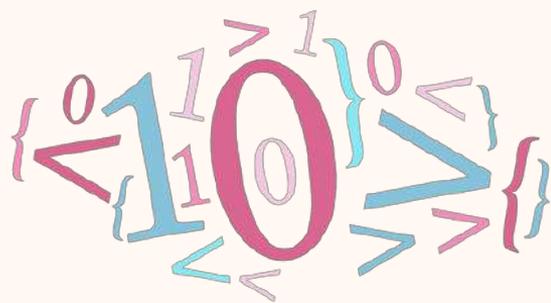




Uma brevíssima introdução à programação com a linguagem Python



codificADAS
USP





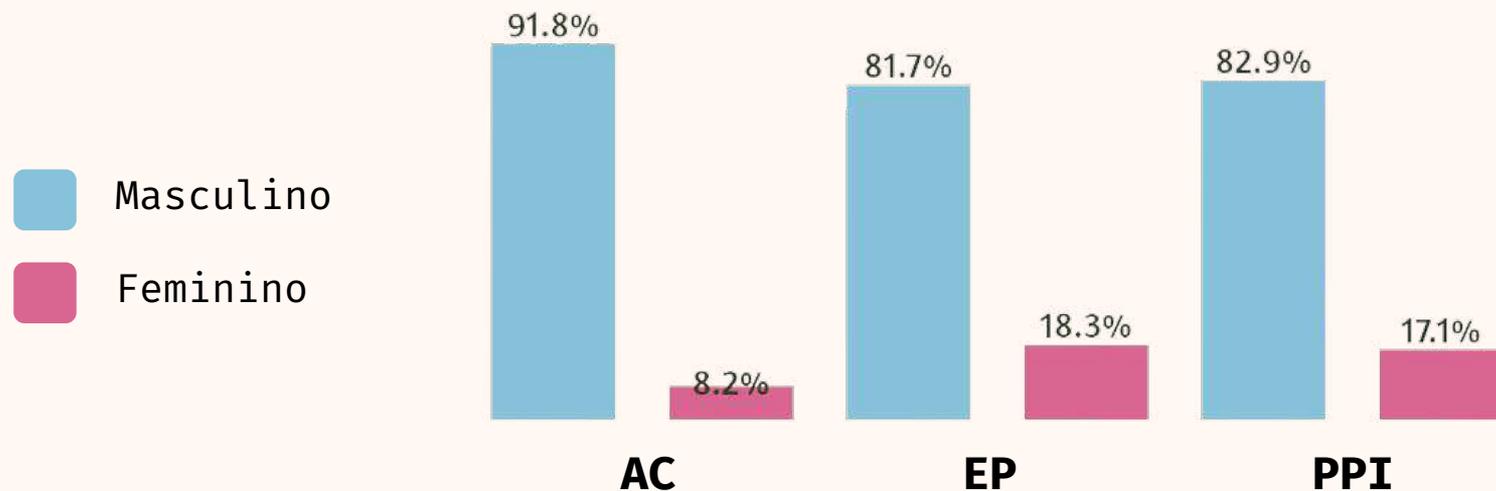
Quem somos nós?

O **CodificADAs** é um grupo que propõe o desenvolvimento de atividades de ensino e divulgação da computação para alunas dos anos finais do ensino fundamental e médio, visando promover a inclusão feminina nos cursos superiores de computação e contribuir no combate da desigualdade de gênero da área.



FUVEST 2021

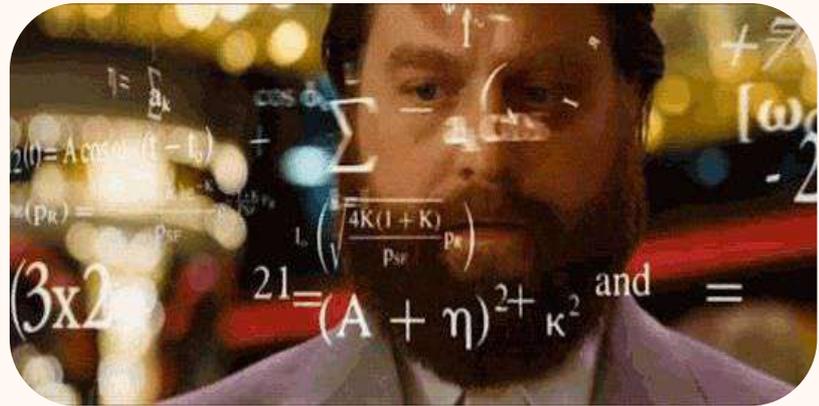
- Questionário de Avaliação Socioeconômica: Candidatos que foram chamados para a 1ª matrícula na carreira de computação.



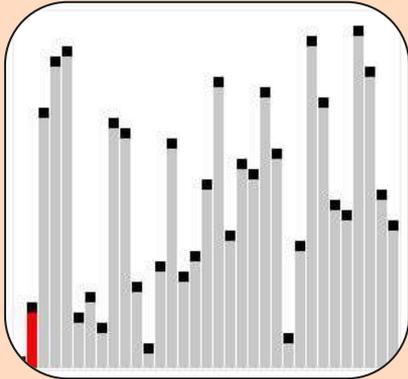
Fonte: https://acervo.fuvest.br/fuvest/2021/FUVEST_2021_qase_chamados_matricula_1_carreira.pdf



Algoritmos



Exemplos de algoritmos



Triângulo


$$A = \frac{b \times h}{2}$$

b = base
 h = altura

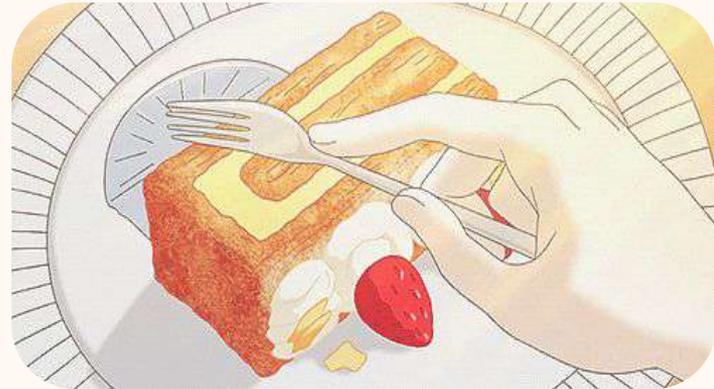


Lógica de programação e algoritmo

Lógica de programação é uma **técnica utilizada para escrever uma sequência de passos de modo a resolver um problema.**

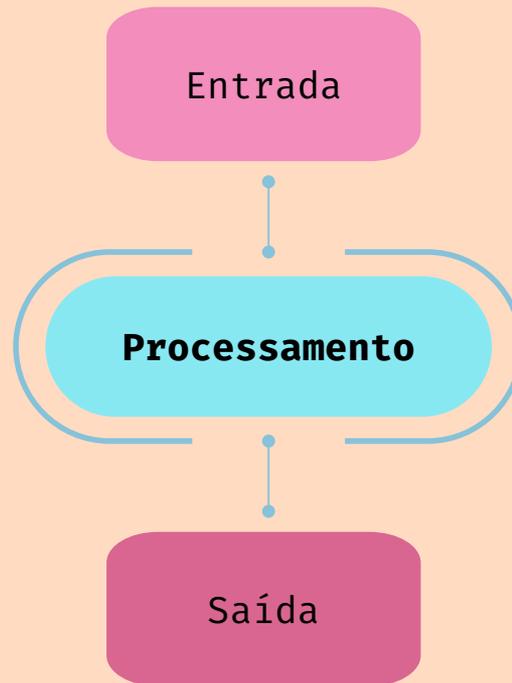
Ela é como uma receita de bolo.

Chamamos essa receita de bolo de **algoritmo.**





Passos para criar um algoritmo



O algoritmo na receita de bolo

Ingredientes (Entrada)

- 4 ovos
- 2 xícaras de chá de açúcar
- 2 xícaras de chá de farinha de trigo
- 1 xícara de chá de fubá
- 3 colheres de sopa de margarina
- 1 xícara de chá de leite





O algoritmo na receita de bolo

Processamento

1. Bater as claras em neve, acrescentar o açúcar, continuar batendo
2. Acrescente aos poucos as gemas, a margarina, o leite, a farinha de trigo, o fubá e continue batendo
3. Coloque por último o fermento misturando com uma colher ou espátula
4. Coloque a massa em uma forma untada e deixe assar em forno médio pré-aquecido por aproximadamente 30 minutos





Saída esperada

Um delicioso bolo de fubá!





Por fim... O que é um algoritmo?

O algoritmo pode ser descrito como uma **sequência finita e ordenada de instruções** claras para **atingir algum objetivo**, ou seja, **solucionar um problema**.





codificADAs
USP



Linguagens de programação



Linguagens de programação

Lembra quando falamos de lógica de programação, que é a técnica que a gente usa para escrever um algoritmo? Com que é escrito?

Com as linguagens de programação!

As linguagens de programação funcionam como idiomas, existem várias no mundo, e na programação também.



C++



Ruby



JavaScript



Java



Python

Google Colab

O **Google Colaboratory** é a plataforma que iremos utilizar para escrever e executar nossos códigos.

- É online e gratuito.
- Além dos códigos podemos adicionar texto, formando uma espécie de caderno.





Entrada e saída de dados





Saída de dados e o print()

Utilizamos a **função print()** quando quisermos **mostrar uma mensagem na tela**.

Podemos colocar qualquer frase dentro do parênteses que o computador vai retornar a sua mensagem!



```
print('Olá mundo!')
```



Entrada de dados e o input()

Utilizaremos a **função input()** quando quisermos **transmitir uma mensagem** (um dado) para o computador!

Lembre-se do **input como uma forma de inserir dados**, você rodando o código acima, percebe-se que vai ser colocado uma caixinha de texto para o usuário digitar!

Ou seja, o computador te ofereceu um meio para você inserir dados.



```
input('Escreva o seu nome: ')
```



Variáveis



O que são variáveis?

Variáveis são **espaços situados na memória do computador** onde podemos **guardar valores ou expressões**.

Ou seja, se quisermos que o computador memorize um número para que possamos usá-lo posteriormente em alguma operação, guardamos esse número na memória utilizando uma **variável**.



As variáveis na vida real

Vamos supor que temos um armário, no qual podemos guardar diversos objetos. E o nosso objetivo é guardar uma bola nesse armário.

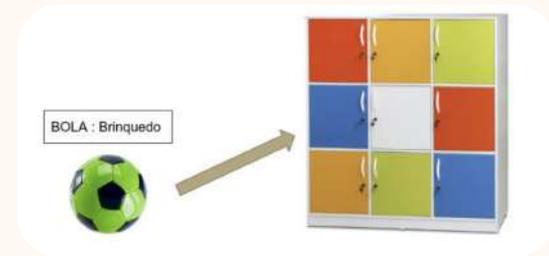
Só que somos organizadas em relação ao que guardamos no nosso armário. Assim, antes de colocarmos um objeto no nosso armário, colocamos uma etiqueta na porta dizendo o **identificador (o nome) desse objeto** que está guardado naquela gaveta, e de que tipo ele é. No caso da bola, colocaremos o identificador BOLA e dizemos que a bola é do tipo Brinquedo.



Armários, memória, brinquedos e variáveis

Portanto, percebemos que:

- O **armário** seria como a **memória do nosso computador**, com várias divisões para guardarmos coisas;
- A **etiqueta** seria o **nome da variável**;
- A **bola** em si seria como a **informação** que queremos guardar.





Definição

Formalmente variáveis são **referências à objetos** capazes de **guardar e representar valores**, ou seja, variáveis são utilizadas para **armazenar valores em memória** (a do computador).

Nela utilizamos um **identificador** para associar esse valor a um nome usando o comando de atribuição que corresponde ao símbolo **=** como no trecho ao lado:



```
nome_variavel = valor
```





Existe algum padrão para declarar uma variável?

Não existem regras para declaração de variáveis, mas sim boas práticas que podemos lembrar antes de definirmos uma:

- Podem possuir caracteres alfanuméricos e o caractere underscore `_`
- É recomendado que comece com letra minúscula
- Não pode ter espaço entre elas
- Não podem começar com números
- Não podem ter caracteres de acentuação tais como `^`, `~`, `!`
- Não podem ter caracteres especiais como `@`, `#`, `&` nem hífen `-`
- Não use nomes que comecem e terminem com dois underscores `__` o Python define vários métodos especiais e variáveis que usam esse padrão.



Tem exemplo aí?

Aqui há alguns exemplos de nomes de variáveis válidos e inválidos, de acordo com as práticas citadas:

Variáveis

válidas

tempo

area_triangulo

samuel_john

nome2

inválidas

2tempo

area triangulo

samuel_&_john

total-val

Não pode começar com números

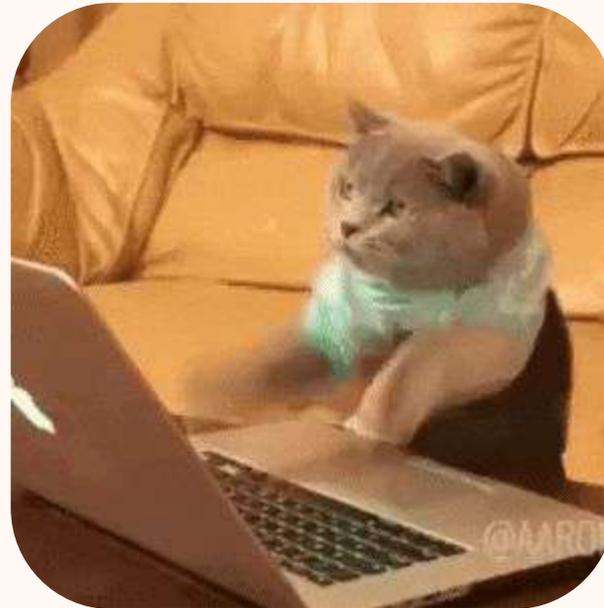
Não é permitido espaço em branco

Não são válidos caracteres especiais

Não são válidos hífens



Tipos de dados





Quais os possíveis tipos em Python?

Tipo	Descrição	Exemplo
int	Números inteiros	15
float	Ponto flutuante (números fracionários)	9.67
booleano	Verdadeiro/ Falso	True
str	Caracteres (frases, nomes, letra, etc.)	“Ana”
list	Conjunto de elementos	[“Maçã”, “Banana”, “Uva”]
tupla	Semelhante ao list, mas com elementos que não se modificam. Não é possível adicionar ou remover elementos, por exemplo.	(“Maçã”, “Banana”, “Uva”)



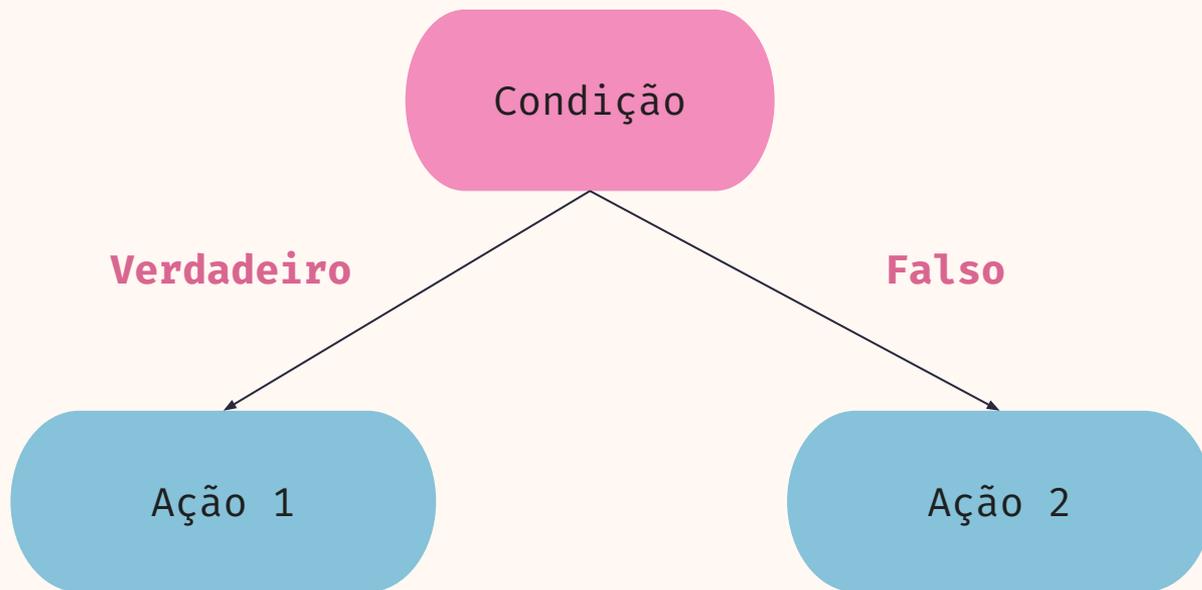


Condicionais

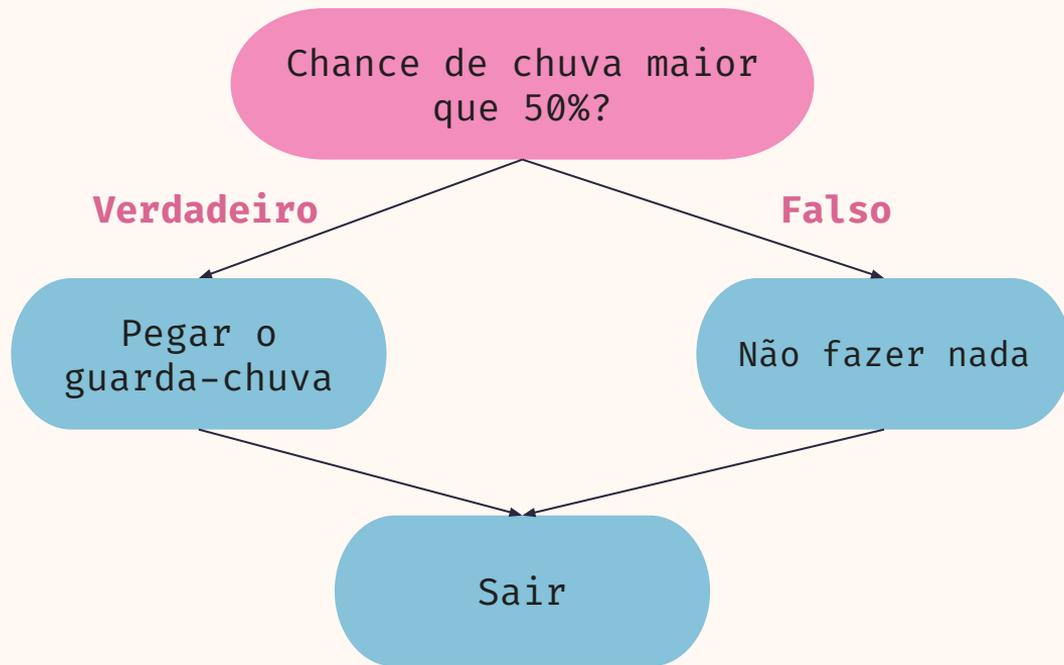




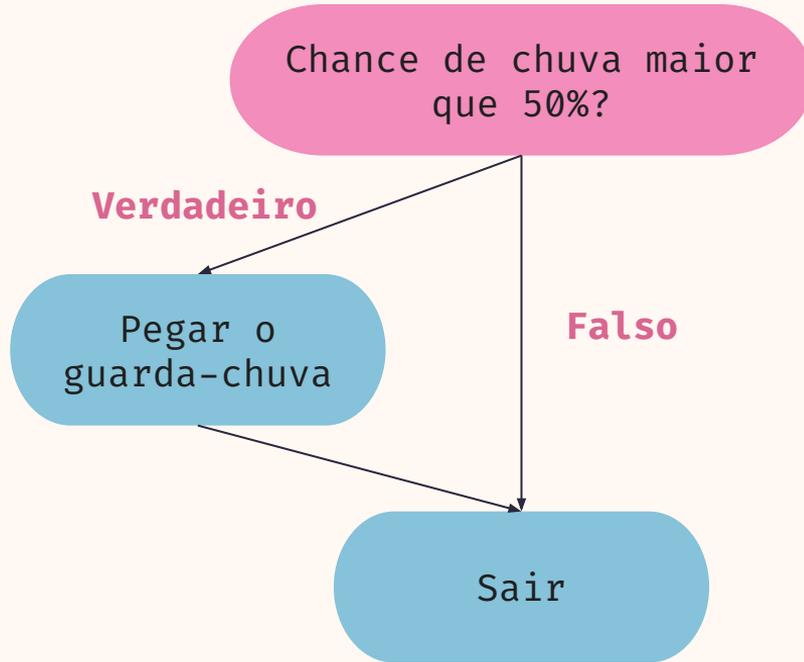
Se...



Exemplo: Você vai sair de casa.



Exemplo: Você vai sair de casa.



Operação If

- O operador **if** (do inglês “se”) serve para executar uma **parte do código** somente **se uma condição for verdadeira**.



```
if condição:
```

```
    código a ser executado caso a condição seja verdadeira
```

```
restante do programa
```

Identação

- O código a ser executado deve ser indentado
- Indentação é quando você dá uma **série de espaços para separar um bloco de código de outro.**
- Para Indentar no colab basta apertar **TAB**





Operadores relacionais

Operador	Descrição	Exemplo	Resultado
==	Igualdade	2 == 3	False
!=	Desigualdade	2 != 3	True
>	Maior	3 > 3	False
>=	Maior ou igual	3 >= 3	True
<	Menor	2 < 3	True
<=	Menor ou igual	4 <= 3	False



Operação Else

- O operador **else** (do inglês “senão”) serve para executar uma parte do código somente **se uma condição posta no if não for verdade**



```
if condição:  
    código a ser executado caso a condição seja verdadeira  
else:  
    código a ser executado caso a condição seja falsa
```

Operação Elif

- O operador **elif** é uma mistura dos dois operadores, ele só roda **se a condição de cima não foi cumprida e a condição atual foi**.





Laços





O que é um laço?

Definição

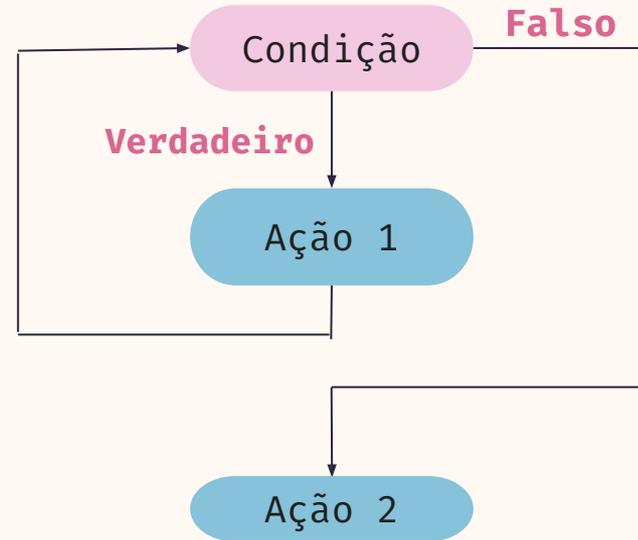
Um laço ou um loop é um **tipo de estrutura** na programação que **simplifica** um código caso seja necessário a repetição de um comando

Tipos

- While
- For



Estrutura básica de um laço





0 comando while



```
controle  
while <condição>:  
    comandos  
    ...  
  
controle
```



O comando for



```
for i in range(<início>, <fim>, <passo>):  
    comandos
```

- Atenção! O intervalo do range é: **[início, fim[** ou seja, inclui o valor de início mas não o de fim.

Em outras palavras, no comando **for i in range(0, 5, 1)** a variável **i** percorrerá os números **0, 1, 2, 3 e 4**



... algumas coisinhas dignas de atenção

- MUITA atenção com o desenho do seu loop:
 - Você não quer que ele acabe antes do que precisa
 - Você também não quer que ele acabe depois do que você precisa
 - E, pior ainda, você definitivamente não quer que ele não acabe nunca!





... algumas coisinhas dignas de atenção

- Chamamos de **contador**, mas é bem comum ser chamado de **iterador** também.
- **Iteração** é o nome dado para cada vez que o computador acaba de executar o **bloco de repetição**.
- Perceba que a **condição é verificada** a primeira vez antes de começar o loop e toda vez que uma iteração acaba





Gratuito

 Universidade de São Paulo

**Introdução à Ciência da Computação com Python
Parte 1**

Habilidades que você terá: Computer Programming, Python Programming,...

★ 4.9 (6,7k avaliações)

Beginner · Course · 1-3 Months

Muito obrigada!

Essa foi nossa brevíssima introdução à programação em Python, esperamos que tenha gostado!

Para quem desejar prosseguir nos estudos, a USP oferece um curso 100% gratuito e online na plataforma Coursera.

[Acesse-o clicando aqui.](#)





Nossas redes sociais!



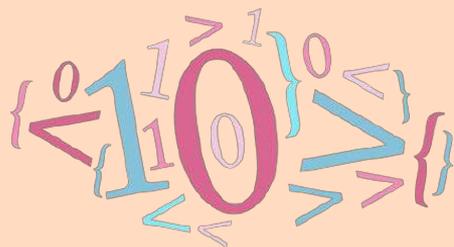
@codificadasusp



facebook.com/codificadasusp



codificadas.ime.usp.br



codificADAS
USP

