

Proposta de Ementa Otimizada para Regravação do Curso EAD de Programação em Python

1 Introdução

Este documento apresenta uma proposta de ementa otimizada para o curso EAD assíncrono de programação em Python do projeto Desenvolve. O objetivo é reorganizar o conteúdo das apostilas existentes, destacando os principais pontos de aprendizado e estimando o tempo de gravação para cada módulo, garantindo aulas compactas, práticas e alinhadas com os objetivos pedagógicos. A ementa foi estruturada para introduzir conceitos fundamentais de programação, promover a prática com exercícios direcionados e culminar em um projeto final baseado em aprendizagem por projetos (PBL) para consolidar o aprendizado.

2 Estrutura da Ementa Otimizada

2.1 Módulo 1 – Algoritmos, Variáveis e Tipos de Dados

Tempo estimado: 25 minutos

Objetivo: Compreender os fundamentos de algoritmos, variáveis e tipos de dados em Python.

Conteúdo:

- **O que é um algoritmo?** (1 vídeo – 6 min)
Compreender como algoritmos estruturam soluções passo a passo.
- **Variáveis e tipos de dados** (1 vídeo – 9 min)
Aprender a declarar variáveis e identificar os tipos int, float e str.
- **Primeiros comandos: print() e type()** (1 vídeo – 8 min)
Executar os primeiros códigos e visualizar saídas no terminal ou Colab.

Exercício: Programa que recebe nome e idade e calcula o ano de nascimento.

2.2 Módulo 2 – Entrada de Dados e Expressões Aritméticas

Tempo estimado: 25 minutos

Objetivo: Capturar dados do usuário e realizar operações matemáticas básicas.

Conteúdo:

- **A função input() e conversão de tipos** (1 vídeo – 9 min)
Aprender a capturar dados do usuário e transformá-los para os tipos corretos.
- **Operadores aritméticos e precedência** (1 vídeo – 8 min)
Realizar operações matemáticas simples e entender a ordem de execução.
- **Instruções vs Expressões e erros comuns** (1 vídeo – 8 min)
Entender a diferença entre instrução e expressão e identificar erros típicos.

Exercício: Programa que recebe duas notas, calcula a média e exibe o resultado.

2.3 Módulo 3 – Condicionais e Operadores Lógicos

Tempo estimado: 25 minutos

Objetivo: Criar lógicas de decisão com operadores relacionais e lógicos.

Conteúdo:

- **Operadores relacionais e lógicos** (1 vídeo – 7 min)
Comparar valores e aplicar lógica com and, or e not.
- **Condicionais: if, elif, else** (1 vídeo – 10 min)
Criar blocos de decisão simples e compostos.
- **Erros de lógica e boas práticas em decisões** (1 vídeo – 8 min)
Evitar erros comuns e melhorar a clareza das condições.

Exercício: Programa que classifica a média do aluno em aprovado, recuperação ou reprovado.

2.4 Módulo 4 – Laços de Repetição: while e for

Tempo estimado: 25 minutos

Objetivo: Implementar repetições controladas por condição ou faixa numérica.

Conteúdo:

- **Laço while com contadores** (1 vídeo – 8 min)
Criar estruturas de repetição controladas por condição.
- **Laço for com range()** (1 vídeo – 8 min)
Iterar por faixas numéricas de forma simples e controlada.
- **break, continue e controle de fluxo** (1 vídeo – 9 min)
Interromper ou pular iterações com controle explícito.

Exercício: Programa que imprime a tabuada de um número fornecido pelo usuário.

2.5 Módulo 5 – Listas, Strings e Iterações

Tempo estimado: 30 minutos

Objetivo: Manipular listas e strings com iterações e métodos específicos.

Conteúdo:

- **Introdução às listas e seus métodos básicos** (1 vídeo – 8 min)
Criar, acessar e modificar listas.
- **Iterando listas e strings com for** (1 vídeo – 9 min)
Navegar por elementos de sequências com laços.
- **Fatiamento, busca e boas práticas com strings** (1 vídeo – 8 min)
Manipular e extrair informações de strings com eficiência.
- **Exercício aplicado com filtragem de lista** (1 vídeo – 5 min)
Consolidar operações de lista e string em uma lógica prática.

Exercício: Programa que recebe nomes e imprime apenas os que começam com uma letra específica.

2.6 Módulo 6 – Funções e Organização de Código

Tempo estimado: 25 minutos

Objetivo: Criar funções reutilizáveis e escrever códigos bem estruturados.

Conteúdo:

- **Definindo funções com def()** (1 vídeo – 9 min)
Criar blocos reutilizáveis de código com parâmetros.
- **Retorno de valores e escopo de variáveis** (1 vídeo – 8 min)
Utilizar return e entender onde variáveis podem ser acessadas.
- **Organização e legibilidade do código** (1 vídeo – 8 min)
Escrever códigos limpos, reutilizáveis e bem estruturados.

Exercício: Criar uma função calcular_imc(peso, altura) e exibir o resultado com classificação.

2.7 Projeto Final – Cadastro de Usuários com Relatórios

Tempo estimado: 30–40 minutos

Objetivo: Integrar todos os conceitos aprendidos em um projeto prático baseado em aprendizagem por projetos (PBL).

Conteúdo:

- **Coleta e armazenamento de dados em lista de dicionários** (1 vídeo – 15 min)
Receber múltiplos cadastros contendo nome, idade, peso e altura.
- **Cálculo de IMC, média de idade e relatório final** (1 vídeo – 20–25 min)
Processar os dados armazenados e exibir estatísticas simples.

3 Resumo e Benefícios

A ementa proposta reorganiza o conteúdo das apostilas em 6 módulos e um projeto final, totalizando 155–165 minutos (aproximadamente 2 horas e 35–45 minutos) de gravação. Cada módulo foi projetado para ser conciso, com vídeos curtos (média de 5–10 minutos) e objetivos claros, promovendo engajamento e aprendizado progressivo. Os exercícios práticos, como o cálculo do ano de nascimento, tabuada e classificação de IMC, reforçam a aplicação dos conceitos. O projeto final baseado em PBL integra todos os tópicos, incentivando a autonomia e a resolução de problemas.

4 Considerações Finais

A ementa otimizada reduz redundâncias, foca nos conceitos essenciais e alinha o conteúdo aos objetivos pedagógicos do curso. O projeto final garante que os alunos apliquem os conhecimentos de forma prática e significativa. Recomenda-se a validação com a equipe pedagógica e a realização de um piloto para ajustar os tempos de gravação, se necessário. Para dúvidas ou ajustes, estou à disposição.