**ExclusivePass Token**

**STF**

|  |  |  |
| --- | --- | --- |
| Versione Documento |  | Data Documento |
| 1.0 |  | 13/04/2024 |
|  |  |  |

SOMMARIO

[1 STORIA DEL DOCUMENTO 3](#_Toc163904082)

[2 INTRODUZIONE 4](#_Toc163904083)

[2.1 Definizioni, abbreviazioni, acronimi 4](#_Toc163904084)

[2.1.1 Definizioni 4](#_Toc163904085)

[2.1.2 Acronimi 4](#_Toc163904086)

[2.2 Documenti di riferimento 4](#_Toc163904087)

[3 INFORMAZIONI GENERALI 5](#_Toc163904088)

[4 Impostazione progetto 5](#_Toc163904089)

[4.1 Tecnologie utilizzate 5](#_Toc163904090)

[5 Dettagli Contratto 5](#_Toc163904091)

[6 PUNTI DI ATTENZIONE 7](#_Toc163904092)

[7 ALLEGATI 8](#_Toc163904093)

# STORIA DEL DOCUMENTO

Le informazioni contenute in questo documento aggiornano e sostituiscono quelle contenute nelle versioni precedenti. Ogni modifica del documento sarà controllata nella apposita procedura.

|  |  |  |  |
| --- | --- | --- | --- |
| **Versione** | **Data** | **Descrizione della Revisione** | **Modificato da** |
| 1.0 | 13/04/2024 | Prima versione | Carlo Blatti |

# INTRODUZIONE

## Definizioni, abbreviazioni, acronimi

### Definizioni

### Acronimi

## Documenti di riferimento

|  |
| --- |
| **Documento** |
| Proposta\_token\_Blatti.docx |

# INFORMAZIONI GENERALI

Il progetto è stato realizzato utilizzando MetaMask per interagire con la testnet Sepolia sulla blockchain Ethereum, e il linguaggio Solidity con librerie di supporto allo sviluppo, come openzeppelin.

# Impostazione progetto

Il progetto dell’applicativo è stato sviluppato utilizzando Remix come ambiente di sviluppo e deploy del contratto su testnet.

## Tecnologie utilizzate

Per lo sviluppo del progetto si è fatto uso delle seguenti tecnologie:

* Remix
* Solidity à v. 0.8.0
* Openzeppelin à v. 19.8.0.0
* Token Erc-20

# Dettagli Contratto

L’ implementazione include le seguenti funzionalità:

* Creazione del token: L'organizzatore può creare un nuovo token ERC-20 specificando il nome, il simbolo, il prezzo e il numero massimo di partecipanti consentiti per l'evento.
* Aggiornamento del prezzo: L’organizzatore può aggiornare il prezzo del biglietto in qualunque momento.
* Acquisto dei token: Gli interessati all'evento possono acquistare i token tramite la funzione purchaseTokens. Viene controllato che il numero massimo di partecipanti non venga superato e che l'importo inviato sia corretto.
* Utilizzo dei token: I partecipanti possono registrare la propria presenza all'evento tramite la funzione recordAttendance. Viene controllato se il partecipante possiede almeno un token e se non ha già registrato la propria presenza.
* Bruciatura dei token: Quando un partecipante registra la propria presenza, il token corrispondente viene bruciato tramite la funzione \_burn, rendendolo non spendibile.
* Ritiro dei fondi: L'organizzatore può ritirare i fondi raccolti dalla vendita dei token tramite la funzione withdrawFunds.

# PUNTI DI ATTENZIONE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | Descrizione | Owner | Decisioni | Stato |
|  |  |  |  |  |
|  |  |  |  |  |

# ALLEGATI

|  |  |  |
| --- | --- | --- |
| **ID** | Allegato | Descrizione |
|  |  |  |
|  |  |  |
|  |  |  |