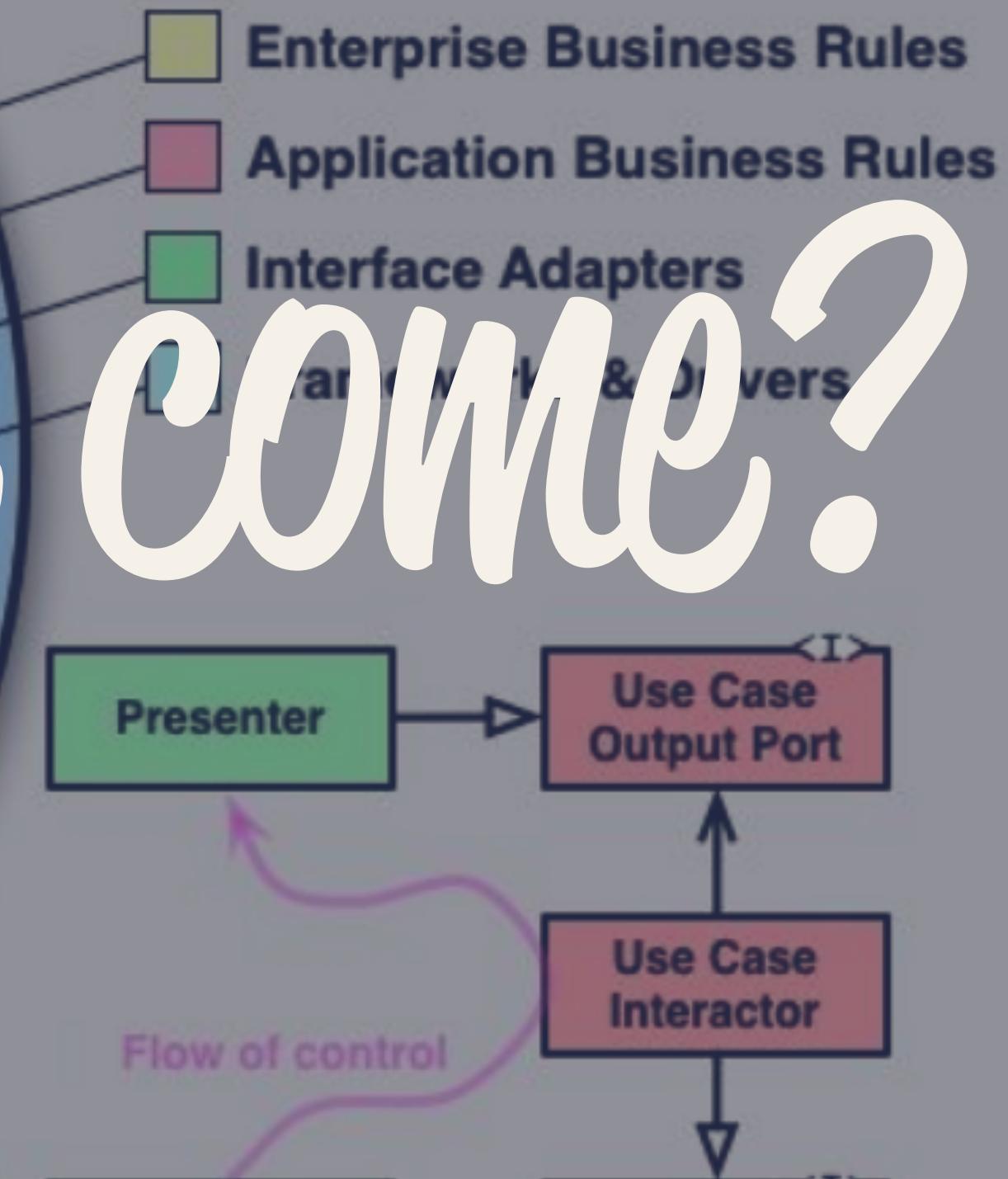
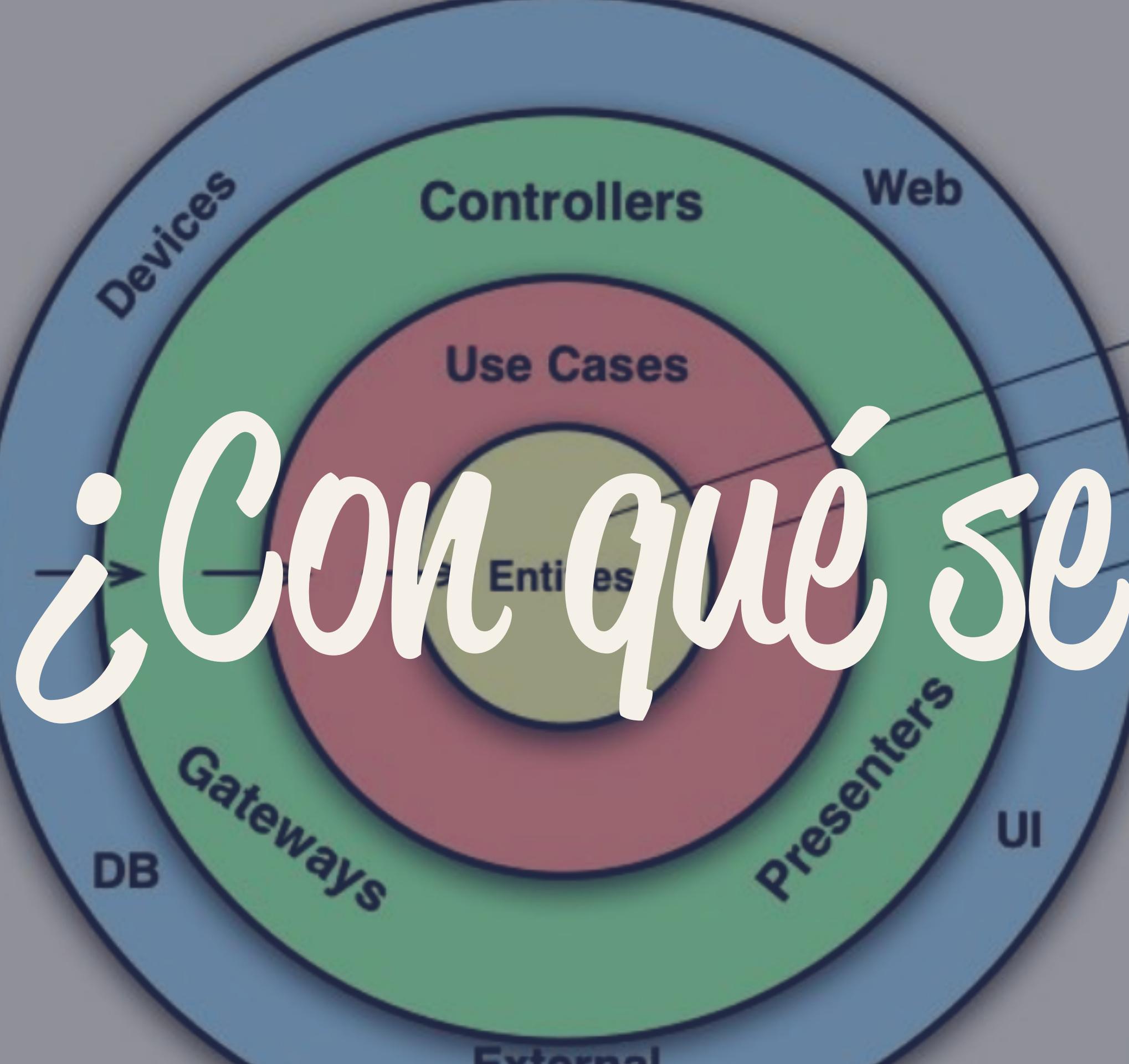
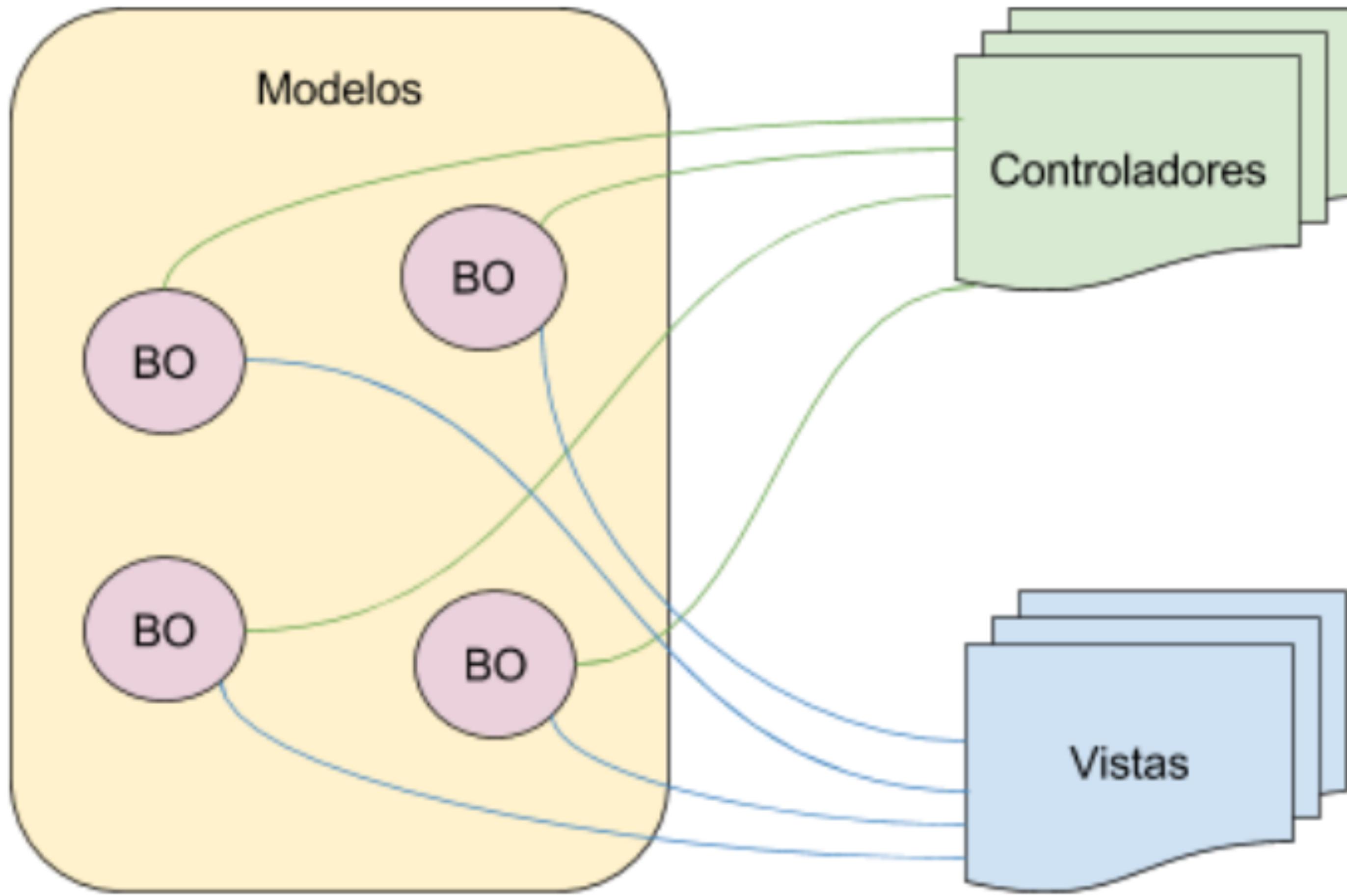


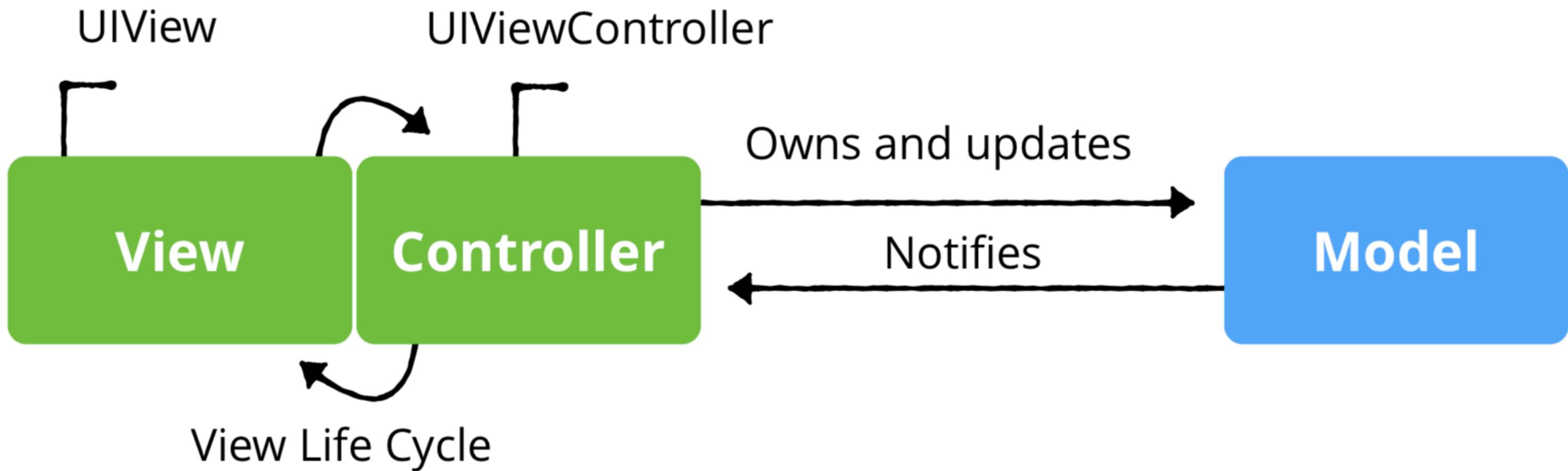
# CLEAN ARCHITECTURE

# ¡Con qué se come?



**¿POR QUÉ USAR CLEAN?**





- ▶ Independiente de frameworks
- ▶ Testable
- ▶ Independiente de la UI...
- ▶ Independiente de la base de datos...

**NO LE DEMOS  
VAMOS A VERLO DE UNA FORMA SENCILLA**

**HERRAMIENTAS PARA APRENDER CLEAN  
QUE ME HAN  
SERVIDO**



<https://github.com/edittler/clean-architecture>

<http://xurxodev.com/por-que-utilizo-clean-architecture-en-mis-proyectos/>

<https://www.raywenderlich.com/5161-clean-architecture-on-ios>

Advanced iOS app Architecture Ray Wenderlich

<https://clean-swift.com/about/>

# ORGANIZACIÓN DE PROYECTOS MÓVILES



# ate.swift yboard CONFINE ARCHIVOS DE LAS ENTIDADES DE LA LÓGICA DE NEGOCIO

# DATA SERVICES

IMPLEMENTACIÓN DE LOS WORKERS Y SERVICIOS DE LA CAPA DE DATOS

```
1 //  
2 // CourseModels.swift  
3 // FIUBA  
4 //  
5  
6 import Foundation  
7  
8 struct CourseModel {  
9     var id: String?  
10    var title: String?  
11    var professor: Instructor?  
12    var teachers: [Instructor]  
13    var vacancies: Int?  
14    var enrolled: Int?  
15 }  
16 }
```



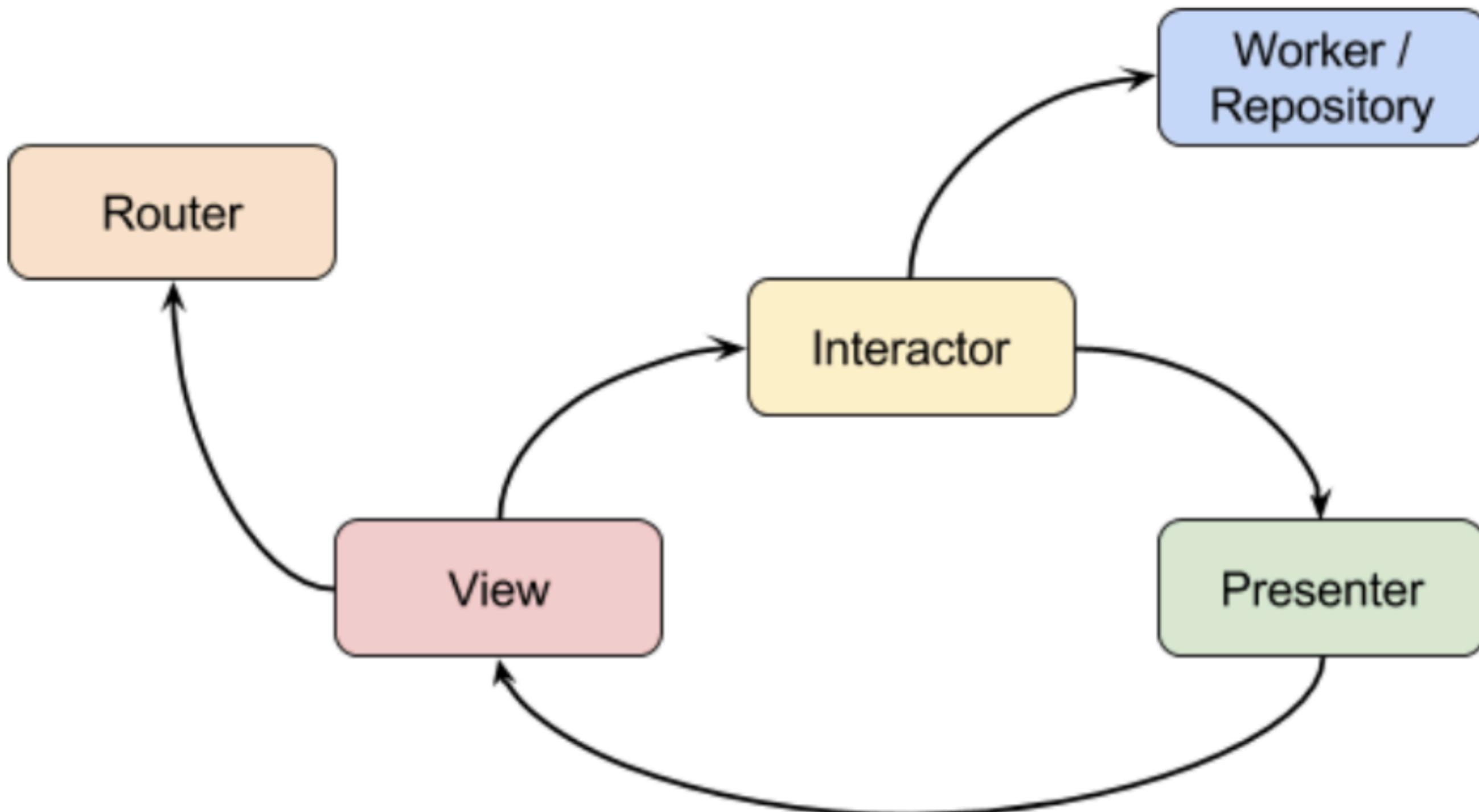
# HERRAMIENTAS PARA LA PERSISTENCIA DE DATOS ENTIDADES DE DOMINIO EN UNA BASE DE DATOS

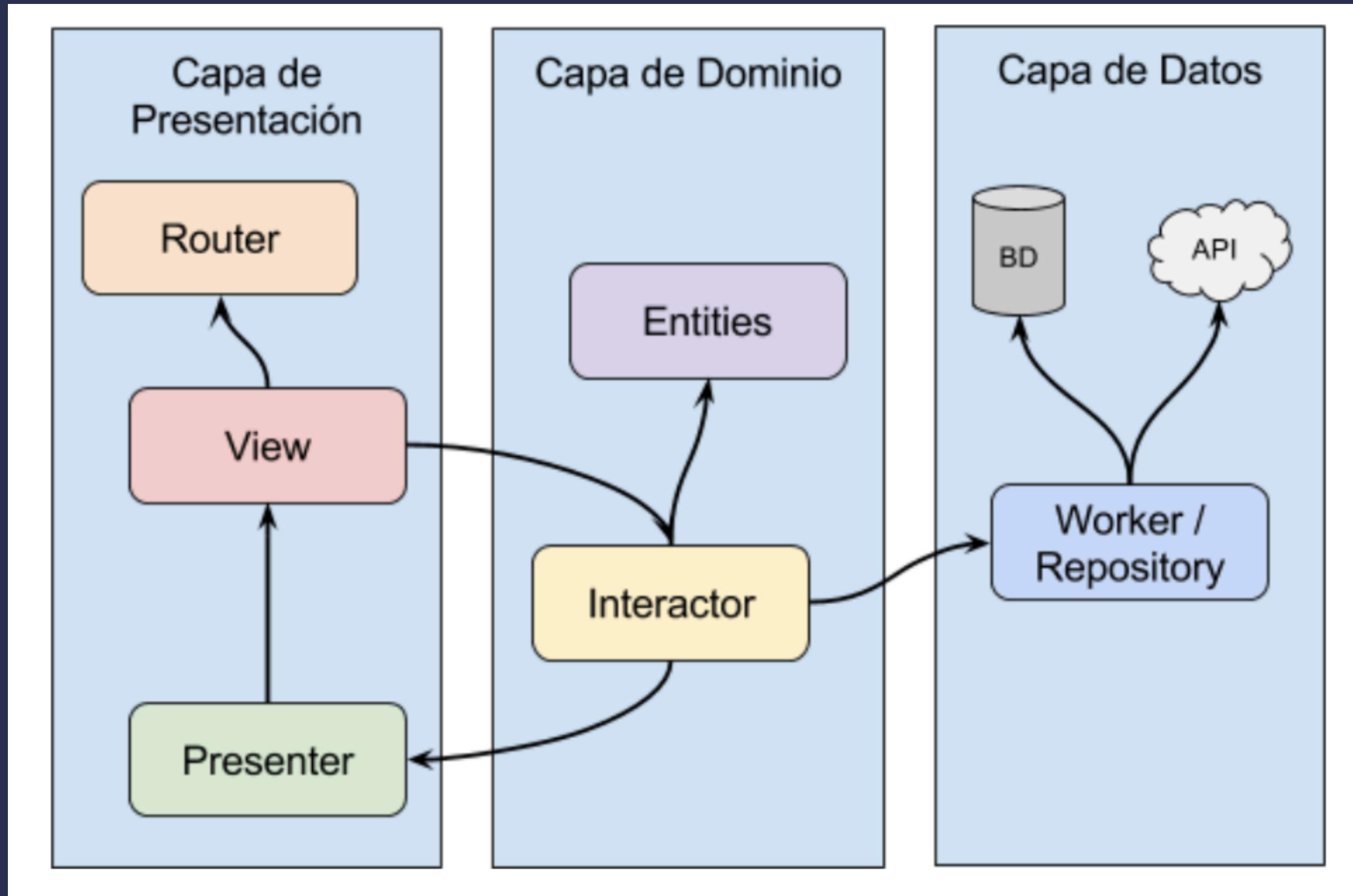
# HERRAMIENTAS Y ELEMENTOS PARA CONECTARSE A UNA API O CUALQUIER RECURSO DE INTERNET

```
1 //  
2 // DataManager.swift  
2 // TopApps  
3 //  
4 //  
5 //  
6 //  
7 //  
8 //  
9 import Foundation  
10 open class DataManager {  
11     open func loadJson(_ fileName: String, completion:@escaping (_ data: Data?, _ e  
12         let filePath = Bundle.main.path(forResource: fileName, ofType: "json")!  
13         let fileURL = URL(fileURLWithPath: filePath)  
14         let contents = try! Data(contentsOf: fileURL)  
15         success(contents)  
16     }  
17     let data = try! Data(contentsOf: URL(fileURLWithPath: filePath!),  
18     options: NSData.ReadingOptions.unCached)  
19     success(data)  
20 }  
21  
22  
23     open class func loadDataFromURL(_ url: URL, completion:@escaping (_ data: Data?, _ e  
24         let session = URLSession.shared  
25 }
```

# SISTEMAS CENTRALES

ELEMENTOS RELACIONADOS CON LA CAPA DE PRESENTACIÓN





# INTERFACTOR

ELEMENTO QUE CONTIENE LAS REGLAS DE NEGOCIO

# PRESENTER

ELEMENTO QUE ADAPTA LOS MODELOS DEL DOMINIO PARA PRESENTARLOS A LA VISTA

# ELEMENTOS DE LA VISTA



# WORKER

ELEMENTO QUE OBTIENE LOS DATOS QUE SOLICITA EL INTERACTOR

# ROUTER

ELEMENTO QUE CONTIENE LA LÓGICA DE NAVEGACIÓN ENTRE LAS VISTAS, REALIZANDO CONFIGURACIONES COMO PASAR DATOS, ESTABLECER VALORES NECESARIOS ETC

