

UD 8

Bibliotecas, APIs y autenticación web

DESARROLLO WEB EN ENTORNO CLIENTE

Técnico de Grado Superior Desarrollo de Aplicaciones Web

2024-25

Contenidos

- Geolocalización.
 - API de geolocalización en HTML 5.
 - Utilización de la geolocalización.
- Mapas: API de Google Maps y alternativas
- Utilización de Mapas
 - Configuración y utilización
 - Controles
 - Eventos
 - Rutas y distancias
- Lectura de ficheros desde lado cliente.
- Gráficos estadísticos en el lado cliente.
- Librerías para generar gráficas y diagramas.
- C3.js

Introducción

HTML, CSS y JavaScript proporcionan unos fundamentos sobre los que *la web actual* consigue funcionalidades más avanzadas y complejas para crear aplicaciones interactivas y dinámicas que satisfagan las necesidades de los usuarios.

El sistema de módulos de JavaScript, la reutilización de código mediante bibliotecas externas, el uso de las Web APIs, así como el uso de servicios de terceros que requieren autenticación serán los asuntos que afrontaremos en esta unidad didáctica para extender nuestras posibilidades en desarrollo web.

Introducción

Para llevar nuestras aplicaciones al siguiente nivel, es necesario integrar funcionalidades avanzadas que van más allá del núcleo del lenguaje.

Estas funcionalidades se pueden implementar utilizando:

- **WebAPIs** (estándar) APIs proporcionadas por el navegador que permiten interactuar con el hardware, el sistema operativo o el propio navegador.
- **Librerías**, que son conjuntos de código preescrito que facilitan tareas específicas, como manipular datos, crear gráficos o integrar mapas.
- **APIs** externas (HTTP), servicios web proporcionados por terceros que permiten acceder a datos o funcionalidades remotas, como redes sociales, servicios de pago, bases de datos, etc.

Antes vamos a ver cómo JavaScript permite trabajar con varios ficheros de código fuente...

Módulos en JavaScript

Los módulos en JavaScript son una forma de organizar y reutilizar código mediante la importación y exportación de funcionalidades entre archivos.

La sintaxis moderna de módulos (ES6+) nos permite:

el sistema de módulos es fundamental para trabajar con bibliotecas y mantener nuestro código organizado

```
// Exportar funcionalidades
export function miFuncion() { ... }
export const miConstante = 42;

// Importar funcionalidades
import { miFuncion, miConstante } from './miModulo.js';
```

- La palabra clave **export** etiqueta las variables y funciones que necesitan ser accesibles desde fuera del módulo actual.
- **import** permite importar funcionalidades desde otros módulos.

Módulos en JavaScript

¿Por qué usar módulos?

Reutilización de código; podemos escribir código reutilizable y compartirlo entre distintos archivos.

Mantenimiento; al dividir el código, es más fácil depurar y trabajar en partes específicas de la aplicación.

Encapsulación; los módulos permiten ocultar detalles de implementación y exponer solo lo necesario.

El atributo `type="module"` en HTML

Para usar módulos debemos añadir el atributo `type="module"` al elemento `<script>` en nuestro archivo HTML. Esto indica al navegador que el archivo JavaScript es un módulo y no bloquea el procesamiento de HTML, se cargan en paralelo junto con otros recursos (mismo comportamiento que `defer`).

Los módulos siempre usan modo estricto

Por ejemplo, asignar a una variable sin declarar nos dará un error.

Alcance a nivel de módulo

Las variables y funciones de un módulo no se ven en otros scripts, cada módulo tiene su propio alcance de nivel superior.

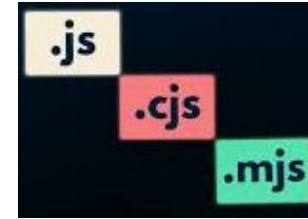
Módulos en JavaScript

Reto:

Practicar la exportación e importación de funciones y variables entre diferentes archivos.
Para ello:

- Crea un archivo llamado `operaciones.js` y **exporta** las siguientes funciones:
 - `sumar(a, b)`: devuelve la suma de dos números.
 - `restar(a, b)`: devuelve la resta de dos números.
 - `PI`: una constante con el valor de π (3.14).
- Crea un archivo llamado `main.js` que **importe** las funciones y la constante de `operaciones.js` y haga lo siguiente:
 - Muestra por consola el resultado de sumar 5 y 10, por ejemplo.
 - Muestra por consola el resultado de restar 15 y 8, por ejemplo.
 - Muestra por consola el valor de `PI`.
- Enlaza el archivo `main.js` en un archivo HTML usando el atributo `type="module"`.

Módulos en JavaScript



La extensión ayuda a identificarlos

Antes, todo el código JavaScript se escribía en un solo archivo

Ante la necesidad de dividirlo, se creó CommonJS y de su uso, los ficheros .cjs.

ES6 introdujo (y los archivos .mjs) para dividir el código en partes más pequeñas y reutilizables

src/index.js

```
import bar from './bar.js';  
  
bar();
```

src/bar.js

```
export default function bar() {  
  //  
}
```

Empaqueta módulos de JS y, ya puestos, otros activos (imágenes, hojas de estilo, etc.) para su uso en el navegador.



Bundle It

Without config or provide custom webpack.config.js

```
const path = require('path');  
  
module.exports = {  
  entry: './src/index.js',  
  output: {  
    path: path.resolve(__dirname, 'dist'),  
    filename: 'bundle.js',  
  },  
};
```

page.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8" />  
    ...  
  </head>  
  <body>  
    ...  
    <script src="dist/bundle.js"></script>  
  </body>  
</html>
```



Los módulos de navegador rara vez se usan “tal cual”, se suelen agrupar con herramientas empaquetadoras.

Web APIs

Estas son solo
algunas

Son interfaces proporcionadas por los navegadores que permiten interactuar con el entorno del usuario.

Estas APIs están estandarizadas y son soportadas por la mayoría de los navegadores modernos.
No requieren importación de código externo.

DOM API

```
document.getElementById('miElemento').innerHTML = '¡Hola, mundo!';
```

Geolocation API

```
navigator.geolocation.getCurrentPosition(position => {  
  console.log(`Lat: ${position.coords.latitude}, Long: ${position.coords.longitude}`);  
});
```

Fetch API

```
fetch('https://api.ejemplo.com/datos')  
  .then(response => response.json())  
  .then(data => console.log(data));
```

Web Storage API

```
localStorage.setItem('nombre', 'Juan');  
console.log(localStorage.getItem('nombre')); // Juan
```

DOM

manipulación de la estructura y contenido HTML.

Canvas API

dibujo y gráficos 2D.

WebGL

gráficos 3D acelerados por hardware.

Geolocation API

ubicación geográfica.

Web Audio API

manipulación y procesamiento de audio.

WebRTC

comunicación en tiempo real entre navegadores.

IndexedDB API

almacenamiento NoSQL.

Local Storage API

almacenamiento clave-valor.

Fetch API

solicitudes HTTP.

Notifications API

notificaciones al usuario.

Battery API

estado de la batería del dispositivo.

Web Workers API

ejecución de scripts en segundo plano.

Service Workers API

PWAs y funcionalidades offline.


Media Devices API

dispositivos multimedia (cámaras, micrófonos).

Pointer Events API

eventos de puntero (ratón, lápiz táctil, dedo).

Web APIs

 mdn web docs



Tecnología para desarrolladores web >
Referencia de la API Web



Referencia de la API Web

Cuando escribimos código para la web utilizando JavaScript, podemos usar gran número de APIs disponibles. A continuación mostramos una lista de todas las interfaces (es decir, tipos de objetos) que puedes usar al desarrollar tu aplicación o sitio Web. Para obtener una lista de las API que contiene cada una de estas interfaces, consulta la referencia de la API Web.

A

- [AbortController](#) (inglés)
- [AbortSignal](#) (inglés)
- [AbsoluteOrientationSensor](#) (inglés)
- [AbstractRange](#) (inglés)
- [Accelerometer](#) (inglés) 
- [AesCbcParams](#) (inglés)
- [AesCtrParams](#) (inglés)
- [AesGcmParams](#) (inglés)
- [AesKeyGenParams](#) (inglés)
- [AmbientLightSensor](#) (inglés) 
- [AnalyserNode](#) (inglés)
- [ANGLE_instanced_arrays](#) (inglés)
- [Animation](#) (inglés)
- [AnimationEffect](#) (inglés)
- [AnimationEvent](#) (inglés)
- [AnimationPlaybackEvent](#) (inglés)
- [AnimationTimeline](#) (inglés)
- [Attr](#) (inglés)
- [AudioBuffer](#) (inglés)
- [AudioBufferSourceNode](#) (inglés)
- [AudioContext](#) (inglés)
- [AudioData](#) (inglés)
- [AudioDecoder](#) (inglés)
- [AudioDestinationNode](#) (inglés)
- [AudioEncoder](#) (inglés)
- [AudioListener](#) (inglés)
- [AudioNode](#) (inglés)
- [AudioParam](#) (inglés)
- [AudioParamDescriptor](#) (inglés)
- [AudioParamMap](#) (inglés)
- [AudioProcessingEvent](#) (inglés) 
- [AudioScheduledSourceNode](#) (inglés)
- [AudioSinkInfo](#) (inglés) 
- [AudioTrack](#) (inglés)
- [AudioTrackList](#) (inglés)
- [AudioWorklet](#) (inglés)
- [AudioWorkletGlobalScope](#) (inglés)
- [AudioWorkletNode](#) (inglés)
- [AudioWorkletProcessor](#) (inglés)
- [AuthenticatorAssertionResponse](#) (inglés)
- [AuthenticatorAttestationResponse](#) (inglés)
- [AuthenticatorResponse](#) (inglés)

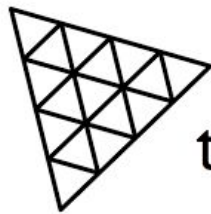
B

- [BackgroundFetchEvent](#) (inglés) 
- [BackgroundFetchManager](#) (inglés) 
- [BackgroundFetchRecord](#) (inglés) 
- [BackgroundFetchRegistration](#)
- [BeforeInstallPromptEvent](#) (inglés) 
- [BeforeUnloadEvent](#) (inglés)
- [BiquadFilterNode](#) (inglés)
- [BluetoothRemoteGATTCharacteristic](#) (inglés) 
- [BluetoothRemoteGATTDescriptor](#) (inglés) 

Librerías (o bibliotecas)

Son conjuntos de código preescrito que facilitan tareas específicas.

Estas pueden ser de uso general o estar especializadas en áreas como gráficos, mapas, etc.



three.js



Hay... muchísimas

Oo a las licencias de uso



APIs externas

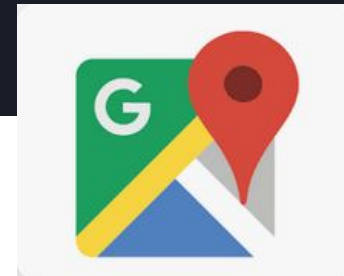
Las APIs externas son servicios web proporcionados por terceros que permiten acceder a datos o funcionalidades remotas.

Estas APIs suelen requerir autenticación y se consumen mediante solicitudes HTTP.

```
fetch('https://api.twitter.com/2/tweets/12345', {  
  headers: {  
    'Authorization': 'Bearer TOKEN_DE_ACCESO'  
  }  
})  
.then(response => response.json())  
.then(data => console.log(data));
```



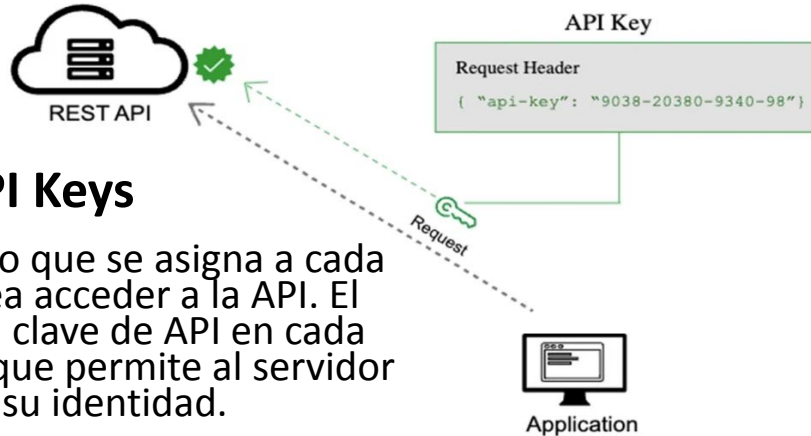
```
const map = new google.maps.Map(document.getElementById('map'), {  
  center: {lat: -34.397, lng: 150.644},  
  zoom: 8  
});
```



El cliente realiza una solicitud HTTP al servidor de la API, especificando la acción que desea realizar y los datos que necesita. El servidor de la API recibe la solicitud, la procesa y envía una respuesta HTTP al cliente, que contiene los datos solicitados o un mensaje de error si la solicitud no pudo ser procesada.

APIs externas - Autenticación

Permite **verificar** la identidad del cliente que realiza la solicitud y garantizar que solo los clientes **autorizados** puedan acceder a los recursos de la API.



API Keys

Identificador único que se asigna a cada cliente que desea acceder a la API. El cliente incluye la clave de API en cada solicitud HTTP, lo que permite al servidor verificar su identidad.

JWT (JSON Web Token)

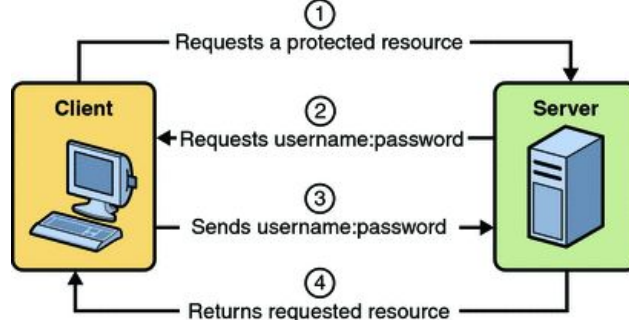
Estándar abierto para representar reclamaciones (claims); un JWT contiene información sobre el cliente y se utiliza para autenticar y autorizar el acceso a los recursos de la API.



Header . Payload . Signature

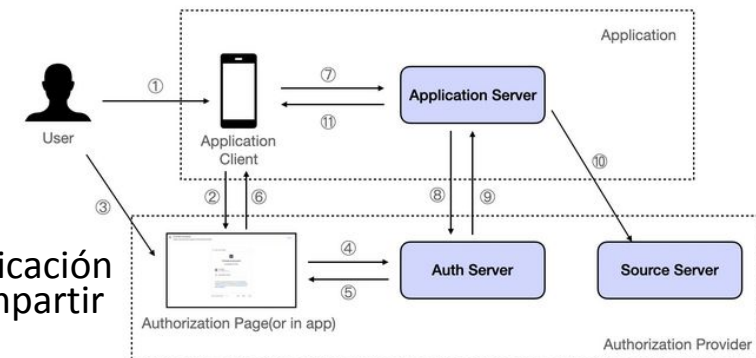
Basic authentication

El cliente envía sus credenciales (nombre de usuario y contraseña) en cada solicitud HTTP



OAuth (2.0)

Protocolo de autorización que permite a una aplicación acceder a los recursos de otra aplicación sin compartir credenciales



`'https://api.twitter.com/oauth/authorize?oauth_token=TOKEN'`

APIs externas - Autenticación

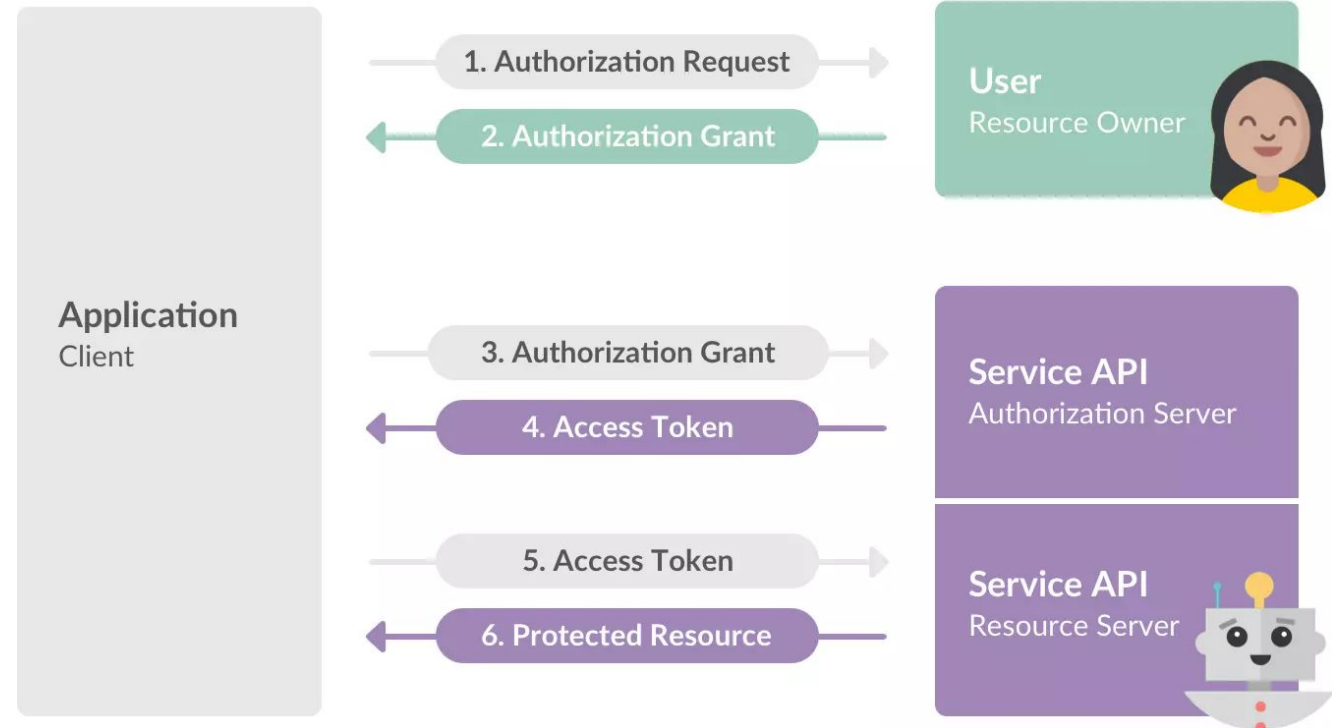
OAuth 2.0

Arquitectura del Protocolo:

- Roles
 - Application (el cliente). Tu aplicación consumidora
 - APIs. El servicio proveedor:
 - Resource
 - Authorization
 - Users. El usuario (y los recursos que quiere compartir)
- Tipos de garantías (grant types)
 - Obtener un código de autorización
 - Password
 - Uso de credenciales
- Flujo de autorización. Depende de la aplicación
 - web server. Basado en el servidor
 - browser-based. Basado en el cliente (el que usaremos)
 - mobile apps. Para apps

Entender el flujo de autorización OAuth

- Al aplicación solicita autorización
- El usuario lo acepta, o lo deniega
- Si lo acepta, la aplicación tiene autorización de acceso y lo solicita al servidor
- El servidor acepta la petición y devuelve un token de acceso
- Con el token de acceso la aplicación accede al recurso



Aplicaciones prácticas



API de Geolocalización

La API de geolocalización en HTML5 permite a las aplicaciones web acceder a la ubicación geográfica del usuario (**si este lo permite**). Esta API es útil para aplicaciones que necesitan mostrar contenido basado en la ubicación, como mapas, restaurantes cercanos, etc.

API de geolocalización

Geolocalización

▼ Métodos de instancia

`clearWatch()`

`getCurrentPosition()`

`watchPosition()`

▼ Páginas relacionadas a Geolocation API

`GeolocationCoordinates`

`GeolocationPosition`

`GeolocationPositionError` (inglés)

`Navigator.geolocation`

Geolocalización

✓ **Baseline** Widely available



La interfaz **Geolocalización** representa un objeto capaz de obtener mediante programación la posición del dispositivo. Se da acceso Web de la ubicación del dispositivo. Esto permite a un sitio Web o aplicación ofrecer resultados personalizados basados en la ubicación del usuario.

Se obtiene un objeto con esta interfaz mediante la propiedad `NavigatorGeolocation.geolocation` implementado por el objeto `Navigator`.

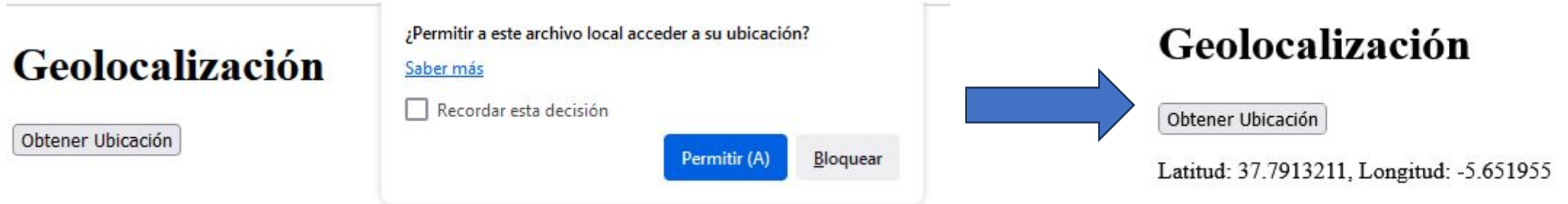
Nota: Por razones de seguridad, cuando una página web intenta obtener acceso a la información de ubicación, se notifica al usuario y le pide conceder el permiso. Tenga en cuenta que cada navegador tiene sus propias políticas y métodos para solicitar este permiso.

API de geolocalización

Reto:

Crea una página que contenga un botón, el cual, al ser pulsado, trata de obtener las coordenadas de la ubicación.

Una vez conseguido, muestra también la precisión de dicha geolocalización.



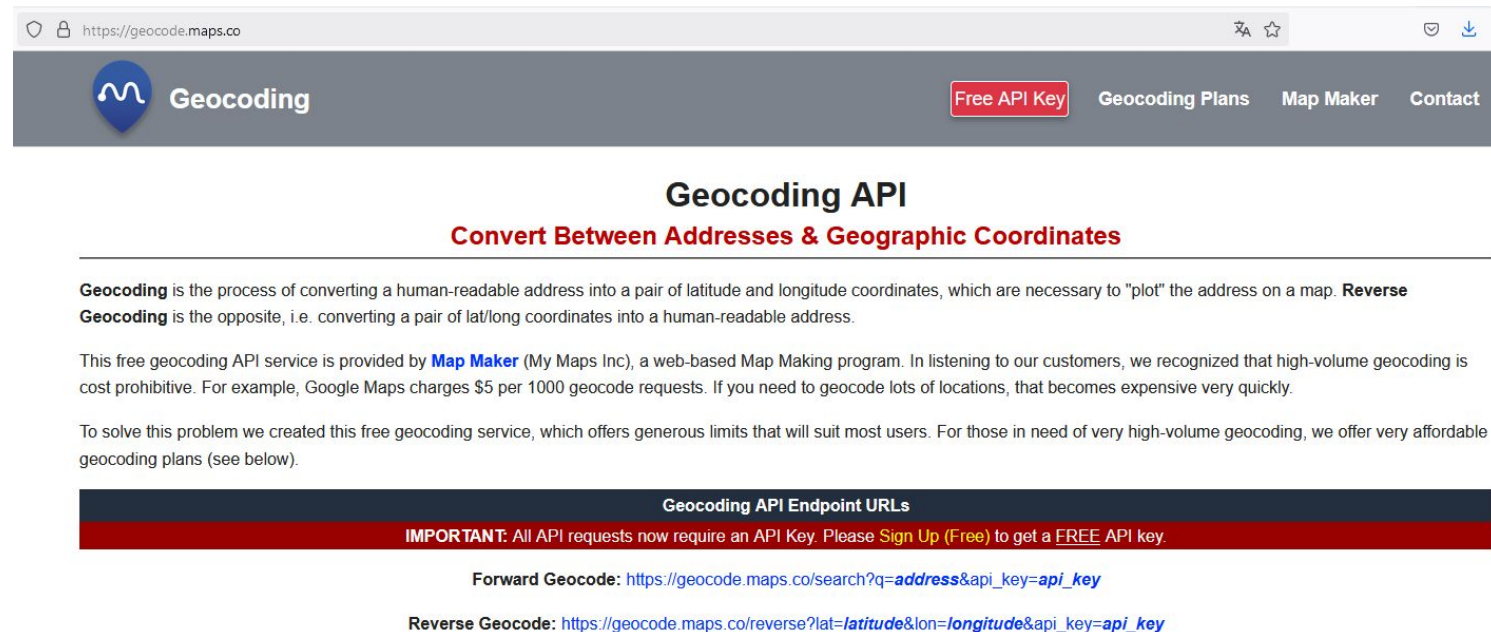
API de geolocalización

Reto:

Una vez obtenidas las coordenadas, trata de encontrar a qué ubicación corresponde.

Esta funcionalidad, llamada GEOCODING, no es nativa, tendrás que utilizar un servicio externo.

Por ejemplo:



The screenshot shows the homepage of the Geocoding API service. The browser address bar displays 'https://geocode.maps.co'. The website has a dark grey header with the 'Geocoding' logo, a 'Free API Key' button, and links for 'Geocoding Plans', 'Map Maker', and 'Contact'. The main content area is titled 'Geocoding API' and 'Convert Between Addresses & Geographic Coordinates'. It explains that geocoding is the process of converting a human-readable address into latitude and longitude coordinates, and reverse geocoding is the opposite. It mentions that the service is provided by Map Maker (My Maps Inc.) and is free, unlike Google Maps which charges \$5 per 1000 requests. It also states that the service offers generous limits for most users and very affordable plans for high-volume users. At the bottom, there is a section for 'Geocoding API Endpoint URLs' with an important note: 'IMPORTANT: All API requests now require an API Key. Please Sign Up (Free) to get a FREE API key.' Below this, the forward geocode endpoint is given as 'https://geocode.maps.co/search?q=address&api_key=api_key' and the reverse geocode endpoint as 'https://geocode.maps.co/reverse?lat=latitude&lon=longitude&api_key=api_key'.

Geocoding API

Convert Between Addresses & Geographic Coordinates

Geocoding is the process of converting a human-readable address into a pair of latitude and longitude coordinates, which are necessary to "plot" the address on a map. **Reverse Geocoding** is the opposite, i.e. converting a pair of lat/long coordinates into a human-readable address.

This free geocoding API service is provided by **Map Maker** (My Maps Inc), a web-based Map Making program. In listening to our customers, we recognized that high-volume geocoding is cost prohibitive. For example, Google Maps charges \$5 per 1000 geocode requests. If you need to geocode lots of locations, that becomes expensive very quickly.

To solve this problem we created this free geocoding service, which offers generous limits that will suit most users. For those in need of very high-volume geocoding, we offer very affordable geocoding plans (see below).

Geocoding API Endpoint URLs

IMPORTANT: All API requests now require an API Key. Please [Sign Up \(Free\)](#) to get a **FREE** API key.

Forward Geocode: https://geocode.maps.co/search?q=address&api_key=api_key

Reverse Geocode: https://geocode.maps.co/reverse?lat=latitude&lon=longitude&api_key=api_key

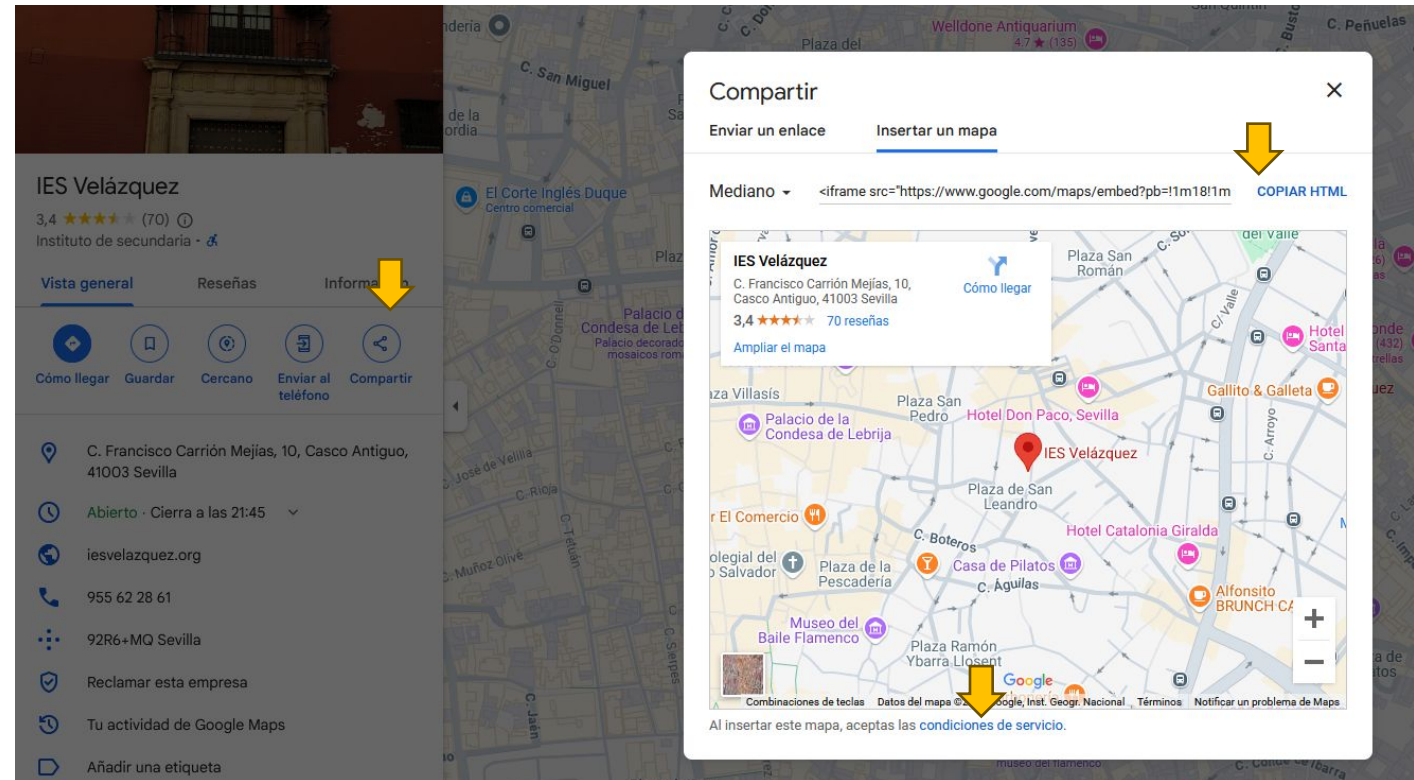
Mapas con Google Maps iframes

Con un iframe podemos mostrar el mapa de forma rápida y sencilla.

- ideal para necesidades básicas de ubicación
- capacidad de interacción y adaptación limitada.

La API de Google Maps proporciona un conjunto de herramientas y funcionalidades avanzadas que permiten crear mapas dinámicos y personalizados (añadir marcadores interactivos, dibujar rutas, integrar información adicional y adaptar el mapa a la estética de tu web).

- puede tener coste (a partir de determinada cuota de uso)
- hay que aprender a utilizar ese API



Mapas con Google Maps APIKey y documentación

Ahora, utilizar APIs de Google requiere haber introducido algún método de pago en nuestra cuenta, aunque no sobrepasemos el uso gratuito

The screenshot displays the Google Maps Platform documentation page for the Maps JavaScript API. The page is divided into several sections:

- Header:** Includes the Google Maps Platform logo and navigation links: Descripción general, Productos, Precios, Documentación (selected), Blog, and Comunidad.
- Left Sidebar:** Contains a 'Filtrar' button and a list of links for the Maps JavaScript API, including 'Descripción general', 'Configura tu proyecto de Google Cloud', 'Usa claves de API', 'Usa la Verificación de aplicaciones para proteger tu clave de API', 'Carga la API de Maps JavaScript', 'Solución de problemas', 'Instructivos', 'Cómo agregar un mapa de Google Maps con un marcador usando HTML', 'Cómo agregar un mapa de Google Maps con un marcador usando JavaScript', 'Agrega un mapa de Google Maps a una app de React', 'Más instructivos', 'Conceptos', 'Control de versiones', 'Localización', 'Prácticas recomendadas', 'TypeScript', 'Promesas', 'Mapa base', 'Cómo agregar un mapa de Google Maps a una página web', and 'Eventos de mapa'.
- Main Content Area:**
 - API de Maps JavaScript:** A section with a map preview and the text 'Crea mapas dinámicos, interactivos y muy personalizados para ubicaciones y experiencias geoespaciales para tu aplicación'.
 - Obtén una clave de API:** A section explaining how to get an API key. It includes a button 'OBTENER UNA CLAVE' and a list of steps: 1. Ingresa a Google API Console, 2. Crea o selecciona un proyecto, 3. Haz clic en **Continue** para habilitar la API y cualquier servicio relacionado, 4. En la página **Credentials**, obtén una **clave de API** (y configura las restricciones para esta). Nota: Si ya tienes una clave de API sin restricciones o una clave con restricciones de navegador, puedes usarla, 5. Para evitar el robo de cuota, **protege tu clave de API siguiendo estas prácticas recomendadas**, 6. (Opcional) Habilita la facturación. Para obtener más información, consulta **Límites de uso**.
- Comenzar:** A section titled 'Comienza a crear mapas con las funciones más usadas de la API de Maps JavaScript.' containing two cards:
 - Comienza a utilizar Google Maps Platform:** 'Sigue la guía de introducción de Google Maps Platform para crear una cuenta, generar una clave de API y comenzar a crear mapas.'
 - Crea tu primer mapa con un marcador:** 'Obtén información sobre cómo cargar la API de Maps JavaScript y agregar un mapa con un marcador a tu app web.'
- Mobile Navigation Menu:** A dropdown menu showing options like 'Configura y crea un mapa', 'Cómo usar las funciones de la API de Maps JavaScript', 'Trabaja con los servicios de la API de Maps JavaScript', 'Trabaja con las bibliotecas de la API de Maps JavaScript', and 'Consulta el código de ejemplo de las funciones de Maps'.

Mapas con Google Maps

Ventajas, inconvenientes y alternativas

La API de Google es sencilla y dispone de muy buena documentación.

Galería de ejemplos, guías para el aprendizaje y descripción de la librería muy completa.

La principal desventaja de Google es... Google

Se trata de una empresa comercial y por tanto sus productos siguen sus normas y precios.

No hay que olvidar que aunque la librería es gratuita, si superamos cierto número de visitas o necesitamos servicios adicionales hay que pagar por ellos. Estamos sujetos a sus condiciones de uso.

Tener que registrarse y solicitar una clave no es un punto a su favor. Si además la adición del API requiere incluir un método de pago, la barrera de entrada es aún mayor.

Además estamos obligados a utilizar el mapa de Google, lo que podría limitar nuestras posibilidades.

¿Alternativas?

Librerías Open Source como Leaflet y OpenLayers.

No proporcionan un mapa base, y que por lo tanto necesitan acudir a otros servicios como los de OpenStreetMap.

En muchos casos serán una solución suficiente para publicar nuestros mapas en la web.



Lectura de ficheros desde el lado cliente

JavaScript permite leer archivos desde el lado cliente utilizando el objeto [FileReader](#).

File API (inglés)

FileReader

▼ Constructor

[FileReader\(\)](#) (inglés)

▼ Propiedades de instancia

[error](#) (inglés)

[readyState](#) (inglés)

[result](#)

▼ Métodos de instancia

[abort\(\)](#) (inglés)

[readAsArrayBuffer\(\)](#)

[readAsBinaryString\(\)](#) (inglés) 

[readAsDataURL\(\)](#)

[readAsText\(\)](#)

FileReader

✓ Baseline Widely available



Sumario

El objeto [FileReader](#) permite que las aplicaciones web lean ficheros (o información en buffer) almacenados en el cliente de forma asíncrona, usando los objetos [File](#) o [Blob](#) dependiendo de los datos que se pretenden leer.

El objeto [File](#) puede ser obtenido desde un objeto [FileList](#) (inglés) devuelto como resultado de la selección de archivos por parte del usuario en un elemento [<input>](#), desde el objeto [DataTransfer](#) (inglés) producido por una operación de arrastre (drag and drop) o desde la API [mozGetAsFile\(\)](#) en un [HTMLCanvasElement](#).

Constructor

```
FileReader FileReader();
```

Lectura de ficheros desde el lado cliente

Se puede:

- Leer archivos de texto: archivos .txt, .csv, .json...
- Leer archivos binarios: imágenes, audio, videos...
- Leer archivos en formato *Data URL*, por ejemplo para previsualizar imágenes o subirlas a un servidor.
- Leer archivos en formato *ArrayBuffer*, para manipular datos binarios directamente.

¿Qué se puede y qué no se puede hacer con FileReader?

No se puede:

- No se puede acceder a archivos locales directamente.
Ya vimos que por razones de seguridad, el navegador no permite acceder a archivos locales sin la interacción del usuario (por ejemplo, mediante un `<input type="file">`).
- No se puede leer archivos del sistema de archivos del servidor.
FileReader solo funciona con archivos seleccionados por el usuario en el cliente.
- Tamaño de archivo limitado: Dependiendo del navegador y la memoria disponible, archivos muy grandes pueden causar problemas de rendimiento.

Lectura de ficheros desde el lado cliente

El navegador **no permite acceder a archivos locales** directamente por razones de seguridad.

Hemos utilizado para conseguir el acceso `<input type="file">`

¿Se os ocurren otras opciones para leer ficheros?

- Arrastrar y soltar (*drag and drop*)

Permite al usuario arrastrar archivos desde su sistema de archivos a una zona específica de la página.



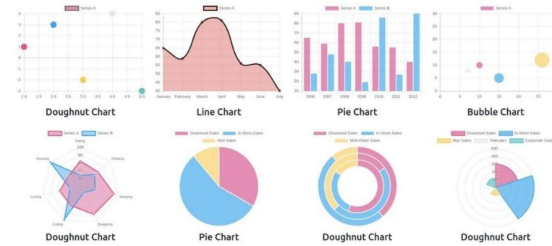
Puede arrastrar y soltar archivos aquí para añadirlos

- Selección de archivos mediante un botón personalizado

Se puede ocultar el `<input type="file">` y activarlo mediante un botón personalizado.



Gráficos estadísticos



permiten representar datos de manera visual en una página web, facilitando su comprensión o el mensaje

Algunas bibliotecas populares para generar gráficos en JavaScript son:

- **Chart.js**: fácil de usar, ideal para gráficos interactivos.
- **D3.js**: permite crear gráficos altamente personalizables manipulando el DOM.
- **C3.js**: basada en D3.js, pero con una API más simple.
- **Charts** de Google: potentes, fáciles de usar y gratuitas.

Gráficos estadísticos



Chart.js

Simple yet flexible JavaScript charting library for the modern web

[Get Started](#)[Samples](#)[Ecosystem](#)[GitHub](#)

New in 4.0 Colors plugin

Default palette of Chart.js brand colors is available as a built-in time-saving zero-configuration plugin.

New in 4.0 Tree-shaking

JavaScript bundle size can be reduced by dozens of kilobytes by registering only necessary components.



New in 3.5 Scale stacking

Layout boxes can be stacked and weighted in groups.

Main title

The JavaScript library for bespoke data visualization

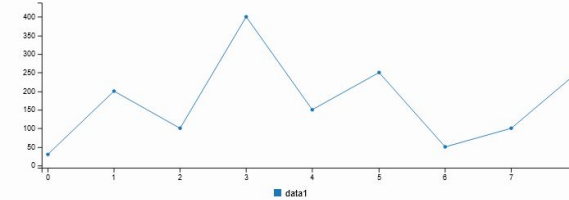
Create custom dynamic visualizations with unparalleled flexibility

[Get started](#)[What is D3?](#)[Examples](#)

C3.js | D3-based reusable chart library

[Getting Started](#)[Examples](#)[Reference](#)[Forum](#)[Source](#)

C3.js D3-based reusable chart library

[Start Demo](#)

Why C3?

Comfortable

C3 makes it easy to generate D3-based charts by wrapping the code required to construct the entire chart. We don't need to write D3 code any more.

Customizable

C3 gives some classes to each element when generating, so you can define a custom style by the class and it's possible to extend the structure directly by D3.

Controllable

C3 provides a variety of APIs and callbacks to access the state of the chart. By using them, you can update the chart even after it's rendered.

Google Charts

Español - A...

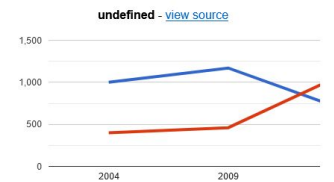


C3 enables deeper integration of charts into:

Muestra datos publicados en tu sitio

Acerca de las herramientas de gráficos de Google

Las herramientas de gráficos de Google son potentes, fáciles de usar y gratuitas. Prueba nuestra extensa galería de herramientas de datos y gráficos interactivos.

[Comenzar](#) [Galería de gráficos](#)

pastel [opcional](#) [área](#) [árbol](#) [más](#)



Galería enriquecida

Elige entre una variedad de gráficos. Encuentra la opción que mejor se adapte a tus datos, desde simples diagramas de dispersión hasta mapas de árboles jerárquicos.



Gratis



Personalizable

Personaliza los gráficos. Configura un amplio conjunto de opciones para que coincida perfectamente con el aspecto de tu sitio web.



Controles y paneles de control



HTML5 o SVG

Compatibilidad entre navegadores (con la adopción de VML para versiones anteriores de IE) y portabilidad multiplataforma a iOS y las nuevas versiones de Android. No se necesitan complementos.

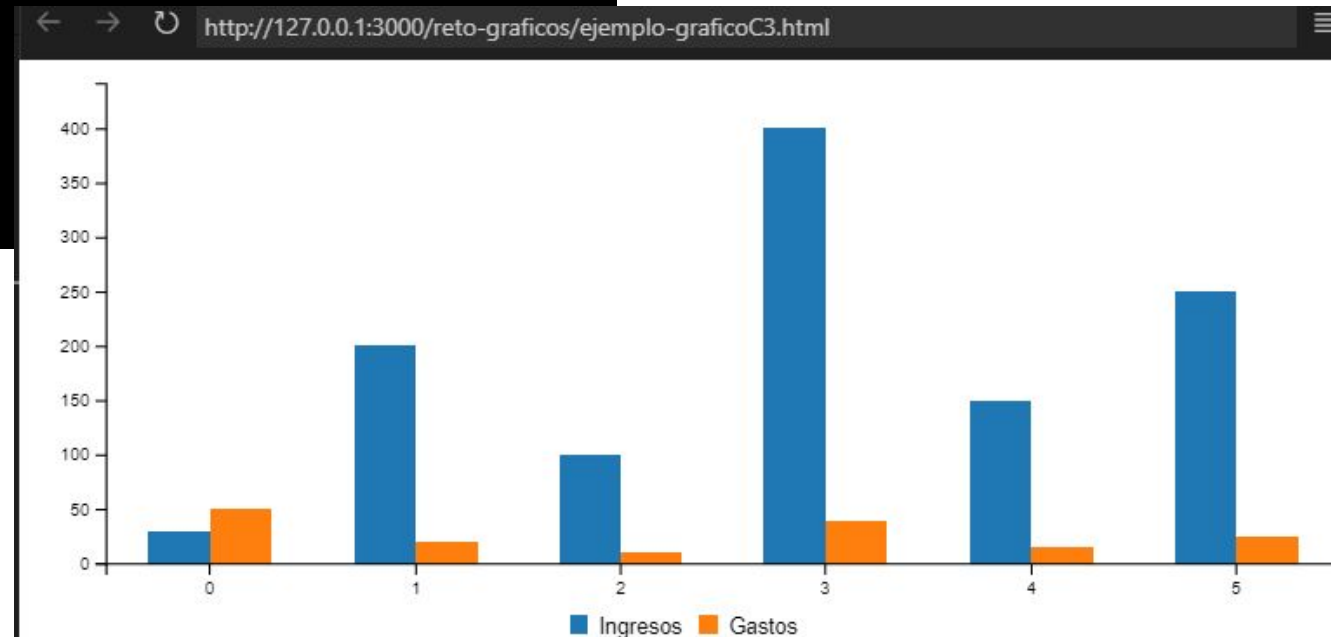


Datos dinámicos

Ejemplo con C3.js

```
<div id="chart"></div>

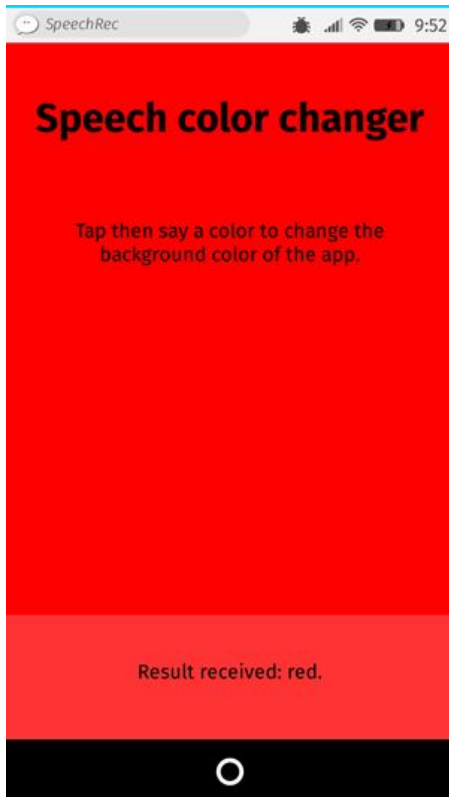
<script>
  var chart = c3.generate({
    bindto: '#chart',
    data: {
      columns: [
        ['Ingresos', 30, 200, 100, 400, 150, 250],
        ['Gastos', 50, 20, 10, 40, 15, 25]
      ],
      type: 'bar'
    }
  });
</script>
```



Otras Web APIs

SpeechRecognition


 Limited availability



¡Podéis probarlo!

Web Speech API Demonstration

Hola cómo estás



[SpeechRecognition.lang](#)

Returns and sets the language of the current `SpeechRecognition`. If not specified, this defaults to the HTML `lang` attribute value, or the user agent's language setting if that isn't set either.

[SpeechRecognition.continuous](#)

Controls whether continuous results are returned for each recognition, or only a single result. Defaults to single (`false`).

[SpeechRecognition.interimResults](#)

Controls whether interim results should be returned (`true`) or not (`false`). Interim results are results that are not yet final (e.g. the `SpeechRecognitionResult.isFinal` property is `false`).

[SpeechRecognition.abort\(\)](#)

Stops the speech recognition service from listening to incoming audio, and doesn't attempt to return a `SpeechRecognitionResult`.

[SpeechRecognition.start\(\)](#)

Starts the speech recognition service listening to incoming audio with intent to recognize grammars associated with the current `SpeechRecognition`.

[SpeechRecognition.stop\(\)](#)

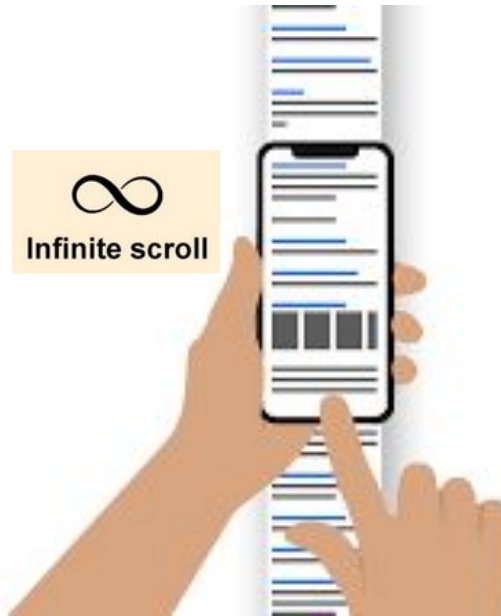
Stops the speech recognition service from listening to incoming audio, and attempts to return a `SpeechRecognitionResult` using the audio captured so far.

Otras Web APIs

Intersection Observer API

Permite observar de forma asíncrona los cambios en la intersección de un elemento con otro elemento (o con el *viewport* del documento)

¿Algún ejemplo de la utilidad de esto?



Otras Web APIs

Web Workers API

hacen posible ejecutar la operación de un script en un hilo en segundo plano separado de la ejecución del hilo principal de la aplicación web. La ventaja de esto es que un proceso laborioso puede actuar en un hilo separado, permitiendo al hilo principal (normalmente la UI) ejecutarse sin ser bloqueado o ralentizado.

Usando Web Workers

Los Web Workers dedicados proveen un medio sencillo para que el contenido web ejecute scripts en hilos en segundo plano. Una vez creado, un worker puede enviar mensajes a la tarea creada mediante envío de mensajes al manejador de eventos especificado por el creador. Sin embargo, **los workers trabajan dentro de un [contexto global](#) (inglés) diferente de la ventana actual** (usar el atajo `window` en lugar de `self` (inglés) con el fin de obtener el scope actual dentro de un `Worker` retornaría, de hecho, un error).

