



## M06. Desenvolupament Web en Entorn Client

### UF4. Comunicació asíncrona client-servidor

#### 4.1 Funcions asíncrones i promeses

#### 4.2 XHR

#### 4.3 Fetch API

#### 4.4 WebSockets

(RA1)

curs 2023-24

Sergi Grau

Daniel Collados

## Descripció.

Es demana realitzar una aplicació que permet obtenir informació de l'API de Marvel,  
<https://developer.marvel.com/>.

Aquesta informació ha de permetre obtenir primer tots els còmics d'un superheroi que introduïm en un camp de cerca. Ens mostra en forma de columnes les portades de tots els còmics i el seu títol. En seleccionar un de concret ens mostra tots els detalls del mateix en un espai lateral.

Aquestes crides a l'API s'han de fer en 3 tecnologies. XHR, Fetch API i WebSockets. Per a fer les crides amb WebSockets caldrà fer en NodeJS un Proxy que faci les crides a Marvel.

## Criteris d'avaluació

La puntuació màxima del projecte és un 10.

La puntuació mínima per a superar-lo es de 5 sobre 10.

## Rúbrica d'avaluació

La rúbrica d'avaluació del projecte és la següent:

Críteris	4	3	2	1
XHR (30%)	La implementació de les crides a l'API és completa. Es fa servir només XHR2. En algun punt s'utilitzen dades binàries.	La implementació de les crides a l'API és completa. Es fa servir només XHR2	La implementació de les crides a l'API és completa. Es fa servir només XHR1	La implementació de les crides a l'API és molt bàsica
Fetch API (30%)	La implementació de les crides a l'API és completa. Es fa servir funcions asíncrones. En algun punt s'utilitzen promeses paral·les i de tipus <i>race</i> .	La implementació de les crides a l'API és completa. Es fa servir funcions asíncrones	La implementació de les crides a l'API és completa. Es fa servir només Promeses, de manera completa amb catch, etc.	La implementació de les crides a l'API és molt bàsica
Web Sockets (40%)	El servidor fa crides REST completes. A més exposa per websockets per a que clients puguin consultar la informació obtinguda. Hi ha un client JS que també utilitza WebSockets, fent ús de la biblioteca Socket.io	El servidor fa crides REST completes. A més exposa per websockets per a que clients puguin consultar la informació obtinguda	El servidor fa crides REST completes.	La implementació del backend de les crides a l'API és molt bàsica

Funcionalitats a implementar.

- El Back-end de l'aplicació per la part de WebSockets ha d'estar desenvolupat amb Node.JS i ExpressJS
- El Front-end de l'aplicació cal desenvolupar-ho amb JS (és opcional utilitzar ES7).
- Les dades del Front-end s'envien mitjançant XHR i Fetchi objectes JSON

Característiques del disseny.

- Cal utilitzar el paradigma de POO.
- El disseny d'interfície web és lliure.
- Les comunicacions C/S utilitzen JSON.
- Es pot utilitzar ExpressJS per a resoldre la part del backend.

Cal lliurar-la abans del dia especificat al l'apartat del classroom de l'assignatura.

S'ha d'entregar en fitxer zip o tar, amb el format

COGNOM\_NOM\_PRACTICA\_ENTORN\_CLIENT\_UF4.zip

- Aquest fitxer contindrà un fitxer en format PDF descrivint les característiques de la implementació realitzada.
- Aquest fitxer zip contindrà tot el codi font, biblioteques emprades i el contingut estàtic de la aplicació web.