

# ETHEREUM DEVELOPER COURSE

## Online Certification Course: ETH-27

**Versión en Español**

duración: 12 semanas

Hay mucho entusiasmo alrededor de blockchain, pero cómo funciona?  
¿Qué es un contrato inteligente?  
¿Por qué todos están tan entusiasmados con los DAO? ¿Y qué tiene que ver todo esto con las criptomonedas?



## Sobre el Curso...

Este curso en línea enseña a los ingenieros de software los conceptos básicos de la tecnología blockchain, así como los lenguajes y herramientas necesarios para crear aplicaciones descentralizadas en la plataforma Ethereum. Este curso está dirigido a programadores experimentados, presentando todo lo necesario para comprender la tecnología, redactar contratos inteligentes y crear aplicaciones que interactúen con ellos.

Los participantes aprenderán sobre la plataforma Ethereum, el lenguaje de programación Solidity, cómo usar web3 y el framework Truffle y, por último, cómo unir todo. Paso a paso, los participantes construyen una aplicación descentralizada en pleno funcionamiento, la despliegan y la prueban.

Este curso fue desarrollado en respuesta a las solicitudes diarias que recibimos para desarrolladores expertos en proyectos de blockchain.





```
mod.use_y = False
mod.use_z = False
action == "MIRROR_Y":
mod.use_x = False
mod.use_y = True
mod.use_z = False
action == "MIRROR_Z":
mod.use_x = False
mod.use_y = False
mod.use_z = True

#selection at the end -add back the deselected
ob.select= 1
context.scene.objects.active = modifier
selected" + str(modifier_ob)) # modifier
modifier.select = 0
key.context.selected_objects[0]
objects[one.name].select = 1
print("please select exactly two objects")
#OPERATOR CLASSES -->
#operator class
class Operator(bpy.types.Operator):
    bl_idname = "object.mirror_mirror"
    bl_label = "mirror to the selected object"
    bl_options = {'REGISTER', 'UNDO'}
    def execute(self, context):
        active_object = context.active_object
        if active_object is None:
            self.report({'ERROR'}, "No active object")
            return {'CANCELLED'}
```

## Objetivos

Al finalizar el curso, los participantes serán capaces de:  
Crear aplicaciones descentralizadas en la plataforma Ethereum, comprender la tecnología, redactar contratos inteligentes y crear aplicaciones que interactúen con ellos.

## Metodología

Este curso se imparte en modalidad E-Learning asincrónica, que permite acceder 24/7 a los contenidos de la capacitación a través de la plataforma.  
El curso incluye variados instrumentos de apoyo de alto nivel, proyectos simulados, videos, guías de estudio y exámenes.

## Requisitos para Participar

Experiencia como desarrollador, al menos 1 año.  
Conocimiento y uso de JavaScript, HTML, CSS, Git y Linux.

NO SE REQUIERE DOMINIO DE IDIOMA INGLÉS.



# Contenidos

## Módulo 1: Fundamentos de blockchain

- ¿Qué es blockchain?
- ¿Qué son las transacciones y los bloques?
- ¿Cómo operan los sistemas P2P?
- Los mecanismos de consenso más prominentes
- Diferencia entre redes privadas, de consorcio y públicas
- ¿Cómo se ve la estructura de datos?
- ¿Cómo funcionan los contratos inteligentes?
- ¿Qué es la Máquina Virtual Ethereum?
- ¿Qué es un Hash?
- ¿Cómo funcionan las claves públicas / privadas?
- ¿Qué es un árbol de Merkle?
- ¿Cuál es el problema del doble gasto?

## Módulo 2: Fundamentos de Ethereum

- El ecosistema Ethereum, DApps y DAOs
- ¿Qué es niebla y cómo funciona?
- ¿Qué es Ether, una cuenta, un Faucet?
- Compilar, implementar y crear instancias de contratos
- Configurando, corriendo y trabajando con el cliente go-ethereum
- Gestión de cuentas y minería
- Comprender las diferentes etapas de una implementación de contrato
- ¿Cómo interactuar con un contrato una vez desplegado?
- ¿Cómo se ve el proceso de configuración de una red privada?

## Módulo 3: Solidity

- Clases contractuales, funciones y condicionales
- Herencia y contratos abstractos
- Bibliotecas
- Tipos y optimización
- Variables globales
- Depuración

# Contenidos

## Módulo 4: Web3 & Truffle

- Construir una interfaz para interactuar con un contrato inteligente
- Configuración de interfaces basadas en eventos
- Adaptación del comportamiento predeterminado de Truffle
- Escribir pruebas funcionales para contratos inteligentes
- ¿Cómo probar manualmente la funcionalidad del contrato?
- Crear una instancia de web3 y comunicarse con un contrato desde una página HTML
- Utilice la firma del lado del cliente y los nodos remotos para DApps livianos
- ¿Cómo funciona web3 y trufa con Angular.js?

