Agora que o projeto está configurado com idioma e horário, precisamos versionar o projeto usando *Git* e *Github*.

Porém, não podemos enviar todas as partes do código para essas ferramentas, por questões de segurança. Se enviássemos o código por inteiro, receberíamos uma mensagem do *Github*, avisando que uma *Django Secret Key* está acessível no *Github*.

A linha que contém a *Secret Key*, chave secreta, está em "setup > settings.py". Vamos clicar em "Ctrl + B" para minimizar a aba da esquerda. A *Secret Key* está na linha 23 do arquivo.

Todo projeto *Django* tem uma chave secreta. Essa *key* não deve ser disponibilizada para outras pessoas. Sempre que a disponibilizarmos para visualização pública no *Github*, receberemos a mesma mensagem.

Vamos aprender a selecionar quais arquivos do projeto enviamos para o *Github* e, ainda mais importante, aprenderemos a não enviar, especificamente, a *Secret Key*. Para dar início ao processo, vamos copiar a *Secret Key* com a ajuda de "Ctrl + C". Depois de copiá-la, vamos removê-la do código para executar um teste.

Se atualizarmos o navegador, receberemos a mensagem de que um erro ocorreu, porque não é possível ficarmos sem a *Secret Key*. Vamos precisar criar algumas variáveis de ambientes. Com elas, conseguiremos evitar que a *Secret Key* e outros dados confidenciais sejam enviados.

No terminal, vamos solicitar que o *Pip* instale uma nova dependência, com o comando pip install python-dotenv. Depois da alteração, vamos atualizar o arquivo de *Requirements* com pip freeze > requirements.txt.

Vamos criar uma novo arquivo na pasta "alura-space", chamado ".env". Nele, vamos inserir a *Secret Key*, seguindo o modelo abaixo:

SECRET\_KEY = sua secret keyCopiar código

Agora, podemos remover a *Secret Key* do arquivo "settings.py". Vamos carregar, agora, as variáveis de ambiente. Na linha 13, onde importamos Path, vamos importar também o os, utilizando from dotenv import load\_dotenv.

Abaixo, criaremos a função load\_dotenv():

**from** pathlib **import** **Path**, os

**from** dotenv **import** load\_dotenv

**load\_dotenv**()Copiar código

Agora, vamos descer até a linha 26 do código, onde antes havia a *Secret Key*. Vamos passar a variável transformada em *string* str(os.getenv('SECRET\_KEY')) no lugar da *Secret Key*.

Agora, voltaremos ao terminal e apertaremos "Ctrl + L". Vamos executar o comando python manage.py runserver, para executar nosso servidor. Ele será carregado. Se atualizarmos a página do navegador, veremos que o projeto terá voltado a funcionar.

Agora, quando enviarmos para o *Github*, ele apresentará a string str(os.getenv('SECRET\_KEY')) ao invés da *Secret Key*, que está armazenada somente no nosso computador. Isso acontece porque nós não enviamos o arquivo .env para nosso repositório.