

Descripción del curso sobre Angular

Visión general

Este curso requiere que los alumnos tengan conocimientos de HTML5, CSS3 y conocimientos básicos de Javascript. Se les ofrecerá los conocimientos necesarios para crear sitios web utilizando el Framework de Angular creado por Google, utilizando uno de los lenguajes con mayor futuro en el desarrollo de sitios web.

Duración

40 hrs

Asistentes

- Profesores de media superior y superior que imparten clase de informática o asignatura relacionada.
- Alumnos de media superior y superior que deseen adquirir los conocimientos necesarios para ser un desarrollador web Frontend.
- Profesionistas que deseen incrementar sus conocimientos en el desarrollo de sitios web profesionales.

Requisitos previos

HTML5, CSS3 y conocimientos básicos de Javascript

Próximos cursos sugeridos

Symfony, Django o algún lenguaje de programación

Temario

Introducción a TypeScript y ES6

- Introducción a TypeScript
- Demostración de TypeScript
- Configuración de TypeScript
- Variables let y const
- Introducción a los tipos de datos
- Excluir archivos a traducir
- Template literales del ES6
- Funciones: Parámetros opcionales, obligatorios y por defecto
- Funciones de Fecha
- Desestructuración de Objetos y Arreglos
- Promesas
- Promesas y su tipado en TypeScript
- Interfaces de TypeScript
- Introducción a las Clases de la POO
- Definición de una clase básica en TypeScript
- Constructores de una clase en TypeScript
- Importaciones * URL
- Decoradores de Clases
- Tipado del retorno de una función

Aplicación Hola Mundo

- Introducción a los componentes y directivas estructurales.
- Nuestra primera interacción en Angular
- Nota de Actualización del AngularCLI
- Creando un entorno local de Angular
- Estructura del proyecto
- Utilizando Bootstrap 4
- templateUrl: Separando el HTML el componente
- Creando el footer.component
- Estructura del body component
- Directivas estructurales: *ngFor y el *ngIf

Aplicación de una sola página (SPA)

- Iniciar el proyecto - SPA
- Creando la estructura de nuestro proyecto
- Instalando el bootstrap cuando usamos el Angular-CLI
- Configurando el navbar y otros componentes
- Rutas en Angular
- RouterLink y RouterLinkActive - Completando las rutas
- Componente Heroes - diseño
- Introducción a los Servicios
- Creando nuestro primer servicio - HeroesService
- Página de Heroes - Diseño con *ngFor
- Rutas con parametros - Router
- Recibiendo parámetros por URL - ActivatedRoute
- Pipes: Transformación visual de la data.
- Buscador de Héroe
- Plus: Mostrando un mensaje cuando no hay resultados.
- @Input - Recibir información de un componente padre a un hijo
- @Output - Emitir un evento del hijo hacia el padre
- Arreglar detalles de la búsqueda

Pipes

- Inicio del proyecto y la introducción a los Pipes
- Pipe: Slice
- Pipe: Decimal
- Pipe: Percent - Porcentajes
- Pipe: Currency - Moneda
- Pipe: Json
- Pipe: Async
- Pipe: Date - Fecha
- Registrar otros idiomas
- Pipes personalizados: Capitalizar palabras
- Pipe Personalizado: Domseguro

Servicios – SpotiApp

- Sitio web de developer de Spotify
- Iniciando el proyecto - SpotiApp
- Creando las rutas de nuestra aplicación
- Introducción a las peticiones HTTP
- Actualización - Token para uso de servicios Spotify
- HTTPClient - Service: Conectándonos a Spotify
- Consumiendo información del servicio de Spotify
- Componente de Búsqueda de artistas
- Operador Map de los Observables
- Centralizar las peticiones hacia Spotify
- Verificación de imagen y pipe para manejar las imágenes
- Componente de tarjetas
- Creando un loading component
- Página del artista, nueva ruta, parámetro por url y servicio
- Obtener artista de Spotify
- Servicio: Top-tracks
- Widgets de Spotify
- Manejo de errores de un observable
- Generar Token de Spotify de forma automática

IONIC

- ¿Qué es ionic?
- Creando el esqueleto de nuestra aplicación
- Componente de Pendientes, estructura y Tabs
- Servicio y clases de nuestra lista de deseos
- Pantalla de pendientes - diseño y documentación de ionic
- Navegación entre pantallas
- Diseño de la página de agregar
- Alert Controller - Agregar una lista a nuestro servicio
- Localstorage - Hacer persistente la información
- Funcionalidad de la pantalla para agregar tareas a la lista
- Eliminar items de la lista de deseos
- Editar elementos de otras listas
- Módulo de componentes y listas component
- Componente listas
- Eliminar una lista
- Pipes impuros
- Editar el título de la lista

Componentes, directiva de atributos, ciclo de vida

- Creando el proyecto de esta sección - Misceláneos
- ngStyle y su uso con directivas de atributos.
- Aplicando CSS a un solo componente
- ngClass - Agregando clases de estilos a nuestros elementos HTML
- Usando procesos asíncronos con indicadores de usuario
- Directivas personalizadas
- ngSwitch - Múltiples opciones con una sola decisión
- Rutas y Rutas Hijas
- Rutas Hijas
- Separar las rutas hijas a un archivo específico
- Obteniendo parámetros del padre, desde las rutas hijas
- Ciclo de vida completo de un componente
- Demostración del ciclo de vida de un componente

Autenticación

- Aplicación con autenticación Auth0 - Inicio del proyecto
- Creando la interfaz de nuestra aplicación
- Comenzando con Auth0 - Componentes y servicios
- Configuraciones en Auth0 dashboard y uso del servicio
- Conectar la los botones con los métodos de Auth0
- Servicio de bloqueo - CanActivate - AuthGuard
- Obteniendo el perfil del usuario que inició sesión

Login tradicional y manejo de tokens - Firebase

- Inicio de proyecto - LoginApp
- Crear modelo para el manejo de los usuarios
- Conectar el formulario de registro con una instancia del modelo de usuario
- Validar la información antes de enviarla a un servidor
- Mostrar errores en pantalla
- Firebase y servicios REST
- Registrar un nuevo usuario
- Login de usuarios
- Guardar Token en el LocalStorage
- SweetAlert2 - Mostrar notificaciones al usuario
- Recordar usuario
- Guard para proteger la ruta si no se está autenticado
- Logout - Cerrar sesión

- Mejorar la validación del token

Formularios

- Inicio del proyecto de formularios
- Creación de componentes - Formularios y Rutas
- Template: ngModel - ngSubmit - Referencias locales a los elementos HTML
- Template: Obteniendo la información del formulario
- Template: Validaciones independientes y cambio de estilo
- Template: Mostrar errores y mensajes
- API de países del mundo
- Servicio Pais - Obtener información de países
- Template: Select y sus validaciones
- Template: Uso de radio buttons
- Reactivo: Aproximación de formularios utilizando código
- Reactivo: Creación del formulario - FormGroup
- Reactivo: Validaciones síncronas
- Reactivo: Validaciones y HTML
- Reactivo: Agrupaciones de los objetos - formGroupName
- Reactivo: Validación visual de los campos anidados
- Reactivo: Reset y carga de la data inicial
- Reactivo: Arreglos de FormControl - FormArray
- Reactivo: Añadir y borrar FormControls de forma dinámica
- Reactivo: Validaciones personalizadas
- Reactivo: Validar que el password2 sea igual al password1
- Reactivo: Validadores asíncronos
- Reactivo: Detectar cambios en los valores, estado del formulario o controles

CRUD – Firebase

- Inicio del proyecto - CRUD - Firebase
- Diseño de la página de héroes
- Modelo para manejar Héroes y formulario de edición
- Configurando Firebase como backend REST
- HTTP - POST - Creando un nuevo registro en Firebase
- HTTP - PUT - Actualizar el registro
- Información del estado de las peticiones
- HTTP - GET - Obtener un listado de todos los héroes
- HTTP - GET - Obtener un nodo específico
- HTTP - DELETE - Para eliminar registros
- Maquillaje para nuestra pantalla de héroes

AngularFire2

- Iniciando el proyecto - FireChat
- Instalación del AngularFire2 - Uso de la librería
- Componente del Chat
- Servicio para controlar las acciones del chat
- Cargar y agregar mensajes al chat
- Desplegar mensajes en la caja del chat correctamente
- Autenticación con Google
- Autenticación con Twitter
- Toques finales a nuestro chat

Themoviedb

- Crear una cuenta en Themoviedb
- JSONP - Peticiones cross domain
- ¿Cómo usar los servicios de Themoviedb?
- Creación de los componentes de la aplicación
- Diseño de la pantalla HOME y uso de servicios

- Pipe para controlar las imágenes de las películas
- Enviar y recibir información de un componente a otro
- Componente de búsqueda
- Búsqueda del Navbar
- Pantalla de película
- Detalles finales de la aplicación de películas

Uso de Google Maps

- Inicio del proyecto - GoogleMap + VSC
- Utilizar componentes de Angular Material
- Módulos personalizados
- Componente del mapa y su diseño
- Mostrando un mapa de Google
- Creando marcadores de forma dinámica
- Guardando los marcadores en el LocalStorage
- Borrar marcadores del mapa
- SnackBar - Retro alimentación en pantalla
- Dialog - Mostrar un modal para la edición del marcador
- Pantalla para editar el marcador
- Actualizar marcador con la información del dialog

API de YouTube

- Enlaces a la API de YouTube
- Inicio del proyecto - YouTubeApp
- Configurar el API de YouTube - Postman
- Servicio para obtener videos de YouTube
- Extraer datos y el tipado de la respuesta
- Mostrando los videos en tarjetas
- Mostrar un modal con el video
- Cargar siguientes videos

Desplegando las aplicaciones a producción

- Generando la carpeta dist (No es para producción)
- Generando el archivo de distribución para producción
- Generar una aplicación para desplegar en un servidor Apache
- Desplegando la aplicación en un hosting apache

Fileupload hacia Firebase

- Inicio del proyecto
- Componentes y rutas
- Modelo de datos
- Configuración de Firebase y Servicio de carga
- Componente de carga - HTML
- Directiva del DropZone
- Validaciones del archivo desde la directiva
- Obtener archivos y validarlos
- Mostrar los archivos en pantalla
- Definiciones de Firebase
- Cargar imágenes a Firebase - Storage
- Mostrar las imágenes subidas a Firebase

Gráficas dinámicas en Angular

- Inicio del proyecto – Gráficos Dinámicos
- Instalaciones necesarias - ng2-charts
- Gráfico de Líneas
- Gráfico de Barras
- Gráfico de Donas

- Gráfico de Radar

Virtual Scroll y Drag-Drop

- Inicio de proyecto - Virtual Scroll y Drag & Drop
- Virtual Scroll
- Mover al inicio, fin y a la mitad el ViewPort
- Drag básico
- Drag & Drop de una lista ordenada
- Más información sobre el Virtual Scroll y Drag & Drop de Angular

Firestore, Firebase Functions

- Instalaciones necesarias
- Inicio de proyecto - Game of the Year
- Creación de páginas y componentes
- Configuración del gráfico inicial
- Cambiar valores de la gráfica de forma aleatoria
- Firebase y Firestore - Nuestra base de datos
- Cloud Functions - Hola Mundo
- Firebase: Credenciales para trabajar local y remotamente
- Cloud Function: GET - Obtener colecciones
- Servidor Express dentro de Firebase Cloud Functions
- Post: Incrementar en 1 el voto del juego
- Firebase deploy: Subir las nuevas funciones
- Angular: HTTP GET - Obtener y mostrar los juegos
- Mostrar los juegos en el HTML
- Angular: Votar por el juego del año
- Manejo de errores en la petición
- Angularfire - Obtener información en tiempo real
- Gráfica en tiempo real
- Desplegar la aplicación de Angular a Firebase Hosting

Para buscar y registrarse en los eventos programados visite el calendario publicado en www.colegioamatulli.mx