



## Ementa

### Curso de Extensão: *Python* e Suas Aplicações para Engenheiros

**Carga horária total:** 20 horas

**Modalidade:** Presencial.

**Público-alvo:** Estudantes e profissionais de Engenharia sem experiência prévia em programação.

**Pré-requisitos:** Conhecimentos básicos de informática, incluindo uso do computador e navegação na internet.

---

#### 1. Justificativa

Este curso foi elaborado seguindo as melhores práticas de instituições internacionais de excelência em engenharia, que reconhecem a programação como competência fundamental na formação contemporânea de engenheiros.

Nossa proposta combina acessibilidade e aplicação prática, oferecendo uma imersão em *Python* voltada para desafios reais da Engenharia de Produção: desde otimização de processos e logística até análise de dados estratégicos para tomada de decisão.

#### 2. Objetivo Geral

Proporcionar aos alunos uma introdução acessível à programação em *Python*, desenvolvendo raciocínio lógico, compreensão de algoritmos e uso de bibliotecas essenciais para manipular dados e criar soluções simples aplicadas a problemas de engenharia.

### 3. Objetivos do Curso

Ao final do curso, espera-se que os participantes sejam capazes de:

- **Desenvolver o raciocínio lógico e computacional**, aplicando-o à resolução de problemas de Engenharia.
- **Compreender os conceitos básicos de programação**, incluindo variáveis, estruturas condicionais e laços de repetição.
- **Introduzir-se aos paradigmas de programação em *Python***, entendendo sua aplicação prática em engenharia.
- **Familiarizar-se com os ambientes de desenvolvimento (*IDE* e terminal)** e com o uso inicial do ***GitHub*** para versionamento de código.
- **Entender os conceitos fundamentais de algoritmos e estruturas de dados**, aplicados a situações simples.
- **Utilizar bibliotecas essenciais do *Python* (*pandas*, *matplotlib*)** para manipulação de dados e criação de gráficos.
- **Manipular planilhas, arquivos e dados textuais**, automatizando tarefas repetitivas.
- **Aplicar a programação para resolver problemas introdutórios típicos da Engenharia de Produção**, desenvolvendo soluções simples e funcionais.

## 4. Metodologia

A metodologia adotada busca proporcionar um ambiente de aprendizado acessível, dinâmico e prático, especialmente voltado para iniciantes em programação, por meio das seguintes abordagens:

- **Aprendizagem progressiva:** introdução gradual aos conceitos de programação, desde a lógica computacional até a aplicação de bibliotecas essenciais, com exercícios.
- **Abordagem *hands-on*:** desenvolvimento de códigos simples desde o primeiro contato, aplicando imediatamente os conceitos apresentados em situações práticas.
- **Aprendizagem ativa:** dinâmicas em sala de aula, realizadas individualmente e em equipe, além de estudos de caso curtos para reforço dos conteúdos trabalhados.
- **Didática inclusiva:** abordagem acessível, promovendo o desenvolvimento gradual da confiança e autonomia na programação, sem exigir experiência prévia.

## 5. Avaliação

Para serem elegíveis à aprovação e recebimento do certificado de conclusão do curso, os alunos deverão atingir, no mínimo, 70% da pontuação total e registrar 75% de frequência nas aulas.

Critério	Peso
Participação e execução dos exercícios práticos (em sala)	25%
Realização dos desafios semanais (em casa)	25%
Entrega e apresentação do projeto final	50%

## 6. Recursos Didáticos

- **Slides de apoio às aulas:** apresentação dos conceitos e exemplos abordados em cada encontro.
- **Repositório no *GitHub*:** disponibilização de exercícios práticos, materiais complementares, links para documentações oficiais, vídeos educativos e plataformas de prática de programação (ex.: LeetCode).
- **Ambientes de desenvolvimento:** utilização do Google Colab) e do Visual Studio Code para implementação dos códigos e realização das atividades propostas.
- **Grupo do *Whatsapp*:** para interação entre os alunos, anúncios e esclarecimento de dúvidas

## 7. Competências a Desenvolver

- **Técnicas:** compreensão dos fundamentos da lógica de programação, manipulação de dados, uso de bibliotecas básicas do *Python* (*pandas*, *matplotlib*) e aplicação prática em problemas simples de engenharia.
- **Gerenciais:** desenvolvimento do pensamento lógico-analítico, organização de informações, automação de tarefas básicas e utilização de ferramentas de versionamento de código (*GitHub*).
- **Transversais:** capacidade de resolução de problemas, raciocínio lógico aplicado à engenharia, colaboração em atividades práticas e comunicação de resultados por meio de código e gráficos.

## 8. Plano de Aulas

### Aula 1

**Tema:** Fundamentos e Ambiente de Desenvolvimento

**Data:** 13/09/2025

**Duração:** 08:00 as 12:00

**Local:** Laboratório de informática

**Recursos necessários:** projetor multimídia, computadores com navegador e acesso à internet, *Visual Studio Code* e *Python* instalados.

#### Conteúdo programático da aula

- **Apresentação do curso:** objetivos, cronograma e expectativas de aprendizagem.
- **Introdução à plataforma *GitHub*:** Instruções sobre acesso aos materiais do curso disponibilizados na plataforma.
- **Contextualização do uso da programação na engenharia:** importância, aplicações práticas e benefícios do *Python* na área.
- **Conceitos básicos de programação:** algoritmos, lógica computacional, linguagens de programação e casos de uso do *Python* em engenharia.
- **Operadores:** aritméticos.
- **Introdução aos ambientes de desenvolvimento (*IDE*):** conceito e primeiros passos no Visual Studio Code.
- **Primeiros exercícios práticos:** escrever e executar programas básicos no *Python* (mensagens na tela, operações matemáticas simples).
- **Variáveis e tipos de dados:** espaços nomeados na memória para armazenar dados.
- **Entrada e saída de dados no *Python*.**

#### Estrutura da aula

- **Exposição teórica intercalada com exercícios práticos:** alternância entre explicação dos conceitos e aplicação imediata.
- **Intervalo:** pausa para descanso e socialização dos alunos.
- **Retomada das atividades:** continuação dos exercícios práticos e esclarecimento de dúvidas pontuais.
- **Encerramento:** revisão dos tópicos trabalhados e espaço para perguntas finais.

## Aula 2

**Tema:** Primeiros passos com a linguagem

**Data:** 20/09/2025

**Duração:** 08:00 as 12:00

**Local:** Laboratório de informática

**Recursos necessários:** projetor multimídia, computadores com navegador e acesso à internet, *Visual Studio Code* e *Python* instalados.

### Conteúdo programático da aula

- **Revisão da aula anterior:** Espaço dedicado à apresentação das resoluções dos exercícios práticos, discussão de dúvidas, aprimoramento de códigos e compartilhamento de diferentes abordagens de solução.
- **Operadores:** relacionais e lógicos.
- **Estruturas condicionais:** *if*, *elif*, *else*.
- **Laços de repetição:** *for*, *while*.
- **Exercícios práticos:** aplicação dos conceitos aprendidos em problemas simples de engenharia (ex.: cálculo de eficiência, médias de produção)
- **Encerramento:** revisão dos tópicos trabalhados e espaço para perguntas finais.

### Estrutura da aula

- **Revisão inicial:** breve recapitulação dos principais conceitos abordados na aula anterior através da resolução de exercícios.
- **Exposição teórica intercalada com exercícios práticos:** alternância entre explicação dos conceitos e aplicação imediata.
- **Intervalo:** pausa para descanso e socialização dos alunos.
- **Retomada das atividades:** continuação dos exercícios práticos e esclarecimento de dúvidas pontuais.
- **Encerramento:** revisão dos tópicos trabalhados e espaço para perguntas finais.

### Aula 3

**Tema:** Estruturas de dados, manipulação de informações, introdução a bibliotecas e ao Google Colab.

**Data:** 27/09/2025

**Duração:** 08:00 as 12:00

**Local:** Laboratório de informática

**Recursos necessários:** projetor multimídia, computadores com navegador e acesso à internet, *Visual Studio Code* e *Python* instalados.

#### Conteúdo programático da aula

- **Revisão da aula anterior:** Espaço dedicado à apresentação das resoluções dos exercícios práticos, discussão de dúvidas, aprimoramento de códigos e compartilhamento de diferentes abordagens de solução.
- **Listas, tuplas e dicionários:** criação, acesso, modificação e iteração.
- **Operações básicas de manipulação de dados:** inserção, remoção, ordenação e busca de informações em listas e dicionários.
- **Exercícios práticos aplicados:** organização e análise simples de dados de produção utilizando estruturas de dados.
- **Primeiros passos no Google Colab:** conhecendo a interface, o formato de *notebooks* e sua utilização prática.
- **Introdução a bibliotecas:** conceito e utilização para criação de gráficos básicos com *matplotlib* para visualização de dados.

#### Estrutura da aula

- **Revisão inicial:** breve recapitulação dos principais conceitos abordados na aula anterior através da resolução de exercícios.
- **Exposição teórica intercalada com exercícios práticos:** alternância entre explicação dos conceitos e aplicação imediata.
- **Intervalo:** pausa para descanso e socialização dos alunos.
- **Retomada das atividades:** continuação dos exercícios práticos e esclarecimento de dúvidas pontuais.
- **Encerramento:** revisão dos tópicos trabalhados e espaço para perguntas finais.

## Aula 4

**Tema:** Funções e análise de dados com o *Pandas*.

**Data:** 04/10/2025

**Duração:** 08:00 as 12:00

**Local:** Laboratório de informática

**Recursos necessários:** projetor multimídia, computadores com navegador e acesso à internet, *Visual Studio Code* e *Python* instalados.

### Conteúdo programático da aula

- **Revisão da aula anterior:** Espaço dedicado à apresentação das resoluções dos exercícios práticos, discussão de dúvidas, aprimoramento de códigos e compartilhamento de diferentes abordagens de solução.
- **Funções em *Python*:** definição, parâmetros, argumentos e retorno de valores.
- **Introdução à biblioteca *pandas*:** leitura de arquivos CSV, visualização de tabelas e operações simples de filtragem e seleção de dados.
- **Integração com *matplotlib*:** criação de gráficos a partir de dados tabulares.
- **Proposta do projeto final:** explicação do desafio a ser desenvolvido, orientações sobre uso do *GitHub* para Armazenamento e organização do trabalho.

### Estrutura da aula

- **Revisão inicial:** breve recapitulação dos principais conceitos abordados na aula anterior através da resolução de exercícios.
- **Exposição teórica intercalada com exercícios práticos:** alternância entre explicação dos conceitos e aplicação imediata.
- **Intervalo:** pausa para descanso e socialização dos alunos.
- **Retomada das atividades:** continuação dos exercícios práticos e esclarecimento de dúvidas pontuais.
- **Encerramento:** revisão dos tópicos trabalhados e espaço para perguntas finais.



## Aula 5

**Tema:** Apresentação do projeto final

**Data:** 11/10/2025

**Duração:** 08:00 as 12:00

**Local:** Laboratório de informática

**Recursos necessários:** projetor multimídia, computadores com navegador e acesso à internet, *Visual Studio Code* e *Python* instalados.

### Conteúdo programático da aula

- **Revisão da aula anterior:** Espaço dedicado à apresentação das resoluções dos exercícios práticos, discussão de dúvidas, aprimoramento de códigos e compartilhamento de diferentes abordagens de solução.
- **Apresentação dos projetos:** exposição breve dos trabalhos pelos participantes, destacando raciocínio lógico, organização do código e resultados obtidos.
- **Discussão coletiva:** feedbacks e troca de experiências entre os participantes.
- **Encerramento do curso:** principais aprendizados, próximos passos para evolução na programação e materiais complementares para estudo.

### Estrutura da aula

- **Revisão inicial:** breve recapitulação dos principais conceitos abordados na aula anterior através da resolução de exercícios.
- **Apresentação do projeto final:** sessão demonstrativa onde as equipes exibem seus trabalhos, explicam a solução implementada e compartilham aprendizados técnicos e desafios superados durante o desenvolvimento.
- **Encerramento do curso:** entrega dos certificados de conclusão, orientações sobre *next steps* na jornada de programação e incentivo à manutenção do portfólio no GitHub.