



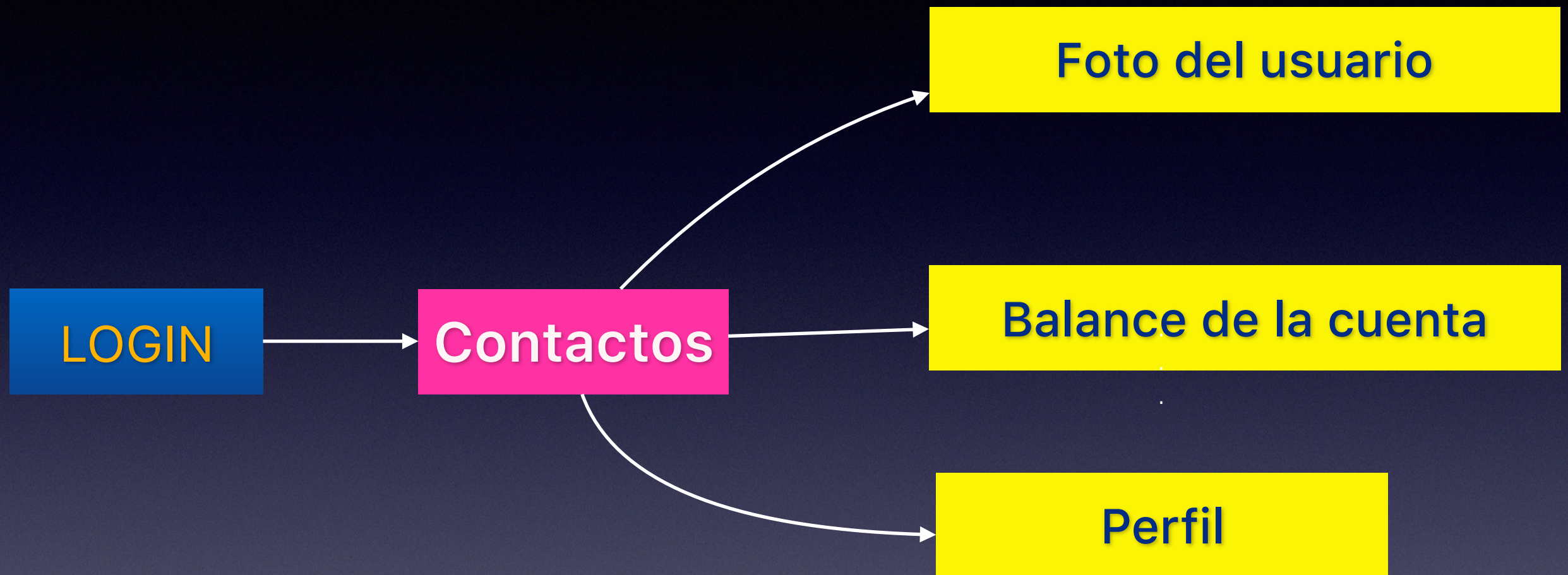
NS0perations a fondo

Nicolas Purita
@nicopuri

Agenda

- Introducción
- Conceptos Básicos
- **NSOperations** a **FONDO**
- Código de ejemplo

Introducción



- Múltiples Request en el Hilo principal.
- Mala performance de la app.
- Bloqueo de la UI.

Classes Principales

```
@available(iOS 2.0, *)  
public class NSOperationQueue : NSObject {  
    ...  
}
```

```
@available(iOS 2.0, *)  
public class NSOperation : NSObject {  
    ...  
}
```

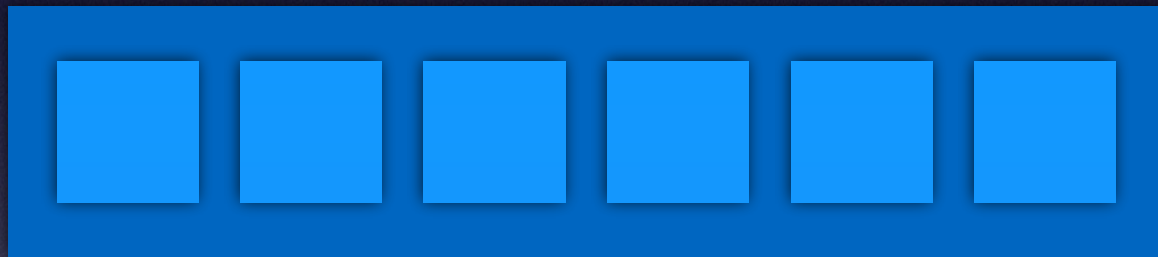

NSOperationQueue

- Alto nivel de manejo sobre **dispatch_queue_t**
- Contenedor de todas las operaciones.
- Prioridades.
- Dependencia entre operaciones.

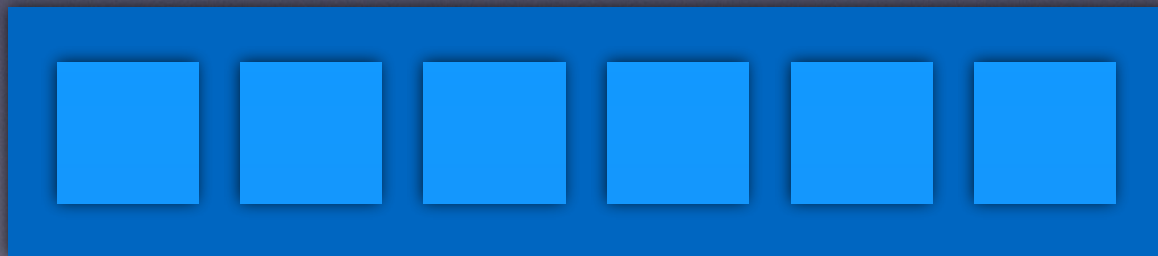
Tipos de colas

```
public var maxConcurrentOperationCount Int
```

- Serial



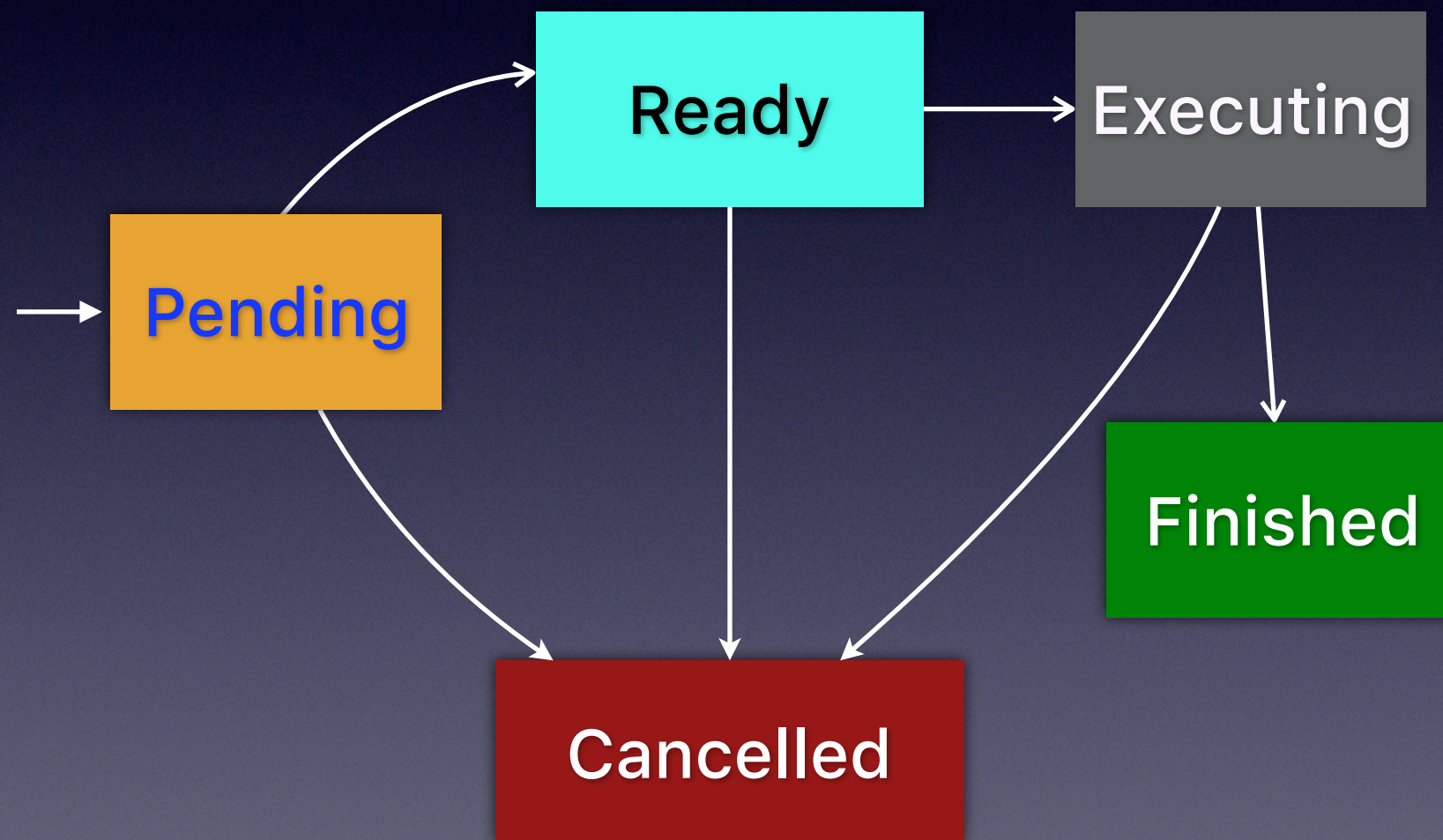
- Concurrent



NSOperation

- Clase Abstracta.
- Contenedor de ejecución de una tarea.
 - `NSInvocationOperation`
 - `NSBlockOperation`

Estados



- isCancelled
- isAsynchronous
- isExecuting
- isFinished
- isReady
- dependencies
- queuePriority
- completionBlock

Dependencias

- Orden de ejecución de las operaciones.
- Una operación se ejecuta siempre que todas sus dependencias hayan finalizado.
- Como lo hago?

```
let operationA = ...  
let operationB = ...  
operationB.addDependency(operationA)
```


Sync vs Async

- Sincrónicas por defecto.
- Se ejecuta en el hilo actual.
- Las asincrónicas retornan inmediatamente luego de llamar al start.
- Las **NSOperationQueue** ignora la propiedad “asynchronous”.

Agenda

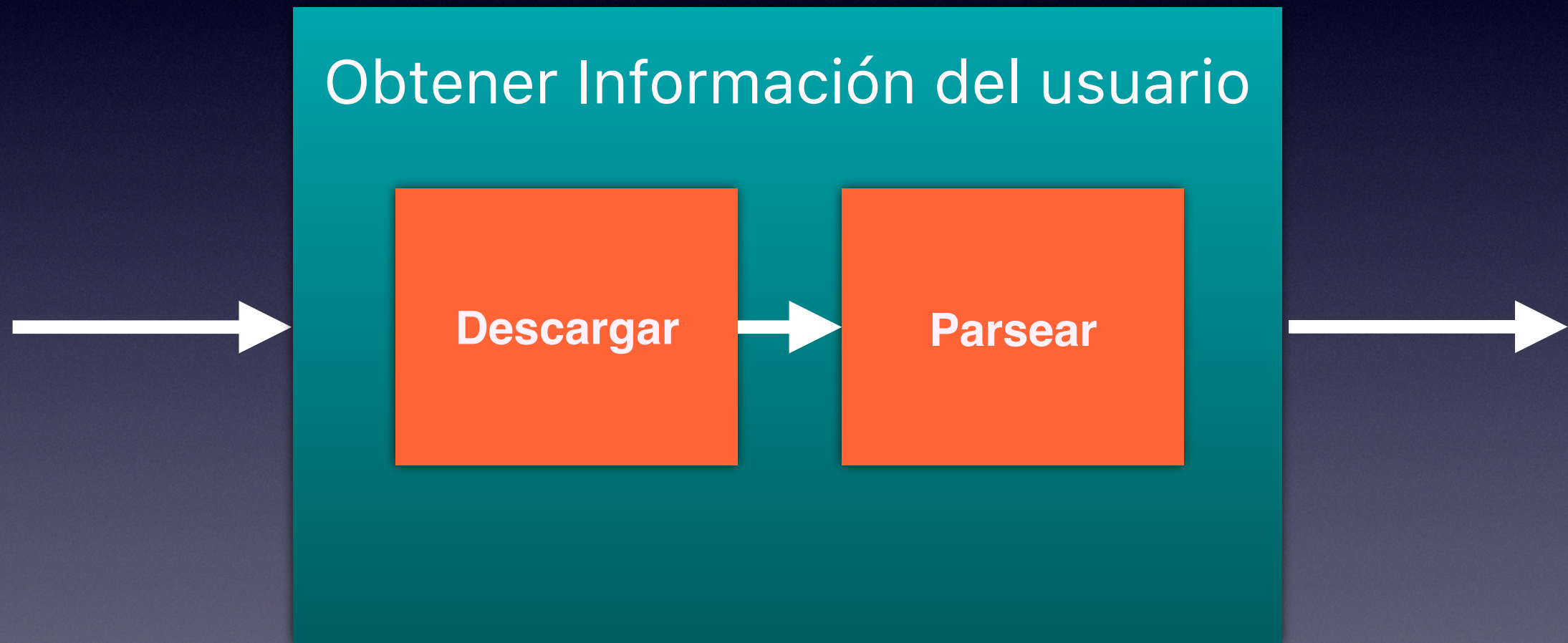
- ✓ Introducción
- ✓ Conceptos Básicos
- **NS0perations** a **FONDO**
- Código de ejemplo



Extensiones a **NSOperations**

- Composición de operaciones
- Condiciones
- Observadores
- Mutuamente excluyentes

Composición



Condiciones

```
protocol OperationCondition {  
    static var name: String { get }  
  
    static var isMutuallyExclusive: Bool { get }  
  
    func dependencyForOperation(operation: Operation) ->  
    NSOperation?  
  
    func evaluateForOperation(operation: Operation,  
    completion: OperationConditionResult -> Void)  
}}
```


Observadores

```
protocol OperationObserver {  
    func operationDidStart(operation: Operation)  
  
    func operation(operation: Operation,  
didProduceOperation newOperation: NSOperation)  
  
    func operationDidFinish(operation: Operation,  
errors: [NSError])  
}
```


Demo



Referencias

- 226 - Advanced NSOperations-HD (WWDC 2015)
- NSOperation Class Reference
- NSOperationQueue Class Reference



Preguntas?



Muchas Gracias!!