Go & TypeScript

Linguagens de Programação

Davi Ventura
Edmilson Lino

Tópicos

01

Introdução

Criada pela **Google** e lançada em **código livre** em **2009**. Sua origem deu-se quando a Google estava com um problema: seus sistemas eram em **C++** e em **C**, e o processo de compilar esses programas para gerar um executável era **complicado** e **demorado**. Com isso, os engenheiros do Google tiveram a ideia de criar uma nova linguagem de programação. Sua **popularidade** vem crescendo e acredita-se que a Go possa **substituir** o **Java**.

Linguagem de programação de código aberto desenvolvida pela **Microsoft** em **2012**. Mais conhecido como um **superset** do **Javascript**, ou seja, um conjunto de ferramentas, o TypeScript foi criado com o objetivo de incluir recursos que não estão presentes no JS. Foi considerada pelo público a **4ª** linguagem **"mais amada"**, de acordo com uma pesquisa do **Stack Overflow** em **2018**, e está entre as **15** linguagens mais **populares**, de acordo a **RedMonk**.

02

Características

Vantagens e Desvantagens das Linguagens

O uso de Go vem **crescendo** em todo o mundo, especialmente no ambiente de **Cloud Computing, como Docker** e **Kubernetes**. Além disso, algumas **empresas** fazem **uso** da linguagem Go, além do próprio Google, sendo elas: **Mercado Livre, iFood, Heroku, SoundCloud, etc**.

Vantagens:

- Facilidade de uso
- Linguagem de Segurança
- Excelente documentação

Desvantagens:

- Menos versátil
- Consome mais recursos computacionais
- Construir um grande ecossistema para Go levará tempo

O TypeScript é utilizado para criar **funções tipadas** e **bem modeladas** e que não tenham qualquer **problema**. Naturalmente, por possuir mais recursos do que o **JavaScript**, o **TypeScript** é considerado uma versão **melhor** pelos desenvolvedores.

Ao utilizarmos o TypeScript, temos a possibilidade de aplicar a **tipagem estática** à programação juntamente com interfaces em um sistema construído com Javascript, ou seja, podemos **turbinar** as nossas aplicações. Entre esses conceitos estão: **Encapsulamento**, **Herança**, **Abstração** e **Polimorfismo**.

Vantagens:

- Apresenta erros no momento da organização
- Tipagem Estática
- Executado em qualquer programa ou motor JavaScript
- Ajuda na organização do código
- Melhor documentação para API's

Desvantagens:

- Documentação de definição não acessíveis em determinados momentos
- A natureza dos documentos de definição de tipo é uma preocupação
- Necessita ser alterado para JavaScript
- Não é totalmente coexpressivo com JavaScript

03

Critérios de Avaliação

Legibilidade:

- **Simplicidade Geral:** sintaxe simplificada e eficiente, fornecendo ao desenvolvedor apenas os recursos suficientes para uma aplicação desejada, bem como facilita manutenções póstumas;
- **Ortogonalidade:** liberdade para se utilizar N ferramentas para aplicações, permitindo fácil interpretação sobre códigos;
- **Instruções de Controle:** *if/else, switch/case, while e for;*
- **Tipo de Dados:** tipos inteiros, pontos flutuantes, double, booleanos, strings char e, no caso o float64;
- **Sintaxe:** uma das premissas norteadoras do Go foi proporcionar uma linguagem com sintaxe de fácil compreensão como o C e Java, proporcionando melhor adaptação para a linguagem.

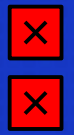
Capacidade de Escrita:

- **Suporte para Abstração:** Apesar de que Go não seja orientada a objeto, a linguagem realiza a abstração de dados através de suas structs de tipos, tendo o mesmo propósito de uma classe. Go também dispõe de pacotes (packages) sem a necessidade de implementar dentro das inclusões de bibliotecas;
- **Expressividade:** Através do construtor ":=", Go associa variáveis e valores sem necessitar declarar seus tipos (exceto em Métodos). Outra função presente em Go é o "fallthrough" (queda), que possui efeito inverso ao break na linguagem C.

Confiabilidade:

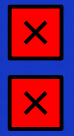
- **Verificação de Tipos:** Por ser uma linguagem fortemente tipada, realiza a checagem de tipos em tempo de compilação, a fim de que não ocorram bugs enquanto executa o código;
- **Manipulação de Exceções:** Go possui funções com o intuito de proteger o código de cometer bugs em caso de aplicações complexas, como sistemas de segurança. Destaca-se: ***Defer, Panic, Recover, Aliasing***.

Custos:



- **Aprendizagem:** Devido à sua praticidade, simplicidade e ampla documentação oficial, Go proporciona melhor nível de aprendizagem para desenvolvedores, de juniors a seniors, devido sua familiaridade com C;
- **Compilação:** Uma das premissas da linguagem, sua compilação é feita de forma rápida, diminuindo portanto o seu custo;
- **Execução do Programa:** Através de goroutines (ou threads), acaba por economizar recursos computacionais, sendo tão rápida quanto a linguagem C;
- **Sistema de Implementação:** Possui suporte a todos os sistemas operacionais, cabendo ao desenvolvedor instalar conforme sua arquitetura (x64);
- **Sistemas Críticos:** Go é recomendado para sistemas de segurança (bancos, por exemplo), entretanto, peca na aplicação em sistemas de tempo real, como controle de tráfego aéreo e rastreamento veicular.

Legibilidade:



- **Simplicidade Geral:** extremamente simples escrever códigos em Javascript. Este é um dos muitos fatores que dão tanta popularidade ao TypeScript atualmente;
- **Ortogonalidade:** liberdade para se utilizar N ferramentas para aplicações, permitindo fácil interpretação sobre códigos;
- **Instruções de Controle:** temos algumas estruturas de decisão que podem nos ajudar na programação no lado do cliente, como por exemplo, *if/else* e *switch;*
- **Tipo de Dados:** *String, Boolean, Number, Array, Tuple, Enum, Any e Void*;
- **Sintaxe:** o desenvolvedor lidará diretamente com uma sintaxe simplificada, mais clara e amplamente suportada por editores de código modernos.

Capacidade de Escrita:

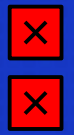
- **Suporte para Abstração:** dispomos de recursos que melhor suportam o uso da Programação Orientada a Objetos, que tem como base quatro princípios fundamentais, sendo um deles justamente a abstração;

- **Expressividade:** O TypeScript introduziu inovações ao longo do ano passado para realmente descrever a expressividade do JavaScript em seu sistema de tipos. Entre tipos condicionais, novas verificações mais rígidas, etc.

Confiabilidade:

- **Verificação de Tipos:** Consiste em ***classes, interfaces, herança,*** **etc**. É estrita e tipificada estaticamente como Java;
- **Manipulação de Exceções:** O TypeScript permite tratar uma exceção, de forma similar ao C#. Basicamente temos três blocos: ***try, catch, finally***.

Custos:



- **Aprendizagem**: Em decorrência da sua praticidade e simplicidade, o TypeScript proporciona melhor nível de aprendizagem para desenvolvedores, de juniors a seniors, devido sua familiaridade com JavaScript;
- **Compilação**: O arquivo TypeScript não roda diretamente no navegador enquanto o JavaScript é executado. Sendo assim, temos que compilar o arquivo TypeScript em JavaScript, então ele funcionará normalmente;
- **Execução do Programa**: Através de goroutines (ou threads), acaba por economizar recursos computacionais, sendo tão rápida quanto a linguagem C;
- **Sistema de Implementação**: Possui suporte a todos os sistemas operacionais, cabendo ao desenvolvedor instalar conforme sua arquitetura (x64);
- **Sistemas Críticos**: Por utilizar tipagem forte, estática, e uma linguagem que facilite a escrita de um código estruturado pode fazer toda diferença para pegar um bug em tempo de desenvolvimento, ou descobri-lo em produção após uma invasão que custe muito caro financeiramente à empresa.

04

Exemplos de Código

Calculadora:

Link para acessar o código completo

```
132  func calcularRaizQuadrada() float32 {
133          resultado := (num1 / 1.0 / 2.0)
134          return resultado
135  }
136
137  func optPotencia() {
138          fmt.Print("deseja calcular a potência de :")
139          fmt.Scan(&num1)
140          fmt.Print("\nelevado á :")
141          fmt.Scan(&num2)
142  }
143
144  func optRaizQuadrada() {
145          fmt.Print("Calcular a raiz Quadrada de :")
146          fmt.Scanf("%f", &num1)
147  }
```

To-Do List:

Link para acessar o projeto completo

Link para Testar via Web

```
    if (content && !statusInput) {
      setStatusInput(true);
    }

    return (
      <form onSubmit={handleSubmit}>
        <HStack mt='4' mb='4'>
          <Input
            h='46'
            borderColor={!statusInput ? 'red.300' : 'transparent'}
            variant='filled'
            placeholder='Digite sua tarefa'
            value={content}
            onChange={(e) => setContent(e.target.value)}
          />
          <Button
          colorScheme='blue'
          px='8'
          pl='10'
          pr='10'
          h='46'
          type='submit'>Adicionar</Button>
        </HStack>
      </form>
    );
}
```

Sua simplicidade, além do suporte nativo de multiprocessamento, é um dos motivos pelos quais Go ganhou o seu espaço no coração dos desenvolvedores, seguindo um dos objetivos principais da linguagem: tornar o desenvolvimento de softwares modernos mais simples, rápido e produtivo.

Como vimos, o TypeScript é uma linguagem de programação de código aberto criada recentemente para aprimorar e aperfeiçoar o JavaScript. O intuito tem dado certo e, por isso, sua popularidade tem crescido a cada ano, permitindo que ela ocupe importante espaço na rotina de trabalho dos programadores.

05 Referências Bibliográficas:

ALURA. **Go - a linguagem do Google**. Disponível em: <https://www.alura.com.br/conteudo/golang>. Acesso em: 29 de agosto de 2022.

ANNEBÄCK, Joakim; STJERNBERG, Johan. **A comparison between Go and C++**. Disponível em: <https://www.csc.kth.se/utbildning/kth/kurser/DD143X/dkand11/Group9Alexander/Joakim_Anneback_Johan_Stjernberg.finalreport.2.pdf>. Acesso em: 29 de agosto de 2022.

CAMPOMORI, Cleber. **Precisamos falar sobre TypeScript**. Disponível em: <https://www.treinaweb.com.br/blog/precisamos-falar-sobre-o-typescript>. Acesso em: 30 de agosto de 2022.

EDSON. **Introdução ao TypeScript**. Disponível em: <https://www.devmedia.com.br/introducao-ao-typescript/36729/>. Acesso em: 30 de agosto de 2022.

GOULARTE, Suelen. **Golang - a Linguagem do Google**. Disponível em: <https://www.ime.usp.br/~gold/cursos/2015/MAC5742/reports/GoLang.pdf>. Acesso em: 30 de agosto de 2022.

Redação Vulpi. **Do JavaScript ao TypeScript, Why?**. Disponível em: <https://blog.vulpi.com.br/javascript-typescript/>. Acesso em: 28 de agosto de 2022.

ROSA, Daniel. **Go explicada e exemplos de código**. Disponível em: <https://www.freecodecamp.org/portuguese/news/genericos-em-go-explicados-com-exemplos-de-codigo/>. Acesso em: 29 de agosto de 2022.

SEBESTA, Robert. **Conceitos de Linguagem de Programação**. Ed.Bookman. Publicado em: 17 de janeiro de 2011.

SESTOFT, Peter. **Programming Language Concepts (Undergraduate Topics in Computer Science)**. Ed.Springer. Publicado em: 31 de agosto de 2017 (2ª edição.

MORAES, Murilo. **Go Search**. Disponível em: <https://github.com/kliff-k/go-search>. Acesso em: 26 de agosto de 2022.

Obrigado!

Alguma Pergunta?