→| CON JAVASCRIPT

DESARROLLO

[ ]

AQUÍ NUESTRO MANIFIESTO

Classe es Redradix y Redradix es Classe. La una sin la otra no tiene sentido. Una, porque si nuestros cursos y contenidos valen la pena es porque los imparte el equipo de Redradix, curtido a diario en mil batallas y proyectos reales. La otra, porque una de las cosas que hacen que llevemos 10 años al pie del cañón es nuestro carácter didáctico.

Classe es la formación que nos habría gustado tener en nuestros primeros años trabajando en productos digitales. Una formación complementaria para quienes se dedican al desarrollo y diseño digital, que no es teoría ni se queda en ella, sino que se basa en el día a día de las personas que la imparten -profesionales en activo- y en todo lo que han ido aprendiendo para mejorar en sus proyectos y sus carreras.

Classe quiere compartir todo esto en cursos abiertos de acceso libre y también con cursos que se adaptan a las empresas para que mejoren las habilidades de sus equipos para dar el salto que necesitan en sus proyectos.

Classe va de aprender para ser mejor profesional para poder ponerlo en práctica después, incluso si crees que ya lo sabes todo.

HABLEMOS DEL CURSO

Básicamente aprenderás JavaScript en profundidad. Los cimientos sobre los que apoyar conocimientos futuros. Descubrirás que se puede pasar de Angular a React en cuestión de horas si las bases son sólidas. No solo eso, tus compañeros agradecerán a partir de ahora que tu código sea robusto, escalable y, sobre todo, legible.

Este es un curso avanzado para profesionales que han trabajado en desarrollo web al menos durante un año. En remoto para que te puedas organizar mejor, pero sin perder la cercanía del profesor. Está centrado en learn by doing con ejercicios divertidos para hacer y corregir en vivo. Live programming al poder!

- → | 600 EUROS
- → | 30 HORAS
- → | PREWORK Y TRABAJO FINAL
- → | 15 PLAZAS
- → | 80% PRÁCTICO

TEMARIO DEL CURSO

SUJETO A CAMBIOS

(1) TIPOS DE DATOS

- Fundamentos
- Arrays
- Mapas (+ weakmap)
- Sets
- Símbolos
- Proxies
- Objetos
  - Propiedades (descriptors)
  - Getters, setters
  - Prototipos

(2) ORIENTACIÓN A OBJETOS [CLASES]

- Intro a OOP
  - Razón de ser
  - Diccionario de conceptos
- Instancias y envío de mensajes
- This
  - Apply, call
  - Bind
- Constructores
  - New
  - Prototype
  - Herencia
  - Limitaciones
- Clases
  - Syntax sugar
  - Herencia
  - Mejoras sobre constructores
  - Limitaciones
- Decoradores
- Principios de diseño
  - SOLID
  - Expression problem

(3) PATRONES DE PROGRAMACIÓN FUNCIONAL

- Intro a FP
  - Razón de ser
  - Diccionario de conceptos
- Funciones de orden superior
  - Curry
    - CurryRight
    - Autocurry
  - Debounce
  - Throttle
  - Memoize
  - Once
- Operaciones sobre listas
  - Array.prototype
  - Map
  - Each
  - Filter
  - Reduce
  - Iterabas
  - Generadores
- Operaciones sobre objetos
  - Map, mapKeys, mapValues
  - Prop, assoc
- Composición de funciones
  - Compose, pipe
  - Branch
  - Maybe
  - Point free programming
- Datos inmutables

(4) PATRONES DE PROGRAMACIÓN ASÍNCRONA

- Intro
  - Razón de ser
  - El problema fundamental: ejecución desordenada
- Callbacks y CPP
  - Pensar en callbacks
  - Pyramid of doom
  - Sincronización de callbacks
  - Manejo de errores
- Promesas
  - Concepto
  - Explicación de Promise
  - Sincronización de promesas
  - Manejo de errores
  - Promise.all
  - Promise.map
  - Promise.reduce
  - Promise.fromCallback
- Async / await
  - Generadores + promises = corrutinas
  - Sincronización de corrutinas
  - Manejo de errores
  - Iteración
  - Limitaciones
  - Async / await

MÓDULOS OPCIONALES

Si quieres profundizar, podemos complementar el curso con módulos de 4 horas:

- Módulo opcional: `Lenguajes tipados con Typscript`
- Módulo opcional: `Estrategias de Testing`
- Módulo opcional: `GraphQL`
- Módulo opcional: `Mejoras en ECMAScript`

Pregúntanos y te contamos sobre estas ampliaciones.