

GRUPO 10

FRAMEWORK

DISEÑO AVANZADO DE APLICACIONES WEB

VUE

RAFAELA KARA - CRISTOBAL MUÑOZ - FLAVIO TARSETTI

CalorieFit

Ingreso de datos:

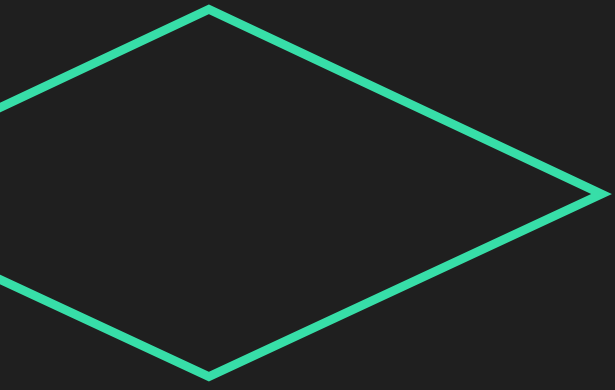
- Edad
- Género
- Estatura
- Peso
- Nivel actividad física
- Objetivo

- Cantidad de comidas al día
- Dieta alimentaria

Escoger
recetas

Crear menú

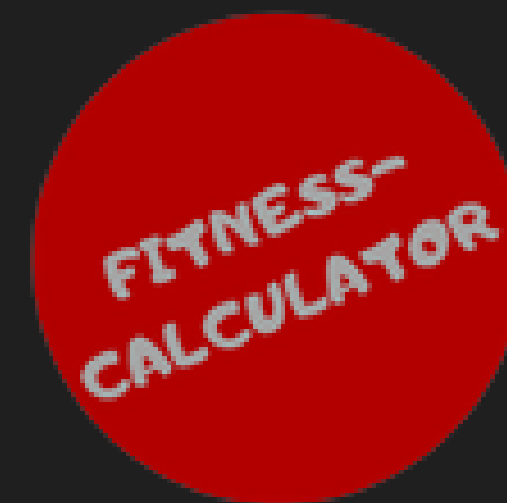




CARACTERÍSTICAS IMPLEMENTADAS



- ✓ No es un "to do list"
- ✓ Más de una componente
- ✓ Manejo no trivial del estado
- ✓ Utilización de API externa



101001100010111
000111010111001
011010101001010
110110110101010
101010101001010

DEMO

MANEJO DE ESTADO

- 1 Creamos un store para guardar las recetas elegidas
- 2 Definimos una acción para agregar recetas
- 3 Definimos una acción para eliminar recetas
- 4 Construimos un getter para agrupar las recetas según el tipo de comida

- A Obtenemos tipos de comida
- B Generamos grupos de recetas para la vista final



```
1 export const useMenuStore = defineStore('menu', {
2   state: (): State => {
3     return {
4       likedRecipes: []
5     }
6   },
7   actions: {
8     addLikedRecipe(recipe: Meal) {
9       this.likedRecipes.push(recipe)
10    },
11    removeLikedRecipe(recipe: Meal) {
12      const index = this.likedRecipes.findIndex((liked) => liked.name === recipe.name);
13      this.likedRecipes.splice(index, 1)
14    }
15  },
16  getters: {
17    4 getLikedRecipesByType: (state) => {
18      const nameDict: { [key:string]: string } = {
19        breakfast: "Desayuno", lunch: "Almuerzo", dinner: "Cena"
20      }
21      const likedTypes = state.likedRecipes.map((recipe) => recipe.typeOfMeal)
22      const uniqueTypes = [...new Set(likedTypes)]
23      const likedRecipesByType = uniqueTypes.map((type) => {
24        return {
25          25 mealType: type,
26          name: nameDict[type],
27          recipes: state.likedRecipes.filter((recipe) => recipe.typeOfMeal === type)
28        }
29      })
30      return likedRecipesByType;
31    }
32  })
33 })
```

COMPONENTES

TEMPLATE

```
1 <template>
2   <div v-for="recipe in recipes" v-bind:key="recipe.label" > 1
3     <h3>{{ recipe.label }}</h3>
4     <div>
5       
6       <div>
7         <p>Calorías: {{ recipe.calories }} kcal</p>
8         <p>Tiempo de preparación: {{ recipe.totalTime }} minutos</p> 2
9         <div>Ingredientes:
10           <p v-for="ingredient in recipe.ingredientLines" v-bind:key="ingredient">{{ ingredient }}</p>
11         </div>
12         4 <button v-if="userInfoStore.caloriesLeft - recipe.calories > 0" v-on:click="addRecipe(recipe)">Agregar</button>
13         <button v-else disabled>Agregar</button> 3
14       </div>
15     </div>
16   </div>
17 </template>
```

1

v-for

Iterar sobre conjunto de recetas

2

v-bind

Asignación dinámica de llaves

3

v-on

Agregar recetas al apretar botón

4

v-if v-else

Mostrar botón sólo si quedan calorías

COMPONENTES

SCRIPT

1 **data()** Llamar a los stores
Inicializar variables

2 **mounted()** Llamada a la API
de Edamam

3 **methods** Agregar recetas
Eliminar recetas
Actualizar calorías
Determinar otras
condiciones

```
1 <script>
2   import { useUserInfoStore } from '../stores/userInfo'
3   import { useMenuStore } from '../stores/menu'
4
5   export default {
6     1 data() {
7       return {
8         userInfoStore: useUserInfoStore(),
9         menuStore: useMenuStore(),
10        recipes: []
11      }
12    },
13    2 mounted() {
14      const options = {method: 'GET'}
15      fetch(urñ, options)
16        .then(response => response.json())
17        .then(data => this.recipes = data.hits)
18    },
19    3 methods: {
20      addRecipe(recipe) {
21        this.userInfoStore.caloriesLeft -= recipe.calories/recipe.yield
22        const meal = {
23          ...
24        }
25        this.menuStore.addLikedRecipe(meal)
26      },
27      hasBeenLiked(recipe) {
28        return this.menuStore.likedRecipes.some(elem => elem.name === recipe.label)
29      },
30      removeRecipe(recipe) {
31        this.userInfoStore.caloriesLeft += recipe.calories/recipe.yield
32        const meal = {
33          ...
34        }
35        this.menuStore.removeLikedRecipe(meal)
36      }
37    };
38  </script>
```




DIFICULTADES

- Originalidad
- Documentación API Recipes



CONCLUSIONES



- No hubieron muchas dificultades con Vue
- Es fácil de empezar a usar y bastante intuitivo :)
- Conociendo Vue ahora lo pensaremos bien antes de llegar y usar React :o



**MUCHAS
GRACIAS**

101001100010111
000111010111001
01101010101010
1101101101010
101010101001010