingresa tus preferencias alimentarias

Comidas diarias (obligatorio):

Desayuno

☐

Brunch

☐

Almuerzo

☐

Colación

☐

Once

☐

Cena

☐

Dieta alimentaria (opcional):

Sin productos lácteos

☐

Sin gluten

☐

Kosher

☐

Bajo en azúcar

☐

Pescetariana

☐

Vegetariana

☐

Vegana

☐

Seleccionar Recetas

Componente  
DietForm

# VISTA RECIPES

## Componente Header

**CalorieFit**

HomeForm

DesayunoAlmuerzoOnce

Calorías totales por día: 2103 kcalCalorías restantes: 2103 kcalMenú del día hasta ahoraTerminar Menú

### Pico de Gallo



Calorías totales: 321.53000000000003 kcal

Etiquetas: Low-Fat

Tiempo de preparación: 10 minutos

Cantidad de porciones: 6

Calorías por porción: 53.58833333333334 kcal

Ingredientes: 

AGREGAR RECETA

### Pain Normand




Calorías totales: 1979.95 kcal

Etiquetas: Low Fat

# VISTA RECIPES

## Componente Header




CalorieFit

HomeForm

DesayunoAlmuerzoOnce

Calorías totales por día: 2103 kcalCalorías restantes: 2103 kcalMenú del día hasta ahoraTerminar Menú

Pico de Gallo




Calorías totales: 321.53000000000003 kcal

Etiquetas: Low-Fat

Tiempo de preparación: 10 minutos


Cantidad de porciones: 6

Calorías por porción: 53.58833333333334 kcal

Ingredientes: 

AGREGAR RECETA

Pain Normand



Calorías totales: 1979.95 kcal

Etiquetas: Low Fat

# EJEMPLO DE COMPONENTE

## RecipeCard

### SCRIPT

- 1 Obtención de props
- 2 Importación de stores
- 3 Creación de una variable reactiva
- 4 Implementación de funciones para agregar y eliminar recetas del menú

```
1 <script>
2   export let recipe;
3   export let hideButtons = false;
4   import { caloriesLeft, mealSelection } from '../stores/userInfoStore.js'
5   import { likedRecipes } from '../stores/menuStore.js'
6   const { label, image, calories, dietLabels, totalTime, ingredientLines } = recipe;
7
8   let hasBeenAdded = $likedRecipes.findIndex((elem) => elem.label = label) !== -1
9
10  3 $: showDeleteButton = hasBeenAdded && $likedRecipes.length > 0
11
12  function addRecipe() {
13    $caloriesLeft -= calories/recipe.yield
14    const meal = {
15      ...
16    }
17    likedRecipes.update(items => [...items, meal]);
18    hasBeenAdded = true
19  }
20
21  function removeRecipe() {
22    $caloriesLeft += calories/recipe.yield
23    const index = $likedRecipes.findIndex((elem) => elem.label = label)
24    $likedRecipes.splice(index, 1)
25    hasBeenAdded = $likedRecipes.some(elem => elem.label = label)
26  }
27 </script>
```



# EJEMPLO DE COMPONENTE

## RecipeCard

### HTML

#### Bloques y Bindings

- 1 `{#if} {:else} {/if}` Determinar qué botones desplegar
- 2 `{#each} {/each}` Mostrar cada ingrediente
- 3 `on:click` Agregar/Eliminar recetas del men

### OTROS

- `bind:value`
- `on:mouseover`
- `on:mouseleave`
- `{#await} {:then} {:catch} {/await}`

```
1 <div>
2   <h2>{label}</h2>
3   <figure></figure>
4 </div>
5 <div>
6   <div>
7     <p>Tiempo de preparación: </p>
8     <p>{totalTime} minutos</p>
9   </div>
10  <div>
11    <p>Calorías por porción: </p>
12    <p>{calories/recipe.yield} kcal</p>
13  </div>
14  <details>
15    <summary>Ingredientes:</summary>
16    {#each ingredientLines as ingredient}
17      <p key={ingredient}>{ingredient}</p>
18    {/each}
19  </details>
20  {#if !hideButtons}
21    {#if showDeleteButton}
22      <button on:click={removeRecipe}>Eliminar Receta</button>
23    {/if}
24    {#if $caloriesLeft - calories/recipe.yield > 0}
25      <button on:click={addRecipe}>Agregar Receta</button>
26    {#else}
27      <button disabled>Agregar Receta</button>
28    {/if}
29  {/if}
30 </div>
```

# DIFICULTADES

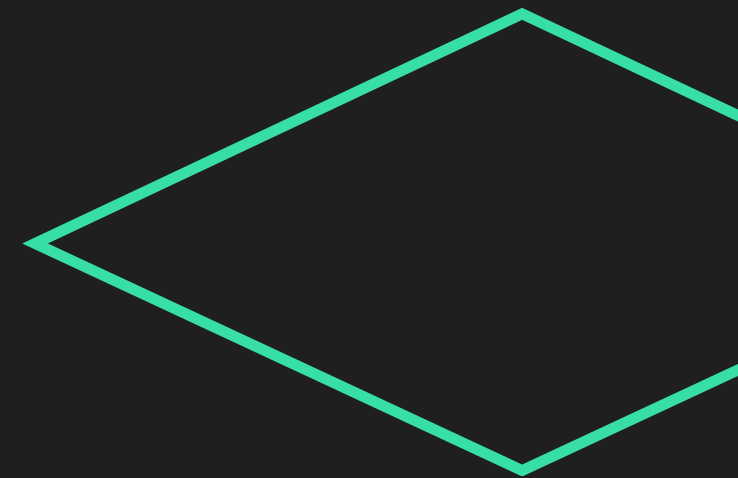


- Funcionamiento de variables reactivas
- Responsive App
- Funcionamiento de `onMount()`

# CONCLUSIONES



- Similar con Vue
- Uso de bloques y bindings simplifica el código
- "Reactividad" puede ser muy útil si se sabe usar bien





**MUCHAS  
GRACIAS**

101001100010111  
000111010111001  
01101010101010  
1101101101010  
101010101001010