

## Actividad evaluable UT4

### I. Descripción:

Página web, cuya funcionalidad está programada en los lenguajes JavaScript / TypeScript, que recupere datos de un servidor. Su funcionalidad es mostrar categorías de platos, para luego mostrar una lista de platos agrupados según su categoría y, al elegir una, mostrar la receta.

### ii. Métodos y tecnologías

Esta página recupera datos del servidor a través de las tecnologías fetch (síncrono y asíncrono) y Axios.

#### A. Versión General (fetch síncrono):

Estructura html / css:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
    body{
      margin: 8px 4px;
      background-color: black;
      color: white;
    }
    #principal{
      width: 100%;
      display: flex;
      flex-direction: row;
      justify-content: center;
      flex-wrap: wrap;
    }
    #secundario{
      width: 100%;
      display: flex;
      flex-direction: row;
      justify-content: center;
      flex-wrap: wrap;
    }
    #comidas{
      width: 30%;
    }
  </style>
</head>
<body>
  <h1>TusRecetasCaseras</h1>
  <p>Elige una categoria: </p>

  <div id="principal">
  </div>
  <br><br>
  <div id="secundario">
    <div id="comidas"></div>
    <div id="receta"></div>
  </div>

  <script src="e1.js"></script>

</body>
</html>
```

A través del método fetch recuperaremos los datos de la url listCategories (API),

```
let listCategories = "https://www.themealdb.com/api/json/v1/1/categories.php";//obtendremos nombre de categoria

//metodo inicial que a partir de la api llama a la funcion mostrarCategorias
function mostrarDatos() {
  fetch(listCategories)
    .then(response => response.json())
    .then(json => mostrarCategorias(json))
    .catch(error => console.log(error));
}
mostrarDatos();
```

Los datos recuperados los pasaremos como parámetro a la función `mostrarCategorias()`, la cual se encarga de 'pintar' en el html un div por cada categoría añadiendo también un manejador de eventos a cada una, el cual se encarga de llamar a otro método `fetch` para así mostrar mostrar datos de la API.

```
function mostrarCategorias(json) {
  console.log(json);

  let body = "";
  let ac = json.categories; //array de categorias

  //agregamos todas las categorias
  for(let i = 0; i < ac.length; i++) {
    body += `<div id="${ac[i].strCategory}"
      style="width: 150px; height=100px; cursor: pointer; text-align: center; border: 3px solid #73AD21; border-radius: 12px;
      <h3>${ac[i].strCategory}</h3>
      
    </div>`;
  }
  document.getElementById("principal").innerHTML = body;

  //agregamos eventos a cada categoria
  for(let i = 0; i < ac.length; i++) {
    const e = document.getElementById(`${ac[i].strCategory}`);
    e.addEventListener("click", function() {fetchComidas(ac[i].strCategory)});
  }

  console.log(body);
}
```

Lo mismo haremos con la función que llama a `fetch` para recuperar la lista de platos de una categoría específica. Esta función pasa el json de la promesa recibida a otra función para pintar los datos en la página.

```
//a traves de la url, muestra todos los platillos a partir de la categoria
function fetchComidas(categoria) {
  let urlCat = `https://www.themeadb.com/api/json/v1/1/filter.php?c=${categoria}`;
  fetch(urlCat)
    .then(response => response.json())
    .then(json => mostrarRecetas(json))
    .catch(error => console.log(error));
}
```

Añade también manejador de eventos por cada uno de los platos.

```
//muestra la lista de platillos por categoria
function mostrarRecetas(json) {
  console.log(json);

  document.getElementById("receta").innerHTML = "";

  let body = "";
  let am = json.meals; //array de comidas
  for(let i = 0; i < am.length; i++) {
    body += `<h4 style="margin: 10px; cursor: pointer;" id="${am[i].strMeal}">${am[i].strMeal}</h4>`;
  }

  document.getElementById("comidas").innerHTML = body;

  for(let i = 0; i < am.length; i++) {
    document.getElementById(`${am[i].strMeal}`).addEventListener("click", function() {fetchReceta(am[i].idMeal)});
  }
  console.log(body);
}
```

Lo mismo haremos para mostrar una receta específica:

```
//a traves de la url recupera datos sobre el platillo
function fetchReceta(idComida){
  let urlReceta = `https://www.themealdb.com/api/json/v1/1/lookup.php?i=${idComida}`;
  fetch(urlReceta)
    .then(response => response.json())
    .then(json => mostrarReceta(json))
    .catch(error => console.log(error));
}
```

```
//muestra los datos del platillo seleccionado
function mostrarReceta(json) {
  console.log(json);

  let body = "";
  let dReceta = json.meals[0]; //datos de receta

  body += `<h2 style="color: gray">${dReceta.strMeal}</h2>
    
    <h5 style="color: yellow; padding: 10px, 0px;">How to prepare?</h5>
    <p style="padding: 10px, 0px; width: 800px; height: 300px; text-align: justify;">${dReceta.strInstructions}</p>`;

  document.getElementById("receta").innerHTML = body;
  console.log("-----");
  console.log(body);
}
```

## B. Version 2: fetch asincrono

Para esta versión, simplemente cambiaremos parte de la sintaxis en los métodos que llaman al método fetch, para que esta se realice de manera asíncrona a través de métodos async y esperando promesas con la palabra clave await.

```
let listCategories = "https://www.themealdb.com/api/json/v1/1/categories.php"; //obtendremos nombre de categoria

//metodo inicial que a partir de la api llama a la funcion mostrarCategorias
const mostrarDatos = async () => {
  try {
    const response = await fetch(listCategories);
    if (!response.ok) { throw Error(); }
    const json = await response.json();
    mostrarCategorias(json);
  } catch (error) {
    console.log(error);
  }
}

mostrarDatos();
```

### C. Versión 3: Axios

Para esta tercera versión, tenemos que añadir la etiqueta `<script>` dos veces en el html:

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script src="e1.js"></script>
```

Simplemente cambiaremos parte de la sintaxis, como se muestra a continuación:

```
// https://thymealdb.com/api.php
//todas las categorias
var listCategories = "https://www.thymealdb.com/api/json/v1/1/categories.php"; //obtendremos nombre de categoria

axios({
  method: 'GET',
  url: listCategories
}).then(res => {
  mostrarCategorias(res.data);
}).catch(err => console.log(err))
```

## III. Origen de datos

Para este proyecto utilizamos la API MealDB. MealDB API es una base de datos abierta que contiene recetas de todo el mundo. Ofrece datos como los ingredientes, categorías e imágenes de los platillos.

**URL con las instrucciones de como usar la api:**

<https://thymealdb.com/api.php>

**URL con todas las categorías de los platillos:**

<https://www.thymealdb.com/api/json/v1/1/categories.php>

A partir de esta lista de categorías, obtendremos los nombres de cada una de las categorías, para así poder utilizarla como parámetro en la url de los platillos por categoría.

**URL con la lista de platillos por nombre de categoría**

<https://www.thymealdb.com/api/json/v1/1/filter.php?c={nombre de la categoría}>

A partir de esta lista, obtendremos los id de cada uno de los platillos para así poder utilizarlo como parámetro en la url del platillo.

**URL del platillo por ID**

<https://www.thymealdb.com/api/json/v1/1/lookup.php?i={id del platillo}>

Muestra gran variedad de datos relacionados con el platillo en específico

## IV. Capturas demostrativas de la página en ejecución

Al iniciar la página, nos mostrará todas las categorías de los platillos.




Al hacer clic en una de ellas, se mostrará la lista de los platillos de esa categoría.

**Kapsalon**  
**Keleya Zaara**  
**Lamb and Lemon Souvlaki**  
**Lamb and Potato pie**  
**Lamb Biryani**  
**Lamb Rogan josh**  
**Lamb Tagine**  
**Lamb tomato and sweet spices**  
**Lamb Tzatziki Burgers**  
**Lancashire hotpot**  
**McSinghs Scotch pie**  
**Rigatoni with fennel sausage sauce**  
**Stuffed Lamb Tomatoes**  
**Tunisian Lamb Soup**

Al elegir hacer clic en algún platillo de esta receta, se mostrará la receta.

### Stuffed Lamb Tomatoes



**How to prepare?**

Heat oven to 180C/160C fan/gas 4. Slice the tops off the tomatoes and reserve. Scoop out most of the pulp with a teaspoon, being careful not to break the skin. Finely chop the pulp, and keep any juices. Sprinkle the insides of the tomatoes with a little sugar to take away the acidity, then place them on a baking tray. Heat 2 tbsp olive oil in a large frying pan, add the onion and garlic, then gently cook for about 10 mins until soft but not coloured. Add the lamb, cinnamon and tomato purée, turn up the heat, then fry until the meat is browned. Add the tomato pulp and juice, the rice and the stock. Season generously. Bring to the boil, then simmer for 15 mins or until the rice is tender and the liquid has been absorbed. Set aside to cool a little, then stir in the herbs. Stuff the tomatoes up to the brim, top tomatoes with their lids, drizzle with 2 tbsp more olive oil, sprinkle 3 tbsp water into the tray, then bake for 35 mins. Serve with salad and crusty bread, hot or cold.

Vistazo general:

### TusRecetasCaseras

Elige una categoría:

Beef

Chicken

Dessert

Lamb

Miscellaneous

Pasta

Pork

Seafood

Side

Starter

Vegan


Vegetarian

Breakfast

Goat

Kapsalon  
Keleza Zaara  
Lamb and Lemon Souvlaki  
Lamb and Potato pie  
Lamb Biryani  
Lamb Rogan josh  
Lamb Tagine  
Lamb tomato and sweet spices  
Lamb Tzatziki Burgers  
Lancashire hotpot  
McSinghs Scotch pie  
Rigatoni with fennel sausage sauce  
Stuffed Lamb Tomatoes  
Tunisian Lamb Soup

#### Stuffed Lamb Tomatoes



**How to prepare?**

Heat oven to 180C/160C fan/gas 4. Slice the tops off the tomatoes and reserve. Scoop out most of the pulp with a teaspoon, being careful not to break the skin. Finely chop the pulp, and keep any juices. Sprinkle the insides of the tomatoes with a little sugar to take away the acidity, then place them on a baking tray. Heat 2 tbsp olive oil in a large frying pan, add the onion and garlic, then gently cook for about 10 mins until soft but not coloured. Add the lamb, cinnamon and tomato purée, turn up the heat, then fry until the meat is browned. Add the tomato pulp and juice, the rice and the stock. Season generously. Bring to the boil, then simmer for 15 mins or until the rice is tender and the liquid has been absorbed. Set aside to cool a little, then stir in the herbs. Stuff the tomatoes up to the brim, top tomatoes with their lids, drizzle with 2 tbsp more olive oil, sprinkle 3 tbsp water into the tray, then bake for 35 mins. Serve with salad and crusty bread, hot or cold.

## V. Investigación y fuentes consultada

[13 APIs gratis para desarrolladores frontend \(Platzi\)](#)  
[API TheMealDb](#)

## VI. Consideraciones finales

Me siento satisfecho con este trabajo. Siento que domino lo necesario para obtener datos de servidores a través de tecnologías populares.