CURSO de Especialización en
DESARROLLO WEB
(Full Stack)

MÓDULOS	LARAVEL	3 semanas
	NODE.JS CON NESTJS	4 semanas
	ANGULAR BASICO - AVANZADO	4 semanas
	VUE.JS	3 semanas
	DESPLIEGUE, CI/CD Y SERVICIOS CLOUD	2 semanas
Requisitos	Conocimientos en Programación orientada a objetos, PHP, HTML, CSS y JavaScript	
Duración	4 MESES Lunes – Miércoles – Viernes, 2 hrs por clase	

LARAVEL

Contenido

1.1. ¿Qué es Laravel y como lo puedo utilizar?
1.2. Instalación y configuración de Laravel
  1.2.1. Requerimientos mínimos
  1.2.2. Instalación vía Composer
1.3. Estructura del Proyecto Laravel
  1.3.1. Configuraciones Fundamentales
1.4. Consola Artisan
1.5. Enrutamiento
1.6. Modelado de datos
  1.6.1. Query Builders
  1.6.2. Migraciones
  1.6.3. Seeders
  1.6.4. Eloquent ORM
  1.6.5. Relaciones entre tablas
1.7. Vistas
  1.7.1. Inertia
  1.7.2. Typescript
  1.7.3. TailwindCSS 4
  1.7.4. Vue 3 Composition API
  1.7.5. Directivas Vue

NODEJS

con base en TypeScript con PostgreSQL y NestJS

Contenido

1. Bases de TypeScript
   - Instalación
   - El compilador
   - Los tipos en TypeScript
   - Clases e Interfaces
2. NPM
   - Comandos básicos
   - Las dependencias en NPM
   - Package.json
   - Los @types para TypeScript
3. NestJS
   - Instalación y creación de proyecto
   - Manejo de módulos
   - Manejo de controladores
   - Manejo de servicios
4. Instalación del controlador de PostgreSQL para NodeJS y TypeORM
   - Instalación y creación de nuestra BBDD
   - El modelo relacional
   - Entidades
   - Relaciones
5. Servicios
   - Injección de dependencias (repositorio)
   - Métodos de repositorio
   - Query builders
   - Consultas SQL directas
6. Módulos static y Modulo config

- Configuración de variables de entorno
- Archivos públicos con static

7. Manejo de archivos en NestJS
   - Decoradores para el manejo de archivos
   - Validaciones
   - Servicio para el almacenamiento de archivos
8. Manejo de errores
   - Manejo mediante try catch
   - Clases para el manejo de errores
9. Validaciones
   - Class validator
   - Class transformer
   - Validaciones personalizadas
10. Controladores
    - Data Transfer Object
    - Manejo de decoradores
    - Agregar OpenAPI (Swagger)

ANGULAR

Contenido

1. Bases de Angular
   - Conceptos básicos
   - Instalación y creación de un primer proyecto
   - Manejo de Angular CLI
   - Entendiendo nuestro primer proyecto
2. Enrutamiento
   - Definición y manejo de rutas
   - Rutas mediante lazy load
   - Parámetros y parámetros de consulta
   - Guardianes de rutas
3. Componentes Standalone
   - Creación de componentes
   - Partes de un componente
   - Directivas principales
   - Eventos
4. Manejo de Formularios
   - ReactiveFormModule
   - FormBuilder y FormGroup
   - Validación de formularios
   - Validaciones personalizadas
5. Variables de entorno en Angular
   - Creación mediante Angular CLI

- Creación de diferentes entornos
- Ejecución y compilación en base a un entorno definido

6. Manejo de servicios
   - Servicios para centralizar información
   - Modulo HTTP
   - Servicios para comunicarse con un servidor REST API
   - Manejo de servicios con los operadores RXJS
   - Subject
   - Observer, Observable, subscription
7. Manejo de errores
   - Manejo de error al suscribirse
8. Manejo de archivos en Angular
   - Manejo de archivos
   - Validaciones para archivos
9. Component Lifecycle
   - ngOnInit
   - ngOnChange
   - ngOnDestroy
10. Input y Output
    - Modularizar componentes con entradas y salidas
11. Manejo de Pipes
    - Conceptos básicos de los Pipes
    - Manejo de Pipes
    - Creación de pipe personalizado
    - Manejo desde un componente
12. Interceptores
    - Creación de un interceptor
    - Configuración de interceptores
13. Manejo de Guards
    - Conceptos básicos
    - Creación de Guards y tipos de Guards
    - Los Guards y las rutas
14. PrimeNG y PrimeFlex
    - Instalación y configuración
    - Diseño con PrimeNG
    - Manejo de la disposición mediante PrimeFlex
    - Manejo de componentes PrimeNG

Curso Vue JS

CONTENIDO

1. Reforzamiento de Javascript

- 1.1 Bases de Javascript
- 1.2 Let vs Var vs Const
- 1.3 Template Literals
- 1.4 Object Literal
- 1.5 Arrays
- 1.6 Arrow functions
- 1.7 Desestructuracion de objetos
- 1.8 Desestructuracion de Arreglos
- 1.9 Importaciones y exportaciones
- 1.10 Promesas
- 1.11 Axios
- 1.12 Async – Await
- 1.13 Ternarios y null check

2. Introducción a Vue JS
   - 2.1 Hola Mundo VueJS
   - 2.2 Representacion declarativa
   - 2.3 Estado del componente – Data
   - 2.4 Introducción a los eventos
   - 2.5 Directiva v-for
   - 2.6 Directiva v-model
   - 2.7 Directiva v-bind
   - 2.8 Modificadores de eventos
   - 2.9 Directivas v-if y v-show

3. Vue + Vite – Scaffolding Application (Option API y Composition API)
   - 3.1 Inicio de proyecto
   - 3.2 Estructura de directorios generada por defecto
   - 3.3 Mi primero componente (Single File Component)
   - 3.4 Componentes modulares
   - 3.5 Option API y Composition API
   - 3.6 Entendiendo el Option API (data, methods, computed, props)
   - 3.7 Entendiendo composition API
   - 3.8 Lifecycle hooks (en Option API y Composition API)
   - 3.9 Ref
   - 3.10 Propiedades computadas – Computed
   - 3.11 Properties – props
   - 3.12 Define props
   - 3.13 Define emit

4. Axios en Vue
   - 4.1 Instalar Axios
   - 4.2 Configuraciones generales de Axios
   - 4.3 Consumir servidor REST API con Axios desde componente

4.4 Manejo de errores con try catch
4.5 Interceptores en Axios

5. Manejo de validaciones
   5.1 Entendiendo las validaciones en Vue
   5.2 Vee validate y Yup
   5.3 Validaciones personalizadas

6. Vue Router
   6.1 RouterLink
   6.2 RouterView
   6.3 Definición y manejo de rutas
   6.4 Definición de rutas mediante (Lazy Load)
   6.5 Parametros y parámetros de consulta
   6.6 Rutas hijas
   6.7 Guard – Proteccion de rutas

7. Pinia
   7.1 Concepto de Pinia y su uso en Vue
   7.2 Creación estado centralizado o almacen (store)
   7.3 Propiedad “state”
   7.4 Actions
   7.5 Getters
   7.6 Su llamada y uso en componentes
   7.7 Como puedo obtener información de mi “store”
   7.8 Como puedo establecer información en mi “store” desde el componente
   7.9 Uso aplicado desde un componente que usa Option API y Composition API

Curso Despliegue, CI/CD y Servicios Cloud

CONTENIDO

1. Contenerización con Docker y Docker Compose

   - Conceptos fundamentales: Imágenes vs. Contenedores.
   - Ventajas frente a máquinas virtuales en entornos web.

- Dockerizar Backend y Frontend:
   - Creación y optimización de Dockerfile para NestJS (uso de builds multietapa para optimizar tamaño).

- Creación de Dockerfile para Laravel y configuración del servidor web (Nginx).
- Empaquetado y servido de aplicaciones frontend (Angular y Vue.js) mediante servidor Nginx en contenedor.

- Orquestación con Docker Compose:
  - Definición de servicios web y bases de datos (PostgreSQL / MySQL).
  - Manejo de volúmenes persistentes y redes internas.
  - Gestión de variables de entorno seguras dentro de los contenedores.

2. Control de Versiones Avanzado y CI/CD con GitHub Actions

- Flujos de trabajo con Git en entornos profesionales:
  - Buenas prácticas de ramas (main, develop, feature).
  - Gestión de *Pull Requests* y revisiones de código.
- Automatización con GitHub Actions:
  - Introducción a la Integración Continua (CI) y Despliegue Continuo (CD).
  - Creación de Workflows (.github/workflows).
  - Automatización de pruebas, *builds* y validación de código con cada push.
  - Gestión de credenciales y secretos (GitHub Secrets).

3. Despliegue Cloud en Producción

- Despliegue de Frontend:
  - Publicación automatizada de aplicaciones en Vercel o Render conectadas al repositorio de GitHub.
- Despliegue de Backend y Base de Datos:
  - Configuración de PostgreSQL / MySQL gestionado en la nube (Supabase / Render).
  - Despliegue de APIs (NestJS / Laravel) en plataformas PaaS (Render / Railway) o máquinas virtuales (AWS EC2).
- Configuración de Red y Seguridad Final:
  - Manejo de CORS (Cross-Origin Resource Sharing) entre el frontend y backend en producción.

- Certificados SSL/HTTPS y mapeo de dominios.
- Monitoreo básico y revisión de logs en la nube.

Preguntas Frecuentes – Curso de Desarrollo Web Full Stack

1. ¿Qué conocimientos previos necesito para tomar el curso?
Se recomienda tener conocimientos básicos de **HTML, CSS, JavaScript y Programación Orientada a Objetos (POO)**, además de nociones de **bases de datos y SQL**. No es necesario ser experto, pero sí haber tenido algún contacto con la programación.

---

2. ¿A quién está dirigido este curso?
A estudiantes, desarrolladores o profesionales que deseen **especializarse en desarrollo web moderno full stack**, aprendiendo a construir aplicaciones completas con tecnologías actuales tanto del lado del servidor como del cliente.

---

3. ¿Qué aprenderé al finalizar el curso?
Podrás desarrollar **aplicaciones web completas** desde el backend hasta el frontend, utilizando frameworks de alto nivel como **Laravel, NestJS, Angular y Vue.js**, implementando autenticación, consumo de APIs, validaciones, enrutamiento y más.

---

4. ¿Cuánto dura el curso y cuál es la modalidad?
El curso tiene una duración total de **4 meses**, con clases tres **veces por semana (lunes, miércoles y viernes)**.
Cada módulo se enfoca en una tecnología específica y combina teoría con práctica.

---

5. ¿Debo instalar algún programa antes de comenzar?
Sí, necesitarás herramientas como **VS Code**, **Node.js**, **Composer**, **Git**, y un **gestor de base de datos** (por ejemplo PostgreSQL o MySQL).
Durante las primeras clases se guía paso a paso la instalación y configuración de todo el entorno.

---

6. ¿Recibiré material de apoyo o acceso a ejercicios?
Sí. Se entregan **archivos de práctica, apuntes digitales y ejemplos de código** para que puedas repasar y practicar fuera de clase, todo desde la plataforma del campus.

---

7. ¿Se realiza algún proyecto práctico durante el curso?
Sí. Cada módulo incluye **ejercicios aplicados** y al final se propone un **proyecto integrador full stack**, donde el estudiante crea una aplicación completa utilizando las tecnologías aprendidas.

---

8. ¿Qué diferencia tiene este curso frente a otros?
Este programa combina **cuatro de los frameworks más demandados del mercado** (Laravel, Nest.JS, Angular y Vuejs) en un solo curso, garantizando una **formación completa y actualizada** para el desarrollo web moderno.

---

9. ¿Entregan certificado al finalizar el curso?
Sí, al finalizar y cumplir con los requisitos académicos se entrega un **Certificado de Aprobación** avalado por el Ministerio de Educación.

---

10. ¿El curso incluye backend y frontend?
Exactamente. Aprenderás **backend con Laravel y NestJS** y **frontend con Angular y Vue.js**, lo que te permitirá desempeñarte como **desarrollador full stack**.

---

11. ¿Necesito tener una computadora potente?
Se recomienda un equipo con al menos **8 GB de RAM** y un procesador i5 o equivalente para ejecutar los entornos de desarrollo sin problemas.

---

12. ¿Qué tipo de soporte tendré durante el curso?
Podrás comunicarte con el instructor para resolver dudas, además de contar con **asistencia técnica y académica** durante todo el proceso formativo.