



Serviço Nacional de Aprendizagem Industrial  
PELO FUTURO DO TRABALHO

SÉRIE TI - SOFTWARE

# LÓGICA DE PROGRAMAÇÃO

## VARIÁVEIS E CONSTANTES



## **CONFEDERAÇÃO NACIONAL DA INDÚSTRIA – CNI**

*Robson Braga de Andrade*  
Presidente

### **GABINETE DA PRESIDÊNCIA**

*Teodomiro Braga da Silva*  
Chefe do Gabinete - Diretor

### **DIRETORIA DE EDUCAÇÃO E TECNOLOGIA - DIRET**

*Rafael Esmeraldo Lucchesi Ramacciotti*  
Diretor de Educação e Tecnologia

### **SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL - SENAI**

*Robson Braga de Andrade*  
Presidente do Conselho Nacional

### **SENAI – Departamento Nacional**

*Rafael Esmeraldo Lucchesi Ramacciotti*  
Diretor-Geral

*Julio Sergio de Maya Pedrosa Moreira*  
Diretor-Adjunto

*Gustavo Leal Sales Filho*  
Diretor de Operações



*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**

SÉRIE TI - SOFTWARE

# LÓGICA DE PROGRAMAÇÃO

VARIÁVEIS E CONSTANTES



© 2020. SENAI – Departamento Nacional

© 2020. SENAI – Departamento Regional de Santa Catarina

A reprodução total ou parcial desta publicação por quaisquer meios, seja eletrônico, mecânico, fotocópia, de gravação ou outros, somente será permitida com prévia autorização, por escrito, do SENAI.

Esta publicação foi elaborada pela equipe de Educação a Distância do SENAI de Santa Catarina, com a coordenação do SENAI Departamento Nacional, para ser utilizada por todos os Departamentos Regionais do SENAI nos cursos presenciais e a distância.

**SENAI Departamento Nacional**

Unidade de Educação Profissional e Tecnológica - UNIEP

**SENAI Departamento Regional de Santa Catarina**

Gerência de Educação

**SENAI**

Serviço Nacional de  
Aprendizagem Industrial  
Departamento Nacional

**Sede**

Setor Bancário Norte • Quadra 1 • Bloco C • Edifício Roberto  
Simonsen • 70040-903 • Brasília – DF • Tel.: (0xx61) 3317-  
9001 Fax: (0xx61) 3317-9190 • <http://www.senai.br>

# Lista de Ilustrações

---

Figura 1 - Tela do VisualG – Declarando variáveis no Portugol.....	13
Figura 2 - Estrutura de Algoritmo no VisualG .....	13
Figura 3 - Tela da IDE Eclipse – Declarando variáveis no Java .....	14
Figura 4 - Estrutura de Algoritmo na IDE Eclipse .....	14
Figura 5 - Tela do VisualG – Informando Valores para as Variáveis.....	15
Figura 6 - Tela da IDE Eclipse – Informando Valores para as Variáveis.....	16
Figura 7 - Tela da IDE Eclipse – Declaração de variável constante FINAL .....	17
Figura 8 - Tela da IDE Eclipse – Criando enum TipoProduto .....	18
Figura 9 - Tela da IDE Eclipse – Inserindo enum na classe de produto da lanchonete.....	18
Figura 10 - Tela do VisualG – Variáveis do Monitor de Fluxo.....	19
Figura 11 - Tela da IDE Eclipse – Variáveis do Monitor de Fluxo .....	19
Figura 12 - Protótipo do Monitor dos Canos .....	20
Figura 13 - Tela do VisualG – Variáveis do Registro de Remédio .....	22
Figura 14 - Tela da IDE Eclipse – Variáveis do Registro de Remédio.....	22
Figura 15 - Tela da IDE Eclipse – Variáveis do Registro de Remédio Inicializadas.....	23
Tabela 1 - Tabela proposta de tipo de dado para Primeira Situação .....	10
Tabela 2 - Tabela proposta de tipo de dado para Terceira Situação.....	11
Tabela 3 - Necessidade para situação de controle de livros .....	12
Tabela 4 - Sintaxe de atribuição de variável em Portugol.....	16
Tabela 5 - Sintaxe de atribuição de variável em Java .....	17
Tabela 6 - Tabela proposta de tipo de dado para Segunda Situação .....	21



# Sumário

---

Variáveis e Constantes.....	9
Apresentação.....	9
Definição .....	9
Estrutura .....	11
Declaração de Variável em Portugol – VisualG .....	13
Declaração de Variável em Java – IDE Eclipse .....	14
Aplicação na indústria .....	19
Exemplos .....	20
Referências.....	24





# Variáveis e Constantes



## APRESENTAÇÃO

Olá! O assunto tratado aqui será sobre Variáveis e Constantes!

Os tipos de dados, são justamente a forma de identificar a natureza de uma determinada informação e qualificá-la para o seu futuro armazenamento. E, como seria esse armazenamento? É justamente esse tema que será apresentado neste tópico, ou seja, como criar variáveis e constantes, recursos esses utilizados para guardar informações dos mais variados tipos.

Bons estudos!

## DEFINIÇÃO

Acompanhe inicialmente os conceitos de variável e de constante. A variável é um recurso que se cria para definir um local apropriado para guardar uma determinada informação, que pode variar infinitamente de valores. Exatamente pelo fato de o valor da informação poder “variar” infinitamente, que se dá o nome de Variável ao recurso.

Na tabela “Tabela proposta de tipo de dado para Primeira Situação” conforme você pode acompanhar na sequência, verifica-se que existe a necessidade de guardar o ano de fabricação dos carros.

PRIMEIRA SITUAÇÃO			
NECESSIDADE	TIPO DE DADO	PORTUGOL	JAVA
Marca do carro	Literal	caractere	String
Modelo	Literal	caractere	String
Ano de fabricação	Número Inteiro	inteiro	int
Ano de modelo	Número Inteiro	inteiro	int
Cor do veículo	Literal	caractere	String
Valor de compra	Número Real	real	double
Valor proposto para venda	Número Real	real	double
Direção hidráulica, vidros elétricos, ar-condicionado, freios ABS, banco em couro e teto solar	Lógico (para cada item)	logico	boolean

Tabela 1 - Tabela proposta de tipo de dado para Primeira Situação  
Fonte: do Autor (2020)

Além disso, consta inclusive na tabela que o tipo do dado são números inteiros. Bem, para que isso realmente ocorra, é preciso criar uma variável no sistema que tenha a competência de guardar números inteiros quaisquer. Veja que o valor do ano de um carro para outro pode variar significativamente, ou seja, não é possível estipular que os resultados seriam constantes entre uma ou outra opção de valor. Ao contrário, é evidente que os anos dos carros vão sempre “variar”, exatamente porque adotam a lógica do ano corrente da fabricação daquele veículo: ora ele pode ser, por exemplo, do ano 2020, ora de 1999, de 2014, ou até mesmo carros que ainda estão por vir enquanto o sistema estiver em uso, como 2100, quem sabe?

E o que seriam constantes? Ao contrário das variáveis, as constantes possuem valores definidos e conhecidos, e as possibilidades de escolha dos valores que serão armazenados estão dentro de um domínio constante de valores. No caso a “Tabela proposta de tipo de dado para Terceira Situação”.

TERCEIRA SITUAÇÃO			
NECESSIDADE	TIPO DE DADO	PORTUGOL	JAVA
Nome do produto	Literal	caractere	String
Valor da venda	Número Real	real	double
Tipo do produto	Literal	caractere	String
Descrição do produto	Literal	caractere	String
Customizável	Lógico	logico	boolean
Descrição da customização	Literal	caractere	String

Tabela 2 - Tabela proposta de tipo de dado para Terceira Situação  
Fonte: do Autor (2020)

Veja que se refere à necessidade de informar o “Tipo do Produto”. Talvez na proposta da terceira situação, seja possível definir uma constante, chamada, por exemplo, **tipo\_produto** e nela fornecer os possíveis valores que podem ser escolhidos em um conjunto predeterminado (e portanto, constante), como: Bebida, Lanche e Porção. Ou seja, para essa constante **tipo\_produto**, o usuário poderia selecionar dentre essas opções. Então, se tivessem, por exemplo, 100 registros de produtos no sistema, eles iriam ser sempre de um dentre os valores constantes apresentados (Bebida, Lanche ou Porção).

## ESTRUTURA

Afinal, como funcionaria a estrutura de declaração de uma variável e de uma constante? É exatamente isso que você irá estudar neste tópico, que será apresentado em pseudocódigo (Portugol) e em uma linguagem tradicional de programação (Java), para que você possa estar bem ambientado com a teoria e com a prática.

Para poder compreender melhor como se declara uma variável, tanto em Portugol (no VisualG) como em Java (no Eclipse), será proposto um caso simples, a partir do qual serão criadas as propostas de declaração de variáveis.

Imagine a seguinte situação: uma pessoa deseja catalogar todos os seus livros e, para isso, gostaria de guardar informações importantes de cada obra em um sistema para, assim, ter de forma mais rápida e confiável as informações acerca de sua coleção. Conversando com a pessoa, ela informa que deseja saber as seguintes informações:

- a) Título do livro;
- b) O autor do livro;
- c) Quantidade de páginas;
- d) Categoria do livro;
- e) Se o livro está emprestado;
- f) Para quem foi emprestado.

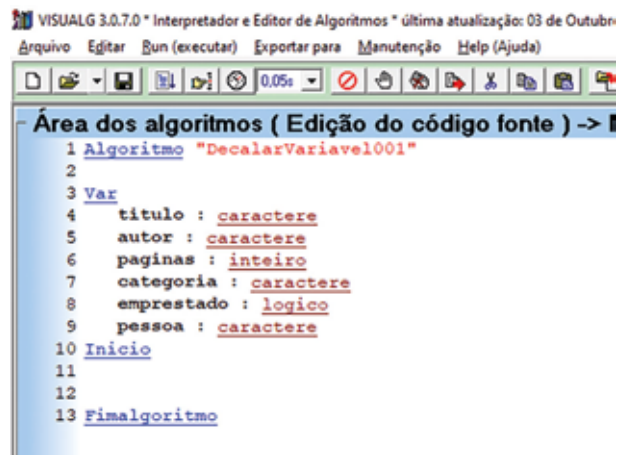
Observe agora a classificação dos tipos de dados para cada uma dessas possíveis variáveis e constantes.

NECESSIDADE	TIPO DE DADO	PORTUGOL	JAVA
Título do Livro	Literal	caractere	String
Autor do Livro	Literal	caractere	double
Quantas Páginas	Número Inteiro	Inteiro	int
Categoria do Livro	Literal	caractere	String
Está Emprestado	Lógico	logico	Boolean
Para quem está Emprestado	Literal	caractere	String

Tabela 3 - Necessidade para situação de controle de livros  
Fonte: do Autor (2020)

Primeiramente, é preciso transformar essas “necessidades” em declaração de variáveis reais, com os seus respectivos tipos de dados identificados.

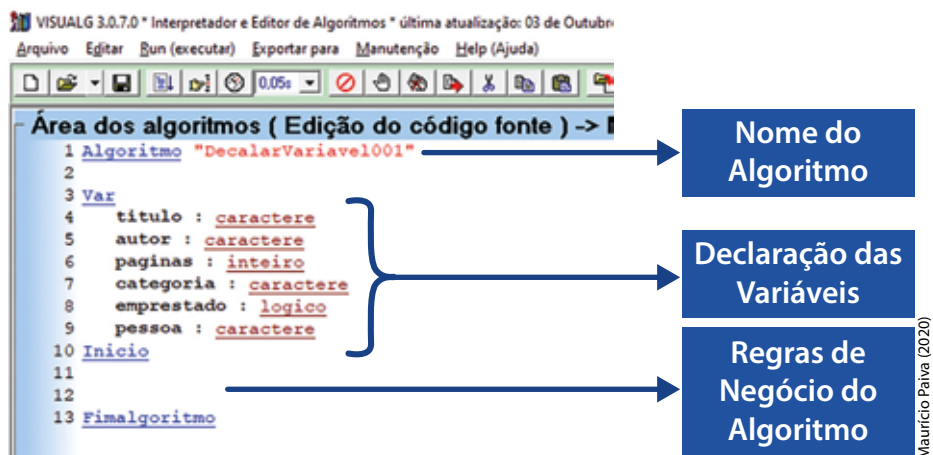
## DECLARAÇÃO DE VARIÁVEL EM PORTUGOL – VISUALG



Unknown (20--?)

Figura 1 - Tela do VisualG – Declarando variáveis no Portugol  
Fonte: do Autor (2020)

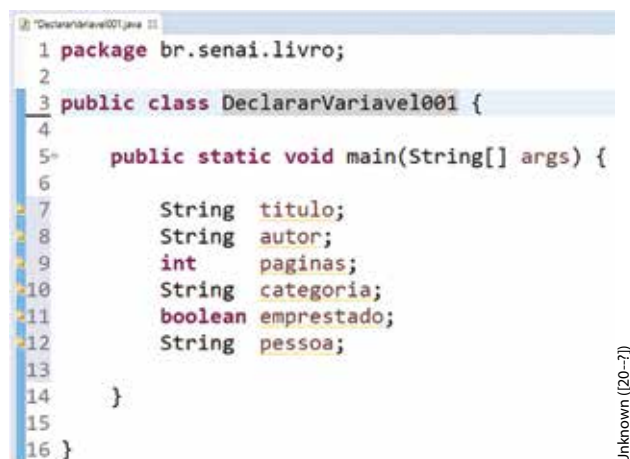
A sintaxe de declaração das variáveis, no caso da linguagem Portugol no VisualG, acontece da seguinte forma:



Maurício Paiva (2020)

Figura 2 - Estrutura de Algoritmo no VisualG  
Fonte: do Autor (2020)

## DECLARAÇÃO DE VARIÁVEL EM JAVA – IDE ECLIPSE



```

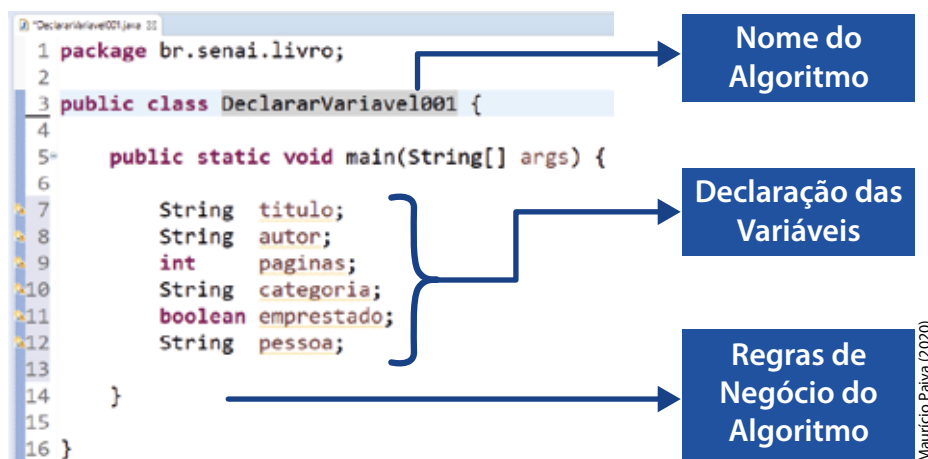
1 package br.senai.livro;
2
3 public class DeclararVariavel001 {
4
5     public static void main(String[] args) {
6
7         String titulo;
8         String autor;
9         int paginas;
10        String categoria;
11        boolean emprestado;
12        String pessoa;
13
14    }
15
16 }

```

Unknown (20--?)

Figura 3 - Tela da IDE Eclipse – Declarando variáveis no Java  
Fonte: do Autor (2020)

Já no Java, a sintaxe para a proposta de um mesmo algoritmo (não utilizando de Programação Orientada a Objetos) poderia ser assim definida:



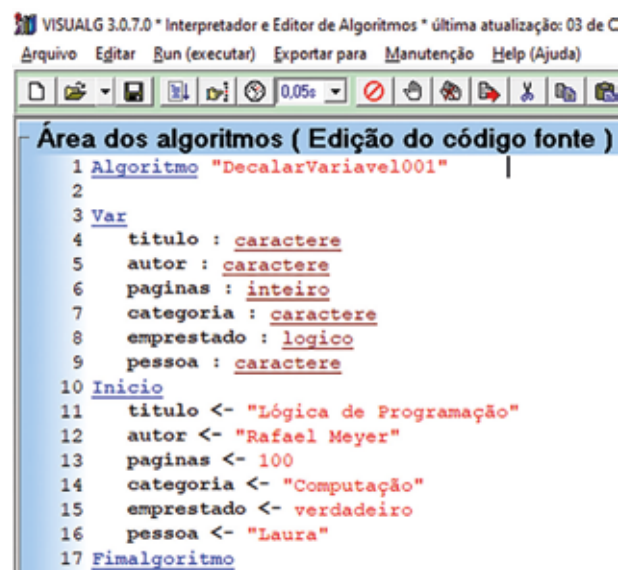
Maurício Paiva (2020)

Figura 4 - Estrutura de Algoritmo na IDE Eclipse  
Fonte: do Autor (2020)

É pertinente destacar que essas ferramentas evoluem constantemente, principalmente a IDE Eclipse. Porém, existem muitos materiais acessíveis na internet que podem auxiliá-lo(a) nessa tarefa. A seguir, algumas dicas de pesquisa para que você possa concluir alguns interessantes passos:

- a) Pesquise: **Como instalar o JDK** (que é Java Development Kit, conjunto de utilitários que permitem criar sistemas de *software* para a plataforma Java);
- b) Pesquise: **Como instalar IDE Eclipse Java** (ferramenta utilizada para programar em Java);
- c) Pesquise: **Como criar um Projeto Java no Eclipse** (para aprender como criar um primeiro projeto na ferramenta Eclipse);
- d) Pesquise: **Como fazer um Hello World em Java no Eclipse** (para aprender a executar seu primeiro programa escrito em Java no Eclipse);
- e) Pesquise: **Manual do VisualG** (para ver toda documentação da ferramenta).

Acompanhe agora essa mesma proposta de declaração de variáveis apresentados na Figura “Declarando variáveis no Portugol” e na Figura “Tela da IDE Eclipse – Declarando variáveis no Java”, porém atribuindo valores para essas variáveis, ou seja, propondo qual a informação que será armazenada em cada uma dessas variáveis.



```
1 Algoritmo "DecalarVariavel001"
2
3 Var
4     titulo : caractere
5     autor : caractere
6     paginas : inteiro
7     categoria : caractere
8     emprestado : logico
9     pessoa : caractere
10 Inicio
11     titulo <- "Lógica de Programação"
12     autor <- "Rafael Meyer"
13     paginas <- 100
14     categoria <- "Computação"
15     emprestado <- verdadeiro
16     pessoa <- "Laura"
17 Fimalgoritmo
```

Figura 5 - Tela do VisualG – Informando Valores para as Variáveis  
Fonte: do Autor (2020)

Unknown (20--?)

```

1 package br.senai.livro;
2
3 public class DeclararVariavel001 {
4
5     public static void main(String[] args) {
6
7         String titulo;
8         String autor;
9         int paginas;
10        String categoria;
11        boolean emprestado;
12        String pessoa;
13
14        titulo = "Lógica de Programação";
15        autor = "Rafael Meyer";
16        paginas = 100;
17        categoria = "Computação";
18        emprestado = true;
19        pessoa = "Laura";
20
21    }
22 }
23

```

Unknown (20--?)

Figura 6 - Tela da IDE Eclipse – Informando Valores para as Variáveis  
Fonte: Fonte: do Autor (2020)

Agora, retornando à declaração das variáveis, analise a sintaxe de atribuição de valor em uma variável tanto no Portugol como no Java.

SINTAXE DA ATRIBUIÇÃO DE INFORMAÇÃO NA VARIÁVEL EM PORTUGOL		
IDENTIFICADOR DA VARIÁVEL	SINAL DE ATRIBUIÇÃO	DECLARAÇÃO DO VALOR
titulo	<-	"Lógica de Programação"
autor	<-	"Rafael Meyer"
paginas	<-	100
categoria	<-	"Computação"
emprestado	<-	verdadeiro
pessoa	<-	"Laura"

Tabela 4 - Sintaxe de atribuição de variável em Portugol  
Fonte: do Autor (2020)

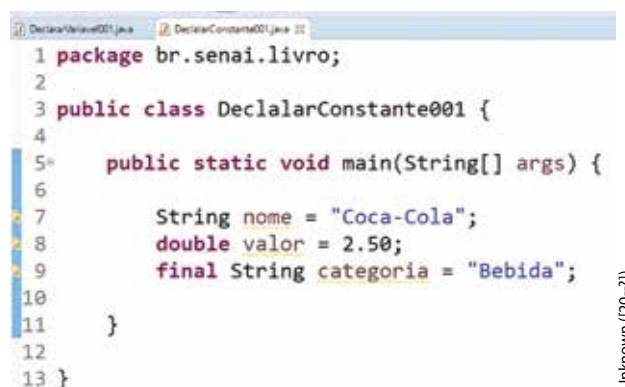


SINTAXE DA ATRIBUIÇÃO DE INFORMAÇÃO NA VARIÁVEL EM PORTUGOL		
IDENTIFICADOR DA VARIÁVEL	SINAL DE ATRIBUIÇÃO	DECLARAÇÃO DO VALOR
titulo	=	"Lógica de Programação";
autor	=	"Rafael Meyer";
paginas	=	100;
categoria	=	"Computação";
emprestado	=	true;
pessoa	=	"Laura"

Tabela 5 - Sintaxe de atribuição de variável em Java  
Fonte: do Autor (2020)

É possível notar que, basicamente, mudou o sinal de atribuição, sendo que no Portugol se usa <- para atribuir um determinado valor a uma variável. No Java, por sua vez, utiliza-se o sinal = para efetuar a mesma ação de atribuir um valor a uma variável. Perceba também que, no caso do Java, no final de cada linha de atribuição, utiliza-se o finalizador de linha de comando ";" para indicar que aquela linha de comando chegou ao fim, recurso esse que não existe no Portugol. A última diferença a ser observada nessas tabelas comparativas é que, no caso do Portugol, o valor lógico atribuído é **verdadeiro**, enquanto no Java é **true**.

Retomando sobre as constantes, que seriam uma forma de criar declarações de variáveis onde os valores serão escolhidos a partir de um conjunto determinado de opções, ou seja, são possibilidades de valores pré-determinados (ou seja, constantes). Nesse caso, mencionamos o exemplo da lanchonete possuir uma constante chamada **tipo\_produto**, onde o valor assumido seria sempre uma das seguintes opções: Bebida, Lanche ou Porção. Então, vamos ver como ficaria uma declaração de constante no Java.



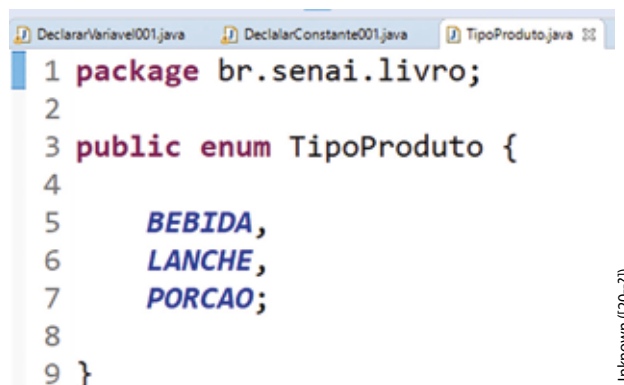
```

1 package br.senai.livro;
2
3 public class DeclaraConstante001 {
4
5     public static void main(String[] args) {
6
7         String nome = "Coca-Cola";
8         double valor = 2.50;
9         final String categoria = "Bebida";
10
11     }
12
13 }

```

Figura 7 - Tela da IDE Eclipse – Declaração de variável constante FINAL  
Fonte: do Autor (2020)

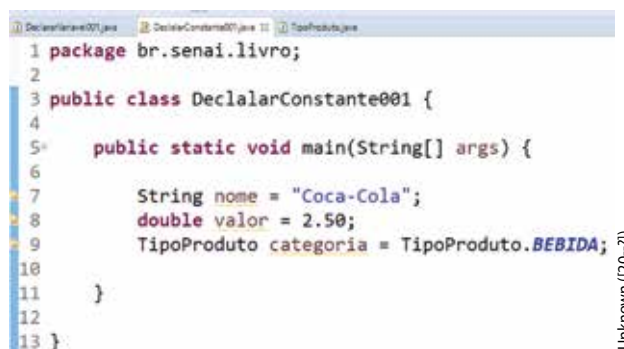
No caso acima, declaramos que a variável categoria seria constantemente atribuída ao valor “Bebida”, ou seja, esse seria seu valor final ao longo de todo o programa, não podendo assumir nenhum outro tipo de valor. Porém podemos também adotar outras formas de declarar variáveis constantes, que podem a partir de um conjunto fixo de opções, alterar seu valor dentro de um conjunto de valores constantes. Vamos ver como podemos criar um tipo **enum** para representar melhor essa variável **tipo\_produto**.



```
1 package br.senai.livro;
2
3 public enum TipoProduto {
4
5     BEBIDA,
6     LANCHE,
7     PORCAO;
8
9 }
```

Figura 8 - Tela da IDE Eclipse – Criando enum TipoProduto  
Fonte: do Autor (2020)

E com isso, poderíamos ter uma outra proposta para aquela classe que estava representando o lanche.



```
1 package br.senai.livro;
2
3 public class DeclalarConstante001 {
4
5     public static void main(String[] args) {
6
7         String nome = "Coca-Cola";
8         double valor = 2.50;
9         TipoProduto categoria = TipoProduto.BEBIDA;
10
11     }
12
13 }
```

Figura 9 - Tela da IDE Eclipse – Inserindo enum na classe de produto da lanchonete  
Fonte: do Autor (2020)

Note que seria possível alterar a categoria desse produto entre três possíveis valores, constantemente apresentados na estrutura e em um TipoProduto. Sendo assim, a variável categoria poderia ser:

- TipoProduto.BEBIDA;
- TipoProduto.LANCHE;
- TipoProduto.PORCAO.

Não sendo possível assumir qualquer outro valor, a não ser um desses da estrutura enum TipoProduto.

## APLICAÇÃO NA INDÚSTRIA

A utilização de variáveis e constantes no setor produtivo é gigantesca, pois há inúmeras aplicações reais que podem acontecer e que necessitam da criação de programas para atender ao universo de negócios.

Para elucidar mais o aprendizado, acompanhe o seguinte cenário, que deverá ser resolvido a partir de um algoritmo computacional utilizando as duas linguagens apresentadas, Portugol e Java.

Uma indústria que fabrica produtos químicos precisa criar um programa que acompanhe a vazão de determinados líquidos em alguns canos na empresa, para poder aumentar sua capacidade de monitoramento de vazamentos.

assim, a fábrica gostaria de ter um programa que monitorasse em cada cano as seguintes questões: o fluxo de litros por segundo que passa naquele ponto do cano, qual o tipo do líquido (sendo que só podem ser 2 tipos: solvente e aditivos) e a temperatura desse líquido. Dessa forma, primeiramente é preciso criar essa proposta em Portugol e depois em Java, declarando as variáveis e seus respectivos tipo de dados.

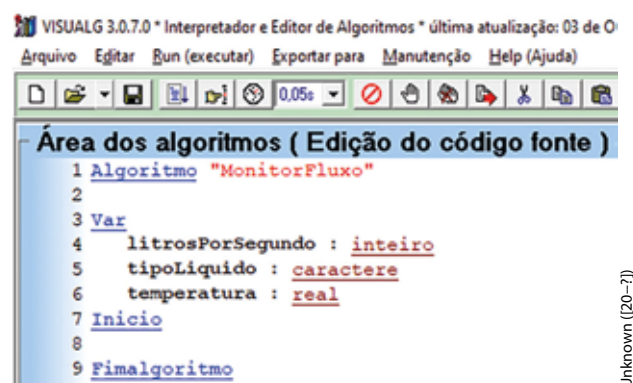


Figura 10 - Tela do VisualG – Variáveis do Monitor de Fluxo  
Fonte: do Autor (2020)

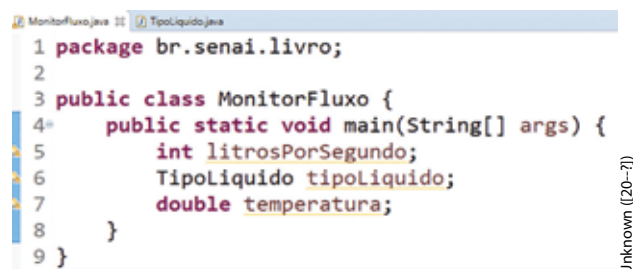


Figura 11 - Tela da IDE Eclipse – Variáveis do Monitor de Fluxo  
Fonte: do Autor (2020)

A variável **tipoLiquido** na proposta em Java é um enum, e seus possíveis valores seriam:

- a) TipoLiquido.SOLVENTE;
- b) TipoLiquido.ADITIVO.

Agora fica fácil imaginar que, a partir de um equipamento que possui um conjunto de sensores, os valores seriam coletados em tempo real e informados para os programas, que seriam exibidos em um terminal acoplado ao cano.

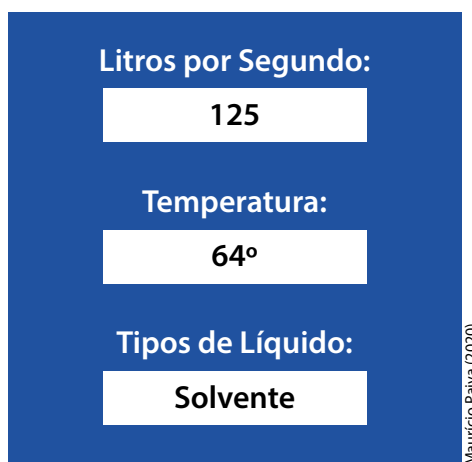


Figura 12 - Protótipo do Monitor dos Canos  
Fonte: do Autor (2020)

Futuramente, conforme avançarmos nos próximos tópicos, vamos conseguir perceber que poderíamos inserir diversas regras de negócio nessas classes, dando um comportamento muito mais interessante do que meramente mostrar os valores dos sensores no monitor. Poderíamos, por exemplo, criar regras de alarmes sonoros se um determinado tipo de líquido fosse colocado por engano naquele cano, ou, quem sabe, outro aviso sonoro caso a temperatura ou vazão de litros por segundo fossem inferiores ou superiores ao apropriado.

## EXEMPLOS

Vamos reforçar mais um pouco o conceito de declaração das variáveis e seus tipos, apresentando soluções para a situação-problema exposta nessa tabela:

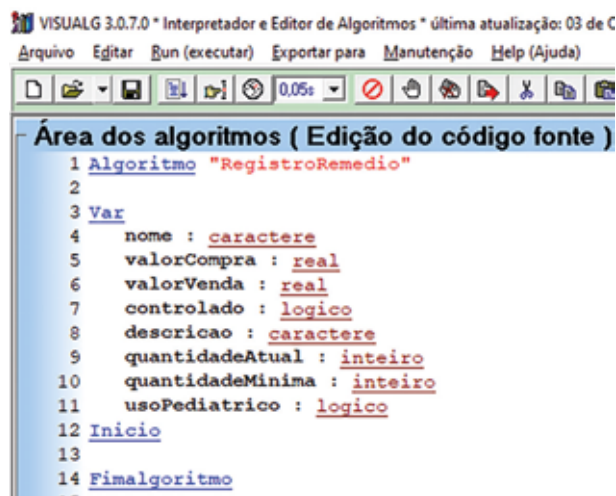
SEGUNDA SITUAÇÃO			
NECESSIDADE	TIPO DE DADO	PORTUGOL	JAVA
Nome do remédio	Literal	caractere	String
Valor de compra	Número Real	real	double
Valor de venda	Número Real	real	double
Remédio é controlado	Lógico	logico	boolean
Descrição sobre o remédio	Literal	caractere	String
Restrições de uso	Literal	caractere	String
Quantidade atual	Número Inteiro	inteiro	int
Quantidade mínima	Número Inteiro	inteiro	int
Uso pediátrico	Lógico	logico	boolean

Tabela 6 - Tabela proposta de tipo de dado para Segunda Situação  
Fonte: do Autor (2020)

A ideia aqui era criar um programa que fosse possível guardar as seguintes informações:

- nome do remédio;
- valor de compra;
- valor de venda;
- remédio é controlado ou não;
- descrição sobre o remédio;
- restrições de uso;
- quantidade atual;
- quantidade mínima;
- uso pediátrico.

Sendo assim, vamos apresentar a proposta da solução em Portugol e em Java.



VisualG 3.0.7.0 \* Interpretador e Editor de Algoritmos \* última atualização: 03 de C

Arquivo Editar Run (executar) Exportar para Manutenção Help (Ajuda)

0.05s

Área dos algoritmos ( Edição do código fonte )

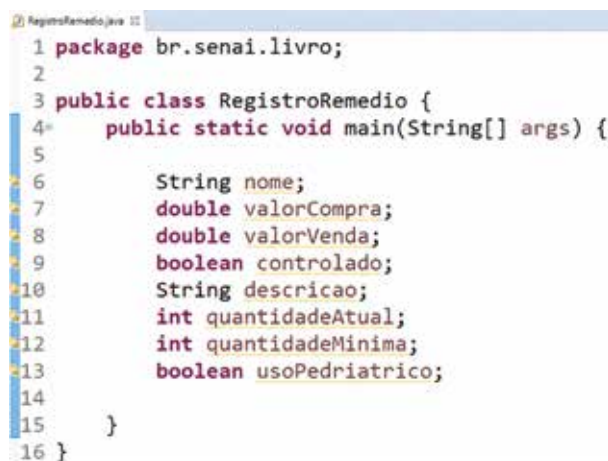
```

1 Algoritmo "RegistroRemedio"
2
3 Var
4   nome : caractere
5   valorCompra : real
6   valorVenda : real
7   controlado : logico
8   descricao : caractere
9   quantidadeAtual : inteiro
10  quantidadeMinima : inteiro
11  usoPediatrico : logico
12 Inicio
13
14 Fimalgoritmo

```

Unknown (20--?)

Figura 13 - Tela do VisualG – Variáveis do Registro de Remédio  
Fonte: do Autor (2020)



```

1 package br.senai.livro;
2
3 public class RegistroRemedio {
4     public static void main(String[] args) {
5
6         String nome;
7         double valorCompra;
8         double valorVenda;
9         boolean controlado;
10        String descricao;
11        int quantidadeAtual;
12        int quantidadeMinima;
13        boolean usoPediatrico;
14
15    }
16 }

```

Unknown (20--?)

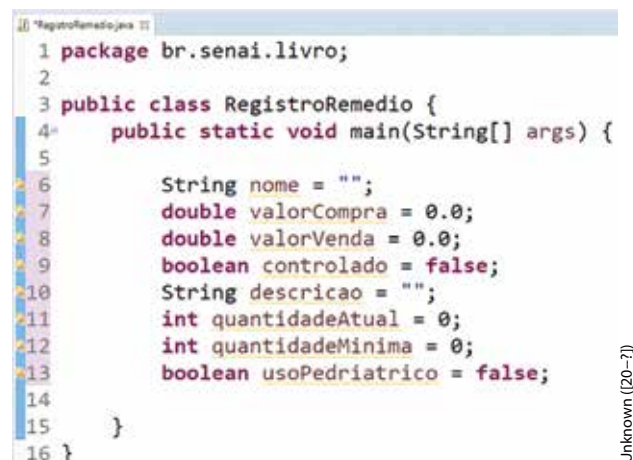
Figura 14 - Tela da IDE Eclipse – Variáveis do Registro de Remédio  
Fonte: do Autor (2020)



**SAIBA  
MAIS**

Ao trabalhar com declaração de variáveis, é possível, muitas vezes, já iniciá-las com algum valor padrão, para evitar, por exemplo, que alguma variável possua um valor nulo (*null*) durante sua execução.

Para exemplificar, vamos pegar a proposta das variáveis de registro de remédio e vamos agora, além de declarar as variáveis e definir seus tipos, também definir quais os valores padrões iniciais dessas variáveis.



```
1 package br.senai.livro;
2
3 public class RegistroRemedio {
4     public static void main(String[] args) {
5
6         String nome = "";
7         double valorCompra = 0.0;
8         double valorVenda = 0.0;
9         boolean controlado = false;
10        String descricao = "";
11        int quantidadeAtual = 0;
12        int quantidadeMinima = 0;
13        boolean usoPediatrico = false;
14
15    }
16 }
```

Figura 15 - Tela da IDE Eclipse – Variáveis do Registro de Remédio Inicializadas  
Fonte: Fonte: do Autor (2020)

Perceba que nesse caso, as variáveis assumem um valor padrão (default) e são atribuídos durante sua declaração. Muitas linguagens de programação solicitam esse tipo de comportamento (inicialização obrigatória), outras, porém, permitem que você possa declarar os valores em outros momentos da execução.

Falando nisso, procure pesquisar também sobre linguagens **fortemente tipadas** e linguagem **fracamente tipadas**, pois são características importantes para que você conheça, principalmente quando falamos de declaração de variáveis.

\*\*\*

## REFERÊNCIAS

---

SOUZA, Marco Antonio de; GOMES, Marcelo Marques; SOARES, Márcio José; CONCILIO, Ricardo. **Algoritmos e lógica de programação**. São Paulo (SP): Thomson Pioneira, 2005.

ARAÚJO, Everton Coimbra de. **Algoritmos**: fundamentos e pratica. Florianópolis: Visual Books, 2005.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. **Algoritmos**: lógica para desenvolvimento de programação de computadores. 24. ed. São Paulo (SP): Érica, 2011.

ANSELMO, Fernando. **Aplicando lógica orientada a objetos em Java**: da lógica à certificação. 3. ed. Florianópolis: Visual Books, 20







*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**