

The background features a collection of 3D blocks in various colors (purple, blue, yellow, green, orange, pink, grey) arranged in a scattered, overlapping manner. Some blocks are solid, while others have a square cutout in the center. Thin, curved lines in blue, red, and green weave through the blocks, adding a sense of movement and connectivity. The overall aesthetic is clean, modern, and geometric.

Mãos à obra! - Respostas

Processamento predefinido

Exercício 1

Neste exercício, devemos elaborar um processamento predefinido que faça a autenticação do código de cargo para, depois disso, passar a informação ao algoritmo se o código é válido ou não. Veja a proposta:

Em um programa que verifica se um CÓDIGO DE CARGO é existente numa tabela com códigos que vão de 1 a 1000, crie o processamento predefinido CHECACODIGO() para validar o código digitado. Se o código for válido, altere o valor da variável CHECK para que, no programa adiante, seja feita a busca da descrição do cargo, e para que esta seja exibida. Senão, exiba mensagem de erro e finalize.

```
VERIFICA_CARGO
  DECLARA  CODCARGO,  VALIDAÇÃO  NUMÉRICAS
  VALIDAÇÃO = 0
  EXIBIR "DIGITE O CÓDIGO DE UM CARGO"
  LER CODCARGO
  CHEGACODIGO( CODCARGO, VALIDAÇÃO)
  SE VALIDAÇÃO < > 0
    BUSCAR A DESCRIÇÃO DO CARGO NA TABELA DE CARGOS
    EXIBIR A DESCRIÇÃO DO CARGO
  FIM SE
FIM
```

A resposta é esta:

```
CHECACODIGO (CODCARGO, VALIDAÇÃO)
  SE COD >= 1  E COD <= 100
    VALIDAÇÃO = 1
    SENÃO EXIBIR "CÓDIGO NÃO EXISTENTE"
  FIM SE
SAÍDA
```

Antes de começar a elaborar o processamento predefinido, vamos dar uma olhada no algoritmo em que ele será utilizado. No algoritmo VERIFICA_CARGO, foram declaradas duas variáveis, uma para receber o código do cargo do funcionário e outra para checar se esse código é válido.

```
DECLARA    CODCARGO,    VALIDAÇÃO    NUMÉRICAS
```

A variável **VALIDAÇÃO** poderá ter apenas dois valores possíveis, **zero (0)** ou **1**. O zero representa um código inválido e o 1 será retornado se o código for válido.

Na sequência, foi atribuído o valor zero na variável **VALIDAÇÃO**. Isso é feito para definir um estado padrão para a variável. Dessa forma, quando formos construir o nosso processamento predefinido, o valor só precisa ser alterado se o código for válido.

```
VALIDAÇÃO = 0
```

Com as variáveis declaradas, é pedido ao usuário que digite o código do cargo e, logo após isso, temos o comando de leitura do teclado. Assim, o número que o usuário digitar será atribuído à variável **CODCARGO**.

```
EXIBIR "DIGITE O CÓDIGO DE UM CARGO"  
LER CODCARGO
```

Agora que o código do cargo foi definido, foi feita uma validação através de um processamento predefinido.

```
CHECACODIGO (CODCARGO, VALIDAÇÃO)
```

É aí que entra a nossa solução. Mas, antes disso, vamos continuar analisando o algoritmo para ver como ele irá tratar o retorno recebido no processamento.

Mãos à obra!

Introdução à Lógica de Programação

4 / 4

Veja que temos uma decisão que irá verificar se a validação que acabou de ser processada continua com o valor zero ou foi alterada para 1. Se o valor não for zero, ou seja, a condição seja verdadeira, será pesquisada a descrição do cargo que o usuário digitou e exibida na tela, logo em seguida.

```
SE VALIDAÇÃO < > 0
    BUSCAR A DESCRIÇÃO DO CARGO NA TABELA DE CARGOS
    EXIBIR A DESCRIÇÃO DO CARGO
FIM SE
```

Vamos voltar para o processamento predefinido. Agora que já sabemos as variáveis que ele irá utilizar e como o valor de validação será tratado pelo algoritmo, fica bem mais fácil saber o que precisamos fazer para validar o código do cargo. Uma das alternativas é inserir uma estrutura de decisão que irá checar se o código recebido é um número entre 1 e 1000.

```
CHECACODIGO (CODCARGO, VALIDAÇÃO)
    SE COD >= 1 E COD <= 1000
```

Se isso for verdade, então definimos que a variável de validação possui valor 1, ou seja, esse código é válido.

```
VALIDAÇÃO = 1
```

Caso contrário, exibimos uma mensagem indicando que o código é inexistente.

```
SENÃO EXIBIR "CÓDIGO NÃO EXISTENTE"
```

Depois disso, basta fechar a nossa estrutura de decisão e inserir o comando SAÍDA, que indica o fim do processamento predefinido.

```
FIM SE
SAÍDA
```