GoLedger

Construa seu primeiro Blockchain

Samuel Venzi
CTO
samuel.venzi@goledger.com.br

Marcos Sarres
CEO
marcos.sarres@goledger.com.br

Perfil Dev: Samuel Venzi

Passado

- C/C++
- Visão Computacional em OpenCV
- Ciência de Dados em Python
- Aprendizagem por Reforço em Go

Hoje

- Hyperledger Fabric
- Golang
- Node.js
- Docker

Perfil Dev: Marcos Sarres

Passado

- QT/PhoneGap
- PHP
- Angular
- MFC/X11
- C++/Assembler (PC)
- Basic/Assembler (MSX)

Hoje

- Hyperledger Fabric
- Golang
- Solidity
- TypeScript/JavaScript
- React.js/React Native

O livro-razão e o desejo humano de contar

O **Livro-Razão** ou ***Ledger*** sempre foi utilizado pela humanidade para registro de valores e propriedades.

Livro-razão da Mesopotâmia

Antes do Blockchain, havia o DLT

Assinatura, cópia dos **ledgers** e armazenamento em lugares distintos como solução de combate à fraude.

O **Livro-Razão Distribuído** ou **Distributed Ledger** ou **DLT** foi a solução encontrada para aumentar a confiabilidade dos valores.

Relato de uso de DLT pelo sistema financeiro no **Império Romano**.

E com o Bitcoin nasce a tecnologia Blockchain, um DLT com regras de consenso.

Que problema o Blockchain resolve?

Blockchains são poderosas ferramentas que permitem contagem de coisas de forma distribuída

- Imutabilidade do Histórico
- Confiabilidade das transações
- Consenso entre participantes
- Auditoria à prova de fraudes
- Integração entre bases
- Escalabilidade

Cases Blockchain Corporativos

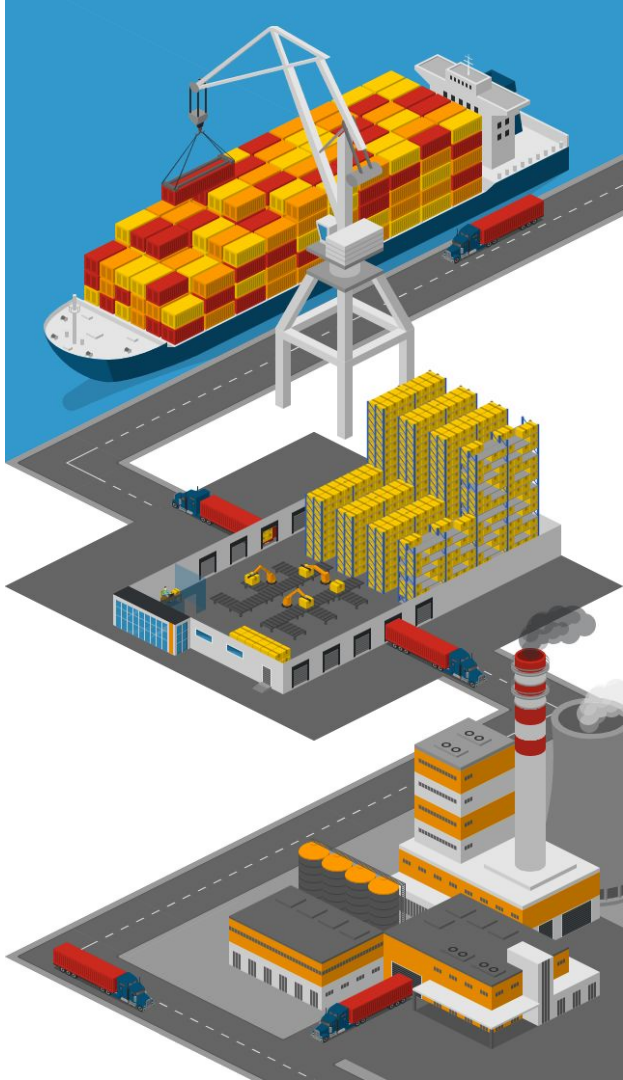
TradeLens: *Maior plataforma de governança rastreabilidade de comércio exterior do mundo.*

Ecotrace: *rastreabilidade de insumos agrícolas JBS, FriGol, etc*

Min. Saúde: *dados de saúde e vacinas*

Diário de Bordo Digital / ANAC: *registro de voos e manutenções de aeronaves*

bCPF: *compartilhamento interno de dados de cidadãos entre entidades governamentais*

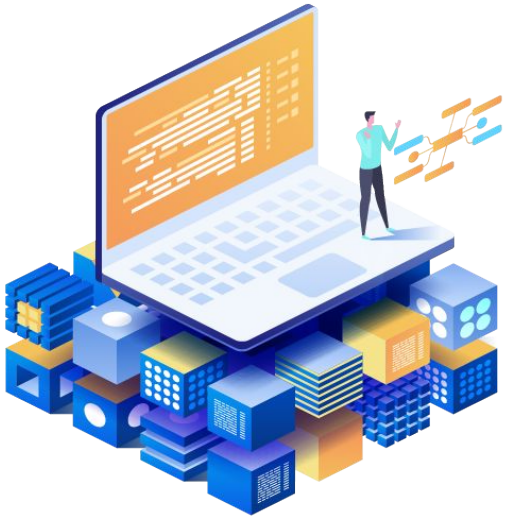


Projetos em que estamos trabalhando

GoLedger

- *Controle de Estoque do Gás Natural Brasileiro*
- *Tokenização de direitos autorais musicais*
- *Rastreabilidade de Placas de Veículos*
- *Gestão de certificados x509 e rastreabilidade de PDFs*
- *Controle de licenciamento de automóveis*

Oportunidades para Devs Blockchains



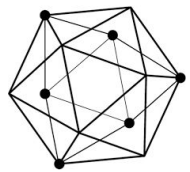
Mercado em crescimento para programadores:

- Exchanges (Crypto)
- Gestão de infra-estrutura (DevOps)
- Contratos inteligentes para Blockchains privados
- APIs/Wallets

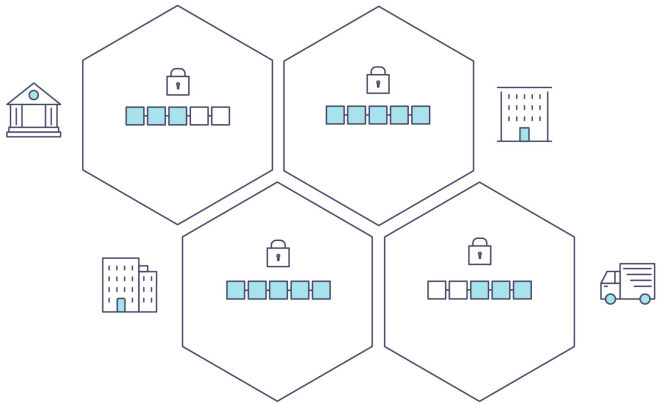
Tecnologias

- Ethereum
- Hyperledger Fabric
- Corda

Hyperledger Fabric

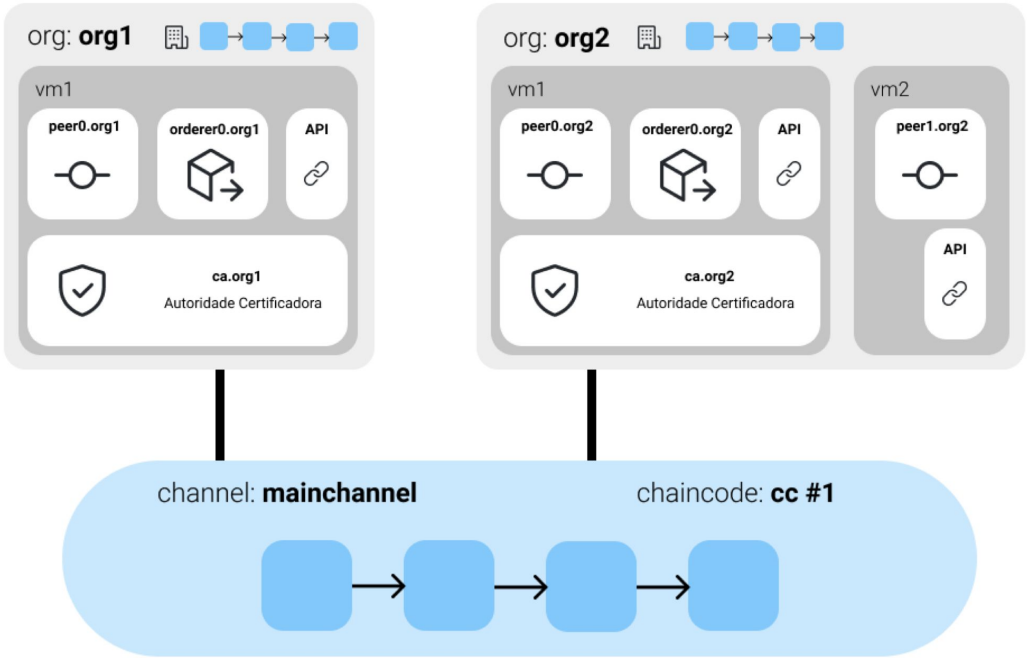


- Redes de **ledgers distribuídos** entre participantes interessados
- Redes **permissionadas**
- Suporte a contratos inteligentes ou **chaincodes**
- **Open-source**
- Mais utilizado em projetos corporativos no mundo



Conceitos Chave de HLF

- Organizações
  - CAs
  - Peers
  - Orderers
- Canais
- ***Chaincodes***



O que guarda o Ledger do HLF?

Chave ➡️ Valor

```
"MEU-REGISTRO-1": "VALOR DO MEU REGISTRO"


"ITEM023": "Carro Sedan #23, Status: Vendido"


"contrato:abc76d57-bc75-4cd8-a038-97fa81054acb": {
    "contratado": "Fulano de Tal",
    "valor": 2000.00,
    "data": "2016-09-01T10:11:12.123456-0500"
}
```

A ***chave*** permite rastrearmos o histórico do ***valor*** registrado.

Chaincodes para Hyperledger Fabric

Etapas:

1. Mapeamento de Dados
2. Desenvolvimento de Transações
3. Arquitetura de Organizações em produção

Pré-requisitos:

1. Binários do Fabric
2. Golang
3. Docker

Chaincode Exemplo #1: `fabric-samples/fabcar`

Definição de um **ativo**:

```
// Car describes basic details of what makes up a car
type Car struct {
    Make   string `json:"make"`
    Model  string `json:"model"`
    Colour string `json:"colour"`
    Owner  string `json:"owner"`
}
```

disponível no GitHub

- Struct padrão do Golang
- Campos com tipos definidos pelo Golang

Chaincode Exemplo #1: `fabric-samples/fabcar`

Definição de uma **transação**:

```
// CreateCar adds a new car to the world state with given details
func (s *SmartContract) CreateCar(ctx contractapi, carNumber, make, model, colour, owner string) error {
     car := Car{
          Make:   make,
          Model:  model,
          Colour: colour,
          Owner:  owner,
     }

     carAsBytes, _ := json.Marshal(car)
     return ctx.GetStub().PutState(carNumber, carAsBytes)
}
```

- `PutState` do SDK do Fabric registra a chave e valor
- Validações de dados manuais

Chaincode Exemplo #1: `fabric-samples/fabcar`

Executar uma **transação**:

```
peer chaincode invoke
      -o orderer.example.com:7050
      -C mychannel
      -n fabcar
      -c '{"function":"createCar","Args":["CAR0", "Tesla", "S", "Silver", "Samuel"]}'
      --waitForEvent
      --tls
      --cafile
/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.exa
mple.com/msp/tlscacerts/tlsca.example.com-cert.pem
      --peerAddresses peer0.org1.example.com:7051
      --tlsRootCertFiles
/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.e
xample.com/tls/ca.crt
```

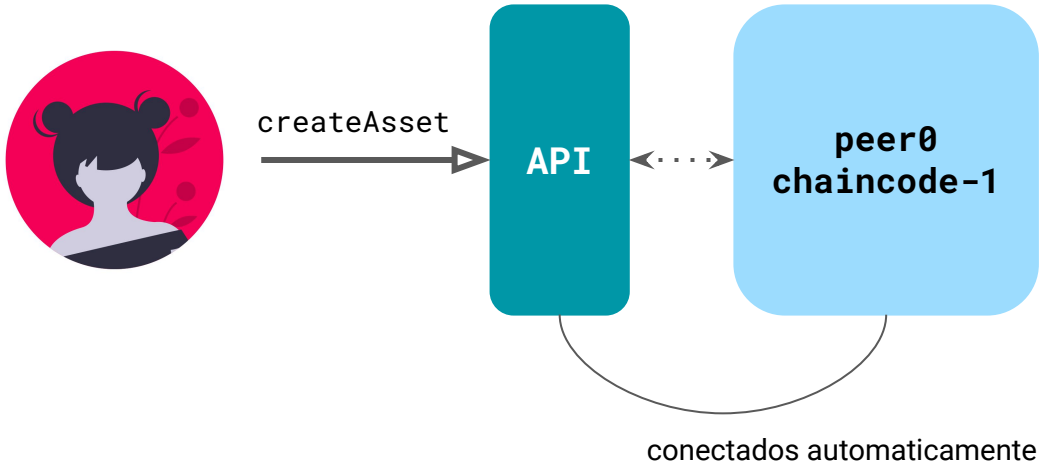
- Execução via *Command Line Interface* (`cli`)
- Argumentos são posicionais
- Acesso aos certificados digitais com autorização

Biblioteca GoLedger CC-Tools

- Padronização de **ativos**, **chaves** e **referências de ativos** (ativo dentro de ativo)
- **Tipos de dados** padrão e customizáveis
- Gerenciamento de **organizações** (MSP)
- Padronização de **Transações**
- Gerenciamento de **permissões de escrita** por propriedade de ativos por MSP
- Gerenciamento de **private data**

Biblioteca GoLedger CC-Tools

- Transações embutidas:
  - Create
  - Read
  - Update
  - Delete
  - Search com paginação
  - Read Asset History



- Integração das transações com a Rest API (GET, PUT, POST, DELETE)

Chaincode Exemplo #2: `cc-tools/gofabcar-cc`

Definição de um **ativo**:

- Definição de chaves
- Referência para um novo ativo *person*
- Validação de parâmetros obrigatórios

disponível no GitHub

```
var Car = assets.AssetType{
	Tag:         "car",
	Label:       "Car",
	Description: "Registry of a Car",

	Props: []assets.AssetProp{
		{
			// Primary key
			Required: true,
			IsKey:    true,
			Tag:      "make",
			Label:    "Make",
			DataType: "string",
		},
		{
			// Primary key
			Required: true,
			IsKey:    true,
			Tag:      "model",
			Label:    "Model",
			DataType: "string",
		},
		{
			Required: true,
			Tag:      "colour",
			Label:    "Colour",
			DataType: "string",
		},
		{
			Tag:      "owner",
			Label:    "Owner",
			DataType: "->person",
		},
	},
}
```

Chaincode Exemplo #2: `cc-tools/gofabcar-cc`

Definição de uma **transação**:

- Definição de argumentos obrigatórios
- Verificação automática de tipos
- Adição de verificações
- Funções de alto nível do cc-tools
  - PutNew
  - Update

```
var CreateNewCar = tx.Transaction{
	Tag:         "createNewCar",
	Label:       "Create New Car",
	Description: "Create New Car",
	Method:      "POST",
	Args: []tx.Argument{
		{
			Required: true,
			Tag:      "make",
			Label:    "Make",
			DataType: "string",
		},
		{
			Required: true,
			Tag:      "model",
			Label:    "Model",
			DataType: "string",
		},
		{
			Required: true,
			Tag:      "colour",
			Label:    "Colour",
			DataType: "string",
		},
		{
			Tag:      "owner",
			Label:    "Owner",
			DataType: "->person",
		},
	},
	Routine: func(stub *sw, req map[string]interface{}) (...){
		...
		return carJSON, nil
	},
}
```

Chaincode Exemplo #2: `cc-tools/gofabcar-cc`

Executar uma **transação**:

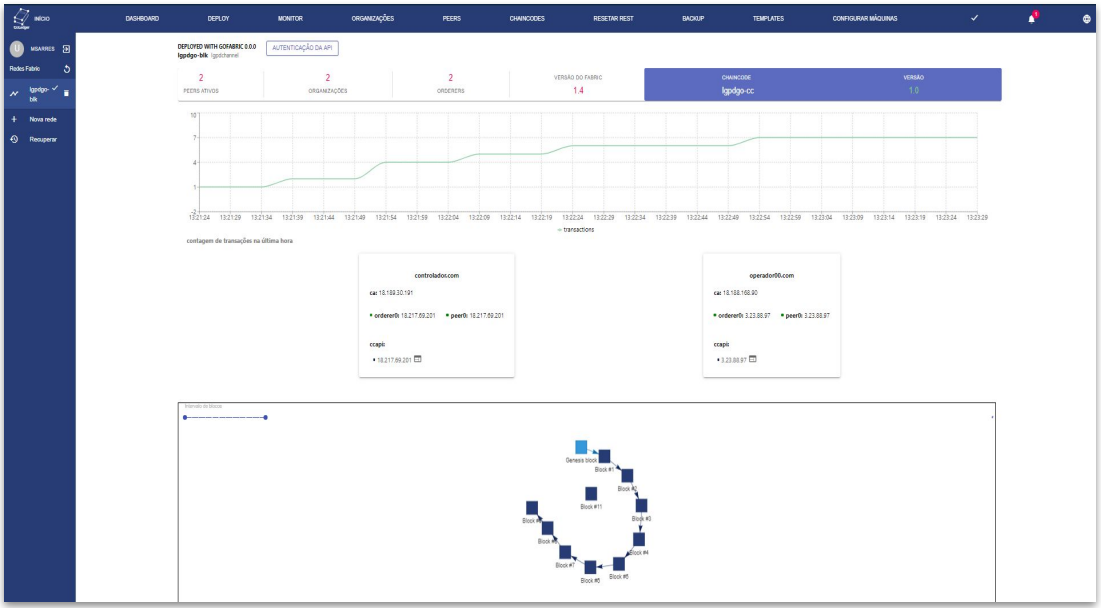
```
curl -X POST "http://localhost:80/api/invoke/createNewCar" \
    -H 'Content-Type: application/json' -H 'cache-control: no-cache'  \
    -d '{"make":"Tesla","model":"S","colour":"Silver"}'
```

- Execução via *API* já disponível
- Argumentos mapeados em um corpo **JSON**
- API possui certificados de autorização, permite auth e customização
- Opção de executar via Interface Web

Demo

Do ambiente de desenvolvimento para produção

GoFabric: orquestrador Hyperledger Fabric/APIs



Demo 2

Samuel Venzi
CTO
samuel.venzi@goledger.com.br

Marcos Sarres
CEO
marcos.sarres@goledger.com.br

GoLedger

faça parte do nosso Discord!

goledger.com.br

contato@goledger.com.br

https://medium.com/@goledger

http://linkedin.com/company/goledger

https://www.youtube.com/goledger