**CENTRO UNIVERSITÁRIO ESTÁCIO DE JUIZ DE FORA**

**COMPARAÇÃO ENTRE FERRAMENTAS PARA DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTON**

**Aluno 1**

**Samuel Moisés Guimarães Macanham**

**Aluno 2**

**Francielly Araújo Nascimento**

**Aluno 3**

**Aluno 4**

**Juiz de Fora, setembro de 2023**

Sumário

[**1) INTRODUÇÃO** 3](#_Toc145580260)

[**2)FRAMEWORKS EM PYTHON** 3](#_Toc145580261)

[**2.1) Áreas De Aplicação** 3](#_Toc145580262)

[**3) FLASK** 3](#_Toc145580263)

[**3.1) História** 3](#_Toc145580264)

[**3.2) Motivação** 4](#_Toc145580265)

[**3.3) Princípios Básicos** 4](#_Toc145580266)

[**3.4) Exemplo de aplicações** 4](#_Toc145580267)

[**4) DJANGO** 4](#_Toc145580268)

[**4.1) História** 5](#_Toc145580269)

[**4.2) Motivação** 5](#_Toc145580270)

[**4.3) Princípios básicos** 5](#_Toc145580271)

[**4.4) Exemplo de aplicações** 5](#_Toc145580272)

[**5) APLICAÇÃO DE EXEMPLO** 6](#_Toc145580273)

[**5.1) Código em Flask** 6](#_Toc145580274)

[**5.2) Código em Django** 6](#_Toc145580275)

[**6) CONCLUSÕES** 6](#_Toc145580276)

[**7) REFERÊNCIAS** 8](#_Toc145580277)

**1) INTRODUÇÃO**

**2)FRAMEWORKS EM PYTHON**

**2.1) Áreas de Aplicação**

**3) FLASK**

**3.1) História**

**Flask é um microframework web Python que foi lançado em 2010 por Armin Ronacher. Diferentemente do Django, que é um framework de alto nível, o Flask é projetado para ser minimalista e flexível, dando aos desenvolvedores mais controle sobre os componentes que desejam usar em suas aplicações.**

**3.2) Motivação**

A motivação por trás do Flask era fornecer aos desenvolvedores uma estrutura leve e extensível para o desenvolvimento web. Ronacher queria criar um framework que fosse fácil de aprender e usar, mas que não impusesse muitas restrições.

**3.3) Princípios Básicos**

Microframework: Flask é um microframework que fornece o mínimo necessário para criar uma aplicação web, deixando a escolha de outros componentes, como um ORM ou um mecanismo de autenticação, para o desenvolvedor.

Extensibilidade: Flask é altamente extensível e permite que os desenvolvedores escolham as extensões e bibliotecas que melhor atendam às necessidades de suas aplicações.

Simplicidade: Flask segue o princípio da simplicidade, tornando-o uma escolha popular para projetos pequenos e médios.

Flexibilidade de Design: Os desenvolvedores têm liberdade para projetar a estrutura de seus aplicativos da maneira que desejarem, o que torna o Flask uma escolha ideal para projetos personalizados.

**3.4) Exemplo de aplicações**

Um exemplo de aplicação Flask poderia ser a criação de um microblog. Você pode definir rotas para criar, exibir e excluir postagens, usando bibliotecas adicionais, como SQLAlchemy para lidar com o banco de dados. A estrutura mínima do Flask facilita a criação de um aplicativo personalizado, escolhendo os componentes específicos que você deseja incorporar.

**4) DJANGO**

**4.1) História**

Django é um framework web Python de alto nível que foi lançado pela primeira vez em 2005 por Adrian Holovaty e Simon Willison. Foi originalmente desenvolvido na World Company, uma agência de notícias online, com o objetivo de acelerar o desenvolvimento de aplicações web robustas e escaláveis. O nome "Django" é uma homenagem ao guitarrista de jazz Django Reinhardt, refletindo a ênfase do framework na simplicidade e na eficiência.

**4.2) Motivação**

A motivação por trás do desenvolvimento do Django era simplificar o processo de criação de aplicações web, fornecendo um conjunto de componentes e convenções que acelerassem o desenvolvimento, promovessem a reutilização de código e incentivasse boas práticas de programação.

**4.3) Princípios básicos**

Django ORM (Object-Relational Mapping): Django possui um ORM poderoso que permite aos desenvolvedores interagir com o banco de dados usando objetos Python, eliminando a necessidade de escrever SQL manualmente.

Administração Automatizada: O Django inclui uma interface de administração automatizada que gera automaticamente uma interface de administração para seu aplicativo com base em seus modelos de dados.

Padrão MVC (Model-View-Controller): Django segue o padrão arquitetural MVC (Model-View-Controller), onde os modelos representam os dados, as visualizações controlam a lógica de apresentação e os controladores são responsáveis pela lógica de negócios.

Rotas e URLs Simples: Django possui um sistema de roteamento simples e eficaz para mapear URLs para funções de visualização.

**4.4) Exemplo de aplicações**

Um exemplo de aplicação Django poderia ser a criação de um blog. Você pode definir modelos para Postagens e Comentários, usar o sistema de administração para gerenciar postagens e, em seguida, criar visualizações para exibir essas postagens no site. O ORM do Django facilita a criação e recuperação de dados do banco de dados, e os recursos de segurança integrados ajudam a proteger seu aplicativo contra ameaças comuns da web.

**5) APLICAÇÃO DE EXEMPLO**

**5.1) Código em Flask**

from flask import Flask

# Crie uma instância do aplicativo Flask

app = Flask(\_\_name\_\_)

# Defina uma rota para a página inicial

@app.route('/')

def hello\_world():

return 'Hello, World!'

if \_\_name\_\_ == '\_\_main\_\_':

# Execute o aplicativo Flask

app.run()

**5.2) Código em Django**

# hello/views.py

from django.http import HttpResponse

def hello\_world(request):

return HttpResponse("Hello, World!")

# hello/urls.py

from django.urls import path

from . import views

urlpatterns = [

path('', views.hello\_world, name='hello\_world'),

]

# helloworldproject/urls.py

from django.contrib import admin

from django.urls import include, path

urlpatterns = [

path('admin/', admin.site.urls),

path('hello/', include('hello.urls')),

]

**6) CONCLUSÕES**

**Em resumo, Django e Flask são dois frameworks Python populares para o desenvolvimento web, cada um com sua própria abordagem e filosofia. Django é uma escolha sólida para projetos complexos e de grande escala, devido à sua estrutura abrangente e recursos integrados, enquanto o Flask é mais adequado para projetos menores e personalizados, oferecendo maior flexibilidade. A escolha entre eles depende das necessidades específicas do projeto e das preferências do desenvolvedor.**

**7) REFERÊNCIAS**