

02

## Entendendo o build manual

### Transcrição

O script que o instrutor segue durante a aula é o seguinte:

```
# Passos para configurar a app e subir manualmente
# Criando o arquivo .env (temporário)
cd /vagrant/jenkins-todo-list/to_do/
vi .env
[config]
# Secret configuration
SECRET_KEY = 'r*5ltfzw-61ksdm41fuul8+hxs$86yo9%k1%k=( !@=-wv4qtyv'

# conf
DEBUG=True

# Database
DB_NAME = "todo_dev"
DB_USER = "devops_dev"
DB_PASSWORD = "mestre"
DB_HOST = "localhost"
DB_PORT = "3306"

# Instalando o venv
sudo pip3 install virtualenv nose coverage nosecover pylint
# Criando e ativando o venv (dev)
cd ../
virtualenv --always-copy venv-django-todolist
source venv-django-todolist/bin/activate
pip install -r requirements.txt
# Fazendo a migracao inicial dos dados
python manage.py makemigrations
python manage.py migrate
# Criando o superuser para acessar a app
python manage.py createsuperuser
# Repetir o processo de migracao para o ambiente de producao:
vi to_do/.env
[config]
# Secret configuration
SECRET_KEY = 'r*5ltfzw-61ksdm41fuul8+hxs$86yo9%k1%k=( !@=-wv4qtyv'

# conf
DEBUG=True

# Database
DB_NAME = "todo"
DB_USER = "devops"
DB_PASSWORD = "mestre"

DB_HOST = "localhost"
DB_PORT = "3306"
```

```
# Fazendo a migração inicial dos dados
    python manage.py makemigrations
    python manage.py migrate
# Criando o superuser para acessar a app
    python manage.py createsuperuser

# Verificar o ip do servidor
    ip addr
# Rodando a app
    python manage.py runserver 0:8000
    http://192.168.33.10:8000
```

[00:00] Bom pessoal, agora tá na hora da gente fazer o build manual da nossa aplicação. Porque que a gente vai fazer isso? Pra que quando a gente automatize toda construção, a gente consiga enxergar a diferença de velocidade e de agilidade de um processo pro outro.

[00:15] Pra fazer o build, a aplicação é muito simples. Primeiro, dentro do meu diretório da aplicação, eu preciso criar um arquivo ".env". Que que esse arquivo ".env" contém? Vou pegar aqui um modelo dele. Basicamente, ele tem algumas configurações pra aplicação rodar, como por exemplo, o secret key que é utilizado pra fazer os hash de senha, por exemplo, nome de usuários de banco de dados, nome do banco de dados, a porta e onde o banco tá localizado.

[00:51] Atentem-se pro seguinte: eu estou conectando do banco de dados de dev. E porque que eu tô criando esse arquivo? Vamos primeiro salvar o arquivo aqui. A gente usou o VI pra quem não tá muito acostumado. Entrou no arquivo, aperto o I, colo o conteúdo, Esc, :x. A gente salvou o arquivo aqui, se eu der um "ls -la" tá aqui o meu arquivo env.

[01:14] Porque que a gente não coloca o arquivo env dentro da aplicação versionada? Porque isso vai pro GitHub, ou até, às vezes, pra um repositório privado. O problema é que o arquivo env contém as informações de conexão de banco, por exemplo, isso não pode ficar público.

[01:30] Com o arquivo copiado, que que a gente faz agora? A gente vai instalar algumas dependências do Python com o PIP, que é o gerenciador de pacotes do Python. Mandei instalar, ele tá instalando, é rápida a instalação, ele busca nos repositórios e faz a instalação pra gente.

[01:55] Legal, a instalação dos pacotes feita, o que eu vou fazer? Eu vou criar um Virtualenv pro Python, pra isolar a minha aplicação. Vocês tão percebendo o trabalho que dá fazer manual o build disso daqui. Então eu vou criar um Virtualenv, ele vai isolar os pacotes da minha aplicação. Legal, ele instalou os pacotes da minha aplicação e agora eu vou ativar o meu Virtualenv.

[02:25] Legal, Virtualenv ativado, eu consigo ver pelo começo do bash agora. Então o que eu tenho que pedir pra ele fazer agora? Instalar todos os requerimentos da minha aplicação. Se eu der um cat nesse arquivo vocês vão ver que tem várias dependências. Então, pra isso, eu vou executar o pip install -r requirements.

[02:44] Isso é legal ver, enquanto tá rodando, que tomar conta de uma aplicação, ou deployar uma aplicação, você não precisa, exatamente, configurar todo o ambiente dela manualmente porque a gente tá tendo muito trabalho pra fazer isso. O legal é a gente automatizar. Então as próximas aulas vão mostrar como que isso é aqui é feito de uma maneira invisível pra quem tá aplicando a Integração Contínua.

[03:09] Normalmente quando a gente roda o pip install eventualmente ele tem algum problema de cache, é só rodar de novo, só pra validar a instalação, mas foi tudo instalado. Isso não é um erro de aplicação e sim um erro do próprio PIP mesmo, que quando a gente automatiza não aparece.

[03:28] Legal, então a gente tá com as dependências instaladas, o que que a gente vai fazer agora? Tem um passo que a gente vai executar agora, uma sequência de passos na verdade, que é específico do Django. Como o nosso curso é de Jenkins a gente vai criar esse ambiente na mão na primeira vez, é uma vez só que a gente vai criar, que são as migrações de banco, que é onde ele vai criar as tabelas do banco de dados.

[03:55] Isso não vai entrar na nossa Integração Contínua porque é específico do Python. Se você trabalha com Java você vai ter que lidar com isso na hora de criar a sua aplicação. Você pode automatizar esses passos, isso é possível, mas como o foco do curso não é Django a gente só vai criar a primeira vez. É uma vez só que a gente executa e não executa mais.

[04:18] Então a gente vai rodar as migrações do banco, a gente vai criar as migrações do banco, baseado nos modelos e a gente vai migrar o banco. Então ele tá fazendo toda a configuração de tabelas e permissões, e agora o que eu tenho que fazer? Eu tenho que criar um super usuário, isso é específico do Django. É o usuário admin, o root do Django pra acessar as aplicações.

[04:51] Então eu vou criar um usuário aqui com o nome alura, o endereço de email vai ser aluno@alura.com.br, a senha vai ser mestre123, vamos manter aquele padronzinho mestre123 pra sempre lembrar. E o super usuário foi criado.

[05:18] Bom, então agora tá na hora da gente ver a aplicação rodar, até agora a gente não viu como ela funciona, qual que é a carinha dela. A gente vai rodar um comando aqui que só é utilizado no ambiente de desenvolvimento, no pipeline nosso a gente vai usar outro que é para produção. Startei a aplicação.

[05:39] Então agora que a gente já colocou a aplicação pra rodar, a gente vai usar o mesmo IP que a gente utilizou pra abrir o Jenkins, pra abrir a aplicação, porque eles estão rodando no mesmo servidor. Só que o que muda agora é a porta que, ao invés de ser a porta 8080, que é onde o Jenkins tá rodando, a gente vai usar a porta 8000. Aí a aplicação apareceu. O usuário e a senha a gente acabou de criar, alura, mestre123. Abrimos a aplicação aqui.

[06:11] Pessoal, tem uma outra coisa que a gente vai fazer, primeiro a gente vai parar o nosso servidor aqui, que é o seguinte: tudo tá apontado pro banco de dev, a gente vai editar o nosso arquivo aqui e vai enxergar. Dentro do nosso arquivo de ambiente tá tudo apontado pra dev, só que a gente vai ter um ambiente de produção que é um outro schema e um outro usuário.

[06:36] Então vamos rodar rapidinho as migrations pro usuário de produção. Então a gente apaga a parte dev, porque a gente já tem essa tabela criada, a senha é a mesma, a gente só vai rodar os migrations de novo, makemigrations, na verdade nem precisava porque já estavam criadas as migrações.

[07:04] Agora o migrate vai fazer o que? Ele vai conectar no banco de produção, a gente trocou as tabelinhas lá e ele criou de novo. Isso é bem específico, só pra que na próxima etapa do build a gente não tenha problema. E a gente vai criar o super usuário da mesma maneira. Vamos criar os mesmos dados pra não ficar confuso, alura, aluno@alura.com.br, mestre123.

[07:37] Legal, criou o usuário, a gente sobe a aplicação pra validar que tá tudo funcionando, atualiza aqui a aplicação. Logou, conectamos. Pra utilizar a aplicação é muito simples, você coloca o título da sua task, o texto, pode marcar como completa ou não, e dar um salvar, sua task tá aqui.

[08:04] Bom pessoal, agora a gente acabou de ver o build manual, como ele é feito. Na próxima aula a gente vai acertar o último detalhe antes de colocar todo o nosso processo automatizado dentro do Jenkins. A gente se vê lá.