Blockchain, Criptomoedas & Tecnologias Descentralizadas

Tecnologias descentralizadas:

Interplanetary File System (IPFS)

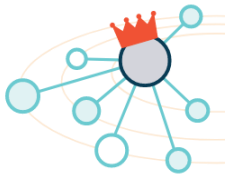
Prof. Dr. Marcos A. Simplicio Jr. – mjunior@larc.usp.br
Escola Politécnica, Universidade de São Paulo

Objetivos

- Conhecer o Sistema de Arquivos Interplanetário (InterPlanetary File System -- IPFS)
  - Uma proposta de web altamente descentralizada
  - Um ótimo exemplo de engenharia: combina diversas tecnologias descentralizadas discutidas no curso!

Problemas com o HTTP

- **Centralizada em servidores**, que podem ser desligados
  - **Conteúdo** acaba sendo **perdido**, proposital ou acidentalmente
- Cria **dependência** de alguns serviços essenciais



  - **Buscas**: Google pode controlar o que usuários encontram
  - **Hosting**: AWS/Azure/Locaweb pode controlar o que usuários armazenam ou veem
  - Resultado: **censura**, invasão de **privacidade**, **espionagem**, manipulação de **opiniões**, possível **interrupção** de serviços…
- **Ineficiente**: servidor de conteúdo popular torna-se gargalo
  - Escolha entre lentidão ou contratação serviços de caching…

- Voltado a **locais**, não a **conteúdos**:

  - Links quebram se local for alterado (404 Not Found)

IPFS: InterPlanetary File System

- Criado em 2015, por Juan Benet (Protocol Labs)
  - Um sistema de arquivos versionado, distribuído globalmente
  - "**Uma web permanente e distribuída**": similar a um enorme swarm bittorrent para troca de objetos
    - Páginas web, imagens, vídeos, código, ...
  - Permite a hospedagem de sites e dados sem um servidor correspondente: **"servidor" distribuído na rede**!

Atualmente, parte de ecossistema maior:

IPFS: arquitetura

- Combina diferentes tecnologias

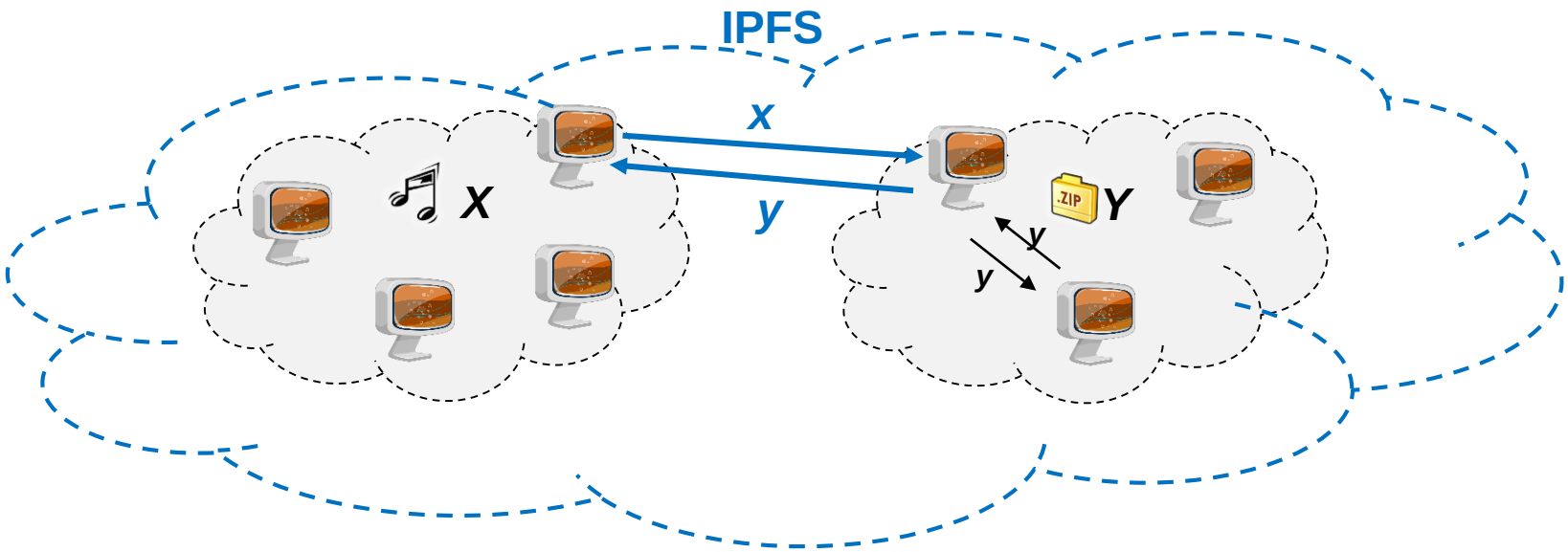
Fonte: https://github.com/ipfs/specs/blob/main/ARCHITECTURE.md

IPFS: roteamento/busca

- Roteamento/busca descentralizada, via **DHT**
  - Se dados pequenos (<1KB): dados armazenados na DHT;
  - Caso contrário, DHT armazena referência para dados (IPs de nós que podem fornecer dados)
- Algoritmos:
  - **S/Kademlia**: estrutura em árvore para busca eficiente
    - Detalhes na aula sobre busca distribuída e DHTs
  - **Coral**: considera **localidade** dos dados para melhorar eficiência das buscas
    - Clusters organizados por região e tamanho: prioriza buscas locais antes de lançar mão de buscas em nós distantes fisicamente
    - “Busca(chave)” retorna subconjunto de IPs que têm o conteúdo em vez de lista completa (“Sloppy DHT”)

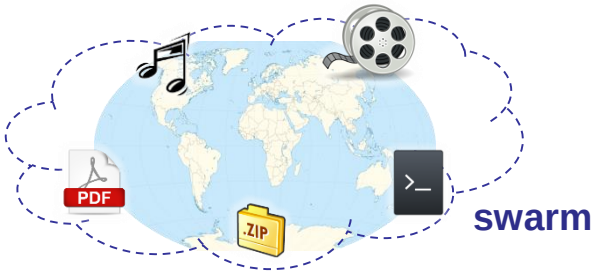
IPFS: troca de dados

- Baseado no **BitSwap**
  - Similar a **BitTorrent**, mas com **um só *swarm*** com todos os conteúdos



IPFS: troca de dados

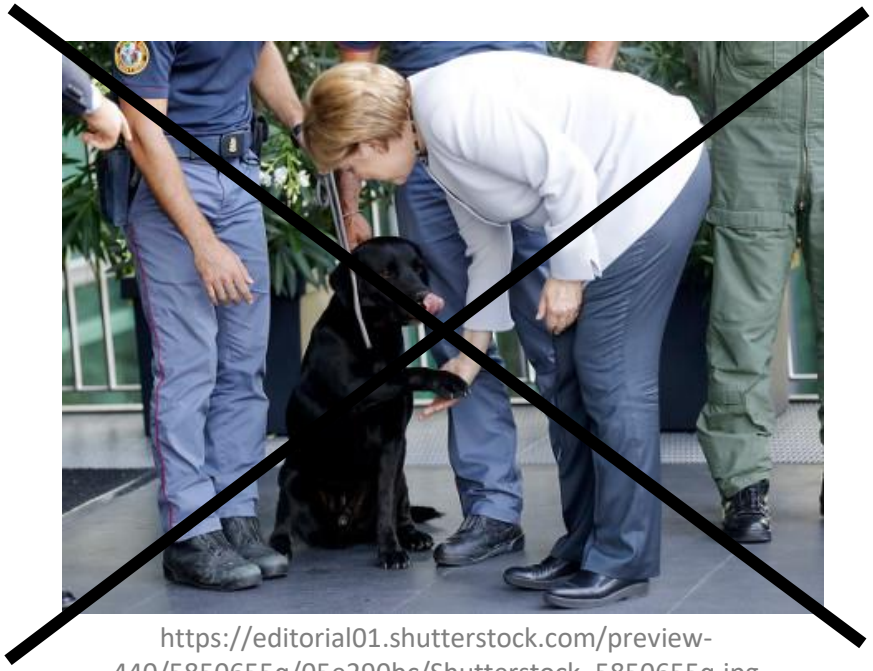
- Baseado no **BitSwap**
  - Similar a **BitTorrent**, mas com **um só *swarm*** com todos os conteúdos



- Mecanismos de **incentivo**:
  - **Tit-for-tat**: incentiva nó A a buscar pedaços que nó B deseja para que então possa receber pedaços vindos de B
    - B anuncia sua "**lista de desejos**" periodicamente a A
  - Nó A mantém lista de "**débitos**" (balanço entre recepção e envio de dados com outros nós) e pode cobrar débito
    - **Leniente** com débito de quem contribuiu no passado: **"nós confiáveis"**
    - Mas não com novos nós, **não confiáveis**: evita ataques de Sybil (um nó assume várias identidades, zerando débitos)
  - Integrável com outros mecanismos
    - Ex.: Filecoin

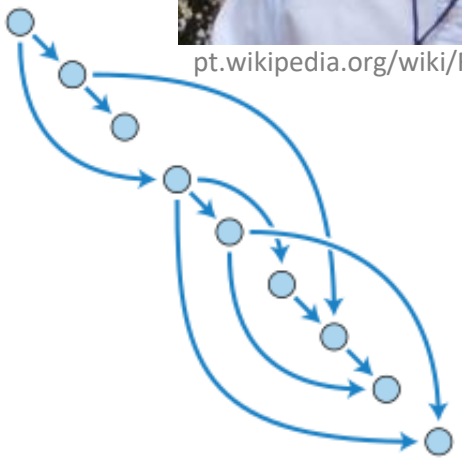
IPFS: versionamento (GIT)

- Usa ~~Merkel DOG~~ Merkle DAG (Grafo Direcionado Acíclico)



https://editorial01.shutterstock.com/preview-440/5850655q/05e290bc/Shutterstock_5850655q.jpg

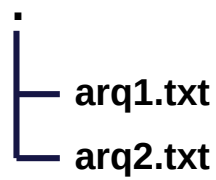
pt.wikipedia.org/wiki/Ralph_Merkle

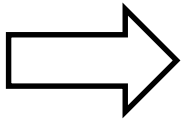


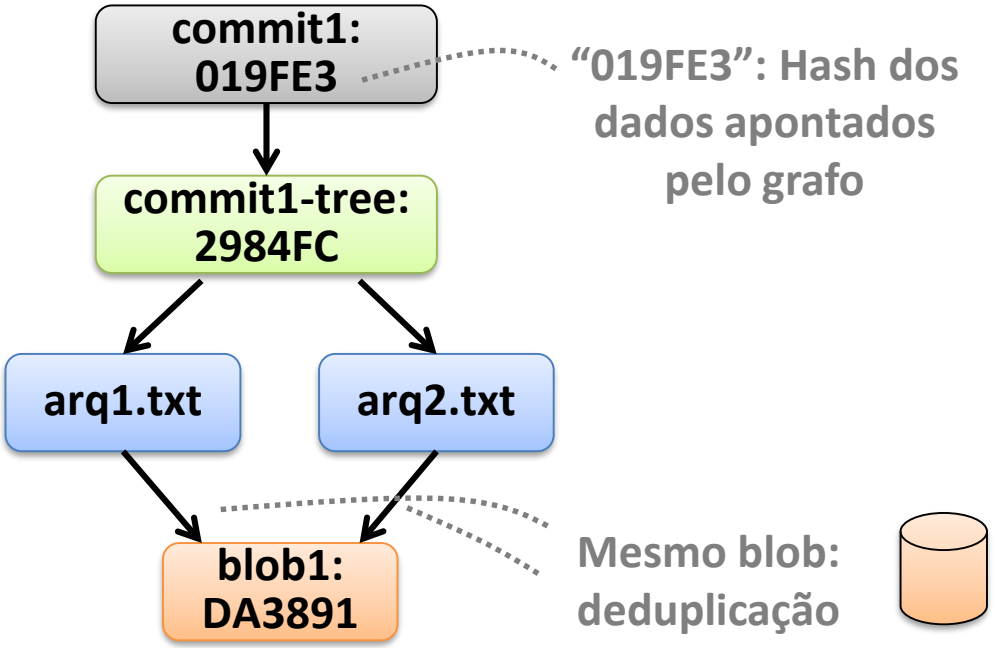
IPFS: versionamento (GIT)

- Usa Merkle DAG (Grafo Direcionado Acíclico)
  - **Commit**: autor, mensagem, ponteiro (hash) para uma árvore
  - **Tree**: ponteiro para árvores e arquivos (estrutura de pastas)
  - **Blob**: dados

Ex.: dois arquivos idênticos na raíz



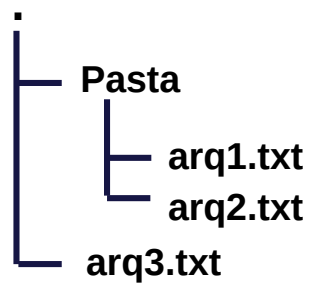


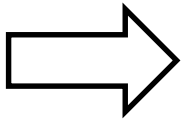


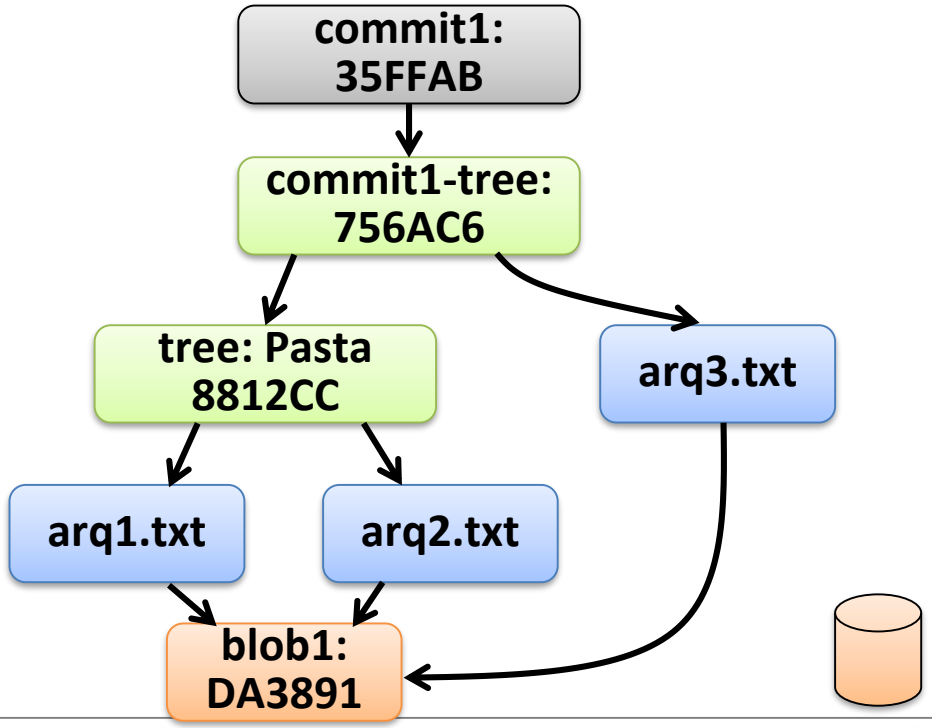
IPFS: versionamento (GIT)

- Usa Merkle DAG (Grafo Direcionado Acíclico)
  - **Commit**: autor, mensagem, ponteiro (hash) para uma árvore
  - **Tree**: ponteiro para árvores e arquivos (estrutura de pastas)
  - **Blob**: dados

Ex.: três arquivos idênticos, em pastas

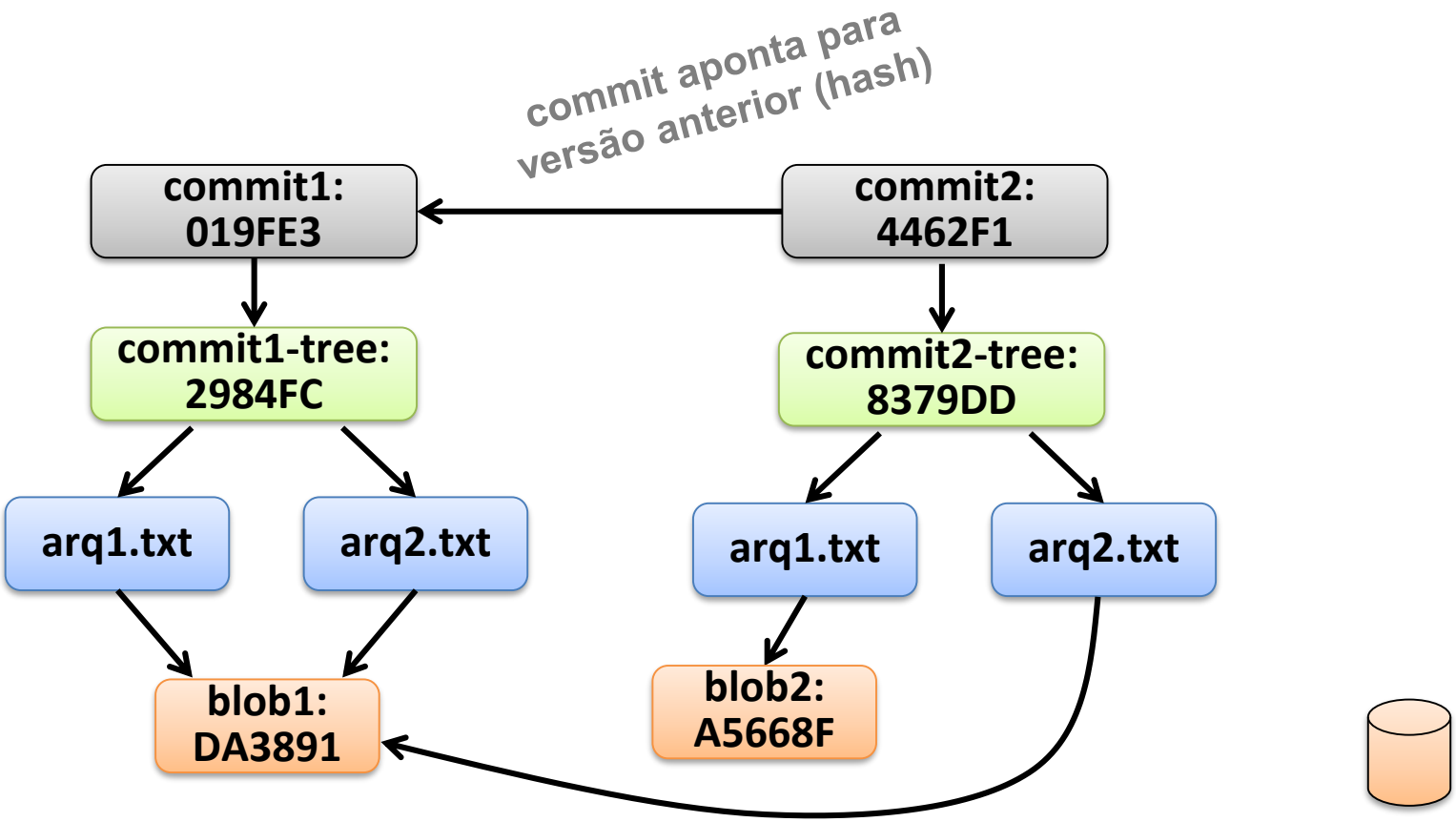






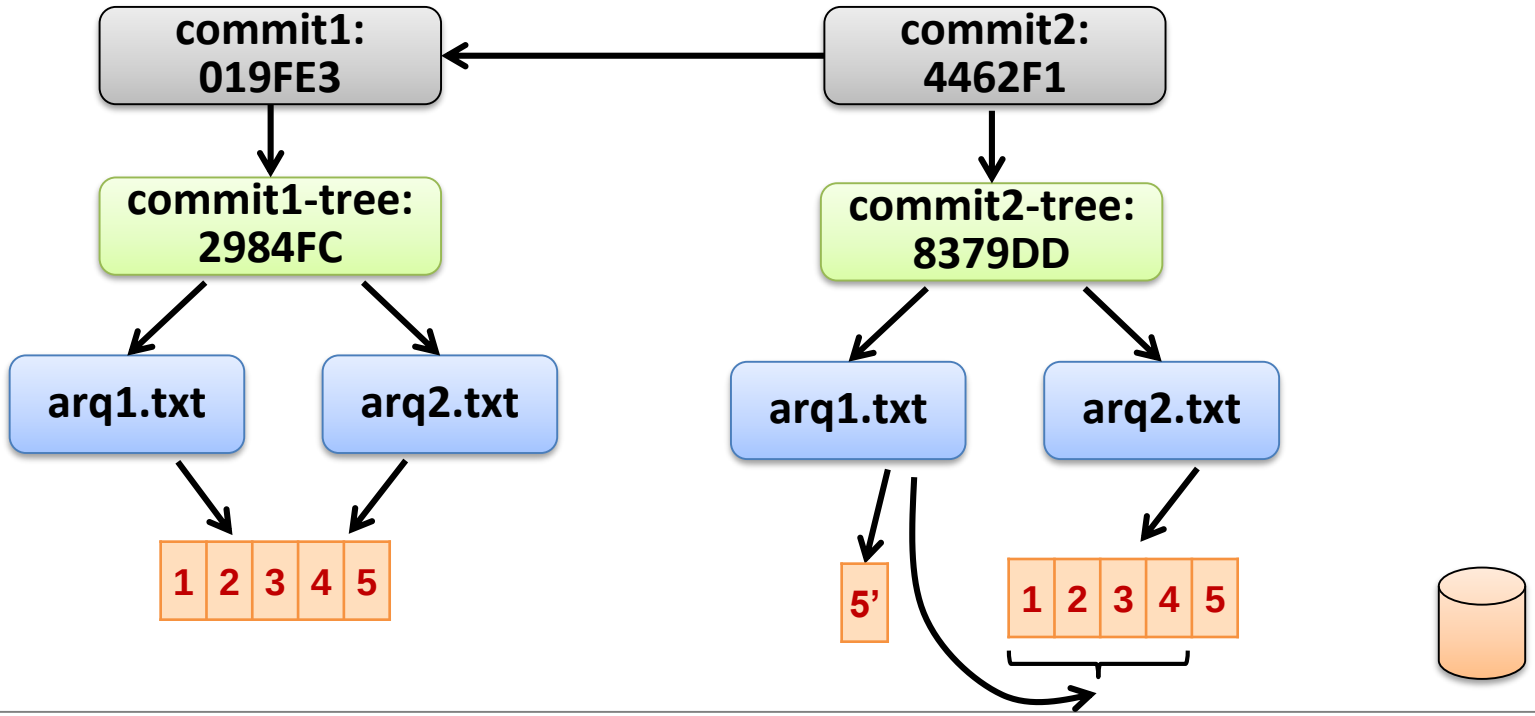
IPFS: versionamento (GIT)

- Controle de versão via hashes
  - Permite acompanhar o caminho das alterações dos arquivos



IPFS: versionamento (GIT)

- Controle de versão via hashes
  - Permite acompanhar o caminho das alterações dos arquivos
  - Se blobs quebrados em blocos: **deduplicação** mais efetiva



IPFS: versionamento (GIT)

- **Objetos** no IPFS: todos identificados pelo seu hash, tanto se forem **arquivos** ou **links**
  - **Multihash** para suporte a diferentes algoritmos
    - Formato: **<Algoritmo><Tamanho_Hash><Bytes_Hash>**
  - Navegação entre links de um domínio = navegação no Merkle DAG, usando o hash de cada link

Formato: `/ipfs/<hash-do-objeto>/<nome-caminho-até-objeto>`

Ex.: `/ipfs/XLYkgq61DYaQ8NhkcqyU7rLcnSa7dSHQ16x/arq.txt`

  - Para acessar o arquivo “fig.png” localizado no caminho “<dominio>/pasta/fig.png, pode-se usar qualquer das opções:

1) Domínio: `/ipfs/<hash-de-dominio>/pasta/fig.png`

2) Pasta: `/ipfs/<hash-de-pasta>/fig.png`

3) Arquivo: `/ipfs/<hash-de-fig.png>`

IPFS: versionamento (GIT)

- Ex.: processo de obtenção do arquivo

`/ipfs/QmdpMvUptHuGysVn6mj69K53EhitFd2LzeHCmHrHasHjVX/test/foo`

1. Começar obtendo o multihash: `get(QmdpMvUpt...)`

`QmdpMvUptHuGysVn6mj69K53EhitFd2LzeHCmHrHasHjVX` ←

```
cat.jpg   QmajFHH...
cat2.jpg  QmajFHH...
test      QmNtpA5...
tree.jpg  QmWNj1p...
```

2. Agora usamos o hash de test: `get(QmNtpA5...)`

`QmNtpA5TBNqHrKf3cLQ1AiUKXiE4JmUodbG5gXrajg8wdv` ←

```
bar       QmTz3oc...
baz       QmX1ebV...
foo       QmYNmQK...
```

3. Agora usamos o hash de foo: `get(QmYNmQK...)`

`QmYNmQKp6SuaVrpgWRsPTgCQCnpxUYGq76YEKBXuj2N4H6` ←

```
foo
```

IPFS: versionamento (GIT)

- Qualquer usuário pode publicar objetos na rede
  - Basta incluir hash do objeto na DHT, se declarar como um peer para o objeto e divulgar caminho para objeto

IPFS: versionamento (GIT)

- Qualquer usuário pode publicar objetos na rede
  - Basta incluir hash do objeto na DHT, se declarar como um peer para o objeto e divulgar caminho para objeto
- Uso de Merkle DAGs: hash não pode ser alterado, logo **objetos são permanentes**!
  - Redução do consumo de banda: **caching**
  - Conteúdo servido por nós **sem confiança**: análogo ao que ocorre no Bittorrent
  - **Links são permanentes**: sem “links quebrados” desde que alguém tenha arquivo
  - Usuários podem escolher fazer backups de **dados específicos** para garantir sua **longevidade**

IPFS: Sistema de nomes

- Sistema de arquivos auto-certificado

  - *Self Certified Filesystem* (SFS)
  - **Identificadores dos nós** (`NodeId`) correspondem ao **hash de suas chaves públicas**
    - `NodeID` usado para roteamento de buscas
    - Dificulta “escolha do ID” para eventuais ataques (S/Kademlia)

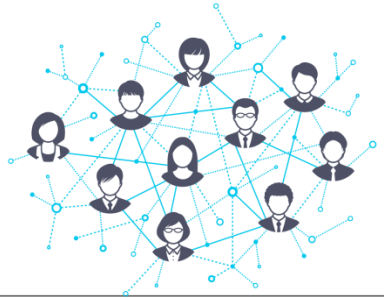
Formato: `/ipns/<NodeID>`

Ex.: `/ipns/hj17rsy89MnOo`

IPFS: Sistema de nomes

- Sistema de arquivos auto-certificado

  - *Self Certified Filesystem* (**SFS**)
  - **Identificadores dos nós** (`NodeId`) correspondem ao **hash de suas chaves públicas**
    - `NodeID` usado para roteamento de buscas
    - Dificulta “escolha do ID” para eventuais ataques (S/Kademlia)
  - Totalmente descentralizado: basta servidor provar **posse da chave privada** para verificar corretude
    - Permite usar chaves para **estabelecer canais seguros**
    - **Similar a HTTPS**, mas **sem** depender de certificados digitais emitidos por **Autoridade Certificadora**



Formato: `/ipns/<NodeID>`

Ex.: `/ipns/hj17rsy89MnOo`

IPFS: Sistema de nomes

- IPNS (Naming System): **mutabilidade** de objetos
  - Nó pode associar seu **domínio** `/ipns/<NodeId>` a um objeto
  - Busca na DHT por domínio retorna **versão atual do objeto** (seu hash), que pode ser modificado quando desejado
    - Assinatura do servidor sobre objeto garante autenticidade
    - Objeto pode ser **commit de página web** completa: carrega **histórico de versões**!

mesma chave

NodeID	Valor
/ipns/hj17rsy89MnOo	98sdfHjyu87q
/ipns/hj17rsy89MnOo	3dfvJ3KdgYr

conteúdos distintos

IPFS: Sistema de nomes

- IPNS (Naming System): **mutabilidade** de objetos
  - Facilita nomes amigáveis a humanos, no lugar de hashes
  - Ex.: criar registro DNS (e.g., blockchainusp.blog) p/ nodeID
    - blockchainusp.blog: "`dnslink= /ipns/hj17rsy89MnOo`"

Serviço centralizado ou distribuído (Namecoin, Blockstack, …)

DNS	NodeID	Valor
/ipns/blockchainusp.blog	**/ipns/**hj17rsy89MnOo	98sdfHjyu87q
/ipns/blockchainusp.blog	**/ipns/**hj17rsy89MnOo	3dfvJ3KdgYr

Leitura: https://decentralized.blog/ten-terrible-attempts-to-make-ipfs-human-friendly.html

IPFS: Teste você mesmo!

- Solução ainda razoavelmente experimental
  - Mas “mão na massa” ajuda a entender funcionamento!

https://docs.ipfs.tech

Blockchain, Criptomoedas & Tecnologias Descentralizadas

Tecnologias descentralizadas:

Interplanetary File System (IPFS)

Prof. Dr. Marcos A. Simplicio Jr. – mjunior@larc.usp.br
Escola Politécnica, Universidade de São Paulo

Apoio financeiro: d24n.org <https://d24n.org>
ripple impact | university blockchain research initiative

Referências

- “IPFS powers the Distributed Web” (online). Página oficial do Interplanetary File System. URL: https://ipfs.io/
- "IPFS docs" (online). URL: https://docs.ipfs.tech
- J. Benet (2019). “IPFS - Content Addressed, Versioned, P2P File System (DRAFT 3). White paper. URL: https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf
- M. Pors (2017). "Ten terrible attempts to make IPFS human-friendly". The Blockchain Train Journal. URL: https://decentralized.blog/ten-terrible-attempts-to-make-ipfs-human-friendly.html