

Soluzione E-commerce Decentralizzata per Corsi Inclusivi



Progetto sviluppato da Antonino Paternò

Inclusione attraverso la formazione

- IncluDO è una startup che mira a fornire formazione professionale inclusiva
- Punta ad offrire opportunità di apprendimento per migranti e persone svantaggiate.
- Si pone come obiettivo la creazione di un sistema di e-commerce decentralizzato per l'acquisto e la gestione dei corsi.

Obiettivi del Progetto

- Sviluppare uno smart contract in Solidity per gestire:
- ✓ Corsi formativi e relative vendite
- ✓ Registrazioni sicure delle transazioni
- ✓ Tracciamento delle vendite
- ✓ Fornire una soluzione trasparente e immutabile su blockchain

Definizione Strutture Corsi ed Iscrizioni

- La prima struttura logica definisce i dettagli di un corso, tra cui titolo, ID e prezzo in wei

```
struct Course {  
    uint256 id;  
    string title;  
    uint256 feeInWei;  
}
```

- L'ulteriore struttura definita direttamente all'interno della libreria custom SalesLibrary permette di associare dettagli di una vendita, come il corso acquistato, il momento dell'acquisto e l'indirizzo dell'acquirente.

```
struct Enrollment {  
    uint256 courseId;  
    address student;  
    uint256 timestamp;  
}
```

Funzionalità Chiave dello Smart contract CourseCommerceManager

1

Aggiunta di nuovi corsi da parte del proprietario fissandone il costo

2

Possibilità per gli utenti di acquistare corsi pagando in Ether

3

Tracciamento automatico delle vendite e gestione delle transazioni

4

Eventi per monitorare in modo trasparente le iscrizioni e le vendite

Ruolo della Libreria SalesLibrary

➤ Funzioni di supporto per:

1

Visualizzare i corsi acquistati da un determinato utente

2

Calcolare l'ammontare delle vendite in un periodo definito

3

Riduzione del codice ripetuto e ottimizzazione delle operazioni complesse

Processo trasparente di vendita

➤ *Gestione delle Transazioni*



Step 1

*Tutte le vendite
vengono registrate
sulla blockchain*

Step 2

*Utilizzo degli eventi
per notificare
l'acquisto di un corso*

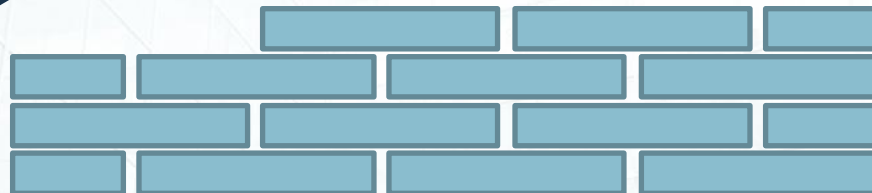
Step 3

***Proprietario può prelevare
i fondi raccolti in Ether in
qualsiasi momento***

Sicurezza e Controlli

➤ L'uso di **modificatori** e strutture di controllo **Require** per verifica condizione di validazione ha consentito, oltre a garantire eleganza e pulizia nel codice, di implementare:

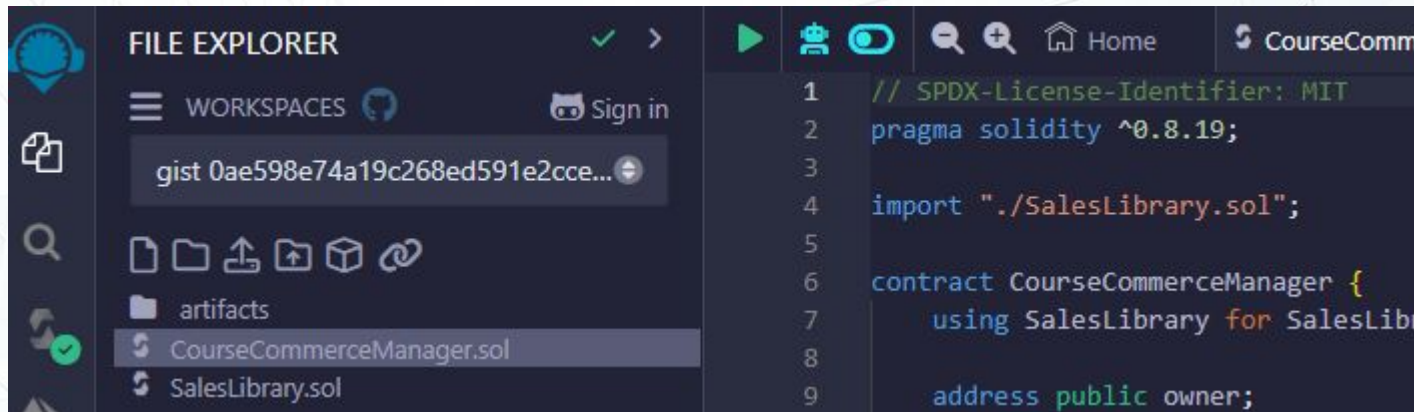
- aggiunta corsi solo all'owner del contratto
- impossibilità per l'owner di auto acquistare corsi da lui stesso pubblicati
- Verificare il corretto pagamento prima di completare un acquisto
- Tracciamento delle iscrizioni per evitare doppie registrazioni allo stesso corso da parte di un EoA



```
modifier onlyOwner() {  
    require(  
        msg.sender == owner,  
        "Solo il proprietario puo' eseguire questa funzione."  
    );  
}
```

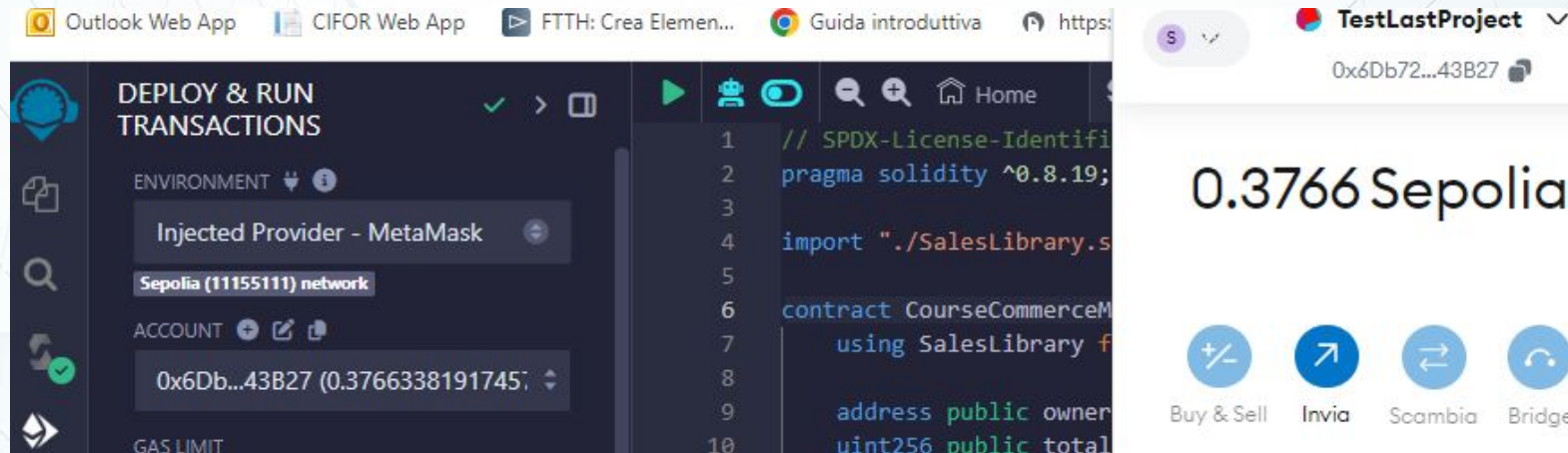

Strumenti per Test del contratto

- *Remix fornisce un'interfaccia user-friendly e una serie di strumenti integrati per facilitare lo sviluppo su Ethereum; grazie a questo tool è stato possibile testare il contratto direttamente sulla blockchain locale e garantire che fosse aderente ai requisiti prefissati.*



Deploy su rete di Test Sepolia

- *Dopo i Test su local Blockchain e verifica delle funzionalità propagando le Tx firmate tramite Metamask , si è passati al deploy su rete di test Sepolia del contratto e relativa libreria in esso importato.*



Address dei contratti su Sepolia:

- **CourseCommerceManager:** [0x3D7111F7282C8750cf6866a7482d1A306f4B80CC](https://sepolia.etherscan.io/address/0x3D7111F7282C8750cf6866a7482d1A306f4B80CC)
- **SalesLibrary:** [0xd52865c6349479a89d0ba1cf7af07f01c94b0cec](https://sepolia.etherscan.io/address/0xd52865c6349479a89d0ba1cf7af07f01c94b0cec)

Deploy e versioning GitHub del codice

- *La fase di sviluppo del progetto è stata facilitata dall'uso del IDE Visual Studio Code pushando le modifiche sul repository GITHUB prima sul branch “development” per poi eseguire il merge sul branch principale “main”*

<https://github.com/antopat1/ProgettoSmartContractconSolidityDiAntoninoPaterno>

- *E' possibile deployare e testare il contratto direttamente su IDE Remix al seguente URL offerto dal servizio GIST di GitHub*

<https://remix.ethereum.org/#version=soljson-v0.8.26+commit.8a97fa7a.js&optimize=false&runs=200&gist=0ae598e74a19c268ed591e2cce5348fd&lang=en&evmVersion=null>