Mozout

Gerado a 15/11/2026 03:29

BASE DE CONHECIMENTO · DOCUMENTO PÚBLICO

Pôr uma aplicação Django a correr

Python e outras linguagens

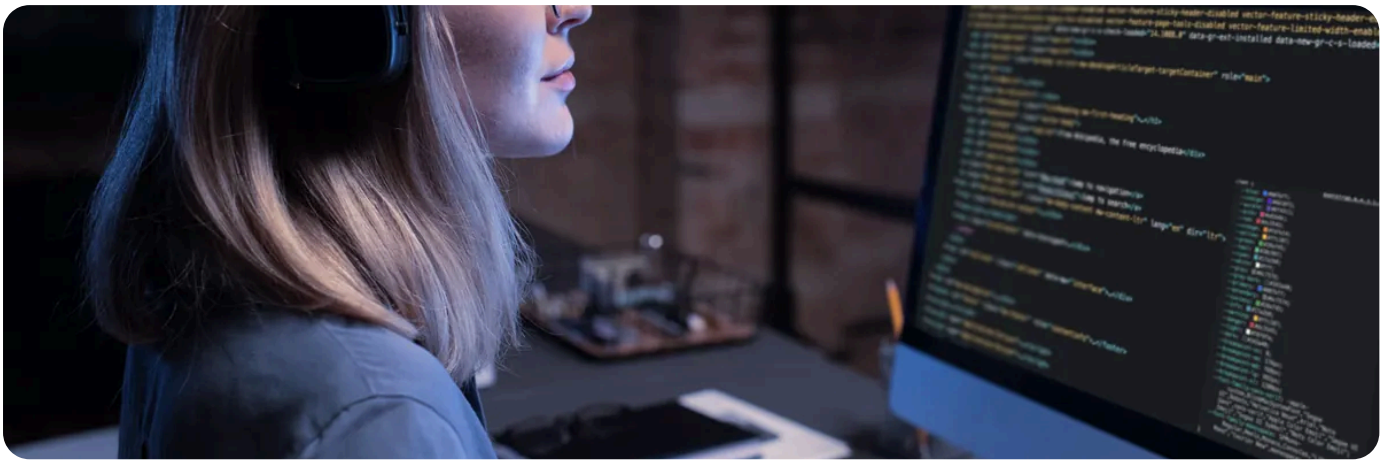
CÓDIGO	VERSÃO	PAÍS · IDIOMA
RLX-MZ-KB-1592	v2026.09.27	Moçambique · Português

MozOut Moçambique
RLX Holdings, S.A. · Grupo RLX Holdings
NUIT 401 191 607 · NUEL 105 045 974 · CAE 62022

https://www.my.mozout.com/knowledgebase/1592

Pôr uma aplicação Django a correr

Python e outras linguagens · Criado em **27 de setembro de 2026** · 4 min de leitura



O **Django** não está no instalador de aplicações do cPanel, e não é falta: aquele instalador é para aplicações em PHP. O Django corre em Python, e para Python o cPanel tem ferramenta própria, no grupo **Software**: o **Setup Python App**.

É essa página que cria um ambiente Python isolado para a sua aplicação, escolhe a versão do interpretador e liga a aplicação a um endereço do seu domínio. As versões disponíveis no dia em que este artigo foi escrito vão da **2.7** até à **3.13**, com todas as intermédias da série 3.

> **Escolha uma versão recente do Python.**
> As antigas continuam na lista para manter aplicações que nunca foram migradas, e não para começar nada. Um Django novo deve nascer nas versões mais altas da lista, porque é com essas que as bibliotecas de que ele depende ainda são publicadas.

Criar a aplicação, passo a passo

1. Entre no **My MozOut** e abra o cPanel do plano.
2. No grupo **Software**, abra o **Setup Python App** e clique em criar aplicação.
3. **Escolha a versão do Python.** Depois de a aplicação estar montada, mudar de versão obriga a reinstalar tudo, por isso pense duas vezes agora.
4. **Application root:** a pasta onde vai ficar o código. Deve ficar **fora** da pasta pública do site.
5. **Application URL:** o domínio, ou o domínio mais um caminho, onde a aplicação vai responder.

6. **Application startup file** e **Entry point:** o ficheiro que arranca a aplicação e o objecto dentro dele. Num Django, é o objecto da aplicação web que o próprio Django gera no projecto.

7. Crie. A página passa a mostrar-lhe, no topo, **o comando que activa este ambiente** na linha de comandos. Copie-o: vai precisar dele.

Pôr lá o código e as bibliotecas

1. **Envie o projecto** para a pasta que indicou, por FTP ou pelo Gestor de Ficheiros: **contas FTP e o FileZilla**. Não envie a pasta do ambiente virtual do seu computador: ela não serve aqui.

2. **Ponha um ficheiro de requisitos na raiz do projecto**, com as bibliotecas e as versões. A página do selector tem um botão que o instala por si, sem linha de comandos nenhuma.

3. **Ou instale pela linha de comandos**, se tiver acesso SSH: primeiro corre o comando de activação que a página lhe deu, e só depois instala. Se instalar sem activar, as bibliotecas vão parar ao sítio errado e a aplicação continua a dizer que não as encontra.

4. **Defina as variáveis de ambiente** na mesma página. É aí que deve ficar a chave secreta do Django e os dados de ligação à base, e não escritos no código.

Depois de cada alteração ao código, reinicie a aplicação.
Há um botão para isso na página do selector. Ao contrário do PHP, uma aplicação Python fica carregada em memória: enquanto não reiniciar, continua a servir o código antigo, e você vai jurar que o ficheiro não foi gravado.

A base de dados, os estáticos e o resto

Assunto	O que fazer
Base de dados	Crie-a no cPanel como para qualquer outra aplicação: **criar bases de dados e usar o phpMyAdmin**. Depois aponte as definições do Django para ela, pelas variáveis de ambiente.
Ficheiros estáticos	Em produção, o Django não serve os estáticos. Recolha-os com o comando próprio dele para uma pasta que fique dentro da área pública, e sirva-os a partir daí.
Endereços permitidos	Ponha o seu domínio na lista de anfitriões permitidos do Django. Se não o fizer, ele recusa todos os pedidos e você vê um erro que não explica nada.
Modo de depuração	**Desligue-o antes de pôr no ar.** Com ele ligado, qualquer erro mostra ao visitante os caminhos, as definições e, por vezes, as credenciais.

Assunto	O que fazer
Tarefas periódicas	Use as tarefas agendadas do cPanel, chamando o comando dentro do ambiente que criou: **tarefas agendadas**.

Processos que ficam a correr sem parar não cabem num plano partilhado.
A aplicação web arranca a pedido e isso funciona bem; um trabalhador de fila permanente, um processo de tarefas assíncronas ou um servidor de *websockets* batem no tecto de processos da conta e são interrompidos. Se a aplicação precisa disso, o sítio dela é um servidor próprio: **VPS, Cloud ou Dedicado**. Sobre os tectos, veja **os limites da conta**.

Para o panorama do que corre aqui, veja **que tecnologias correm aqui**. Se a aplicação é noutra linguagem, veja **Ruby on Rails** ou **Laravel**.

PRECISA DE AJUDA?

A aplicação foi criada e o endereço dá erro? Diga-nos o domínio e a pasta, e lemos o registo consigo.

Abrir um pedido de suporte →

VEJA TAMBÉM

- **Planos de hospedagem e o que cada um inclui** →
- **Servidor VPS** →
- **Política de Suporte** →

MozOut Moçambique

RLX Holdings, S.A. · Grupo RLX Holdings

NUIT	401 191 607
NUEL	105 045 974
CAE	62022
SEDE	Bairro 7 de Setembro Vilankulo 1304 · CEP 0411-02, Inhambane, Moçambique
CONTACTOS	info@mozout.com · +258 84 635 1111 · +258 86 635 1111 · www.mozout.com
HORÁRIO	2ª. à 6ª. Feira - 08 à 16 Horas Sábado & Domingo - Fechado

Versão online deste documento

RLX-MZ-KB-1592 · v2026.09.27 · Gerado a 15/11/2026 03:29