

Como executar localmente — Frontend + Backend (Point Tils)

Este documento descreve, passo a passo, como um desenvolvedor novo deve configurar e executar o frontend (Expo / React Native) e o backend (Java / Spring Boot) localmente, incluindo todas as variáveis de ambiente e *secrets* usadas pelos dois repositórios.

Observação de segurança: nunca comite secrets reais no repositório. Use `.env` locais (`.gitignore`) em desenvolvimento e um Secret Manager / GitHub Secrets para ambientes remotos.

Resumo rápido (comandos principais)

- Backend (com Docker Compose):
 - `git clone https://github.com/PointTils/Backend.git`
 - `cd Backend`
 - `cp .env.example .env` (preencher valores)
 - `docker-compose up --build | docker-compose-dev up --build | docker-compose-prod up --build`
- Frontend (Expo):
 - `git clone https://github.com/PointTils/Frontend.git`
 - `cd Frontend`
 - `cp .env.example .env` (preencher valores)
 - `npm install` ou `yarn`
 - `npx expo start`

Pré-requisitos (máquina de desenvolvimento)

- Git
- Docker & Docker Compose (para rodar Backend + PostgreSQL + Prometheus + Grafana + Sonarqube em container) (em produção recomendamos utilizar Kubernetes)
- Java 17+ e Maven (opcional se não for usar Docker)
- Node.js 18+ e Yarn ou npm
- Expo CLI (opcional): `npm i -g expo-cli` (ou usar `npx`)
- Dispositivo com Expo Go

Backend (PointTils/Backend)

1) Clonar e preparar

```
git clone https://github.com/PointTils/Backend.git
cd Backend
```

2) Variáveis de ambiente necessárias (lista completa com valores atuais/configurações)

Abaixo estão as variáveis atualmente usadas/configuradas no projeto. Valores sensíveis devem ser mantidos como secrets.

- ADMIN_EMAIL
 - `point.tils.ages@gmail.com`
- BREVO_SENDER_EMAIL
 - `point.tils.ages@gmail.com`
- BREVO_SENDER_NAME
 - `PointTils`
- BREVO_SMTP_HOST
 - `smtp.sendgrid.net`
- BREVO_SMTP_PORT
 - `587`
- CLIENT_IBGE_CITY_URL
 - `https://servicodados.ibge.gov.br/api/v1/localidades/estados/{state}/municipios`
- JWT_EXPIRATION_TIME
 - `3600000`
- JWT_REFRESH_EXPIRATION_TIME
 - `86400000`
- SERVER_PORT
 - `8080`
- SPRINGDOC_API_DOCS_ENABLED
 - `true`
- SPRINGDOC_SWAGGER_UI_ENABLED
 - `true`
- SPRINGDOC_SWAGGER_UI_PATH

- /swagger-ui.html
- SPRING_DATASOURCE_URL
 - jdbc:postgresql://pointtils-db:5432/pointtils-db
- SPRING_FLYWAY_BASELINE_ON_MIGRATE
 - true
- SPRING_FLYWAY_ENABLED
 - true
- SPRING_FLYWAY_LOCATIONS
 - classpath:db/migration
- SPRING_FLYWAY_VALIDATE_ON_MIGRATE
 - true
- SPRING_JPA_HIBERNATE_DDL_AUTO
 - validate
- SPRING_JPA_SHOW_SQL
 - true

Secrets

- API_DEV_BASE_URL
- API_PRD_BASE_URL
- AWS_ACCESS_KEY_ID
- AWS_ACCOUNT_ID
- AWS_REGION
- AWS_SECRET_ACCESS_KEY
- BREVO_SMTP_PASSWORD
- BREVO_SMTP_USERNAME
- CLOUD_AWS_BUCKET_NAME
- DB_NAME
- DB_PASSWORD
- DB_USERNAME
- DEADLINE_TOKEN
- DISCORD_DEADLINE_WEBHOOK
- DISCORD_PR_WEBHOOK
- DISCORD_USERS
- ECR_REPOSITORY
- FIREBASE_SERVICE_ACCOUNT
- GITLAB_PASSWORD
- GITLAB_TOKEN
- GITLAB_USERNAME
- JWT_SECRET
- POSTGRES_PASSWORD
- SONAR_TOKEN
- SSH_PRIVATE_KEY
- SSH_PUBLIC_KEY

Além das variáveis acima, localmente você deve fornecer:

- SPRING_DATASOURCE_USERNAME
- SPRING_DATASOURCE_PASSWORD
- JWT_SECRET (valor forte; obrigatório)
- SPRING_PROFILES_ACTIVE (ex.: dev)

3) Exemplo de `.env.example` (Backend)

Copie para `.env` e preencha os valores sensíveis:

```
# Aplicação
SPRING_PROFILES_ACTIVE=dev
SERVER_PORT=8080
SPRING_APPLICATION_NAME=pointtils-backend

# Database (ajuste conforme seu ambiente)
SPRING_DATASOURCE_URL=jdbc:postgresql://pointtils-db:5432/pointtils-db
SPRING_DATASOURCE_USERNAME=postgres
SPRING_DATASOURCE_PASSWORD=changeme

# Flyway
SPRING_FLYWAY_ENABLED=true
SPRING_FLYWAY_LOCATIONS=classpath:db/migration
SPRING_FLYWAY_BASELINE_ON_MIGRATE=true
SPRING_FLYWAY_VALIDATE_ON_MIGRATE=true

# JWT
JWT_SECRET=please_change_this_to_a_strong_secret
JWT_EXPIRATION_TIME=3600000
JWT_REFRESH_EXPIRATION_TIME=86400000

# Springdoc / Swagger
SPRINGDOC_API_DOCS_ENABLED=true
SPRINGDOC_SWAGGER_UI_ENABLED=true
SPRINGDOC_SWAGGER_UI_PATH=/swagger-ui.html

# Email (BREVO / SendGrid)
BREVO_SENDER_EMAIL=teste@gmail.com
BREVO_SENDER_NAME=PointTils
BREVO_SMTP_HOST=smtp.sendgrid.net
BREVO_SMTP_PORT=587
ADMIN_EMAIL=teste@gmail.com

# Integrações externas
CLIENT_IBGE_CITY_URL=https://servicodados.ibge.gov.br/api/v1/localidades/estados/{state}/municipios
```

4) Rodando com Docker Compose (recomendado)

Executar:

```
docker-compose-dev up --build
```

6) Testes e docs

- Swagger: <http://localhost:8080/swagger-ui.html> (se habilitado)
- Health: /actuator/health (se actuator configurado)

Frontend (PointTils/Frontend)

1) Clonar e preparar

```
git clone https://github.com/PointTils/Frontend.git
cd Frontend
```

2) Instalar dependências

Usando npm:

```
npm install
npx expo prebuild # somente se necessário
npx expo start
```

Ou usando Yarn:

```
yarn
yarn expo start
```

3) Variáveis de ambiente e *secrets* (Frontend)

Lista de `secrets` esperadas pelo GitHub Actions no frontend

- DISCORD_PR_WEBHOOK
- DISCORD_RELEASE_WEBHOOK
- EXPO_TOKEN
- GITLAB_PASSWORD
- GITLAB_USERNAME
- PERSONAL_TOKEN

4) Exemplo de `.env.example` (Frontend)

```
# API
EXPO_PUBLIC_API_URL=http://<ip_address>:8080/v1 -> backend local
```

Coloque `.env` no `.gitignore`.

5) Executando no dispositivo / emulador

- Expo Go (dispositivo físico): escaneie o QR code exibido pelo `expo start`. Se o backend estiver em localhost, ajuste `EXPO_PUBLIC_API_URL` para URL pública.

Passos combinados (ordem recomendada para primeiro setup)

1. Configurar `.env` do backend (preencher `SPRING_DATASOURCE_*` e `JWT_SECRET`)
2. Subir backend e DB:
 - `docker-compose-dev up --build`
 - ou rodar aplicação diretamente com banco local
3. Acessar Swagger / endpoints para verificar health
4. Configurar `.env` do frontend (definir `EXPO_PUBLIC_API_URL` apontando para backend)
5. `npm install` && `npx expo start`
6. Rodar app no dispositivo e testar autenticação / endpoints

Onde guardar secrets em desenvolvimento e CI

- Local: `.env` `.gitignore`
- CI / GitHub Actions: GitHub Secrets
- Produção: AWS Secrets Manager / Azure KeyVault / GCP Secret Manager / HashiCorp Vault
- Kubernetes: Kubernetes Secrets (integrado a Secrets Manager quando possível)

Contatos e referências rápidas

- Backend repo: <https://github.com/PointTils/Backend>
- Frontend repo: <https://github.com/PointTils/Frontend>
- Swagger (backend local): <http://localhost:8080/swagger-ui.html>
- Grafana (backend rodando): <http://localhost:3001/dashboards>

Documentações Oficiais

- <https://docs.expo.dev/>
- <https://reactnative.dev/docs/getting-started>
- <https://docs.oracle.com/en/java/javase/17/>
- <https://docs.docker.com/>
- <https://docs.spring.io/spring-boot/index.html>
- <https://prometheus.io/docs/introduction/overview/>
- <https://grafana.com/docs/>
- <https://www.postgresql.org/docs/>
- <https://docs.sonarsource.com/sonarqube-cloud/>
- <https://docs.aws.amazon.com/>
- <https://developer.hashicorp.com/terraform/docs>