

# Desenvolvimento de Sistemas

Unidade curricular:

Programação Web Back-End  
- PWBE -

# Desenvolvimento de Sistemas

## O que veremos ?

- Framework
- Django

# Desenvolvimento de Sistemas

## O que é um Framework?

- Um **framework** é um conjunto de bibliotecas e ferramentas que fornecem uma estrutura predefinida para o desenvolvimento de software.
- Ele facilita o desenvolvimento, ao oferecer soluções prontas para tarefas comuns e uma arquitetura para organizar o código, permitindo que os desenvolvedores se concentrem mais nas funcionalidades do que em detalhes técnicos repetitivos.

# Desenvolvimento de Sistemas

## Tipos de Frameworks

Os frameworks podem ser divididos em diferentes tipos, dependendo da sua aplicação.

- **Frameworks para Desenvolvimento Web:** São usados para criar aplicações web. Exemplo: **Django**, Flask (Python), Ruby on Rails (Ruby), Laravel (PHP).
- **Frameworks para Desenvolvimento de Aplicações Móveis:** São utilizados para criar aplicativos móveis. Exemplo: Flutter, React Native.
- **Frameworks para Desenvolvimento de Jogos:** Utilizados para criar jogos. Exemplo: Unity, Unreal Engine.
- **Frameworks para Desenvolvimento de Software de Escritório:** Facilitam o desenvolvimento de programas de desktop. Exemplo: Qt, GTK.

# Desenvolvimento de Sistemas

## Vantagens de usar Frameworks

- **Produtividade Aumentada:** Você ganha tempo ao não precisar reinventar a roda.
- **Organização do Código:** Eles seguem padrões que ajudam a manter o código mais organizado e limpo.
- **Segurança:** Frameworks frequentemente incluem funcionalidades de segurança, como proteção contra SQL injection, CSRF, etc.
- **Facilidade de Manutenção:** O uso de um framework bem estruturado facilita a manutenção e evolução do projeto.

# Desenvolvimento de Sistemas

## Django

- **Produtividade Aumentada:** Você ganha tempo ao não precisar reinventar a roda.
- **Organização do Código:** Eles seguem padrões que ajudam a manter o código mais organizado e limpo.
- **Segurança:** Frameworks frequentemente incluem funcionalidades de segurança, como proteção contra SQL injection, CSRF, etc.
- **Facilidade de Manutenção:** O uso de um framework bem estruturado facilita a manutenção e evolução do projeto.

# Desenvolvimento de Sistemas

## Django

### O que é o Django?

- O Django é um **framework web** de alto nível para o desenvolvimento de aplicações web, escrito em **Python**.
- Ele segue o princípio DRY (Don't Repeat Yourself) e oferece uma maneira rápida e simples de desenvolver sites dinâmicos, com uma poderosa administração, segurança e escalabilidade.

# Desenvolvimento de Sistemas

## Django

### Características principais:

- **Administração Automática:** Oferece uma interface de administração que facilita a gestão dos dados da aplicação.
- **ORM (Object-Relational Mapping):** Facilita a interação com o banco de dados, permitindo que você trabalhe com objetos Python ao invés de escrever SQL manualmente.
- **Segurança:** Protege contra muitos ataques de segurança como CSRF, SQL Injection e Clickjacking.
- **Escalabilidade e Desempenho:** Pode lidar com grandes quantidades de tráfego e dados.



# Desenvolvimento de Sistemas

## **Django**

### **Instalação e Configuração do Django**

# Desenvolvimento de Sistemas

## Django

- **Instalar o Django:** *pip install django*

# Desenvolvimento de Sistemas

## Django

- **Instalar o Django:** *pip install django*
- **Verificar a instalação:** *django-admin --version*

# Desenvolvimento de Sistemas

## Django

- **Instalar o Django:** *pip install django*
- **Verificar a instalação:** *django-admin --version*
- **Criando um projeto Django:** *django-admin startproject nome\_do\_projeto*

# Desenvolvimento de Sistemas

## Django

- **Instalar o Django:** *pip install django*
- **Verificar a instalação:** *django-admin --version*
- **Criando um projeto Django:** *django-admin startproject nome\_do\_projeto*
- **Rodando o servidor de desenvolvimento:**
  - *cd nome\_do\_projeto*
  - *python manage.py runserver*
- <http://127.0.0.1:8000/>

# Desenvolvimento de Sistemas

## Criando um Projeto Django Simples

- **Passo 1:** Criar uma aplicação dentro do projeto.
  - `python manage.py startapp nome_da_aplicacao`
- **Passo 2:** Definir uma Rota e uma View. Dentro do diretório da aplicação, no arquivo `views.py`, defina uma função simples de view:
  - `from django.http import HttpResponse`
  - `def ola(request):`
  - `return HttpResponse("Olá, DS14!")`
- Agora, crie uma rota (URL) para essa view. No arquivo `urls.py` dentro do diretório do projeto, faça a seguinte configuração:
  - `from django.urls import path`
  - `from nome_da_aplicacao import views`
  - `urlpatterns = [`
  - `path('ola/', views.ola),`
  - `]`

# Desenvolvimento de Sistemas

## Criando um Projeto Django Simples

- **Passo 3:** Rodar o servidor e acessar a página.
  - *python manage.py runserver*