

## Objetivo genérico de la UT:

- Escribir código para comunicarse con el servidor web, definiendo parámetros de petición y de recepción, procesamiento y presentación de la respuesta.

## Objetivo específico de la Tarea

- Realizar peticiones con **Fetch** al tiempo que aprendemos a comunicarnos con distintas APIs sin precisar ninguna librería como axios y jQuery
- Toma de contacto con el uso de **bootstrap** para realizar la maquetación de los datos consumidos



**Fetch** es una API (*Interfaz de programación de aplicaciones*) de JS que permite realizar **peticiones http** para obtener recursos de red. Es una alternativa más moderna al uso de XMLHttpRequest. Recuerda que las peticiones más utilizadas utilizando el protocolo http son: **get, post, put, delete y patch**. En este caso la petición a utilizar es **get**, ya que vamos a consumir/tomar datos de una API

Fetch devuelve una [promesa](#), por tanto utilizaremos:

- ☐ el método **.then()** para indicar la acción a realizar cuando la promesa se cumpla (es decir cuando se obtiene respuesta de la petición realizada) y
- ☐ el método **.catch()** para actuar cuando se rechaza la promesa

## Ejemplo:

```
fetch("/robots.txt")
  .then(function(response) {
    /* Código a realizar cuando se cumpla la promesa */
  })
  .catch(function(error) {
    /* Código a realizar cuando se rechaza la promesa */
  });
```

(\* Si se siguen generando promesas, podemos encadenar varios .then() \*)

## Tarea:

Obtener desde la API pública [JSONPlaceholder](#) los datos de los 10 usuarios que se ofrecen en el end-point **users** y presentarlos en una tabla HTML.

*Nota: los end-points no son otra cosa que las rutas/url a las cuales el servicio consumidor deberá navegar para realizar el request*

☐ sin estilos:

Id	Name	Email
1	Leanne Graham	Sincere@april.biz
2	Ervin Howell	Shanna@melissa.tv
3	Clementine Bauch	Nathan@yesenia.net
4	Patricia Lebsack	Julianne.OConner@kory.org
5	Chelsey Dietrich	Lucio_Hettinger@annie.ca
6	Mrs. Dennis Schulist	Karley_Dach@jasper.info
7	Kurtis Weissnat	Telly.Hoeger@billy.biz
8	Nicholas Runolfsdottir V	Sherwood@rosamond.me
9	Glenna Reichert	Chaim_McDermott@dana.io
10	Clementina DuBuque	Rey.Padberg@karina.biz

☐ el estilo "a pelo":

ID	NOMBRE	EMAIL
1	Leanne Graham	Sincere@april.biz
2	Ervin Howell	Shanna@melissa.tv
3	Clementine Bauch	Nathan@yesenia.net
4	Patricia Lebsack	Julianne.OConner@kory.org
5	Chelsey Dietrich	Lucio_Hettinger@annie.ca
6	Mrs. Dennis Schulist	Karley_Dach@jasper.info
7	Kurtis Weissnat	Telly.Hoeger@billy.biz
8	Nicholas Runolfsdottir V	Sherwood@rosamond.me
9	Glenna Reichert	Chaim_McDermott@dana.io
10	Clementina DuBuque	Rey.Padberg@karina.biz

☐ haz uso de **bootstrap** para obtener este diseño

Id	Name	Email
1	Leanne Graham	Sincere@april.biz
2	Ervin Howell	Shanna@melissa.tv
3	Clementine Bdluch	Nathan@yesenia.net
4	Patricia Lebsack	Julianne.OConner@kory.org
5	Chelsey Dietrich	Lucio_Hettinger@annie.ca
6	Mrs. Dennis Schulist	Karley_Dach@jasper.info
7	Kurtis Weissnat	Telly.Hoeger@billy.biz
8	Nicholas Runolfsdottir V	Sherwood@rosamond.me
9	Glenna Reichert	Chaim_McDermott@dana.io
10	Clementina DuBuque	Rey.Padberg@karina.biz

[Puedes consultar este vídeo para obtener el diseño propuesto](#)

Sube a moodle las soluciones