

**SÓ PROGRAMAÇÃO: A PROJEÇÃO DE UM SISTEMA EM  
DISTINTAS LINGUAGENS DE PROGRAMAÇÃO**

**ONLY PROGRAMMING: THE PROJECTION OF A SYSTEM IN  
DIFFERENT PROGRAMMING LANGUAGES**

**SÓLO PROGRAMA: LA PROYECCIÓN DE UN SISTEMA EN  
DISTINTAS LENGUAJES DE PROGRAMACIÓN**

**Carla Gonçalves Távora**  
**carlag.tavora@hotmail.com**

**Faculdade de Tecnologia do Estado de São Paulo – FATEC Garça/SP**

**Gabriel Lino Garcia**  
**gabriel\_linao@hotmail.com**

**Faculdade de Tecnologia do Estado de São Paulo – FATEC Garça/SP**

**Carlos Eduardo da Trindade Ribeiro**  
**Carlos.trindade@cps.sp.gov.br**

**Universidade Estadual Paulista “Júlio de Mesquita Filho”, UNESP-Bauru/SP**

**Everton Simões da Motta**  
**evmotta08@gmail.com**

**Universidade Estadual Paulista “Júlio de Mesquita Filho”, UNESP-Bauru/SP**

**Resumo**

O artigo tem como finalidade fornecer breves contextualizações e diferentes instruções de linguagens envolvidas no projeto – onde foram selecionadas pelas revistas TecMundo e Exame – com o propósito em auxiliar usuários iniciantes em seus estudos e projetos. O trabalho realizou o desenvolvimento teórico através de uma pesquisa sobre a história da linguagem de programação, além de utilizar a apostila – Universidade Federal do Rio Grande do Norte: Centro de Tecnologia, Departamento de Computação e Automação como referência para a explicação da lógica de programação computacional, resultando no desenvolvimento do grupo de pesquisa intitulado “Só Programação”, composto pelos discentes da Fatec da cidade de Garça. O grupo foi realizado no ano de 2018 com os alunos do curso de Análise e Desenvolvimento de Sistemas, o objetivo em questão é desenvolver um projeto de ensino em linguagens de programação, desenvolvendo sistemas simples para exemplos, conceituando raciocínio lógicos e matemáticos.

Palavras chave: Programas Computacionais; ciência; tecnologia.

**Abstract**

The article aims to provide brief different instructions of languages, selected by the magazines TecMundo and Exame with the purpose to assist novice users in their studies and projects. The work carried out the theoretical development through a research on the history of the programming language, besides using the apostille – Federal University of Rio Grande do Norte: Center for Technology, Department of Computing and Automation as a reference for the explanation of the computational programming logic, resulting in the development of the research

group called "Only Programming", composed by the students of Fatec of the city of Garça. The group was held in the year 2018 with the students of the course of Analysis and Development of Systems, the objective in question is to develop a teaching project in programming languages, developing simple systems for examples, conceptualizing logical and mathematical reasoning.

Keywords: Computer Programs; science; technology.

### Resumen

El artículo tiene como finalidad proporcionar breves contextualizaciones y diferentes instrucciones de lenguajes involucrados en el proyecto - donde fueron seleccionadas por las revistas TecMundo y Examen - con el propósito de auxiliar a los usuarios principiantes en sus estudios y proyectos. El trabajo realizó el desarrollo teórico a través de una investigación sobre la historia del lenguaje de programación, además de utilizar la apostilla - Universidad Federal de Rio Grande do Norte: Centro de Tecnología, Departamento de Computación y Automatización como referencia para la explicación de la lógica de programación y en el desarrollo del grupo de investigación titulado "Sólo Programación", compuesto por los alumnos de Fatec de la ciudad de Garça. El grupo fue realizado en el año 2018 con los alumnos del curso de Análisis y Desarrollo de Sistemas, el objetivo en cuestión es desarrollar un proyecto de enseñanza en lenguajes de programación, desarrollando sistemas simples para ejemplos, conceptualizando raciocinio lógico y matemático.

Palabras clave: Programas Computacionales; ciencia; tecnología.

### INTRODUÇÃO

A linguagem de programação é uma ferramenta utilizada pelos profissionais de desenvolvimento de software para escrever programas, ou seja, conjuntos de instruções representadas por uma combinação de símbolos a ser seguida pelo computador de modo a realizar determinada tarefa ou processo (VAREJÃO, 1995). “Os programas de computadores nada mais são do que algoritmos escritos numa linguagem de computador e que são interpretados e executados por uma máquina, no caso um computador” (MORAES, 2000, p.5).

A história da linguagem de programação é apresentada e descrita por cinco gerações, segundo Lauer & Sobrinho (2008):

- Primeira geração: linguagem de máquina, programação binária;
- Segunda geração: primeiras linguagens Assembly;
- Terceira geração: primeiras linguagens de meio e alto nível;
- Quarta geração: capazes de gerar código por si só;
- Quinta geração: linguagens orientadas a inteligência artificial.

Complementando as gerações de linguagens, a 1ª Geração (1GL) é caracterizada como linguagem de máquina, pois é, baseada no código binário com as sequências de zeros e uns. A 2ª Geração, uma linguagem assembler (Assembly) ou linguagem de montagem, considerada uma linguagem de baixo nível por ser facilmente manipulado. A 3ª Geração representa a criação da primeira linguagem de alto nível, uma linguagem entendida com mais facilidade pelos humanos, pois assemelhava-se a linguagem humana. A 4ª Geração (4GL) também conhecida por uma linguagem de alto nível por ser uma linguagem orientada ao objeto, facilitando o desenvolvimento para os usuários e reutilizando códigos para outros programas. A 5ª Geração é uma linguagem orientada a inteligência artificial (IA), além de ser pouco desenvolvida e com muitas restrições tem o propósito de entender a mente humana. A criação da 6ª Geração ainda permanece em processo de construção, pois, envolve as redes neurais artificiais (RNA's), ou seja, a interconexão entre um sistema nervoso central com a computação.

As linguagens de programação tem como função servir como um meio de comunicação entre computadores e humanos, segundo Morimoto (2007), as gerações de linguagens citadas foram desenvolvidas com o propósito de facilitar o processo dessa comunicação.

O objetivo do trabalho é explicar a elaboração de um programa computacional, ou seja, “uma ordem ou um comando que indica ao computador uma ação elementar a ser executada” (MORAES, 2000, p.4), denominado como uma sequência lógica padronizada permitindo resoluções de problemas utilizando os computadores. Além de apresentar o objetivo do grupo – formado pelos alunos da FATEC da cidade de Garça – em que disponibilizam opções de linguagens com a intenção de que os usuários iniciantes possam escolher qual tem mais afinidade, concluindo na formação de projetos computadorizados em diversas linguagens de programação com a função de desenvolver técnicas inteligentes e logicas, a fim de identificar um algoritmo eficaz.

## **METODOLOGIA**

O estudo predominou-se no levantando teórico através de pesquisas em sites, livros e artigos científicos dentro da área. Esses materiais tiveram um importante papel para o entendimento da linguagem de programação e como o planejamento de um programa é interpretado, possibilitando a projeção de sistemas em diferentes linguagens.

Embasando-se em um sistema consistente a uma composição de algoritmos ordenados com uma lógica clara e concisa no qual possam ser analisados, identificados e modificados até alcançar uma melhor performance no processamento. De acordo com Pinochet (2014, p.41), o sistema computacional é criado pelo ser humano, tendo como função uma determinada finalidade para satisfazer uma necessidade que é a razão de sua existência.

O significado de um programa é definido pela sua semântica, ou seja, quando um programa é executado, o efeito de cada comando sobre os valores das variáveis no programa é dado pela semântica da linguagem. Assim, quando escrevemos um programa, devemos entender ideias básicas como o efeito exato que uma atribuição possui sobre as variáveis do programa. Se tivermos um modelo semântico que seja independente de alguma plataforma específica, podemos aplicá-lo a uma diversidade de máquinas nas quais essa linguagem possa ser implementada (TUCKER & NOONAN, 2010, p.3).

As dificuldades existentes em diversas linguagens dentro e fora do curso de Ciência da Computação, originou na criação do projeto “Só Programação” com objetivo de auxiliar usuários nas primeiras etapas de estudo ao disponibilizar uma linguagem mais informal e breves sintaxes para a elaboração de um algoritmo envolvendo lógicas matemáticas. O grupo de alunos da FATEC da cidade de Garça estudaram as diferentes linguagens para a projeção do trabalho, resultando em etapas específicas e descritivas:

1. Esse trabalho representa uma breve descrição de cada linguagem estudada para a elaboração do projeto “Só Programação”;
2. Uma introdução sobre as sintaxes utilizadas para o entendimento da elaboração de um algoritmo;
3. Cada linguagem citada será utilizada para o desenvolvimento de sistemas;
4. Cada sistema elaborado terá uma apostila complementar sobre a linguagem designada.

## **ANALISE DE ALGORITMO**

A área da tecnologia é um processo contínuo de crescimento, a linguagem de programação está sempre sendo aperfeiçoada, atualmente são mais de 600 linguagens de programação, mas a forma de elaboração nunca muda, segundo o portal TecMundo (2017).

O entendimento da linguagem é necessário para compreensão da elaboração do algoritmo, a apostila feita pela Universidade Federal do Rio Grande do Norte: Centro de

Tecnologia, Departamento de Computação e Automação (2004), expõe algumas formas de algoritmos onde tratam os problemas apenas em nível lógico, abstraindo-se de detalhes de implementação muitas vezes relacionados com alguma linguagem de programação específica. Por outro lado existem formas de representação de algoritmos que possuem uma maior riqueza de detalhes e muitas vezes acabam por obscurecer as ideias principais do algoritmo, dificultando seu entendimento. Dentre as formas de representação de algoritmos mais conhecidas podemos citar:

- Descrição Narrativa;
- Fluxograma Convencional (representação gráfica de algoritmos implicando ações);
- Pseudocódigo (conhecido como Linguagem estruturada ou Portugol, uma representação mais detalhada com a utilização de tipos de variáveis).

A elaboração desses três exemplos de algoritmos é necessário o entendimento dos operadores de cálculo; lógicos e relacionais para uma maior compreensão nas etapas até encontrar uma solução. São 3 exemplos de operadores:

Tabela 1 – A Operação Aritméticas em um algoritmo realizará cálculos utilizando expressões matemáticas ou aritméticas.

<b>Operadores Aritméticos</b>	<b>Símbolos</b>
Soma	+
Subtração	-
Multiplicação	*
Divisão	/
Exponenciação	** OU /

Fonte: Créditos dos Autores

Tabela 2 – As Operações Relacionais fazem comparações entre dados numéricos e literais (caracteres):

<b>Operadores Relacionais</b>	<b>Símbolos</b>
Maior que	>
Menor que	<
Maior ou igual a	>=
Menor ou igual a	<=
Igual a	=
Maior que	<> OU # OU !=

Fonte: Créditos dos Autores

Tabela 3 – As Operações Lógicas – chamados de booleanos – permitem comparar ou relacionar resultados de expressões:

<b>Operadores Lógicos</b>	<b>Símbolos</b>
Conjunção	E
Disjunção (não-exclusiva)	OU
Negação	NÃO
Disjunção (exclusiva)	XOU

Fonte: Créditos dos Autores

A lógica orienta o funcionamento do computador através de instruções ou comandos, para a eficiência do processo é necessário a codificação por meio de elementos ou atributos de programação. O algoritmo é representado por começo, meio e fim para alcançar um objetivo, representando uma estrutura de controle com mais complexidade ao código.

“Para que o computador execute um algoritmo, o mesmo deve ser escrito de uma forma que o processador esteja apto a executá-lo” (MEDINA, 2005). Assim, abaixo representaremos exemplos de como o algoritmo é planejado e interpretado:

A descrição narrativa é composta por ações até atingir um fim específico, a apostila da UFRN usa como exemplo a ação de uma pessoa tomando banho:

1. Entrar no banheiro e tirar a roupa
2. Abrir a torneira do chuveiro
3. Entrar na água
4. Ensaboar-se
5. Sair da água
6. Fechar a torneira
7. Enxugar-se
8. Vestir-se

Uma pessoa pode desenvolver um algoritmo com as tarefas que precisam ser concluídas, assim como um computador realizando tarefas, a lógica da descrição narrativa orienta as decisões do dia a dia, definindo as etapas necessárias para a conclusão. A técnica da descrição narrativa é considerada mais fácil e simples de ser entendida e utilizada, pois, é a linguagem mais próxima da comunicação humana.

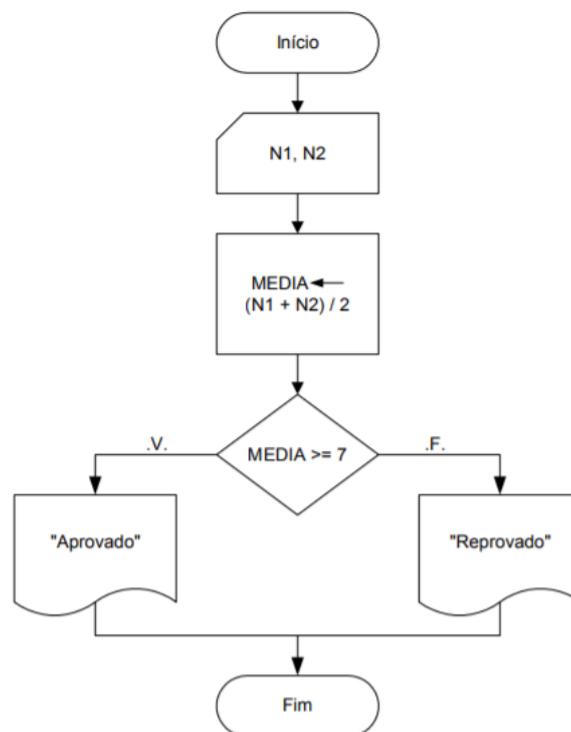
A estrutura de seleção utilizado pela apostila da UFRN são as condições “Se” e “Senão”, nesse exemplo as condições tem finalidade em calcular a média do aluno em uma disciplina e estabelecer caminhos diferentes de instruções a serem percorridos a partir de tomadas de decisões, ou seja, demonstra dois caminhos que o aluno poderá obter dependendo da nota final.

O algoritmo em descrição narrativa utiliza as condições citadas para calcular a média de um aluno:

1. Obter as suas 2 notas de provas
2. Calcular a média aritmética
3. **Se** a média for maior que 7, o aluno foi aprovado, **senão** ele foi reprovado

Esse mesmo algoritmo também é representado na apostila de UFRN como um algoritmo de fluxograma convencional, uma representação gráfica da tarefa que o computador realizará para atingir um final desejado:

Figura 1: Fluxograma Convencional



Fonte: UFRN, 2008.

Esse exemplo, representa um algoritmo formado por diagramada de blocos, a função desse diagrama é demonstrar como o computador realizará o processamento de calcular a média, obtendo um caminho final para cada aluno.

Um pseudocódigo depende de regras rígidas para a elaboração, em outras palavras, o pseudocódigo utiliza comandos Leia e Escreva para descrever a tarefa, respectivamente. A apostila UFRN utiliza o mesmo algoritmo “Calcular a média do aluno” para demonstrar um algoritmo em forma de pseudocódigo:

1. **Algoritmo** Calculo\_Media
2. **Var** N1, N2, MEDIA: **real**
3. **Inicio**
4. **Leia** N1, N2
5.  $MEDIA \leftarrow (N1 + N2) / 2$
6. **Se** MEDIA  $\geq$  7 **então**
- 6.1 **Escreva** “Aprovado”
7. **Senão**
- 7.1 **Escreva** “Reprovado”
8. **Fim\_Se**
9. **Fim.**

As linhas do algoritmo pseudocódigo apresentam diferentes significados, a linha 1 representa o nome criado para o seu algoritmo, a linha 2 caracteriza os tipos de variáveis e expressões aritméticas que serão utilizados para calcular a média, as variáveis são expansões em memórias que tem o objetivo de guardar um tipo de dado ou informação, como por exemplo, as três variáveis N1, N2 e Media são do tipo **real**, ou seja, podem representar números inteiros ou quebrados com a função de diferenciar os nomes das variáveis. As instruções do corpo do algoritmo é escrito entre as linhas 3 e 9, o algoritmo utiliza a Estrutura de Controle, as condições **Se** e **Senão** nas linhas 6 e 7 propondo dois caminhos para o programa ser executado, a fim de solucionar o fim das medias dos alunos.

## RESULTADOS E DISCUSSÕES

A linguagem de programação entendida pelo computador é composta somente por números também conhecida como linguagem de máquina, desta forma, seria um trabalho extremamente difícil para o programador escrever algoritmos nesta linguagem,

sendo assim, novas linguagens de programação foram desenvolvidas ao longo dos anos para facilitar o processo de programação e escrita desses algoritmos (MEDINA, 2005).

De acordo com o portal TecMundo (2017) foi elaborado uma lista das linguagens de programação mais importantes em 2018:

1. Java;
2. JavaScript;
3. Python;
4. C++;
5. C#.

A revista Exame (2017) utiliza os sites Coding Dojo e o Indeed como referência para a pesquisa das principais linguagens de programação e acrescenta mais 2 linguagens nas que o site TecMundo (2017) citou:

1. PHP;
2. Perl.

Estas diferentes linguagens não podem ser traduzidas diretamente entre si, pois além de serem representadas de modos diferentes, também referem-se a coisas muito distintas (FERRARI & CECHINEL 2008, p.19), devido a isso, o trabalho demonstrará uma breve contextualização e explicação de cada linguagem de programação citada pelos sites TecMundo e Exame para uma melhor compreensão do seu desenvolvimento.

## LINGUAGEM DE PROGRAMAÇÃO JAVA

Java é uma linguagem computacional completa, adequada para o desenvolvimento de aplicações baseadas na rede Internet, redes fechadas ou ainda programas *stand-alone* (CAMPIONE,1996).

Para uma maior compreensão, o exemplo abaixo é uma representação da estrutura de controle, a utilização da condição enquanto, mais conhecida como *While* (enquanto):

Figura 2: Linguagem JAVA

```
1  import javax.swing.JOptionPane;
2
3  float n1,n2,media;
4  int aluno=1;
5
6  while (aluno<=15) {
7      n1= Float.parseFloat(JOptionPane.showInputDialog("Informe a primeira nota: "));
8      n2= Float.parseFloat(JOptionPane.showInputDialog("Informe a segunda nota: "));
9      media=(n1+n2)/2;
10     if (media>=6) {
11         JOptionPane.showMessageDialog(null,"Aluno Aprovado com a média igual a = "+media);
12     } else {
13         JOptionPane.showMessageDialog(null,"Aluno reprovado com a média igual a = "+media);
14     }
15     aluno++;
16 }
```

Fonte: Créditos dos Autores

A resolução para um problema que envolve uma turma, notas e a média final dos alunos desta determinada turma. O algoritmo envolve os Tipos de variáveis *Float* (ponto flutuante) para as notas dos alunos, e *Integer* indicando cada aluno da classe. No caso, a classe possui 15 alunos e a média para que o estudante seja aprovado é igual ou maior que 6. A partir, de uma condição de controle *While*, o algoritmo executa o bloco especificado dentro dessa condição devolvendo ao usuário telas para que o mesmo, possa introduzir as notas dos respectivos alunos e também, mostrar a média de cada aluno informado.

O algoritmo importa a *javax.swing.JOptionPane* para mostrar ao usuário as telas indicadas no decorrer do código. Para que isso ocorra sem problemas, o código possui um *Casting* envolvendo as notas dos alunos, no código, as variáveis *n1* e *n2* que são declaradas do Tipo *Float*. O *casting* é quando a linguagem Java não suporta atribuições arbitrárias entre variáveis de tipos diferentes, ou seja, é preciso ordenar explicitamente que o número indicado seja moldado para um outro determinado tipo de variável. No caso, utilizamos o *parseFloat* para que o *JOptionPane* consiga entender as atribuições das variáveis. Na última linha do código, “aluno ++” adverte ao final do bloco de controle *While*, dando assim, um incremento para que não se torne um loop infinito e, conseqüentemente voltando ao comando de controle para uma nova validação.

Indo mais afundo no modelo de exibição *JOptionPane*, existem dois tipos utilizados neste algoritmo. O primeiro para inserção de valores e o outro de somente exibição das médias. Para compreensão, *ShowInputDialog* nada mais é uma caixa de diálogo onde o usuário insere os dados de forma simples e direta, no algoritmo exemplificado, o usuário insere os valores das notas respectivamente.

`ShowMessageDialog(null)` é um método utilizado para exibição, onde o primeiro parâmetro `null`, é usado pois o código acima não possui nenhum frame padrão.

A explicação de um algoritmo na linguagem Java exibe sua facilidade na compreensão e aprendizagem de cada código, ressaltando que é uma linguagem orientada a objeto, ou seja, caracterizado por um padrão sequencial de desenvolvimento.

## LINGUAGEM DE PROGRAMAÇÃO JAVASCRIPT

A denominação da linguagem, JavaScript, se deve a similaridades com a sintaxe do Java e embora as duas linguagens não tenham nenhuma outra relação além desta, os nomes ainda causam confusão para alguns usuários. JavaScript permite criar pequenos programas embutidos no próprio código de uma página HTML e capazes de gerar números, processar alguns dados, verificar formulários, alterar valor de elementos HTML e criar elementos HTML. Tudo isso diretamente no computador cliente, evitando a troca de informações com o servidor e o tempo passa a depender somente do processamento local do cliente, não mais da latência da rede (GRILLO & FORTES, 2008, p.4)

A linguagem JavaScript é um código utilizado dentro de uma página HTML, ou seja, dentro de uma página a utilização do Script para a abertura do código e a explicação da linguagem e tipo que estará utilizando para a criação do seu programa, a linguagem JavaScript vem logo após – com seus códigos e comandos –, assim, para finalizar é necessário fechar o Script que foi aberto no começo, assim como a representação abaixo:

Figura 3: Linguagem JavaScript Cabeçalho

```
<script language="javascript" type="text/javascript">  
  //JavaScript código  
</script>
```

Fonte: Créditos dos Autores

Os autores Grillo & Fortes (2008) em sua apostila sobre “*Aprendendo JavaScript*” disponibiliza vários exemplos de códigos e operadores para a representação da linguagem JavaScript, o exemplo abaixo é uma representação da estrutura de controle, a utilização das condições Se e Senão, mas na linguagem inglês *If e Else*:

Figura 4: Linguagem JavaScript

```
1  var a = 12;  
2  var b = 5;  
3  if (a == b) {  
4      window.alert("12 é igual a 5?!?!");  
5  }  
6  else {  
7      window.alert("a é diferente de b");  
8  }  
9
```

Fonte: Créditos dos Autores

O desenvolver criou duas variáveis A e B, assim cada variável tem seu designado valor como por exemplo: 12 e 5, respectivamente. O objetivo do código é verificar se o valor de A é igual ao de B utilizando as condições Se e Senão. A linha *If* ou Se representa a comparação de A com B utilizando os operadores relacionais (==) iguais, assim, o código terá dois caminhos para o seu desenvolvimento: **Se** o valor da variável A for igual ao B, o usuário receberá essa mensagem “12 é igual a 5?!?!”, **Senão** forem iguais aparecerá essa “a é diferente de b”. Cada chaves dessas “{” representa que a condição *If* e *Else* foram abertas para a descrição do algoritmo e depois fechada “}” e cada sinal de ponto-e-vírgula (;) indica o final de um comando.

## LINGUAGEM DE PROGRAMAÇÃO PYTHON

Aprender uma nova linguagem de programação é sempre algo desafiador e, aprender uma linguagem dinâmica representa uma mudança de paradigma ainda mais desafiador. Python é uma linguagem extremamente popular e, que vem crescendo muito ordinalmente. Com diversas funcionalidades, a linguagem Python pode ser usada para desenvolvimento para desktop, análise de dados, desenvolvimento web, inteligência artificial e muitas outras funções. Com uma especialidade muito peculiar, a linguagem Python possui tipagem dinâmica e trabalha com endentação do código, o que significa respectivamente que o tipo de uma variável é inferido pelo interpretador em tempo de execução e que o código será executado somente se ocorrer uma endentação correta.

No desenvolver da linguagem Python, existem várias distribuições e atualizações. O grupo optou por utilizar a versão 3.6 e para um maior entendimento da linguagem, a seguir encontra-se um exemplo de tipagem dinâmica junto a uma estrutura de controle *While* (enquanto).

Figura 5: Linguagem Python

```
1  variável = 10
2  while (variável > 0):
3      print (variável)
4      variável = variável - 1
5
6
```

Fonte: Créditos dos Autores

O trecho de código acima resulta em um contador que começa no número atribuído, `variável = 10`. Na segunda linha de código, o interpretador verifica o comando *While*, onde a partir da condição imposta (`variável > 0`): seja verdadeira, o interpretador entra no bloco abaixo que será executado.

A indentação na terceira linha, mostra que o comando `print(variável)` irá nos mostrar na tela os respectivos números de 10 a 1, ou seja, o bloco está dentro do comando *While* e assim, será executado até que a condição imposta na linha acima seja verdadeira.

Na última linha do código, acontece a subtração da variável declarada na primeira linha, onde fecha o bloco de controle e volta a condição do *While* novamente até que ele não seja mais verdadeiro.

## LINGUAGEM DE PROGRAMAÇÃO C++

O C++ é uma linguagem de programação que possibilita o desenvolvimento de aplicações em alto e baixo nível. Uma das grandes vantagens dessa linguagem é a transferência de códigos para a linguagem C, pois o C++ foi criado para ter compatibilidade com C. Outra vantagem desta linguagem é que ela não precisa de um ambiente muito potente para o seu desenvolvimento.

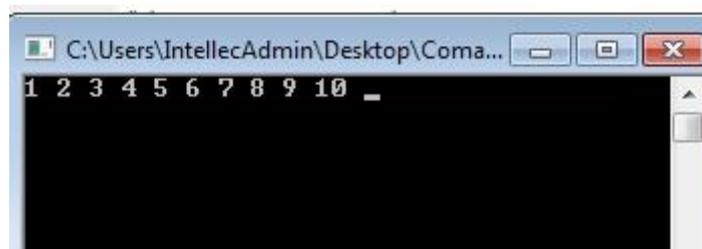
Abaixo segue uma estrutura de repetição muito utilizada na programação chamada “WHILE” onde o seu papel é mostrar para o usuário os n números de “1” a “10”.

Figura 6: Linguagem C++

```
1  #include <stdio.h>
2  #include <conio.h>
3  int main(void)
4  {
5      int contador = 1;
6      while (contador <= 10)
7      {
8          printf("%d ", contador);
9          contador++;
10     }
11     getch();
12     return 0;
13 }
14
```

Fonte: Créditos dos Autores

Figura 6: Tela de execução do programa mostrado a cima



Fonte: Créditos dos Autores

Inicialmente, a variável de controle denominada contador foi declarada e inicializada com o valor 1. O teste da condição *while* é realizado, e como o contador é menor que 10, então a condição é verdadeira. Como a condição seja verdadeira, o programa entra dentro do corpo do laço e executa o *printf*, exibindo o valor da variável contador. Após executar o comando *printf*, o contador é incrementado. Depois do incremento, a condição é testada novamente e enquanto for verdadeira, os comandos são executados de novo, até que a condição se torne falsa. A condição falsa faz com que o laço seja encerrado.

## LINGUAGEM DE PROGRAMAÇÃO C#

O C# é uma linguagem de programação orientada a objetos desenvolvida pela Microsoft que permite o desenvolvimento de diversas aplicações onde são executados no *.NET Framework* que é um ambiente de execução dos aplicativos desenvolvidos na linguagem C#. Sua sintaxe foi baseada em C++, Java e *Object Pascal*.

O C# é uma linguagem muito indicada para os iniciantes pois sua sintaxe é bem simples e fácil de aprender onde uma das ferramentas indicadas para o desenvolvimento é o Visual Studio.

Um exemplo de aplicação feita em C# é utilizando uma estrutura de repetição chamada “For” onde fazemos uma verificação e imprimimos somente os números que forem menores ou iguais a 5:

Figura 7: Linguagem C#

```
1  Class Exemplo
2  {
3      static void Main()
4      {
5          for(int i=1; i<=5; i++)
6          {
7              Console.WriteLine(i);
8          }
9      }
10 }
```

Fonte: Créditos dos Autores

A condição “For” acima serve para mostrar a repetição de um conjunto de números onde esta condição começa do valor “1” e vai até o número menor ou igual a “5”, fazendo com que mostre os números sempre somando “1” a mais em sua repetição. A linha “*Console.WriteLine(i);*” Serve apenas para mostrar na tela do usuário os números aplicados na repetição.

## LINGUAGEM DE PROGRAMAÇÃO PHP

A linguagem PHP é uma sigla recursiva que significa PHP HyperText Preprocessor. O PHP é uma linguagem de código-fonte aberto, muito utilizada na Internet e especialmente criada para o desenvolvimento de aplicativos Web (LIMA, 2003, p.3).

O professor Lima (2003) desenvolveu uma apostila sobre a linguagem PHP para seus alunos da disciplina Tópicos Especiais em Sistemas de Informação, mas também para usuários iniciantes e não-alunos da Universidade da Região da Campanha Centro de Ciências da Economia e Informática Curso de Informática.

Assim como outras linguagens o PHP é utilizada para o desenvolvimento Web, ou seja, uma linguagem dentro do HTML, para a identificação do PHP dentro do HTML

é através da sua construção de códigos, a imagem abaixo representa um algoritmo de estrutura de controle:

Figura 8: Linguagem PHP

```
1 <html>
2 <body>
3 <?php
4     $a=True;
5     If($a){
6         Echo "Verdadeiro";
7     }
8     Else {
9         Echo "Falso";
10    }
11 >?
12 </body>
13 </html>
14
```

Fonte: Créditos dos Autores

Observa-se que cada linha contém um sinal de abertura e fechamento, “a abertura é formada por um sinal de menor (<) e um final de interrogação seguida pela sigla PHP, a de fechamento é formada por um ponto de interrogação e sinal de maior (>)” (LIMA, 2003, p.6).

Os sinais de menor (<) e maior (>) para abrir a linguagem HTML, o corpo do algoritmo (*body*) e a linguagem PHP. Dentro da linguagem PHP existe a variável. A armazenando o valor Verdadeiro (*True*), “no PHP elas são representadas por um cifrão (\$) mais o nome da variável” (LIMA, 2003, p.8), as condições *If* e *Else* verificam se a variável “\$a” é verdadeira ou não, o comando *Echo* exibira a mensagem que está entre as aspas. Cada chave e ponto-e-vírgula é utilizado para fechar o comando, assim como, cada linguagem é fechada quando o código termina.

## LINGUAGEM DE PROGRAMAÇÃO PERL

Segundo o site CCM Brasil (2017), a linguagem Perl (*Practical Extraction and Report Language*) é derivada dos scripts *shell* criada em 1986 por Larry Wall para desenvolver um sistema de *News* entre duas redes. Sua maior vantagem é a adaptação à manipulação de cadeias de caracteres. Além disso, suas funcionalidades de manipulação

de arquivos, diretórios e bancos de dados transformaram a Perl na linguagem preferida para escrita de interfaces CGI.

O exemplo a seguir, mostra um algoritmo em Perl com duas variáveis que recebem o nome e a idade do usuário e que ao serem inseridas são exibidas na tela.

Figura 9: Linguagem PERL

```
1 $nome;  
2 $idade;  
3 $nome = "Carla";  
4 $idade = 20;  
5 print $nome;  
6 print $idade;  
7
```

Fonte: Créditos dos Autores

Para que a variável seja identificada na linguagem Perl, utiliza-se o símbolo “\$” antes do nome da variável. O símbolo de atribuição de valores é o símbolo = e, para que o usuário consiga visualizar as informações já inseridas, usa-se o comando print + \$ + nome da variável.

## CONSIDERAÇÕES FINAIS

O trabalho desenvolvido visou o estudo de algumas das linguagens de programação com suporte à mobilidade código, com o intuito de identificarem-se os tipos de abordagens utilizadas e trabalhar-se com uma linguagem de cada tipo de abordagem encontrada (DUARTE & DOTTI, 2001).

O objetivo de contextualizar cada linguagem de programação é para que cada usuário entenda suas diferenças e dificuldades, os autores Duarte & Dotti (2001) explicam que algumas linguagens enfocam mais certos aspectos do que as outras e, devido a isso, a escolha passa a não ser pela melhor linguagem, mas pela linguagem que forneça a maioria dos requisitos desejados para o tipo de aplicação que se pretenda desenvolver.

O projeto deve ser entendido como três fases. A fase 1 contempla em breves contextualizações de conceitos e aplicações; técnicas de aplicações e estruturas de montagem de cada linguagem auxiliando no entendimento da lógica de programação. A fase 2 a construção de sistemas em cada linguagem de programação citada, segundo os autores Fontes & Silva (2008), as linguagens de programação de computadores evoluíram muito nos últimos cinco anos. Atualmente elas oferecem uma vasta gama de recursos,

que permitem desde a elaboração de simples programas para cálculos matemáticos até a construção de sistemas de computação com engenharia complexa, a integração de dados alfanuméricos, geográficos, fotográficos ou sonoros, que podem ser lidos e armazenados em bases de dados localizadas em qualquer parte do planeta. Por fim, fase 3, o desenvolvimento de apostilas completas para o ensino-aprendizado de usuários iniciantes, mas também auxiliar usuários com questionamento sobre alguma linguagem específica.

## REFERÊNCIAS

AGRELA, Lucas. EXAME, 2017. **Estas são as linguagens de programação para ficar de olho em 2018**. Disponível em: <<https://exame.abril.com.br/tecnologia/estas-sao-as-linguagens-de-programacao-para-ficar-de-olho-em-2018/>> Acesso em: 12 março 2018.

CAMPIONE, M; WALRATH, K. **The Java Tutorial: Object-Oriented Programming for the Internet**. [S.l.]: SunSoft Press, 1996.

CCM BRASIL, 2017. **Linguagem Perl**. Disponível em: <<https://br.ccm.net/contents/845-linguagem-perl>> Acesso em: 23 maio 2018.

DUARTE, L.M.; DOTTI, F.L. **Estudo de Linguagens de Programação com Suporte à Mobilidade de Código**, 2001. FACULDADE DE INFORMÁTICA PUCRS – Brasil. Disponível em: <<http://www.pucrs.br/facin-prov/wp-content/uploads/sites/19/2016/03/tr015.pdf>> Acesso em: 23 maio 2018.

FERRARI, F.; CECHINEL, C. **Introdução a Algoritmos e Programação**. Bagé: Universidade Federal do Pampa, Campus Bagé, 2008.

FONTES, C.R.; SILVA, F.W.O. **O ensino da disciplina linguagem de programação em escolas técnicas**, 2008. Disponível em: <[http://www.cienciasecognicao.org/pdf/v13\\_2/m318242.pdf](http://www.cienciasecognicao.org/pdf/v13_2/m318242.pdf)> Acesso em: 23 maio 2018.

GRILLO, F.D.N.; FORTES, R.P.M. **Aprendendo JavaScript**. 2008. Disponível em: <[http://conteudo.icmc.usp.br/CMS/Arquivos/arquivos\\_enviados/BIBLIOTECA\\_113\\_ND\\_72.pdf](http://conteudo.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_ND_72.pdf)> Acesso em 22 maio 2018.

LIMA, C.C. **Apostila de PHP**. Universidade da Região da Campanha Centro de Ciências da Economia e Informática Curso de Informática – Disciplina: Tópicos Especiais em Sistemas de Informação, 2003. Disponível em: <<http://www.alkem.com.br/download/apostilaphp.pdf>> Acesso em: 22 maio 2018.

LAUER, D., SOBRINHO, W. P. **Comparação entre linguagens de programação**. Curitiba, 2008. Disponível em: <<https://pt.scribd.com/document/31787143/Artigo-Final-Comparacao-Entre-Linguagens-de-Programacao>> Acesso em: 17 maio 2018.

MEDINA, Marco, FERTIG, Cristina. **Algoritmo e Programação: Teoria e Prática**. 1. ed. São Paulo: Novatec, 2005.

MORAES, P. S. **Lógica de Programação**. Unicamp - Centro de Computação, 2000, p.4. Disponível em: <<http://www3.dsi.uminho.pt/iiee/repos/logica.pdf>> Acesso em 8 maio 2018.

MORIMOTO, Carlos E. 2007. **Linguagens de Programação**. Disponível em: <<https://www.hardware.com.br/artigos/linguagens/>> Acesso em: 17 maio 2018.

PAYÃO, Felipe. TECMUNDO, 2017. **As 5 principais linguagens de programação de 2018**. Disponível em: <<https://www.tecmundo.com.br/software/125237-5-principais-linguagens-programacao-2018.htm>> Acesso em: 12 março 2018.

PINOCHET, Luis H. C. **Tecnologia da Informação e Comunicação**. Rio de Janeiro: Elsevier, Ed. 1, 2014, p. 40-215.

SOUZA, L. M. TECMUNDO, 2017. **Conheça as linguagens de programação que mais crescem no mercado de TI**. Disponível em: <<https://www.tecmundo.com.br/software/125425-conheca-linguagens-programacao-crescem-mercado-ti.htm>> Acesso em: 12 março 2018.

TUCKER, A.B. NOONAN, R.E. **Linguagens de programação: princípios e paradigmas**. Ed. 2, Porto Alegre: AMGH, 2010.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE (UFRN), Centro de Tecnologia Departamento de Computação e Automação (2004). **Algoritmo e Lógica de Programação, Algoritmos – Parte 1**. Disponível em: <[https://www.dca.ufrn.br/~affonso/DCA800/pdf/algoritmos\\_parte1.pdf](https://www.dca.ufrn.br/~affonso/DCA800/pdf/algoritmos_parte1.pdf)> Acesso em: 12 março 2018.

VAREJÃO, F. **Linguagens de Programação: Conceitos e Técnicas**. 2. ed., Rio de Janeiro: Campus, 1995.