# Resumen y Ejercicios de Fetch API con Giphy API

## 1. Introducción

En este ejercicio trabajamos con Fetch API y Giphy API para aprender a realizar peticiones HTTP desde el navegador utilizando JavaScript. La Fetch API es una herramienta nativa del navegador que nos permite interactuar con endpoints sin necesidad de bibliotecas externas.

## 2. Configuración de Giphy API

Para trabajar con Giphy API, es necesario crear una cuenta y obtener un API Key. A continuación los pasos resumidos:  
1. Ir a la página de Giphy Developers.  
2. Crear una cuenta y acceder a la sección de 'Create an App'.  
3. Seleccionar 'API' y completar el formulario con un nombre y descripción.  
4. Copiar el API Key proporcionado.

## 3. Uso del Fetch API

El Fetch API nos permite realizar peticiones HTTP de forma nativa. Funciona devolviendo una promesa que se puede manejar con los métodos `.then()` y `.catch()`.

### Ejemplo básico:

const apiKey = 'TU\_API\_KEY';  
const url = `https://api.giphy.com/v1/gifs/random?api\_key=${apiKey}`;  
  
fetch(url)  
 .then(response => response.json())  
 .then(data => {  
 console.log(data);  
 const imgUrl = data.data.images.original.url;  
 console.log('URL del GIF:', imgUrl);  
 })  
 .catch(error => console.warn('Error:', error));

## 4. Encadenamiento de Promesas

Para hacer el código más limpio y manejable, podemos encadenar promesas de la siguiente forma:

### Ejemplo de código mejorado:

fetch(url)  
 .then(response => response.json())  
 .then(data => console.log('Datos del GIF:', data))  
 .catch(error => console.warn('Error:', error));

## 5. Notas adicionales

Para visualizar datos JSON de forma más legible, se recomienda usar extensiones como JSON Viewer en el navegador. También es importante manejar errores con `.catch()` para depuración.