

**Guia de Implantação**

Versão 1

**IFSP\_Anote**

Histórico de Revisões

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| *12/11/2025* | *3.0* | *Preenchimento Inicial* | *Sahel Luis* |
| *26/11/2025* | *5.0* | *Correção e atualização* | *Gabriel Timpone* |

SUMÁRIO

[1. Introdução 4](#_7rko24t1xbz2)

[1.1. Referências 4](#_aa8yi0qofkre)

[2. Planejamento de Implantação 4](#_it7cqudp5ssl)

[2.1. Responsabilidades 4](#_ahpwimadhe5)

[2.2. Cronograma 4](#_bd78eca8e8kz)

[3. Recursos NECESSÁRIOS PARA IMPLANTAR O PRODUTO 4](#_rqtficlnu8c9)

[3.1. Ambiente 4](#_iznjfx48nqil)

[3.2. Hardware 4](#_1u4lvsci9x7k)

[3.3. Software 4](#_v0rj9giriare)

[3.4. Pessoas 5](#_ehsp47uwvuda)

[4. ROTEIRO DE implantação 5](#_mvkrzh2bpcrg)

[5. Treinamento e suporte 5](#_htfya7xsoelj)

# Introdução

Este documento tem como objetivo descrever o processo de implantação do Bloco de Notas Virtual, um sistema web voltado ao registro, organização e gerenciamento de anotações em ambiente digital.

O guia detalha as etapas necessárias para implantação, desde a preparação do ambiente até o treinamento dos usuários finais, incluindo recursos técnicos, responsabilidades, cronograma e suporte pós-implantação.

O produto foi desenvolvido utilizando React no front-end, Spring Boot (Java) no back-end, MariaDB como banco de dados relacional, Docker para conteinerização e orquestração, e hospedagem em nuvem AWS.

## 1.1 Referências

· Documentação Amazon Linux 2023 – [https://aws.amazon.com/linux/](https://aws.amazon.com/linux/" \t "_new)

· Documentação Docker – [https://docs.docker.com/](https://docs.docker.com/" \t "_new)

· Documentação MariaDB – [https://mariadb.org/](https://mariadb.org/" \t "_new)

· Documentação Java Corretto – [https://docs.aws.amazon.com/corretto/](https://docs.aws.amazon.com/corretto/" \t "_new)

· Documentação Spring Boot – [https://spring.io/projects/spring-boot](https://spring.io/projects/spring-boot" \t "_new)

· Documentação React – [https://react.dev/](https://react.dev/" \t "_new)

· Documentação Vite – [https://vitejs.dev/](https://vitejs.dev/" \t "_new)

· Documentação Nginx – [https://nginx.org/](https://nginx.org/" \t "_new)

# Planejamento de Implantação

· Criação e acesso inicial à instância EC2 (Amazon Linux 2023).

· Instalação do Docker e execução do container MariaDB.

· Instalação do Java Corretto 21 para rodar o backend Spring Boot.

· Configuração e build do backend com Gradle 8.14.

· Instalação do Node.js e NPM para build do frontend.

· Configuração do Nginx para servir o frontend em produção.

· Configuração da API acessível pelo frontend por IP público.

· Testes finais de comunicação API ↔ Frontend.

· Documentação final do ambiente.

## 2.1 Responsabilidades

|  |  |
| --- | --- |
| Responsável | *Atividade* |
| Gabriel | *Preparar o ambiente, realizar testes de integração e empacotar o produto para implantação.* |
| Aline | *Configurar servidores, containers Docker e ambiente de nuvem AWS.* |
| Gabriel | *Executar o processo de instalação e garantir que todos os componentes estejam operando corretamente.* |
| Pedro | *Validar o funcionamento do sistema após a implantação.* |

## 2.2 Cronograma

|  |  |  |
| --- | --- | --- |
| Etapa | Responsavel | Prazo |
| Criação da EC2 e acesso SSH | Cliente | 20/11/2025 |
| Instalação do Docker + MariaDB | Gabriel | 21/11/2025 |
| Instalação do Java + Gradle | Gabriel | 21/11/2025 |
| Deploy Backend | Gabriel | 21/11/2025 |
| Instalação Node + Deploy Frontend | Gabriel | 21/11/2025 |
| Configuração Nginx | Aline | 23/11/2025 |
| Testes finais e ajustes API\_URL | Pedro | 24/11/2025 |

# Recursos Necessários para implantar o produto *3.1. Ambiente*

· Instância EC2 rodando **Amazon Linux 2023**

· Acesso SSH com chave .pem

· Porta 80 liberada (HTTP)

· Porta 8080 liberada (Backend)

· Porta 3307 liberada internamente para banco

### 3.2. Hardware

Uma instância simples atende a aplicação:

### ****Configuração utilizada****

Tipo: t2.micro / t3.micro (1 vCPU, 1GB RAM)

Armazenamento: 8–16 GB SSD

Processador: x86\_64

Fabricante: AWS

### 3.3. Software

| **Software** | **Versão** | **Origem** |
| --- | --- | --- |
| Amazon Linux | 2023.9.20251117 | AWS |
| Docker | 25.0.13 | docker.com |
| MariaDB | 12.1.2 (via Docker) | MariaDB Foundation |
| Java | Corretto 21.0.9 LTS | AWS |
| Gradle | 8.14 | Gradle |
| Node.js | 20.19.5 | Node Foundation |
| npm | 10.8.2 | Node Foundation |
| Nginx | 1.28.0 | Nginx.org |
| Git | Última versão do repositório | AWS |

# ROTEIRO DE implantação

## ****Atualização do sistema****

sudo dnf update -y

### ****Instalar e configurar Git / clonar repositório****

sudo dnf install git -y

git --version#

git config --global user.name "Seu Nome"

git config --global user.email ["seu@email.com"](mailto:\"seu@email.com\")

cd ~

git clone [https://gitserver/GaTimpone/IFSP\_Anote---Back.git](https://gitserver/seu-usuario/seu-repo.git) api

cd ~

git clone [https://gitserver/GaTimpone/IFSP\_Anote---Front.git](https://gitserver/seu-usuario/seu-repo.git) frontend

## ****Instalação do Docker****

sudo dnf install docker -y

sudo systemctl enable docker

sudo systemctl start docker

sudo usermod -aG docker ec2-user

## ****Banco MariaDB via Docker****

Docker pull mariadb

## **docker run -d --name db -e MYSQL\_ROOT\_PASSWORD=ifsp -v mariadb\_data:/var/lib/mysql -p 3307:3306 mariadb**

Verificar:

docker ps -a

## ****Instalar Java 21 (Corretto)****

sudo dnf install java-21-amazon-corretto -y

java -version

## ****Build do Backend (Spring Boot)****

cd ~/api

./gradlew clean build

Executar:

java -jar build/libs/app-0.0.1-SNAPSHOT.jar

## ****Instalação do Node.js e NPM****

sudo dnf install nodejs npm -y

node -v

npm -v

## ****Build do Frontend****

cd ~/frontend

npm install

npm run build

## ****Instalação e configuração do Nginx****

Copiar build:

sudo rm -rf /usr/share/nginx/html/\*

sudo cp -r dist/\* /usr/share/nginx/html/

Reiniciar:

sudo systemctl restart nginx

Arquivo nginx.conf utilizado:

server {

listen 80;

server\_name \_;

root /usr/share/nginx/html;

index index.html;

location / {

try\_files $uri /index.html;

}

location /api/ {

proxy\_pass http://localhost:8080/;

}

}

## ****Testes de Comunicação****

### Testar API local no servidor:

curl http://localhost:8080/usuarios

### Testar do navegador:

http://SEU\_IP\_PUBLICO/

# Treinamento e suporte

### Treinamento

Será fornecido um treinamento básico aos usuários finais, abrangendo:

* Acesso ao sistema e login;
* Criação, edição e exclusão de notas;
* Organização por categorias;
* Salvamento e sincronização automática;
* Boas práticas de uso.

O treinamento pode ser realizado de forma **presencial** ou **remota**, com material de apoio em formato PDF ou vídeo demonstrativo.

### Suporte

Após a implantação, a equipe técnica oferecerá suporte para:

* Correção de eventuais falhas técnicas;
* Monitoramento de logs e desempenho;
* Atualizações e manutenções programadas;
* Apoio aos usuários no uso da ferramenta.

O suporte inicial será mantido durante o período de **monitoramento pós-implantação** (entre 7 e 14 dias), podendo ser estendido conforme contrato.