

# PROGRAMAÇÃO DE INTERNET

Prof. Richard Brosler

E-mail: [richard.brosler@sp.senai.br](mailto:richard.brosler@sp.senai.br)

Grupo WhatsApp: <https://bit.ly/3K8jFIg>





# AGENDA

- Exercícios
- Procedimentos e funções
- Mais Exercícios 😊



# EXERCÍCIOS

- 1) montar um algoritmo que monte essa tela abaixo. Dica: Você deverá usar 2 laços de repetição. Um para a linha e dentro um para coluna. (usar as 3 estruturas de repetição)

1-1

2-2

3-3

4-4

5-5

6-6

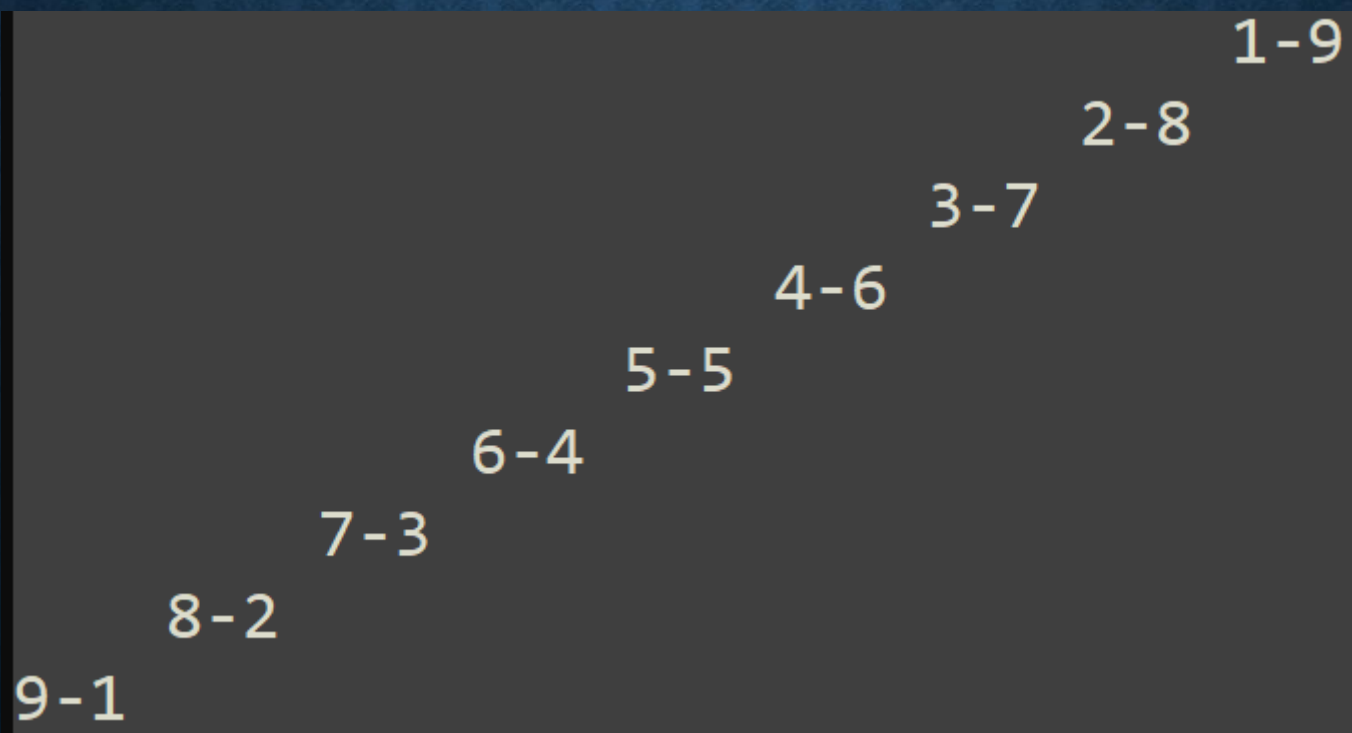
7-7

8-8

9-9

# EXERCÍCIOS

- 2) montar um algoritmo que monte essa tela abaixo. Dica: Você deverá usar 2 laços de repetição. Um para a linha e dentro um para coluna. (usar as 3 estruturas de repetição)



# EXERCÍCIOS

- 3) montar um algoritmo que monte essa tela abaixo. Dica: Você deverá usar 2 laços de repetição. Um para a linha e dentro um para coluna. (usar as 3 estruturas de repetição)

1-1 1-9

2-2 2-8

3-3 3-7

4-4 4-6

5-5

6-4 6-6

7-3 7-7

8-2 8-8

9-1 9-9



# EXERCÍCIOS

- 4) montar um algoritmo para solicitar qual termo da frequência de Fibonacci ele deseja (sequencia deve ser maior que 4). Você deverá imprimir a sequencia até o termo desejado pelo usuário.
- Frequência de Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89....
- Se o usuário escolher por exemplo o 7º termo, teremos o resultado:
- 0, 1, 1, 2, 3, 5, 8

# EXERCÍCIOS

- 5) montar um algoritmo para solicitar qual um número inteiro. Você calculará o fatorial desse número e apresentará o resultado para o usuário.
- Exemplo de fatorial:  $5! \Rightarrow 5 * 4 * 3 * 2 * 1 \Rightarrow 120$
- 6) montar um algoritmo para solicitar o cpf (tipo caractere, somente números). Você deverá verificar se ele é um cpf válido, se for, mostrar a mensagem “CPF Válido”, senão, “CPF inválido”. Para saber se um cpf é válido ele não pode ser 111.111.111-11, ou 222.222.222-22, assim sucessivamente até o 999.999.999-99. Deve ter 11 posições numéricas, e os 2 dígitos finais deve ser calculados conforme a regra.



# EXERCÍCIOS

- Regras:
  - 1º dígito. Da esquerda para direita iremos pegar cada número individualmente, multiplicaremos por 11-posição do item, ou seja, vamos ao exemplo para o CPF 123.456.789-00.

1º	2º	3º	4º	5º	6º	7º	8º	9º	DV1
1	2	3	4	5	6	7	8	9	0
10	9	8	7	6	5	4	3	2	Soma
10	18	24	28	30	30	28	24	18	210



# EXERCÍCIOS

- Regras:
  - 2º dígito. Da esquerda para direita iremos pegar cada número individualmente, multiplicaremos por 12-posição do item até a penúltima posição , ou seja, vamos ao exemplo para o CPF 123.456.789-00.

1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	DV2
1	2	3	4	5	6	7	8	9	0	9
11	10	9	8	7	6	5	4	3	2	<b>Soma</b>
11	20	27	32	35	36	35	24	27	0	<b>255</b>

- Note que os dígitos calculados foram 0 para o primeiro dígito e 9 para o segundo dígito, ou seja, o cpf é inválido.

# EXERCÍCIOS

- Apoios para resolução da atividade:
  - Para essa atividade vamos precisar de algumas funções do visualg, são elas:
  - Compr(campostring) => retorna o número de caracteres que a string possui
  - copia(texto,posinicial,quantidadeCaracteres) => pega parte de uma string.
  - Caracpnum(texto) => converte o texto para número



# PROCEDIMENTOS

- Blocos de códigos que não devolvem resposta após sua execução
- Vejam o exemplo ao lado, onde usaremos um procedimento para montagem da tela

```
1 Algoritmo "semnome"
2
3 procedimento montaTela()
4 inicio
5     limpatela()
6     escreval("Menu de opções")
7     escreval("1 - cadastrar")
8     escreval("2 - consultar")
9     escreval("3 - alterar")
10    escreval("4 - excluir")
11    escreval("9 - fim")
12    escreval()
13    escreva("Digite sua opção:")
14 fimprocedimento
15
16 Var
17     opc : inteiro
18 Inicio
19     repita
20         montaTela()
21         leia(opc)
22     ate opc = 9;
23 Fimalgoritmo
```

# FUNÇÕES

- Funções, assim como procedimentos, são blocos de código que serão executados, porém, diferentemente de um procedimento, elas retornam valor.

```
1 Algoritmo "semnome"  
2  
3 funcao somar() : inteiro  
4 var ret : inteiro  
5 inicio  
6   ret := a + b  
7   retorne ret  
8 fimfuncao  
9  
10 Var  
11   a, b : inteiro  
12 Inicio  
13   a := 10  
14   b := 20  
15   escreval("a soma é ", somar())  
16 Fimalgoritmo
```