

Linguagens de Programação Importantes do Passado e do Presente e Paradigmas de Programação

Eduardo Henrique de A. Izidorio, Gabriel Peixoto M. da Costa, Marcia Gabrielle B. de Oliveira

Departamento De Ciência Da Computação – Universidade Federal De Roraima
(UFRR)

`eduardo57izidorio@gmail.com, gabrielpeixoto371@gmail.com,
gabybonifacio2@gmail.com`

Histórico de Linguagens de Programação

Entre os anos de 1943 à 1946, o cientista alemão Konrad Zuse desenvolveu a primeira linguagem de programação, chamada de Plankalkül, porém, sua primeira tentativa de publicar, em 1948, não adquiriu relevância profissional para a época, então acabou em desuso. Posteriormente, na década de 50 que surgiram as primeiras linguagens de programação modernas, como exemplo o Cobol e o Fortran, em 1955, e futuramente o Lisp e o Algol 60, em 1958.

Durante a década de 60, a Linguagem de Programação Simula foi criada, sendo ela a primeira linguagem a comportar o conceito de classes. Consequentemente, os paradigmas da programação foram criados em sua maior parte na década de 70, tal como:

- **C:** Uma das primeiras linguagens de programação, sendo uma das favoritas e tendo uma das maiores influências do mundo atual, além disso, ela é uma linguagem imperativa de uso geral.
- **Prolog:** Foi a primeira linguagem de programação com paradigma lógico, desenvolvida em 1972, utilizada até os dias atuais.
- **Pascal:** É uma linguagem de programação estruturada, sendo comum usa-la para fins acadêmicos, porém sendo deixada de lado para usar linguagens de fácil Legibilidade.

Os anos 80 foram marcados pelo investimento nas linguagens de programação desenvolvidas nas décadas de 60 e 70, como:

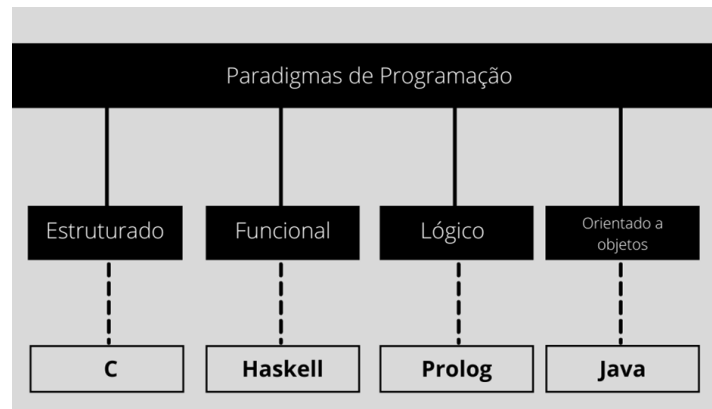
- **C++:** É uma evolução melhorada da linguagem C, considerada uma linguagem de nível intermediário, orientada a objetos.
- **Perl:** Trata de uma linguagem interpretada cuja maior vantagem é sua adaptação à manipulação de cadeias de caracteres.

A década de 90 foi um marco para a linguagem da programação moderna, pois originaria das linguagens antigas. As linguagens modernas vieram com um foco inovador para a rede mundial de computadores e para a orientação a objeto, tais como:

- **Java:** Considerada e classificada uma linguagem de programação orientada a objeto. Que reuniu características de várias outras de linguagens nela, afim de produzir uma ferramenta que "rodasse" em qualquer ambiente.
- **Python:** É uma linguagem de programação de alto nível, dinâmica, interpretada, modular, multiplataforma e orientada a objetos. Por ser uma linguagem de sintaxe relativamente simples e de fácil compreensão, ganhou popularidade entre profissionais da indústria tecnológica.

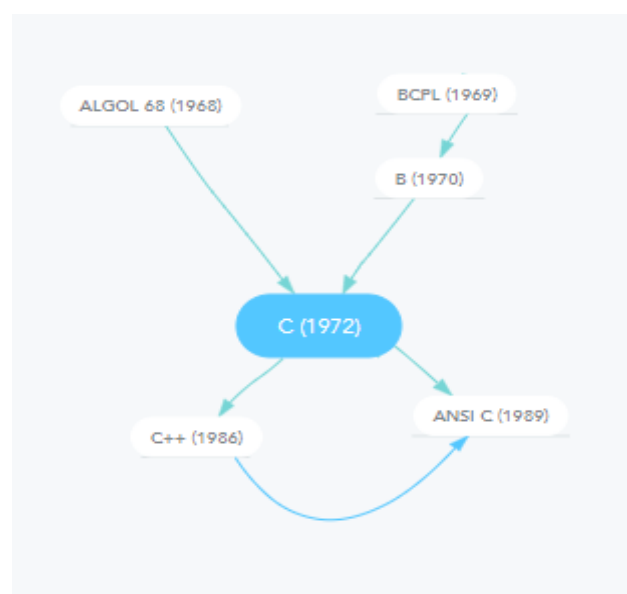
Paradigmas de Programação

Para cada um dos paradigmas (Estruturado, Funcional, Lógico e Orientado a objetos), escolher pelo menos uma linguagem de programação, e escrever, com suas próprias palavras, um breve resumo das características diferenciais, assim como sua relação histórica com outras linguagens que a precederam ou sucederam.



Linguagem C

A linguagem em C foi criada em 1972, desenvolvida nos laboratórios Bell pelo cientista da computação Dennis Ritchie, e derivada por outras duas linguagens: Algol 68 e BCPL. Inicialmente o foco da Linguagem C era o desenvolvimento de sistemas operacionais e compiladores. Teve um grande sucesso na construção de uma versão nova do UNIX (Sistema operativo portátil, multitarefa e multiutilizador), que foi inicialmente escrita na linguagem Assembly. Com esse grande sucesso obtido no mundo de Unix, fez que grande maioria dos grandes sistemas operacionais atualmente são construídos com as linguagens C/C++.



Origem e evolução da Linguagem C

O ano de 1978 foi marcado pela publicação do livro: The C Programming Language e na época fez bastante sucesso e auxiliou muito a divulgação da Linguagem C. Início dos anos 80 a linguagem C houve o reconhecimento como uma linguagem de propósito geral e com isso contava com uma variedade de compiladores construídos por diversos fabricantes, e durante essa época iniciou-se uma padronização, pois existia vários compiladores e frequentemente apresentavam discrepâncias, grande parte dos fabricantes como opção de compatibilidade usava o padrão ANSI C.

Aspectos principais

C é considerada uma linguagem de alto nível

Dizer que a linguagem é de alto nível é falar que a sintaxe da linguagem se aproxima mais da nossa linguagem e se distancia da linguagem da máquina, mas mesmo sendo de alto nível a linguagem C tem alguns aspectos de baixo nível, por exemplo a incorporação de código Assembly.

Portabilidade

A linguagem C tem o aspecto de ser muito portátil entre vários sistemas, inicialmente construída para UNIX, ela roda em diversas outras arquiteturas, sistemas operacionais e compiladores diferentes.

Simplicidade

C tem uma sintaxe muito simples para o aprendizado e entendimento, improvavelmente ela trará problema para quem a usa.

Modularidade

Um programa em C tem divisão em vários blocos de programação diferentes, isto é, funções que não estão ligadas entre si. Deste modo, por ora, em que a sua função é fechada, aquilo que você escreveu não vai prejudicar os blocos seguintes – a não ser que seja chamada.

```

1  ▾ //Linguagem da programação 2021
2      #include<stdio.h>
3
4      int main()
5
6      {
7          printf("Hello world!");
8          return 0;
9      }
10

```

Exemplo de um “Hello World” em Linguagem C

Considerações finais

Atualmente a linguagem C tem aspectos comuns quando comparado as linguagens de hoje em dia, mas na época que foi criada, seus conceitos foram inovadores entre as linguagens que já existia naquela década (ALGOL 68 e a B).

Haskell

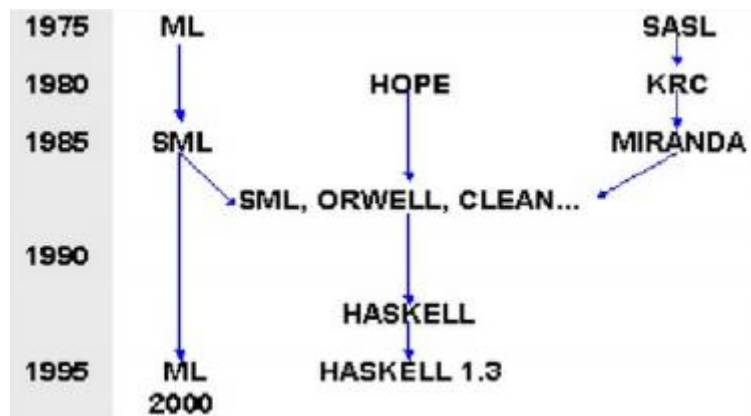
Contexto Histórico

Anos antes da criação da linguagem Haskell em 1980, foi o ano de ascensão da programação funcional, durante esse período surgiu várias linguagens funcionais, porém só a linguagem Miranda era madura, comercial e tinha um bom projeto. No ano de 1987, ocorreu uma reunião, na conferência de Linguagens de Programação Funcionais e Arquitetura de Computadores, para debater sobre o número grande de linguagens que estavam sendo criadas na época, então decidido que para criar a linguagem deveria ter uma comissão.

Com uma comissão formada, eles tinham em mente algumas diretrizes para a criação da nova linguagem, são elas:

- Uma comunicação bem mais rápida de novas ideias;
- Ter um fundamento estável para a criação de aplicações reais;
- Ser um meio, para que a Programação Funcional seja estimulada.

Essa comissão decidiu que partiria da linguagem Miranda para o desenvolvimento da nova linguagem, pois Miranda na época era a mais bem projetada entre as outras. Sendo assim no ano de 1990, foi liberada a primeira versão, que veio a ser desenvolvida pelos cientistas Simon Peyton-Jones, Paul Hudak e Philip Walder e em homenagem ao matemático Haskell Brooks Curry a linguagem foi nomeada com o seu nome. E em 1999 foi anunciado o Haskell 98, que era uma versão estável e portável da linguagem Haskell.



Influencias para a linguagem Haskell

Aspectos Principais

Paradigma Funcional

Programas escritos em linguagens funcionais tendem a ser mais concisos do que seus equivalentes em linguagens imperativas. Assim, a redução em linhas de código pode variar numa razão de 2 a 10.

Forte Sistema de Tipo

Haskell é fortemente tipada, eliminando a grande parte de erros comuns durante a compilação. De uma forma geral, um sistema de tipo permite que o programador indique à priori como devem ser tratados os dados, associando-os a um tipo. Com um forte sistema de tipos não existe, por exemplo, a possibilidade de tratar um inteiro como um apontador ou seguir um apontador nulo.

Funções Recursivas

Com a imutabilidade, o conceito de laços de repetição também não existe em linguagens funcionais. Isso se dá porque eles são implementados através de funções recursivas.

```
int x = 1;

for (int i = 1; i <= 10; i++) {

    x = x * 2;

}

printf("%d\n", x);
```

Avaliação preguiçosa

Ao aplicar uma função, o resultado será computado apenas quando requisitado. Isso permite evitar computações desnecessárias, estimula uma programação modular e permite estruturas de dados infinitas.

```
listaInf = [1..] -- 1, 2, 3, ...  
  
print (take 10 listaInf)
```

Considerações Finais

Haskell é a sobre a qual mais se realizam pesquisas atualmente. Muito utilizada no meio acadêmico, é uma linguagem relativamente nova, derivada de outras linguagens funcionais, como por exemplo Miranda e ML. Existe uma série de linguagens inspiradas pelo Haskell, mas com sistemas de diferentes tipos, foram desenvolvidas, incluindo: Agda, Idris, Epigram, entre outros.

Prolog

Contexto Histórico

Prolog foi criado por Alain Colmerauer em 1973, na Universidade de Marseille, França. Seu nome deriva da expressão “Programming in Logic”, que é baseado na lógica de predicados. Desde então prolog é utilizada para aplicações de computação simbólica, compreensão de linguagens naturais, análise de estruturas bioquímicas, sistemas especialistas, entre outros. Sua estrutura conceitual está totalmente ligada à lógica matemática, o que faz dela uma ferramenta ótima para estudos de lógica.



Alain Colmerauer – um dos principais criadores da linguagem Prolog

Prolog é uma linguagem declarativa, ou seja, que se limita a fornecer uma descrição do problema onde utilizando de fatos e regras (lógica) exprimem o problema a ser resolvido. Por ser uma linguagem declarativa, podemos citar fatos de que se difere das outras linguagens, prolog não possui estruturas de controle, assim como também outros tipos de dados que existem na maioria das outras linguagens, não são empregados nele.

Aspectos Principais

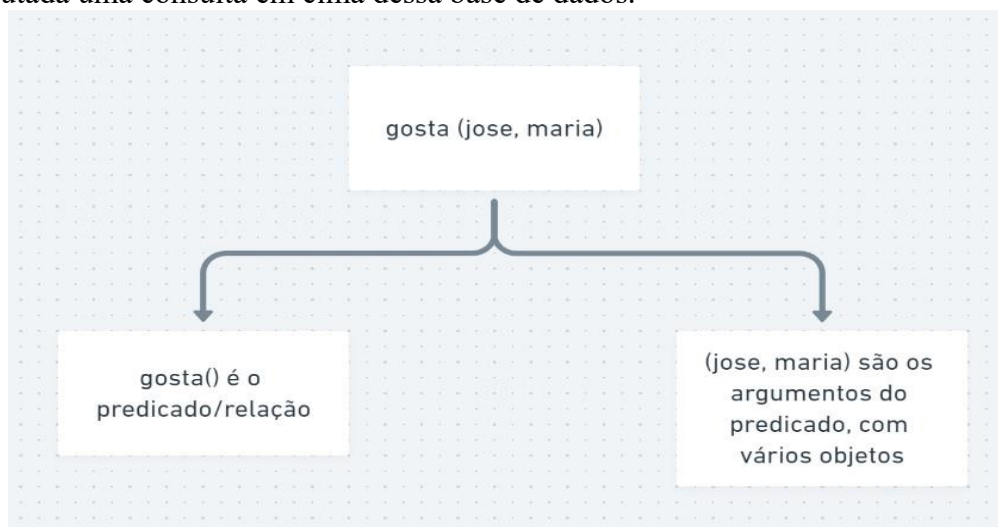
Paradigma lógico

Emergindo na década de 1970, o paradigma lógico teve uma abordagem diferente dos outros paradigmas expressando programas na forma lógica simbólica e usando um processo de inferência lógica assim produzindo resultados. Sua principal aplicação foi nas áreas de inteligência artificial.

Uma linguagem lógica é aquela que faz uso da lógica matemática, que envolve conceitos como a declaração de fatos e regras, e usamos essas regras e fatos para chegar a conclusões.

Fatos em Prolog

Programar em Prolog é bem diferente de como programamos nas outras linguagens. No Prolog é dado fatos e regras para uma base de dados, consequentemente depois será executada uma consulta em cima dessa base de dados.



Estrutura e Características de um fato em Prolog:

- Estrutura formada por um predicado, objeto e um ponto(.) para finalizar a instrução.
Por exemplo: predicado (argumento1, argumento2...).
- Os predicados são escritos primeiros e depois os objetos, sendo eles sempre escritos com letras minúsculas.
- E os objetos devem ser sempre escritos entre parênteses separados por vírgulas.

Questões em Prolog

Uma questão varia de acordo como os compiladores existentes, mas é praticamente escrita da mesma forma de um fato, diferenciando de que é antecedido por um ponto de

interrogação.

Quando é feita uma questão, o Prolog realiza uma busca na Base de Conhecimento, procurando um fato que seja igual ao da questão, assim podemos dizer que dois fatos ou um fato e uma questão são iguais se:

- Seus predicados são os mesmos;
- Eles possuem o mesmo número de argumentos e se os argumentos são iguais;

Quando o Prolog encontra um fato que seja igual a questão, ele retorna “YES”, indicando assim resposta verdadeira para a questão. Caso contrário, se ele não conseguir se igualar a questão a um fato da Base de Conhecimento, ele retornará “NO”.

Unificação em Prolog

A Unificação é o elemento básico do interpretador ou compilador Prolog. O nome Unificação é dado ao processo de que tenta casar um objetivo (goal) com uma cláusula, que inclui a busca, o casamento de padrões (pattern-matting) e as instanciações (Yin, 1987).

Basicamente, dizemos que a unificação teve resultado positivo se os objetos envolvidos na comparação são idênticos ou se as cláusulas possuem variáveis que podem ser instanciadas. Caso contrário, dizemos que a unificação falhou.

Variáveis em Prolog

Uma variável pode ser instanciada ou não-instanciada. Vale lembrar que uma variável em Prolog sempre deve começar com uma letra maiúscula. Dizemos que a variável está instanciada quando ela estiver assumindo o valor de um objeto, caso contrário, não-instanciada.

Um exemplo para que entenda melhor:

Considere a seguinte Base de Conhecimento:

- gosta (joão, flores).
- gosta (joão, maria).
- gosta (Paulo, maria).

Ao ser realizada a questão?

? – gosta (joão, X).

A variável X inicialmente está não-instanciada, logo o Prolog procura na Base de conhecimento um fato que se iguale a questão, isto é, tendo um mesmo predicado, mesmo número de argumentos e que o primeiro argumento seja “joão”. É importante ressaltar que a procura tem que seguir a ordem em que os fatos foram passados dentro da Base de Conhecimento. Assim, depois de verificar se todos os requisitos foram atendidos, é instanciado a variável X com o objeto “flores”.

Considerações finais

Como podemos ver, o Prolog é uma linguagem muito poderosa, apesar de termos

mostrado os principais comandos e apenas o básico dessa linguagem, ainda existem muitas outras propriedades e implementações diferentes, cada um com seu próprio padrão, que não foram abordadas no artigo. Podemos citar algumas dessas implementações como, Visual Prolog (Turbo Prolog), GNU Prolog, Amzi! Prolog, entre muitas outras já existentes. Como já falado anteriormente o Prolog é utilizado principalmente na área de inteligência artificial onde é líder absoluto até os tempos atuais.

Java

Introdução e Contexto Histórico

Java é uma linguagem de programação orientada a objetos, criada na década de 90 por James Gosling e sua equipe da Sun Microsystems. O nome Java foi inspirado no café que o time de desenvolvimento consumia, oriundo da ilha de Java.

A tecnologia Java tinha sido projetada para se mover por meio das redes de dispositivos heterogêneos, redes como a internet. Agora aplicações poderiam ser executadas dentro dos navegadores nos Applets Java e tudo seria disponibilizado pela internet instantaneamente. Foi o estático HTML dos navegadores que promoveu a rápida disseminação da dinâmica tecnologia Java. A velocidade dos acontecimentos seguintes foi assustadora, o número de usuários aumentou rapidamente, grandes fornecedores de tecnologia, como a IBM anunciaram suporte para a tecnologia Java.



Logomarca do Java inspirado no café

A linguagem Java é compilada para um bytecode que é interpretado por uma máquina virtual, Java Virtual Machine (JVM), que é diferente das linguagens de programação convencionais, que são compiladas para código nativo. Será abordado as principais características e também as linguagens que influenciaram na criação do Java.

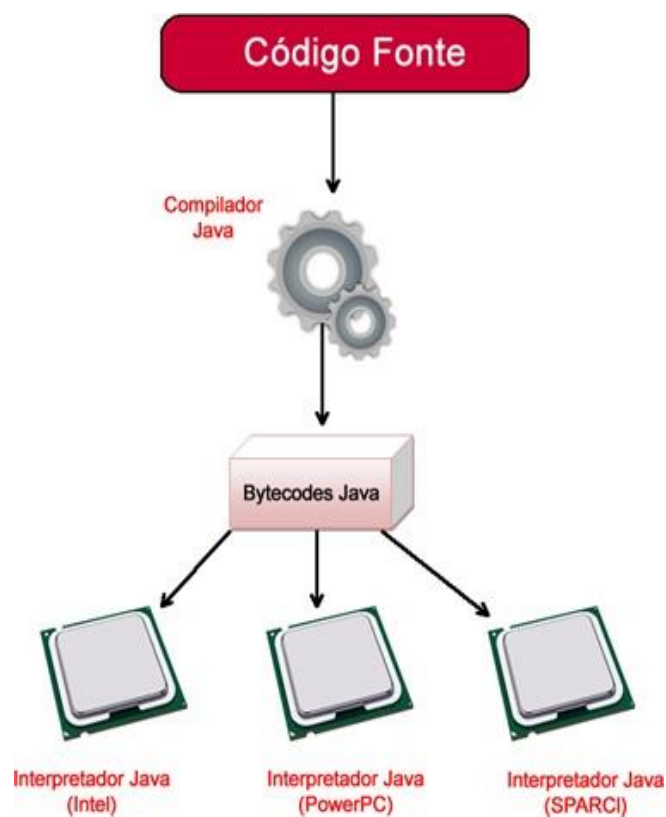
Aspectos Principais

Orientado a objetos

O modelo orientado a objetos (POO) é um dos estilos de programação mais populares. Ele permite desenhar o software para que os diferentes tipos de dados utilizados sejam vinculados às suas operações.

Independência de Plataforma

Atualmente, grande parte das linguagens sofrem na transferência de plataforma quando o sistema desenvolvido tem que mudar para outra plataforma, pois quando compilado um programa a ação do compilador é transformar o arquivo-fonte em código de máquina. A independência de plataforma já fala por si, possibilita o programa ser executado em diferentes plataformas e sistemas operacionais, através de um emulador conhecido como a **Máquina Virtual Java** ou **JVM** (Java Virtual Machine) que ajuda rodar os sistemas baseados em Java. Pode-se também se denominar como uma máquina virtual baseada em software que é executada dentro dos aparelhos eletrônicos onde irá ler e executar os bytecodes do Java.



Processo da compilação de um programa desenvolvido por Java.

Tipada

A linguagem Java é fortemente tipada, utilizando a tipagem estática que não permite ao desenvolvedor alterar o tipo da variável depois de declarada. Geralmente a verificação de tipo é feita em tempo de compilação. Podemos ver o exemplo abaixo na linguagem Java:

```
public class MyClass {  
    public static void main(String args[]) {  
        int variavel = 10;  
  
        variavel = "Elton Fonseca"; //error: incompatible types: String cannot  
        be converted to int  
    }  
}
```

Relação histórica com outras linguagens

Com a intenção de criar um projeto simples e ao mesmo tempo robusto, dinâmico e com alto desempenho, Gosling recebeu influências do C e do C++, buscou-se o aperfeiçoamento dos pontos que continham desvantagens em suas precursoras, criando assim uma linguagem mais simples e voltada para atender as necessidades do desenvolvimento de aplicações em ambientes distribuídos e heterogêneos. E o Java influenciou algumas linguagens como Ada 2005, BeanShel e Fantom.

Considerações Finais

Atualmente, o Java é uma das linguagens mais usadas no mundo, isso se dá pelo fato de não ser apenas uma linguagem, mas também uma plataforma de desenvolvimento. E como a maior parte dos aplicativos desenvolvidos para dispositivos móveis é desenvolvido em Java, isso faz com que os programadores que dominam essa linguagem sejam profissionais bastante procurados e valorizados no mercado de trabalho.

Referências

Histórico da linguagem de programação, Disponível em:

<https://www.infoescola.com/informatica/historia-da-programacao/> ,

<http://www.facom.ufu.br/~madriana/PF/Intro.pdf> e

<https://www.programador.com.br/historia-da-programacao.html> > Acesso em: 27 de julho de 2021.

Contexto Histórico linguagem C, Disponível em: <http://linguagemc.com.br/breve-historia-da-linguagem-c/> e <https://www.passeidireto.com/arquivo/1504406/resumo-linguagem-c> > Acesso em: 28 de julho de 2021.

Características de linguagem C, Disponível em: <https://canaltech.com.br/software/c-a-linguagem-de-programacao-que-esta-em-tudo-o-que-voce-usa-19512/> , <https://idocode.com.br/blog/programacao/exemplos-e-aplicacoes-da-linguagem-c/> e <https://blog.betrybe.com/linguagem-de-programacao/linguagem-alto-e-baixo-nivel/> > Acesso em: 28 de julho de 2021.

Contexto Histórico de haskell. Disponível em:

<https://marceloparrela.wordpress.com/2013/11/01/haskell-a-historia/> ,
<http://www.inf.ufes.br/~vitorsouza/archive/2020/wp-content/uploads/teaching-lp-20132-seminario-haskell.pdf> , <https://pt.slideshare.net/renzopetri/seminario-haskell> e
<http://www.inf.ufes.br/~vitorsouza/archive/2020/wp-content/uploads/teaching-lp-20192-seminario-haskell.pdf> > Acesso em: 29 de julho de 2021.

Aspectos Principais de Haskell, Disponível em: <http://ninjadolinux.com.br/haskell-a-linguagem-puramente-funcional/> ,
[https://pt.wikipedia.org/wiki/Haskell_\(linguagem_de_programa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Haskell_(linguagem_de_programa%C3%A7%C3%A3o)) e
<https://haskell.pesquisa.ufabc.edu.br/posts/haskell/03.haskell.basico.1.html> > Acesso em: 29 de julho de 2021.

Contexto Histórico de Prolog, Disponível em:
<http://www.linhadecodigo.com.br/artigo/1697/descobrindo-o-prolog.aspx> e
<https://pt.wikipedia.org/wiki/Prolog> > Acesso em: 28 de julho de 2021.

Paradigma lógico, Disponível em: https://leandromoh.gitbooks.io/tcc-paradigmas-de-programacao/content/6_paradigma_logico/index.html > Acesso em: 29 de julho de 2021.

Aspectos Principais de Prolog, Disponível em:
<http://www.linhadecodigo.com.br/artigo/1697/descobrindo-o-prolog.aspx> > Acesso em: 29 de julho de 2021.

Contexto Histórico de Java, Disponível em:
[https://pt.wikipedia.org/wiki/Java_\(linguagem_de_programa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Java_(linguagem_de_programa%C3%A7%C3%A3o)) >
Acesso em: 28 de julho de 2021.

Aspectos Principais de Java, Disponível em: <https://www.java.com/pt-BR/about/> ,
<https://www.zup.com.br/blog/java> > Acesso em: 28 de julho de 2021.