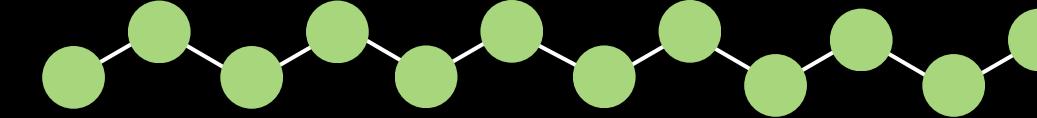


TAREA 6

Diseño avanzado de aplicaciones web



Grupo 3

GOURMET GUIDE

- Tu nueva app favorita para expandir tus habilidades culinarias!
- Busca entre miles de recetas.
- Guarda tus favoritas.
- ¿Vegetariano? ¿Vegana? Selecciona tus recetas en base a tus necesidades.
- ¿Te gusta una receta? Te recomendamos similares
- Usamos Spoonacular API



Recipes

[GET Search Recipes](#)[GET Search Recipes \(Deprecated\)](#)[GET Search Recipes Complex \(Deprecated\)](#)[GET Search Recipes by Nutrients](#)[GET Search Recipes by Ingredients](#)[GET Get Recipe Information](#)[GET Get Recipe Information Bulk](#)[GET Get Similar Recipes](#)[GET Get Random Recipes](#)[GET Autocomplete Recipe Search](#)[GET Taste by ID](#)[GET Equipment by ID](#)[GET Price Breakdown by ID](#)[GET Ingredients by ID](#)[GET Nutrition by ID](#)[GET Get Analyzed Recipe Instructions](#)[GET Extract Recipe from Website](#)

Search through thousands of recipes using advanced filtering and ranking. NOTE: Since this method combines searching by query, by ingredients, and by nutrients into one endpoint, each request counts as 3 requests.

B Personal Account
Benito Andres Tondreau vega

RapidAPI App default-application_7679786 REQUIRED

Request URL rapidapi.com REQUIRED

Header Parameters

X-RapidAPI-Key d1bacda54dmshd0f8781440b0905p174b5djsnd8c ENUM REQUIRED

X-RapidAPI-Host spoonacular-recipe-food-nutrition-v1.p.rapidapi.com STRING REQUIRED

(Node.js) Axios ▾

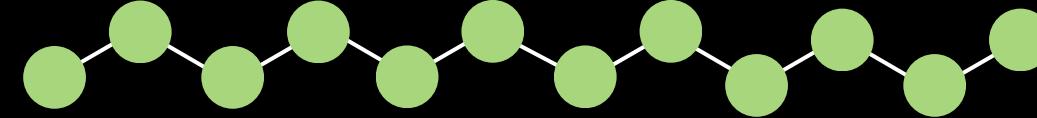
[Copy Code](#)

```
const axios = require('axios');

const options = {
  method: 'GET',
  url: 'https://spoonacular-recipe-food-nutrition-v1.p.rapidapi.co
m/recipes/complexSearch',
  params: {
    query: 'pasta',
    cuisine: 'italian',
    excludeCuisine: 'greek',
    diet: 'vegetarian',
    intolerances: 'gluten',
    equipment: 'pan',
    includeIngredients: 'tomato,cheese',
    excludeIngredients: 'eggs',
    type: 'main course',
    instructionsRequired: 'true',
    fillIngredients: 'false',
    addRecipeInformation: 'false',
    titleMatch: 'Crock Pot',
    maxReadyTime: '20',
    ignorePantry: 'true',
    sort: 'calories',
    sortDirection: 'asc',
    minCarbs: '10',
    maxCarbs: '100',
    minProtein: '10',
    maxProtein: '100',
    minCalories: '50',
    maxCalories: '800',
    minFat: '10',
    maxFat: '100',
    minAlcohol: '0',
    maxAlcohol: '100',
    minCaffeine: '0'.
```



DEMO



Grupo 3



Navbar



We have all the recipes you need!

Search recipes...

SearchBar



Kale and Quinoa Salad
with Black Beans

By Fat Free Vegan

⌚ 50 minutes

Vegetarian Vegan
Glutenfree Dairyfree



Citrus Sesame Kale

By Oh My Veggies

⌚ 15 minutes

Vegetarian Vegan
Glutenfree Dairyfree



Mexican Salad with
Lime Dressing

By Natashas Kitchen

⌚ 15 minutes

Vegetarian Vegan
Glutenfree Dairyfree



Quinoa Kale Tomato
Corn Salad

By Jeanettes Healthy Living

⌚ 15 minutes

Vegetarian Vegan
Glutenfree Dairyfree



Matcha Green Tea and
Pineapple Smoothie

By Feasting at Home

⌚ 10 minutes



Low Carb Frosty

By Buns in My Oven

⌚ 5 minutes

Glutenfree



Pesto Pasta with
Lemon, Spinach,
Edamame & Toasted



Sweet Potato and Black
Bean Mexican Salad

By Two Peas and Their Pod

⌚ 50 minutes

RecipeCard

RecipeTags



Vegetarian Vegan Glutenfree Dairyfree

Cinnamon toast crunch roasted chickpeas

⌚ 5 minutes 🙀 4 servings

RecipeTags

Ingredients

1 Tbsp cinnamon

425.243 g chickpeas

2 Tbsps sugar

1 Tbsp tahini

Instructions

1. Preheat oven to 425 degrees and grease a baking sheet.
2. Combine sugar and cinnamon in a small bowl until fully combined. Toss dried beans with tahini first to coat and then sprinkle cinnamon sugar and mix until all are coated.
3. Spread out onto baking sheet and roast for 15-20 minutes, stirring every 5 minutes so they don't burn on any one side.

You might be interested

Cinnamon Toast Crunch® French Toast Fingers
⌚ 25 minutes

Cinnamon Toast Crunch French Toast
⌚ 15 minutes

Cinnamon Toast Crunch Cookies
⌚ 20 minutes

Cinnamon Toast Crunch® Cocktails
⌚ 10 minutes

Similar Recipes

ESTRUCTURA DE TRABAJO

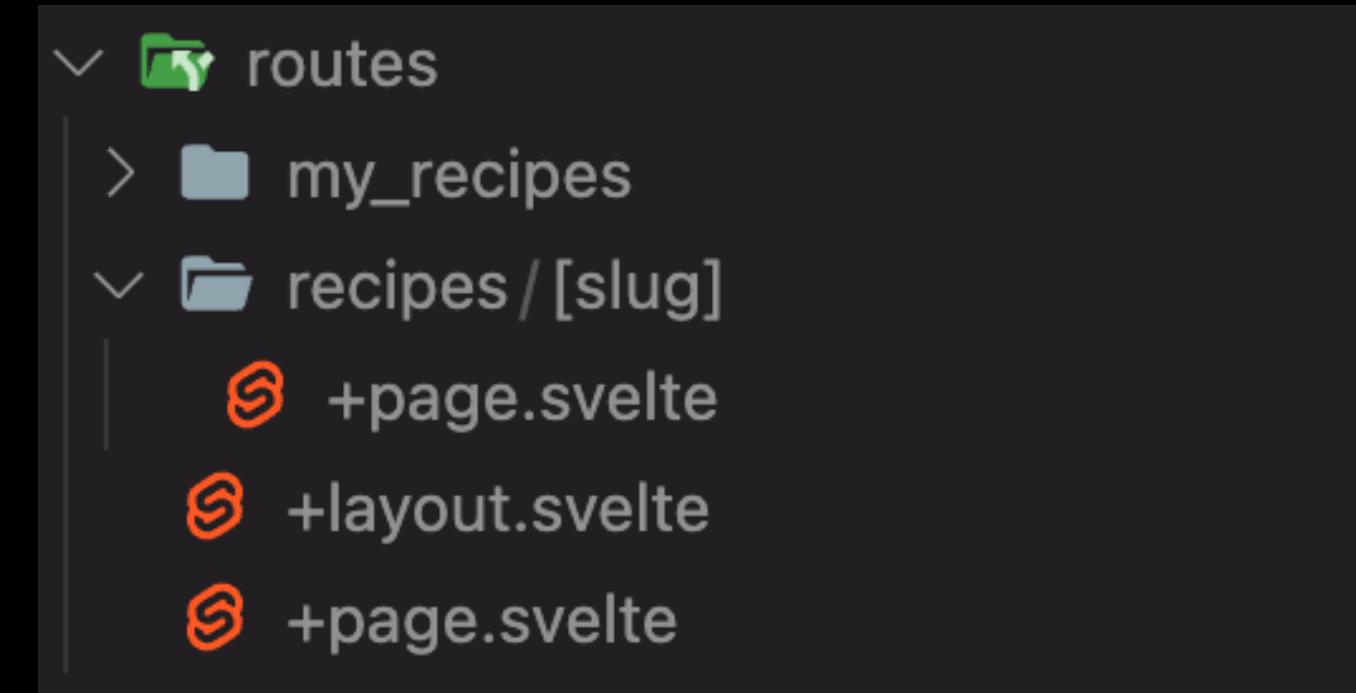
```
✓ SVELTE-GRUPO-03
  > 📁 .idea
  ✓ 📁 src
    ✓ 📁 lib
    ✓ 📁 components
      ✓ 📁 common
        S LoadingIcon.svelte
        S Navbar.svelte
        S Pagination.svelte
        S RecipeCard.svelte
        S RecipeTags.svelte
        S SearchBar.svelte
        S SimilarRecipes.svelte
      ✓ 📁 stores
        TS savedRecipes.ts
        TS useFetchQuery.ts
      ✓ 📁 types
        TS analyzedInstructions.ts
        TS ingredient.ts
        TS recipe.ts
        TS dataProvider.ts
        TS env.ts
        TS httpClient.ts
        TS queries.ts
    } Componentes
    } Stores
    } Types
    } Manejo API's
  } Rutas/Vistas
```

The image shows a file explorer interface with a dark theme. On the left, the project structure for 'SVELTE-GRUPO-03' is displayed. It includes a '.idea' folder, a 'src' folder containing 'lib', 'components' (which has a 'common' subfolder with several Svelte components), 'stores' (with TypeScript files for 'savedRecipes' and 'useFetchQuery'), and 'types' (with TypeScript files for 'analyzedInstructions', 'ingredient', 'recipe', 'dataProvider', 'env', 'httpClient', and 'queries'). Brackets on the right group these into 'Componentes', 'Stores', 'Types', and 'Manejo API's'. On the right side of the interface, there is a list of files and folders: 'routes' (containing 'my_recipes' and 'recipes/[slug]' with various Svelte files like '+page.svelte', '+layout.svelte', and '+page.svelte'), 'static' (containing configuration files like '.eslintignore', '.eslintrc.cjs', '.gitignore', '.npmrc', '.prettierignore', '.prettierrc', and build scripts like 'package-lock.json', 'package.json', 'postcss.config.js', 'README.md', 'svelte.config.js', 'tailwind.config.js', 'tsconfig.json', and 'vite.config.ts'). Brackets on the right group these into 'Rutas/Vistas'.

Rutas/Vistas

ROUTING

- `src/routes` es el root
- `src/routes/recipes` genera la ruta `/recipes`
- `src/routes/recipes/[slug]` genera una ruta con un parámetro.



Slug es usado para cargar información de forma dinámica mediante la store \$page



```
1  $: recipeId = $page.params.slug;
```

STORES

- Es un objeto con un método `subscribe()` que permite a todos los componentes interesados ser notificados cuando el `store value` cambie.
- Mantiene la consistencia entre componentes
- Svelte provee funciones para crear `readable`, `writable` y `derived stores`.



```
1 import { writable } from 'svelte/store';
2 import type Recipe from '../types/recipe';
3
4 export default writable<Partial<Recipe>[]>([]);
```

STORES

Svelte permite la auto suscripción a una store, mediante el prefijo \$

De esta manera podemos acceder al valor de una store y evitar tener que suscribirnos y desuscribirnos a ella en cada componente.

```
● ● ●  
1 import savedRecipes from '../stores/savedRecipes';  
2  
3 $: saved = $savedRecipes?.includes(recipe);
```

BINDINGS

Svelte nos permite pasar información **de un componente hijo al padre**, actualizando el valor de la variable relacionada



```
1 <SearchBar bind:search={searchText} />
```



```
1 $: searchParams = {  
2   query: searchText,  
3   addRecipeInformation: true,  
4   number: number,  
5   offset: (curPage - 1) * number,  
6 }  
7
```



```
1 <script lang="ts">  
2   export let search = '';  
3 </script>
```

earchBar.svelte



SVELTE - QUERY

- Svelte no cuenta con una forma específica de realizar **fetch** o actualizar data proveniente de servicios externos.
- Esto provoca utilizar métodos tradicionales para el manejo de estados que no funciona muy bien para manejar el **estado del request**.
- Beneficios de Svelte Query:
 - Manejo de data cache tras un request.
 - Mayor performance optimizando paginación y datos de carga lenta.
 - Actualización de data desactualizada en el background.



SVELTE-QUERY



```
1 const searchRecipesQuery = useQuery(
2   makeResourceQuery<Recipe>('recipes/complexSearch', searchParams)
3 );
4 $: searchRecipesQuery.setOptions(
5   makeResourceQuery<Recipe>('recipes/complexSearch', searchParams)
6 );
7 $: searchRecipes = $searchRecipesQuery.data?.results || [];
```

Performance

Ecosistema



No utiliza DOM virtual. Reconocido por su gran performance

Al ser bastante nuevo, no posee un ecosistema tan grande como otros frameworks

V
S



Aunque posee manejo de DOM eficiente, no es tanto como Svelte

Posee un gran ecosistema con múltiples librerías y herramientas disponibles



SVELTE

Comunidad

Comunidad más reducida debido a ser un framework relativamente nuevo

Tamaño

Aplicaciones son más livianas debido a que no utilizan el virtual Dom

V
S

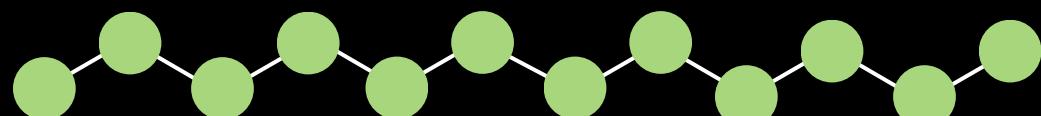


Gran comunidad activa que permite encontrar valiosos recursos y guías de buenas prácticas

Aplicaciones de mayor tamaño que es Svelte. Puede afectar tiempos de carga especialmente en conexiones de red lentas

APRENDIZAJES

- Svelte es un framework minimalista y con una destacable velocidad frente a sus competidores.
- Todo en Svelte es reactivo, lo cual es una gran ventaja a la hora de programar.
- Svelte posee fortalezas y debilidades frente a otros frameworks, por lo que es importante estudiar y analizar los requerimientos y preferencias del proyecto a realizar.





Grupo 3



```
1 export const makeResourceQuery = <ResType extends AnyResource = AnyResource>(
2   resource: string,
3   queryParams?: any
4 ): UseQueryOptions<ListResponse<ResType>> => ({
5   queryKey: [resource, queryParams],
6   queryFn: () =>dataProvider.getList<ResType>(resource, queryParams),
7 })
```