

# Operadores Lógicos e Relacionais e Comandos de Seleção

## Laboratório 02 - Prática da Aula 02

ECT3201 - Linguagem de Programação (C++)

Prof. Éverton Santi

# Revisão

Escreva um programa que leia:

1. `idade`
2. `possui_documento` ( `1` para sim, `0` para não)
3. `autorizado_responsavel` ( `1` para sim, `0` para não)

Classifique:

1. `"Entrada permitida"` para pessoas maiores de idade
2. `"Entrada permitida com responsavel"` para adolescentes com documento e autorização do responsável
3. `"Entrada negada"` nos demais casos

# Exercício 1 - Par ou Ímpar

Crie um programa que:

1. Leia um número inteiro
2. Mostre "Par" quando o número for par

## Exercício 2 - Senha Numérica

Crie um programa que:

1. Leia uma senha inteira digitada
2. Considere a senha correta igual a 1234
3. Mostre:
  - "Acesso liberado" se a senha estiver correta
  - "Acesso negado" caso contrário

## Exercício 3 - Triagem de Pedido

Crie um programa que leia:

1. `valor_pedido`
2. `cliente_premium` ( `1` ou `0` )

Classifique:

1. `"Frete gratis"` para pedidos de maior valor ou para clientes premium
2. `"Frete normal"` nos demais casos

Use uma constante simbólica para definir o valor mínimo de um produto apto a receber desconto:

```
#define VALOR_MINIMO 500.0f
```

## Exercício 4 - Ano Bissexto

Crie um programa que:

1. Leia um ano inteiro
2. Classifique o ano considerando as regras do calendário gregoriano:
  - anos divisíveis por 400 são bissextos
  - anos divisíveis por 100, mas não por 400, não são bissextos
  - anos divisíveis por 4, mas não por 100, são bissextos
  - nos demais casos, o ano não é bissexto
3. Mostre "Bissexto" ou "Nao bissexto"

# Exercício 5 - Classificação de IMC

Crie um programa que:

1. Leia `peso` e `altura`
2. Calcule `imc = peso / (altura * altura)`
3. Classifique:
  - `"Baixo peso"` para IMC abaixo de 18,5
  - `"Peso normal"` para IMC de 18,5 até 24,9
  - `"Acima do peso"` para IMC a partir de 25,0

# Exercício 6 - Aprovação em Processo

Crie um programa que leia:

1. `nota_prova`
2. `nota_entrevista`
3. `documentacao_ok` ( `1` ou `0` )

Classifique:

1. `"Aprovado"` quando prova e entrevista atingirem nota mínima e a documentação estiver regular
2. `"Lista de espera"` se: a documentação estiver regular e pelo menos uma das notas estiver em faixa intermediária
3. `"Reprovado"` nos demais casos

Use constantes simbólicas para definir:

1. Nota mínima para aprovação;
2. Nota intermediária para lista de espera;

## Exercício 7 - Categoria de Entrega

Crie um programa que leia:

1. `distancia_km`
2. `peso_kg`
3. `urgente` ( `1` ou `0` )

Classifique:

1. `"Entrega expressa"` para pedidos urgentes, próximos e leves
2. `"Entrega padrao"` para pedidos com distância e peso dentro da faixa regular
3. `"Entrega especial"` nos demais casos

# Desafio Extra - Risco de Fraude

Crie um programa que leia:

1. `valor_compra`
2. `tentativas_24h`
3. `pais_igual` ( `1` ou `0` ) (se o país da compra é igual ao país usual do cliente)
4. `cartao_verificado` ( `1` ou `0` )

Classifique:

1. `"Alto risco"` para ausência de verificação do cartão, compra muito alta fora do país usual ou muitas tentativas recentes
2. `"Risco medio"` para compra alta fora do país usual ou quantidade intermediária de tentativas
3. `"Baixo risco"` nos demais casos