

Deploy do Back end da API no Heroku

O que é Deploy?

O verbo **deploy**, em inglês, significa **implantar**.

Em programação, seu sentido está intimamente relacionado à sua tradução literal: fazer um deploy, em termos práticos, significa colocar no ar alguma aplicação que teve seu desenvolvimento concluído.

Quando um site é finalizado por um desenvolvedor e, após seus testes, é finalmente hospedado na nuvem e colocado no ar, ele passa pelo processo de deploy.

De mesmo modo, quando um sistema sofre alguma melhoria ou alteração em seu código-fonte, implementar essa alteração ao sistema que está no ar também é um tipo de deploy.

O que veremos por aqui?

Esse documento é um passo a passo para você subir (deploy) a sua API criada no SPRING gratuitamente para a nuvem e isso irá gerar um link de acesso a sua página que poderá ser acessado em qualquer lugar a partir de qualquer dispositivo com acesso a Internet.

Para realizar esse deploy vamos precisar fazer algumas modificações em nosso projeto, que serão detalhadas nas próximas páginas.

#Passo 01 - Criar a Documentação da API no Swagger

Para criar a Documentação da API no Swagger, utilize o ebook do Swagger.

#Passo 02 - Criação do usuário em memória

Vamos criar um usuário padrão em memória para simplificar o acesso a nossa API. O usuário em memória é um usuário para testes, que dispensa o cadastro no Banco de Dados. Quando a API estiver em produção este usuário deve ser desabilitado.

Na camada Security, abra o arquivo **BasicSecurityConfig** e altere o método **protected void configure(AuthenticationManagerBuilder auth) throws Exception** de:

```
@Override
protected void configure(AuthenticationManagerBuilder auth)
throws Exception {
    auth.userDetailsService(userDetailsService);
}
```

Para:

```
@Override
protected void configure(AuthenticationManagerBuilder auth)
throws Exception {
    auth.userDetailsService(userDetailsService);

    auth.inMemoryAuthentication()
        .withUser("root")
        .password(passwordEncoder().encode("root"))
        .authorities("ROLE_USER");
}
```

#Passo 03 - Testar a API no seu computador

1. Execute a sua aplicação localmente pelo Eclipse ou pelo STS
2. Abra o endereço: <http://localhost:8080/> no seu navegador
3. Verifique se o Swagger abre automaticamente
4. Caso a API solicite Usuário e senha, experimente o **Usuário: root** e a **Senha: root**, que foram criados em memória.
5. Aproveite para testar todos os Endpoints da aplicação no Swagger ou no Postman (/postagens, /temas e /usuarios).
6. Antes de continuar pare a execução do Projeto.

IMPORTANTE: Antes de fazer o Deploy a API deve estar funcionando sem erros.

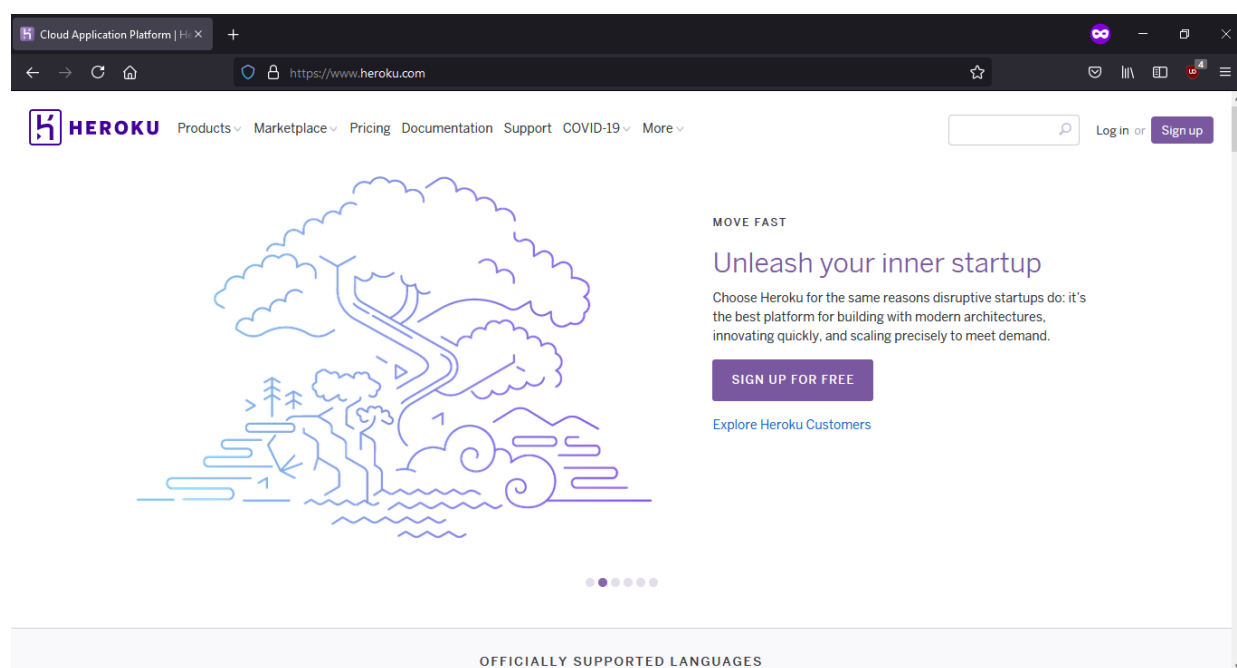
#Passo 04 - Criar uma conta grátis no Heroku

O **Heroku** é uma plataforma na nuvem que permite efetuar o Deploy de aplicações Back End ou Front End seja para hospedagem, testes em produção ou escalar as suas aplicações.

O grande diferencial do Heroku são as contas gratuitas, que permitem implantar até 5 Aplicações na mesma conta com Banco de Dados PostgreSQL incluso.

Para criar a sua conta grátis:

1- Acesse o endereço: <https://www.heroku.com>



2- Crie a sua conta grátis no Heroku clicando no botão **SIGN UP FOR FREE** e siga as instruções.

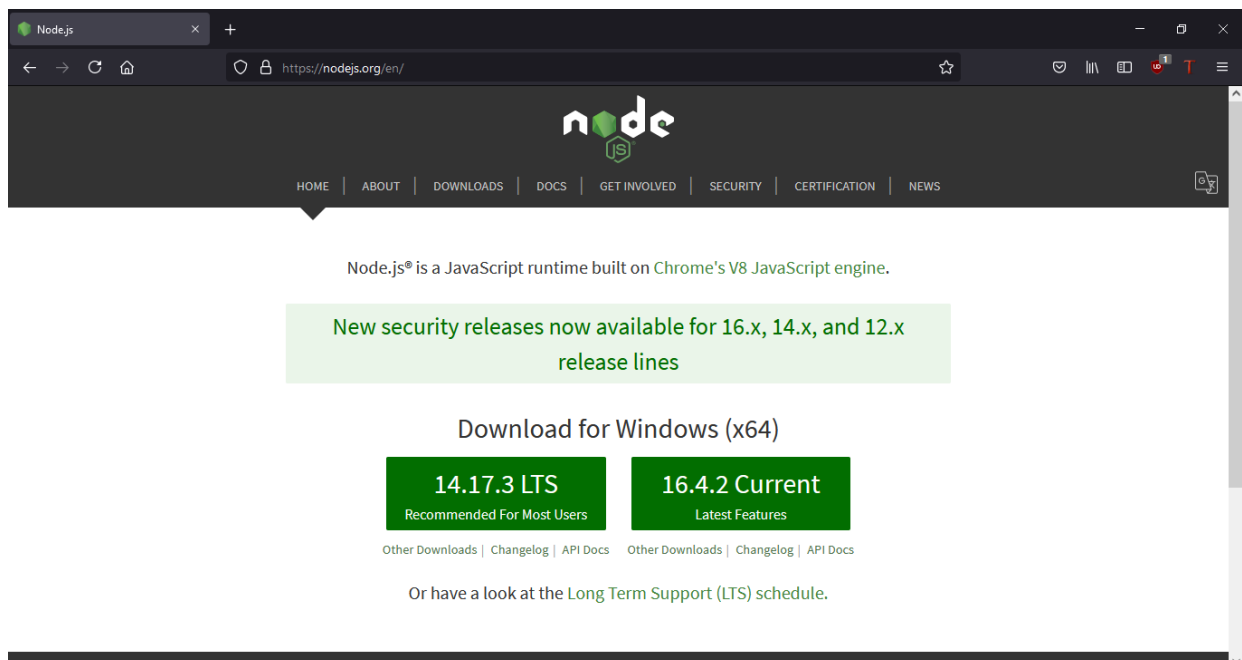
#Passo 05 - Instalação do Node.js

O Node.js pode ser definido como um **ambiente de execução Javascript**. Isso significa que com o Node.js é possível criar aplicações Javascript e executar como uma aplicação local em uma máquina, não dependendo de um navegador, como estamos acostumados.

O **NPM** é um gerenciador de pacotes para a linguagem de programação Javascript, semelhante ao Maven do Java, que permite instalar pacotes Javascript no nosso computador.

Para realizarmos o nosso Deploy utilizaremos o NPM para instalar o **Heroku Client**.

1- Acesse o endereço: <https://nodejs.org/en/>



2- Faça o download da versão LTS do Node.js e instale no seu computador.

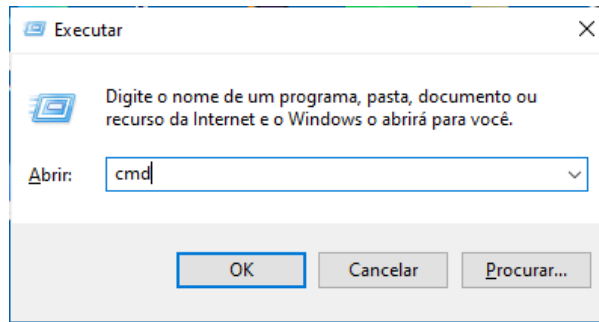
Em caso de dúvidas, acesse o Guia de instalação do Node.js

#Passo 06 - Instalação do Heroku Client

O **Heroku Client** é uma Interface de linha de comando (CLI) do Heroku, que facilita a criação e o gerenciamento de seus aplicativos Heroku diretamente do terminal.

Para instalar e executar os comandos do Heroku Client usaremos o Prompt de comando do Windows.

1- Para instalar, execute o atalho  +  para abrir a janela Executar



2- Digite o comando **cmd** para abrir o **Prompt de comando do Windows**

3- Antes de instalar o **Heroku Client**, verifique se o Node já está instalado através do comando:

```
npm -version
```

```
7.19.1
```

**A versão pode ser diferente da imagem*

4- Para instalar o **Heroku Client** digite o comando:

```
npm i -g heroku
```

```
npm WARN deprecated strip-eof@2.0.0: Renamed to `strip-final-newline` to better represent its functionality.
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js
environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated uuid@3.2.1: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances,
which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated uuid@3.3.2: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances,
which is known to be problematic. See https://v8.dev/blog/math-random for details.

added 13 packages, changed 620 packages, and audited 691 packages in 2m

22 packages are looking for funding
  run `npm fund` for details

7 vulnerabilities (2 moderate, 5 high)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 7.17.0 -> 7.19.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.19.1
npm notice Run npm install -g npm@7.19.1 to update!
npm notice
```

5- Confirme a instalação do Heroku Client através do comando:

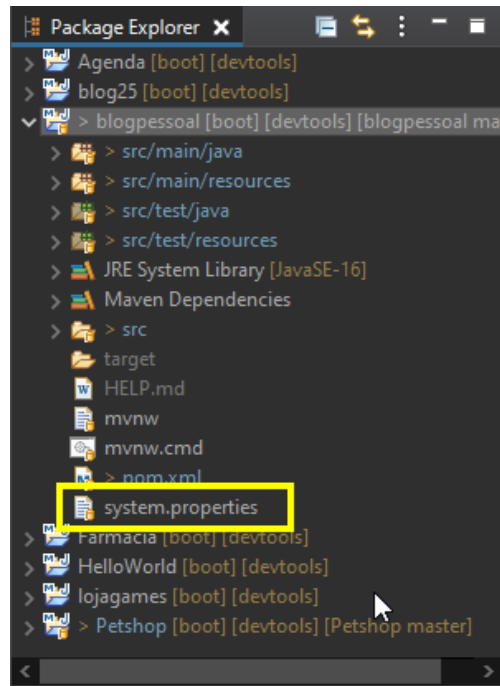
```
heroku version
```

```
heroku/7.56.0 win32-x64 node-v14.17.3
```

**A versão pode ser diferente da imagem*

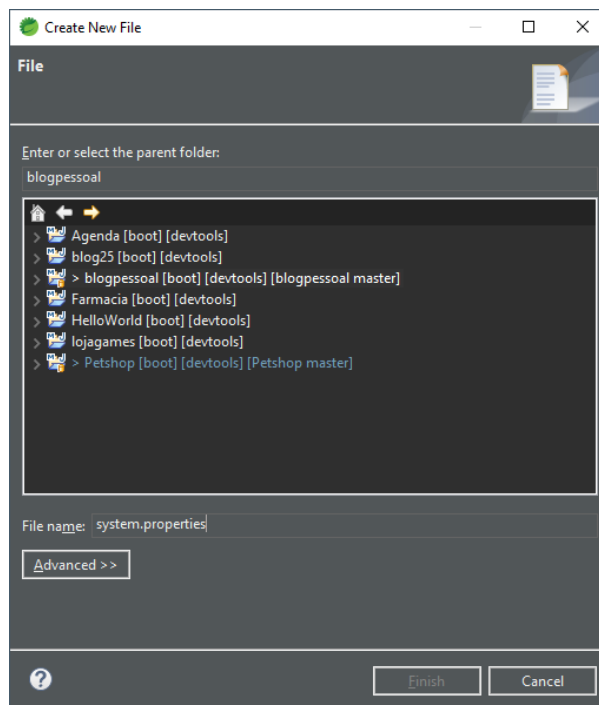
#Passo 07 - Criação do arquivo system.properties

1- Na raiz do seu projeto (na pasta blogpessoal), crie o arquivo **system.properties**.



2- Na Guia **Package explorer**, clique com o botão direito do mouse e clique na opção **New->File**.

3- Em **File name**, digite: **system.properties** e clique no botão **Finish**.



4- No arquivo **system.properties** indique a versão do Java que será utilizada no Heroku:

```
java.runtime.version=16
```

#Passo 08 - Configuração do PostgreSQL no arquivo pom.xml

No arquivo, **pom.xml**, vamos alterar as linhas:

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <scope>runtime</scope>
</dependency>
```

Para:

```
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
</dependency>
```

#Passo 09 - Configuração do Banco de Dados no arquivo application.properties

No arquivo, **application.properties**, vamos alterar as linhas:

```
spring.jpa.hibernate.ddl-auto=update
spring.jpa.database=mysql
spring.datasource.url=jdbc:mysql://localhost/db_blogpessoal?
createDatabaseIfNotExist=true&serverTimezone=America/Sao_Paulo&useSSL=false
spring.datasource.username=root
spring.datasource.password=root
spring.jpa.properties.hibernate.dialect =
org.hibernate.dialect.MySQL8Dialect

spring.jpa.show-sql=true

spring.jackson.date-format=yyyy-MM-dd HH:mm:ss
spring.jackson.time-zone=Brazil/East
```

Observe que a linha 3 do arquivo *application.properties* (nome do banco de dados) está dividida em 2 linhas no pdf. No STS/Eclipse mantenha em uma única linha.

Para:

```
spring.jpa.generate-ddl=true
spring.datasource.url=${JDBC_DATASOURCE_URL}
spring.jpa.show-sql=true

spring.jackson.date-format=yyyy-MM-dd HH:mm:ss
spring.jackson.time-zone=Brazil/East
```

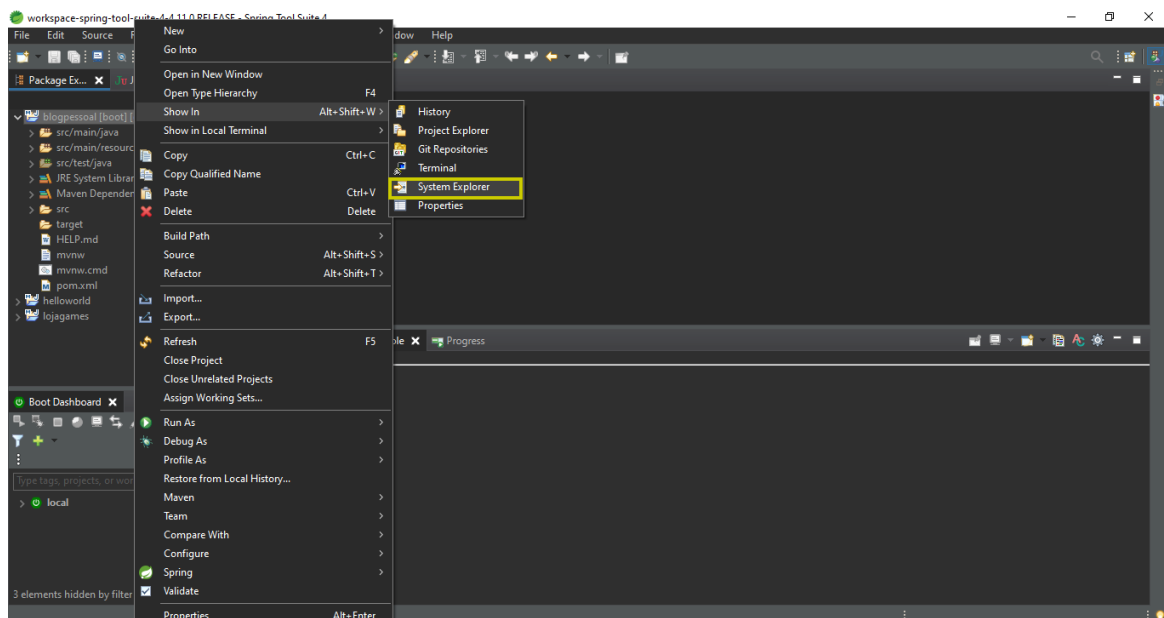
*** Importante ***

A partir deste ponto, **o seu projeto não executará mais localmente (http://localhost:8080/)**. Para voltar a executar localmente, será necessário **desfazer as configurações efetuadas nos arquivos pom.xml e application.properties**.

#Passo 10 - Deploy com o Git

Vamos preparar o nosso repositório local para subir a aplicação para o Heroku utilizando o Git.

1- Na pasta do projeto, clique com o botão direito do mouse e na sequência clique na opção: **Show in => System Explorer**



2- Será aberta a pasta Workspace onde o Eclipse/STS grava os seus projetos:

- Se você estiver usando o STS geralmente a pasta fica em: **c:\Usuarios\seu usuario\Documents\workspace-spring-tool-suite-4-4.11.0.RELEASE** (a versão pode ser diferente).
- Se você estiver utilizando o Eclipse, geralmente a pasta fica em: **c:\Usuarios\seu usuario\eclipse-workspace**.

**seu usuario = Usuário do seu computador*

3- Copie a pasta da API: **blogpessoal** (o nome da sua pasta pode ser diferente)

Nome	Data de modificação	Tipo	Tamanho
.metadata	20/07/2021 00:26	Pasta de arquivos	
blogpessoal	02/08/2021 01:54	Pasta de arquivos	
helloworld	23/07/2021 09:19	Pasta de arquivos	
lojagames	30/07/2021 09:30	Pasta de arquivos	

4- Cole a pasta no mesmo diretório

Nome	Data de modificação	Tipo	Tamanho
.metadata	20/07/2021 00:26	Pasta de arquivos	
blogpessoal	20/07/2021 00:29	Pasta de arquivos	
blogpessoal - Copia	02/08/2021 01:54	Pasta de arquivos	
helloworld	23/07/2021 09:19	Pasta de arquivos	
lojagames	30/07/2021 09:30	Pasta de arquivos	

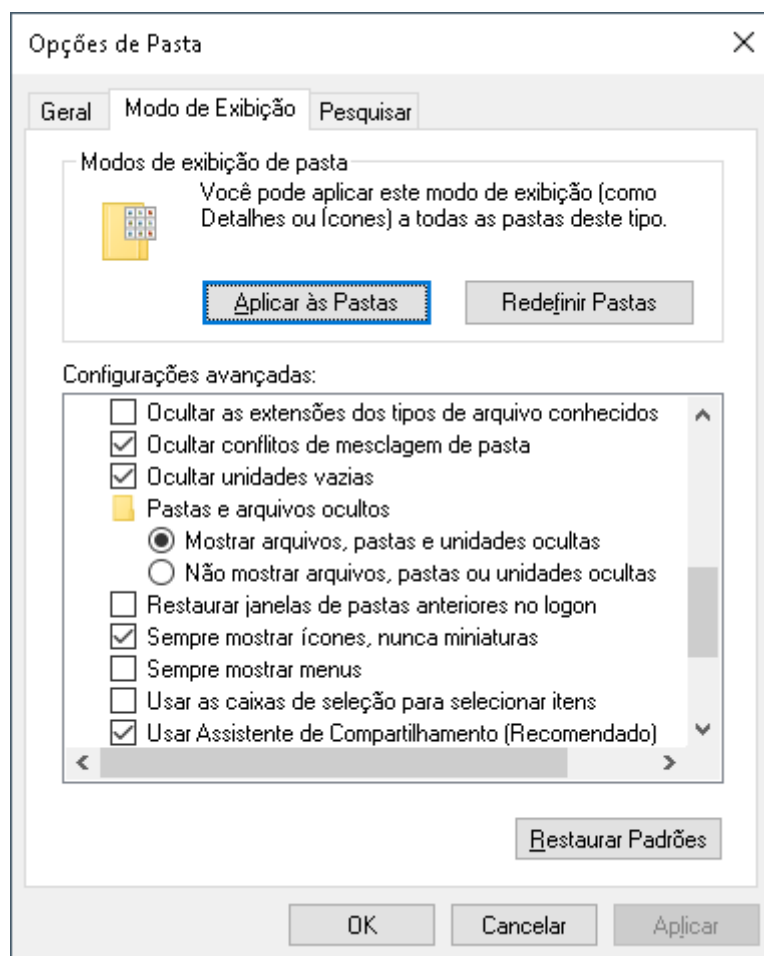
5- Renomeie a pasta para deploy_blogpessoal

Nome	Data de modificação	Tipo	Tamanho
.metadata	20/07/2021 00:26	Pasta de arquivos	
blogpessoal	20/07/2021 00:29	Pasta de arquivos	
deploy_blogpessoal	02/08/2021 01:54	Pasta de arquivos	
helloworld	23/07/2021 09:19	Pasta de arquivos	
lojagames	30/07/2021 09:30	Pasta de arquivos	

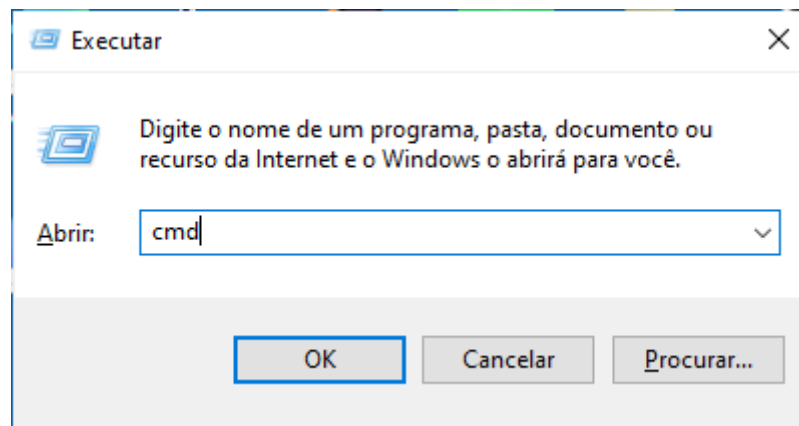
6- Abra esta pasta e verifique se existe uma pasta chamada .git. Caso exista, apague esta pasta. ****Esta pasta estará presente APENAS se você inicializou o git dentro dela.****

Nome	Data de modificação	Tipo	Tamanho
.git	13/07/2021 16:50	Pasta de arquivos	
.mvn	13/07/2021 07:39	Pasta de arquivos	
.settings	13/07/2021 07:39	Pasta de arquivos	
.vscode	13/07/2021 07:39	Pasta de arquivos	
src	13/07/2021 07:39	Pasta de arquivos	
target	13/07/2021 07:39	Pasta de arquivos	
.classpath	13/07/2021 22:27	Arquivo CLASSPA...	3 KB
.gitignore	25/05/2021 22:47	Documento de Te...	1 KB
.project	27/05/2021 18:05	Arquivo PROJECT	1 KB
HELP.md	25/05/2021 22:47	Markdown File	2 KB
mvnw	25/05/2021 22:47	Arquivo	10 KB
mvnw.cmd	25/05/2021 22:47	Script de Comand...	7 KB
pom.xml	13/07/2021 22:27	Documento XML	3 KB
system.properties	13/07/2021 22:26	Arquivo PROPERTI...	1 KB

Caso esta pasta não esteja sendo exibida, na janela do Windows Explorer, clique na **Guia Exibir** e na sequência no botão **Opções**. Na janela **Opções de Pasta**, na **Guia Modo de Exibição**, no item **Configurações avançadas**, localize a opção: **Pastas e arquivos ocultos** e marque a opção **Mostrar arquivos, pastas e unidades ocultas** (como mostra a figura abaixo). Em seguida clique em **OK** para concluir.



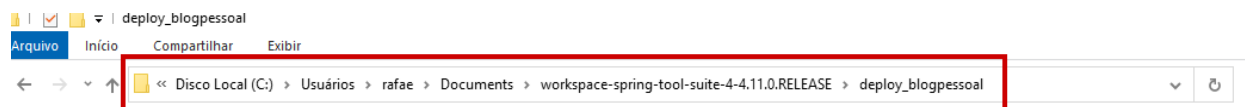
7- Execute o atalho  +  para abrir a janela Executar



8- Digite o comando abaixo para abrir o **Prompt de Comando do Windows**:

```
cmd
```

9- Na pasta do seu projeto, no Windows explorer, copie o caminho da pasta conforme a figura abaixo:



10- No Prompt de comando do Windows digite os comando cd e cole na frente do comando o caminho copiado:

```
cd C:\Users\seu usuario\Documents\  
workspace-spring-tool-suite-4-4.11.0.RELEASE\deploy_blogpessoal
```

**o nome da pasta pode ser diferente*

11- Digite a sequência de comandos abaixo para inicializar o seu repositório local para efetuar o Deploy no Heroku:

```
git init  
git add .  
git commit -m "Deploy inicial - Blog Pessoal"
```

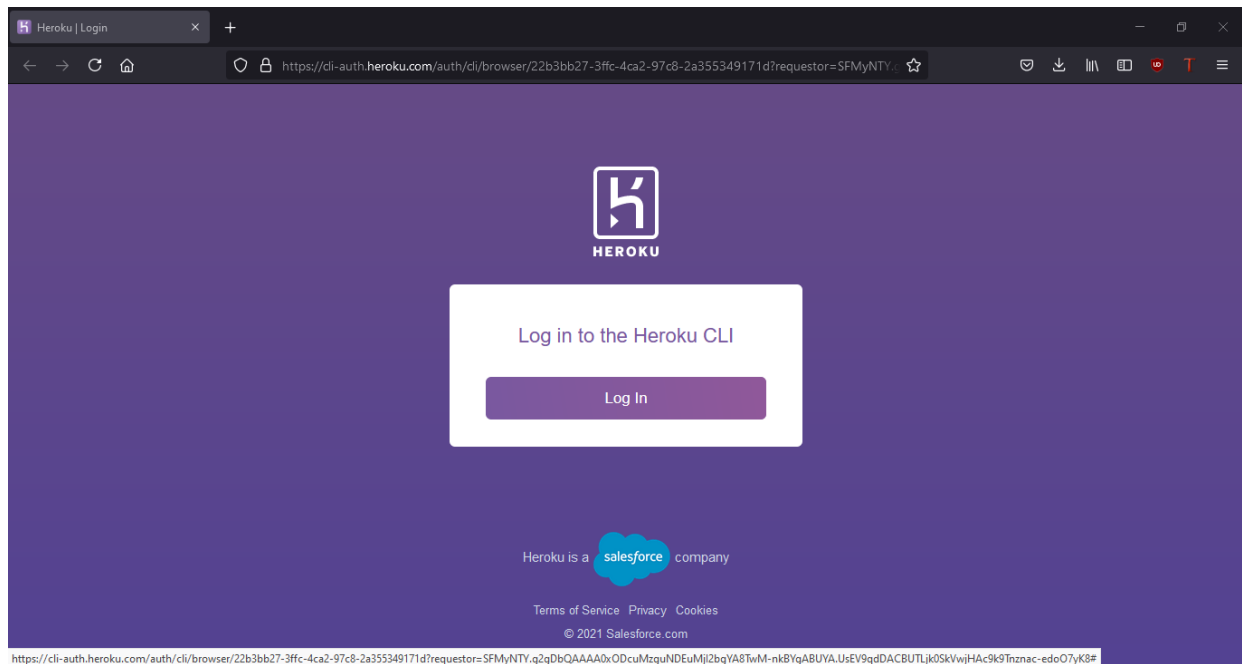
#Passo 11 - Login no Heroku

1- Digite o comando:

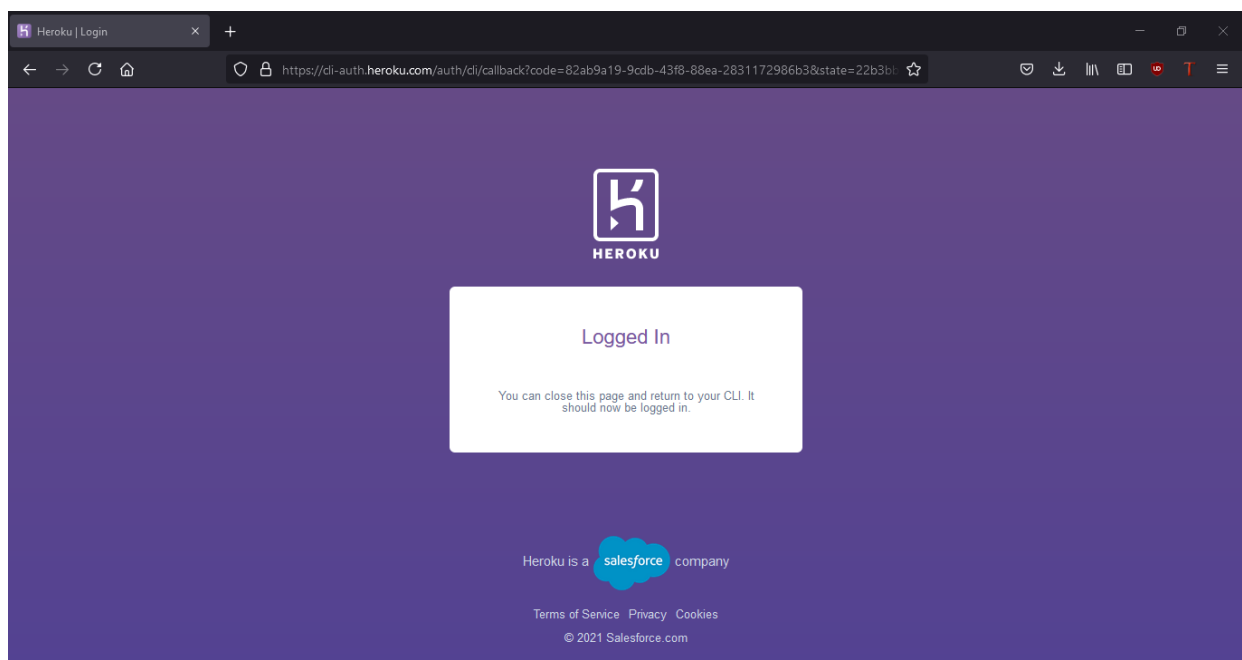
```
heroku login
```

```
>> Warning: Our terms of service have changed: https://dashboard.heroku.com/terms-of-service
heroku: Press any key to open up the browser to login or q to exit:
```

2- Será aberta a janela abaixo. Clique no botão **Log in**



3- Após efetuar o login na sua conta, será exibida a janela abaixo.



4- Volte para o Prompt de comando para continuar o Deploy.

```
» Warning: Our terms of service have changed: https://dashboard.heroku.com/terms-of-service
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/22b3bb27-3ffc-4ca2-97c8-2a355349171d?requestor=SFMyNTY.g
2gDbQAAAA0xODcuMzguNDUuMjI2bgYA8TwM-nkBYgABUYA.USeV9qdDACBUTLjk0SkVwjHAc9k9Tznac-edo07yK8
Logging in... done
Logged in as rafaelpinfo@gmail.com
```

#Passo 12 - Criar um novo projeto no Heroku

Para criar um novo projeto na sua conta do Heroku, digite o comando:

```
heroku create nomedoprojeto
```

***** Importante *****

O NOME DO PROJETO NÃO PODE TER LETRAS MAIUSCULAS, NUMEROS OU CARACTERES ESPECIAIS. ALÉM DISSO ELE PRECISA SER UNICO DENTRO DA PLATAFORMA HEROKU.

Se o nome escolhido já estiver em uso no Heroku, será exibida a mensagem abaixo:

```
Creating ☐ bloggen... !
! Name bloggen is already taken
```

Se o nome escolhido estiver disponível, será exibida a mensagem abaixo:

```
Creating ☐ bprfp... done
https://bprfp.herokuapp.com/ | https://git.heroku.com/bprfp.git
```

#Passo 13 - Adicionar o Banco de dados (PostgreSQL) no Heroku

Para adicionar um Banco de Dados PostgreSQL no seu projeto, digite o comando:

```
heroku addons:create heroku-postgresql:hobby-dev
```

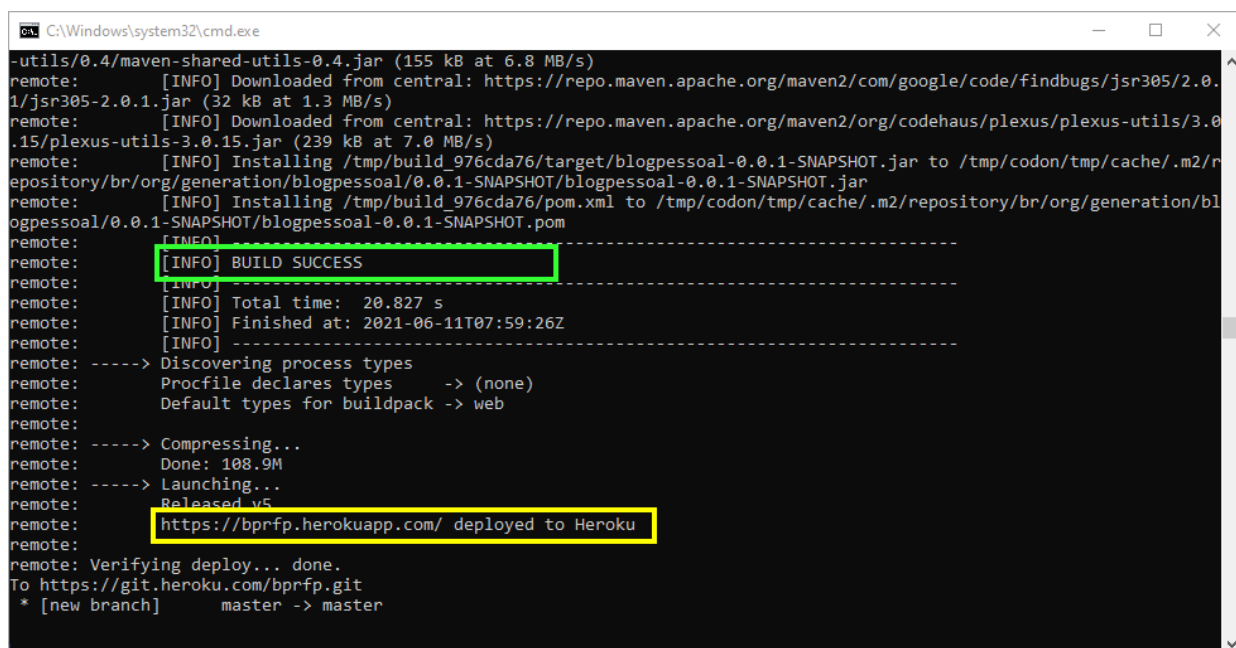
```
Creating heroku-postgresql:hobby-dev on ▯ bprfp... free
Database has been created and is available
! This database is empty. If upgrading, you can transfer
! data from another database with pg:copy
Created postgresql-flexible-10004 as DATABASE_URL
Use heroku addons:docs heroku-postgresql to view documentation
```

#Passo 14 - Efetuar o Deploy

Para concluir o Deploy, digite o comando:

```
git push heroku master
```

Se tudo deu certo, será exibida a mensagem **BUILD SUCESS** (destacado em verde na imagem) e será exibido o endereço (<https://nomedoprojeto.herokuapp.com>) para acessar a API na Internet (destacado em amarelo na imagem)



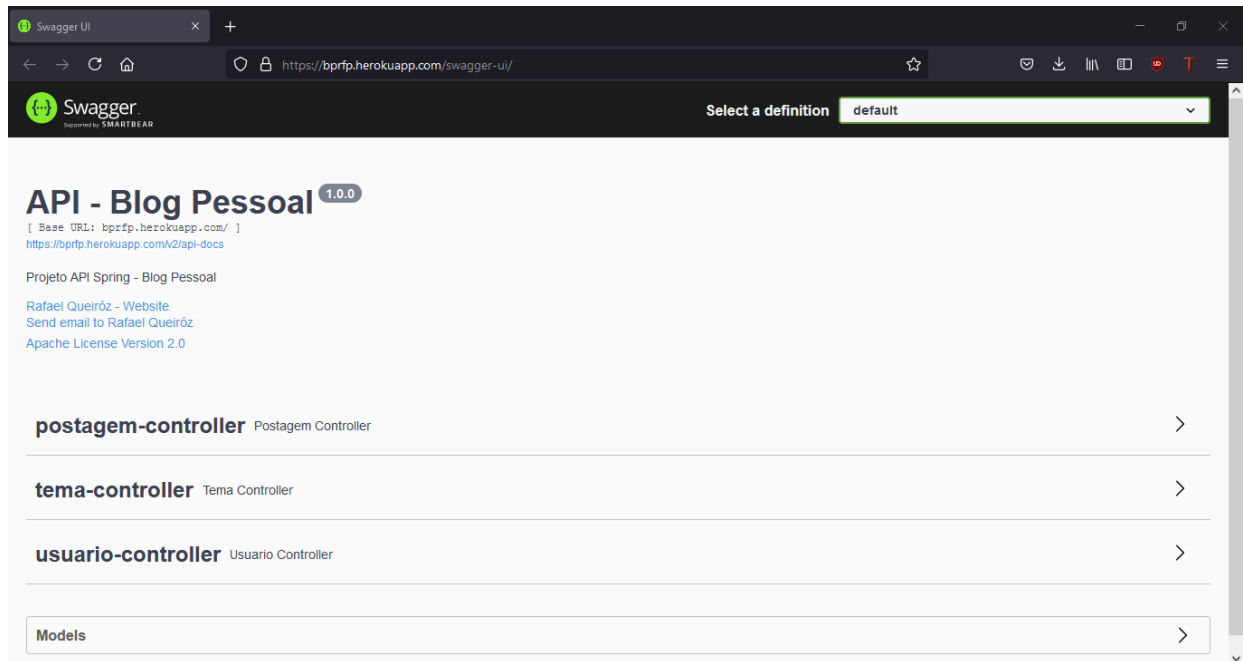
```
C:\Windows\system32\cmd.exe
-utils/0.4/maven-shared-utils-0.4.jar (155 kB at 6.8 MB/s)
remote: [INFO] Downloaded from central: https://repo.maven.apache.org/maven2/com/google/code/findbugs/jsr305/2.0.
1/jsr305-2.0.1.jar (32 kB at 1.3 MB/s)
remote: [INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0
.15/plexus-utils-3.0.15.jar (239 kB at 7.0 MB/s)
remote: [INFO] Installing /tmp/build_976cda76/target/blogpessoal-0.0.1-SNAPSHOT.jar to /tmp/codon/tmp/cache/.m2/r
epository/br/org/generation/blogpessoal/0.0.1-SNAPSHOT/blogpessoal-0.0.1-SNAPSHOT.jar
remote: [INFO] Installing /tmp/build_976cda76/pom.xml to /tmp/codon/tmp/cache/.m2/repository/br/org/generation/bl
ogpessoal/0.0.1-SNAPSHOT/blogpessoal-0.0.1-SNAPSHOT.pom
remote: [INFO] -----
remote: [INFO] BUILD SUCCESS
remote: [INFO] -----
remote: [INFO] Total time: 20.827 s
remote: [INFO] Finished at: 2021-06-11T07:59:26Z
remote: [INFO] -----
remote: ----> Discovering process types
remote: Procfile declares types -> (none)
remote: Default types for buildpack -> web
remote: ----> Compressing...
remote: Done: 108.9M
remote: ----> Launching...
remote: Released v5
remote: https://bprfp.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/bprfp.git
* [new branch] master -> master
```

#Passo 15 - Testar o link e a API

1- Abra o navegador, copie o endereço do Deploy e cole na Barra de endereço do navegador.

2- Será solicitado o usuário e a senha. Digite **root** para ambos.

3- Sua API deverá abrir o Swagger.



4- Faça alguns testes via Swagger para certificar-se de que tudo está funcionando

Atualizar o Deploy no Heroku (Quando houver necessidade)

Uma vez que o Deploy foi feito no Heroku, assim como no Github, basta atualizar os arquivos na pasta **deploy_blogpessoal** e efetuar a sequência de comandos abaixo para subir as atualizações efetuadas no Backend.

```
git add .  
git commit -m "Atualização do Deploy - Blog Pessoal"  
git push heroku master
```