**Resumen API**

Una API es una interfaz integrada a una aplicación o página web que añade características propias externas a la misma página.

Un ejemplo cotidiano puede ser una aplicación de pedidos y delivery, la cual requiere de un seguimiento en tiempo real con mapas, para utilizar esta tecnología se puede utilizar una API tipo Google Maps.

**Transferencia de datos:** JSON, XML e incluso texto plano, siendo la primera la más utilizada en la actualidad.

Existen API´S locales y remotas.. donde las locales se ejecutan de manera local con elementos que componen el sistema como lo puede ser la vibración de un teléfono. Y de manera Remota aquellas que requieren de un servicio externo WEB para su funcionalidad.

**Códigos de estado:** Para saber el estado de una petición a una API se utilizan códigos numéricos..donde:

Codigos 200 | 2XX - Indican Éxito en la petición

Codigos 300 | 3XX - Indican Redirección

Codigos 400 | 4XX - Indican Error en la petición, peticiones invalidas

Codigos 500 | 5XX - Indican Errores directamente del servidor

***Video: ¿Qué es una API y para qué sirve? - EDteam YouTube***

**¿Cómo utilizar Google API?**

Para poder utilizar google api se debe crear una cuenta en google developers, la página brindará credenciales de autorización, las cuales servirán para el funcionamiento de la API.

Por lo demás solo restaría añadir el código proporcionado a un determinado proyecto.

**Funciones Sincronas**

Funciones Sincronas: Funciones tradicionales, se ejecutan con recursos existentes en el código..como lo puede ser una estructura de datos, un array, etc.

**Funciones Asíncronas**

Funciones Asíncronas: Son funciones que esperan parámetros y recursos externos al código..como puede ser información de una base de datos, JSON externo, etc.

Al no existir estos recursos directamente en el código existe un retraso en la ejecución.

Para tratar textos retrasos y no caer en errores se trabaja con promesas.

**API FETCH:**

Fetch es una api con métodos incorporados en Javascript que permite acceder a datos externos al código de un proyecto, aplicación, etc.

Fetch funciona por medio de peticiones asíncronas y promesas.

**Promesas en Javascript:**

En javascript una promesa es un resultado pendiente, como lo dice su nombre “promesa”, al utilizar fetch se espera que la información solicitada llegue para poder trabajar y manipular datos con ella.

***VIDEO: Lo básico sobre Fetch / Promesas y Async Await - DEVYAN - Youtube.***