



Gracias a



OpenWebinars certifica que

**José Carlos García-Ramos Medina**

Ha superado con éxito

**Programador Full Stack MEAN**

Duración de la ruta

**26 horas**

Fecha de expedición

**09 abril 2020**

A handwritten signature in black ink, likely belonging to Manuel Agudo, the CEO of OpenWebinars.

CEO de OpenWebinars

**Manuel Agudo**

# Programador Full Stack MEAN

## 1. Curso de HTML5 y CSS3

### 1. Introducción

Presentación

¿Qué es HTML?

¿Qué es CSS?

Entorno de trabajo

Estructura del resto del curso

Test de autoevaluación

### 2. Documentos HTML

Evolución de HTML hasta HTML5

Etiquetas, atributos y comentarios

Estructura de una página web

Cabecera de una página web

Test de autoevaluación

### 3. Añadiendo contenido. Etiquetas Básicas

Etiquetas relacionadas con texto

Imágenes, enlaces y rutas

Práctica: Imágenes y enlaces

Listas

Práctica: Listas

Tablas

Bordes de tablas

Práctica: Tablas.

Test de autoevaluación

#### **4. Formularios en HTML**

Formularios HTML

Tipos de inputs en formularios

Práctica: Formularios

Test de autoevaluación

#### **5. Más etiquetas accesibilidad**

Etiquetas multimedia

Etiquetas semánticas

Otras etiquetas

Accesibilidad en HTML

Test de autoevaluación

#### **6. CSS**

Evolución de CSS hasta CSS3

Añadir hojas de estilo a nuestro HTML

El modelo de caja. Etiquetas en línea y en bloque

Selectores CSS

Propiedades interesantes

Práctica: Selectores

Pseudoselectores

Práctica: Pseudoselectores

Estilos por defecto y reseteo de propiedades

Prefijos específicos para navegadores

Optimización de CSS

Herramientas relacionadas con CSS

Test de autoevaluación

## **2. Curso de JavaScript para principiantes**

### **1. Introducción**

Presentación

¿Qué es JavaScript?

Estándares de JavaScript

Utilizando la consola

Creando variables

### **2. Tipos de datos primitivos**

Numbers (Parte I)

Numbers (Parte II)

Práctica: Numbers

Strings

Práctica: Strings

Boolean

Práctica: Boolean

Null y undefined

### **3. Funciones**

Funciones (Parte I)

Funciones (Parte II)

### **4. Aplicando lógica con JavaScript**

Aplicando lógica con JavaScript

## **3. Curso de JavaScript intermedio**

### **1. Introducción**

Presentación

### **2. Tipos de datos avanzados**

Objetos

Trabajando con objetos

Práctica: Objetos

Objeto Date

Objeto Math

Objeto Array

Bucles

Práctica: Arrays

### **3. DOM**

¿Qué es el DOM?

Selectores

Window y Document

Insertar y eliminar nodos

### **4. Eventos**

¿Para qué sirven los eventos?

Eventos de teclado, ratón y navegador

Lógica de nuestro proyecto

### **5. Expresiones regulares**

¿Qué son las expresiones regulares?

Creación de una expresión regular

## **4. Curso de JavaScript avanzado**

### **1. Introducción**

Presentación

### **2. Estructuras de datos**

Tipado débil y coerción

Expresiones y operadores

Estructuras de datos avanzadas

### **3. Promesas y asincronía**

Trabajando con el objeto Promise

Callbacks y errores

Uso de Async/Await

### **4. Programación funcional**



Funciones puras e idempotencia

Funciones de orden superior y closures

Metodología funcional

POO vs. Programación funcional

## **5. Módulos en Javascript**

¿Qué es un módulo?

Patrón modular

Módulos de ES6

## **5. Curso de Git.**

### **1. Introducción**

Presentación

Conceptos básicos

### **2. Herramientas de trabajo**

Instalación y configuración

Servidor GIT básico e introducción a GitLab

### **3. Organización del código fuente y otros elementos del proyecto**

Creación de un repositorio

Aprobando cambios (Parte I)

Aprobando cambios (Parte II)

Trabajo remoto

Deshacer cambios

Reparando conflictos de unión con merge

### **4. Flujos de trabajo**

Experimentando con ramas (Parte I)

Experimentando con ramas (Parte II)

Experimentando con ramas (Parte III)

Encontrando errores

Git Blame

Siendo "quisquilloso" con Git

Reescribir la historia de un proyecto con Rebase

Etiquetas

Moverse adelante y atrás entre estados de un repositorio git

Git Stash

## **5. Repositorios Corporativos**

Ventajas de usar un servicio en la nube: GitHub y GitLab

Flujo DevOps

## **6. Curso de Angular 11 para principiantes**

### **1. Introducción**

Presentación

Preparación del entorno

Instalación de nvm y Node.js

Requisitos previos: TypeScript (Parte I)

Requisitos previos: TypeScript (Parte II)

Preparando VSCode

### **2. Conociendo Angular**

¿Qué es una SPA? (Single-Page Application)

¿Por qué utilizar Angular?

Instalando Angular y conociendo Angular CLI

Crear y ejecutar nuestra primera aplicación

### **3. Arquitectura Angular**

Estructura básica en Angular

Partes de un proyecto creado en Angular

Estructura de trabajo

### **4. Módulos en Angular**

Metadatos de un NgModule

### **5. Componentes, el corazón de Angular**

Estructura de un componente

Ciclo de vida

Metadatos de un @Component

Plantillas y vistas en Angular

Práctica: Crear nuestros primeros componentes

Data-Binding en Angular (Parte I)

Data-Binding en Angular (Parte II)

Directivas: Manejo del DOM

Variable #ref para referencias en HTML

Práctica: Manejar datos en nuestros componentes

## **6. Enrutado en nuestro proyecto**

Inicializando la navegación

Creando rutas en nuestro proyecto

Añadiendo RouterOutlet para navegar entre vistas

Práctica: Navegando en nuestro proyecto

## **7. Servicios e inyector de dependencias**

Inyector de dependencias

Servicios

Proporcionando servicios a nuestra aplicación

## **8. Peticiones API REST**

Añadiendo la librería HttpClient

Introducción a los Observables

Tipos de peticiones HTTP

Práctica: Petición HTTP a REST API

## **9. Introducción a Formularios**

Formularios en Angular y formularios Template-Driven

Validación de formularios Template-Driven

Práctica: Creando un formulario de login

## **7. Curso de Angular: Ampliando conceptos**

### **1. Introducción**

Presentación





Preparando el entorno de trabajo

## **2. Ampliando conocimientos de componentes**

Tipos de relaciones entre componentes

Relación a través del decorador Input ()

Creando eventos con Output ()

Relación mediante el uso de servicios

Práctica: Creando usos de relación entre componentes

Utilizando NgClass y NgStyle

Directiva NgSwitch

Operaciones especiales con Pipes

Práctica: NgClass y Pipes

## **3. Ampliando conceptos de Angular**

Añadiendo parámetros a nuestras rutas

Rutas anidadas

CanActivate

Práctica: Crear nuevas rutas

Práctica: Añadiendo seguridad

Sub-módulos y componentes

Creando rutas para nuestros submódulos

Práctica: Transformar nuestra aplicación en multimodular

## **4. Formulario Reactivos**

Creación de un formulario reactivo

Validación de formularios reactivos

Práctica: Creación de un formulario reactivo

## **5. Material Design**

Instalar Material Design

¿Cómo usar Material Design?

Práctica: Añadir estilo a nuestra aplicación

## **6. Últimos apuntes**

¿Cómo hacer Debug?

Compilar nuestra aplicación

## **8. Curso de buenas prácticas en Angular**

### **1. Introducción**

Presentación del profesor y curso

Introducción

### **2. Generando una app con Angular CLI**

Angular CLI: Archivo de configuración

Tslint para hacer tu aplicación mantenible

### **3. Refactorizando una aplicación compleja I**

Nombres de ficheros

Principio de responsabilidad única

Funciones pequeñas

Pensando en componentes

Estructura de carpetas

### **4. Refactorizando una aplicación compleja II**

Refactorización de componentes

Refactorización de servicios

Refactorización del core

Refactorización de controles UI

### **5. Refactorizando una aplicación compleja III**

Lazy loading de módulos

Puesta en marcha y solución de errores

### **6. Optimización y despliegue**

Optimización de bundles

Compilación para producción

## **9. Curso de MongoDB: Creación y gestión de bases de datos NoSQL**

### **1. Introducción**

Presentación

¿Qué son las bases de datos SQL?

Características de las bases de datos noSQL

¿Qué es el formato JSON?

Comparativa entre bases de datos SQL y noSQL

## **2. Primeros pasos con MongoDB**

Instalación de MongoDB

Trabajando con MongoDB desde la consola

Creación y gestión de bases de datos

Colecciones y documentos

Tipos de datos

Práctica: Creación de una base de datos

## **3. Operaciones con datos I (CRUD)**

Insertar

Eliminar

Actualizar

Práctica: Insertar, eliminar y actualizar

## **4. Diseñando el modelo de datos**

Concepto de Schemaless

Diseño I: Documentos embebidos

Diseño II: Documentos referenciados

## **5. Operaciones con datos II (CRUD)**

Consultas con tipos de datos simples

Consultas avanzadas: Arrays

Consultas avanzadas: Documentos embebidos

Práctica: Consultas avanzadas

## **6. Mejorando el rendimiento**

¿Qué son los índices?

Índices simples

Índices compuestos

Índices únicos

Práctica: Crear y eliminar índices