



DIPARTIMENTO DI FISICA

METODI COMPUTAZIONALI DELLA FISICA

---

# Sviluppo di un Token ERC20 e di un Exchange decentralizzato

---

*Autore: Marta Ferrero*

*Autore: Cristian Angelo Crespi*

*Autore: Carlo Amedeo Melloni*

10 Luglio 2024

# Indice

<b>1</b>	<b>Introduzione agli exchange decentralizzati</b>	<b>2</b>
1.1	Vantaggi dei DEX . . . . .	2
1.2	Svantaggi dei DEX . . . . .	3
1.3	Possibili attacchi ai DEX . . . . .	3
<b>2</b>	<b>Introduzione ai market making automatico</b>	<b>4</b>
2.1	Tipologie di AMM . . . . .	5
2.1.1	AMM a prodotto costante . . . . .	5
2.1.2	AMM a somma costante . . . . .	5
<b>3</b>	<b>Implementazione</b>	<b>5</b>
3.1	Smart Contract . . . . .	6
3.1.1	Token . . . . .	6
3.1.2	Marketplace . . . . .	7
3.1.3	Challenge . . . . .	8
3.2	Bot e script . . . . .	9
3.3	Integrazione Telegram . . . . .	9
<b>4</b>	<b>Analisi della gara</b>	<b>9</b>
4.1	Andamento dei prezzi . . . . .	9
4.2	Andamento dei wallet dei partecipanti . . . . .	10
4.3	Guadagni e perdite per ciascun utente . . . . .	12
4.4	Swap effettuati . . . . .	12
4.5	Fee pagate . . . . .	13
4.6	Challenge . . . . .	14
<b>5</b>	<b>Strategia gruppo QTC</b>	<b>15</b>
<b>6</b>	<b>Comportamenti anomali</b>	<b>16</b>
6.1	Impossibilità di utilizzare la rete SepoliaETH . . . . .	16
6.2	Funzione day_mint . . . . .	16
6.3	Challenge . . . . .	16
6.4	Memoria del server . . . . .	16
6.5	Malfunzionamento iniziale dei bot . . . . .	17
<b>7</b>	<b>Conclusione</b>	<b>17</b>
<b>8</b>	<b>Appendice</b>	<b>18</b>
8.1	Andamento dei token nei wallet dei partecipanti . . . . .	18
8.2	Andamento delle pool . . . . .	19
8.3	Analisi dei bot . . . . .	20

## Sommario

In questa relazione, descriviamo la realizzazione di 4 token ERC-20, di un exchange decentralizzato con market making automatico basato su una funzione di prezzo a prodotto costante. Gli smart contract sviluppati in Solidity sono stati lanciati sulla blockchain di ganache. Sono stati sviluppati in Python dei sistemi di monitoraggio e dei bot per simulare rumore, è stata anche costruita un' interfaccia Telegram per facilitare l'utilizzo del sistema. Successivamente si è svolta una gara di trading della durata di 5 giorni tra 3 gruppi: LDR,QTC e BOR. Di seguito sarà discussa l'implementazione e analizzati i risultati della gara.

## 1 Introduzione agli exchange decentralizzati

Gli exchange decentralizzati (DEX) rappresentano una delle più grandi innovazioni nel mondo delle criptovalute e delle *blockchain*. La caratteristica distintiva dei DEX è la loro natura decentralizzata. Essi permettono agli utenti di effettuare scambi direttamente tra loro (*peer-to-peer*), utilizzando la tecnologia blockchain per garantire trasparenza, sicurezza e immutabilità. Questo approccio elimina la necessità di un'entità centrale che supervisiona e facilita le transazioni, riducendo così il rischio di frodi e attacchi informatici.

Uno dei principi fondamentali della criptovaluta è la disintermediazione della finanza, poiché aumenta la libertà economica degli individui ovunque nel mondo, indipendentemente da chi siano. I DEX hanno rapidamente guadagnato popolarità su *Ethereum*, la blockchain più popolare, per poi espandersi su tutte le blockchain dotate di funzionalità di *smart contracts* generici. Essi sono un elemento cruciale nell'ecosistema di una blockchain, poiché rappresentano la prima applicazione decentralizzata (dApp) con cui gli utenti interagiscono quando approdano su una nuova catena. I DEX permettono di effettuare scambi di criptoasset e offrono anche opportunità per guadagnare criptoasset senza necessità di fare *trading* attivamente.

### 1.1 Vantaggi dei DEX

La decentralizzazione offre numerosi vantaggi rispetto alle controparti centralizzate. Innanzitutto, i nuovi criptoasset appaiono quasi sempre prima sui DEX rispetto agli exchange centralizzati (CEX). Questo perché i CEX devono aggiungere manualmente i criptoasset al loro sistema, un processo che richiede vari livelli di conformità, test e autorizzazione da parte della direzione. Con un DEX, qualsiasi nuovo token può essere automaticamente aggiunto all'exchange senza bisogno di autorizzazione.

I DEX non richiedono misure di conoscenza del cliente KYC (*Know Your Customer*) o di anti-riciclaggio di denaro AML (*Anti Money Laundering*). Poiché lo scambio non esiste in una singola giurisdizione, non è soggetto a un quadro normativo specifico. Questo comporta diverse implicazioni. Innanzitutto, i DEX preservano la privacy: è possibile utilizzarli in modo anonimo senza dover divulgare informazioni personali, inclusi i dati finanziari come il saldo del proprio conto. Inoltre, si può iniziare a fare trading immediatamente su un DEX, senza dover superare lunghi processi di verifica.

I DEX sono particolarmente utili nei paesi in via di sviluppo, dove la mancanza di un'infrastruttura finanziaria robusta ha impedito alle aziende finanziarie tradizionali di servire molti clienti. La Banca Mondiale stima che un miliardo di persone non possiedono una prova ufficiale di identità. Queste persone non potrebbero superare nemmeno le misure KYC e AML più lasche, ma con accesso a internet possono commerciare su un DEX.

Infine, non è necessario commerciare per guadagnare su un DEX. Esistono diversi modi per generare profitti, come fornire liquidità e guadagnare una parte delle commissioni. Inoltre, molti DEX offrono premi bonus, tipicamente pagati nel token di *governance* del DEX, incentivando ulteriormente la partecipazione degli utenti. I token di governance permettono di democratizzare le scelte governative del protocollo blockchain.

## 1.2 Svantaggi dei DEX

La principale difficoltà nell'utilizzo dei DEX riguarda l'interfaccia utente. Trasferire criptoasset in un portafoglio e navigare le interfacce dei DEX può risultare complicato, richiedendo un notevole sforzo di apprendimento iniziale prima di poter effettuare una singola operazione di scambio. Questa difficoltà è ulteriormente aggravata dal fatto che le transazioni spesso falliscono per vari motivi, e il supporto disponibile è limitato.

Poiché chiunque può aggiungere un criptoasset a un DEX, le truffe sono comuni. Ogni progetto legittimo può generare decine di progetti truffa con nomi simili, progettati per ingannare gli utenti e appropriarsi dei loro criptoasset. Inoltre, poiché questi progetti non devono essere verificati in anticipo, anche progetti apparentemente legittimi possono rivelarsi truffe. Per operare in questo spazio non regolamentato è essenziale fare ricerche approfondite prima di scambiare nuovi token. Questo può includere la lettura di *white paper*, la partecipazione a gruppi Telegram e Discord, la ricerca su X (ex Twitter) e la lettura di blog e articoli su Medium. Un white paper è un documento tecnico che descrive dettagliatamente un progetto di criptovaluta o blockchain, spiegando come funziona, quali problemi risolve e come intende raggiungere i suoi obiettivi. Fornisce informazioni su tecnologia, obiettivi, meccanismi di funzionamento e benefici del progetto, ed è utilizzato per comunicare con potenziali investitori e utenti.

Ogni azione su un DEX richiede l'uso di gas, il che può essere costoso. Su Ethereum il costo di una singola operazione di scambio può superare i 100 dollari, a seconda dello stato di congestione della rete. Più è congestionata la rete e più il prezzo è alto. Altre blockchain offrono commissioni più basse, ma hanno anche meno attività. Inoltre, poiché l'attività dei DEX è tutta *on-chain*, ovvero direttamente sulla blockchain, e pubblica, i bot possono monitorare le transazioni e anticipare gli ordini degli utenti, facendo aumentare i costi degli scambi.

Fornire liquidità a una *pool*, ossia "contenitori" di token che possono essere venduti e comprati, comporta un certo livello di rischio rispetto agli exchange centralizzati. Questo perché aggiungere o rimuovere liquidità da una pool richiede una certa quantità di gas, più tempo e competenze tecniche superiori. Per gli investitori meno esperti, potrebbe essere più redditizio mantenere semplicemente i propri criptoasset piuttosto che inserirli in una pool di liquidità.

## 1.3 Possibili attacchi ai DEX

I DEX possono essere soggetti a diversi tipi di attacchi, nonostante la loro struttura decentralizzata offra una maggiore sicurezza rispetto agli exchange centralizzati. Ecco alcuni dei possibili attacchi ai DEX:

### 1. Front-running

I bot monitorano le transazioni in sospeso e inseriscono ordini con commissioni di gas più elevate per essere eseguiti prima di altre transazioni. Questo permette ai bot di trarre profitto dalle variazioni di prezzo che si verificano a seguito di ordini di grandi dimensioni.

## 2. Attacco di liquidità bassa

I DEX con pool di liquidità limitata sono vulnerabili a manipolazioni dei prezzi. Un aggressore può spostare significativamente il prezzo di un asset con un ordine relativamente piccolo, creando opportunità di arbitraggio e sfruttando la mancanza di liquidità.

## 3. Attacchi di reentrancy

Questi attacchi sfruttano una vulnerabilità negli smart contracts, in cui un contratto chiama un altro contratto esterno che poi richiama il primo contratto prima che la prima transazione sia completata. Questo può portare a una situazione in cui i fondi vengono sottratti più volte.

## 4. Attacchi di arbitraggio

Gli attacchi di arbitraggio sfruttano differenze di prezzo temporanee tra diversi DEX o tra un DEX e un CEX. Gli aggressori possono spostare rapidamente asset tra le piattaforme per trarre profitto da queste discrepanze di prezzo, spesso utilizzando bot automatizzati.

## 5. Flash Loan Attacks

Un attacco con prestiti *flash* sfrutta la possibilità di prendere in prestito grandi somme di criptovaluta senza garanzie, eseguire una serie di operazioni complesse e restituire il prestito nella stessa transazione. Questo può manipolare i prezzi nelle pool di liquidità o sfruttare vulnerabilità negli smart contracts.

## 6. Attacchi di phishing

Gli utenti dei DEX possono essere ingannati tramite siti web falsi o link dannosi che imitano le piattaforme DEX legittime. Gli aggressori possono rubare le credenziali di accesso o ingannare gli utenti a eseguire transazioni non autorizzate.

## 7. Attacchi di governance

In alcuni DEX, le decisioni vengono prese attraverso meccanismi di governance basati sui token di governance al fine di democratizzare la gestione governativa del protocollo. Ovvero chi possiede più token di governance ha più potere decisionale. Gli aggressori possono acquisire una quantità significativa di token di governance per influenzare le decisioni a proprio favore, potenzialmente compromettendo la sicurezza o l'integrità del DEX.

## 8. Attacchi alla rete blockchain

Un DEX può essere vulnerabile agli attacchi alla rete blockchain sottostante, come gli attacchi del 51%, dove un gruppo di *miner* o validatori ottiene il controllo della maggioranza della rete, permettendo loro di alterare le transazioni e manipolare il mercato.

Affrontare queste minacce richiede un'attenta progettazione degli smart contracts, una robusta gestione della liquidità, l'implementazione di meccanismi di sicurezza avanzati e l'educazione degli utenti sulle pratiche di sicurezza.

## 2 Introduzione ai market making automatico

Un Market Making Automatico (AMM) è un tipologia di DEX utilizzato per facilitare le operazioni di trading senza la necessità di un intermediario centrale. Al posto di usare piattaforme di scambio convenzionali dove acquirenti e venditori presentano prezzi diversi per gli stessi asset, gli AMM utilizzano algoritmi matematici per determinare i prezzi degli asset e fornire liquidità. Alla base del

funzionamento di un AMM ci sono le pool di liquidità, queste sono raccolte di due o più asset (nel nostro caso due token) fornite dai partecipanti. Questi depositano asset in una pool e ricevono una fee, nel nostro caso del 0.3%, ogni volta che operazioni di trading vengono effettuate sulla pool in cui hanno depositato i loro asset. Gli AMM si differenziano in base al tipo di algoritmo utilizzato per la determinazione del prezzo.

Gli AMM sono una componente fondamentale della finanza decentralizzata, permettendo il trading di asset digitali in modo decentralizzato ed efficiente, pur presentando rischi dovuti a eventuali errori negli Smart Contract.

## 2.1 Tipologie di AMM

Gli AMM utilizzano diversi algoritmi per la determinazione dei prezzi degli asset. Gli algoritmi più diffusi, come quello di Uniswap, si basano su funzioni costanti come il prodotto o la somma.

### 2.1.1 AMM a prodotto costante

Gli AMM a prodotto costante si basano sulla formula  $x * y = k$ , dove  $x$  è la quantità del primo asset nella pool,  $y$  la quantità del secondo asset e  $k$  è una costante. Quando un utente vuole fare effettuare operazioni sugli asset interagisce con la pool. Se vuole scambiare una quantità  $\Delta x$  di asset  $x$  in asset  $y$  la quantità  $k$  deve rimanere invariata in seguito all'aggiunta di  $\Delta x$  nella pool, quindi si avrà:

$$(x + \Delta x)(y - \Delta y) = k. \quad (1)$$

Usando questa formula si può determinare la quantità di asset  $\Delta y$  che ottiene in cambio della quantità  $\Delta x$  immessa nella pool. La funzione corrispondente a questa tipologia di AMM è data da un'iperbole.

### 2.1.2 AMM a somma costante

Gli AMM a somma costante usano come algoritmo  $x + y = k$ . In questo caso se un utente vuole scambiare una quantità  $\Delta x$  di asset  $x$  e ottenere una quantità  $\Delta y$  di asset  $y$  la formula da seguire è data da:

$$(x + \Delta x) + (y - \Delta y) = k. \quad (2)$$

Questa tipologia di AMM impone dei limiti massimi alla quantità di asset presenti nella pool.

## 3 Implementazione

Il lavoro necessario a questo progetto è diviso principalmente in tre parti:

- Lo sviluppo degli Smart Contract scritti in Solidity che implementano i token, il marketplace e le challenge.
- Lo sviluppo di bot scritti in Python che compiono stocasticamente azioni di compravendita per simulare il rumore del mercato.
- Lo sviluppo di bot Telegram che permettono di effettuare azioni come comprare e vendere direttamente da smartphone.

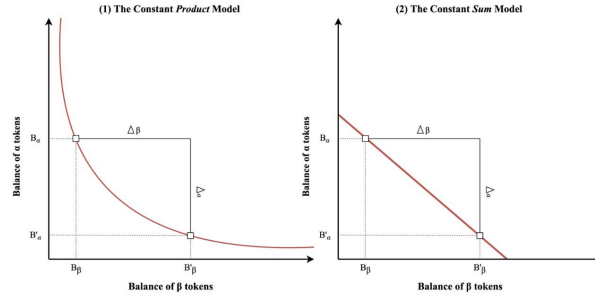


Figura 1: A sinistra grafico dell'andamento del prezzo in un AMM a prodotto costante a destra AMM a somma costante.

### 3.1 Smart Contract

Gli Smart Contract sono dei programmi che vengono eseguiti sulla blockchain di Ethereum, le cui funzioni possono essere chiamate da chiunque possieda un conto su Ethereum, a patto che egli paghi il gas necessario ad eseguire le suddette funzioni. Visto che le funzioni sono eseguite in maniera decentralizzata sulla blockchain, in maniera quindi trasparente e sicura, gli Smart Contract risultano molto utili per la creazione di nuovi token e l'implementazione di un Exchange su cui scambiare i suddetti token. Le componenti principali di uno Smart Contract sono:

- Il costruttore, attraverso cui viene inizializzato il contratto e vengono impostate le variabili iniziali.
- Le funzioni, che possono essere eseguite dopo che il contratto è stato inizializzato.
- I modificatori, che permettono di controllare che delle condizioni vengano soddisfatte prima di eseguire le funzioni.
- Gli eventi, ovvero oggetti contenenti dei parametri che vengono salvati alla fine dell'esecuzione delle funzioni, e che se indicizzati possono essere utilizzati per cercare l'evento stesso.

Di seguito descriviamo gli Smart Contract utilizzati.

#### 3.1.1 Token

Nell'Exchange realizzato, i Paycoin sono l'equivalente della moneta pubblica, mentre i Token sono la moneta di proprietà del loro creatore. Gli Smart Contract di entrambe seguono il protocollo ERC-20 e sono funzionalmente identici, venendo generati chiamando lo stesso costruttore con argomenti diversi durante il Deploy. Il suddetto costruttore permette di specificare il proprietario del token, il nome del token ed il suo simbolo. Altre variabili rilevanti sono `_balance`, ovvero un mapping da indirizzo a `uint256` che determina quanti token siano in possesso di quell'indirizzo, `_allowance`, ovvero un mapping da due indirizzi a `uint256` che specifica quanti Token il secondo indirizzo possa prelevare dal primo e `isMinter`, un mapping da indirizzo a `bool` che determina se quell'indirizzo ha il permesso di coniare nuovi token.

Abbiamo inoltre due modificatori, `onlyOwner` e `onlyMinter`, usati per assicurarsi che alcune funzioni possano essere chiamate solamente rispettivamente dal proprietario del Token e dai minter abilitati.

Le principali funzioni sono le seguenti:

- `minting`, chiamabile solo dal proprietario e dai minter, e attraverso cui vengono conati nuovi token. Questa funzione inoltre emette l'evento `Mint`, che registra minter, il destinatario e la quantità di token conati.
- `transfer`, utilizzata per trasferire token dal proprio indirizzo ad un altro. Questa funzione emette l'evento `Transfer`, che registra il mandante, il destinatario e la quantità di token trasferiti.
- `transferFrom`, utilizzata per trasferire token da un indirizzo diverso dal proprio, e che necessita di sufficiente allowance. Anche questa funzione emette l'evento `Transfer` come la precedente.
- `increaseAllowance` e `decreaseAllowance`, usati rispettivamente per aumentare e diminuire l'allowance.
- `balanceOf` permette di visualizzare il balance di qualsiasi indirizzo.

### 3.1.2 Marketplace

Lo Smart Contract Marketplace implementa un DEX con AMM del tipo a prodotto costante, come descritto nella sezione precedente. Tramite mapping viene associato ad ogni indirizzo le variabili `liquidity`, `price`, e `k`, che descrivono rispettivamente la quantità di paycoin presenti nella pool associata al token con quell'indirizzo, il prezzo del token (ovvero il rapporto tra liquidità e token presenti nella pool), e la `k` associata al token; la quantità di token presenti nella pool corrisponde al numero di token presenti sul balance associato all'indirizzo del contratto Marketplace. E' inoltre presente una fee del 0.3%, ovvero ad ogni transazione che coinvolge un Token, viene inviato lo 0.3% dei paycoin al proprietario del Token.

Abbiamo come modificatori `onlyOwner` e `Open`; quest'ultimo assicura che le relative funzioni possano essere chiamate con successo solo durante gli orari di apertura del Marketplace.

Le principali funzioni sono le seguenti:

- `addToken`, funzione chiamabile solo dal proprietario del marketplace che permette di inizializzare una pool associata a un Token.
- `buy`, utilizzabile per comprare dei token usando paycoin. Emette un evento in cui vengono salvati il mittente, il token acquistato e la quantità di paycoin spesi.
- `sell`, utilizzabile per vendere dei token ottenendo paycoin. Emette un evento in cui vengono salvati il mittente, il token venduto e la quantità di paycoin ottenuti.
- `swap`, utilizzabile per scambiare dei token con altri token. Emette un evento in cui vengono salvati il mittente, i token scambiati, le loro quantità ed il loro valore in paycoin.
- `mint`, chiamabile dal proprietario del Token per aumentare la quantità sia di token che di paycoin presenti nella pool corrispondente, mantenendo il prezzo invariato. I token e i paycoin aggiunti alla pool in questa maniera provengono dal saldo del proprietario del Token. La



quantità massima di token nella pool che si possono raggiungere tramite questa operazione è data da 1.5 volte la quantità iniziale. Viene anche emesso un evento che salva il Token, il mittente e le quantità di paycoin e di token coinvolte.

- **burn**, chiamabile dal proprietario del Token per diminuire la quantità sia di token che di paycoin presenti nella pool corrispondente, mantenendo il prezzo invariato. I token e i paycoin tolti alla pool in questa maniera sono trasferiti al proprietario del Token. La quantità minima di token nella pool che si possono raggiungere tramite questa operazione è data da 0.5 volte la quantità iniziale. Viene anche emesso un evento che salva il Token, il mittente e le quantità di paycoin e di token coinvolte.
- **day\_mint**, funzione che permette una volta al giorno al proprietario di ogni Token di generare un numero di paycoin uguale al N10% della quantità presente nella pool del Token, dove N è il numero del giorno della sfida.

E' importante notare che quando vengono comprati e venduti dei token la quantità di paycoin corrispondente non è mai esattamente uguale alla quantità di token moltiplicata per il prezzo. Infatti dato che la  $k$ , ovvero il prodotto tra la quantità di token e di paycoin deve rimanere invariato dalle transazioni, la quantità di paycoin corrispondente viene calcolata attraverso una funzione non lineare. Consideriamo  $x, y$  tali che  $xy = k$ , ed  $x' = x + \Delta x$ ,  $y' = y - \Delta y$ . Imponendo  $x'y' = k$  troviamo:

$$\Delta y = y - \frac{k}{x + \Delta x}. \quad (3)$$

Questo ci permette di calcolare quanti paycoin vanno aggiunti alla pool quando ne viene tolta una certa quantità di token per mantenere  $k$  costante, e viceversa. Scambiare un token con un altro tramite **swap** corrisponde funzionalmente a vendere il primo e usare i paycoin ottenuti per comprare il secondo; questo è tuttavia molto utile al proprietario del token, visto che a lui non è permesso comprare e vendere il proprio token tramite le funzioni **buy** e **sell**.

### 3.1.3 Challenge

Lo Smart Contract Challenge permette di lanciare e rispondere a delle sfide. La sfida viene vinta chiamando per primo l'apposita funzione dopo minimo 20 secondi che la sfida è stata lanciata. Per permettere la creazione e gestione di più sfide in contemporanea abbiamo definito all'interno del contratto uno struct **ChallengeData** contenente tutte le informazioni pertinenti ad una particolare sfida, come l'orario in cui essa è stata lanciata, i partecipanti e se essa sia già stata vinta e da chi. Quando viene lanciata una sfida quindi viene creato un oggetto **ChallengeData** ed aggiunto ad una lista, e quando si vuole rispondere ad una sfida bisogna specificare l'indice nella lista della sfida a cui si vuole rispondere. E' anche presente **attempts**, un mapping da indirizzo e uint256 a bool, che viene usato per registrare se l'indirizzo ha già cercato di vincere la sfida con un tale indice. In Le funzioni implementate sono quindi le seguenti:

- **one\_vs\_one**, con cui viene lanciata una sfida contro una sola altra persona. Vengono conati 1000 paycoin e trasferiti allo sfidante, e viene emesso un evento **One\_vs\_One** che contiene lo sfidante, lo sfidato e l'indice della sfida.
- **one\_vs\_two**, con cui viene lanciata una sfida contro due altre persone. Vengono conati 2000 paycoin e trasferiti allo sfidante, e viene emesso un evento **One\_vs\_Two** che contiene chi lo sfidante, i due sfidati e l'indice della sfida.

- **win\_one**, funzione usata per vincere una sfida **one\_vs\_one**, specificandone l'indice. Fallisce se viene chiamata prima dei 20 secondi da quando la sfida è stata lanciata, se il mittente non è tra i partecipanti alla sfida, se il mittente ha già provato a vincere la sfida e se la sfida è già stata vinta. In caso di successo vengono conati e trasferiti al vincitore 10000 paycoin, e viene emesso l'evento di vittoria **Win\_One**, su cui vengono registrati il vincitore e la quantità di paycoin vinti.
- **win\_two**, funzione usata per vincere una sfida **one\_vs\_two**, specificandone l'indice. Fallisce se viene chiamata prima dei 20 secondi da quando la sfida è stata lanciata, se il mittente non è tra i partecipanti alla sfida, se il mittente ha già provato a vincere la sfida e se la sfida è già stata vinta. In caso di successo vengono conati e trasferiti al vincitore 50000 paycoin, e viene emesso l'evento di vittoria **Win\_Two**, su cui vengono registrati il vincitore e la quantità di paycoin vinti.

### 3.2 Bot e script

Per simulare un mercato reale abbiamo costruito 100 bot utilizzando la libreria Brownie di Python, ognuno con il proprio indirizzo, che fanno azioni di compravendita in maniera randomica. Per l'esattezza ogni secondo viene estratto un bot casuale dai 100 e gli viene fatto compiere una azione casuale tra buy, sell e swap dei vari token. I bot sono stati inizializzati con una quantità random di paycoin uniformemente distribuita tra 1000 e 100000, e ad ogni operazione muovono tra il 20% e l'80% dei token o paycoin che possiedono relativi a quell'operazione. Dato che inizialmente i bot erano in possesso solamente di paycoin, e non di altri token, il mercato che simulavano, almeno per le prime ore era quello relativo al lancio di dei nuovi token.

Abbiamo inoltre scritto uno script Python in grado di salvare su file tutte le informazioni relativi al mercato per successive analisi. Il nostro gruppo ha anche sviluppato uno script in grado di rispondere automaticamente alle sfide, in modo da non rimanere vulnerabili nei momenti in cui nessuno era pronto a rispondere.

### 3.3 Integrazione Telegram

Abbiamo utilizzato Python per scrivere due bot Telegram per poter effettuare operazioni direttamente da smartphone senza dover passare necessariamente da un computer. Il primo, Bosco, è in grado di effettuare azioni di compravendita sul Marketplace, oltre che a chiamare le funzioni **mint** e **burn**, e di monitorare in tempo reale le principali informazioni sul mercato quali i prezzi dei token, la dimensione delle pool ed il proprio balance. Il secondo, Gabigol, è in grado di lanciare e rispondere alle sfide, avvisando anche quando viene lanciata una sfida che coinvolge l'user.

## 4 Analisi della gara

### 4.1 Andamento dei prezzi

Nella Figura 2 è riportato l'andamento del prezzo dei 3 token durante lo svolgimento della gara. Si può notare che all'inizio i prezzi sono cresciuti notevolmente questo è dovuto al fatto che inizialmente i bot avevano solo PC e quindi le prime azioni di trading dei bot erano necessariamente dei buy. Durante il primo giorno ci sono state grosse oscillazioni dei prezzi, specialmente verso la fine del

mercato dove è stato raggiunto il picco massimo del prezzo dei tre token. Il sistema di monitoring utilizzato monitorava l'andamento dei prezzi anche a mercato chiuso, per questo motivo si possono notare dei tratti dove il prezzo rimane invariato. Si può notare che a partire dal terzo giorno sono diminuite notevolmente le oscillazioni di prezzo durante la giornata, questo può essere dovuto a un minore volume di token comprati o venduti. Infine si può notare che l'ultimo giorno i prezzi sono scesi perchè i partecipanti alla gara hanno venduto tutti i token che possedevano facendo aumentare i token nelle diverse liquidity pool e facendo crollare i prezzi dei token. Prezzi all'apertura del mercato dei singoli token:

- Il token LDR partiva da un prezzo di 1000.0 PC.
- Il token QTC inizialmente aveva un prezzo di 10,00 PC.
- Il token BOR è stato lanciato con un prezzo di 0,10 PC.

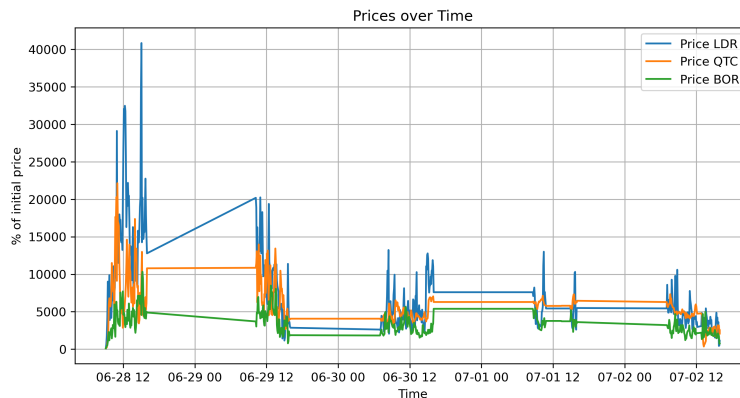


Figura 2: Andamento dei prezzi dei tre token in funzione del tempo.

## 4.2 Andamento dei wallet dei partecipanti

Nel Figura 3 è riportato l'andamento dei PC delle tre squadre. Come si può notare il portafoglio LDR e BOR è cresciuto inizialmente per poi mantenersi abbastanza costante fino alla chiusura dell'ultimo giorno quando sono stati venduti tutti gli asset in loro possesso in modo tale da aumentare il loro balance di PC. Il balance del gruppo QTC si è mantenuto costante nei primi giorni per poi subire una forte crescita nella giornata finale dovuta a un'attenta strategia di vendita dei token combinata con azioni di burn\_stake e swap in modo tale da poter trasferire PC dalla pool al balance del gruppo.

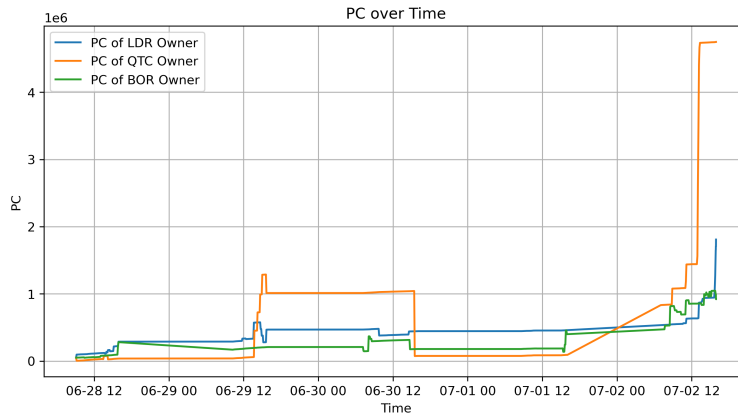


Figura 3: Andamento dei PC presenti nei tre wallet in funzione del tempo.

Nella Figura 4 è riportato l'andamento complessivo del portafoglio dato dalla somma dei PC con il valore dei Token posseduti. I due grafici risultano essere molto diversi, nel secondo sono presenti molte più oscillazioni dovute al variare dei prezzi. Specialmente per LDR le oscillazioni hanno toccato picchi molto alti durante i primi due giorni del mercato, questo fenomeno è dovuto al prezzo di LDR nelle fasi iniziali della sfida. Il portafoglio di QTC è variato molto durante i primi due giorni per poi stabilizzarsi e crescere nelle fasi conclusive della gara, il portafoglio di BOR è leggermente cresciuto durante tutto il mercato. Questo grafico rappresenta tuttavia una situazione virtuale in quanto la ricchezza dei gruppi è espressa in PC, il possesso degli asset di per se non genera ricchezza fino a quando questi non vengono venduti al prezzo di mercato.

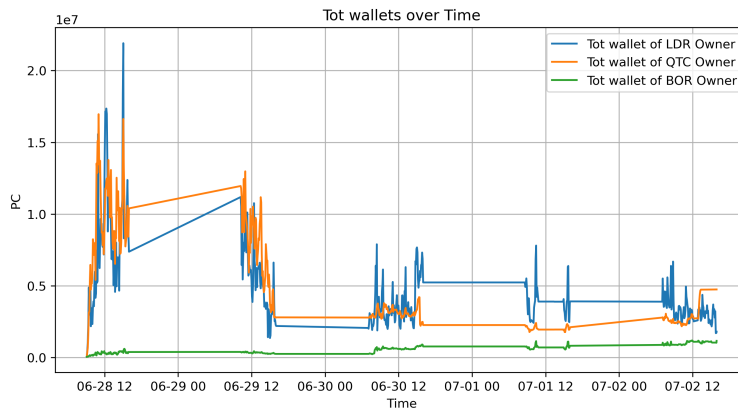


Figura 4: Andamento complessivo del portafoglio in funzione del tempo.

In Appendice sono riportati gli andamenti dei tre token, delle liquidity pool e della quantità di token presenti nelle diverse pool.

### 4.3 Guadagni e perdite per ciascun utente

Nella Tabella 1 sono riportati i guadagni per ciascun partecipante e la somma dei guadagni dei bot. In Appendice sono riportati i dati dei singoli bot. Tutti e tre i partecipanti alla gara hanno chiuso la sfida in positivo, i bot al contrario hanno un bilancio negativo. Questo è dovuto al fatto che i bot effettuavano azioni completamente casuali comprando e vendendo nei momenti meno opportuni. La squadra che ha avuto un guadagno maggiore è stata QTC seguita da LDR e BOR.

squadra	PC iniziali	PC finali	Differenza	% differenza
QTC	50000	4748152	4698152	9396%
BOR	50000	1014077	964077	1928%
LDR	50000	1912602	1862602	3725%
tot bot	458183	109040	-349143	-76,20%

Tabella 1: Guadagni o perdite per ciascun partecipante e per i bot

Come si può vedere dalla Figura 5 i guadagni sono dovuti dalle fee, dalla vittoria delle challenge, dal day mint e dal trading. I guadagni dovuti dalle fee sono paragonabili, questo è dovuto al fatto che la maggior parte delle operazioni è stata svolta dai bot. Il gruppo BOR ha guadagnato principalmente dal day mint (nonostante non abbia usato il day mint un giorno) e dalle challenge e in misura minore dalle fee e dal trading. Il gruppo LDR ha guadagnato principalmente dal trading e dal day mint e solo in parte dalle challenge e dalle fee. Per il gruppo QTC i guadagni principali sono dati dalle strategie di trading e dal day mint, successivamente dalla vittoria delle challenge e infine dalle fee.

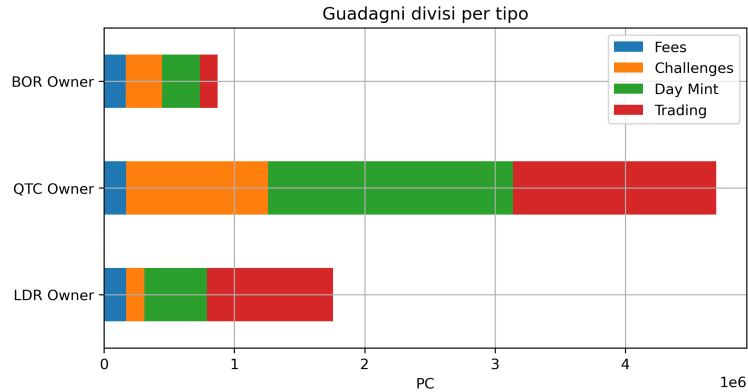


Figura 5: Dettaglio delle fonti dei guadagni di ogni utente.

### 4.4 Swap effettuati

Nella Tabella 2 sono riportati il numero di operazioni buy e sell effettuati in ciascuna pool.

pool	$n^\circ$ buy	$n^\circ$ sell
QTC	1224	1212
BOR	1249	1210
LDR	1272	1236

Tabella 2: numero di operazioni buy e sell nelle pool

La quantità di operazioni effettuate su ciascuna pool è equiparabile, ciò è dovuto al fatto che la maggior parte delle operazioni sono state effettuate dai bot che sceglievano casualmente se vendere o comprare e da quale pool. Si può notare che in tutte le pool sono stati effettuati più operazioni buy rispetto a operazioni sell, in aggiunta siccome all'inizio i bot non possedevano token ma solo PC le prime azioni sell sono state "vuote". Questo fa pensare che generalmente durante le operazioni buy sono stati acquistati meno token rispetto a quando sono state effettuate operazioni sell. Nella Tabella 3 sono state analizzate le operazioni swap, in particolare nelle colonne è indicato la pool da cui è stato preso il token in ingresso mentre nelle righe quello in uscita.

in\out	QTC	BOR	LDR
QTC	0	610	655
BOR	683	0	596
LDR	584	625	0

Tabella 3: numero di operazioni swap tra le pool

Dall'analisi della tabella si deduce che le pool di QTC e LDR sono state interessate maggiormente da swap in entrata al contrario nella pool di BOR sono stati effettuati più swap in uscita. Complessivamente sono state effettuate 3745 operazioni buy, 3658 sell e 3753 swap.

## 4.5 Fee pagate

Nella Tabella 4 sono state riportate le fee pagate da ciascun utente a ogni squadra e il totale delle fee pagate. Il dettaglio delle fee pagate dai bot è riportato in Appendice

squadra	fee a LDR	fee a QTC	fee a BOR	fee totali
LDR	3639	5244	3347	12230
QTC	8125	3731	4541	16397
BOR	8938	2808	669	12415
bot	145606	157159	156871	459636

Tabella 4: Fee pagate da ciascun componente e bot

Il totale delle fee pagate dal gruppo LDR e dal gruppo BOR è uguale mentre il gruppo QTC ha pagato più fee. Il totale delle fee pagate da ciascun gruppo è indice dell'attività di un gruppo, infatti le fee rappresentano il 0.3% dei PC spesi durante le transazioni. Da questa analisi si può dedurre che il volume totale di operazioni effettuati da QTC è stato superiore rispetto a quello di BOR e

LDR. Inoltre il gruppo LDR ha guadagnato dalle fee 166308 PC, il gruppo QTC 168942 PC e il gruppo BOR 165428 PC.

## 4.6 Challenge

Nella Tabella 5 sono riportate le statistiche relative alle challenge lanciate e vinte da ciascuna squadra

squadra	1v1 lanciate	1v2 lanciate	1v1 vinte	1v2 vinte	PC guadagnati
QTC	10	10	11	19	1090000
BOR	10	10	5	4	280000
LDR	4	4	8	1	142000

Tabella 5: Challenge lanciate e vinte da ciascun gruppo suddivise per tipologia.

Come si può notare dalla tabella il gruppo QTC ha guadagnato di più dalle challenge (1090000 PC) seguito da BOR (280000 PC) e LDR (142000 PC). I gruppi QTC e BOR hanno lanciato tutte le challenge a loro disposizione, 10 per tipologia, mentre il gruppo LDR ha lanciato solo 4 challenge per tipo. Confrontando i dati relativi alle challenge con quelli dei guadagni di ciascuna squadra si può notare che il 23,2% dei guadagni di QTC, il 29,0% dei guadagni di BOR e il 7,6% dei guadagni di LDR sono stati ottenuti grazie alle challenge. Nelle tabelle 6 e 7 sono stati riportati i dettagli di ogni challenge lanciata.

giorno	tipo	sfidante	sfidato	vincitore
3	1v1	QTC	BOR	QTC
3	1v1	QTC	BOR	QTC
3	1v1	QTC	BOR	QTC
3	1v1	QTC	BOR	QTC
3	1v1	QTC	BOR	QTC
3	1v1	QTC	LDR	QTC
3	1v1	QTC	LDR	QTC
3	1v1	QTC	LDR	QTC
3	1v1	QTC	LDR	QTC
3	1v1	QTC	LDR	QTC
3	1v1	BOR	LDR	BOR
3	1v1	BOR	LDR	LDR
3	1v1	BOR	LDR	LDR
3	1v1	BOR	LDR	LDR
3	1v1	BOR	LDR	LDR
3	1v1	BOR	LDR	LDR
3	1v1	BOR	QTC	QTC
3	1v1	LDR	BOR	LDR
3	1v1	LDR	BOR	LDR
3	1v1	LDR	BOR	LDR
4	1v1	LDR	BOR	LDR
4	1v1	BOR	LDR	BOR
4	1v1	BOR	LDR	BOR
4	1v1	BOR	LDR	BOR
4	1v1	BOR	LDR	BOR

Tabella 6: Sfide 1v1

giorno	tipo	sfidante	vincitore
3	1v2	QTC	QTC
3	1v2	QTC	QTC
3	1v2	QTC	QTC
3	1v2	QTC	QTC
3	1v2	QTC	QTC
3	1v2	QTC	QTC
3	1v2	QTC	QTC
3	1v2	QTC	QTC
3	1v2	QTC	QTC
3	1v2	QTC	QTC
3	1v2	BOR	QTC
3	1v2	BOR	QTC
3	1v2	LDR	BOR
4	1v2	BOR	QTC
4	1v2	BOR	QTC
4	1v2	BOR	BOR
4	1v2	BOR	QTC
4	1v2	BOR	QTC
4	1v2	BOR	BOR
4	1v2	BOR	QTC
4	1v2	BOR	BOR
4	1v2	LDR	QTC
4	1v2	LDR	LDR
4	1v2	LDR	QTC

Tabella 7: Sfide 1v2

Il gruppo QTC ha lanciato e vinto tutte le challenge nel primo giorno in cui erano disponibili (il terzo giorno di mercato). Il gruppo BOR le ha lanciate tutte durante il terzo e il quarto giorno di mercato, il gruppo LDR ha lanciato solo 4 sfide per tipologia e non ne ha lanciate nessuna durante il quinto giorno di mercato.

## 5 Strategia gruppo QTC

Il gruppo QTC ha attuato diverse strategie:

- All'apertura del mercato è stato fatto un burn\_stake massimo, equivalente al 50% della pool in modo tale da aumentare la liquidità del gruppo e poter acquistare subito LDR e BOR al prezzo di apertura del mercato.
- Aumentare la liquidity pool attraverso azioni di mint\_stake prima di effettuare il day\_mint in modo tale da guadagnare più PC.



- Lanciare tutte le sfide il primo giorno disponibile (il terzo del mercato) in modo tale da avere più PC per i day\_mint, questa strategia ha incontrato delle difficoltà dovute a un errore nel codice del day\_mint che non ha permesso l'utilizzo nel terzo e quarto giorno di mercato.
- Sviluppo di un bot automatico per rispondere alle challenge in modo tale da non dover essere costantemente attivi, questo bot ha permesso di vincere 30 challenge su 35.
- In chiusura di mercato vendita dei diversi token e cicli di burn\_stake e swap in modo da diminuire la liquidity pool e aumentare il balance di PC della squadra.

## 6 Comportamenti anomali

Durante lo svolgimento della gara sono stati riscontrati diversi problemi che hanno richiesto modifiche ai programmi sviluppati.

### 6.1 Impossibilità di utilizzare la rete SepoliaETH

Originariamente il progetto doveva essere lanciato sulla rete di test di Sepolia, tuttavia poco prima dell'inizio della sfida i tre gruppi si sono resi conto che 100 ETH non erano sufficienti per effettuare transazioni per 5 giorni, di conseguenza è stato deciso di utilizzare ganache e di creare un server in modo tale che tutti potessero accedere al progetto.

### 6.2 Funzione day\_mint

Tutte le operazioni che coinvolgono il DEX potevano essere effettuate tramite il bot Telegram Bosco ad eccezione della funzione day\_mint, per ovviare a questo problema è stato sviluppato in corso d'opera uno script Python per lanciare questa funzione. Inoltre nel terzo e nel quarto giorno non è stato possibile effettuare il day\_mint a causa di un errore in un parametro dello Smart Contract in Solidity. Insieme alle altre squadre è stato deciso di non sistemare questo errore in corso d'opera per rimanere fedeli alla filosofia della blockchain.

### 6.3 Challenge

Gabigol, bot Telegram sviluppato per gestire le challenge rallentava il server fino a renderlo quasi inutilizzabile. Per questo motivo si è deciso di far andare lo script delle challenge in locale, offrendo la possibilità a tutti di far girare eventuali script sviluppati. Di comune accordo è stato deciso che i guadagni relativi alle challenge venivano aggiunti tramite un apposito script alla fine del mercato giornaliero. Per risolvere i problemi relativi alle challenge è stato necessario bloccare il loro utilizzo durante i primi due giorni in modo tale da riuscire a risolvere i problemi e a testare in locale la loro nuova implementazione.

### 6.4 Memoria del server

Tutto il progetto è stato sviluppato su un server e non sulla rete di test di Sepolia, arrivati a metà del quinto giorno non era più possibile effettuare operazioni tramite i bot Telegram, dopo un'analisi si è giunti a capire che la memoria del server era finita. Per risolvere questo problema sono stati

eliminati alcuni dati relativi al primo giorno che erano stati salvati due volte per controllare che il sever funzionasse correttamente.

## **6.5 Malfunzionamento iniziale dei bot**

Circa due ore dopo l'inizio della sfida i bot hanno smesso di funzionare e non è stato possibile riattivarli immediatamente, dopo una riunione di gruppo è stato deciso di bloccare la sfida e di far ripartire i bot dopo aver sistemati alcuni problemi che si erano presentati sullo script dei bot. La gara è stata quindi fatta ripartire da zero, in seguito i problemi ai bot non sono più stati riscontrati.

## **7 Conclusione**

In conclusione il progetto di sviluppo di token ERC20 e del DEX con annesso trading è stato molto interessante. Ci ha permesso di migliorare le nostre capacità di programmazione in Solidity e Python e di collaborazione all'interno dei gruppi mantenendo comunque uno spirito competitivo. Sviluppare un progetto così articolato è stata una sfida ma riuscire a coordinare tutte le sue sottoparti è stato molto gratificante. La strategia utilizzata da QTC ha permesso di ottenere un guadagno superiore alla somma dei guadagni degli altri due gruppi. Un altro aspetto estremamente interessante di questa sfida è stato l'approccio alla gara e alla gestione degli imprevisti occorsi durante i cinque giorni.

## 8 Appendice

### 8.1 Andamento dei token nei wallet dei partecipanti

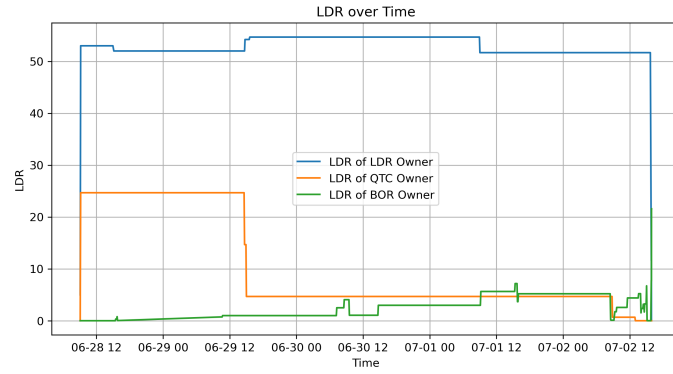


Figura 6: Andamento del token LDR nel portafoglio delle squadre in funzione del tempo.

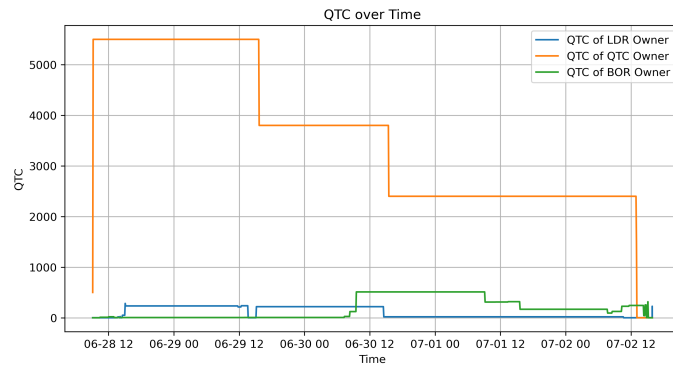


Figura 7: Andamento del token QTC nel portafoglio delle squadre in funzione del tempo.

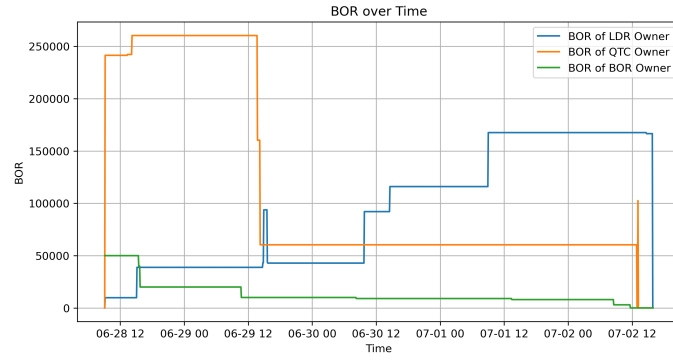


Figura 8: Andamento del token BOR nel portafoglio delle squadre in funzione del tempo.

## 8.2 Andamento delle pool

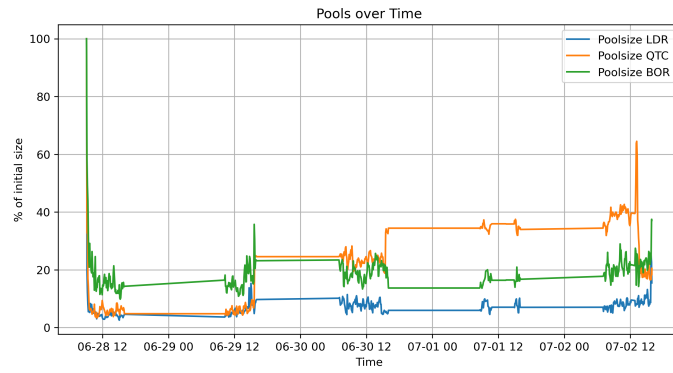


Figura 9: Percentuale sui valori iniziali dei token presenti nelle pool in funzione del tempo.

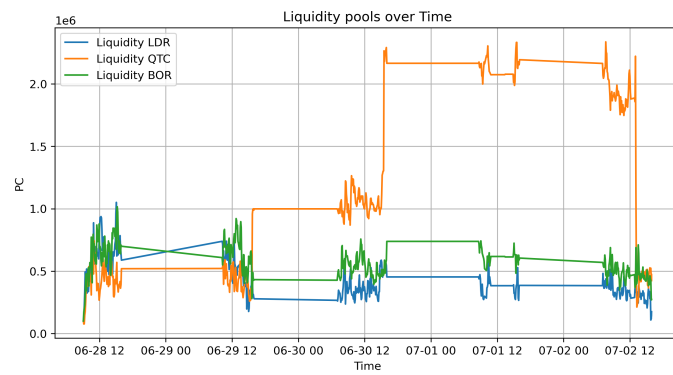


Figura 10: Andamento delle tre liquidity pool in funzione del tempo.

### 8.3 Analisi dei bot

In queste tabelle sono riportati i dati dei guadagni o delle perdite di ciascun bot

squadra	PC iniziali	PC finali	Differenza	% differenza
0	5763	188	-5576	-96,7
1	4030	634	-3396	-84,3
2	1835	410	-1425	-77,7
3	3195	1230	-1964	-61,5
4	2887	830	-2057	-71,3
5	5854	69	-5784	-98,8
6	1342	189	-1153	-85,9
7	3357	628	-2730	-81,3
8	3765	70	-3695	-98,2
9	807	106	-702	-86,9
10	6654	294	-6360	-95,6
11	6082	8	-6074	-99,9
12	448	518	70	15,7
13	6085	3075	-3009	-49,5
14	303	44	-259	-85,5
15	6606	209	-6397	-96,8
16	8542	300	-8242	-96,5
17	5038	1638	-3401	-67,5
18	4047	123	-3924	-97,0
19	4012	528	-3484	-86,8
20	555	3	-551	-99,4
21	9060	157	-8903	-98,3
22	6395	304	-6091	-95,2
23	9069	674	-8395	-92,6
24	2880	1434	-1446	-50,2
25	3094	294	-2801	-90,5
26	5384	3696	-1688	-31,4
27	4926	635	-4290	-87,1
28	3038	184	-2853	-93,9
29	5303	5876	573	10,8
30	7463	105	-7358	-98,6
31	1135	9	-1126	-99,2
32	7518	11876	4358	58,0
33	5386	1026	-4360	-81,0

Tabella 8: Guadagni o perdite per i bot 0-33

squadra	PC iniziali	PC finali	Differenza	% differenza
34	6884	607	-6277	-91,2
35	823	54	-768	-93,4
36	2534	1185	-1349	-53,2
37	3924	8857	4933	125,7
38	5624	4	-5620	-99,9
39	2326	831	-1495	-64,3
40	6755	3220	-3535	-52,3
41	5993	1513	-4480	-74,8
42	881	244	-637	-72,3
43	9918	168	-9750	-98,3
44	3654	183	-3471	-95,0
45	9651	1447	-8204	-85,0
46	4567	606	-3961	-86,7
47	327	6	-321	-98,2
48	1110	21	-1089	-98,1
49	8737	6	-8731	-99,9
50	556	4310	3754	675,2
51	2987	378	-2609	-87,3
52	6555	277	-6277	-95,8
53	4100	197	-3902	-95,2
54	2601	13	-2587	-99,5
55	3349	4476	1126	33,6
56	6316	227	-6089	-96,4
57	6814	2189	-4625	-67,9
58	4955	81	-4874	-98,4
59	1179	94	-1086	-92,1
60	4105	0	-4104	-100,0
61	611	369	-242	-39,6
62	6882	2244	-4638	-67,4
63	3761	5090	1329	35,3
64	4824	1302	-3522	-73,0
65	4301	11	-4290	-99,8
66	6596	222	-6374	-96,6
67	3923	17	-3906	-99,6

Tabella 9: Guadagni o perdite per i bot 34-66

squadra	PC iniziali	PC finali	Differenza	% differenza
68	450	21	-428	-95,2
69	622	106	-515	-82,9
70	3136	2210	-926	-29,5
71	3507	366	-3141	-89,6
72	378	10	-368	-97,3
73	6053	652	-5401	-89,2
74	5033	354	-4679	-93,0
75	2644	682	-1962	-74,2
76	7532	1339	-6193	-82,2
77	5279	1915	-3363	-63,7
78	693	605	-88	-12,7
79	9554	1987	-7566	-79,2
80	4256	600	-3656	-85,9
81	8594	253	-8341	-97,1
82	4479	45	-4433	-99,0
83	7157	1172	-5984	-83,6
84	1856	52	-1804	-97,2
85	1411	68	-1344	-95,2
86	7377	73	-7304	-99,0
87	7934	1382	-6552	-82,6
88	6229	5714	-515	-8,3
89	2326	741	-1585	-68,2
90	9296	4430	-4866	-52,3
91	7614	1061	-6552	-86,1
92	8399	700	-7699	-91,7
93	685	54	-631	-92,2
94	4315	1131	-3184	-73,8
95	6152	206	-5946	-96,7
96	2970	210	-2760	-92,9
97	8070	707	-7364	-91,2
98	4263	385	-3878	-91,0
99	9945	1996	-7949	-79,9

Tabella 10: Guadagni o perdite per i bot 68-99

In queste tabelle sono riportati i dati relativi al pagamento delle fee di ciascun bot

squadra	fee a LDR	fee a QTC	fee a BOR	fee totali
0	898	851	1052	2801
1	721	928	1134	2783
2	338	317	141	796
3	1702	3918	3333	8953
4	315	294	379	988
5	1311	814	820	2945
6	151	147	181	479
7	159	1350	1874	3383
8	358	918	703	1979
9	117	196	166	479
10	889	1057	1610	3556
11	1131	852	1313	3296
12	581	726	743	2050
13	6416	6753	5513	18682
14	48	60	41	149
15	1182	1177	721	3080
16	910	1567	724	3201
17	1611	1328	1802	4741
18	363	357	317	1037
19	563	888	790	2241
20	68	54	48	170
21	2682	2925	2922	8529
22	5560	7615	9432	22607
23	1232	637	625	2494
24	828	571	797	2196
25	1785	1336	582	3703
26	2650	2255	2578	7483
27	1147	922	869	2938
28	816	867	715	2398
29	3260	2294	3735	9289
30	1074	988	1746	3808
31	466	405	231	1102
32	6880	8416	6530	21826
33	2286	681	1742	4709

Tabella 11: Fee pagate dai bot 0-33



squadra	fee a LDR	fee a QTC	fee a BOR	fee totali
34	1255	1186	1519	3960
35	135	74	68	277
36	2502	3369	2202	8073
37	3140	3084	2898	9122
38	806	1384	1299	3489
39	1408	1880	1948	5236
40	1142	1179	1545	3866
41	1324	1737	2272	5333
42	226	204	194	624
43	2020	1894	3281	7195
44	3358	4308	3796	11462
45	4957	2815	4506	12278
46	1112	1383	1558	4053
47	59	72	111	242
48	77	117	119	313
49	737	572	850	2159
50	2078	2243	2090	6411
51	506	1106	575	2187
52	2393	2359	1963	6715
53	326	1026	969	2321
54	2177	2975	1574	6726
55	649	1253	1440	3342
56	2416	2256	1863	6535
57	1844	2331	1979	6154
58	654	652	944	2250
59	404	348	497	1249
60	756	1062	1276	3094
61	262	158	227	647
62	1329	1064	622	3015
63	1962	2121	1197	5280
64	1812	1659	1395	4866
65	912	759	785	2456
66	5647	2349	5648	13644
67	951	558	804	2313

Tabella 12: Fee pagate dai bot 34-67

squadra	fee a LDR	fee a QTC	fee a BOR	fee totali
68	74	63	38	175
69	173	129	141	443
70	2274	2629	1998	6901
71	574	650	865	2089
72	621	469	434	1524
73	1524	1931	1066	4521
74	466	694	481	1641
75	1465	2090	1644	5199
76	2221	2934	3142	8297
77	641	699	756	2096
78	371	240	210	821
79	2614	2267	3795	8676
80	753	316	759	1828
81	684	1380	1052	3116
82	346	212	317	875
83	1681	1758	1570	5009
84	5771	8176	6629	20576
85	79	184	201	464
86	2035	2484	2222	6741
87	1029	1219	1000	3248
88	1220	2100	2872	6192
89	612	813	761	2186
90	2923	3529	2331	8783
91	2757	3249	2154	8160
92	2197	1352	2030	5579
93	183	134	156	473
94	1855	1945	1916	5716
95	949	974	511	2434
96	456	422	382	1260
97	761	1164	1070	2995
98	2430	2481	1859	6770
99	2033	2471	2586	7090

Tabella 13: Fee pagate dai bot 68-99