

**PROGRAMAÇÃO PARA WEB**

**ATIVIDADE 2 – PESQUISA**

**LINGUAGENS PARA BACK-END**

Beatriz Alves Ribeiro – 0030482111036

Sorocaba

Agosto, 2022

Sumário

[Introdução 3](#_Toc112064430)

[O que é Back-End? 4](#_Toc112064431)

[O que é uma linguagem de programação? 5](#_Toc112064432)

[Linguagem de programação de alto nível 5](#_Toc112064433)

[Linguagem de programação de nível intermediário 5](#_Toc112064434)

[Linguagem de programação baixo nível 5](#_Toc112064435)

[C# .NET 6](#_Toc112064436)

[Características 6](#_Toc112064437)

[Limitações 7](#_Toc112064438)

[Java 8](#_Toc112064439)

[Características 8](#_Toc112064440)

[Limitações 9](#_Toc112064441)

[JavaScript 10](#_Toc112064442)

[Características 10](#_Toc112064443)

[Limitações 11](#_Toc112064444)

[PHP 12](#_Toc112064445)

[Características 12](#_Toc112064446)

[Limitações 13](#_Toc112064447)

[Phyton 14](#_Toc112064448)

[Características 14](#_Toc112064449)

[Limitações 15](#_Toc112064450)

[Ruby 16](#_Toc112064451)

[Características 16](#_Toc112064452)

[Limitações 17](#_Toc112064453)

[Conclusão 18](#_Toc112064454)

[Referências 19](#_Toc112064455)

# Introdução

O presente trabalho tem como objetivo apresentar o que é back- end, as principais linguagens mais utilizadas atualmente e níveis de linguagem de programação.

Linguagens pelas quais é necessário conhecer e saber suas características e suas limitações, também saber quais estão sendo mais utilizadas, assim tendo uma facilidade na hora de escolher qual linguagem aprender.

# O que é Back-End?

Back-end é basicamente o que fica por trás de uma aplicação, a estrutura com suas funcionalides.

Segundo o site Kenzie, “No universo da programação é basicamente tudo que fica por trás de uma aplicação web, ou seja, são códigos que fazem a ponte entre o navegador rumo ao banco de dados e vice-versa. Quando você usa um aplicativo bancário existem ícones e botões para que você decida o que fazer, correto? E quando você define uma ação, como realizar um pix, por exemplo, fica a cargo do back-end validar informações, verificar se a conta está ativa, se existe saldo na conta e se a senha foi informada corretamente. Aí sim, decidir se é possível realizar o pix ou não. Logo, tudo que é back-end acontece “atrás dos bastidores”. ”

# O que é uma linguagem de programação?

A linguagem de programação é um conjunto de comandos e instruções digitais que usam sintaxes específicas para criar aplicativos de computador. Elas são classificadas por três tipos, que são respectivamente:

## Linguagem de programação de alto nível

Essa linguagem utiliza palavras e símbolos que utilizamos no nosso dia a dia, ou seja, mais perto do ser humano.

## Linguagem de programação de nível intermediário

Também existe a linguagem de nível intermediário, porém ela é basicamente a linguagem de alto nível só que com direcionamentos que facilitam a vida do programador, ademais, ela possui características que podem ser tanto de baixo nível quanto de alto nível.

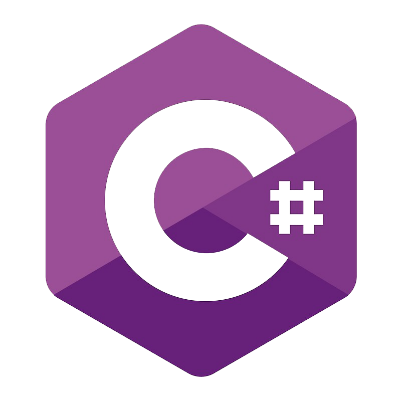
## Linguagem de programação baixo nível

Esse tipo de linguagem utiliza linguagens de máquina, ou seja, as linguagens de baixo nível enviam instruções diretas ao processador e são constituídas de sequências de 0 e 1, conhecido como código binário.

# C# .NET

Uma das linguagens de programação mais populares para o desenvolvimento de back-end é C #, comumente chamada de C-Sharp. Ele faz um excelente trabalho na automação de codificação em servidores Windows e plataformas web (usando o framework ASP.Net.) C-Sharp, que é mais ou menos uma extensão do C ++, já existe há muito tempo.

C # encontra ampla aplicação no desenvolvimento de aplicativos de desktop e também em sistemas incorporados. Ele executa o código mais rápido do que a maioria das linguagens de programação. C # foi usado para criar plataformas semelhantes ao Unity para desenvolvimento de jogos e para criar aplicativos CLI.



## Características

* **Desenvolvimento de plataforma cruzada**: os aplicativos com back-ends C-Sharp podem ser executados em vários sistemas operacionais, como macOS, Windows, Linux, etc. Este artigo sobre estruturas de plataforma cruzada discute o desenvolvimento de plataforma cruzada mais adiante.
* **Vantagem orientada a objetos**: Como uma linguagem orientada a objetos, o código C # faz uso de classes e relacionamentos. Essa abordagem permite a reutilização de trechos de código e fácil solução de problemas de código.
* **Compatibilidade extensiva**: aplicativos C # apresentam compatibilidade com sistemas legados. Por exemplo, organizações que ainda usam versões mais antigas de estruturas de programação considerariam C # inestimável.
* **Recurso de coleta de lixo**: C-Sharp tem um recurso que elimina todos os resíduos do sistema. Este é um ótimo recurso que acelera a execução do programa. A melhor parte é que o sistema continua a funcionar perfeitamente durante o procedimento de coleta de lixo.

## Limitações

* Como uma linguagem de programação de alto nível, o código C # não pode se comunicar diretamente com o hardware.
* C # é altamente rígido em comparação com outras tecnologias de back-end porque ele só funciona no .Net Framework e só pode ser instalado e executado em um computador Windows.

# Java

O índice 2021 TIOBE classifica Java em segundo lugar entre as tecnologias de desenvolvimento de backend hoje. Essa classificação indica que Java é uma das linguagens de programação mais influentes hoje. James Gosling inventou a linguagem de programação em 1991, mas não foi até 1995 que a Sun Microsystems a publicou.

Ao longo dos anos, o Java surgiu como a plataforma de escolha para desenvolvedores que preferem criar aplicativos da web com recursos personalizados e inovadores. Java também é útil para desenvolvimento móvel e aplicativos em dispositivos incorporados.



## Características

* **Escalável e direto**: Java Enterprise Edition é útil para criar aplicativos escalonáveis, permitindo que o servidor execute várias instâncias. Os componentes Java estão prontamente disponíveis e a sintaxe é fácil de entender. Todos esses recursos o tornam excelente como uma tecnologia de desenvolvimento de backend.
* **Multi-Threading**: Java é capaz de lidar com solicitações em threads independentes em um servidor web multi-thread. É por isso que o Java tem um desempenho excelente com aplicativos que exigem grande capacidade de CPU.
* **Bibliotecas de código aberto massivas**: os desenvolvedores podem aproveitar bibliotecas de código aberto massivas para agilizar as tarefas de desenvolvimento do lado do servidor. Algumas dessas bibliotecas incluem testes de unidade, excel, mensagens, análise JSON e muito mais.
* **Segurança aprimorada**: Java é conhecido por sua segurança rígida; ele fornece vários recursos que reduzem os riscos de segurança. Por exemplo, a Java Virtual Machine verifica os bytecodes java para manter os vírus afastados. Outros recursos que aumentam a segurança são o modelo de segurança Java e o teste de código reutilizável.

## Limitações

* A programação Java é cara e demorada.
* A programação de baixo nível e o comando para coleta de lixo estão visivelmente ausentes.
* O alto custo dos requisitos de hardware torna cara a implementação de back-ends baseados em java.
* O kit de ferramentas Swing, o aplicativo de desenvolvimento Java GUI, carece da interface intrigante de outras ferramentas GUI da moda.

# JavaScript

Por quase uma década, o JavaScript emergiu como a linguagem de programação mais popular na Pesquisa Anual StackOverflow. Na recente Pesquisa para Desenvolvedores de 2020, 69,7% dos entrevistados escolheram JavaScript como a linguagem de programação mais amplamente usada.

JavaScript não é mais apenas uma linguagem de script do lado do cliente, porque tecnologias como Node.Js permitem que você execute operações do lado do servidor. O Node.js depende das estruturas do lado do servidor chamadas Express.js para criar uma plataforma que permite aos desenvolvedores escrever códigos que são executados no servidor.

Assim que o Express.Js estiver instalado e funcionando com o Node.Js, os desenvolvedores podem usar JavaScript como linguagem de desenvolvimento de front-end e back-end. Ele também oferece uma interface de programação de aplicativos (API) para a criação de vários aplicativos, incluindo aplicativos móveis, híbridos, da web, de uma e várias páginas.



## Características

* **Desenvolvimento rápido:** como a mesma linguagem é usada para o desenvolvimento de front-end e back-end, os programadores podem criar aplicativos da web e móveis mais rapidamente com JavaScript.
* **Menos sobrecarga de script:**Alguns recursos integrados do JavaScript, como DOM e hoops, aumentam a eficiência da codificação e melhoram o desempenho e menos sobrecarga de script.
* **Tecnologia Lean Backend:** JavaScript usa plataformas de backend como Express.js, que serve como um middleware para resolver vários desafios de desenvolvimento. Eles fornecem várias funções, como dados POST, cabeçalhos de sutura, tratamento de erros, registrador de solicitações HTTP e muito mais. Além disso, as regras de programação do JavaScript são menos rígidas, dando aos desenvolvedores mais liberdade para trabalhar com seu middleware preferido.
* **Corte de custos:**a eficiência do JavaScript reduz os esforços de programação e os custos necessários para desenvolver aplicativos. Uma vez que a mesma linguagem de programação é usada para criar códigos de front-end e back-end.
* **Vantagem de código aberto**: Ter uma comunidade vibrante que oferece suporte à tecnologia de programação é essencial para seu aprimoramento contínuo. Felizmente, JavaScript e suas contrapartes de back-end são plataformas de código aberto com comunidades ativas de desenvolvedores de back-end e front-end que impulsionam inovações contínuas.
* **Recursos de I / O**: a natureza robusta do Express.js e do Node.js permite que ele lide com toneladas de solicitações de I / O e notificações de aplicativos conectados.

## Limitações

* As funções orientadas a eventos dos back-ends JavaScript são um tanto complicadas, fazendo com que os desenvolvedores com experiência em outras plataformas interpretem mal os retornos de chamada. Este desafio foi resolvido principalmente nas versões mais recentes.
* Muitos programadores que usam o desenvolvimento do lado do servidor JavaScript não entendem como o middleware funciona.
* Hospedar estruturas de back-end de JavaScript com banco de dados MySQL é complicado.
* Alguns desenvolvedores acham que a abordagem de design padrão para o desenvolvimento é preferível à liberdade que as estruturas de back-end JavaScript não definidas oferecem.

# PHP

PHP foi desenvolvido por Rasmus Lerdorf em 1994; desde então, o PHP evoluiu para uma das principais linguagens de programação do lado do servidor hoje.

PHP é uma linguagem de programação multifuncional fácil de usar. Funciona perfeitamente com uma ampla variedade de bancos de dados e sistemas operacionais. Estruturas modernas, uma base de código enorme e a comunidade PHP ativa são fatores que impulsionam a evolução contínua do PHP.



## Características

* **Código aberto e versátil**: há muitas bibliotecas PHP online gratuitas que os desenvolvedores podem aproveitar para desenvolver códigos de back-end rapidamente. Quase todos os sistemas operacionais, como Windows e Linux, oferecem suporte a PHP. Além disso, os aplicativos PHP também podem ser iniciados em qualquer servidor web.
* **Econômico**: o PHP está disponível gratuitamente e conta com o suporte de uma vibrante comunidade de desenvolvedores. Contratar um desenvolvedor PHP não é caro. O 2020 Stack Overflow Developer Survey sobre as tecnologias mais bem pagas classificou o PHP da penúltima posição.
* **Fácil de usar**: surgiram muitos frameworks PHP que simplificam a programação, eliminando a necessidade de escrever códigos SQL tediosos. Alguns deles usam o sistema Object Relational Mapping (ORM) que funciona como o model-view-controller (MVC) para escrever funções do lado do servidor rapidamente.
* **Excelente para iniciantes**: a simplicidade do PHP o torna uma linguagem ideal para novos desenvolvedores. Eles podem começar a funcionar rapidamente devido à curta curva de aprendizado.
* **Funções de automação**: o recurso de script do PHP o torna útil para a criação de automação, como autenticação, mapeamento de URL, gerenciamento de sessão e muito mais.
* **Segurança embutida**: Embora muitas pessoas pensem que o PHP não é seguro, ele tem muitos recursos de segurança embutidos que permitem a você mitigar várias ameaças.

## Limitações

* A influência do PHP como tecnologia de desenvolvimento está diminuindo. Hoje em dia, as pessoas dificilmente incluem PHP em suas habilidades de desenvolvimento.
* PHP não pode competir de forma eficiente com tecnologias de desenvolvimento modernas como Ruby e Python devido a bibliotecas limitadas.
* Como uma plataforma de código aberto, o PHP é suscetível ao uso indevido e à criação de códigos com erros.

# Phyton

Desde que o Python foi criado em 1991 por Guido van Rossum, ele cresceu e se tornou uma das linguagens de programação multifuncionais líderes no mundo hoje. Os desenvolvedores de back-end utilizam seus códigos organizados e altamente legíveis para criar scripts funcionais para lidar com atribuições de back-end.

Uma pesquisa de 2020 feita por Stack Overflow revelou que é a principal linguagem de programação de back-end que os desenvolvedores esperam aprender. Python liderou essa categoria por quatro anos consecutivos. Ele também foi classificado em terceiro lugar entre as linguagens de programação mais amadas existentes.



## Características

* **Relativamente fácil de aprender**: um dos benefícios do Python é o estilo de codificação semelhante ao inglês, que o torna altamente legível. Portanto, codificar e ler códigos Python é relativamente fácil para programadores novos e experientes.
* **Bibliotecas enormes**: Python conta com o suporte de bibliotecas enormes que reduzem a necessidade de escrever códigos manualmente. Algumas bibliotecas contêm códigos que aprimoram tarefas como e-mail, navegação, atribuições de banco de dados, teste de unidade e muito mais.
* **Eficiente em termos de custos**: além de Python ser uma plataforma de código aberto para download gratuito, ele também oferece várias ferramentas e recursos gratuitos que aprimoram os projetos de desenvolvimento de aplicativos.
* **Recursos de IoT**: os desenvolvedores podem aproveitar os recursos modernos do Python para criar objetos físicos Raspberry Pi.
* **Códigos incorporáveis**: com a regra Write Once Run Anywhere (WORA), o código python pode ser incorporado no código-fonte de outras linguagens, como C ++.

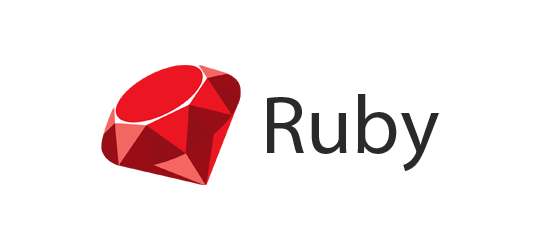
## Limitações

* A execução do código Python fica lenta quando é interrompida. Essa deficiência prejudica todo o projeto de desenvolvimento de aplicativos.
* A camada de acesso ao banco de dados Python é menos desenvolvida em comparação com outras linguagens de programação de back-end.
* Testes extensivos são necessários para detectar erros e bugs nos códigos Python.
* É altamente dependente de bibliotecas e estruturas de terceiros.

# Ruby

Ruby foi desenvolvido em 1990 por Yukihiro Matsumoto, um especialista japonês em computadores. A linguagem de programação que compartilha sintaxe semelhante com Python, Java e Perl também surgiu como uma das tecnologias de back-end líderes hoje. É uma plataforma popular para o desenvolvimento de aplicativos da web e é preferida pela maioria dos desenvolvedores para prototipagem.

Algumas das principais plataformas online, como Shopify, Airbnb e Etsy, usam Ruby para automatizar tarefas de back-end. Os recursos orientados a objetos, flexíveis e escaláveis o tornam a plataforma preferida.



## Características

* **Altamente produtivo**: Ruby é altamente produtivo para realizar atribuições de back-end devido à sua alta legibilidade de código e muitas bibliotecas de terceiros. Os desenvolvedores podem acessar todas as ferramentas de que precisam no Ruby; eles dificilmente precisam fazer referência à documentação externa.
* **Metaprogramação**: Ruby suporta metaprogramação, o que permite que um programa adquira dados de outros programas e se modifique durante a execução. Junto com outras ferramentas, como macros, scripts e introspecção, agilize a entrega de projetos de aplicativos em Ruby.
* **Biblioteca massiva**: Ruby pode tirar proveito de vários módulos de terceiros disponíveis em RubyGems.
* **Recursos de teste**: Ruby é equipado com uma biblioteca enorme contendo vários recursos de teste e automatizados que aprimoram o desenvolvimento de produtos de alta qualidade.
* **Rápido e confiável**: os scripts de back-end podem ser criados mais rapidamente com Ruby em comparação com outras tecnologias de script. Os métodos de programação orientados a objetos tornam o Ruby mais resiliente e excelente para fazer consultas e processar resultados. Ele pode desenvolver códigos 40% mais rápido do que muitas outras linguagens de script.

## Limitações

* O tempo de execução de outras linguagens de script é relativamente mais rápido do que Ruby.
* Apesar de ter uma comunidade de código aberto, Ruby tem fontes e bibliotecas limitadas.
* Ruby é novo em comparação com outras linguagens de programação; os desenvolvedores de back-end existentes acham difícil aprender.
* Os códigos Ruby são difíceis de depurar.

# Conclusão

Em suma, podemos concluir que existem diversas linguagens de programação, assim possuindo várias possibilidades para a pessoa que quer trabalhar na área de TI como back-end.

Utilizando suas características, limitações e popularidade no mercado de trabalho como critério de avaliação, para saber se vale a pena ou não aprender a linguagem.

# Referências

AS 11 LINGUAGENS DE BACK-END MAIS USADAS PARA PROGRAMAR. In: KENZIE. 2022. Disponível em: <https://kenzie.com.br/blog/linguagens-de-programacao-backend/>

TOP 10 LINGUAGENS DE PROGRAMAÇÃO PARA BACKEND. In: BACKAPP. Disponível em: https://blog.back4app.com/pt/top-10-linguagens-de-programacao-para-backend/