

# Módulo de Programação Python

## Trilha Python - Aula 16/17: Revisão e Resolução de Problemas



### Objetivo:

Revisar e consolidar o conteúdo de **NumPy** avançado e introdução a **Pandas**

### Revisão do exercício proposto para determinação do perfil da turma.

Na última instrução prática foram coletados dados dos membros de cada equipe. Dado o pequeno tamanho das equipes a coleta pode ser feita manualmente. Entretanto, para coletar os dados de todos os alunos da residência precisamos melhorar o processo de coleta. Vamos desenvolver um aplicativo para esta coleta.

1. Primeiramente vamos criar um ambiente isolado que possa ser reproduzido posteriormente. Com esta finalidade escolha a ferramenta a ser adotada pela equipe e criem um ambiente virtual contendo os pacotes **Python** 3.10 ou superior, **NumPy** e **Pandas**. Crie o arquivo `requirements.txt` contendo as versões dos pacotes.
2. Criar uma pasta para o pacote `restic`. Que arquivos serão necessários para configurar e executar a aplicação que desenvolveremos?

```
In [ ]: identificador = []  
idade = []  
formação = []  
formaçãoGera1 = []  
formaçãoEspecífica = []  
andamentoGraduação = []  
tempoFormação = []  
experiênciaPrevia = []
```

## Analizando a coleta de dados

3. O identificador de cada aluno será, como definido anteriormente, um código de tipo `string` formado `tic18Py` seguido dos três primeiros números do **CPF** e os dois últimos números do ano de nascimento. Exemplo `tic18Py06472`. Para poder contemplar os restantes alunos da residência precisamos teremos que disponibilizar três raízes diferentes para os identificador. Uma proposta pode ser:
  - `tic18Py` para a turma de **Python**
  - `tic18Net` para a turma de **.NET**
  - `tic18Jav` para a turma de **Java**
4. A idade, próximo campo a ser preenchido, tem que bater com o ano de nascimento. Isto significa que uma pessoa que nasceu em 1990, quando questionado em 2024 sobre sua idade, pode responder que tem 33 ou 34 anos, dependendo da data exata de nascimento.
5. A campo `formação` tem um conjunto restrito de opções que podem ser traduzidas em códigos numéricos a serem armazenados. Já a `formação geral` pode ser armazenada como um boolean mas depende da resposta sobre a `formação`.
6. Veja como implementar o campo de `formação específica` de acordo com o que foi definido pela equipe na instrução prática anterior.
7. O campo `andamento da graduação` e `tempo de formação` são campos excludentes. Isto é, se preenche um o outro não se aplica.
8. O campo final, `experiência previa de programação` pode ser melhorado? Analisando a informação coletada até aqui, a resposta `True` ou `False` permitem caracterizar realmente a experiência dos entrevistados?

## Trabalhando com classes e objetos.

9. Após a análise anterior podemos desenhar classes específicas para armazenar:

- Os dados de um entrevistado de forma a garantir que eles estejam sempre preenchidos corretamente.
- Os dados dos entrevistados numa trilha específica. Esta classe tem um atributo de tipo `DataFrame` que será preenchido com os dados coletados de cada entrevistado. Entretanto o identificador dos entrevistados desta classe será único.
- Os dados de uma turma da residência com várias trilhas diferentes.

Vamos discutir a implementação destas classes.