

Pré-requisitos

Certifique-se de ter as seguintes ferramentas instaladas em sua máquina:

- **Docker** e **Docker Compose** (Essencial para o Blockchain e o Banco de Dados)
- **Python 3.10+** e **pip** (Para o Backend e o Módulo de IA)
- **Node.js** e **npm/yarn** (Para o Frontend Mobile)
- **Go 1.18+** (Para compilar o Chaincode do Hyperledger)
- **Hyperledger Fabric Binaries** (Para gerenciar a rede Blockchain)

Passo a Passo de Execução

Assuma que você está no diretório raiz do projeto (nexus_cidadao/).

1. Configuração e Inicialização do Blockchain (Camada de Confiança)

O Blockchain é o primeiro componente a ser inicializado, pois ele é a base de confiança do sistema.

Passo	Comando Sugerido	Descrição
1.1. Setup da Rede	<code>cd blockchain_network/</code>	Navegue para o diretório da rede.
1.2. Inicialização	<code>./start_network.sh</code>	Script que levanta os <i>peers</i> , a <i>Orderer</i> e o <i>Certificate Authority</i> (CA) via Docker Compose.
1.3. Deploy do Chaincode	<code>peer chaincode install -n nexuscc -v 1.0 -p ./chaincode/consensus.go</code>	Instala o <i>Smart Contract</i> (<u>consensus.go</u>) na rede.
1.4. Instanciação	<code>peer chaincode instantiate -n nexuscc -v 1.0 -C mychannel -c '{"Args":["init"]}'</code>	Instancia o Chaincode, tornando-o pronto para receber transações.

2. Configuração e Execução do Backend (API Gateway e Bancos de Dados)

O Backend orquestra a comunicação entre o Frontend, a IA e o Blockchain.

Passo	Comando Sugerido	Descrição
2.1. Setup do Ambiente	<code>cd ../backend/api_gateway/</code>	Navegue para o diretório do Backend.
2.2. Bancos de Dados	<code><u>docker-compose up -d postgres neo4j</u></code>	Levanta os containers do PostgreSQL e Neo4j (bancos de dados de suporte).
2.3. Dependências Python	<code><u>pip install -r requirements.txt</u></code>	Instala as dependências (FastAPI, bibliotecas de Blockchain SDK, etc.).
2.4. Execução da API	<code><u>uvicorn main:app --reload</u></code>	Inicia o servidor da API Gateway em modo de desenvolvimento (geralmente em <u>http://127.0.0.1:8000</u>).

3. Configuração e Execução do Módulo de IA (Análise de Intenção)

O Módulo de IA é um microserviço separado, acessado pelo Backend via API.

Passo	Comando Sugerido	Descrição
3.1. Setup do Ambiente	<code><u>cd ../../ia_module/pln_model/</u></code>	Navegue para o diretório da IA.
3.2. Dependências Python	<code><u>pip install -r requirements.txt</u></code>	Instala as dependências de Machine Learning (TensorFlow, PyTorch, Transformers).
3.3. Treinamento (Simulado)	<code><u>python train.py</u></code>	Executa o script de treinamento do modelo de Intenção (em um ambiente real, isso seria um processo demorado).
3.4. Execução da API de IA	<code><u>uvicorn ia_api:app --port 8001</u></code>	Inicia a API de inferência da IA (em <u>http://127.0.0.1:8001</u>).

4. Configuração e Execução do Frontend (Aplicativo Cidadão)

O Frontend é a interface de interação com o usuário.

Passo	Comando Sugerido	Descrição
4.1. Setup do Ambiente	<code>cd ../../frontend/app_cidadao/</code>	Navegue para o diretório do Frontend.
4.2. Dependências Node	<code>npm install</code> ou <code>yarn install</code>	Instala as dependências do React Native/Flutter.
4.3. Execução	<code>npm run android</code> ou <code>npm run ios</code>	Inicia o aplicativo em um emulador ou dispositivo conectado.

Teste de Integração (Fluxo Básico)

- 1 Verifique se todos os serviços estão rodando (Blockchain, Backend, IA).
- 2 Abra o aplicativo móvel (Frontend).
- 3 Submeta um texto de teste na tela "Verificar Notícia".
- 4 Monitore os logs do **Backend (API Gateway)** para confirmar que a requisição foi:
 - Enviada para a **API de IA** (Passo 2.4).
 - Enviada para o **Blockchain** (Passo 1.4) para iniciar o consenso.
- 5 Simule o voto de um verificador (usando um *script* de teste ou uma interface de administração).
- 6 Verifique se o veredito final aparece no aplicativo móvel.

Este guia fornece a estrutura técnica necessária para que um desenvolvedor possa iniciar o trabalho de implementação e teste do **Nexus Cidadão**.