

Lección 3

Taller de Smart contracts

M2- Unidad 1

Tiempo de ejecución: 4 horas



PLANTEAMIENTO DE LA SESIÓN	Materiales
<p>En el taller, los participantes se sumergirán en un análisis profundo de las principales plataformas que sustentan el desarrollo y la ejecución de Smart Contracts. Se explorarán las diferencias fundamentales entre estas plataformas, centrándose en aspectos como los lenguajes de programación utilizados, las funcionalidades disponibles y los protocolos de consenso empleados. Este análisis detallado permitirá a los participantes comprender cómo estas variaciones pueden afectar el diseño y la implementación de los contratos inteligentes, destacando la importancia crítica de elegir la plataforma adecuada para cada aplicación específica. En última instancia, se resaltarán el papel fundamental de estas plataformas en el fomento del crecimiento y la innovación en el ecosistema de los contratos inteligentes en la blockchain.</p>	<p>PC con conexión a internet.</p>



Desarrollo de la sesión

Durante la sesión, los participantes serán guiados a través de un proceso estructurado y completo para explorar los Smart Contracts. El programa comienza con una sólida introducción que establece los fundamentos conceptuales de los contratos inteligentes, proporcionando a los asistentes una comprensión clara de su importancia y funcionamiento en el ecosistema blockchain. Este primer paso sienta las bases para la inmersión en la exploración de varias plataformas blockchain líderes.

A lo largo de la sesión, se analizarán detenidamente las diferencias y ventajas de estas plataformas, brindando a los participantes una visión detallada de sus características distintivas, protocolos de consenso y capacidades para el desarrollo de contratos inteligentes. Este enfoque comparativo permitirá a los asistentes comprender mejor cómo seleccionar la plataforma más adecuada para sus necesidades específicas de desarrollo.

- Continuando con el enfoque práctico, los participantes serán guiados a través de la configuración de su entorno de desarrollo para la creación y despliegue práctico de contratos inteligentes. Se proporcionarán instrucciones detalladas y ejemplos prácticos para ayudar a los asistentes a familiarizarse con el proceso de desarrollo y ejecución de Smart Contracts en el entorno de su elección.

+ info

Inicio

Taller de Smart Contracts

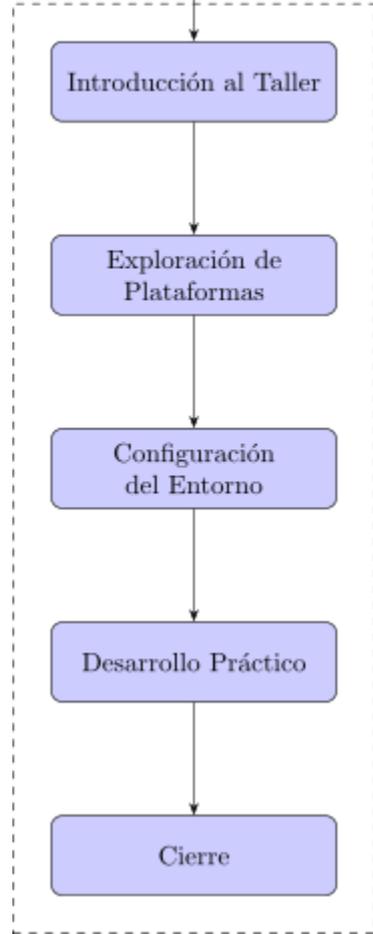
Introducción al Taller

Exploración de
Plataformas

Configuración
del Entorno

Desarrollo Práctico

Cierre



Parte 1: Ejercicios (2 horas)

1. Introducción a Smart Contracts:

- En esta primera parte del taller, se introducirá a los participantes en el concepto de Smart Contracts y su relevancia en el ámbito de la blockchain. Se destacará cómo los Smart Contracts son piezas de código autónomas que ejecutan automáticamente acuerdos digitales sin la necesidad de intermediarios.
- Ejercicio: Los participantes realizarán una breve investigación sobre un caso de uso real de Smart Contracts en cualquier industria. Esto les permitirá comprender mejor cómo se aplican estos contratos inteligentes en situaciones del mundo real y cómo pueden proporcionar soluciones eficientes y seguras para diversas necesidades comerciales.

2. Exploración de Plataformas:

- En esta segunda parte, se presentarán diferentes plataformas populares para el desarrollo de Smart Contracts, como Ethereum, Binance Smart Chain y Tezos. Se discutirán las diferencias clave entre estas plataformas en términos de lenguajes de programación, funcionalidades y protocolos de consenso.
- Ejercicio: Los participantes compararán y contrastarán las características de al menos dos plataformas diferentes. Este ejercicio les ayudará a comprender mejor las fortalezas y debilidades de cada plataforma y les permitirá tomar decisiones informadas al elegir una plataforma para sus proyectos de Smart Contracts.



```
// SPDX-License-Identifier: MIT  
pragma solidity ^0.8.0;  
contract MiContrato  
{ string public mensaje = "Hola  
Mundo!";  
}
```

**Compilación de Contratos
Inteligentes en Solidity**

Parte 2: Desarrollo Práctico (2 horas)

1. Configuración del Entorno:

- En esta parte práctica, los participantes aprenderán a configurar un entorno de desarrollo de Ethereum utilizando herramientas como Remix, Metamask y Ganache. Se proporcionarán instrucciones detalladas paso a paso para que los participantes puedan seguir fácilmente.
- Ejercicio: Los participantes seguirán las instrucciones para configurar el entorno en una máquina local. Esto les permitirá tener un entorno de desarrollo funcional listo para escribir y desplegar contratos inteligentes en Ethereum.

+ Desarrollo

```
// SPDX-License-Identifier: MIT pragma solidity ^0.8.0;
contract SimpleToken { mapping(address => uint256) public balances;
constructor() { balances[msg.sender] = 1000; }
function transfer(address recipient, uint256 amount) public {
require(balances[msg.sender] >= amount, "Insufficient balance");
balances[msg.sender] -= amount;
balances[recipient] += amount;
}
}
```





```
// SPDX-License-Identifier: MIT  
pragma solidity ^0.8.0;  
contract MiContrato  
{ string public mensaje = "Hola  
Mundo!";
```

```
} Compilación de Contratos  
Inteligentes en Solidity
```

Para compilar y probar estos contratos, puedes seguir estos pasos utilizando Remix IDE:

1. Abre Remix IDE en el navegador web.
2. En la pestaña "File Explorer", haz clic en el botón "Create File" para crear un nuevo archivo.
3. Copia y pega el código del contrato correspondiente en el nuevo archivo.
4. Cambia a la pestaña "Solidity Compiler" en Remix IDE.
5. Selecciona el contrato que deseas compilar en el menú desplegable.
6. Haz clic en el botón "Compile" para compilar el contrato.
7. Verifica que no haya errores de compilación en la pestaña "Compile Results".
8. Para probar la funcionalidad del contrato, selecciona la pestaña "Deploy & Run Transactions".
9. Selecciona "Injected Web3" como environment y asegúrate de que esté conectado a tu cuenta de MetaMask.
10. Haz clic en el botón "Deploy" para desplegar el contrato en Remix IDE.

Creación y Despliegue de un Contrato Inteligente en Solidity

Paso 1: Introducción a Solidity y Ethereum

Los participantes serán introducidos a Solidity, el lenguaje de programación utilizado en Ethereum para desarrollar contratos inteligentes. Se explicarán los conceptos básicos de Solidity y su papel en la creación de contratos inteligentes en la red Ethereum.

Paso 2: Diseño del Contrato Inteligente

Se guiará a los participantes a través del diseño de un contrato inteligente básico que implemente una función de transferencia de tokens ficticios entre direcciones. Se definirán las variables necesarias, así como la estructura general del contrato inteligente.

+ INFO



```
// SPDX-License-Identifier: MIT  
pragma solidity ^0.8.0;  
contract MiContrato  
{ string public mensaje = "Hola  
Mundo!";  
}
```

**Compilación de Contratos
Inteligentes en Solidity**



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Token {
    mapping(address => uint256) public balances;

    constructor() {
        balances[msg.sender] = 1000;
    }

    function transfer(address recipient, uint256 amount) public {
        require(balances[msg.sender] >= amount, "Insufficient balance");
        balances[msg.sender] -= amount;
        balances[recipient] += amount;
    }
}
```



Paso 3: Escritura del Contrato Inteligente en Solidity

Los participantes escribirán el código del contrato inteligente utilizando Solidity. Se explicará la sintaxis del lenguaje y se proporcionarán ejemplos para ayudar a los participantes a completar el código del contrato inteligente.

Paso 4: Despliegue del Contrato en una Red de Prueba Local

Los participantes utilizarán herramientas como Ganache o Truffle para desplegar su contrato inteligente en una red de prueba local. Se proporcionarán instrucciones detalladas sobre cómo configurar el entorno de desarrollo y realizar el despliegue en la red de prueba.

Paso 5: Prueba y Verificación del Contrato Desplegado

Una vez desplegado, los participantes probarán y verificarán el funcionamiento del contrato inteligente utilizando herramientas como Remix o Truffle. Se realizarán pruebas de transferencia de tokens entre direcciones simuladas para asegurarse de que el contrato se comporte según lo esperado.

Paso 6: Discusión y Recapitulación

Se llevará a cabo una discusión y recapitulación final, donde los participantes tendrán la oportunidad de compartir sus experiencias y hacer preguntas sobre el proceso de creación y despliegue de contratos inteligentes en Ethereum.

4. Creación y Despliegue de un Contrato Inteligente:

- En esta fase, los participantes aprenderán los fundamentos de la escritura y despliegue de un contrato inteligente simple en Solidity, (como se muestra en Creación y Despliegue de un Contrato Inteligente en Solidity), el lenguaje de programación utilizado en Ethereum. Se guiará a los participantes a través de la creación de un contrato básico que implemente una función de transferencia de tokens ficticios entre direcciones.
-
- Ejercicio: Los participantes escribirán y desplegarán su propio contrato inteligente básico en una red de prueba local. Este ejercicio les dará experiencia práctica en el desarrollo de contratos inteligentes en Ethereum.

+ INFO



Recursos Adicionales

- [Documentación Oficial de Solidity](#)
- [Guía de Remix](#)
- [Ganache](#)
- [Truffle](#)
- [Tutorial de Deploy de Contratos en Ethereum](#)



```
// SPDX-License-Identifier: MIT  
pragma solidity ^0.8.0;  
contract MiContrato  
{ string public mensaje = "Hola  
Mundo!";  
}
```

**Compilación de Contratos
Inteligentes en Solidity**