Multi-Threading with GCD/NSOperations

The main queue is a serial queue globally available and supplied by GCD (Grand Central Dispatch) that executes tasks in the app main thread. We should be careful to execute tasks in this queue, because we don't want to block the main thread as it is used to update the UI, whose operation consumes many resources. Even one fraction of a second delay can affect the smoothness of the UI. In order to avoid this, alongside with the main queue, the GCD gives us global queues that are used to perform tasks that don't belong to the main thread, that is, anything that doesn't update the UI.

Async/Await

The use of sync gives control back to the caller only after the execution is finished, on the other hand, async initiates and gives back control immediately, without blocking the thread. Usually, we use async for API calls or to execute a task with high CPU requirements. Furthermore, we can use async in the main thread, or global queues based on the project.

We know GCD is a low level API that allows multi-threading and operation queues are an abstraction built on top of GCD. They allow us to add priorities and dependencies between the tasks. Therefore, dispatch queues are purely FIFO and operational queues are not. Besides, dispatch queues can be serial or concurrent and operational queues are always concurrent. We can define certain sequences and dependencies, but they will never be purely sequential.

Finally, an async function is different from the others because it can be paused during execution and wait until the end of an operation. It is indicated by the keyword async right after its parameters and the keyword await indicates where it can be paused. When this code line is executed, it is sent to another thread and the execution in the main thread is carried on until that value is necessary.

Mauro Arantes