**UNIVERSIDADE DE FRANCA – UNIFRAN**

FABIO TOLEDO BONEMER DE SALVI

**Monitoramento do Preço da Cesta Básica Usando Pseudocódigo**

Experiência Prática I

ARARAQUARA – SÃO PAULO

2025

FABIO TOLEDO BONEMER DE SALVI

**Monitoramento do Preço da Cesta Básica Usando Pseudocódigo**

Experiência Prática I

Trabalho apresentado ao curso de Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade de Franca – UNIFRAN, como requisito para obtenção de nota de Experiência Prática I na disciplina Algoritmos e Pensamento Computacional.

ARARAQUARA – SÃO PAULO

2025

# Introdução

O presente trabalho tem como objetivo desenvolver a habilidade de construção de algoritmos utilizando fluxograma e pseudocódigo. Para isso foi proposto a implementação de um problema do cotidiano real da vida do brasileiro envolvendo a análise da variação de preços.

Este problema foi implementado em forma de fluxograma no software *Flowgorithm*.

O diretório github deste projeto está disponível em Salvi (2025).

# Descrição do Problema

Desenvolver uma ferramenta que realize a análise da variação de preço dos produtos de uma cesta básica para ser utilizada por órgãos de defesa do consumidor.

O objetivo é verificar se os preços dos produtos estão aumentando de forma abusiva e gerar relatórios que ajudem na tomada de decisão.

## Requisitos que devem ser atendidos

Para dada produto:

### Ler: nome, preço anterior e preço atual;

### Calcular: Variação percentual;

### Exibir:

#### Se o preço aumentou, diminuiu ou permaneceu estável;

#### Se o aumento foi **superior a 10%**, sinalizar como **abuso**.

## Definição das variáveis

Todas as variáveis foram utilizadas em forma de **vetor(array)** para otimizar o algoritmo e permitir que o usuário defina o número de produtos que será passado para análise.

Segue as varáveis utilizadas neste algoritmo:

### numeroProdutos: tipo real (número decimal).

Variável utilizada para definir o número de produtos que serão analisados.

### nome: tipo caractere (texto/string).

Array utilizado para armazenar o nome dos produtos.

### precoAnterior: tipo real (número decimal).

Array utilizado para armazenar o preço dos produtos antes da atualização.

### precoAtual: tipo real (número decimal).

Array utilizado para armazenar o preço dos produtos após a atualização.

### variacao: tipo real (número decimal).

Array utilizado para armazenar a variação percentual do preço dos produtos.

### situacao: tipo caractere(texto/string).

Array utilizado para armazenar a classificação da variação, podendo ser “AUMENTO”, “AUMENTO ABUSIVO”, “QUEDA” ou “ESTÁVEL”.

## Definição das funções

* + - estaEmBranco**:** verifica se uma variável do tipo **caractere(texto/string)** possui apenas espaços em branco.

A função recebe como argumento uma variável **texto** do tipo **caractere(texto/string)** onde é passado o nome do produto e retorna uma variável do tipo **booleana** com o valor **true** para o caso do usuário não ter digitado o nome do produto e retorna **false** para o caso de o usuário ter digitado o nome do produto.

* + - variacaoPercentual:calcula a variação percentual de um produto utilizando seus valores de preço do mês anterior e do preço do mês atual.

A função recebe duas variáveis do tipo **real (número decimal)** representando os valores de **precoAnterior** e **precoAtual** e calcula a variação percentual do preço conforme a seguinte expressão:

|  |  |
| --- | --- |
|  | **(Eq. 1)** |

É retornado um valor **real (número decimal)** com o valor armazenado na variável **variacao**.

* + - situacaoProduto**:** determina o tipo de variação de preço utilizando o resultado o calcula da variação percentual do valor do produto.

As opções de variação de preço são:

* + - * QUEDA: para uma variação de preço negativa.
      * ESTÁVEL: para uma variação de preço nula.
      * AUMENTO: para uma variação de preço positiva e até 10%.
      * AUMENTO ABUSIVO: para uma variação de preço positiva acima de 10%.

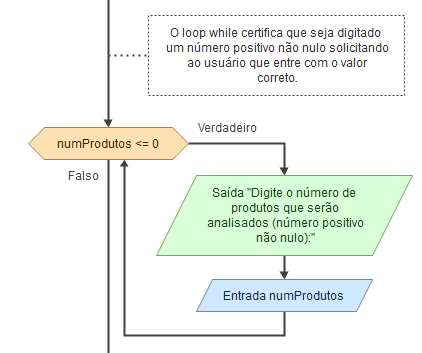
## Fluxograma

### Diagrama O conteúdo gerado por IA pode estar incorreto.Função Principal

Figura - Função Principal. Parte 1 de 2.

Diagrama

O conteúdo gerado por IA pode estar incorreto.



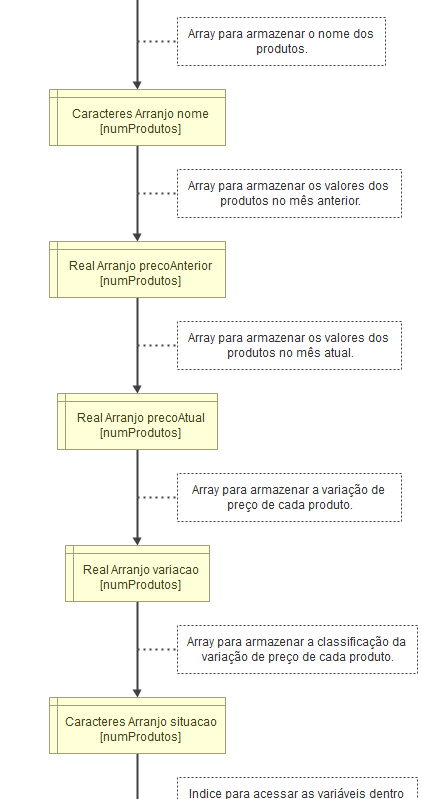
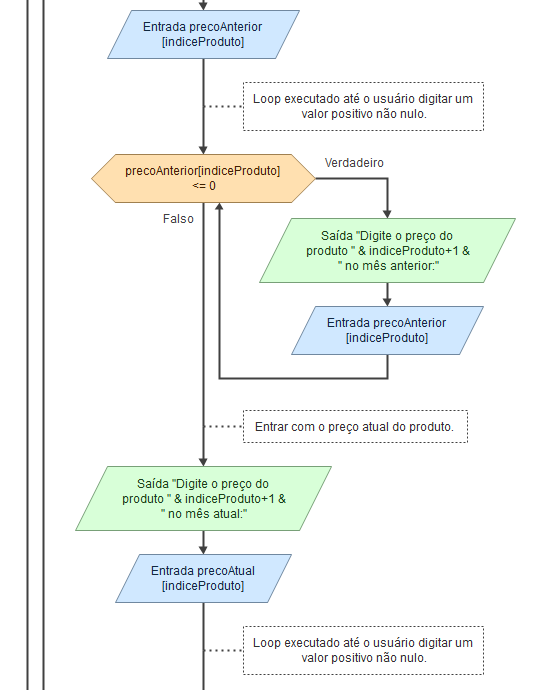
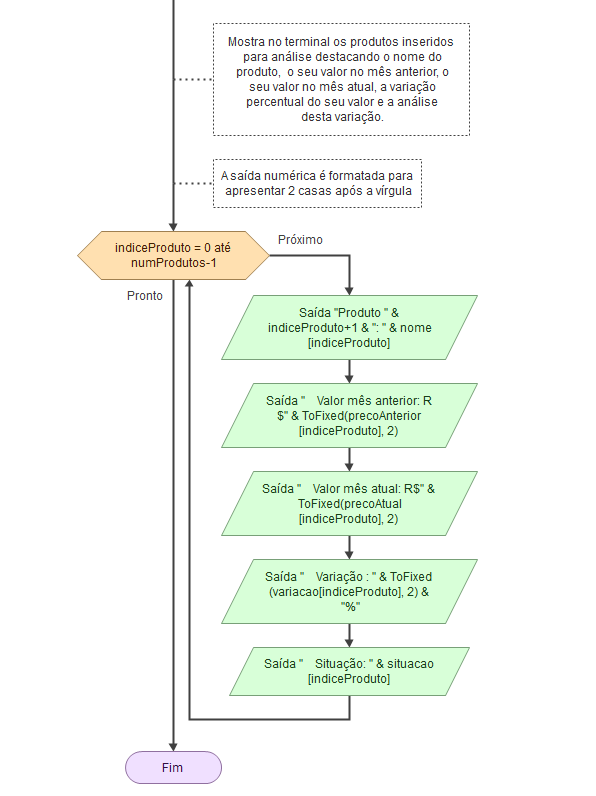
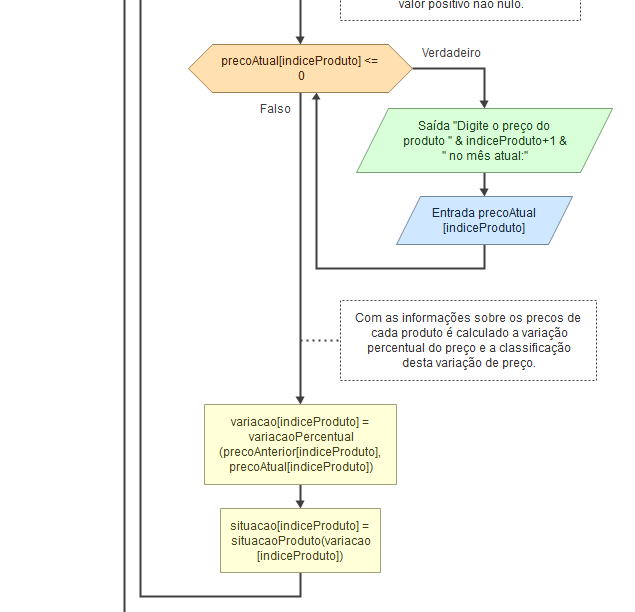


Figura - Função Principal. Parte 2 de 2.







### Função estaEmBranco

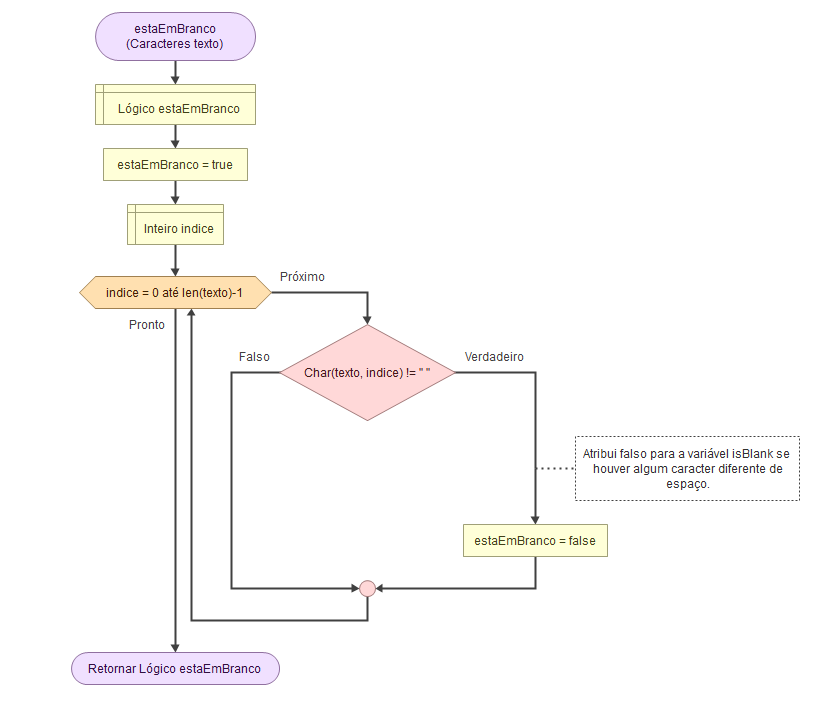


Figura - Função estaEmBranco.

### Função situacaProduto

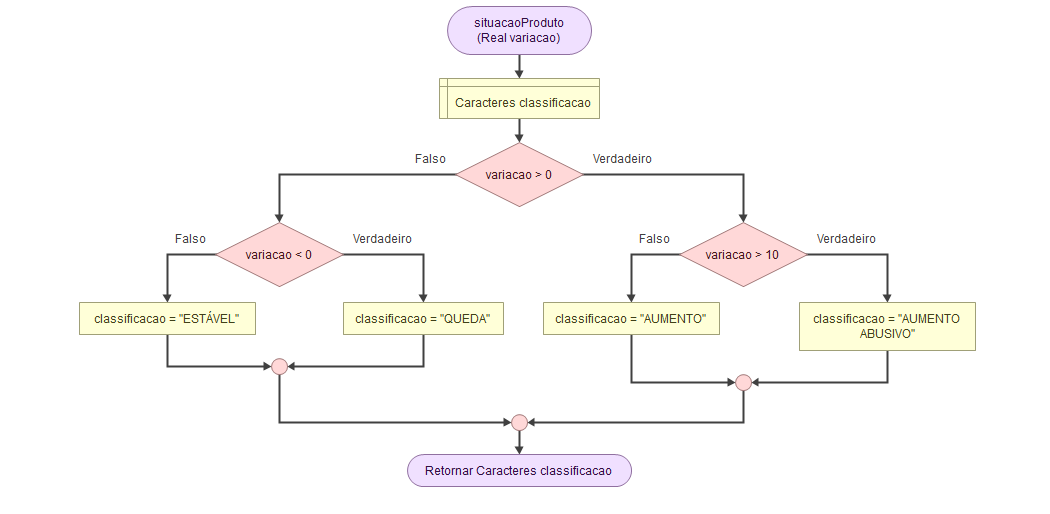


Figura - Função situacaoProduto

### Função variacaoPercentual

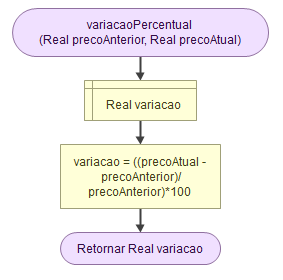


Figura - Função variacaoPercentual

# Pseudocódigo

O pseudocódigo foi gerado utilizando o software *Flowgorithm*(COOK, Devin).

Primeiro foi necessário montar um fluxograma que descrema o problema.

Após esta etapa foi possível obter o pseudocódigo do algoritmo desenvolvido.

Abaixo é apresentado uma descrição dos componentes que formam o pseudocódigo, exemplificando a forma de utilização.

Após esta descrição é apresentado o pseudocódigo com as funções que formam o fluxograma que soluciona o problema proposto.

### Descrição dos componentes e estruturas que formam o pseudocódigo:

* + - * **Comentário:** um comentário é inserido após o pseudocódigo **... (três pontos seguidos)**.

**...** Comentário.

* + - * **Definindo uma variável:** uma varável é definida utilizando o pseudocódigo **Definir** seguido do tipo de variável e nome da variável.

**Definir TipoDado** nomeVariaval

* + - * **Definindo uma variável vetor:** uma variável vetor é definida utilizando o pseudocódigo **Definir** seguido do tipo de variável, do pseudocódigo **Arranjo** e nome da variável.

**Definir TipoDado** **Arranjo** nomeVariaval

* + - * **Função saída:** uma função saída é definida utilizando o pseudocódigo **Saída** seguido da expressão que será analisada e mostrada em um prompt virtual.

**Saída** expressão

* + - * **Função entrada:** uma função entrada é definida utilizando o pseudocódigo **Entrada** seguido da variável que armazenará o dado digitado no prompt virtual.

**Entrada** variavel

* + - * **Laço de repetição para (for):**  o laço de repetição para (for) é delimitado pelos pseudocódigos **Para, até**, **crescente** ou **decrescente**, **passo** e **Fim**.

Os pseudocódigos **crescente** e **passo** são omitidos no software *Flowgorithm* (COOK, Devin) quando o passo é unitário, mas serão incluídos neste pseudocódigo.

**Para** variavel = valor\_inicial **até** valor\_final **crescente**/**decrescente** **passo** valor\_passo

**...** Aqui é incluída a lógica do algoritmo.

**Fim**

* + - * **Laço de repetição enquanto (while):**  o laço de repetição enquanto (while) é delimitado pelos pseudocódigos **Enquanto** e **Fim**. A condição de execução do laço é inserida após o pseudocódigo **Enquanto**.

**Enquanto** condição

**...** Aqui é incluída a lógica do algoritmo.

**Fim**

* + - * **Função principal:** a função principal é denominada por “Principal”, sendo delimitada pelos pseudocódigos **Função** e **Fim**.

**Função** Principal

**...** Aqui é incluída a lógica do algoritmo.

**Fim**

* + - * **Função que não retorna valor:** uma função que não retorna valor é delimitada pelos pseudocódigos **Função** e **Fim**.

**Função** nomeFuncao

**...** Aqui é incluída a lógica do algoritmo.

**Fim**

* + - * **Função que retorna um valor:** uma função que retorna um valor é delimitada pelos pseudocódigos **Função** e **Fim**. Possui o pseudocódigo **Retornar** dentro do corpo da função para retornar uma variável declarada no corpo da função.

**Função** nomeFuncao

**Definir TipoDado** nomeVariaval

**...** Aqui é incluída a lógica do algoritmo que manipula as variáveis antes do valor ser retornado.

**Retornar TipoDado** nomeVariaval

**Fim**

* + - * **Função que recebe um argumento e retorna um valor:** uma função que recebe um valor e retorna um valor é delimitada pelos pseudocódigos **Função** e **Fim**.

A função recebe um argumento declarado entre parênteses após o nome da função.

No corpo da função deve ter o pseudocódigo **Retornar** para retornar uma variável declarada no corpo da função.

**Função** nomeFuncao(**TipoDato** nomeVariavel1)

**Definir TipoDado** nomeVariaval2

**...** Aqui é incluída a lógica que manipula as variáveis antes do valor ser retornado.

**Retornar TipoDado** nomeVariaval2

**Fim**

* + - **Pseudocódigo:**

**Função** Principal

**Saída** "Análise da Variação de Preços de Produtos 1.0"

**Saída** "Desenvolvido por Fabio Toledo Bonemer De Salvi"

**Saída** "Sobre: Ferramenta para análise da variação de preços dos produtos que compõem uma cesta básica."

**...** Número de produtos que serão analisados.

**Definir Inteiro** numProdutos

**Saída** "Digite o número de produtos que serão analisados (número positivo não nulo):"

**...** Solicita ao usuário que entre com um número de produtos que serão analisados.

**Entrada** numProdutos

**...** O loop while certifica que seja digitado um número positivo não nulo solicitando ao usuário que entre com o valor correto.

**Enquanto** numProdutos <= 0

**Saída** "Digite o número de produtos que serão analisados (número positivo não nulo):"

**Entrada** numProdutos

**Fim**

**...** Array para armazenar o nome dos produtos.

**Definir** **Caracteres** **Arranjo** nome[numProdutos]

**...** Array para armazenar os valores dos produtos no mês anterior.

**Definir Real Arranjo** precoAnterior[numProdutos]

**...** Array para armazenar os valores dos produtos no mês atual.

**Definir Real Arranjo** precoAtual[numProdutos]

**...** Array para armazenar a variação de preço de cada produto.

**Definir Real Arranjo** variacao[numProdutos]

**...** Array para armazenar a classificação da variação de preço de cada produto.

**Definir Caracteres Arranjo** situacao[numProdutos]

**...** Indice para acessar as variáveis dentro do laço for.

**Definir Inteiro** indiceProduto

**...** Laço para o usuário entrar com os valores: nome do produto, preço no mês anterior e preço no mês atual referente aos produtos.

**...** Com essas informações é calculado a variação percentual do produto e a classificação da variação percentual do preço do produto.

**Para** indiceProduto = 0 **até** numProdutos-1 **crescente** **passo** 1

**...** Entrar com o nome do produto.

**Saída** "Digite o nome do produto " & indiceProduto+1 & ":"

**Entrada** nome[indiceProduto]

**...** Loop executado até o usuário digitar o nome do produto. Este loop tenta simular a função trim().

**Enquanto** estaEmBranco(nome[indiceProduto])

**Saída** "Digite o nome do produto " & indiceProduto+1 & ":"

**Entrada** nome[indiceProduto]

**Fim**

**...** Entrar com o preço anterior do produto.

**Saída** "Digite o preço do produto " & indiceProduto+1 & " no mês anterior:"

**Entrada** precoAnterior[indiceProduto]

**...** Loop executado até o usuário digitar um valor positivo não nulo.

**Enquanto** precoAnterior[indiceProduto] <= 0

**Saída** "Digite o preço do produto " & indiceProduto+1 & " no mês anterior:"

**Entrada** precoAnterior[indiceProduto]

**Fim**

**...** Entrar com o preço atual do produto.

**Saída** "Digite o preço do produto " & indiceProduto+1 & " no mês atual:"

**Entrada** precoAtual[indiceProduto]

**...** Loop executado até o usuário digitar um valor positivo não nulo.

**Enquanto** precoAtual[indiceProduto] <= 0

**Saída** "Digite o preço do produto " & indiceProduto+1 & " no mês atual:"

**Entrada** precoAtual[indiceProduto]

**Fim**

... Com as informações sobre os preços de cada produto é calculado a variação percentual do preço e a classificação desta variação de preço.

**Atribuir** variacao[indiceProduto] = variacaoPercentual(precoAnterior[indiceProduto], precoAtual[indiceProduto])

**Atribuir** situacao[indiceProduto] = situacaoProduto(variacao[indiceProduto])

**Fim**

... Mostra no terminal os produtos inseridos para análise destacando o nome do produto, o seu valor no mês anterior, o seu valor no mês atual, a variação percentual do seu valor e a análise desta variação.

**...** A saída numérica é formatada para apresentar 2 casas após a vírgula.

**Para** indiceProduto = 0 **até** numProdutos-1 **crescente** **passo** 1

**Saída** "Produto " & indiceProduto+1 & ": " & nome[indiceProduto]

**Saída** " Valor mês anterior: R$" & ToFixed(precoAnterior[indiceProduto], 2)

**Saída** " Valor mês atual: R$" & ToFixed(precoAtual[indiceProduto], 2)

**Saída** " Variação : " & ToFixed(variacao[indiceProduto], 2) & "%"

**Saída** " Situação: " & situacao[indiceProduto]

**Fim**

**Fim**

**Função** estaEmBranco (Caracteres texto)

**Definir** Lógico estaEmBranco

**Atribuir** estaEmBranco = true

**Definir Inteiro** índice

**Para** indice = 0 **até** len(texto)-1 **crescente** **passo** 1

**Se** Char(texto, indice) != " "

... Atribui falso para a variável isBlank se houver algum caracter diferente de espaço na variável texto.

**Atribuir** estaEmBranco = false

**Fim**

**Fim**

**Retornar** Lógico estaEmBranco

**Fim**

**Função** situacaoProduto (Real variacao)

**Definir** Caracteres classificacao

**Se** variacao > 0

**Se** variacao > 10

Atribuir classificacao = "AUMENTO ABUSIVO"

**Senão**

Atribuir classificacao = "AUMENTO"

**Fim**

**Senão**

**Se** variacao < 0

**Atribuir** classificacao = "QUEDA"

**Senão**

**Atribuir** classificacao = "ESTÁVEL"

**Fim**

**Fim**

**Retornar** Caracteres classificacao

**Fim**

**Função** variacaoPercentual (Real precoAnterior, Real precoAtual)

**Definir** Real variacao

**Atribuir** variacao = ((precoAtual - precoAnterior) / precoAnterior) \* 100

**Retornar** Real variacao

**Fim**

# Teste de Execução

O fluxograma foi e simulado no software *Flowgorithm* (COOK, Devin) sendo possível a interação utilizando uma interface que simula um prompt de entrada e saída de dados.

Abaixo temos as capturas de telas da simulação do fluxograma que representa o pseudocódigo apresentado no tópico anterior.

## Dados utilizados para análise

Os dados utilizados para análise da variação de preço foram retirados do material de orientação desta atividade (UNIVERSIDADE DE FRANCA, 2025) e estão dispostos na Tabela abaixo.

Tabela - Relação de produtos e preços que serão utilizados para validar a lógica apresentada em forma de fluxograma e pseudocódigo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Produto** | **Preço no mês anterior (R$)** | **Preço no mês atual (R$)** | **Variação (%)** | **Situação** |
| Arroz | 20,00 | 22,50 | +12,5 | Aumento abusivo |
| Feijão | 8,00 | 8,00 | 0,0 | Estável |
| Óleo de soja | 6,50 | 6,00 | -7,7 | Redução de preço |
| Açúcar | 4,00 | 4,40 | +10,0 | Limite de aumento |

## Tela de apresentação

Ao iniciar a simulação do fluxograma as seguintes mensagens de apresentação são apresentadas no prompt de comandos simulado.

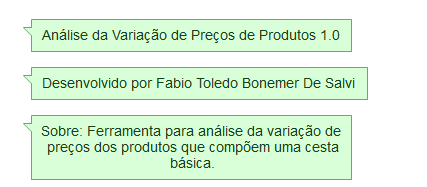


Figura - Mensagens de apresentação ao iniciar a simulação.

## Definindo o número de produtos

Define-se o número de produtos a serem analisados digitando um número positivo não nulo no prompt de comandos simulado.

O número de produtos definido para esta simulação foi 4, referente ao número de produtos que consta na Tabela 1.

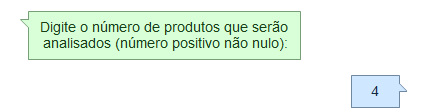


Figura - Definindo o número de produtos que serão analisados.

## Definindo os produtos

Os produtos são definidos um a um, sendo as informações digitadas no prompt de comandos simulado conforme são solicitadas pela simulação do fluxograma.

Após definir o número de produtos a simulação solicitará as informações de nome, preço anterior e preço atual referente ao primeiro produto.

O nome do produto deve ser digitado e não pode estar em branco.

Os valores referentes aos preços devem ser valores positivos não nulo.

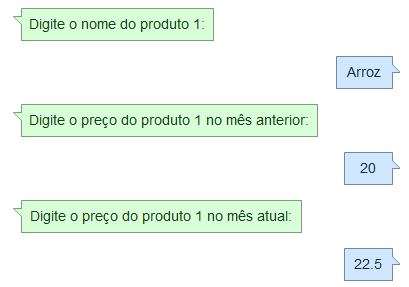


Figura - Informações de nome, preço anterior e preço atual do primeiro produto.

Na sequência é solicitado as informações referentes ao segundo produto.

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

O conteúdo gerado por IA pode estar incorreto.

Figura - Informações de nome, preço anterior e preço atual do segundo produto.

Na sequência é solicitado as informações referentes ao terceiro produto.

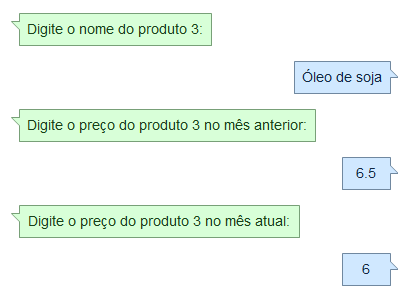


Figura - Informações de nome, preço anterior e preço atual do terceiro produto.

Finalizando com as informações referente a quarto e último produto.

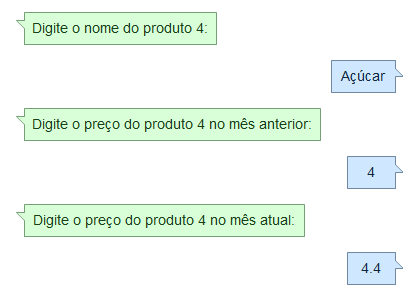


Figura - Informações de nome, preço anterior e preço atual do quarto e último produto.

## Apresentação dos produtos e da análise dos preços.

Após digitar as informações referentes aos quatro produtos que serão analisados a simulação mostrar no prompt simulado as informações referentes a cada produto incluindo a análise dos preços.

É computado a variação percentual de cada produto e verificado o tipo de variação percentual que este produto sofreu.

As informações são mostradas na sequência, iniciando pela análise dos preços do primeiro produto:

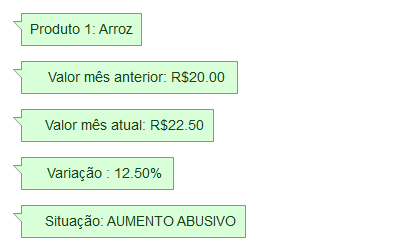


Figura - Análise da variação percentual dos preços do primeiro produto.

Seguindo com a análise dos preços do segundo produto:

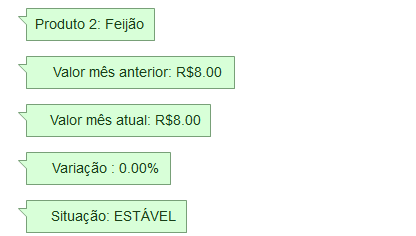


Figura - Análise da variação percentual dos preços do segundo produto.

Depois com a análise dos preços do terceiro produto:

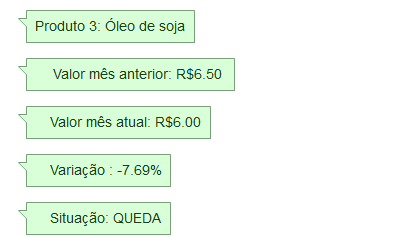


Figura - Análise da variação percentual dos preços do terceiro produto.

E finalizando com a análise dos preços do quarto produto:

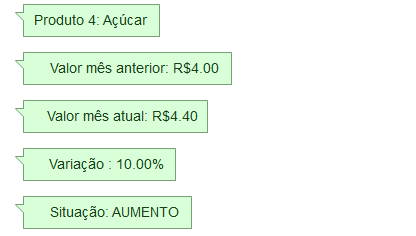


Figura - Análise da variação percentual dos preços do quarto produto.

# Conclusão

O objetivo desta experiência prática ao desenvolver um fluxograma que computasse e analisasse a variação de preços dos produtos de uma cesta básica era apresentar um caso real para aplicação de um algoritmo computacional.

Com o sucesso dos resultados do computo e análise dos dados de variação de preços é possível afirmar que o objetivo de apresentar o pensamento computacional e exercitar a lógica de programação teve êxito.

# Referências

COOK, Devin. *Flowgorithm: Flowchart Programming Language*. Versão 4.5. Disponível em: https://www.flowgorithm.org/ Acesso em: 19 set. 2025.

UNIVERSIDADE DE FRANCA (UNIFRAN). *Experiências Práticas – Fundamentos da Introdução à Programação e ao Pensamento Computacional*. Franca: UNIFRAN, [2025]. Apostila da disciplina Algoritmos e Pensamento Computacional, Curso EAD. Disponível em: https://www.unifran.edu.br/. Acesso em: 19 set. 2025.

SALVI, Fabio Toledo Bonemer De. *basic-food-basket-flowchart*. GitHub, 2025. Disponível em: <https://github.com/engfabiodesalvi/basic-food-basket-flowchart>. Acesso em: 19 set. 2025.