



Guia para cursos de Blockchain para desenvolvedores Ethereum

Ambiente para aplicação web

Temos duas opções para rodar nosso projeto web.

1. A primeira opção é utilizar o gerenciador de pacotes NPM, Bower e Gulp para subir nossa aplicação.

Instalar npm

<https://nodejs.org/en/download/>

Para testar a instalação execute **npm version**

```
[~] $npm version  
{ npm: '5.5.1',
```

Instalar o nvm para gerenciar as versões do node

npm install nvm

nvm use 8.9.1

Instalar Bower

npm install -g bower

Instalar Gulp

npm install -g gulp

2. A segunda opção é a gente acessar diretamente os arquivos index.html.

Instalar Metamask

<https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn>

[Vídeo no nosso canal do youtube](#)

Download do projeto:

Acesse o GitHub para baixar o projeto.

<https://github.com/rodgasp/blockmasters/>



Ambiente Ethereum

GETH

<https://geth.ethereum.org/downloads/>

Alternativas

Linux

<https://gethstore.blob.core.windows.net/builds/geth-linux-amd64-1.8.14-316fc7ec.tar.gz>

Mac

<https://gethstore.blob.core.windows.net/builds/geth-darwin-amd64-1.8.14-316fc7ec.tar.gz>

Windows

<https://gethstore.blob.core.windows.net/builds/geth-windows-amd64-1.8.14-316fc7ec.exe>

Configurar:

MAC / Linux

vi ~/.bash_profile

dentro do vi colocar as linhas a seguir

apertar a tecla i para edição

#GETH ETHEREUM

export GETH_HOME=**CAMINHO DE DOWNLOAD DO GETH**

export PATH=\$PATH:\$GETH_HOME

Cliquem em esc + :wq para salvar

Agora temos que atualizar o .bash_profile

source ~/.bash_profile

Windows

Adicionar nas variáveis de ambiente o executável.

** Feche o console e abra novamente

Agora vamos ver se a versão que instalamos é que está configurada.

No terminal executar **geth version**

```
~ $geth version
Geth
Version: 1.8.13-stable
Git Commit: 225171a4bfcc16bd12a1906b1e0d43d0b18c353b
Architecture: amd64
Protocol Versions: [63 62]
Network Id: 1
```



Conheça o help - [Documentação](#)

geth help

```
~ $geth help
NAME:
  geth - the go-ethereum command line interface

  Copyright 2013-2018 The go-ethereum Authors

USAGE:
  geth [options] command [command options] [arguments...]

VERSION:
  1.8.13-stable-225171a4

COMMANDS:
  account      Manage accounts
  attach       Start an interactive JavaScript environment (connect to node)
  bug          opens a window to report a bug on the geth repo
  console      Start an interactive JavaScript environment
  copydb       Create a local chain from a target chaindata folder
  dump         Dump a specific block from storage
```

Para subir o nó apontando para testnet:

```
geth --datadir test_net --testnet
```

```
[temp $geth --datadir test_net --testnet
INFO [08-05|14:02:21.667] Maximum peer count
INFO [08-05|14:02:21.681] Starting peer-to-peer node
INFO [08-05|14:02:21.681] Allocated cache and file handles
ata cache=768 handles=128
```

Para subir o nó apontando para prod:

```
geth --datadir prod_net
```

Para subir o nó em desenvolvimento:

```
geth --datadir aula_web --dev --rpc --rpcapi
"admin,db,eth,debug,miner,net,shh,txpool,personal,web3" console
```



Ao executar com console vai abrir um cliente em javascript para a gente criar uma conta.

```
/geth.ipc  
Welcome to the Geth JavaScript console!
```

Listar a conta existente

```
personal.listAccounts
```

```
[> personal.listAccounts  
["0xb3757d047ce7a1f152db436ff6d848b3ace2a2ec",
```

Essa conta é criada pois estamos em modo de desenvolvimento e essa versão do Geth já cria uma conta com saldo para que nosso nó possa validar as transações.

Agora vamos verificar o saldo:

```
eth.getBalance(personal.listAccounts[0])
```

```
[> eth.getBalance(personal.listAccounts[0])  
1.15792089237316195423570985008687907853269984665640564039457584007913129639927e+77
```

Vamos criar uma nova conta

```
personal.newAccount("frase de segurança")
```

```
[> personal.newAccount("1")  
"0x40c611cc0b863f4215a988b1196a176e700d9b52"
```

1. Qual é o saldo dessa nova conta?

Saia do cliente javascript com "exit"

Suba novamente o nó agora sem o **console**



```
geth --datadir aula_web --dev --rpc --rpcapi  
"admin,db,eth,debug,miner,net,shh,txpool,personal,web3" --rpcaddr "localhost" --rpcport  
"8545" --rpccorsdomain ""
```

--datadir - informa o local onde vai criar seu nó, é nessa pasta que vão ser criados suas contas, baixar os registros dos blocos e logs.

--dev - informa que é para subir o geth em modo de desenvolvimento

--rpcport - informa qual porta queremos utilizar para chamadas via RPC

--rpcaddr - Endereço RPC

--rpccorsdomain "" - libera chamadas de outros domínios

--rpc possibilita que o nó libera o protocolo de [RPC](#) para comunicação

```
HTTP endpoint opened          url=http://127.0.0.1:8545
```

--rpcapi libera as funções que podem ser executadas no nó, mais detalhes sobre as APIs [clique aqui](#)

console - possibilita que sejam executados comandos no nó

Mais detalhes sobre os parâmetros [clique aqui](#).

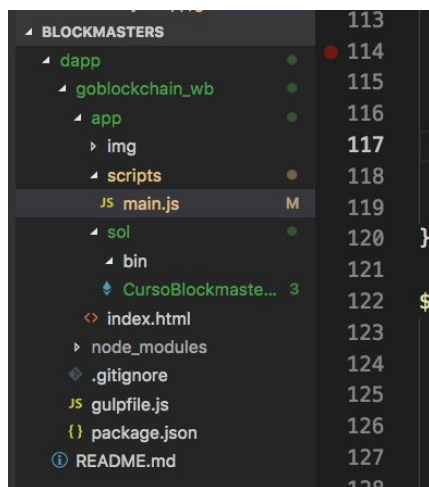
Wallet

Aqui iniciamos nosso primeiro passo para aprender a criar aplicações descentralizadas.

Existem wallets como [Mist](#), [Harmony](#), [Fether](#) e [Ethereum Wallet](#).

Instalação do projeto:

Eu estou usando o [Visual Code](#) e adicionei a pasta **blockmasters** dentro do meu editor.





Após isso vamos instalar as dependências:

1. Via terminal vá até o diretório blockmasters/dapp/goblockchain_wb

```
ethereum_workbench $ cd goblockchain_wb/
```

2. Execute o comando **npm install**

```
goblockchain_wb $ npm install
( [REDACTED] ) : loadDep:os-tmpdir: sill install loadAllDepsIntoIdealTree
```

3. Execute o comando **bower update**

```
goblockchain_wb $ bower update
bower bootstrap-sass#~3.3.5    cached https://github.com/twbs/bootstrap-sass.git#3.3.7
bower bootstrap-sass#~3.3.5    validate 3.3.7 against https://github.com/twbs/bootstrap-sass.git#~3.3.5
bower modernizr#~2.8.1         cached https://github.com/Modernizr/Modernizr.git#2.8.3
bower modernizr#~2.8.1         validate 2.8.3 against https://github.com/Modernizr/Modernizr.git#~2.8.1
bower web3#^0.18.2             cached https://github.com/ethereum/ethereum.js.git#0.18.4
```

4. Vamos subir o projeto

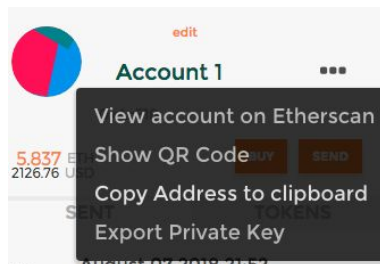
```
goblockchain_wb $ gulp serve
[14:19:43] Using gulpfile ~/Documents/henrique/GoBlockchain/ethereum_workbench/goblockchain_wb/gulpfile.js
[14:19:43] Starting 'serve'...
[14:19:43] Starting 'clean'...
```

Nosso projeto vai subir na porta 3000

<http://localhost:3000/>

Como conseguir ether para testnet Rinkeby

Copie sua chave pública no metamask





Faça uma publicação no Twitter ou Google + com a sua chave pública



Acesse <https://faucet.rinkeby.io/> e adicione o link da publicação e selecione a quantidade de Ether.

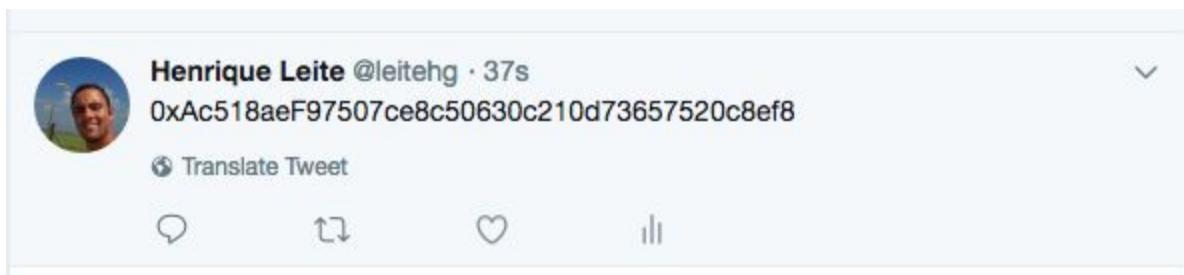
Rinkeby Authenticated Faucet

<https://plus.google.com/107987283654537876593/posts/X6MomoSPHDc>

Give me Ether ▼

 7 peers  2778046 blocks  9.046256971665328e+56 Ethers  174601 funded

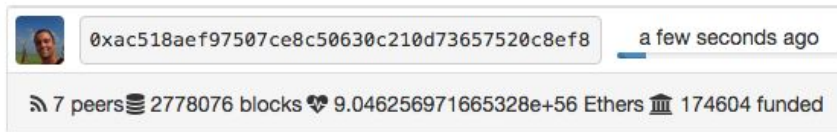
Ou podem utilizar o Twitter





<https://twitter.com/leitehg/status/1027211292883476481>

Give me Ether ▼



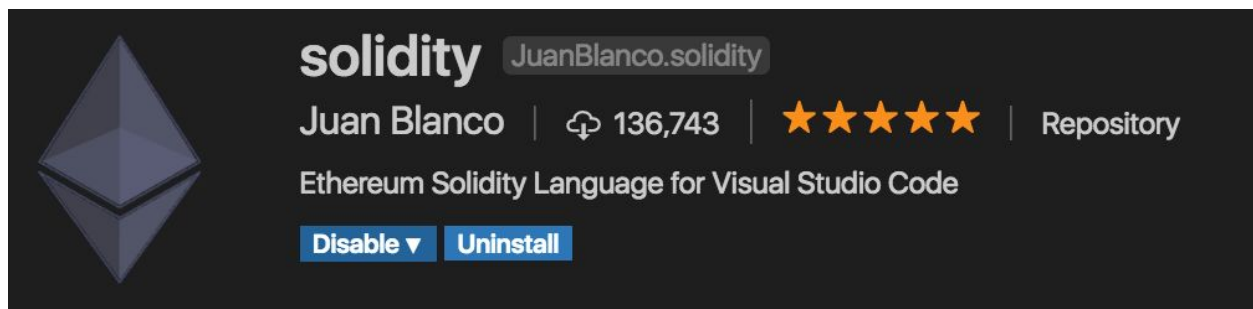
Ambiente de programação

Para programar

Visual Code

<https://code.visualstudio.com/download>

Plugin:



IDE Online:

[Remix.ethereum.org](https://remix.ethereum.org)

<https://ethfiddle.com/>

Metamask:

Gerenciador de carteiras Ethereum.

Pode adicionar ele ao navegador Chrome, Firefox e Brave.

<https://metamask.io/>



Vídeo passo a passo da instalação.

<https://youtu.be/iPlymNvgA2g>

Ganache:

Simula ambiente Ethereum para desenvolvimento

<http://truffleframework.com/ganache/>

Compilador solcjs:

sudo npm install solc -g

solcjs help

Util

Timestamp

<http://www.timestampgenerator.com/1522797840/#result>

geth erro porta em uso

sudo lsof -i :30303

kill -9 <PID>

geth attach ipc: ENDERECO_IPC

windows -

geth attach ipc:"\\.\pipe\geth.ipc"

-----truffle-----

Instalar o node

<https://nodejs.org/en/download/>

truffle

npm install truffle -g

Download ganache

<http://truffleframework.com/ganache/>

1- criar projeto com truffle com nome do jogo



- mkdir {meujogo}
- cd meujogo
- truffle init
- abrir projeto no visual code
- criar contrato dentro da pasta de contracts com o nome MeuJogo.sol
- criar estrutura básica de contrato (pragma, contract e constructor)

2- configurar truffle.js

```
module.exports = {  
  networks: {  
    ganache: {  
      host: "127.0.0.1",  
      port: 7545,  
      network_id: "*",  
      gas: 4600000  
    },  
    development: {  
      host: "127.0.0.1",  
      port: 8545,  
      network_id: "*",  
      gas: 4600000  
    }  
  }  
};
```

3- Escrever migration

4- Verificar contratos foram compilados

- truffle compile

5 - Realizar deploy

- truffle migrate --network ganache

<https://etherscan.io>

<https://github.com/OpenZeppelin/zeppelin-solidity/tree/master/contracts>

desbloquear usuario através de geth attach

- personal.unlockAccount(personal.listAccounts[0], "", 200000)



```
--instalar solcjs  
sudo npm install solc -g  
solcjs contracts/Contrato.sol --abi  
solcjs app/sol/Person.sol --abi -o app/sol/bin --bin
```

```
--converter para timestamp  
http://www.timestampgenerator.com/date-from-timestamp/1522697837/
```

```
-- frações de unidades do ether  
http://www.moeda.cc/entendendo-as-fracoes-do-ether/
```

Filmes e materiais:

Ethereum

[White paper](#)

[Vídeo sobre Ethereum](#)

[Qual tamanho máximo de uma transação](#)

[Como baixar o bloco rápido](#)

Gerais

<https://blockchain.info/>

Vídeo P2P Foundation - The Blockchain: Change Everything Forever
<https://blog.p2pfoundation.net/the-blockchain-change-everything-forever/2016/10/05>

Vídeo - Harvard in Tech - Bitcoin & the Blockchain Primer
<https://www.youtube.com/watch?v=5les7wTmIPE>

Online Program - BlockChain 101
<https://onemonth.com/courses/blockchain/preview>

Série de vídeos Tech Crunch - Trust Disrupted: Bitcoin and the Blockchain



<https://techcrunch.com/video/truth-disrupted-bitcoin-and-the-blockchain-s1e1-in-the-beginning/57f581cb5095497719f427b2/>

Texto - A Beginners Guide to Blockchain Essentials (for Dummies)

<https://www.linkedin.com/pulse/beginners-guide-blockchain-essentials-dummies-collin-thompson>

Texto - O Enigma das Bitcoins

<https://capitalaberto.com.br/extra/bitcoins/>

Vídeo - How Bitcoin Works

<https://www.youtube.com/watch?v=l9jOJk30eQs&feature=youtu.be>

Vídeo - From Competition to Cooperation | Primavera De Filippi | TEDxCambridge

<https://www.youtube.com/watch?v=aYOPcHRO3tc>

Vídeo - Bitcoin News - Bitcoin Explained In Details

<https://www.youtube.com/watch?v=Cd6i3JrHBng>

Texto Tech Crunch - A decentralized web would give power back to the people online

<https://techcrunch.com/2016/10/09/a-decentralized-web-would-give-power-back-to-the-people-online/>

Why you should care about blockchains non-financial uses of blockchain technology

<http://www.nesta.org.uk/blog/why-you-should-care-about-blockchains-non-financial-uses-blockchain-technology>