

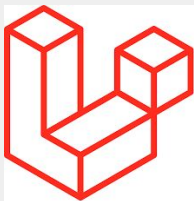
Conhecendo as linguagens de *Beckend*

Objetivo

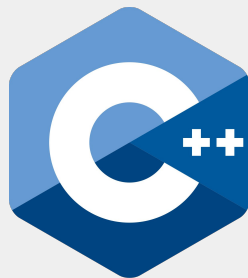
- Conhecer a existência de diferentes linguagens de programação utilizadas para o desenvolvimento backend.
- Entender as vantagens e desvantagens de cada uma delas.
- Entender onde cada uma pode se sair melhor



express



django



Linguagens que vamos conhecer

- NodeJs (JavaScript)
- PHP
- Python
- Java
- C#
- C++
- Ruby
- Go
- Rust



Rust

Principais utilizações

- Sistemas de alto desempenho e segurança;
- Sistemas críticos

Vantagens

- Segurança de memória
- Excelente performance
- Moderna e inteligente

Desvantagens

- Curva de aprendizado elevada (difícil)
- Ainda em amadurecimento
- Desenvolvimento mais lento (mais rigoroso)



Firefox



C#

Principais utilizações

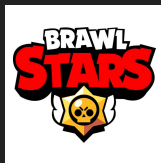
- Aplicações corporativas
- Jogos
- APIs

Vantagens

- Excelente integração com Windows e Azure
- Excelente integração com banco de dados SQL Server
- Ferramentas poderosas (Visual Studio)
- Boa performance

Desvantagens

- Historicamente mais associada ao Windows
- Menor comunidade fora do Windows





C++

Principais utilizações

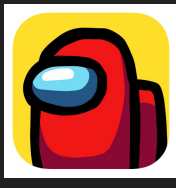
- Jogos
- Sistemas embarcados
- Banco de dados
- Motor gráficos

Vantagens

- Alto desempenho
- Controle total da memória*
- Adequado para sistemas críticos

Desvantagens

- Complexo
- Curva de aprendizado alta (difícil)
- Mais propenso a erros
- Ciclo de desenvolvimento mais lento





Ruby

Principais utilizações

- Desenvolvimento Web
- MVPs rápidos

Vantagens

- Alta produtividade (com Rails)
- Código limpo e elegante
- Boa para prototipação e startups

Desvantagens

- Performance inferior
- Dificuldade em escalar
- Comunidade em declínio



GoLang

Principais utilizações

- APIs
- Micro-serviços
- Ferramentas CLI
- Cloud

Vantagens

- Sintaxe simples
- Curva de aprendizado menor (mais fácil)
- Excelente performance
- Concorrência nativa (go routines)
- Compilado com binários pequenos

Desvantagens

- Falta de recursos mais avançados
- Gestão de dependência evolui devagar
- Poucas palavras reservadas (exige muita implementação do zero)





PHP

Principais utilizações

- Web
- Ecommerces

Vantagens

- Fácil de aprender e bem documentado
- Grande comunidade
- Suporte nativo em quase todos os servidores web

Desvantagens

- Histórico de más práticas*
- Menor desempenho
- Inconsistência na linguagem (nome de funções, parâmetros, etc)





Python

Principais utilizações

- APIs
- Automações
- Ciência de dados
- IA

Vantagens

- Sintaxe simples e legível
- Ecossistema muito rico
- Alta produtividade

Desvantagens

- Performance inferior (interpretado)
- Multithreading limitado
- Menos adequado para aplicações altamente concorrentes



angelohank / recomendacao-curso-GA



angelohank / merge-data-analyzer





Java

Principais utilizações

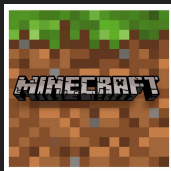
- Sistemas corporativos
- Aplicações financeiras
- APIs
- Android

Vantagens

- Extremamente robusta e escalável
- Grande ecossistema de bibliotecas e frameworks
- Amplo suporte corporativo
- Garbage Collector

Desvantagens

- Verbosidade
- Curva de aprendizado maior
- Consumo de memória alto (pesado)





Node

Principais utilizações

- APIs Rest
- Apps de tempo real (websocket, chats, etc)
- Micro-serviços

Vantagens


- Alta performance com I/O assíncrono
- Ecossistema grande via npm
- Ideal para aplicações *real time*
- Permite usar a mesma linguagem no frontend e backend


Desvantagens

- Callback hell
- Performance inferior em CPU-bound
- Maturidade menor em comparação com outras linguagens





Hello World!


 hello-cpp.cpp


 hello-csharp.cs


 hello-go.go


 hello-java.java

 hello-node.js

 hello-php.php

 hello-python.py

 hello-ruby.rb

 hello-rust.rs