16/05/2024

Deselvolvimento BackEnd II

* Arquitetura Client server
* Client: Todo serviço que faz uma requisição a um servidor
* Server: Tudo Recurso Capaz de responder a requisição de um cliente
* Protocolo 🡪 Conjunto de regras
  + Protocolo HTTP (Hyper text trasnfer protocol)
* Principais servers Nuvem
  + Aws (Amazon)
  + Azure(Microsoft)
  + Google Cloud(Google)

Cloud é um recurso Caro

* Desenvolver um servidor Python

Apresentados 3 Grandes criadores de servidores python

* FastAPI

<https://fastapi.tiangolo.com/>

Framework (intermediário) Criação de API

* Django

<https://www.djangoproject.com/>

Framework

Se concentra em variedade de conteúdo

Conttend driven

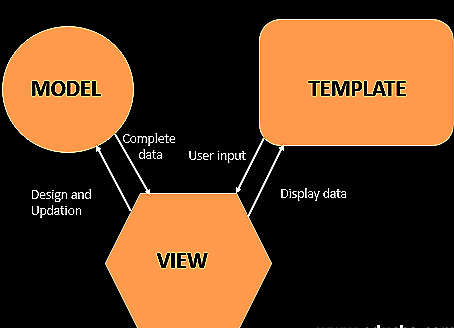
Ideal para site de noticias

Aplicacoes baseadas em monolito

Guiado por criação rápida de conteúdo

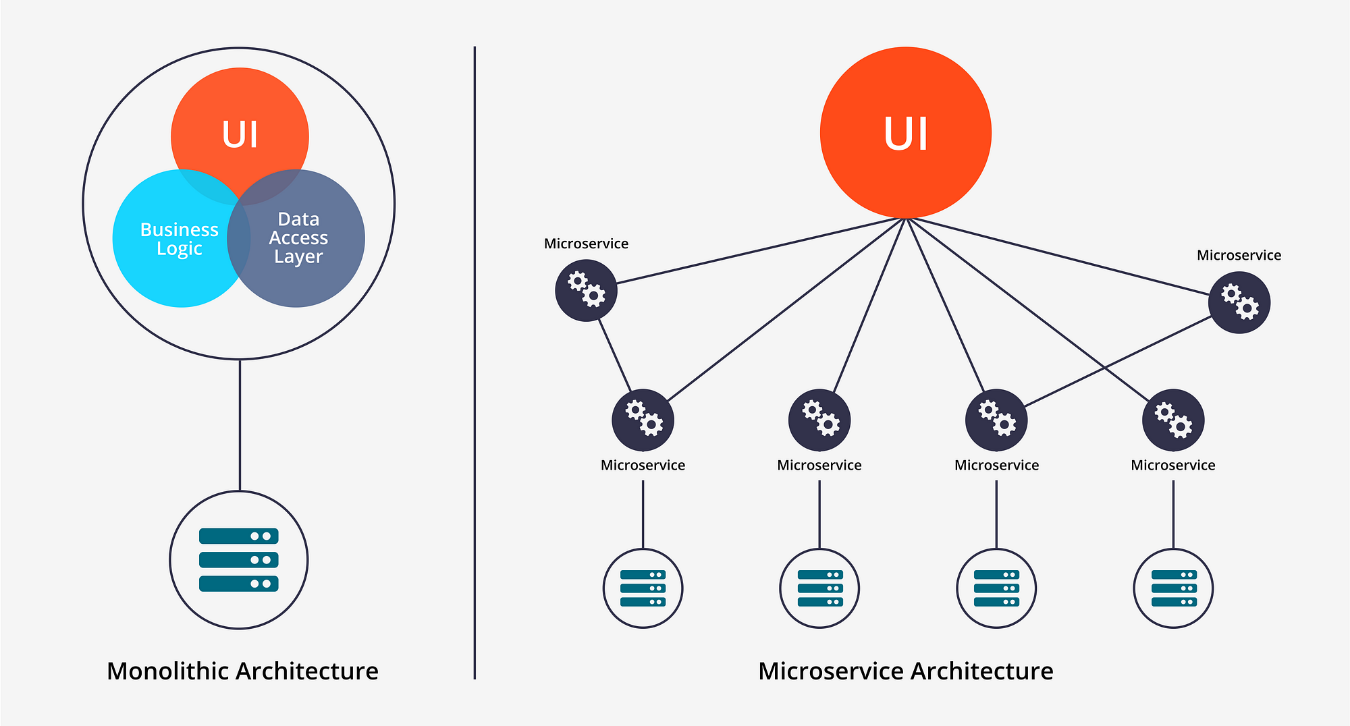
\* Globo tem seu DJANGO próprio

Django não usa MVC mas sim MVT



Aonde

* Model corresponde a model
* View corresponde a control :s 😕
* Template corresponde ao view
* Arquitetura Monolítica X Arquitetura de micro serviços



* Cors (Cross origin Resource sharing)

Mecanismo que permite que recursos restritos em uma pagina web sejam recuperados por outro domínio fora do domínio ao qual pertence o recurso que será recuperado

<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>

* É possível criar PAI com pythin Vanila (Sem frameworks) :

Sim é possível MAS não e viável, consome muito tempo e recurso e não traz nenhuma vantagem comparativa em relação a performance.

* ThunderClient

Extensão VSCODE RESTAPI como se fosse um light postman

Mercado de Ruby e grande na europa pesquisar sobre Ruby

Comandos docker

<https://www.alura.com.br/artigos/ambientes-virtuais-em-python>

docker-compose up -d --build

python -m venv ./venv

source venv/bin/activate

pip install django

pip freeze

pip freeze > requirements.txt

django-admin startproject setup .