**Introdução**

Go é uma linguagem de programação compilada e livre (gratuita/ de código aberto). Associada aos melhores salários( 3º lugar de acordo com [StackOverflow Developer Survey 2019](https://insights.stackoverflow.com/survey/2019#top-paying-technologies)), Go vem sendo cada vez mais utilizada e adorada por programadores. Poderosa e de fácil aprendizagem, Golang é incrivelmente ‘leve’ em termos de memória, o que aumenta drasticamente sua praticidade.

**Origens e influências**

Em 2007, A linguagem de programação Go foi criada como resposta a alguns problemas que surgiram a partir do uso de processadores multinúcleos, sistemas distribuídos em redes, computação massiva e modelo de computação web, que apesar de potenciar a computação, agregou também uma série de problemas, assim como acontece com as novas tecnologias.

Com esses requisitos, surgiram tentativas frustradas de desenvolver uma solução com sistemas de desenvolvimento já existentes, a maior parte em C++, Java e Python. Contudo, o mais desejável era encontrar todas as características( Garbage collector, estaticamente tipada, concorrências) dessas linguagem em uma só.

A linguagem foi desenvolvida por engenheiros da Google em 2007 como um projeto de tempo parcial. Seus fundadores foram Ken Thompson, Robert Griesemer e Rob Pike.

Em 2008, Go passou a ser um projeto de tempo integral, sendo assim, muitos outros engenheiros se juntaram ao projeto. Em 2009, se tornou Open Source e em 2010, a linguagem começou a ser adotada por desenvolvedores de fora do Google.

**Classificação**

* **Compilada:**
  + Por meio de compilador( gc e gccgo), cria um arquivo único executável em linguagem de máquina (binário).
* **Paradigmas:**
  + Concorrente
  + Imperativa
  + Procedural
  + Reflexiva
  + Estruturada
* **Multiplataforma**:
  + Pode ser usada nos mais diversos sistemas operacionais (Windows, Linux, Mac Os, mobile, FreeBSD).
* **Inferência de tipo e tipagem estática:** 
  + O tipo da variável não precisa ser necessariamente declarado( o tipo da variável é inferido a partir do valor do lado direito), porém, não pode mudar de tipo ao longo da execução. Tem tipagem estática, ou seja, existe uma validação no tempo de compilação para evitar erros.
* **Open-source:** 
  + Código fonte disponível e permite a colaboração de qualquer desenvolvedor.

**Avaliação comparativa**

Os dois programas a seguir, calculam o Fibonacci de 0 até 40 e guardam em um arquivo.

Fibonacci em Go:

package main

import (

"fmt"

"os"

"strconv"

)

func fib(n int ) int {

if n==0 { return 0

}else if n==1 {return 1

}else {return fib(n-1)+ fib(n-2)

}

}

func main() {

f, err := os.Create("text.txt")

if err != nil {

panic("cannot create file")

}

defer f.Close()

for i:=0; i <= 40 ; i++ {

fmt.Fprintln(f,strconv.Itoa(fib(i)))

}

}

Fibonacci em Lisp:

(defun fibonacci(n)

(cond

((eq n 0) 0)

((eq n 1) 1)

( (+ (fibonacci (- n 1)) (fibonacci (- n 2))))))

(with-open-file (str "text.txt"

:direction :output

:if-exists :supersede

:if-does-not-exist :create)

(loop for a from 0 to 40

do (format str (write-to-string(fibonacci a)))

(format str "~%" )) )

As duas linguagens diferem muito uma da outra, por conta da tipagem, e da sintaxe das mesmas.

**Readability:**

Na questão readability, Go ganha de Lisp, pelo fato de que em Lisp, todas as operações que acessam as suas estruturas de dados são tratadas como funções, dificultando o entendimento do código.

**Writeability:**

Analisando Writeability, claramente Go ganha essa disputa, já que em Lisp você precisa de um alto nível de abstração, especialmente quando as funções são utilizadas, suprimindo muitos detalhes da programação.

**Expressiveness:**

Quanto a expressiveness, a estrutura de ambas as linguagens são bem diferentes, porém Go é mais expressiva que Lisp. Go é mais intuitiva, simples, concisa e eficiente. Lisp só usa funções recursivas e suas funções devem ser parêntizadas, fazendo com que a estruturação do programa fique mais 'pesada', portanto, precisando de um nível de abstração muito elevado. Além de Go ser muito mais rápida que Lisp por conta das goroutines.

**Conclusão**

Go vem cada vez mais sendo utilizado por ser rápido, tem tipagem estática e é uma linguagem compilada, mas que faz você se sentir em uma linguagem dinâmica e interpretada.

Portanto, o aprendizado rápido,a velocidade, a utilização no desenvolvimento por companhias famosas, e ser a terceira linguagem mais rentável para os programadores, e o fato de ser open-source, fazem com que Go seja, sem dúvida, uma aposta para o futuro das linguagens no desenvolvimento de sistemas.

**Bibliografia**

<https://pt.wikipedia.org/wiki/Go_(linguagem_de_programa%C3%A7%C3%A3o)>

<https://insights.stackoverflow.com/survey/2019#top-paying-technologies>

<https://golang.org/>