Progettazione e rilascio smart contract di raccolta fondi

Utilizzo della tecnologia blockchain per automatizzare e rendere trasparente una campagna di raccolta fondi sul portale di una società no-profit operante nel settore information

This project was developed by Antonino Paternò

Obiettivo:

- Creare un portale di raccolta fondi trasparente e sicuro usando uno smart Contract ovvero un contratto immutabile distribuito su una blockchain pubblica di tipo EVM
- Totale trasparenza delle donazioni
- Sicurezza dei fondi raccolti
- Automazione del processo di raccolta fondi e gestione

Scenario Reale:

1. Il manager crea una campagna di raccolta fondi sulla piattaforma web della società ; il backend di tale piattaforma implementa i meccanismi e sfrutta le opportune librerie di interazione web3 attraverso cui l EoA associato al manager distribuisce sulla blockchain lo smartcontract che astrae e governa la campagna avendone inoltre i privilegi sulla fase di chiusura e ritiro fondi.
2. Gli utenti visitano il sito, individuano la campagna e donano Ether attraverso l'interazione del proprio Wallet con la piattaforma stessa.
3. Le donazioni vengono registrate automaticamente sulla blockchain.
4. Quando l'obiettivo è raggiunto, il manager chiude la donazione per prelevare successivamente i fondi raccolti.

Elementi chiavi dello SmartContract:

- Astrazione di partecipante alla campagna attraverso la struttura **Participant** che, grazie al campo *role* ne stabilisce il ruolo (manager o donante) e grazie agli altri campi in comune ne differenzia l'address del relativo EoA, l 'importo donato ed altre informazioni utili alla logica del contratto

```
struct Participant {
    address addr; // Indirizzo del partecipante
    string name; // Nome del partecipante
    string role; // Ruolo del partecipante (manager o donatore)
    uint256 amountDonated; // Importo donato
    bool hasDonated; // Flag per indicare se ha donato
    bool exists; // Campo aggiunto per indicare se il partecipante esiste
}
```

- Costruttore che , in fase di creazione e deploy del contratto fissa l'obiettivo della raccolta fondi espresso in Ether , il nome del manager che sarà poi l'unico partecipante ad aver i privilegi di chiudere e prelevare i fondi raccolti.

```
constructor(uint256 _goalInEther, string memory _managerName) {
    require(bytes(_managerName).length > 0, "Il nome del manager deve essere specificato.");
    manager = msg.sender; // L'account che deploya il contratto è il manager
    goal = etherToWei(_goalInEther); // Convertiamo l'obiettivo da Ether a Wei
    participants[manager] = Participant(manager, _managerName, "manager", 0, false, true); // Crea il partecipante manager
}
```

Implementazione sicurezza SmartContract:

- Utilizzo di ReentrancyGuard di OpenZeppelin , libreria ampiamente testata e mantenuta per fornire soluzioni di sicurezza standardizzate. Garantisce protezione contro attacchi di tipo «richiamo ricorsivo» non autorizzato, proteggendo funzioni critiche come withdrawFunds.
- L’utilizzo di questa libreria ampiamente testata e mantenuta per fornire soluzioni di sicurezza standardizzate garantisce affidabilità.
- L’applicazione di un semplice modificatore ***nonReentrant*** riduce il rischio di errori umani durante lo sviluppo.
- L’allineamento con tali standard di sicurezza migliora la robustezza del contratto.

Funzioni disponibili 1/2:

Funzione	Descrizione	Utilizzo	Restrizioni
Donate	Permette agli utenti di donare Ether al contratto.	Gli utenti inviano Ether insieme al proprio nome per fare una donazione.	Ogni utente può donare solo una volta e le donazioni sono bloccate se l'obiettivo è raggiunto o la raccolta è chiusa.
withdrawFunds	Permette al manager di prelevare tutti gli Ether raccolti nel contratto.	Solo il manager può chiamare questa funzione.	I fondi possono essere prelevati solo se la raccolta fondi è chiusa.
closeFundraiser	Chiude la raccolta fondi.	Solo il manager può chiamare questa funzione per chiudere la raccolta fondi.	Dopo la chiusura, non è possibile fare ulteriori donazioni.
checkGoalReached	Verifica se l'obiettivo della raccolta fondi è stato raggiunto o superato.	Può essere chiamata da chiunque per controllare lo stato dell'obiettivo.	Nessuna
getGoalInEther	Restituisce l'obiettivo della raccolta fondi in Ether.	Può essere chiamata da chiunque per conoscere l'obiettivo in unità di Ether.	Nessuna

Funzioni disponibili 2/2:

Funzione	Descrizione	Utilizzo	Restrizioni
getTotalBalanceInEther	Restituisce il saldo totale raccolto in Ether.	Può essere chiamata da chiunque per conoscere il saldo totale in unità di Ether.	Nessuna
getTotalDonors	Restituisce il numero totale di donatori.	Può essere chiamata da chiunque per conoscere il numero totale di donatori.	Nessuna
getFundraiserStatus	Restituisce lo stato della raccolta fondi (aperta o chiusa).	Può essere chiamata da chiunque per conoscere se la raccolta fondi è aperta o chiusa.	Nessuna
getParticipant	Restituisce le informazioni di un partecipante dato il suo indirizzo.	Può essere chiamata da chiunque per ottenere i dettagli di un partecipante.	Se nessun partecipante ha ancora donato, restituisce un messaggio che indica l'assenza di donatori.
getManagerName	Restituisce il nome del manager che ha creato il contratto.	Può essere chiamata da chiunque per conoscere il nome del manager.	Nessuna
getAllDonors	Restituisce un elenco di tutti i donatori con i rispettivi indirizzi, nomi e importi donati.	Può essere chiamata da chiunque per ottenere una lista completa dei donatori.	Se nessun donatore ha ancora partecipato, restituisce un messaggio che indica l'assenza di donatori.

Scelte Tecnologiche:

- **Solidity** linguaggio di programmazione orientato agli smart contract, progettato per essere semplice e leggibile, consente la scrittura di contratti che sono eseguibili direttamente sulla blockchain di Ethereum e tutte le altre chain EVM L1/L2 compatibili.
- **Ethereum** piattaforma decentralizzata che permette di eseguire smart contract e applicazioni decentralizzate (dApps) senza tempi di inattività, frodi, controlli o interferenze da parte di terzi.
- **Remix IDE** 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 del Progetto 1/2:

- Il Codice del progetto è stato versionato su GitHub al Repository:

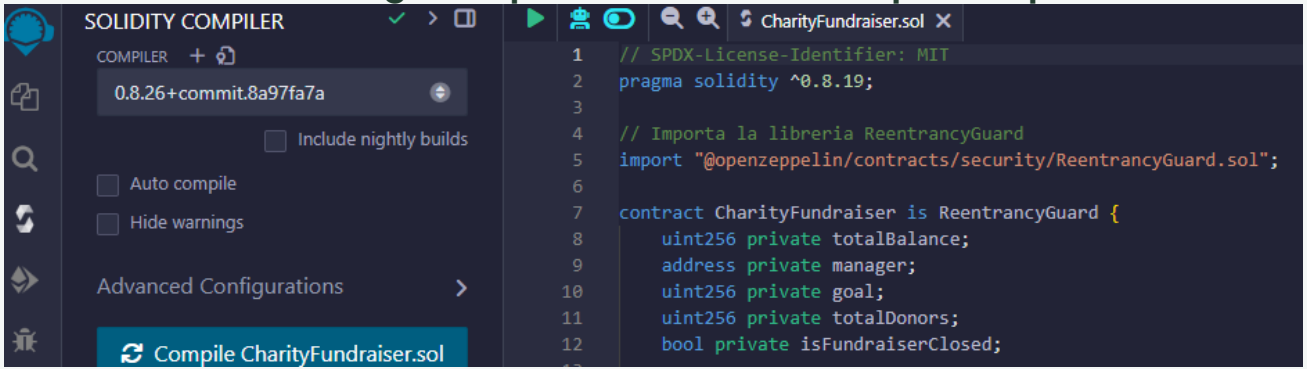
  https://github.com/antopat1/solidityFundraiser

- E' possibile deployare e testare direttamente il contratto su IDE Remix al seguente URL:

https://remix.ethereum.org/#version=soljson-v0.8.19+commit.7dd0d32e.js&optimize=false&runs=200&gist=f1bc0670c2f908726cdd012bdea6939c

Deploy del Progetto 2/2:

- In questo ambiente sarà già disponibile il contratto pronto per essere compilato:



Una volta compilato si potrà scegliere uno degli EoA di test (con bilancio fittizio 100Eth) per la fase di deploy indicando il target in Eth ed il nome del manager che ha aperto la donazione e che ha privilegi di chiusura e ritiro fondi.

Accesso alle funzioni del contratto:

- I diversi EoA in relazione alla visibilità di variabili e funzioni disponibili del contratto, potranno propagare delle transazioni sulla blockchain locale di Test per leggere o modificare lo stato come ad esempio durante la fase di donazione.