# JS en la Web: CRUD con JavaScript asíncrono

## Conclusión 1: Comunicación asíncrona

* Hacer requisiciones utilizando xmlhttprequest.
* Lidias con las promises.
* Hacer una requisición http utilizando xmlhttprequest para buscar todos los clientes del servidor.
* Refactorizar el código utilizando promises mejorando su lectura.
* Utilizar template literals para crear un template html.

## Conclusión 2: Fetch API

## Conclusión 3: Crear y borrar clientes

## Conclusión 4: Editar datos

## Conclusión 5: Async/Await

## Anexos:

* <https://www.npmjs.com/package/json-server>
* <https://chrome.google.com/webstore/detail/json-formatter/bcjindcccaagfpapjjmafapmmgkkhgoa/related?hl=es---------->
* <http://localhost:3000/perfil>
* <https://www.cdmon.com/es/blog/la-ejecucion-de-scripts-esta-deshabilitada-en-este-sistema-te-contamos-como-actuar#:~:text=Para%20poder%20ejecutar%20los%20scripts,scripts%20maliciosos%20en%20el%20sistema>.
* <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Promise>
* <https://app.aluracursos.com/course/js-web-crud-javascript-asincrono/task/77095>
* <https://www.aluracursos.com/blog/empezando-con-fetch-en-javascript?utm_source=gnarus&utm_medium=timeline>

## Código:

## Vocabulario:

* AJAX: JS asíncrono y xml}
* CRUD: Create Read Update Delete (métotodos http)
  + Create – POST
  + Read – GET
  + Update – PUT/PATCH
  + Delete – DELETE
* Backticks
* Callback hell
* Asincronia

## Capturas

|  |
| --- |
|  |