



[Cursos](#) > [Introdução a Blockchain \(Ethereum\)](#) > [Aula 06 - Prova](#)

## Introdução a Blockchain (Ethereum)

Iniciou em **22/06/2022 às 10:16**

# Aula 06 - Prova

A seguir, dois modelos de provas, porém, você vai escolher apenas um de sua preferência.

### Prova - Frontend:

Criar um layout de um jogo de azar, chamado LubyGame. O jogo é composto por várias perguntas e respostas onde cada jogador poderá startar e se arriscar no mundo de Play to Earn.

O Layout fica à escolha de cada um, pois o foco é a integração com o contrato inteligente Ethereum (**Mas caprichem**).

O jogo deve ser composto por pelo menos 3 questões, e cada questão tem 4 opções e apenas 1 correta.

Para simular um game, devemos iniciar "doando" algum valor em LubyCoin para o player poder jogar (em ambiente de produção, eles comprariam essas moedas por um determinado valor).

Com a cryptomoeda na carteira, o jogo deve ser iniciado através da função StartGame do contrato inteligente. Após isso, cada questão certa ou errada, serão chamadas as respectivas funções do contrato.

O player pode sacar todo o dinheiro do jogo utilizando a função ClaimBalance.

### Documentação:

A partir de agora, você vai integrar o contrato inteligente LubyCoin, com uma carteira MetaMask localmente usando o Ganache e React.

No contrato existem os seguintes métodos:

- **MintLBC(value):** Doar "value" de LBCs para um player
- **StartGame(value):** Iniciar o jogo e pagando o "value" para a banca
- **CorrectAnswer(value):** Resposta correta e o "value" vai para a carteira do player no jogo
- **IncorrectAnswer(value):** Resposta errada e o "value" é retirado da carteira do player no jogo
- **Withdraw:** Sacar todo o dinheiro do jogo para a carteira do dono do jogo (apenas admins)
- **ClaimBalance(bonus):** O player retira o saldo do jogo + o "bônus".
- **getBalanceIndividual:** Retorna o valor de cada player **no jogo**

**Dica:** Para realizar a integração do contrato com o React, vocês podem utilizar a Lib Web3JS encontrado no NPM.

`npm i web3`

**Atenção:** É necessário baixar o código abaixo e realizar deploy na máquina de vocês utilizando a rede Ganache, logo após isso, o será atribuído como owner do contrato a conta do Ganache selecionado por vocês. Após isso, é gerado um JSON do contrato inteligente e será esse o responsável por fornecer as funções para o seu FRONTEND.

**Link do Repo:** [https://github.com/aristotelesFerreira/Blockchain\\_Crypto\\_LubyGame](https://github.com/aristotelesFerreira/Blockchain_Crypto_LubyGame)

### Prova - Backend:

Criar um novo contrato em solidity para permitir a criação de uma nova moeda digital, sendo possível, transferir ou doar moedas entre outras carteiras MetaMask. O modelo utilizado para a criação da criptomoeda deve seguir o padrão ERC 20.

- **Deve** ser possível transferir moedas entre address
- **Deve** ser possível estipular o limite máximo de moedas a serem criadas
- **Toda** transação, terá uma **taxa de 10%** que ficará para o contrato (Não poderá cobrar essas taxas para clientes vips)
- **Apenas** o **OWNER** do contrato pode sacar todo o valor acumulado de taxas
- **Clientes vips** são alguns address que informaremos a qualquer momento para o contrato, eles serão isentos de taxas
- **Deve** ser **possível modificar** a porcentagem da taxa cobrada a qualquer momento
- A função de **DOAR** só pode ser efetuada 1x no mês e com valor máximo de 1 ether
- **Deve** ser possível pausar todas as transferencias
- O contrato inteligente deve registrar todas transações

Atenção: As dúvidas desse módulo devem ser direcionadas para o Aristóteles Ferreira.

### Sobre a Luby

Desde 2002, criamos softwares para facilitar o dia a dia de empresas de diversos segmentos. Por meio do desenvolvimento de profissionais qualificados, nos tornamos referência em desenvolvimento de soluções digitais.

### Localização e Contatos

Rua Galeno de Almeida, 188 - Pinheiros, São Paulo - SP

+55 (11) 3055-3404

+55 (11) 93805-8897

### Redes Sociais

[Site oficial](#)  
[Blog](#)



Finalizar