

## Kotlin – Suspend fun (Parte 1)

**Funções** suspensas estão no centro de todas as Coroutines. Uma função de suspensão é simplesmente uma função que pode ser pausada e retomada depois. Elas podem executar uma operação de longa duração e esperar que ela seja concluída sem bloquear.

A sintaxe de uma função de suspensão é semelhante à de uma função regular, exceto pela adição da palavra – chave `suspend`. Pode receber um parâmetro e ter um tipo de retorno. No entanto, as funções de suspensão só podem ser chamadas por outra função de suspensão ou dentro de uma Coroutine.

```
suspend fun backgroundTask(param: Int): Int {  
    // long running operation  
}
```

Por dentro, as funções de suspensão são convertidas pelo compilador em outra função sem a palavra-chave `suspend`, que recebe um parâmetro de adição do tipo `Continuation<T>`. A função acima, por exemplo, será convertida pelo compilador para esta:

```
fun backgroundTask(param: Int, callback: Continuation<Int>): Int {  
    // long running operation  
}
```

`Continuation<T>` é uma interface que contém duas funções que são chamadas para retomar a coroutine com um valor de retorno ou com uma exceção se um erro ocorreu enquanto a função estava suspensa.

```
interface Continuation<in T> {  
    val context: CoroutineContext  
    fun resume(value: T)  
    fun resumeWithException(exception: Throwable)  
}
```

Por hoje é só pessoal, sucesso nos estudos.