

LA FORMACIÓN ES LA CLAVE
DEL ÉXITO

Guía del Curso

IFCD89 JAVASCRIPT AVANZADO

Modalidad de realización del curso: [Online](#)

Titulación: [Diploma acreditativo con las horas del curso](#)

OBJETIVOS

CONTENIDOS

UNIDAD DIDÁCTICA 1. INTRODUCCIÓN

1. Potenciación de las generaciones de codificación limpia y funcional en desarrolladores con experiencia anterior en JavaScript.
2. Estructuración con facilidad de un mejor código y en módulos diferenciados.
3. Exploración de patrones y prácticas vigentes de diseño en la industria actual y cómo usarlos.
4. Aprendizaje para la realización de pruebas unitarias.
5. Control y gestión adecuado de errores en JavaScript.

UNIDAD DIDÁCTICA 2. CONCEPTOS INICIALES PARA LA GENERACIÓN DE CÓDIGO LIMPIO (CLEAN CODE) CON JAVASCRIPT

1. Variables.
2. Qué es Lexical Scope (static scope). Uso de la instrucción "let".
3. Funciones.
4. Todo sobre la palabra clave "this".

5. Objetos and Clases.
6. Prevención de errores y formato homogéneo del código gracias a la herramienta Linter.
7. Actividad de evaluación de los conocimientos adquiridos por el alumno.

UNIDAD DIDÁCTICA 3. CONCEPTOS INICIALES DE GENERACIÓN DE CÓDIGO FUNCIONAL CON JAVASCRIPT

1. ¿Qué la programación funcional?
2. Las Funciones Puras en JavaScript.
3. El correcto uso del método Object. assign y el operador Spread.
4. Optimización de bucles en JavaScript.
5. Reducción de arrays.
6. Filtrado en arrays.
7. Actividad de evaluación de los conocimientos adquiridos por el alumno

UNIDAD DIDÁCTICA 4. MÓDULOS EN JAVASCRIPT

1. Reutilización de código con patrón Módulo (Module Pattern)
2. Utilización del gestor de paquetes Node Package Manager (NPM)
3. El proyecto CommonJS.
4. La API de definición asíncrona de módulos (AMD)
5. Módulos de ES6.
6. Eliminación de código inactivo (Tree-Shaking) con Webpack.
7. Actividad de evaluación de los conocimientos adquiridos por el alumno.

UNIDAD DIDÁCTICA 5. PATRONES Y PRINCIPIOS DE DISEÑO

1. Qué es el principio de responsabilidad única (Single Responsibility Principle)
2. El principio Open/Closed.
3. El principio de sustitución Liskov.
4. El principio de segregación de la interfaz.
5. El principio de Inversión de Dependencias (Dependency Inversion)
6. El patrón Singleton.
7. El patrón Observer.

8. El patrón Estrategia (Strategy)
9. Actividad de evaluación de los conocimientos adquiridos por el alumno.

UNIDAD DIDÁCTICA 6. PRUEBAS DEL CÓDIGO GENERADO

1. Qué son y qué no son y la importancia de las pruebas unitarias.
2. Instalación y uso del framework de testing Jasmine.
3. Desarrollo guiado por pruebas de software, o Test-Driven Development (TDD)
4. Ejercicio guionizado para consolidar formación del alumno.

UNIDAD DIDÁCTICA 7. MANEJO DE ERRORES

1. Estudio de buenas prácticas.
2. Detección de errores en operaciones asíncronas (Objeto Promise)
3. Estrategias y recomendaciones en el manejo de errores.
4. Generación de logs con JavaScript.
5. Actividad de evaluación de los conocimientos adquiridos por el alumno.



C/ San Lorenzo 2 - 2
29001 Málaga



Tlf: 952 215 476
Fax: 951 987 941



www.academiaintegral.com.es
E-mail: info@academiaintegral.com.es