

[ConfigServer-Parte 01] Guia para Configuração Ambientes (dev e prod) e Deploy do Microservice no Heroku

1. Configuração Ambiente **DEV** para ConfigServer Microservice

1. Criar branch local dev: **git checkout -b dev**
2. Renomear o arquivo application.yaml para application-dev.yaml para propriedades exclusivas do ambiente de dev que são diferentes em outros ambientes;
3. Criar outro arquivo application.yaml apenas com configurações de spring.application.name, recortar essa propriedade de application-dev.yaml;
4. No arquivo application.yaml criado no passo 1.3 acima, incluir spring.profiles.active=dev;
5. Fazer o commit das alterações;

2. Configuração Ambiente **PROD** para ConfigServer Microservice

1. Criar branch local prod: **git checkout -b prod**
2. No arquivo application.yaml alterar spring.profiles.active=dev para spring.profiles.active=prod ;
3. Criar arquivo application-prod.yaml dentro do microservice (copiar todas as propriedades de dev inicialmente para alterá-las).
4. Em application-prod.yaml, inserir variáveis de ambiente para produção e também encrypt.key:

```
server:
  port: ${PORT}

spring:
  cloud:
    config:
      server:
        git:
          uri: https://github.com/MichelliBrito/ead-config-server-repo
          default-label: main
          username: ${GIT_USERNAME}
          password: ${GIT_PASSWORD}

encrypt:
  key: z%C*F-
JaNdRgUkXp2s5u8x/A?D (G+KbPeShVmYq3t6w9y$B&E) H@McQfTjWnZr4u

ead:
  configServer:
    username: ${CONFIG_SERVER_USERNAME}
    password: ${CONFIG_SERVER_PASSWORD}

eureka:
```

```
client:
  serviceUrl:
    defaultZone: ${EUREKA_URL}
instance:
  hostname: ${APP_DOMAIN_NAME}
  non-secure-port: 80
```

5. Na raiz do diretório da aplicação, inserir novo arquivo `system.properties` contendo a versão do java utilizado (`java.runtime.version=11`).
6. Fazer o commit das alterações realizadas para ambiente de prod

3. Deploy de ConfigServer Microservice no Heroku Platform

1. Realizar login no heroku via terminal: **heroku login**
2. Criar app heroku utilizando o seguinte command line via terminal:
 - a. **heroku create -a <app-name> --remote heroku-prod**
 - b. Verificar a criação do app no dashboard do Heroku
 - c. Para verificar o endereço do repositório remoto git da app dentro do heroku que foi criada utilizar **git remote -v** ou visualizar a url em heroku -> Settings (Opcional)
3. No Heroku, em Settings -> Reveal Config Vars do app criar variáveis de ambiente necessárias (key-value):
 - a. APP_DOMAIN_NAME : <app-name>.herokuapp.com
 - b. EUREKA_URL: <https://username:password@<app-name>.herokuapp.com/eureka>
 - c. CONFIG_SERVER_USERNAME : username
 - d. CONFIG_SERVER_PASSWORD : password
 - e. GIT_USERNAME : username-git
 - f. GIT_PASSWORD : token-git
4. Realizar o deploy enviando o código-fonte para o repositório remoto Git do Heroku criado no passo 3.2(a) utilizando o seguinte comando: **git push heroku-prod prod:master**
5. Para verificar logs pode-se utilizar o comando: **heroku logs -tail** (opcional)
6. Pode-se verificar logs no dashboard do Heroku também clicando no botão More -> View logs
7. Verificar registro do configServer no ServiceRegistry acessando o dashboard Eureka
8. Voltar em Resources e alterar dynos para o plano Hobby Dev de 7 dólares (opcional).