

## **semana 6- aula 01**

### **Linguagens de programação back-end – Python**

#### **Introdução ao Python para back-end**

**Código da aula: [SIS]ANO2C2B1S6A1**

#### **Objetivos da Aula:**

- ❖ Identificar as características do Python para back-end.
- ❖ Identificar as características do Python para back-end.
- ❖ Incentivar a curiosidade ao explorar as características e as vantagens do Python para desenvolvimento back-end.
- ❖ Recurso audiovisual para exibição de vídeos e imagens; • Lápis e caderno para anotações

#### **Exposição**

##### **Python**

Python é uma linguagem de programação versátil e poderosa que se destaca no desenvolvimento de back-end. Aqui está um resumo dos principais pontos que definem seu perfil para essa área:

#### **Pontos Fortes:**

- \* Simplicidade e Legibilidade:
  - \* Sua sintaxe clara e concisa facilita o aprendizado e a escrita de código, tornando o desenvolvimento mais rápido e eficiente.
- \* Vasta Biblioteca Padrão:
  - \* Python oferece uma rica coleção de módulos e bibliotecas que simplificam tarefas comuns, como manipulação de dados, comunicação de rede e acesso a bancos de dados.
- \* Frameworks Web Robustos:
  - \* Frameworks como Django e Flask fornecem ferramentas e estruturas poderosas para construir aplicações web complexas e escaláveis.
  - \* Django: Ideal para projetos grandes e complexos, com recursos integrados para gerenciamento de banco de dados, autenticação e muito mais.
  - \* Flask: Ideal para projetos menores, ou microserviços, onde se preza a flexibilidade, e a modularidade.
- \* Comunidade Ativa e Suporte:
  - \* Uma grande comunidade de desenvolvedores oferece suporte, recursos e bibliotecas de terceiros, facilitando a resolução de problemas e o aprendizado contínuo.
- \* Escalabilidade:
  - \* Python permite a criação de aplicações de back-end escaláveis, capazes de lidar com grandes volumes de tráfego e dados.
- \* Versatilidade:

- \* Python se integra facilmente com outras tecnologias e sistemas, permitindo a construção de soluções completas e personalizadas.

### **Aplicações Comuns:**

- \* Desenvolvimento de APIs (Interfaces de Programação de Aplicativos)
- \* Criação de aplicações web dinâmicas
- \* Automação de tarefas de servidor
- \* Processamento de dados e análise
- \* Desenvolvimento de microsserviços

Python oferece um equilíbrio ideal entre facilidade de uso, poder e versatilidade, tornando-o uma escolha popular para desenvolvedores de back-end que buscam construir aplicações web eficientes e escaláveis.

Ele possui uma variedade de frameworks robustos, cada um com seus pontos fortes e casos de uso específicos. Aqui estão alguns dos mais conhecidos:

#### **\* Django:**

- \* Um framework completo e de alto nível, ideal para o desenvolvimento rápido de aplicações web complexas e escaláveis.

- \* Oferece recursos integrados para gerenciamento de banco de dados, autenticação, roteamento de URLs e muito mais.

#### **\* Flask:**

- \* Um microframework leve e flexível, que permite aos desenvolvedores construir aplicações web com maior controle e personalização.

- \* Ideal para projetos menores, APIs e microsserviços.

#### **\* FastAPI:**

- \* Um framework moderno e de alto desempenho para a criação de APIs web rápidas e eficientes.

- \* Utiliza tipagem estática e async/await para otimizar o desempenho.

#### **\* Pyramid:**

- \* Um framework flexível e extensível, que oferece uma abordagem minimalista e permite aos desenvolvedores escolher as ferramentas e bibliotecas que melhor se adequam ao seu projeto.

#### **\* Tornado:**

- \* Um framework web assíncrono e escalável, ideal para aplicações que exigem alto desempenho e manipulação de conexões simultâneas.

#### **\* Web2py:**

- \* Um framework web completo, que visa a facilidade de uso e a segurança. Ele fornece ferramentas para desenvolvimento rápido de aplicações web com banco de dados.

Esses frameworks oferecem diferentes abordagens e recursos, permitindo que os desenvolvedores escolham a opção mais adequada para cada projeto.

## **Pandas**

Uma biblioteca Python essencial para manipulação e análise de dados. Aqui estão os pontos-chave sobre o Pandas:

- \* **Propósito Principal:**

- \* O Pandas fornece estruturas de dados de alto desempenho e fáceis de usar, projetadas para trabalhar com dados rotulados ou relacionais.

- \* É amplamente utilizado para limpeza, transformação e análise de dados.

- \* **Estruturas de Dados Fundamentais:**

- \* **Series:** Uma matriz unidimensional rotulada, semelhante a uma coluna em uma planilha ou tabela SQL.

- \* **DataFrame:** Uma estrutura de dados tabular bidimensional, semelhante a uma planilha ou tabela SQL, composta por Series.

- \* **Funcionalidades Principais:**

- \* Leitura e gravação de dados em vários formatos (CSV, Excel, SQL, etc.).

- \* Limpeza e tratamento de dados ausentes.

- \* Filtragem, seleção e indexação de dados.

- \* Agregação e transformação de dados.

- \* Análise estatística e manipulação de séries temporais.

- \* Permite a realização de análise exploratória de dados (EDA).

- \* **Importância:**

- \* O Pandas é uma ferramenta fundamental para cientistas de dados, analistas e engenheiros que trabalham com dados em Python.

- \* É amplamente utilizado em conjunto com outras bibliotecas Python, como NumPy e Matplotlib, para análise e visualização de dados.

- \* **Origem do nome:**

- \* O nome Pandas é derivado do termo "panel data", um termo usado em econometria para conjuntos de dados que incluem várias unidades amostrais acompanhadas ao longo do tempo.

Em resumo, o Pandas simplifica a manipulação e análise de dados em Python, tornando-o uma ferramenta indispensável para qualquer pessoa que trabalhe com dados.

## **semana 6- aula 02**

### **Linguagens de programação back-end – Python**

#### **Primeiros passos com Django**

**Código da aula: [SIS]ANO2C2B1S6A2**

#### **Objetivos da Aula:**

- ❖ Identificar as características do Python para back-end.
- ❖ Identificar as características do Python para back-end.
- ❖ Incentivar a curiosidade ao explorar as características e as vantagens do Python para desenvolvimento back-end.
- ❖ Recurso audiovisual para exibição de vídeos e imagens; • Lápis e caderno para anotações

#### **Exposição**

##### **Definição:**

O Django é um framework web de alto nível, escrito em Python, que permite o desenvolvimento rápido e limpo de aplicações web<sup>1</sup> robustas e seguras. Ele segue o padrão de arquitetura Model-Template-View (MTV) e oferece diversas funcionalidades "de fábrica", como um ORM (Object-Relational Mapper) para interação com bancos de dados, um sistema de templates para geração de HTML dinâmico e uma interface administrativa automática. Em resumo, o Django facilita a construção de sites complexos com menos código e em menos tempo.

Vários sites e aplicações web de alto perfil utilizam o Django como seu framework principal. Alguns exemplos notáveis incluem:

- Redes Sociais: Instagram, Pinterest
- Plataformas de Música: Spotify
- Serviços de Armazenamento em Nuvem: Dropbox
- Organizações de Notícias: The Washington Post, National Geographic
- Plataformas de Discussão: Disqus, Reddit, Quora
- Desenvolvimento de Software: Bitbucket
- Educação: Udemy
- Governo e Ciência: NASA
- Outros: YouTube (algumas partes), Mozilla (alguns sites e ferramentas internas), The Onion, Eventbrite, Prezi.

Essa lista demonstra a versatilidade e escalabilidade do Django, sendo utilizado em projetos de diversos tamanhos e complexidades.

## **semana 6- aula 03**

### **Linguagens de programação back-end – Python**

#### **Primeiros passos com C#**

**Código da aula: [SIS]ANO2C2B1S6A3**

#### **Objetivos da Aula:**

- ❖ Identificar as características do Python para back-end.
- ❖ Identificar as características do Python para back-end.
- ❖ Incentivar a curiosidade ao explorar as características e as vantagens do Python para desenvolvimento back-end.
- ❖ Recurso audiovisual para exibição de vídeos e imagens; • Lápis e caderno para anotações

#### **Exposição**

C# (pronuncia-se "C Sharp") é uma linguagem de programação moderna, orientada a objetos e de propósito geral desenvolvida pela Microsoft<sup>1</sup> como parte da sua plataforma .NET.

Em resumo, o C# é caracterizado por:

- Orientação a Objetos: Suporta os princípios de encapsulamento, herança e polimorfismo.
- Tipagem Forte: Os tipos de dados são definidos explicitamente, o que ajuda a prevenir erros em tempo de execução.
- Multiplataforma: Através do .NET, pode ser usado para desenvolver aplicações para diversas plataformas (Windows, Linux, macOS, mobile, web, jogos, etc.).
- Alto Nível: Oferece abstrações que facilitam o desenvolvimento sem a necessidade de gerenciar detalhes de baixo nível da memória.
- Sintaxe Familiar: Sua sintaxe é influenciada por C, C++ e Java, tornando-a familiar para muitos desenvolvedores.
- Parte do Ecossistema .NET: Integrado com a vasta biblioteca de classes e ferramentas do .NET, o que acelera o desenvolvimento.

É uma linguagem poderosa e versátil, amplamente utilizada no desenvolvimento de diversos tipos de aplicações.

C# é amplamente utilizado em diversos domínios, sendo mais proeminente em:

#### **1. Desenvolvimento de Aplicações Windows:**

- Historicamente, C# tem sido a linguagem principal para criar aplicações desktop para o sistema operacional Windows, utilizando tecnologias como Windows Forms (WinForms) e Windows Presentation Foundation (WPF).

## 2. Desenvolvimento Web com ASP.NET:

- O framework ASP.NET, parte da plataforma .NET, é construído em C# e é uma das tecnologias mais populares para o desenvolvimento de aplicações web robustas, escaláveis e seguras, incluindo APIs e microsserviços.

## 3. Desenvolvimento de Jogos com Unity:

- A Unity, uma das engines de desenvolvimento de jogos mais populares do mundo, utiliza C# como sua principal linguagem de scripting. Isso torna C# essencial para muitos desenvolvedores de jogos, tanto para plataformas desktop quanto mobile e outras.

## 4. Desenvolvimento Mobile com .NET MAUI:

- O .NET Multi-platform App UI (.NET MAUI) permite que desenvolvedores usem C# para criar aplicativos nativos para iOS, Android, macOS e Windows a partir de uma única base de código.

## 5. Desenvolvimento de Serviços e APIs:

- C# é frequentemente usado para construir serviços back-end, APIs (Application Programming Interfaces) e microsserviços que alimentam aplicações web, mobile e desktop.

## Outras áreas importantes incluem:

- Desenvolvimento na Nuvem (Azure): C# tem forte integração com a plataforma de nuvem da Microsoft Azure, sendo usado para desenvolver diversos serviços e aplicações cloud.
- Inteligência Artificial e Machine Learning: Bibliotecas como ML.NET permitem o uso de C# no desenvolvimento de aplicações de machine learning.
- Internet das Coisas (IoT): C# pode ser usado no desenvolvimento de aplicações para dispositivos IoT.
- Aplicações Empresariais: Muitas grandes empresas utilizam C# para desenvolver sistemas corporativos complexos, soluções financeiras, ERPs e CRMs.

Em resumo, C# é uma linguagem versátil com uma forte presença no ecossistema Microsoft, mas sua adoção tem se expandido para outras plataformas graças ao .NET Core (agora apenas .NET), tornando-a uma escolha popular em diversos setores da indústria de tecnologia.