

# Micro servicios con .NET Core

**Nro. Horas: 40 horas**

## **Acerca del Curso**

Desarrollar aplicaciones web modernas usando Patrones de Micro servicios y Contenedores como Docker y Kubernetes.

En este curso, aprenderás sobre arquitectura de Micro servicios, patrones, ecosistema de contenedores, orquestación, Docker y Kubernetes.

Aprender a construir una infraestructura moderna usando contenedores docker y kubernetes. Desarrollar y desplegar tu aplicación ASP.NET Core usando Docker. Aproveche para aprender la arquitectura de micro servicios para construir su aplicación ASP.NET Core e implementar varios patrones de micro servicios en su aplicación.

## **Objetivos del Curso**

Después de este curso, los participantes adquirirán estos conocimientos:

- Entender la tecnología de contenedores
- Ecosistema de contenedores
- Docker para Desarrolladores
- Orquestación usando Kubernetes
- Usando Kubernetes con Docker
- Desarrollar aplicaciones ASP.NET Core usando Docker
- Implementar arquitectura de Micro servicios usando ASP.NET Core
- Entendiendo los conceptos básicos y avanzados de una arquitectura de Micro servicios
- Despliegue de Micro servicios usando Docker y Kubernetes

## **Para quien es este Curso**

Para todos los profesionales .NET que desean desarrollar aplicaciones web modernas, ligeras y basadas en Cloud.

## **Pre-requisitos**

Quien desea sacarle el mejor provecho a este curso debería tener conocimiento básico de C# y HTML.

## **Herramientas/IDE**

Visual Studio 2019 o mayor, Docker Desktop, Kubernetes Tools, SQL Server 2016+.

## Contenidos

### Módulo 1

- ¿Qué son Contenedores?
- ¿Por qué Contenedores?
- Contenedores vs Maquinas Virtuales
- Contenedores Linux, Windows y Hyper-V
- Orquestación de Contenedores
- ¿Qué es Docker?
- Arquitectura de Docker
- Entendiendo imágenes y contenedores
- Ejecutando Hello World en Docker
- Configurando y administrando Contenedores
- Imagen Base
- Docker File
- Trabajando con contenedores
- Optimización de Docker File
- Publicando imágenes en Docker Hub
- Registry privado

### Módulo 2

- Introducción a ASP.NET Core
- Contenedores oficiales Docker .NET
- Trabajando con Base de Datos
- Migrando aplicaciones existentes
- Introducción a volúmenes Docker
- Creando y administrando volúmenes
- Introducción a servicios Docker
- Creando y Escalando servicio Docker

### Módulo 3

- Orquestación de Contenedores
- Kubernetes
- Arquitectura Kubernetes
- Kubernetes Community
- Instalando Kubernetes
- Configurando un clúster de Kubernetes de simple nodo
- Bloques de construcción de Kubernetes
- Servicios

- Desplegando una aplicación stand-alone
- Administración de Volúmenes de Kubernetes
- Ingress
- Kubernetes con Docker
- Trabajando con Kubernetes y ASP.NET Core
- Migrando aplicaciones existentes

#### Módulo 4

- Introducción a Micro servicios
- Principios de Micro Servicios
- Monolítico vs SOA vs Micro servicios
- Arquitectura de Micro servicios
- Compañías usando Micro servicios
- Ventajas de Micro servicios
- Desventajas de Micro servicios
- Introducción a ASP.NET Core
- Configurando ambientes de Desarrollo
- Creando micro servicios usando ASP.NET Core
- Agregando soporte Docker a Micro servicios
- Probando micro servicios usando Postman
- Introducción a API Gateway
- Funcionalidades de un API Gateway
- Conociendo como trabaja el API Gateway Ocelot
- Configurando Ocelot para Agregación
- Ejecutando y probando Micro servicios

#### Módulo 5

- Patrones para Micro servicios
- Patrones de Descomposición
- Patrones de Integración
- Patrones de Base de Datos
- Patrones de Despliegue
- Patrones de Observabilidad
- Patrones de Cross-Cutting Concern

#### Módulo 6

- Entendiendo la arquitectura del Proyecto
- Entendiendo las entidades de Base de Datos

- Entendiendo los componentes del Proyecto
- Construyendo los componentes del Proyecto
- Implementando el API Gateway Ocelot
- Construyendo UIs y consumiendo APIs
- Entendiendo varias opciones de Despliegue de micro servicios
- Desplegando micro servicios usando contenedores Docker
- Desplegando micro servicios usando VMs

JOEDAYZ