e você hospedar o protótipo no **Heroku**, seu supervisor poderá acessar facilmente a aplicação através de um link público gerado pela plataforma. Aqui estão os passos detalhados para configurar e compartilhar o acesso:

**1. Como Configurar o Heroku**

**Passo 1: Criar uma Conta no Heroku**

* Acesse o site [Heroku](https://www.heroku.com/) e crie uma conta gratuita (se ainda não tiver uma).

**Passo 2: Instalar a CLI do Heroku**

* Baixe e instale a **Heroku CLI** para gerenciar o projeto via terminal:
  + Download da CLI

**Passo 3: Fazer Login no Heroku pela CLI**

* Abra o terminal e execute:

bash

Copy code

heroku login

* Isso abrirá o navegador para autenticação.

**Passo 4: Crie um Novo App no Heroku**

* Crie um app no Heroku:

bash

Copy code

heroku create <nome-do-app>

* Isso gerará um link público como: https://<nome-do-app>.herokuapp.com/.

**2. Deployment da Aplicação**

**Backend (Spring Boot):**

1. **Empacote o Projeto em um JAR:**
   * Compile o backend usando Maven:

bash

Copy code

mvn clean package

* + Isso gerará um arquivo .jar na pasta target/.

1. **Crie um Arquivo Procfile:**
   * Adicione um arquivo chamado Procfile na raiz do backend com o seguinte conteúdo:

bash

Copy code

web: java -jar target/photomap.jar

1. **Configure Variáveis de Ambiente:**
   * Configure no Heroku as variáveis necessárias, como a conexão ao banco de dados:

bash

Copy code

heroku config:set SPRING\_DATASOURCE\_URL=jdbc:postgresql://<host>:<porta>/<banco>

heroku config:set SPRING\_DATASOURCE\_USERNAME=<usuario>

heroku config:set SPRING\_DATASOURCE\_PASSWORD=<senha>

1. **Faça o Deploy:**
   * Inicialize o Git no projeto (se ainda não estiver configurado):

bash

Copy code

git init

* + Adicione o Heroku como remoto:

bash

Copy code

heroku git:remote -a <nome-do-app>

* + Faça o commit e envie para o Heroku:

bash

Copy code

git add .

git commit -m "Deploy do protótipo"

git push heroku main

**Frontend (React):**

1. **Crie a Build do Frontend:**
   * Compile o projeto React:

bash

Copy code

npm run build

1. **Use um Servidor de Arquivos Estáticos:**
   * Configure o backend para servir os arquivos estáticos da pasta build/.
   * Outra opção: hospedar o frontend separadamente no **Netlify** ou **Vercel** e apontar para o backend no Heroku.

**3. Compartilhando com o Supervisor**

Assim que o deployment estiver concluído, o Heroku fornecerá um link público, por exemplo:

* https://photomap-prototype.herokuapp.com/

Envie esse link ao supervisor para que ele possa acessar a aplicação diretamente pelo navegador.

**4. Monitoramento e Logs**

Para demonstrar controle sobre o sistema:

* Use o comando abaixo para monitorar logs de acesso ou erros:

bash

Copy code

heroku logs --tail

* Mostre que você está preparado para gerenciar e ajustar o protótipo conforme necessário.

**5. Configurações Opcionais para Melhor Experiência**

1. **Banco de Dados (Heroku PostgreSQL):**
   * Use o **addon gratuito Heroku PostgreSQL** para configurar o banco de dados diretamente na plataforma:

bash

Copy code

heroku addons:create heroku-postgresql:hobby-dev

1. **Configurar HTTPS:**
   * O Heroku já oferece HTTPS para todos os aplicativos hospedados.
2. **Criação de Subdomínio:**
   * Se preferir, configure um subdomínio como prototype.photomap.com e aponte para o app no Heroku.

**Conclusão**

Ao hospedar no Heroku, seu supervisor terá acesso ao protótipo apenas com o link público gerado. Isso simplifica o processo de compartilhamento e demonstra que você está utilizando práticas modernas de deployment.

Se precisar de ajuda com a configuração ou o deployment, posso orientar mais detalhadamente!

escolho o Heroku, como eu faco para hospedar minha aplicacao ?

**ChatGPT said:**

ChatGPT

Hospedar sua aplicação no **Heroku** envolve alguns passos simples. Aqui está um guia passo a passo para configurar o Heroku e hospedar seu protótipo.

**1. Preparação Inicial**

1. **Instale a Heroku CLI:**
   * Faça o download da CLI (Command Line Interface) do Heroku:
     + Heroku CLI Download
   * Após instalar, verifique a instalação:

bash

Copy code

heroku --version

1. **Crie uma Conta no Heroku:**
   * Acesse [Heroku](https://www.heroku.com/) e registre-se.
2. **Login na CLI:**
   * No terminal, faça login na sua conta do Heroku:

bash

Copy code

heroku login

* + Isso abrirá o navegador para autenticação.

**2. Configuração do Backend (Spring Boot)**

1. **Compile o Projeto:**
   * Gere o arquivo JAR usando Maven:

bash

Copy code

mvn clean package

* + Isso criará um arquivo photomap.jar (ou nome similar) na pasta target/.

1. **Crie um Arquivo Procfile:**
   * Adicione um arquivo chamado Procfile na raiz do projeto com o seguinte conteúdo:

bash

Copy code

web: java -jar target/photomap.jar

1. **Configure Variáveis de Ambiente:**
   * Certifique-se de que seu application.properties ou application.yml esteja configurado para usar variáveis de ambiente:

properties

Copy code

spring.datasource.url=${SPRING\_DATASOURCE\_URL}

spring.datasource.username=${SPRING\_DATASOURCE\_USERNAME}

spring.datasource.password=${SPRING\_DATASOURCE\_PASSWORD}

* + No Heroku, configure as variáveis usando o comando:

bash

Copy code

heroku config:set SPRING\_DATASOURCE\_URL=jdbc:postgresql://<host>:<porta>/<banco>

heroku config:set SPRING\_DATASOURCE\_USERNAME=<usuario>

heroku config:set SPRING\_DATASOURCE\_PASSWORD=<senha>

**3. Subindo o Backend para o Heroku**

1. **Inicie o Git no Projeto:**
   * Certifique-se de que o projeto esteja versionado com Git:

bash

Copy code

git init

git add .

git commit -m "Deploy inicial do backend"

1. **Crie o App no Heroku:**
   * No terminal, crie o app:

bash

Copy code

heroku create <nome-do-app>

* + Isso criará um link como: https://<nome-do-app>.herokuapp.com.

1. **Envie o Código para o Heroku:**
   * Envie o código para o Heroku usando Git:

bash

Copy code

git push heroku main

1. **Teste o App:**
   * Após o deploy, acesse o link gerado pelo Heroku para testar o backend.

**4. Configuração do Banco de Dados (Heroku PostgreSQL)**

1. **Adicione o Add-on PostgreSQL:**
   * No terminal, adicione o PostgreSQL gratuito ao seu app:

bash

Copy code

heroku addons:create heroku-postgresql:hobby-dev

* + O Heroku configurará automaticamente as variáveis de ambiente DATABASE\_URL.

1. **Ajuste o Backend para Usar DATABASE\_URL:**
   * Atualize o application.properties:

properties

Copy code

spring.datasource.url=${DATABASE\_URL}

1. Copy code
2. **Verifique a Configuração:**
   * Teste a conexão com o banco de dados no Heroku.

**5. Configuração do Frontend (React)**

1. **Crie a Build do React:**
   * No diretório do frontend, compile o projeto:

bash

Copy code

npm run build

1. **Integre o Frontend ao Backend (Opcional):**
   * Sirva os arquivos do React diretamente no backend adicionando:

java

Copy code

@Configuration

public class WebConfig extends WebMvcConfigurer {

@Override

public void addResourceHandlers(ResourceHandlerRegistry registry) {

registry.addResourceHandler("/\*\*")

.addResourceLocations("classpath:/static/");

}

}

1. **Deploy do Frontend no Heroku (Separado):**
   * Use o serve para criar um servidor de arquivos estáticos:

bash

Copy code

npm install -g serve

serve -s build

**6. Teste e Compartilhe o Link**

1. **Acesse o Link do App:**
   * Após o deploy, seu app estará acessível em:

perl

Copy code

https://<nome-do-app>.herokuapp.com

1. **Envie o Link para o Supervisor:**
   * Compartilhe esse link para que o supervisor teste o protótipo.

**7. (Opcional) Logs e Debugging**

* Monitore os logs do Heroku para verificar erros:

bash

Copy code

heroku logs --tail