Corrotinas

Paradigmas de Programação

Centro Universitário Senac

Prof. Celso Crivelaro

celso.vcrivelaro@sp.senac.br

O que são corrotinas?

São threads leves, chamadas simplificadas

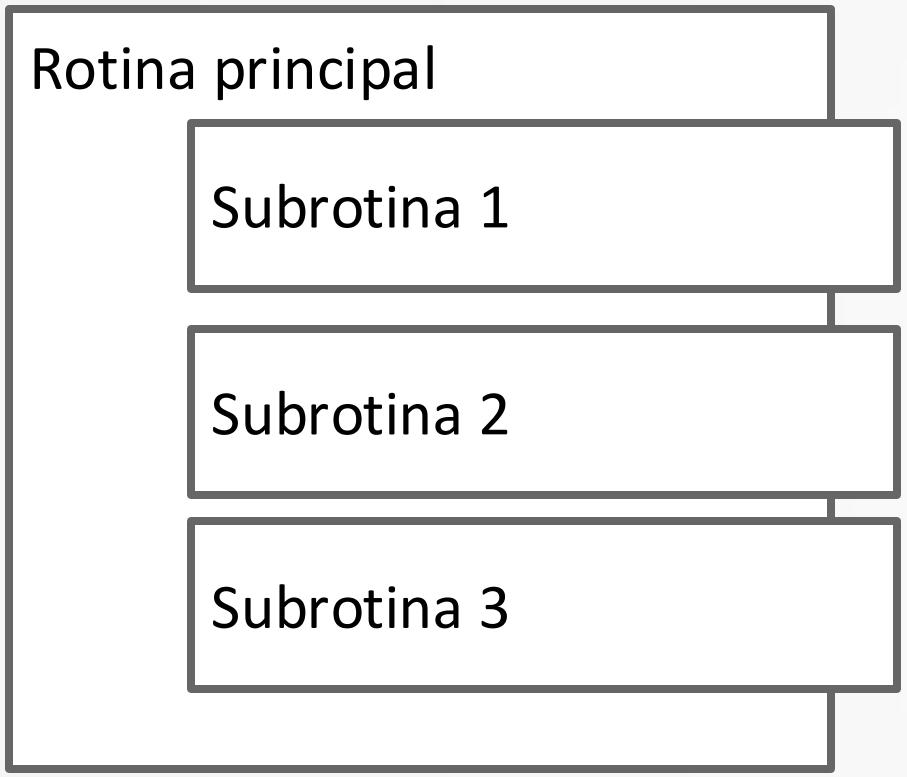
Cada runtime gerencia chamada, parada e troca de informações

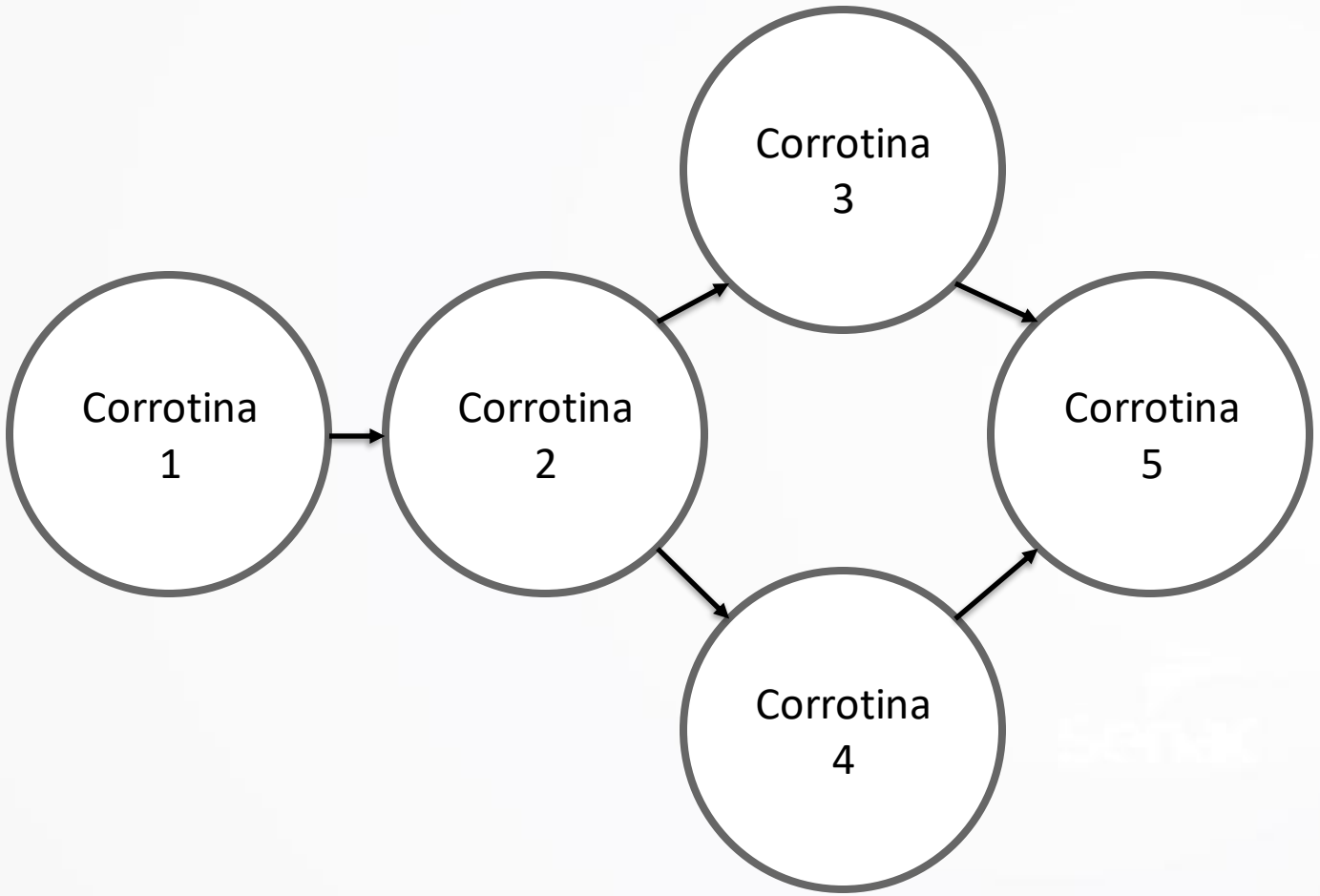
Computação não-bloqueante, passando direto ou parando para esperar

Pode ser suspensa

https://medium.com/@kishor007sutar/kotlin-coroutines-3d70d16addd1

Subrotina x Corrotina

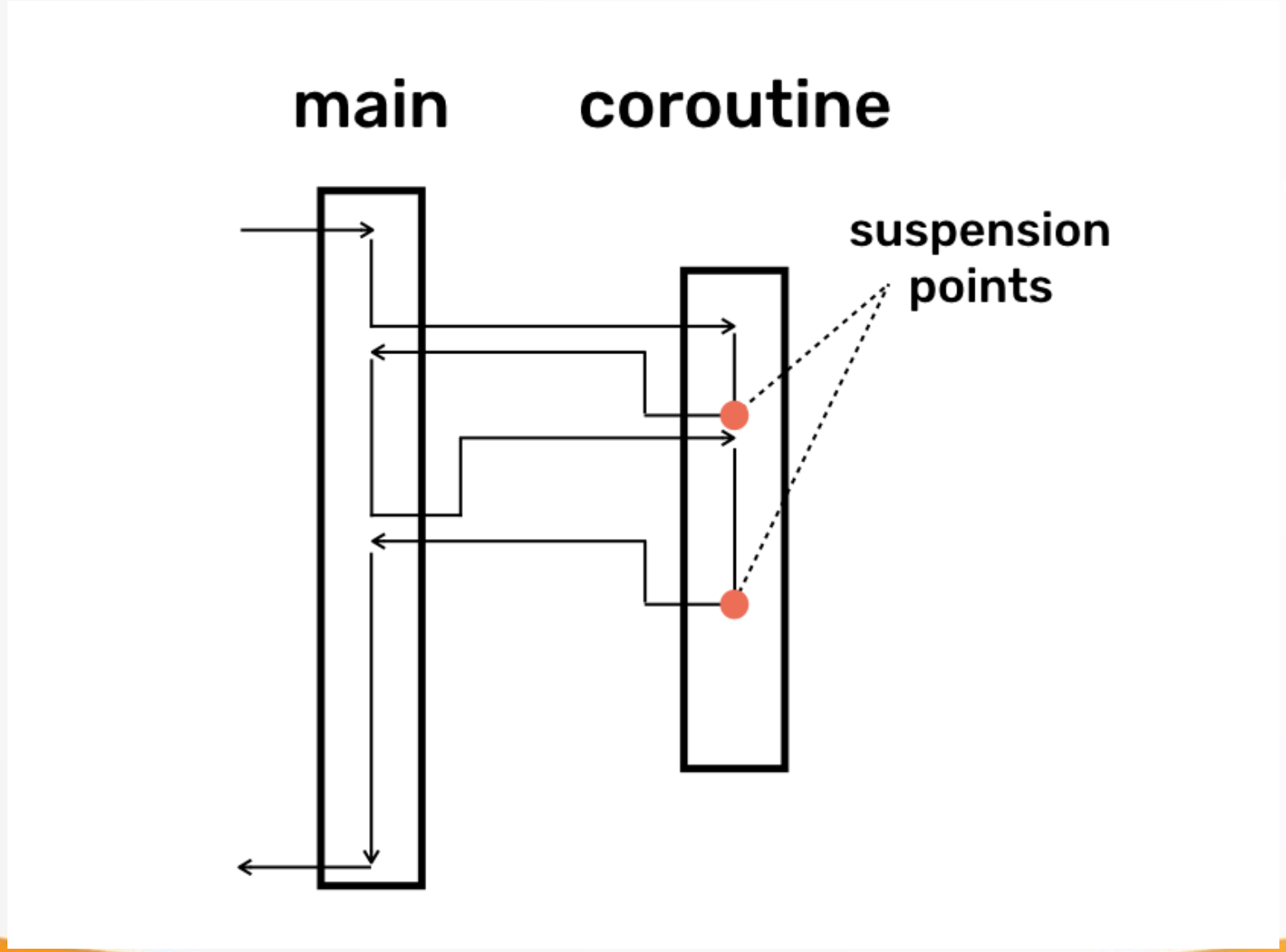




Sequencial

Concorrente

Corrotina




GO


Lua


R

Estudo de caso: Go

Robert Griesemer, Rob Pike, Ken Tompson, criadores do Go

Criada em 2009 pelo Google

Compiladada, estaticamente tipada e com foco em concorrência

Criadores estavam querendo algo melhor do que o C++

Fun fact: Ken Tompson foi cocriador do C

Aplicações

- Web servers
- Game servers
- Docker e Kubernets
- Etherium

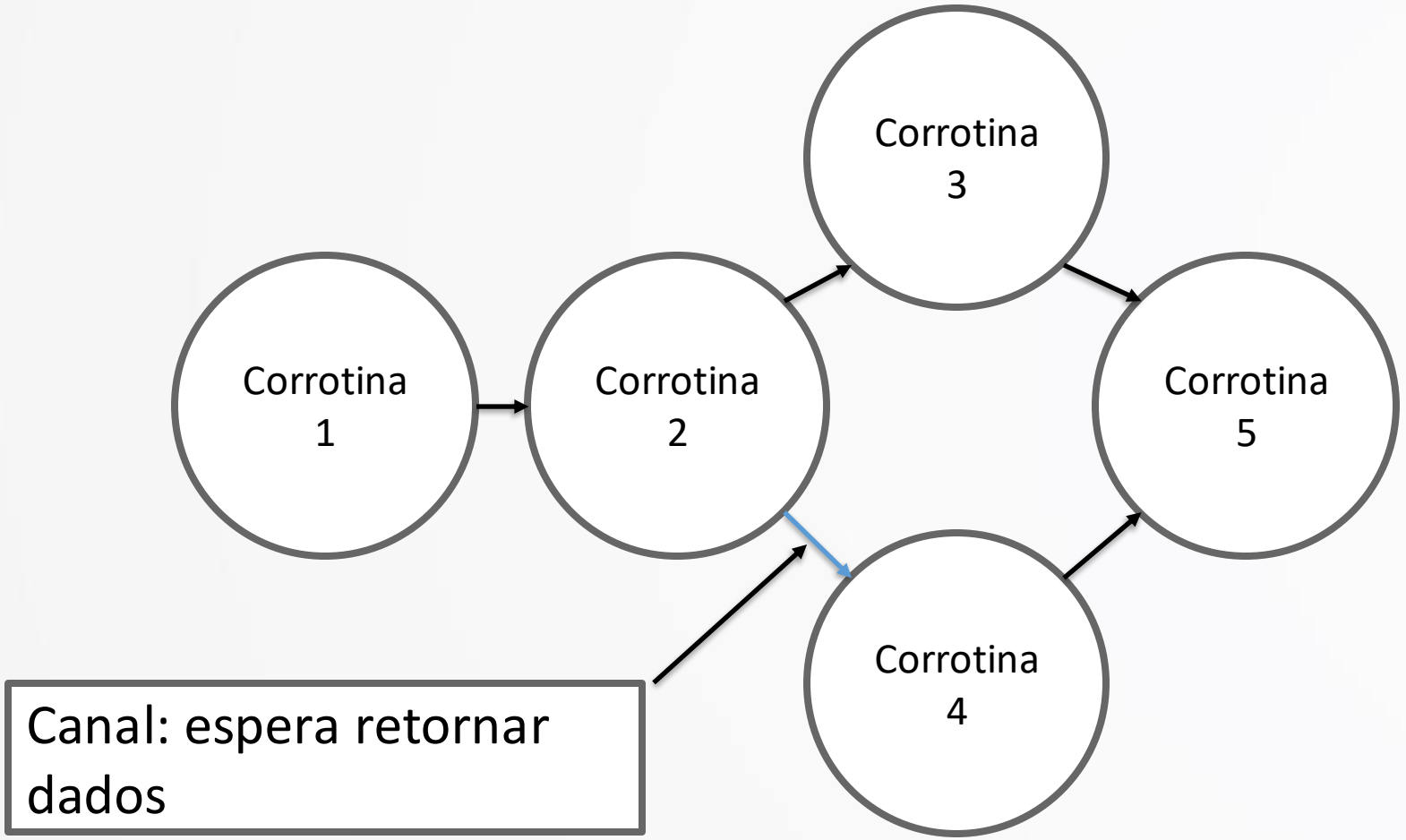
Corrotinas em GO

Simplesmente se chama “go função”

Execução começa em segundo plano

https://repl.it/@celsosenac/concorrencia-corrotina

Como se comunicam? Canais



Canais

Evitam o compartilhamento de memória

Troca de informações entre as corrotinas

https://repl.it/@celsosenac/concorrencia-corrotina-canal

https://repl.it/@celsosenac/concorrencia-corrotina-select

GO by Example

Go by Example

Go is an open source programming language designed for building simple, fast, and reliable software.

Go by Example is a hands-on introduction to Go using annotated example programs. Check out the first example or browse the full list below.

Hello World
Values
Variables
Constants
For
If/Else
Switch
Arrays
Slices
Maps
Range
Functions
Multiple Return Values
Variadic Functions

https://gobyexample.com/

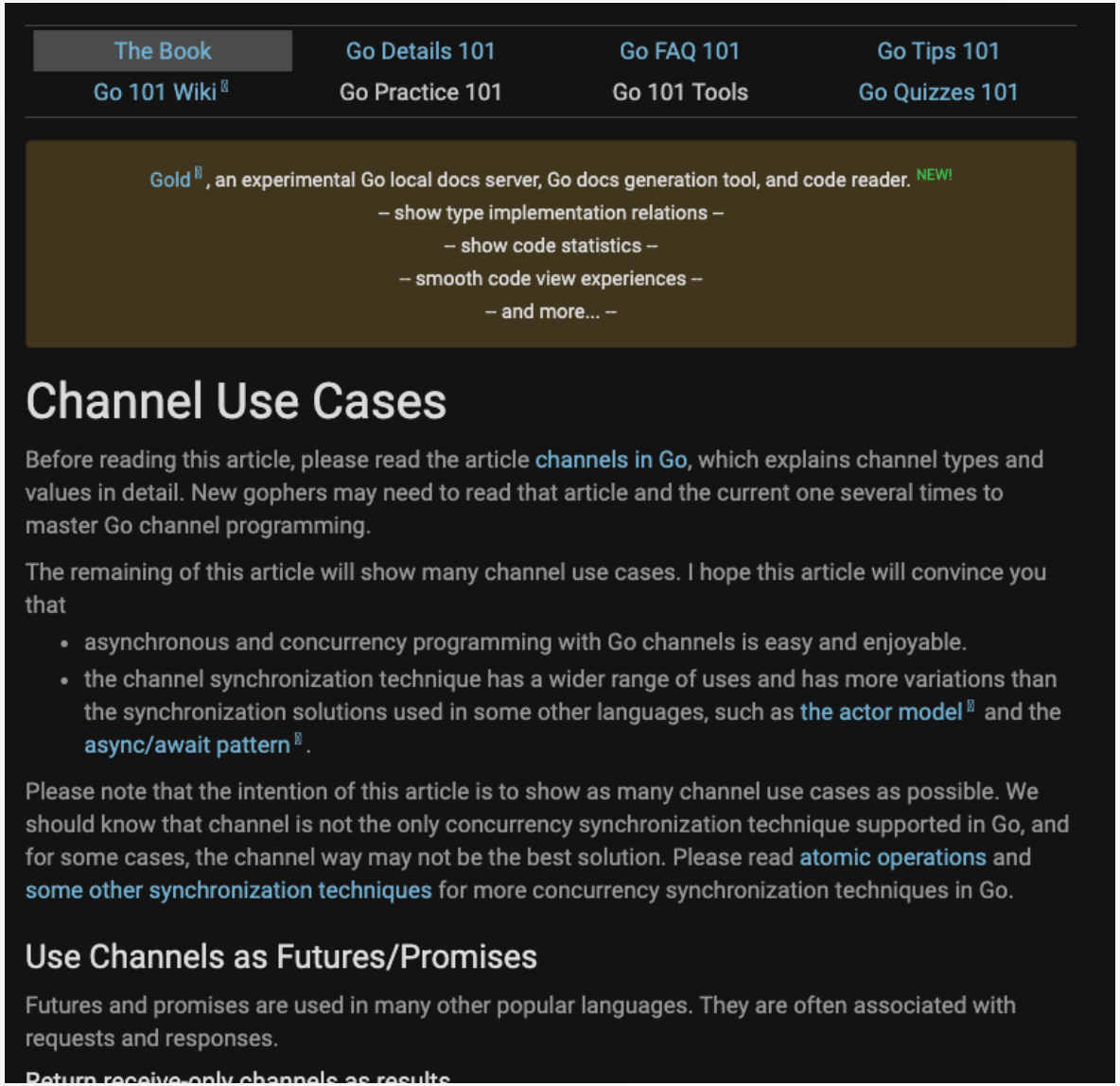
Padrões

Go Concurrency Patterns

Rob Pike
Google

https://talks.golang.org/2012/concurrency.slide#1

Padrões



The Book | Go Details 101 | Go FAQ 101 | Go Tips 101
Go 101 Wiki | Go Practice 101 | Go 101 Tools | Go Quizzes 101

Gold, an experimental Go local docs server, Go docs generation tool, and code reader. NEW!
– show type implementation relations –
– show code statistics –
– smooth code view experiences –
– and more... –

Channel Use Cases

Before reading this article, please read the article channels in Go, which explains channel types and values in detail. New gophers may need to read that article and the current one several times to master Go channel programming.

The remaining of this article will show many channel use cases. I hope this article will convince you that

- asynchronous and concurrency programming with Go channels is easy and enjoyable.
- the channel synchronization technique has a wider range of uses and has more variations than the synchronization solutions used in some other languages, such as the actor model and the async/await pattern.

Please note that the intention of this article is to show as many channel use cases as possible. We should know that channel is not the only concurrency synchronization technique supported in Go, and for some cases, the channel way may not be the best solution. Please read atomic operations and some other synchronization techniques for more concurrency synchronization techniques in Go.

Use Channels as Futures/Promises

Futures and promises are used in many other popular languages. They are often associated with requests and responses.

Return receive-only channels as results

https://go101.org/article/channel-use-cases.html

Muito Obrigado!

Centro Universitário Senac

Prof. Celso Crivelaro
celso.vcrivelaro@sp.senac.br