



# **INSTRUÇÃO PRÁTICA - IP-P002**

## **PP - PROGRAMAÇÃO EM PYTHON**

### **OBJETIVOS DA ATIVIDADE**

Avaliar as limitações dos tipos de dados básicos em python e a alternativa oferecida pelos **ndarrays** do pacote **NumPy**, na manipulação de grandes volumes de dados. Trabalhar com os atributos e métodos disponibilizados pelas classes do **NumPy**. Explorar as diferentes formas de acessar matrizes e vetores criados com **NumPy**.

### **Contextualização**

Na atividade anterior trabalhamos com a implementação de um conjunto de classes para armazenar diferentes tipos de dados visando a realização posterior de análise estatística dos mesmos. Estes dados foram armazenados internamente na forma de listas **Python**. Entretanto, o desempenho do processamento das listas é um fator limitante, particularmente quando se trabalha com grandes volumes de dados.

## Exercício 1: avaliando o desempenho das listas

As classes `ListaIdade` e `ListaSalarios` armazenam tipos de dados inteiros e de ponto flutuante. Vamos explorar o que acontece quando trabalhamos com grandes volumes de dados.

- Implemente uma função `geraListaIdades` para a qual você fornece a quantidade de itens a ser gerados e ela retorna um objeto da classe `ListaIdade` com idades geradas de forma aleatória. A função tem como parâmetro de entrada também a idade mínima e máxima, ou seja o intervalo em que as idades devem ser geradas. Se o usuário não fornecer o intervalo a função gera idades entre 18 e 65 anos.

**Observação:** Repare que não conseguimos utilizar o método da classe `entradaDeDados` para adicionar as idades, geradas de forma aleatória, na lista. Será necessário então criar novos métodos ou modificar os atuais.

**Sugestão:** Modificar o construtor da classe para que e possa receber uma lista de inteiros e construa a instância a partir desta lista.

**Sugestão:** Explore o pacote `random` e a função `randint` do mesmo.

```
In [ ]: def geraListaIdade(n, iMin = 18, iMax = 65):  
        pass
```

- Implemente uma função `geraListaSalario` para a qual você fornece a quantidade de itens a ser gerados e ela retorna um objeto da classe `ListaSalario` com salários gerados de forma aleatória. A função tem como parâmetro de entrada também o salário mínimo e máximo, ou seja o intervalo em que os salários devem ser gerados. Se o usuário não fornecer o intervalo a função gera valores entre um e 10 salários mínimos.

**Observação:** Repare que não conseguimos utilizar o método da classe `entradaDeDados` para adicionar os salários, gerados de forma aleatória, na lista. Será necessário então criar novos métodos ou modificar os atuais.

**Sugestão:** Modificar o construtor da classe para que e possa receber uma lista de valores de ponto flutuante e construa a instância a partir desta lista.

**Sugestão:** Explore o pacote `random` e a função `random` do mesmo.

```
In [ ]: # Pesquisa sobre o valor do salário mínimo no Brasil  
def geraListaSalarios(n, sMin = 1000, sMax = 5000):  
    pass
```

- Agora vamos testar o desempenho das classes implementadas.
  - Gere um objeto da classe `Listaldade` com 5000 idades.
  - Com ajuda do comando mágico `%timeit` identifique quanto tempo demora para executar o método `mostraMediana`.
  - Com ajuda do comando mágico `%timeit` identifique quanto tempo demora para executar o método `mostraMenor`.
  - Com ajuda do comando mágico `%timeit` identifique quanto tempo demora para executar o método `mostraMaior`.
  - Gere um objeto da classe `ListaSalario` com 5000 salários.
  - Com ajuda do comando mágico `%timeit` identifique quanto tempo demora para executar o método `mostraMediana`.
  - Com ajuda do comando mágico `%timeit` identifique quanto tempo demora para executar o método `mostraMenor`.
  - Com ajuda do comando mágico `%timeit` identifique quanto tempo demora para executar o método `mostraMaior`.

## Exercício 2: Utilizando NumPy

Antes de fazer modificações nas classes exploradas no exercício anterior.

- Pesquise na documentação de **NumPy** pelos recursos do módulo `random` e veja se temos uma alternativa para gerar um `ndarray` com valores inteiros aleatórios num determinado intervalo (idade mínima e máxima).
- Vamos criar um `ndarray` a partir da `Listaldade` criada no exercício anterior.
- Pesquise na documentação de **NumPy** pelos recursos do módulo `random` e veja se temos uma alternativa para gerar um `ndarray` com valores de ponto flutuante aleatórios num determinado intervalo (salário mínimo e máximo).
- Vamos criar um `ndarray` a partir da `ListaSalario` criada no exercício anterior.
- Pesquise na documentação de **NumPy** pelos recursos disponíveis para determinar a mediana, o valor mínimo e máximo de um `ndarray`. Compare o desempenho dos recursos disponíveis com o dos métodos testados no exercício anterior.

## Fechamento da atividade

Utilizem o **Discord** para tirar suas dúvidas sobre as atividades e interagir com os membros da sua equipe, com outros residentes, com a tutora ou com o professor; Garanta que seu repositório do **GitHub** esteja atualizado com todos os branch e versão final do branch principal.



Universidade  
Estadual de  
Santa Cruz



MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA  
E INOVAÇÃO



