

## **Documento di Tracce per Hackathon su Avalanche**

### **1. Dashboard per Validatori**

Serve un backend che interroga la P Chain tramite gli endpoint RPC ufficiali. Il backend raccoglie uptime, data di inizio della validazione, storico delle commissioni e quantità totale di deleghe. Espone poi queste informazioni tramite API semplici che la webapp può richiamare.

La webapp permette di cercare un validatore, mostra i dati ricevuti dal backend e consente di inviare una richiesta di staking. L'operazione di staking può essere preparata dal backend che restituisce alla webapp la transazione da firmare con il wallet dell'utente.

Non è richiesto alcuno smart contract, poiché la P Chain gestisce direttamente i validatori tramite le proprie primitive.

### **2. Forum on chain**

Serve uno smart contract sulla C Chain che gestisce la registrazione degli utenti con nome e immagine, la creazione dei post e dei commenti. Le funzioni devono essere molto semplici e limitate alle operazioni essenziali.

Serve un backend che ascolta gli eventi generati dallo smart contract e indicizza i profili, i post e i commenti per permettere ricerche rapide e una navigazione fluida. Il backend espone API per la webapp.

Serve una webapp che permette la registrazione dell'utente e l'upload dell'immagine, mostra i contenuti indicizzati dal backend e consente di creare post e commenti tramite interazione diretta con lo smart contract.

In alternativa il forum può essere costruito sulla P Chain senza smart contract, usando solo un backend che scrive e legge metadati legati ai validatori per creare uno spazio di discussione semplificato.

### **3. Dust Collector**

Serve una webapp che permette di collegare il wallet tramite librerie come web3 punto js o compatibili.

La webapp legge i saldi del wallet e identifica i piccoli importi. Per ogni piccolo importo costruisce una transazione di swap oppure una transazione verso un indirizzo di burn.

Non è necessario un backend e non servono smart contract custom, perché la webapp può interagire direttamente con i DEX esistenti oppure generare semplici transazioni di invio.

### **4. Address Scoring**

Serve un backend che si collega alla C Chain, scarica lo storico delle operazioni di un gruppo di indirizzi e applica una logica basilare di analisi. Il backend assegna un punteggio insieme a una classificazione come bot, utente reale, istituzione oppure contratto.

Serve una webapp che consuma i risultati del backend e mostra elenco degli indirizzi con il punteggio associato.

Non servono smart contract, poiché l'analisi riguarda solo i dati già presenti sulla blockchain.

### **5. Fantacalcio on chain**

Serve uno smart contract basato su ERC-721 per creare gli NFT che rappresentano i giocatori. Il codice può essere preso da repository open source già pronti.

Serve uno smart contract che gestisce le aste: creazione delle aste, offerte e assegnazione finale dell'NFT.

Serve uno smart contract che gestisce la formazione e i punteggi. Deve permettere di registrare la formazione e calcolare i punti alla fine della partita.

Serve un backend opzionale che indica le aste, i giocatori e le formazioni, così la webapp può recuperare velocemente i dati.

Serve una webapp che permette di partecipare alle aste, visualizzare i giocatori, impostare la formazione e leggere la classifica generata dagli smart contract.

## **6. Gaming on chain**

Serve una rete EVM compatibile come Avalanche C Chain o Subnet EVM.

Servono smart contract che gestiscono gli oggetti del gioco, gli account dei giocatori e qualsiasi token necessario. Ogni oggetto o risorsa del gioco può essere rappresentato tramite token o strutture equivalenti.

Serve un backend che recupera i dati del gioco dagli smart contract, gestisce la logica esterna non critica e fornisce informazioni aggiornate alla webapp o al gioco vero e proprio.

Serve una webapp o un gioco web che interagisce direttamente con gli smart contract per lettura e azioni essenziali. Deve permettere di raccogliere oggetti, visualizzare l'inventario e collegare le azioni nel gioco alle operazioni on chain.