






Objetivo:

- Escribir código para comunicarse con el servidor web, definiendo parámetros de petición y de recepción, procesamiento y presentación de la respuesta.
- Obtener datos del servidor y utilizarlos para actualizar el contenido de una página web.

Una tarea muy común en los sitios web y aplicaciones modernas es la de recuperar elementos de datos individuales del servidor para actualizar secciones de una página web sin tener que recargar la página completa. Este detalle, aparentemente pequeño, tiene gran impacto en el rendimiento y comportamiento de los sitios. En la actividad propuesta el alumno deberá poner en práctica **al menos dos** de las siguientes tecnologías que lo hacen posible:

Método	Descripción
XMLHttpRequest	Se suele abreviar como XHR . El más antiguo, y también el más verbose. Nativo. 
fetch	Nuevo sistema nativo de peticiones basado en promesas. Sin soporte en IE.
Axios 	Librería basada en promesas para realizar peticiones en Node o navegadores. 
superagent 	Librería para realizar peticiones HTTP tanto en Node como en navegadores.
frisbee 	Librería basada en fetch. Suele usarse junto a React Native.

Estas tecnologías permiten que las páginas web manejen directamente las solicitudes HTTP para recursos específicos disponibles en un servidor y formatee los datos resultantes según sea necesario antes de que se muestren.

Se pide:

Página/s web (*deberá ser diferente a la de tus compañeros*) que utilice JavaScript / TypeScript para hacer peticiones http que actualicen contenidos de forma parcial, de modo que se actualice la página «en vivo» (solamente una pequeña parte de ella) sin necesidad de recargar toda la página.

Debes realizar al menos **tres versiones de la misma página** empleando diferentes métodos de petición AJAX.

- **versión 1** (empleando la API Fetch)
- **versión 2** (empleando la API Fetch y [async / await](#) para gestionar las Promises). [Aquí tienes una demo de async await](#)
- **versión 3** (empleando una librería externa, Axios o SuperAgent). [Demo Axios](#); [Ejemplo Fetch, Axios, Request y SuperAgent](#)

Funcionalidad de la página:

A determinar por el autor. Puedes extraer la información:

- ☐ desde una API WEB
 - ☐ [TOP 7 mejores APIs GRATIS para tus DESARROLLOS](#)
 - ☐ [APIs públicas para practicar](#)
- ☐ desde archivos txt
- ☐ desde archivos json

Entregables:

- **documentación.** Documento pdf explicativo que incluya:
 - temática elegida y funcionalidad lograda
 - métodos/tecnologías empleadas para el acceso asíncrono al servidor (*fragmentos del código relevante*)
 - origen de los datos consumidos (*breve descripción del uso de la API: qué ofrece y cómo se consume, url y parámetros precisos; en el caso de consumir datos de archivos txt o json contenido y detalle de la estructura de los mismos*)
 - capturas de pantalla demostrativas de la página en ejecución
 - investigación y fuentes consultadas
 - otros aspectos: qué parte de la actividad te ha resultado más laboriosa, qué parte la más sencilla, quedas satisfecho de tu trabajo o lo consideras mejorable (mejoras..), si dispusieras de un plazo de entrega mayor en qué profundizarías...

- **código desarrollado**

Fecha límite de entrega: 28 de enero