



Plano de Ensino: Módulo 8 Python e C#

Instrutor: João Quentino

Duração: 15 Dias (2h50min por aula)

Objetivo do Curso

Capacitar desenvolvedores a construir APIs robustas, compreendendo as diferenças fundamentais entre linguagens dinâmicas (Python) e estáticas (C#/.NET), com foco em demandas reais de mercado como Machine Learning, Arquitetura Limpa e Segurança.

Estrutura da Aula (2h 50min)

- **40 min:** Teoria Expositiva e Casos de Uso.
- **40 min:** Code-along (Codificando junto com o instrutor).
- **1h 10 min:** Desafio Prático (Mão na massa).
- **20 min:** Review, Correção e Dúvidas.

1 Fase 1: Fundamentos Acelerados e Ferramentas

Dia 01 | Introdução e Setup

Tópicos:

- Instalação (Python, .NET SDK, VS Code).
- Diferença Arquitetural: Interpretado vs Compilado.
- Olá Mundo e Entrada/Saída de dados.
- Variáveis e Tipagem básica.

Prática: Calculadora de 13º Salário e Crachá Digital.

Dia 02 | O Choque de Realidade: Coleções e Decisões

Tópicos:

- Listas (Py) vs Arrays/List<T> (C#).
- Dicionários vs Dictionary<K,V>.
- Introdução ao LINQ (Where, Select) para filtrar dados.
- Estruturas de Decisão (if/else).

Prática: Sistema de Gerenciamento de Estoque (CRUD em memória).

Dia 03 | Controle de Fluxo e Debugging Profissional

Tópicos:

- Loops: for, foreach e while.
- Modularização: Funções e Escopo.
- Tratamento de Erros: try/except vs try/catch.
- Debugging avançado no VS Code (Breakpoints, Watch).

Prática: Automação de Arquivos (Leitura de CSV e Processamento).

Dia 04 | Orientação a Objetos

Tópicos:

- Classes, Atributos e Métodos.
- Encapsulamento e Properties (C#).
- Herança e Polimorfismo.
- O conceito de Injeção de Dependência (DI).

Prática: Modelagem de Classes para E-commerce com Interfaces.

Dia 05 | Assincronismo e Consumo de APIs

Tópicos:

- Programação Assíncrona: async e await.
- Consumindo APIs Externas (Requests vs HttpClient).
- JSON Parsing e Serialização.

Prática: Buscador de CEP e Clima (Consultas paralelas).

2 Fase 2: Especialização de Mercado

Dia 06 | Python para Dados: ETL com Pandas

Tópicos:

- O ecossistema de Data Science.
- **Tech:** Biblioteca Pandas (DataFrames, Series).
- Leitura, Limpeza e Transformação de dados (ETL).

Prática: Processador de Relatórios de Vendas (Excel -> JSON).

Dia 07 | Python para IA: Machine Learning

Tópicos:

- Conceitos de ML: Regressão vs Classificação.
- **Tech:** Scikit-learn.
- Treinando um modelo simples e fazendo previsões.

Prática: Predição de Preços de Imóveis (Regressão Linear).

Dia 08 | C# Corporativo: Entity Framework Core

Tópicos:

- Persistência de Dados em grandes empresas.
- **Tech:** EF Core e abordagem Code-First.
- Conexão com Banco de Dados (SQLite para aula).

Prática: Mapeamento do Banco de Dados do E-commerce.

Dia 09 | C# Avançado: Arquitetura e LINQ

Tópicos:

- Consultas complexas com LINQ (Join, GroupBy).
- Padrão DTO (Data Transfer Object).
- Uso de AutoMapper (Conceito).

Prática: Refatoração da Camada de Dados com DTOs.

3 Fase 3: Construção de APIs Modernas

Dia 10 | O Servidor Web: Hello API

Tópicos:

- **Python:** FastAPI + Pydantic.
- **C#:** ASP.NET Core (Minimal APIs).
- Swagger/OpenAPI (Documentação Automática).

Prática: API "Hello World" estruturada e documentada.

Dia 11 | RESTful: O CRUD Profissional

Tópicos:

- Verbos HTTP: GET, POST, PUT, DELETE.
- Status Codes (200, 201, 400, 404, 500).
- Conectando a API ao Banco de Dados (EF Core / SQLAlchemy).

Prática: API de Gerenciamento de Tarefas (Jira Simplificado).

Dia 12 | Segurança: Autenticação JWT

Tópicos:

- Hashing de Senhas (BCrypt).
- Geração e Validação de Tokens JWT.
- Proteção de Rotas (Authorize).

Prática: Implementar Login e Cadastro na API de Tarefas.

Dia 13 | Qualidade de Código: Testes Automatizados

Tópicos:

- A importância do QA no Backend.
- **Python:** Pytest.
- **C#:** xUnit.
- Testes Unitários de Regra de Negócio.

Prática: Blindando as regras da API de Tarefas com testes.

Dia 14 | Integração Fullstack: O Casamento

Tópicos:

- Configuração de CORS (Cross-Origin Resource Sharing).
- Consumo no React (Axios/Fetch).
- Tratamento de erros do Backend no Frontend.

Prática: Interface React consumindo a API de Tarefas Real.

4 Encerramento

Dia 15 | Hackathon Final

Desafio: Sistema de Agendamento (Ex: Barbearia ou Consultório).

- **Requisito 1:** Backend em C# ou Python (escolha do aluno).
- **Requisito 2:** Banco de Dados persistente.
- **Requisito 3:** Frontend em React funcional.
- **Bônus:** Uso de IA ou Relatórios (Pandas/LINQ).

Contato e Dúvidas

João Quentino

Machine Learning Developer & Instrutor

- **Email:** joao.quentino@dcomp.ufs.br
- **GitHub:** @felipequentino

Sinta-se à vontade para enviar dúvidas sobre quaisquer assuntos