

Programa IT Academy – Processo Seletivo – Edição #20

Nome Completo: Lucas Navarini Lanferdini

E-mail: lanfer2002@gmail.com

Exercício Técnico

RESUMO DA SOLUÇÃO

Introdução:

Um resumo do arquivo enviado por e-mail é que temos um desafio de desenvolvimentos de um sistema de controle de apostas, parecido com a Mega-Sena.

Nesse sistema é necessário realizar 6 funcionalidades básicas, que são: 1- iniciar uma nova edição de sorteio, 2- registrar uma nova aposta com nome, cpf e 5 números informados pelo usuário, 3- listar todas as apostas realizadas, 4- executar o sorteio, sorteando 5 números, se não existir nenhum ganhador sortear mais um número, realizar esse processo 25 vezes até achar um vencedor, 5- realizar a apuração do sorteio, mostrando a lista de números sorteados, quantas rodadas de sorteio foram realizadas, quantidade de apostas vencedoras, lista de apostas vencedoras, lista de todos os números apostados, 6- premiar as apostas vencedoras.

Para resolver o desafio, utilizei o ambiente de programação Jupyter Notebook, pelo Anaconda Navigator, pois o acesso para bibliotecas importantes usadas no código se tornam mais fáceis, também foi usado o MySQL Workbench para registro dos dados e a API do Metamask, para realizar a premiação. As bibliotecas necessárias foram: Pandas para criar dataframes, random para sortear números aleatórios, Time para dar função de sleep() e deixar os output mais visíveis, numpy para trabalhar melhor com números, web3 para se conectar a Blockchain e realizar suas operações e mysql.connector para se conectar ao banco de dados.

Começando:

Antes de começar a realizar o código eu segui alguns passos antes. O primeiro foi ler e tentar desenvolver uma lógica para seguir durante todo o código, o segundo passo foi criar um ambiente de desenvolvimento com o arquivo principal, o arquivo do banco de dados e um arquivo para ir realizando testes sem precisar alterar o programa principal.

Banco de Dados:

No MySQL, criei duas tabelas, com o comando "CREATE TABLE", uma para gravar as informações dos usuários e outra para gravar as informações das apostas. Na tabela dos usuário (USERDATA), o nome é registrado como VARCHAR(1000), o cpf é registrado como VARCHAR(11), winner é registrado como TINYINT(1), e o cpf é usado para como chave primária, não sendo possível salvar 2 CPF iguais. Na tabela das apostas (BET), o id é registrado como INT AUTO_INCREMENT, que quando um novo registro é feito um valor é adicionado automaticamente, cpf é registrado da mesma forma, entretanto não é uma chave primária, e os 5 números das apostas são registradas como VARCHAR(2) em n1, n2, n3, n4, n5, e o id é usado para como chave primária da tabela.

No banco de dados também é necessário usar a função “ALTER TABLE BET AUTO_INCREMENT = 1000;”, para que o id comece como 1000, conforme o enunciado. No python o banco de dados é conectado pela função “mysql.connector.connect”, onde é necessário informar o host, user, passwd e database. Para registro nas tabelas é usado os comandos “cursor.execute()” e “db.commit()”, e para buscar dados na tabela é usado os comandos “cursor.execute()” e “cursor.fetchall()”.

1.[Iniciar]:

Primeiro é perguntado se o usuário deseja apagar todos os dados do sorteio e começar uma nova edição, se não, o usuário é direcionado para o menu inicial, se sim, todos os dados das tabelas do banco de dados são apagados com o comando “TRUNCATE TABLE”, e o valor do id é redefinido para 1000.

O código já começa com uma edição do sorteio em andamento, então não é necessário usar a função 1 para começar.

2.[Registrar nova aposta] :

Para registrar uma nova aposta, primeiro é perguntado o nome do apostador e seu CPF. Quando o CPF é registrado é importante cuidar de 2 pontos, o primeiro é se o CPF é válido, e o segundo é se ele já está cadastrado no banco, pois é uma chave primária. Para tratar o primeiro ponto é necessário saber como um CPF é válido, que consiste na seguinte regra: O número do CPF é escrito na forma ABCDEFGHI/JK, sendo J o primeiro dígito verificador e K o segundo, para descobrir J é necessário realizar a operação $10*A + 9*B + 8*C + 7*D + 6*E + 5*F + 4*G + 3*H + 2*I$, e o resultado da soma será dividido por 11, se o resto da divisão for igual a 0 ou 1, então J tem que ser 0, se não J vai ser o resultado de 11 - o resto. Para obter K, é muito parecido, sendo necessário realizar a operação $11*A + 10*B + 9*C + 8*D + 7*E + 6*F + 5*G + 4*H + 3*I + 2*J$, o resultado da soma será dividido por 11, se o resto da divisão for igual a 0 ou 1, então K tem que ser 0, se não K vai ser o resultado de 11 - o resto. Para verificar no programa, desenvolvi meu próprio algoritmo de validação que consiste em seguir a regra, através de uma laço de repetição que vai fazendo a multiplicação do dígitos e a soma, tanto para descobrir J e K, depois com “%” para obter o resto da divisão e por último usando “if” para saber se o número informado pelo usuário corresponde ao encontrado pelas operações do código. Para o segundo ponto com o “try” e “except” é capturado o erro e mostra para o usuário duas opções, se ele deseja informar outro CPF, que repete o processo, ou se deseja usar o mesmo CPF para realizar outra aposta.

Depois, é pedido para informar os 5 números de 1 a 50, um de cada vez, ou 0 para o sistema escolher um número aleatório. O array que armazena os números escolhidos já tem tamanho definido e 5 números negativos atribuídos, para facilitar a encontrar se o número informado for repetido, junto com a função set(), que armazena itens iguais em uma variável só, e então é comparada o tamanho do array dos números escolhidos, com o tamanho do set(). Depois dos 5 números informados, todos os dados registrados são salvos nas tabelas do banco de dados.

Se o sorteio estiver encerrado, novas apostas não são aceitas, somente se a opção 1 for selecionada.

3. [Listas apostas]:

É realizada uma busca no banco de dados pelo ID da aposta, o nome e CPF do apostador, e os 5 números apostados. Para retornar os valores de duas tabelas diferentes juntas foi utilizado o “INNER JOIN” pelo CPF, que conectou as duas tabelas.

Após a busca, se as tabelas não estiverem vazias, é criado um DataFrame para exibir os dados de maneira mais organizada, cada dado em sua coluna, e caso o banco esteja vazio, é exibida uma mensagem informando que não foram realizadas nenhuma aposta. Essa funcionalidade funciona mesmo após o sorteio ser encerrado, mostrando as apostas do sorteio.

4. [Finalizar apostas e executar o sorteio]:

Ao selecionar a opção 4 do menu, é mostrado uma mensagem de confirmação de encerrar as apostas e realizar o sorteio, quando aceito, o sorteio começa. Usando a mesma lógica para salvar os números apostados, é criado um array de tamanho definido e com 30 números negativos diferentes, onde será salvo os números sorteados. Na primeira rodada do sorteio são sorteados os 5 números, sempre conferindo se são diferentes um do outro. Depois na função “draw()”, é feito a primeira busca no banco de dados para saber se houve algum ganhador com os 5 números, se houver é exibido o vencedor e seu cpf, e é feito um “UPDATE” na variável winner da tabela “USERDATA”, que sempre é definida como 0, fica definida como 1 para saber qual apostador ganhou e parte para a apuração, se não houver nenhum ganhador, outro número é sorteado, e testado de novo se houve algum ganhador, se repetindo no máximo 25 vezes, se não houver nenhum ganhador é exibido uma mensagem alertando que o sorteio foi encerrado sem nenhum ganhador e parte para a apuração.

Após qualquer sorteio, com ganhador ou sem, é registrado que o sorteio está encerrado, e para reiniciar outra edição do sorteio é necessário usar a funcionalidade 1. Mesmo com o sorteio encerrado se o usuário tentar começar outra rodada do sorteio, é exibido que o sorteio está finalizado e mostra a apuração.

Se não houver nenhuma aposta registrada, o sorteio também não acontece.

5. [Fim da apuração]:

Na mesma opção 4, após o sorteio ter acabado, a apuração começa automaticamente na função “counting()”, que recebe como parâmetro os números positivos sorteados.

Primeiro é exibido um Data Frame que contém todos os números sorteados em ordem, depois a função “len(numsDraw)-5” é usada para saber quantos sorteios extras foram realizados. Após é pesquisado no banco o nome e CPF, “por winner = 1” e ordenando por ordem alfabética, para mostrar o número de vencedores e mostrar seus dados com um DataFrame, se não tiver vencedor é informado através de uma mensagem, e por último, a parte mais difícil da apuração, é buscado do banco todos os números apostados e registrado em um único array, depois todos os números entre 1 e 51, por um laço de repetição, passam pela função “.count()”, os números que aparecem 0 vezes são removidos, e os que aparecem 1 vez ou mais são salvos numa lista de arrays[,], com a função “.append()”, a quantidade de vezes que eles foram apostados e o próprio número. Com todos os números percorridos, é usado a função “sorted(,reverse = True)”, para organizar eles pelo número que mais aparece até o que menos aparece, e então é exibido o número apostado e a quantidade de vezes em que ele foi apostado.

6. [Premiação]:

Essa, sem dúvida, foi a parte mais difícil, primeiro por ter uma ideia boa o suficiente, e depois para implementar ela. Tive duas ideias antes de encontrar a minha, a primeira foi enviar um PIX através do CPF do vencedor, entretanto, tanto da API do banco Inter quanto na API do Efí Bank, mas eles só aceitam contas PJ para se cadastrarem na API. Minha segunda ideia foi criar um Gift-Card da Binance, mas é preciso ter acesso a verificação KYB, que somente empresas podem ter. Por fim, cheguei a minha última ideia, tomei a liberdade de pedir mais informações ao usuário para poder realizar minha ideia, que foi distribuir ethereum (ETH) para o vencedor, ethereum é a segunda maior criptomoeda do mundo e tem uma excelente proposta por trás dela.

Para isso usei da API do Metamask, que é uma carteira de criptomoedas, carteira essa de onde vai sair o ETH e ir para outra carteira. Então ao existir um vencedor, automaticamente após a apuração, na função prize(), é perguntado para o vencedor qual sua carteira no MetaMask e por fim a transição é realizada. As funções de Web3 fazem as operações da blockchain.

Por se tratar de uma moeda que está muito cara, não posso ficar realizando diversas operações, mas na “except(ValueError)” é capturado o erro que diria que não tem valor suficiente na conta.

Menu:

O menu foi a parte mais simples do código, ele começa mostrando as opções de operações para realizar através de um número indicado pelo usuário, o programa só segue para as operações caso o número informado esteja entre 1 e 4, se não ele repete até o usuário digitar um número certo. No final de qualquer operação, o programa volta, após alguns segundos para o menu. Também, tem um indicador do status do sorteio, para mostrar se ele está em andamento ou se está encerrado.

```
===== SISTEMA DE CONTROLE DE APOSTAS DE ETHEREUM =====  
Status Sorteio: Em andamento
```

Bem vindos!

Escolha umas das opções abaixo:

- [1] Novas edição do sorteio
- [2] Realizar nova aposta
- [3] Exibir a lista de aposta
- [4] Finalizar apostas e executar o sorteio

Opção Escolhida:

Exceções:

Todas as exceções foram tratadas, tanto para entrada de valores numéricos, quanto para a entrada das cidades. Em valores numéricos a entrada de dados se repete até ser digitado um dos números corretor, e caso um valor não inteiro for digitado, o programa vai pedir para informar o dado novamente, usando while para repetir a entrada de dados e try except para capturar um erro, em valores numéricos o if também foi usado.



TESTES (aqui você deverá colar capturas de tela de todas as funcionalidades desenvolvidas e realizar comentários, use o espaço que julgar necessário)

FUNCIONALIDADE 1 – Iniciar

Como esse funcionalidade só tem valor depois do sorteio realizado, vou exibir juntamente com ela o que acontece quando está informado que o sorteio foi realizado no menu inicial:

Status Sorteio: Realizado

Opcao Escolhida: 2

=====

Sorteio ja realizado, nao aceitando novas apostas
Para conferir a apuracao selecione a opcao 4
Para realizar um novo sorteio selecione a opcao 1

Opcao Escolhida: 4

=====

Sorteio ja realizado
Vamos para a apuracao das apostas
Espere um pouquinho enquanto realizamos os calculos
Os numeros sorteados foram:

Numeros sorteados

0	10
1	20

A funcionalidade 3 continua a mesma.

Opção 1:

```

Opcao Escolhida: 1

=====

Tem certeza que deseja apagar todos os dados do sorteio e comecar uma novo?
[1]Nao
[2]Sim

2
===== SISTEMA DE CONTROLE DE APOSTAS DE ETHEREUM =====
Status Sorteio: Em andamento

    Bem vindol

    Escolha umas das opcoes abaixo:
    [1] Novas edicao do sorteio
    [2] Realizar nova aposta
  
```

Antes:

	nameUser	cpf	winner
▶	lucas	85169250053	0
•	NULL	NULL	NULL

	id	cpf	n1	n2	n3	n4	n5
▶	1000	85169250053	30	7	19	17	14
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Depois:

	id	cpf	n1	n2	n3	n4	n5
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	nameUser	cpf	winner
•	NULL	NULL	NULL

FUNCIONALIDADE 2 – Registrar nova aposta

Para o teste eu vou incluir 3 apostas diferentes, a primeira com o usuário escolhendo os 5 números, a 2 com o sistema escolhendo os 5 números, e a terceira com o mesmo CPF da primeira e com alguns números escolhidos pelo sistema outro pelo usuário.

1:

Opcao Escolhida: 2

=====

```
Informe o nome do usuario: Lucas
Informa o CPF (somente o numero): 85169250053
CPF VALIDO
Informe o 1º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 32
O 1º Numero apostado foi: 32

Informe o 2º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 10
O 2º Numero apostado foi: 10

Informe o 3º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 50
O 3º Numero apostado foi: 50

Informe o 4º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 10
Informe um numero diferentes dos escolhidos anteriormente
Informe o 4º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 41
O 4º Numero apostado foi: 41

Informe o 5º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 14
O 5º Numero apostado foi: 14

Aposta realiza com sucesso!
Boa Sorte!
```

2:

Opcao Escolhida: 2

=====

Informe o nome do usuario: Natanael
Informe o CPF (somente o numero): 02195614005
CPF VALIDO
Informe o 1º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 0
O 1º Numero apostado foi: 12

Informe o 2º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 0
O 2º Numero apostado foi: 37

Informe o 3º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 0
O 3º Numero apostado foi: 18

Informe o 4º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 0
O 4º Numero apostado foi: 45

Informe o 5º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 0
O 5º Numero apostado foi: 20

Aposta realiza com sucesso!
Boa Sorte!

3:

Opcao Escolhida: 2

=====

Informe o nome do usuario: Lucas
Informe o CPF (somente o numero): 85169250052
CPF INVALIDO
Informe o CPF (somente o numero): 85169250053
CPF VALIDO
CPF ja cadastrado
[1] Cadastrar novo cpf
[2] Usar o mesmo cadastro para outra aposta
2
Informe o 1º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 12
O 1º Numero apostado foi: 12

Informe o 2º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 0
O 2º Numero apostado foi: 21

Informe o 3º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 36
O 3º Numero apostado foi: 36

Informe o 4º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 0
O 4º Numero apostado foi: 38

Informe o 5º numero da aposta (1 a 50)
Digite 0 para escolher um numero surpresinha(aleatorio): 45
O 5º Numero apostado foi: 45

Aposta realiza com sucesso!
Boa Sorte!

Banco antes:

Result Grid							
	id	cpf	n1	n2	n3	n4	n5
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

nameUser	cpf	winner
NULL	NULL	NULL

Banco depois:

	id	cpf	n1	n2	n3	n4	n5
▶	1000	85169250053	32	10	50	41	14
	1001	2195614005	12	37	18	45	20
	1002	85169250053	12	21	36	38	45
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

nameUser	cpf	winner
Natanael	2195614005	0
Lucas	85169250053	0
NULL	NULL	NULL

FUNCIONALIDADE 3 – Listar apostas

Para mostrar mais dados, vou adicionar mais apostas antes.

	ID	Name	CPF	N1	N2	N3	N4	N5
0	1000	Lucas	85169250053	32	10	50	41	14
1	1001	Natanael	2195614005	12	37	18	45	20
2	1002	Lucas	85169250053	12	21	36	38	45
3	1003	Bella	9846552041	6	27	48	39	46
4	1004	Nate	8991768016	24	28	35	34	30
5	1005	Nicolas	78381696070	50	32	38	26	23
6	1006	Lucas	85169250053	21	44	7	31	2
7	1007	Nicolas	78381696070	10	20	30	40	50
8	1008	Ryan	75586956004	10	20	32	45	15
9	1009	Natanael	2195614005	10	20	30	40	50

Opcao Escolhida: 3

=====

Nenhuma aposta registrada

FUNCIONALIDADE 4 – Finalizar apostas e executar o sorteio

Para a funcionalidade 4, eu vou fazer 2 teste, o primeiro deixar o sistema sortear os números, o segundo, definir os primeiros 5 números sorteados como 10, 20, 30, 40, 50, para mostrar quando tem mais que um ganhador.

1.

```
Desejas nao fazer mais nenhuma aposta e realizar o sorterio?
[1] NAO
[2] SIM      2

APOSTAS ENCERRADAS
VAMOS PARA O SORTEIO

O 1° NUMERO SORTEADO É...
3!!!!!!!!!!!!!!

O 2° NUMERO SORTEADO É...
40!!!!!!!!!!!!!!

O 3° NUMERO SORTEADO É...
43!!!!!!!!!!!!!!
```

```
O 4° NUMERO SORTEADO É...
30!!!!!!!!!!!!!!

O 5° NUMERO SORTEADO É...
23!!!!!!!!!!!!!!
Agora vamos ver se temos um ganhador!

AH NAO!
Parece que nao temos nenhum ganhador ainda
Vamos sortear mais um numero!

O novo numero sortado e...
21!!!
Vamos ver se temos um ganhador

-----
AH NAO!
```

Depois dos 25 números extras sorteados:

```
AH NAO!  
Parece que nao temos nenhum ganhador ainda  
Vamos sortear mais um numero!  
  
O novo numero sortado e...  
49!!!  
Vamos ver se temos um ganhador  
  
-----  
  
NUMERO MAXIMO DE NUMEROS SORTEADOS  
INFELIZMENTE NAO TEMOS NENHUM VENCEDOR DESSE SORTEIO  
BOA SORTE NA PROXIMA  
  
-----  
  
Sorteio ja realizado
```

2.

```
Agora vamos ver se temos um ganhador!  
  
TEMOS UM GANHADOR!!!!!!!!!!!!!!!!!!!!!!  
  
Se apresente Nicolas que tem o cpf 78381696070 para receber seu premio!!!!  
  
-----  
  
TEMOS UM GANHADOR!!!!!!!!!!!!!!!!!!!!!!  
  
Se apresente Natanael que tem o cpf 2195614005 para receber seu premio!!!!  
  
-----
```

3. Nenhuma aposta:

```
Opcao Escolhida: 4  
  
=====:  
  
Nenhuma Aposta registrada
```

FUNCIONALIDADE 5 – Fim da apuração

Para esse funcionalidade, vou usar o resultados dos teste da funcionalidade 4, então primeiro vai ser a apuração sem nenhum vencedor e depois com dois vencedores

1.

Sorteio já realizado
Vamos para a apuração das apostas
Espere um pouquinho enquanto realizamos os cálculos
Os números sorteados foram:

Números sorteados	
0	3
1	40
2	43
3	30
4	23
5	21
6	41
7	5
24	25
25	34
26	39
27	14
28	24
29	49

A quantidade de sorteios extras realizados: 25

Quantidade de apostas vencedoras: 0

Os vencedores são:
Não houve vencedores

Nro apostados	Qtd de apostas
50	4
20	4
10	4
45	3
32	3
30	3
40	2
38	2
21	2
12	2
48	1
46	1
44	1
41	1
39	1
37	1
36	1
35	1
34	1
31	1
28	1
27	1
26	1
24	1
23	1
18	1
15	1
14	1
7	1
6	1
2	1

2.

Sorteio já realizado
Vamos para a apuração das apostas
Espere um pouquinho enquanto realizamos os cálculos
Os números sorteados foram:

Números sorteados	
0	10
1	20
2	30
3	40
4	50

A quantidade de sorteios extras realizados: 0

A quantidade de sorteios extras realizados: 0

Quantidade de apostas vencedoras: 2

Os vencedores são:

	Vencedores	CPF
0	Natanael	2195614005
1	Nicolas	78381696070

Nro apostados	Qtd de apostas
50	4
20	4
10	4
45	3
32	3
30	3
40	2
38	2
21	2
12	2
48	1
46	1
44	1
41	1
39	1
37	1
36	1
35	1
34	1
31	1
28	1
27	1
26	1
24	1
23	1
18	1
15	1
14	1
7	1
6	1
2	1

FUNCIONALIDADE 6 – Premiação

Para essa funcionalidade, vou realizar 3 teste, os dois primeiros sendo pra quando tiver nenhum ganhador e quando tiver 2 ganhadores, entretanto eu só tenho 2 conta (uma para enviar e outra para receber) e enviar duas vezes não ficaria muito fácil de visualizar, então vai mostrar uma mensagem informando um que não foi possível realizar transação, mas seria possível se eu tivesse a quantidade disponível. O terceiro teste, vai ser com um ganhador, onde vai ser possível transferir então o ETH, mostrando nas duas carteiras

1.

Agora vamos para a distribuição do premio:

Infelizmente nao tivemos vencedores

===== SISTEMA DE CONTROLE DE APOSTAS DE ETHEREUM =====

Status Sorteio: Realizado

2.

Agora vamos para a distribuição do premio:

Natanael de CPF: 2195614005 informe o endereço da sua carteira do MetaMask (exemplo de formato: 0xe34FC27644eaE4271435F66784F11Cf31e797E1a): 0xe34FC27644eaE4271435F66784F11Cf31e797E1a

Infelizmente o preco do eth esta muito caro para ficar fazendo tantas transações

Nicolas de CPF: 78381696070 informe o endereço da sua carteira do MetaMask (exemplo de formato: 0xe34FC27644eaE4271435F66784F11Cf31e797E1a): 0xe34FC27644eaE4271435F66784F11Cf31e797E1a

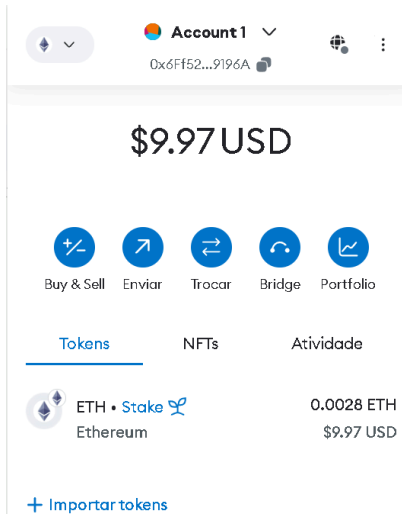
Infelizmente o preco do eth esta muito caro para ficar fazendo tantas transações

===== SISTEMA DE CONTROLE DE APOSTAS DE ETHEREUM =====

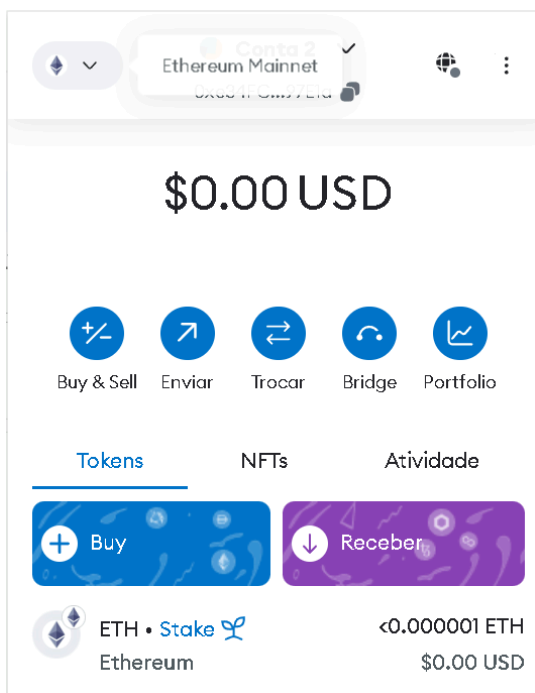
Status Sorteio: Realizado

3.

Conta na Meta Mask que envia antes:



Conta da Meta Mask de quem recebe antes:



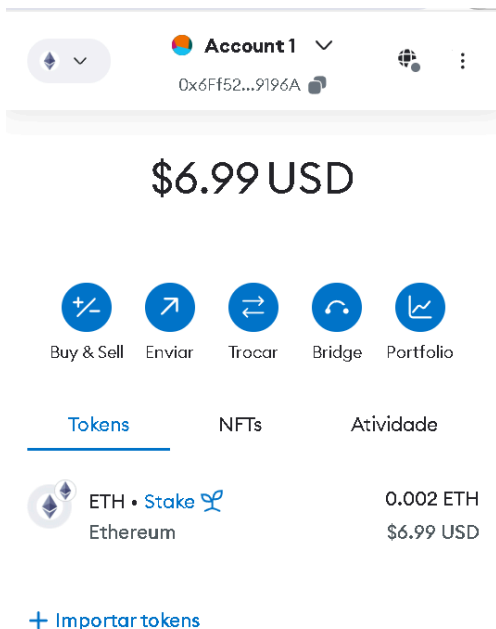
Premiação Enviada:

Agora vamos para a distribuição do prêmio:
Natanael de CPF: 2195614005 informe o endereço da sua carteira do MetaMask (exemplo de formato: 0xe34FC27644eaE4271435F66784F11Cf31e797E1a): 0xe34FC27644eaE4271435F66784F11Cf31e797E1a

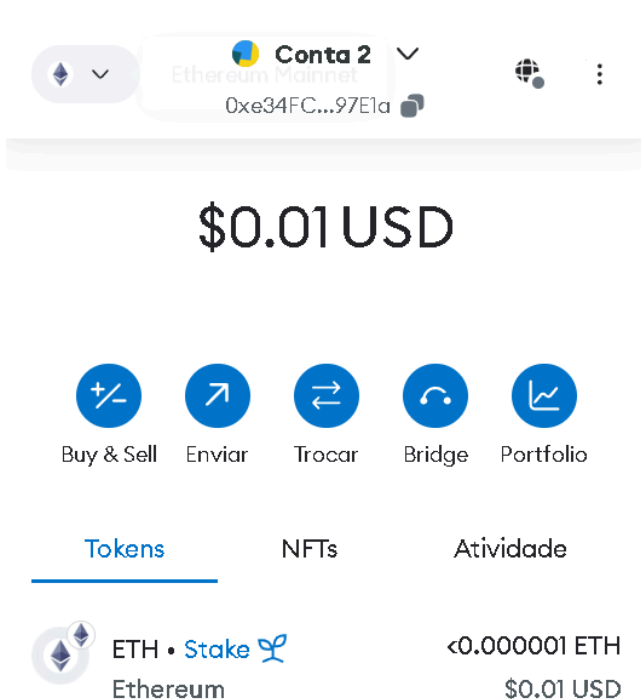
PREMIACAO ENVIADA!!!!

===== SISTEMA DE CONTROLE DE APOSTAS DE ETHEREUM =====
Status Sorteio: Realizado

Conta na Meta Mask que envia antes:



Conta na Meta Mask que envia depois:



O valor não mudou, pois a quantia foi muito pequena, da outra conta reduziu mais devido ao valor de gas mínimo, valor de gas é o custo que é cobrado para realizar as transações, por isso não é possível realizar tanta transações mesmo enviando pouco dinheiro.

Comprovante de transação:

Send



Mar 20, 2024 at 08:33 pm


From (You)

 0x6ff5...59196a (0x6ff...196a) 

To

 0xe34f...797e1a (0xe34...7e1a) 

Transaction Hash

0xeeb4...c586f8 

Transaction Fee 

0.001 ETH

Sent



Transaction Details

Blockchain

 Ethereum

Status

 Success
Ativar o

Timestamp

7 minutes ago (Mar 20, 2024 at 08:33 pm) Co

AUTOAVALIAÇÃO

Você concluiu a implementação de 100% das funcionalidades solicitadas?

(X) Sim () Não

Para as 6 principais funcionalidades solicitadas, como você avalia a sua solução?

Marque um 'X'.

	Inexistente/ Insuficiente	Pouco satisfeito(a)	Satisfeito(a)	Muito satisfeito(a)
Funcionalidade 1				X
Funcionalidade 2				X
Funcionalidade 3				X
Funcionalidade 4				X
Funcionalidade 5				X
Funcionalidade 6				X

Principais dificuldades

Minhas principais dificuldades foram: a primeira foi deixar o código mais fácil de entender, acredito que o uso de notebook dificulte isso, por não conseguir separar o código em diversos arquivos, mas analisando bem o código com paciência é possível entender.

Minha segunda dificuldade foi com a API, foi o que mais levou tempo para ser implementado, devido a falta de recursos não pude realizar diversos testes, e tinha que ir realizando muitas partes sem testar.

Desempenho Geral

Acredito que meu desempenho tenha sido muito bom, consegui realizar tudo que foi solicitado, e acredito ter tido uma boa ideia de premiação. Foi a primeira vez que eu utilizei banco de dados e API com o python, só havia usado em outras linguagens de programação. As técnicas utilizadas não fugiram muito do que eu aprendi na faculdade, só tive que juntar os ensinamentos de POO e de Banco de Dados para conseguir realizar a atividade. Poderia ter usado outra linguagem de programação como Java que poderia

me ajudar mais na parte de objetos, mas optei pelo Python pela sua facilidade e velocidade, deixando o código menor e mais fácil de se entender.

Obrigado por participar deste processo seletivo.
Salve o documento em PDF com o seu nome.