

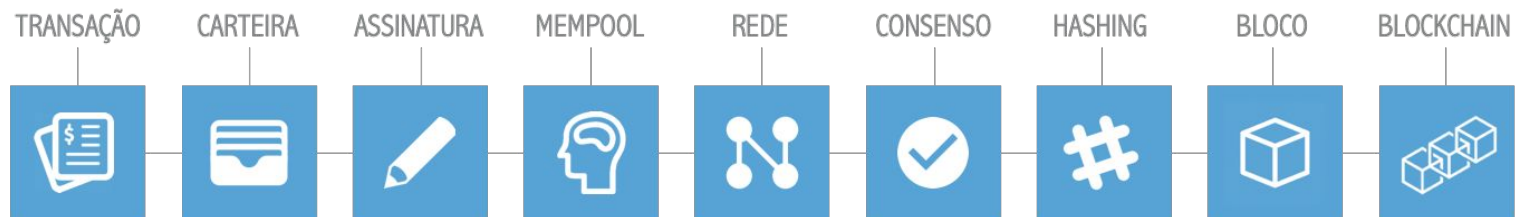
IMD0913

# ARQUITETURA DE UM BLOCKCHAIN

## CARTEIRAS (WALLETS)

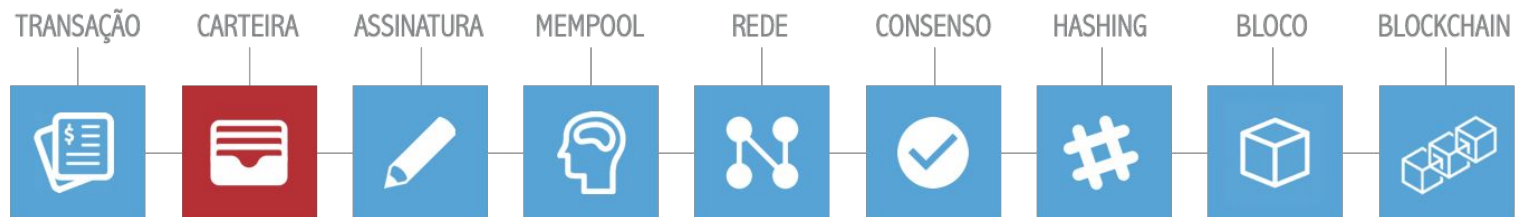
## ARQUITETURA DE UM **BLOCKCHAIN**

---



## ARQUITETURA DE UM **BLOCKCHAIN**

---



## Carteira (*wallet*)

Conceito abstrato para coleção de chaves privadas e públicas.

## **Chave privada** (no contexto do Bitcoin)

Uma chave secreta que permite que você “gaste” bitcoins da sua carteira.

## **Chave pública** (no contexto do Bitcoin)

Uma chave compartilhada publicamente utilizada para receber bitcoins.

# Endereço Bitcoin

## SHA256

Gera um *hash* de 256 *bits*

## RIPEMD

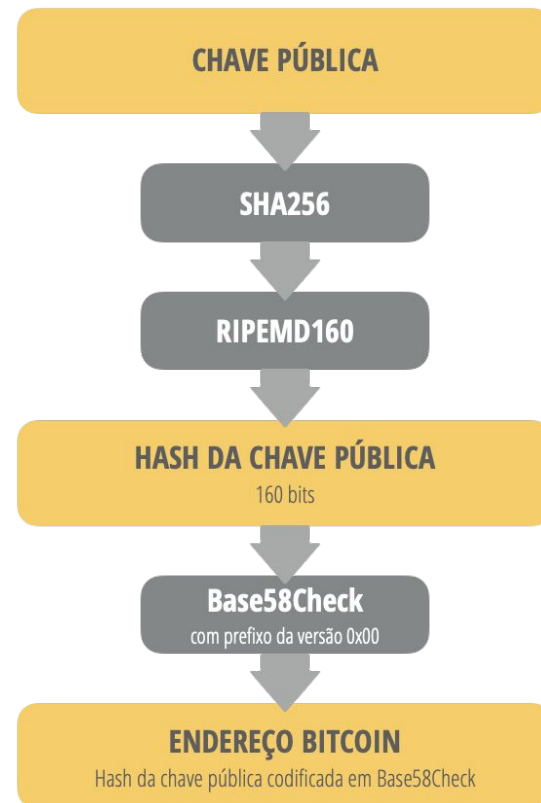
Gera um *hash* de 160 *bits*

## Base58Check

Codificação em base 58

123456789ABCDEFGHJKLMNPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz

Exclui 0, O, I, l



# Base58

Base	Caracteres
2 (binário)	01
10 (decimal)	0123456789
16 (hexadecimal)	0123456789abcdef
58	123456789ABCDEFGH JKLMN PQRSTUVWXYZabcdefghijk mnopqrstuvwxyz

$\text{base2}(9999) = 10011100001111$

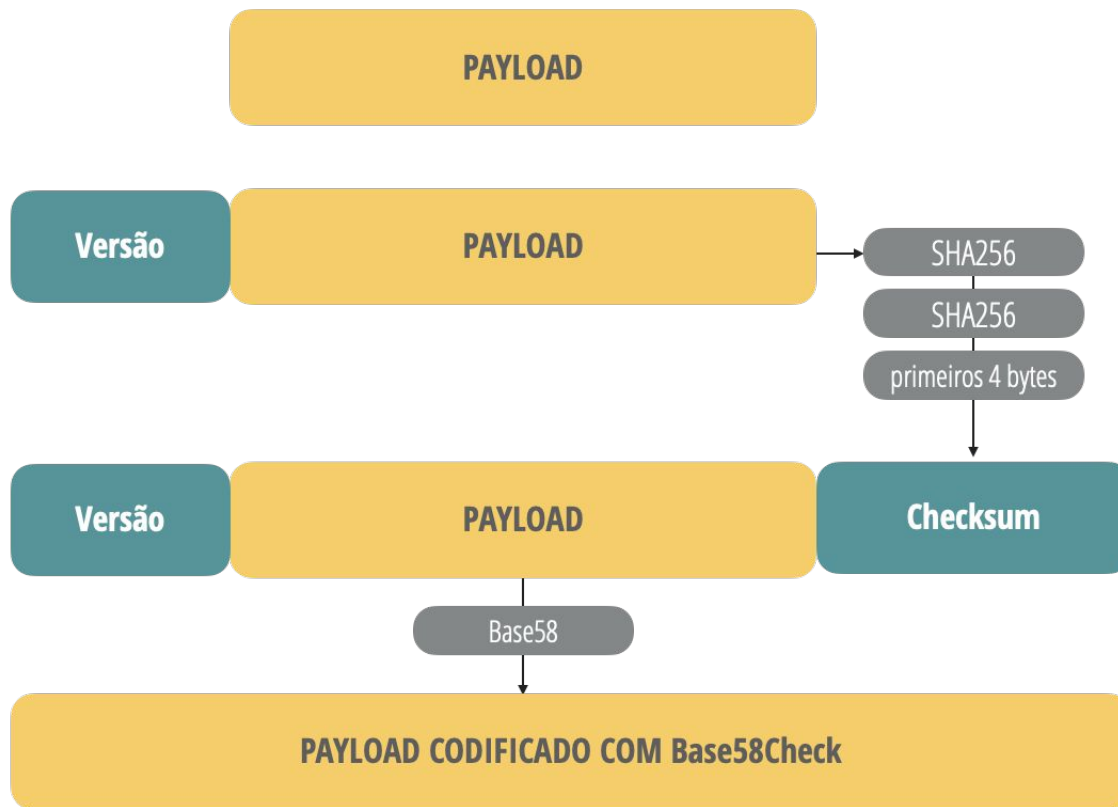
$\text{base10}(9999) = 9999$

$\text{base16}(9999) = 270f$

$\text{base58}(9999) = 3yQ$



# Base58Check



# Endereço Bitcoin

Tipo	Prefixo da versão (hex)	Prefixo resultante em Base58
Endereço Bitcoin	0x00	1
Endereço Bitcoin ( <i>testnet</i> )	0x6F	m ou n

\* existem ainda outros prefixos. Ex: P2SH

**Chave privada(256 bits / 32 bytes):**

9B680320758E0E26BEBA32C9576F5D5A99ABB8D899A8B17F405AE96C09B476F8

**Chave pública (520 bits / 65 bytes):**

0416173A8EE74D83AFCE5C5AFE08CEEE4CFBCA60D719CF4155BDD429FD937EE81  
858B28C5A5515EEE06461923B45C1579E29ACF423D9F526D2057D4B6512877B04

**SHA256 da chave pública (256 bits / 32 bytes):**

FADD7E4B37C823886E1ECD9EDCD17FD5802D0B11850C293E9F777E1CFA24FF03

**RIPE160 do SHA256 da chave pública (160 bits / 20 bytes):**

D04193B453D5AD087AD1967818DF25922D5C8D13

**Endereço bitcoin (Base58Check):**

1KzA7GaFQ63u5J5rUHcrn5rGxWAtNyd2FM

# Formas de representação da chave privada

Tipo	Prefixo	Descrição
Raw	Nenhum	32 bytes
Hex	Nenhum	64 dígitos hex
WIF	5	Codificação Base58Check: Base58 com prefixo de versão 128 (0x80) e checksum de 4 bytes (32 bits)
WIF-compressed	K ou L	Igual a de cima, adicionando o sufixo 0x01 antes de codificar

Uma chave privada WIF é apenas outra maneira de representar sua chave privada original. Se você tiver uma chave privada WIF, sempre poderá convertê-la de volta ao formato original.

# Formas de representação da chave privada

Tipo	Exemplo
Raw	1101101001000110101101010101100111110010000110110011111010010101 01011011110110001100100100101110010010110010010101100010111000011 10110011110101110010111111000011011111001101110100011101101010 0001000001001011000011100111001110010110000000100111101101100101
Hex	DA46B559F21B3E955BB1925C964AC5C3B3D72FE1BF37476A104B0E7396027B65
WIF	5KUR9tz4iDTpW2xQkNvJDKyGHYWT9q8LriTLH29Tv8Thiyqvy9A
WIF-compressed	L4Y1cGSsNv1Nf9dZpTkEyQjLU24zRyRQeRyE5i4MoVvrjrr15Koy

As chaves privadas e públicas podem ser representadas em vários formatos diferentes.  
Todas essas representações codificam o mesmo número!

# Carteiras

Para garantir nossa **identidade**, precisamos proteger nossa **chave privada**

Como gerenciamos todas as nossas chaves? com **CARTEIRAS** ou **WALLETS**

**ENDEREÇO:**

1JJQmRbU9JT9mfxp756Y  
MuxV6yksKtbk5

**CHAVE PRIVADA:**

L1fm3iAFdDHwSD3CZuZmWp54G  
XpQ6QzUjmrACVfKKE8BkggW99u3

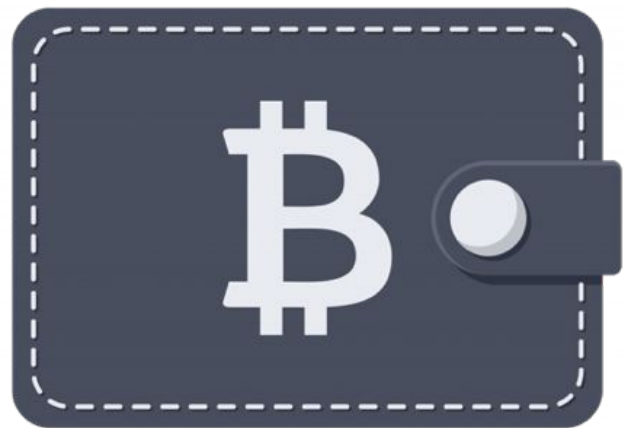
# Carteiras

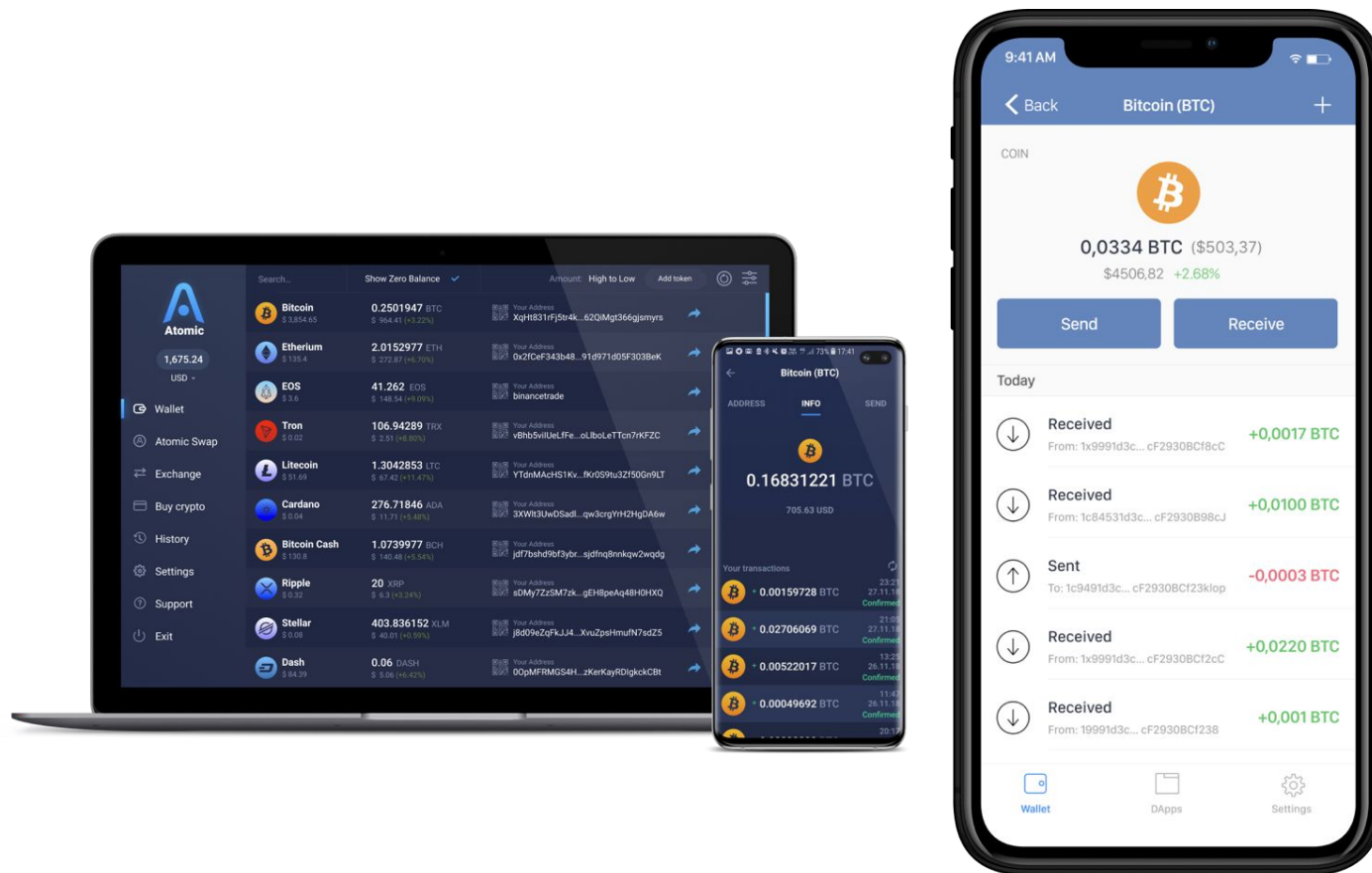
O que carteiras fazem?

- Mantém registro de sua chave privada

- Armazenam, enviam, recebem e listam transações

- Opcionalmente alguma outra funcionalidade







# Carteiras

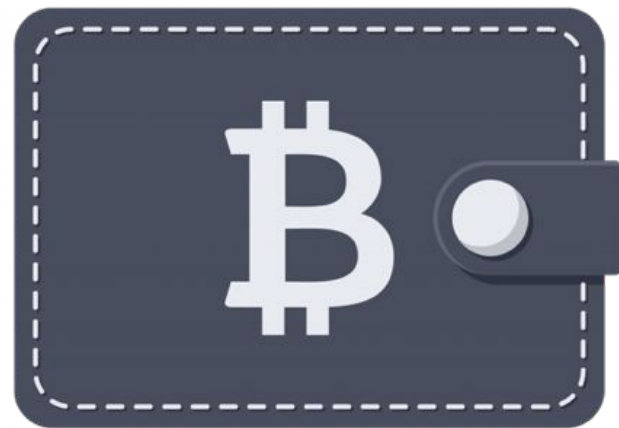
Carteiras não contém bitcoins

**Carteiras contém chaves** (pares de chaves privada/pública)

Tipos de carteiras:

- Não determinísticas

- Determinísticas



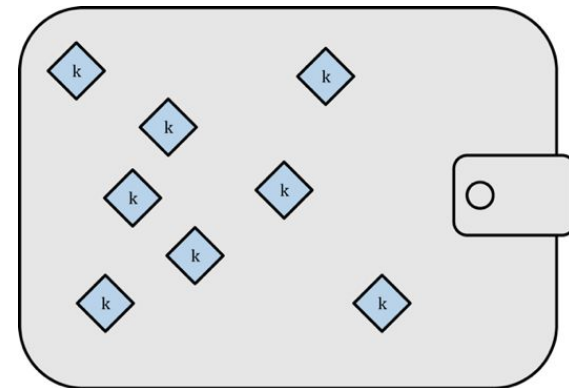
# Tipos de carteiras

## Não determinística (tipo 0)

Cada chave é gerada independentemente e aleatoriamente

As chaves não tem nenhuma relação

Carteira também chamada de JBOK (*Just a Bunch Of Keys*)



número aleatório  $\Rightarrow$  chave privada  $\Rightarrow$  chave pública  $\Rightarrow$  endereço

# Tipos de carteiras

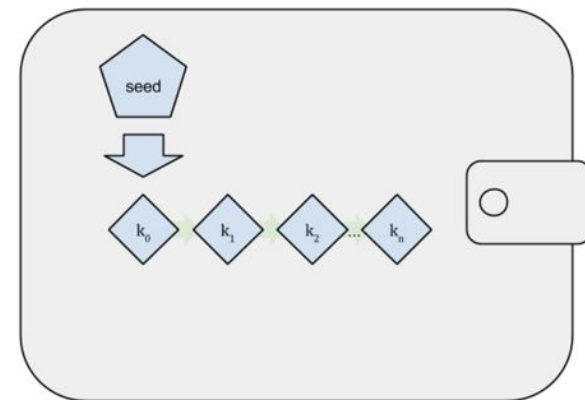
## Determinística baseada em seed (tipo 1)

Contém chaves privadas que derivam de uma semente (*seed*) comum através de uma função *hash*

A semente é gerada aleatoriamente, e a partir dela combinada com outros dados são derivadas as chaves privadas

A semente é suficiente para recuperar todas as chaves derivadas

Fácil de exportar e importar a carteira



$seed \Rightarrow \text{chave mestre} \Rightarrow \text{chave(s) privada(s)} \Rightarrow \text{chave(s) pública(s)} \Rightarrow \text{endereço(s)}$

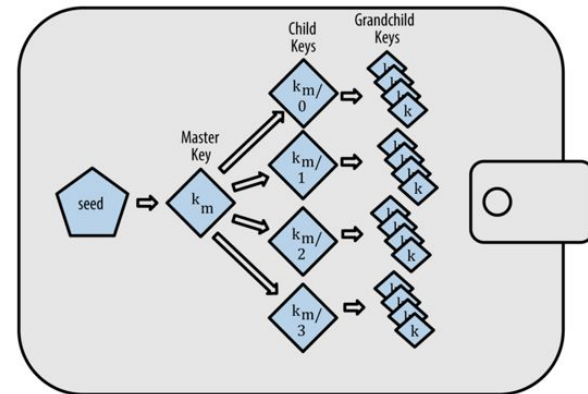
# Tipos de carteiras

## Determinística do tipo HD (tipo 2)

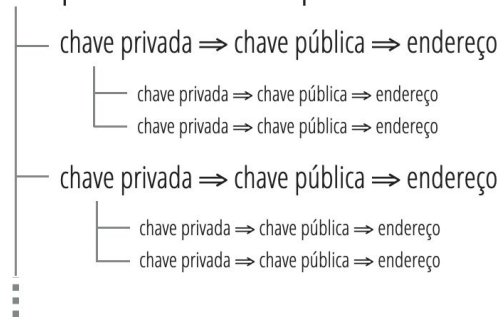
HD = *Hierarchical Deterministic*

Facilita a derivação das chaves a partir da semente comum (determinística)

Organizadas em formato de árvore (hierárquica)



$seed \Rightarrow$  chave mestre  $\Rightarrow$  chave privada  $\Rightarrow$  chave pública  $\Rightarrow$  endereço



# HD Wallets: como funcionam?

## 1. Semente (seed)

Gerar 64 *bytes* aleatórios



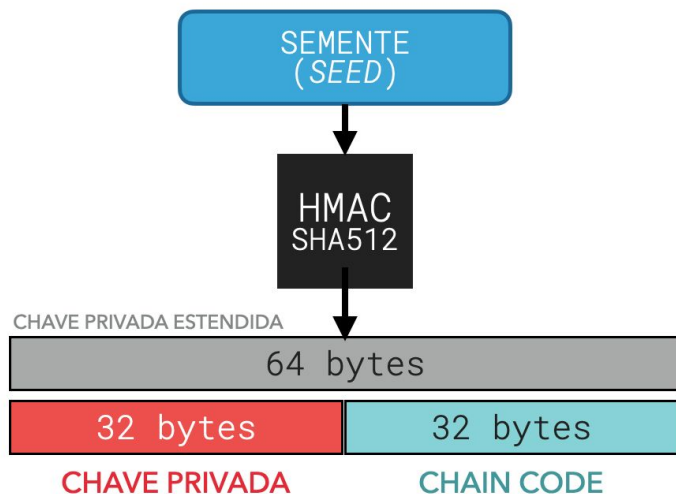
### EXEMPLO:

2f9159acc5566abd10925ee1623f433d2c9f6e3aa385c021d6a1022ad06dcd11  
f162f3ccab192d4fcd5fea3823a2ed2a9c7f9fffe11bab49be5b28deb7805707

# HD Wallets: como funcionam?

## 2. Chave privada mestra (*master private key*)

A chave mestra é gerada passando a semente em um função *hash* (HMAC-SHA512)



### EXEMPLO:

2f9159acc5566abd10925ee1623f433d2c9f6e3aa385c021d6a1022ad06dcd11  
f162f3ccab192d4fcd5fea3823a2ed2a9c7f9fffe11bab49be5b28deb7805707

# HD Wallets: como funcionam?

## 3. Chaves filhas (*child keys*)

Novas chaves privadas filhas são geradas a partir da HMAC da chave privada estendida

Um índice (*index*) também é incluído cada vez que o processo é repetido, para conseguir criar múltiplas chaves filhas a partir de uma única chave

semente (seed): 2f9159acc5566abd10925ee1623f433d2c9f6e3aa385c021d6a1022ad06dcd11  
f162f3ccab192d4fcd5fea3823a2ed2a9c7f9fffe11bab49be5b28deb7805707

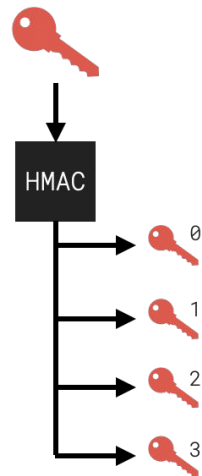
chave privada mestre estendida (master extended private key):  
chave privada: 1efa2e9900a404a7936d4f3f17de93ae524df20e59ba13ab840b36d43855fc22  
chain code: 7dffee063d56a488357fb7be24dea4e54b076fa0628f82dc9426e0f7cd934f3

filho 0:  
chave privada: ff2767602ba9cafe1bcadc6549491693fbb1b943ed2b9cf789990d7aed205f2a

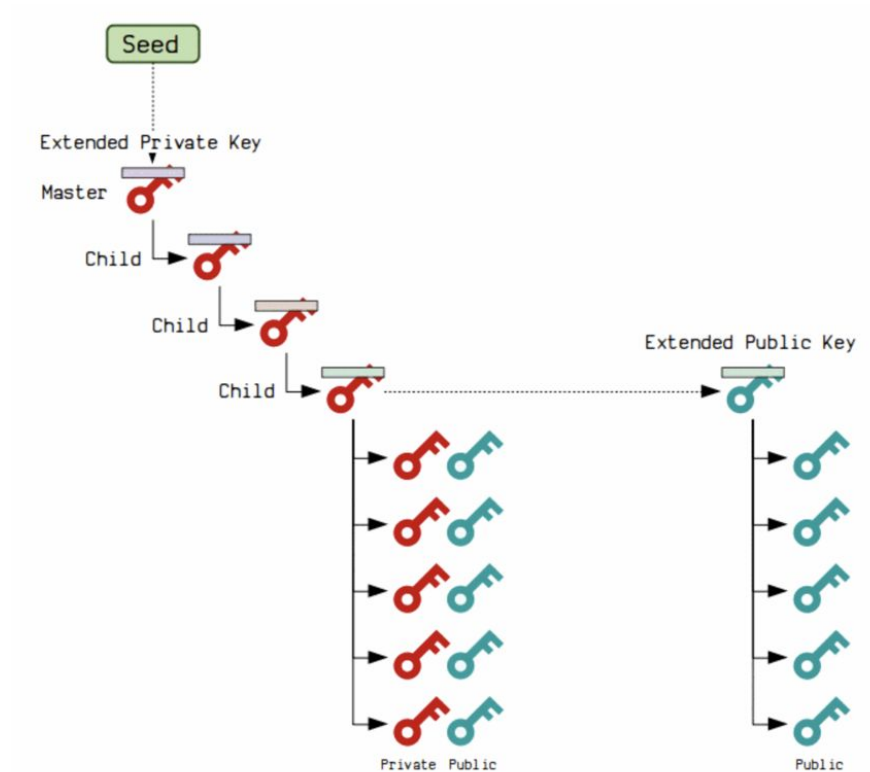
filho 1:  
chave privada: f1cb6942fd88c8cf62af00a1023abf1c8c5e2d54a32d51479f0f85b4a7b4e47d

filho 2:  
chave privada: c3752635e82648576fefa4f7a84aa12f1869c2f1ec582fd40551dbf85861a2db

CHAVE PRIVADA ESTENDIDA



# HD Wallets





# Formas de carteiras

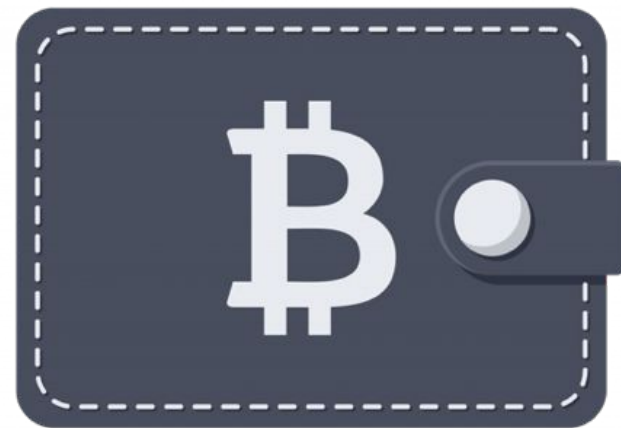
apps

web-wallets

paper wallets

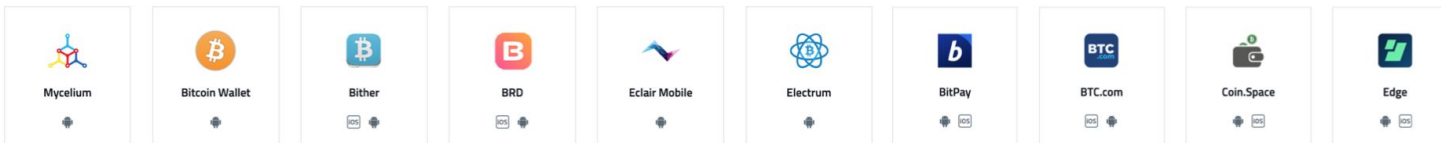
hardware wallets

brain wallet

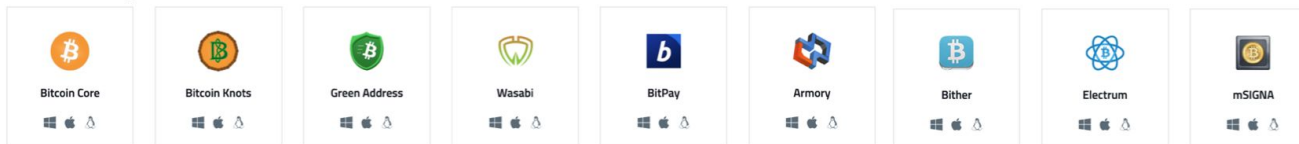


# Formas de carteiras: *apps e webwallets*

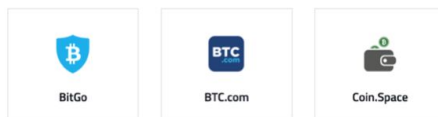
## Smartphone



## Desktop

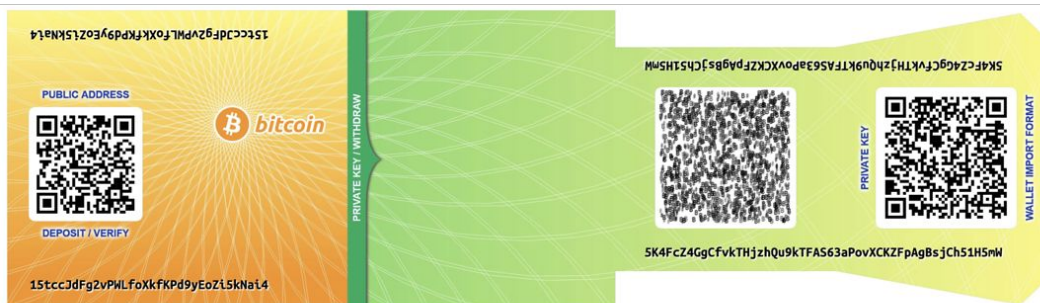
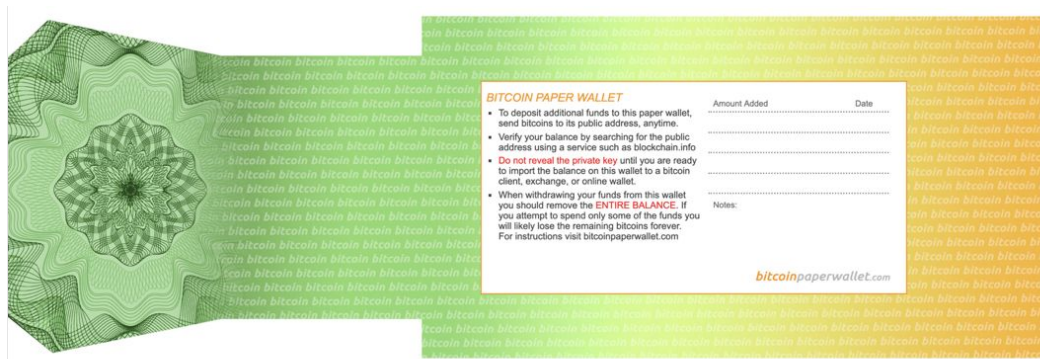


## Web



hot wallet

# Formas de carteiras: *paper wallets*



Your **public** key is: 15tccJdFg2vPWLfoXkFPd9yEoZi5kNaI4  
Receive bitcoin to your wallet using your PUBLIC key.

Your **private** key is: 5K4FcZ4GgCfVktHjzhQu9kTFAS63aPovXCKZfAgBsJCh51H5mW  
Access bitcoin in your wallet using your PRIVATE key.

cold wallet

# Formas de carteiras: *hardware*

## Hardware



cold wallet

## Formas de carteiras: *brain wallets*

**Simplesmente memorize sua chave privada!** (em qualquer formato)

L2Skyj3pJK3nc7wgr9afokGL89dPWV3iHQJvZiy2zEwvXDQReAgg



cold wallet

# Sementes mnemônicas

Comumente *seeds* são gerada a partir de uma **mnemônica**, ou uma coleção de palavras/frases

Maneira conveniente de memorizar sua chave privada

Não tão seguro pois humanos não são tão aleatórios como pensamos

Existe uma documentação para geração das palavras (BIP-39)

<https://github.com/bitcoin/bips/blob/master/bip-0039/english.txt>

<https://github.com/bitcoin/bips/blob/master/bip-0039/portuguese.txt>

hope mouse focus family animal near chest february pipe access sudden please

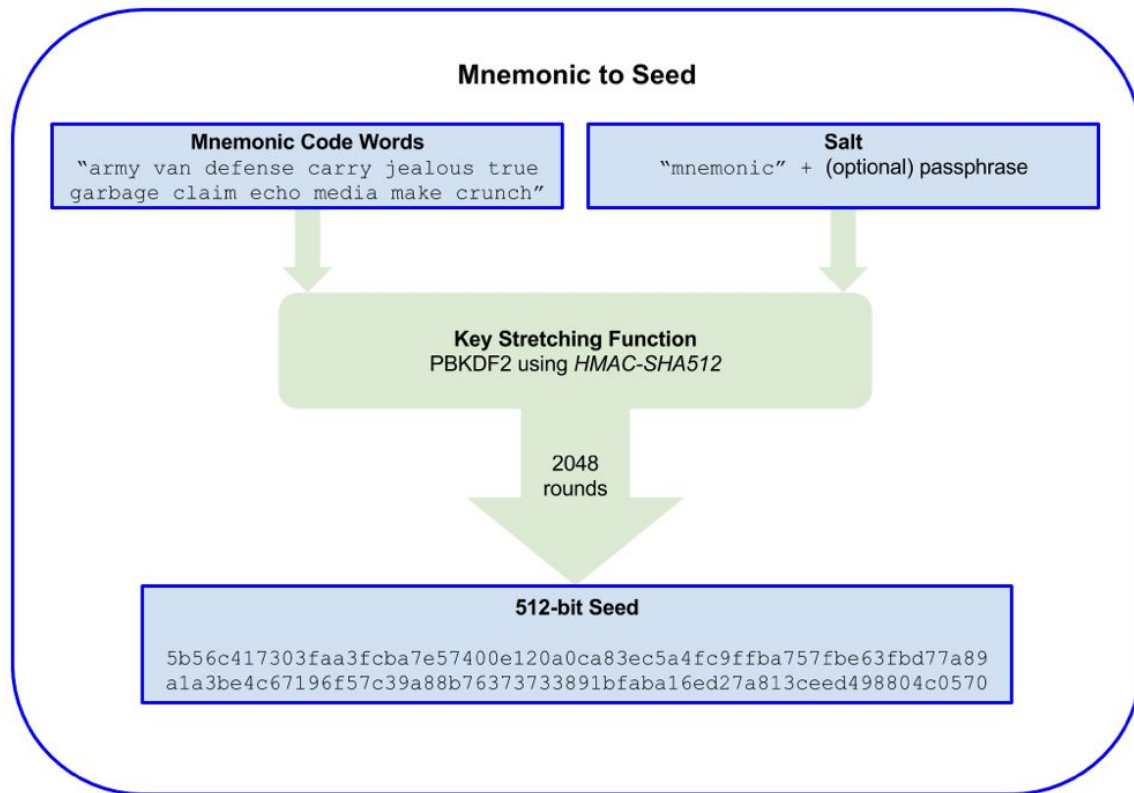
# Sementes mnemônicas

hope mouse focus family animal near chest february pipe access sudden please



7f662bb1013a72f20e0c4c3f5320b31b2ea9e836a64f06daa322c10a20d89a4f

# Sementes mnemônicas - BIP39





# Carteiras

Melhor prática: não reutilizar endereços

Por que?

- Para ninguém conseguir determinar quanto BTC você tem

- Comprometer uma chave é independente das outras

- Chaves são facilmente (computacionalmente falando) geradas

*Software* da carteira vai fazer isso!

# Endereços em Python

```
1  @staticmethod
2  def getWifCompressedPrivateKey(private_key=None):
3      # Retorna a chave privada no formato WIF-compressed da chave privada hex.
4      if private_key is None:
5          private_key = bitcoinlib.random_key()
6      return bitcoinlib.encode_privkey(bitcoinlib.decode_privkey((private_key + '01'), 'hex'), 'wif')
7
8  @staticmethod
9  def getBitcoinAddressFromWifCompressed(wif_pkey):
10     # Retorna o endereço Bitcoin da chave privada WIF-compressed.
11     return bitcoinlib.pubtoaddr(bitcoinlib.privkey_to_pubkey(wif_pkey))
```

