

Pet Shop Tutorial (roteiro de vídeo)

1. Sobre	2
2. Verificando a autoria deste documento	3
3. Introdução	4
4. Executando um Pet Shop descentralizado	5
4.1. Passo a passo para execução	5
5. Principais diferenças em um Pet Shop descentralizado	6
5.1. Número de computadores (ou nós)	6
5.2. Corrupção de dados	6
5.3. Anonimato	7
6. Desenvolvimento	8
6.1. Passo a passo para construção.	8

1. Sobre

Escrito por Paulo Jerônimo (<https://keybase.io/paulojeronimo>).

Última atualização em 2022-04-17 11:17:13 -0300.

Este é o roteiro para a gravação do vídeo **Pet Shop Tutorial** que será composto de vídeos gravados com um dos procedimentos a seguir:

- **Procedimento 1:**

- Gravação pelo celular.
- Fala seguindo o roteiro escrito.
- Uso do [Nano Teleprompter](#).
- Sem abertura de links.
- Imagens dos links inseridas na edição do vídeo.

- **Procedimento 2:**

- Gravação pelo computador.
- Fala de forma livre.
- Uso do [OBS](#) para *Display Capture*.
- Reflexo da *minha imagem no display* pela *webcam* do computador.
- Com abertura de links.

O vídeo final desse roteiro está sendo produzido para os mentorados (do grupo [ftecm221](#)) da mentoria [Desenvolvimento de aplicações Web 3.0 para a Blockchain do Ethereum](#).

Esse vídeo será disponibilizado de forma pública assim que for finalizado.

2. Verificando a autoria deste documento

Isso pode ser feito através do PGP (ou GPG):

Importação da chave PGP do autor (Paulo Jerônimo):

```
$ curl https://keybase.io/paulojeronimo/pgp_keys.asc | gpg --import
```

Download do arquivo original original e de sua assinatura:

```
$ curl -O https://finisher.tech/ethereum-pet-shop-tutorial.roteiro-video.pdf  
$ curl -O https://finisher.tech/ethereum-pet-shop-tutorial.roteiro-video.pdf.sig
```

Verificação do arquivo:

```
$ gpg --verify ethereum-pet-shop-tutorial.roteiro-video.pdf.sig ethereum-pet-shop-tutorial.roteiro-video.pdf  
gpg: Signature made Sun Apr 17 11:03:27 2022 -03  
gpg: using RSA key 22E89A835E4BFF88EA17419310BF59F678EC5B47  
gpg: Good signature from "Paulo Jerônimo Neves Ribeiro <paulojeronimo@gmail.com>"  
[ultimate]
```

3. Introdução

Vídeo gravado com o [Procedimento 1](#).

Uma das palavras chaves que motivam a migração de sistemas descentralizados para centralizados é **CONFIANÇA**. Neste vídeo aprenderemos como uma aplicação já migrada de um modelo centralizado para o descentralizado possui vantagens no aspecto da confiança. Também entenderemos outras diferenças entre esses modelos.

A aplicação descentralizada (**DApp**) que veremos é bastante conhecida em algumas comunidades de desenvolvedores de software, principalmente de programadores Java, desde o tempo em que a Sun Microsystems (adquirida pela Oracle), criou uma [versão de um Pet Shop utilizando tecnologias enterprise](#) em um modelo de desenvolvimento centralizado.

Mesmo ainda trabalhando no modelo tradicional (centralizado), eu venho me distanciando cada vez mais dele pela convicção de que, através da construção de soluções descentralizadas, eu ajudarei a construir um mundo mais confiável.

Por conta desse meu distanciamento dos sistemas centralizados, esse vídeo que você está vendo está vendo aqui no YouTube é apenas um corte do vídeo completo que está disponibilizado em [meu canal no Odysee](#).

Apesar de eu ter visto [uma crítica dizendo que o Odysee não é uma plataforma descentralizada](#), eu ainda não investiguei isso a fundo. Então, eu ainda espero utilizá-lo para publicar vídeos de forma descentralizada. O Odysee é executado em uma Blockchain, assim como esse Pet Shop que apresentarei.

Por causa dessa descentralização o Odysee não pode ser censurado, por governos ou por ninguém, da mesma forma como o YouTube ou o Twitter. E falando desse último, [o Elon Musk, que tem milhares de bitcoins, até fez proposta de comprar o Twitter](#). Isso preocupou os cerceadores de liberdade de expressão.

[O código dos projetos do Odysee está no GitHub](#), assim como o do Pet Store que apresentarei. DApps deveriam ser, por padrão, *open source*. Em próximos vídeos falarei sobre a importância disso.

Por favor, caso goste do conteúdo desse vídeo, crie uma conta no Odysee, inscreva-se em meu canal e dê um *like* nesse meu vídeo. Fazendo isso, **nós dois ganharemos a criptomoeda LBRY**. Leia um pouco sobre ela [em seu site](#) e no [CoinMarketCap](#).

4. Executando um Pet Shop descentralizado

Vídeo gravado com o [Procedimento 1](#).

A construção de um Pet Shop descentralizado é descrita em um [tutorial do Truffle](#) que eu simplifiquei em um [tutorial mais fluido](#), de forma que você consiga, utilizando o Visual Studio Code, ou até mesmo o Vim, chegar a um [repositório da aplicação final](#) criada por esse tutorial.

O Truffle é um framework que simplifica a construção de DApps. Ele está sendo utilizado, assim como diversas outras ferramentas, na minha mentoria sobre o [Desenvolvimento de aplicações Web 3.0 para a Blockchain do Ethereum](#).

Se você fizer uma pesquisa por "Truffle Pet Shop Tutorial" no YouTube, também encontrará vários vídeos a respeito dessa DApp que podem complementar esse meu vídeo de alguma forma.

Então, partindo agora em direção ao nosso ponto principal, vamos baixar o Pet Shop do GitHub e colocá-lo em execução.

4.1. Passo a passo para execução

Vídeo gravado com o [Procedimento 2](#).

Demonstração, passo a passo, de execução do projeto <https://github.com/paulojeronimo/ethereum-pet-shop-ftecm221>.

5. Principais diferenças em um Pet Shop descentralizado

5.1. Número de computadores (ou nós)

Vídeo gravado com o [Procedimento 1](#).

Observando esse DApp, a primeira coisa que precisamos saber que o diferencia de um aplicativo centralizado é que, nesse modelo, geralmente temos no máximo alguns poucos nós executando a aplicação Pet Shop.

Com certeza, o Pet Shop pequeno desse exemplo, que possui espaço para a adoção de apenas 16 *pets*, deverá ter apenas um único nó no modelo centralizado. Mas, em um mundo descentralizado, se estivermos rodando esse Pet Shop na rede de produção de Ethereum, ele estará em execução em milhares de nós.

Você pode ter uma ideia de quantos nós Ethereum estão ativos, por país, acessando o [Etherscan](#) ou o [ethernodes.org](#). Obviamente, você também pode [colocar um nó próprio em atividade](#) e participar de qualquer uma de suas redes (de teste ou de produção).

Vamos imaginar que, por algum motivo, faltou energia para alimentar o nó que executa a aplicação do Pet Shop no modelo centralizado. Nessa situação, o que acontece é bem óbvio: o Pet Shop simplesmente deixa de oferecer sua funcionalidade de adoção de *pets*.

No modelo descentralizado do Pet Shop você é simplesmente redirecionado para a aplicação web servida por outro nó, das [centenas de nós ativos na Blockchain do Ethereum](#). Tenha em mente que, de forma a parar uma aplicação em execução nessa rede seria necessário pararmos todos esses nós. Isso, na prática, talvez seja algo que poderia ocorrer no caso de uma terceira guerra mundial.

5.2. Corrupção de dados

Vídeo gravado com o [Procedimento 1](#).

Por ser executado em uma Blockchain, uma vez que um *pet* foi adotado por alguém (que tenha um endereço válido na rede Ethereum) é impossível que os dados de adoção sejam alterados. A não ser, é claro, que o *smart contract* que executa o app ofereça a possibilidade de alteração de titularidade. Mas, de qualquer forma, isso deverá ser realizado através de uma outra transação que alterará, na Blockchain, a informação que identifica o dono do *pet*.

Em um modelo tradicional que utilize um banco de dados, um *pet* poderia ser alterado por alguém com acesso a esse banco de forma que até mesmo o registro de transações anteriores, relativas a quem pertence o *pet*, poderia ser alterado.

5.3. Anonimato

Vídeo gravado com o [Procedimento 1](#).

Para o caso de um Pet Shop, talvez o anonimato que é decorrente do fato de um *pet* poder ser comprado não por alguém conhecido mas, sim, por alguém que tenha um endereço válido na rede do Ethereum, não seja uma vantagem. Contudo, para várias aplicações, alguma forma de anonimato é algo desejável e está sempre presente em aplicações baseadas em Blockchain.

6. Desenvolvimento

Vídeo gravado com o [Procedimento 1](#).

Se você é desenvolvedor e quer saber, passo a passo, como esse DApp é construído do zero até a execução local em sua máquina, fique comigo até o final desse vídeo que você também aprenderá sobre isso a partir de agora.

6.1. Passo a passo para construção

Vídeo gravado com o [Procedimento 2](#).

Demonstração, passo a passo, da construção do projeto <https://github.com/paulojeronimo/ethereum-pet-shop-ftecm221> seguindo o tutorial <https://paulojeronimo.com/ethereum-pet-shop-tutorial>.