

Simplifica Python



---

# APOSTILA

---

## SEU PRIMEIRO PROJETO COMPLETO EM PYTHON



**Alexandre Sauer**

@prof.alesauer

- Especialista em Python
- Mestre em Sistemas da Informação

# SUMÁRIO

SEU PRIMEIRO PROJETO EM PYTHON	03
CAPÍTULO 1: INTRODUÇÃO	04
CAPÍTULO 2: CONFIGURAÇÃO DO AMBIENTE	06
CAPÍTULO 3: ESTRUTURA DO CÓDIGO	07
CAPÍTULO 4: BIBLIOTECAS NECESSÁRIAS	09
CAPÍTULO 5: CRIANDO O PROJETO	11
CAPÍTULO 6: PERSONALIZANDO O CERTIFICADO	14
CAPÍTULO 7: GERANDO O CERTIFICADO	16
CONCLUSÃO	18



# SEU PRIMEIRO PROJETO EM PYTHON

Nesta aula, iremos desenvolver um projeto para **geração de certificados de cursos em PDF**, igual ao demonstrado na imagem abaixo.

Você será guiado passo a passo para entender os recursos necessários em Python para já, nesta aula, sair do Zero até criar o seu primeiro projeto.



# CAPÍTULO 1: INTRODUÇÃO

No mundo atual, a automação de tarefas repetitivas tornou-se uma necessidade para aumentar a eficiência e garantir a precisão nos processos. Uma dessas tarefas é a geração de certificados de conclusão de curso para alunos, algo essencial em qualquer programa educacional.

Embora possa ser feita manualmente, essa atividade pode ser bastante demorada e sujeita a erros, especialmente quando o número de alunos é grande.

Este eBook apresenta uma solução prática para esse desafio, utilizando a linguagem de programação Python.

Python é uma das linguagens mais versáteis e poderosas disponíveis hoje, amplamente utilizada em diversas áreas, incluindo automação, análise de dados, desenvolvimento web, e muito mais.

# CAPÍTULO 1: INTRODUÇÃO

Sua simplicidade e a vasta quantidade de bibliotecas disponíveis tornam Python a escolha ideal para a criação de scripts que automatizam tarefas como a geração de certificados.

Neste guia, você aprenderá como construir um sistema que lê os dados dos alunos de um arquivo CSV e gera automaticamente certificados personalizados em formato PDF. Vamos explorar passo a passo como configurar o ambiente, entender o código, personalizar os certificados, e finalmente, como executar o projeto de forma eficiente.

Ao final deste eBook, você não apenas terá criado uma ferramenta poderosa para gerar certificados, mas também terá expandido seu conhecimento em Python e na automação de tarefas. Este é um recurso valioso tanto para desenvolvedores iniciantes quanto para educadores que buscam otimizar seus processos administrativos.

## CAPÍTULO 2: CONFIGURAÇÃO DO AMBIENTE

Antes de começarmos a desenvolver o projeto, é necessário garantir que o ambiente de desenvolvimento esteja corretamente configurado. Para isso, você precisará instalar o Python e o Visual Studio Code.

Caso ainda não tenha feito isso, você pode seguir o passo a passo detalhado no vídeo [Como Instalar e Configurar seu Ambiente de Programação Python](#).

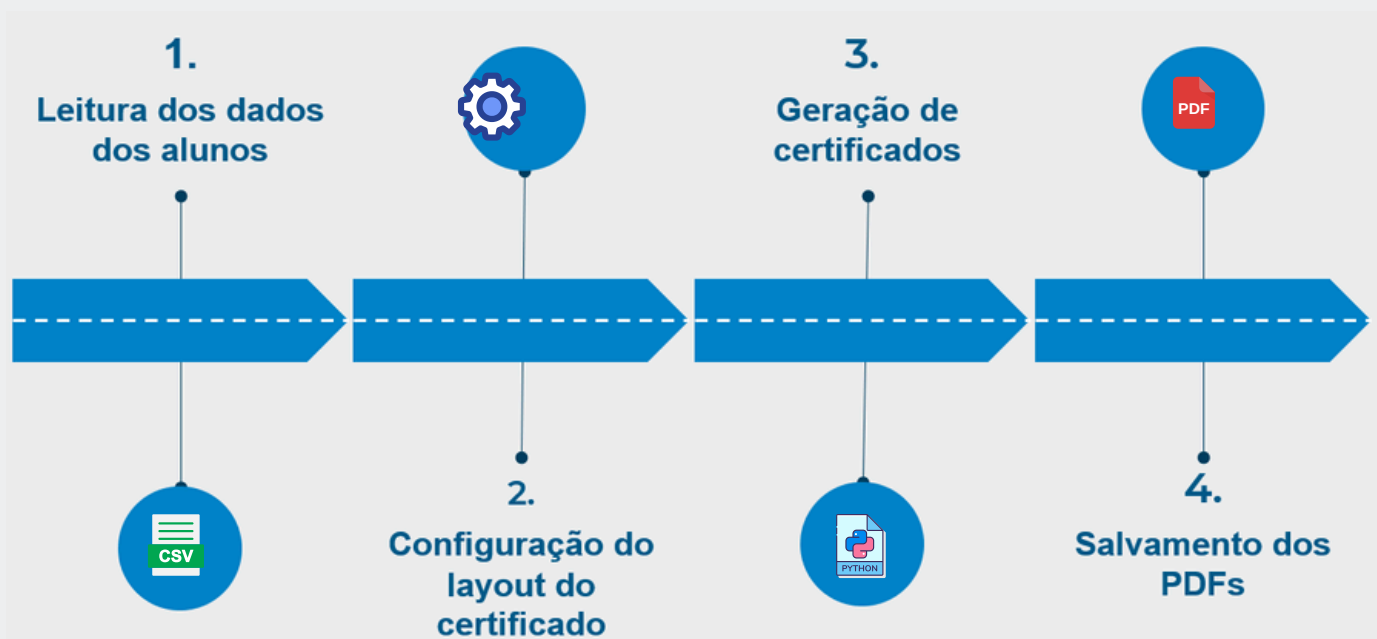


O vídeo fornecerá todas as instruções necessárias para:

- Baixar e instalar o Python em seu computador.
- Configurar o Visual Studio Code para trabalhar com projetos em Python.
- Verificar se o ambiente está corretamente configurado.

# CAPÍTULO 3: ESTRUTURA DO CÓDIGO

Antes de mergulharmos no código propriamente dito, é importante entender a lógica da aplicação e os passos que serão seguidos para gerar os certificados de conclusão de curso. A aplicação seguirá a seguinte sequência lógica:



# CAPÍTULO 3: ESTRUTURA DO CÓDIGO

Explicando passo a passo o que iremos desenvolver:

1. **Leitura dos dados dos alunos:** O programa começará lendo os dados dos alunos de um arquivo CSV. Esses dados incluem informações essenciais, como nome completo, que serão usadas para personalizar os certificados.
2. **Configuração do layout do certificado:** Em seguida, o layout do certificado será configurado, definindo textos como o título, subtítulo, e informações do curso. Além disso, uma imagem de fundo (template) será inserida.
3. **Geração de certificados individuais:** O código percorrerá cada registro no arquivo CSV e, para cada aluno, criará um PDF personalizado. O nome do aluno e outras informações serão inseridos no lugar correto no certificado.
4. **Salvamento dos PDFs:** Por fim, cada certificado será salvo como um arquivo PDF individualmente nomeado.



## CAPÍTULO 4: BIBLIOTECAS NECESSÁRIAS

A primeira etapa para configurar seu ambiente de desenvolvimento é instalar as bibliotecas que serão utilizadas para gerar os certificados e manipular os dados dos alunos. Vamos usar duas bibliotecas muito utilizadas pelos desenvolvedores:

- **pandas**: é uma biblioteca essencial para manipulação e análise de dados em Python. Vamos usá-la para ler o arquivo CSV com os dados dos alunos.
- **fpdf**: é uma ferramenta poderosa para criar documentos PDF em Python. Ela permite adicionar texto, imagens e outros elementos ao PDF de forma simples e intuitiva.
- Para instalação destas bibliotecas, iremos usar o comando **pip** que vem instalado no python. Caso não saiba como proceder, acesse o link do vídeo que disponibilizei para relizar [\[LINK DO VÍDEO\]](#)

## CAPÍTULO 4: BIBLIOTECAS NECESSÁRIAS

Para instalar as bibliotecas necessárias para desenvolvermos o projeto, o **pandas** e o **fpdf**. Usaremos o comando pip. A sintaxe (forma correta de escrever) do comando é:

**pip install <nome da biblioteca>**

Vamos então partir para a instalação. Em seu terminal, digite os comandos necessários para instalar as bibliotecas:

1. pip install pandas
2. pip install fpdf

Se tiver alguma dificuldade em instalar as bibliotecas, reveja a aula 01 que disponibilizamos nessa jornada.

## CAPÍTULO 5: CRIANDO O PROJETO

Antes de começarmos a escrever o código, vamos organizar nosso ambiente de trabalho. Para facilitar o desenvolvimento e a execução do projeto, siga as instruções abaixo:

1. Crie uma pasta para o projeto: No diretório C:\, crie uma nova pasta chamada python, se ela ainda não existir. Dentro da pasta python, crie uma subpasta chamada aula01. Esta será a pasta onde todo o código e os arquivos relacionados ao projeto serão armazenados.
2. Crie o arquivo Jupyter Notebook: Dentro da pasta aula01, crie um novo arquivo Jupyter Notebook e nomeie-o como gera\_certificado.ipynb. Este arquivo será onde você escreverá e executará o código que gera os certificados de conclusão de curso.

Com o ambiente organizado, vamos explorar a estrutura do código que você utilizará para gerar os certificados.

# CAPÍTULO 5: CRIANDO O PROJETO

## Importação das Bibliotecas:

Para construir nossa aplicação, utilizaremos duas bibliotecas principais: `fpdf` e `pandas`. O código para importar essas bibliotecas é simples:

```
1. from fpdf import FPDF  
2. import pandas as pd
```

Com essas bibliotecas importadas, estamos prontos para começar a trabalhar com os dados dos alunos.

## Leitura dos Dados:

O próximo passo é ler os dados dos alunos, que estão armazenados em um arquivo CSV. Esse arquivo contém informações como o nome completo dos alunos, que será usado para personalizar cada certificado.

Usaremos o `pandas` para ler e manipular esses dados. A função `read_csv` permite carregar o arquivo CSV de forma fácil e eficiente. Veja como isso é feito:

```
1. dados = pd.read_csv("dados.csv")
```

## CAPÍTULO 5: CRIANDO O PROJETO

Com este comando, o conteúdo do arquivo dados.csv é carregado em um DataFrame, que é uma estrutura de dados semelhante a uma tabela. A partir desse DataFrame, podemos acessar e manipular os dados de cada aluno, preparando-os para serem usados na geração dos certificados.

Agora que os dados foram lidos e estão prontos para uso, vamos avançar para a personalização e geração dos certificados, que será abordada nos próximos capítulos.

# CAPÍTULO 6: PERSONALIZANDO O CERTIFICADO

Vamos definir o conteúdo textual do certificado e como ajustar a aparência do documento, garantindo que cada certificado seja único e visualmente agradável.

## Definição dos Textos do Certificado

A primeira etapa na personalização do certificado é definir os textos que aparecerão em cada documento.

Esses textos incluem:

- **Título:** O título principal do certificado, como "CERTIFICADO DE PARTICIPAÇÃO".
- **Subtítulo:** Uma breve declaração, como "Este certificado comprova que".
- **Nome do Aluno:** Este é o campo que será preenchido dinamicamente com o nome de cada aluno, lido do arquivo CSV.
- **Informações do Curso:** Texto que descreve o curso, incluindo detalhes como o nome do instrutor, as datas do curso e a carga horária.

# CAPÍTULO 6: PERSONALIZANDO O CERTIFICADO

Veja como esses textos podem ser definidos no código:

```
1. titulo = "CERTIFICADO DE PARTICIPAÇÃO"  
2. subtítulo = "Este certificado comprova que"  
3. texto2 = "concluiu com êxito o curso GRATUITO DE PYTHON ministrado por"  
4. texto3 = "PROF. SAUER entre xx/xx/xxxx e xx/xx/xxxx,"  
5. texto4 = "com carga horária de aproximadamente xx horas."
```

Essas variáveis armazenam as informações básicas que aparecerão em todos os certificados. O nome do aluno será adicionado a essas informações durante a geração do certificado.

# CAPÍTULO 7: GERANDO O CERTIFICADO

Com o layout do certificado definido e os textos configurados, o próximo passo é automatizar a geração dos certificados para todos os alunos listados no arquivo CSV. Isso é feito utilizando um loop for, que percorre cada nome de aluno e cria um certificado personalizado. Vamos detalhar como isso funciona:

```
1. # Loop para gerar certificados para cada participante
2. for nome in dados['nomecompleto']:
3.     # Criação do objeto PDF
4.     pdf = FPDF()
5.     pdf.add_page()
6.     pdf.set_font("Arial", 'B', size=15)
7.
8.     # Adicionando a imagem de fundo (template)
9.     pdf.image("template.png", x=0, y=0)
10.
11.    # Configurando a cor do texto
12.    pdf.set_text_color(33, 24, 136)
13.    # Adicionando o texto ao PDF
14.    pdf.text(65, 95, titulo)
15.    pdf.text(67, 120, subtítulo)
16.    pdf.text(70, 145, nome)
17.    pdf.text(20, 165, texto2)
18.    pdf.text(50, 175, texto3)
19.    pdf.text(50, 185, texto4)
20.
21.    # Salvando o PDF com um nome único
22.    pdf.output(f"Certificado_{nome}.pdf")
```



# CAPÍTULO 7: GERANDO O CERTIFICADO

## Iniciando o Loop:

```
2. for nome in dados['nomecompleto']:
```

Este loop for percorre cada nome na coluna 'nomecompleto' do DataFrame dados. Para cada nome, o código dentro do loop será executado, gerando um certificado exclusivo.

## Criação do Objeto PDF:

```
4. pdf = FPDF()  
5. pdf.add_page()  
6. pdf.set_font("Arial", 'B', size=15)
```

Dentro do loop, um novo objeto PDF é criado para cada aluno. Em seguida, uma página é adicionada ao PDF, e a fonte é configurada. Aqui, estamos usando a fonte Arial em negrito com tamanho 15.

## Adicionando a Imagem de Fundo:

```
9. pdf.image("template.png", x=0, y=0)
```

A imagem de fundo (template) é inserida na página. Isso garante que todos os certificados tenham o mesmo layout visual, com o texto personalizado sobreposto.

# CONCLUSÃO

Você foi guiado passo a passo no desenvolvimento de um projeto Python para gerar certificados de conclusão de curso de forma automatizada.

Este projeto é um excelente exemplo de como Python pode ser utilizado para automatizar tarefas repetitivas e garantir precisão e eficiência. Ao concluir este eBook, você não apenas aprendeu a gerar certificados, mas também adquiriu habilidades valiosas em manipulação de dados, criação de documentos PDF e automação de processos.

Esperamos que esta aula tenha sido útil e inspirador. O conhecimento de automação com Python é uma ferramenta poderosa que pode ser aplicada a inúmeros outros projetos e cenários. Continue explorando, aprendendo e criando soluções inovadoras com Python!

PROF.

SAUER

# IMPULSIONE A SUA CARREIRA



**Simplifica**  
TREINAMENTOS