Javascript avanzado: programación funcional y Test Driven Development

Taller práctico donde se desarrollarán una serie de funcionalidades para una aplicación javascript existente. Durante el desarrollo del mismo se irán introduciendo distintos conceptos de programación funcional y desarrollo dirigido por test para mejorar la aplicación.

### Objetivo

El objetivo del taller es profundizar en conceptos más complejos de programación, cuya ventaja cuesta ver al principio, y que en general los desarrolladores no aplican perdiendo la oportunidad de escribir código más limpio, expresivo y fiable.

### Dirigido a

El taller está dirigido a personas/profesionales que desarrollan habitualmente en Javascript y que quieren sacarle el 100% al lenguaje. El taller está diseñado para poder reengancharte al siguiente ejercicio si te has perdido en el anterior, aunque son ejercicios donde la base se da por supuesta. No se usará ningún framework avanzado (tipo angularjs o react). Quienes ya hayan realizado el curso “Buenas prácticas en desarrollo con Javascript” pueden considerar éste curso como una segunda parte.

### Temario

* Un poco de ES6 (ECMASCRIPT 2015)
  + Arrow functions
  + Destructuring
  + const/let
  + Módule Import
* Programación con funciones
  + Curry y aplicación parcial
  + Combinar, decorar y pipear
  + Inmutabilidad/pureza/transparencia-referencial
  + Simplifica tu código usando programación funcional
* TDD
  + **Diseño** dirigido por tests: encontrando problemas de diseño en tu código
  + TDD vs Test First vs Unit Testing
  + Code economics: programar para reducir el coste de evolucionar
  + Cómo testear código asíncrono
  + Estrategia de testing: tests visuales,

# Recursos

JavaScript:

* [Funciones](https://elabismodenull.wordpress.com/2016/11/04/programacion-funcional-en-javascript-las-funciones/#more-3474)
* [Map, filter, reduce](https://elabismodenull.wordpress.com/2016/11/10/programacion-funcional-en-javascript-los-metodos-funcionales/)
* [Composición](https://elabismodenull.wordpress.com/2016/12/13/programacion-funcional-en-javascript-la-composicion/)
* [Curry](https://elabismodenull.wordpress.com/2016/12/08/programacion-funcional-en-javascript-la-currificacion/)
* Librerías FP:
  + [RamdaJS](http://ramdajs.com/)
  + [LodashFP](https://github.com/lodash/lodash/wiki/FP-Guide)
* Tooling - librerías:
  + [BabelJS](https://babeljs.io/)
  + [Mocha](https://mochajs.org/), [Should.js](https://shouldjs.github.io/)
  + [NPM](https://www.npmjs.com/), [Yarn](https://yarnpkg.com)
  + [Mozilla MDN](https://developer.mozilla.org/es/docs/Web/JavaScript): [Array](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Array), [Promise](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Promise), ...
  + [Webstorm](https://www.jetbrains.com/webstorm/)
* Libro: [JavaScript Allongé, the “Six” edition](https://leanpub.com/javascriptallongesix/read)
* Libro: [Async JavaScript](https://pragprog.com/book/tbajs/async-javascript)

ES6 - ECMAScript2015

* [ES6 Features](https://codetower.github.io/es6-features/)
* [ECMAScript compatibility table](https://kangax.github.io/compat-table/es6/)
* [es6katas.org](http://es6katas.org/)
* Libro: [Understanding ECMAScript](https://leanpub.com/understandinges6/read)