LAB - Consumir servicios REST



Objetivos

* Consumir servicios API Rest desde iOS
* Configurar permisos info.plist
* Uso de Librerías swiftJSON Alomafire

Preparar proyecto base:

Abrir el proyecto

Install Node JS:

<https://nodejs.org/en/download/>

Version of Swift

xcrun swift --version

URL Alomafire

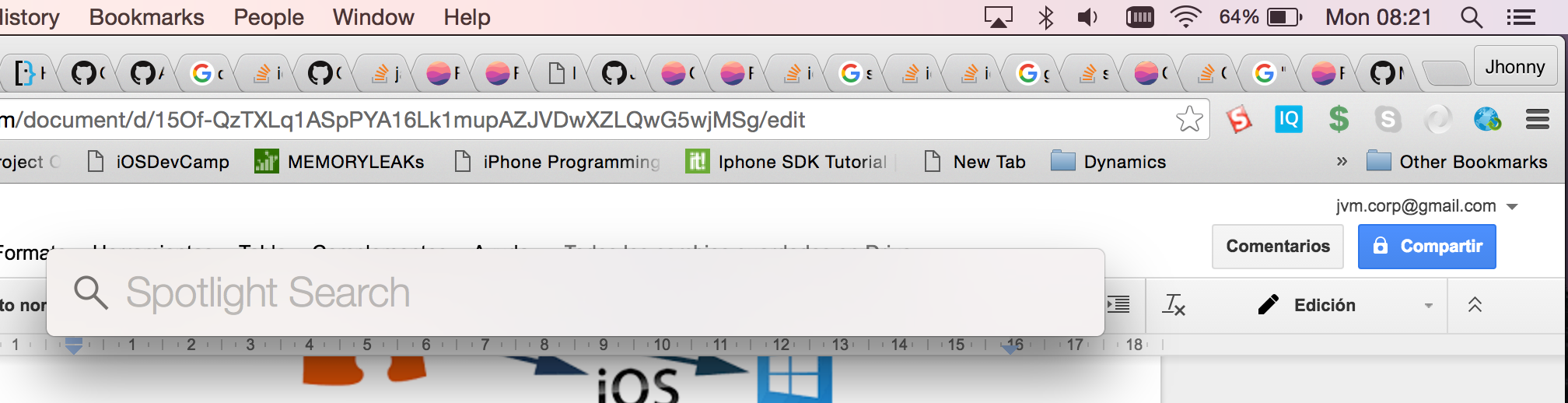
<https://github.com/Alamofire/Alamofire/tree/3.0.1>

URL SwiftJSON

<https://github.com/SwiftyJSON/SwiftyJSON>

## Crear Cartfile

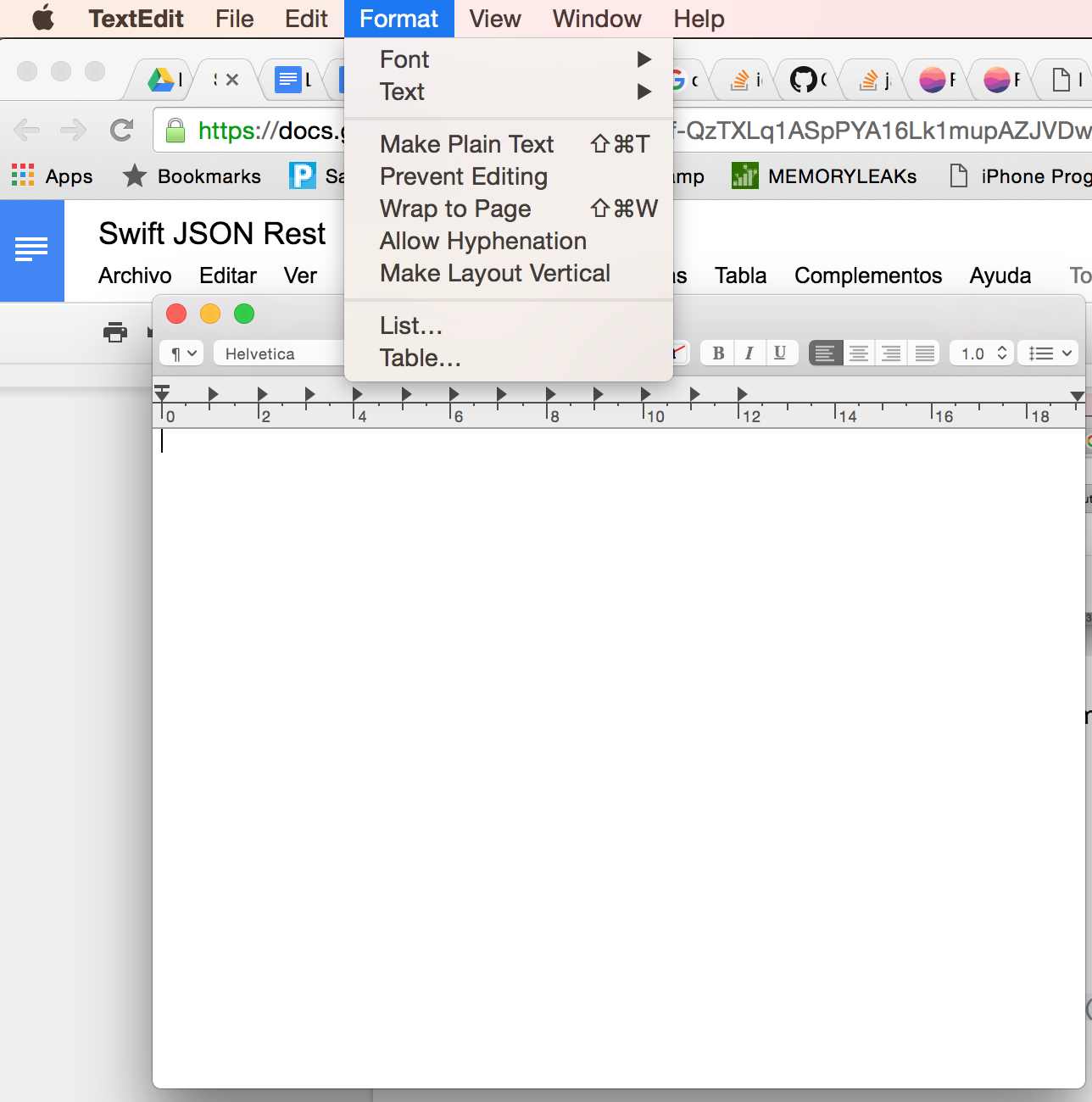
Click en el icono de búsqueda esquina superior derecha



En la barra de busqueda escribir : “textEdit” esta mostrar un diálogo con el programa textEditor.

Configurar el editor de texto para crear archivos de texto Plano:

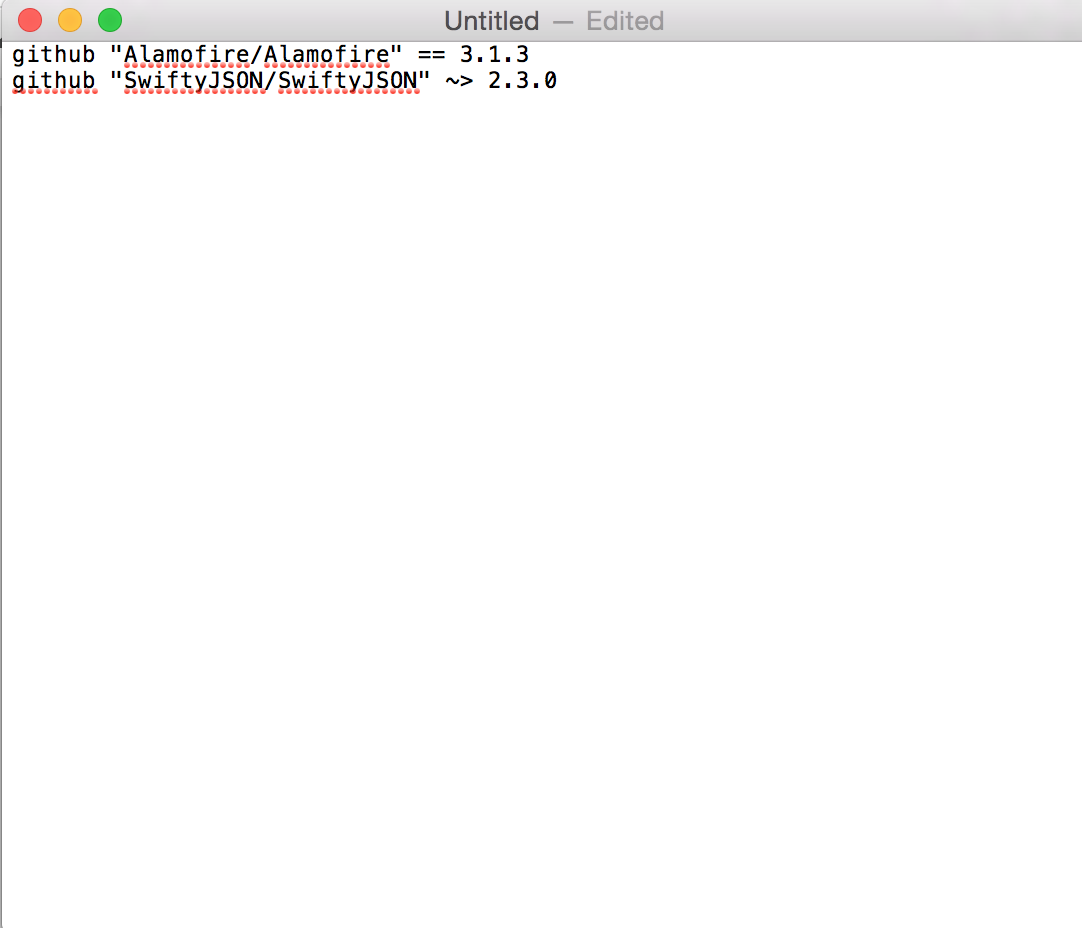
* En el menú del editor seleccionar la opción Format > Make Text Plain



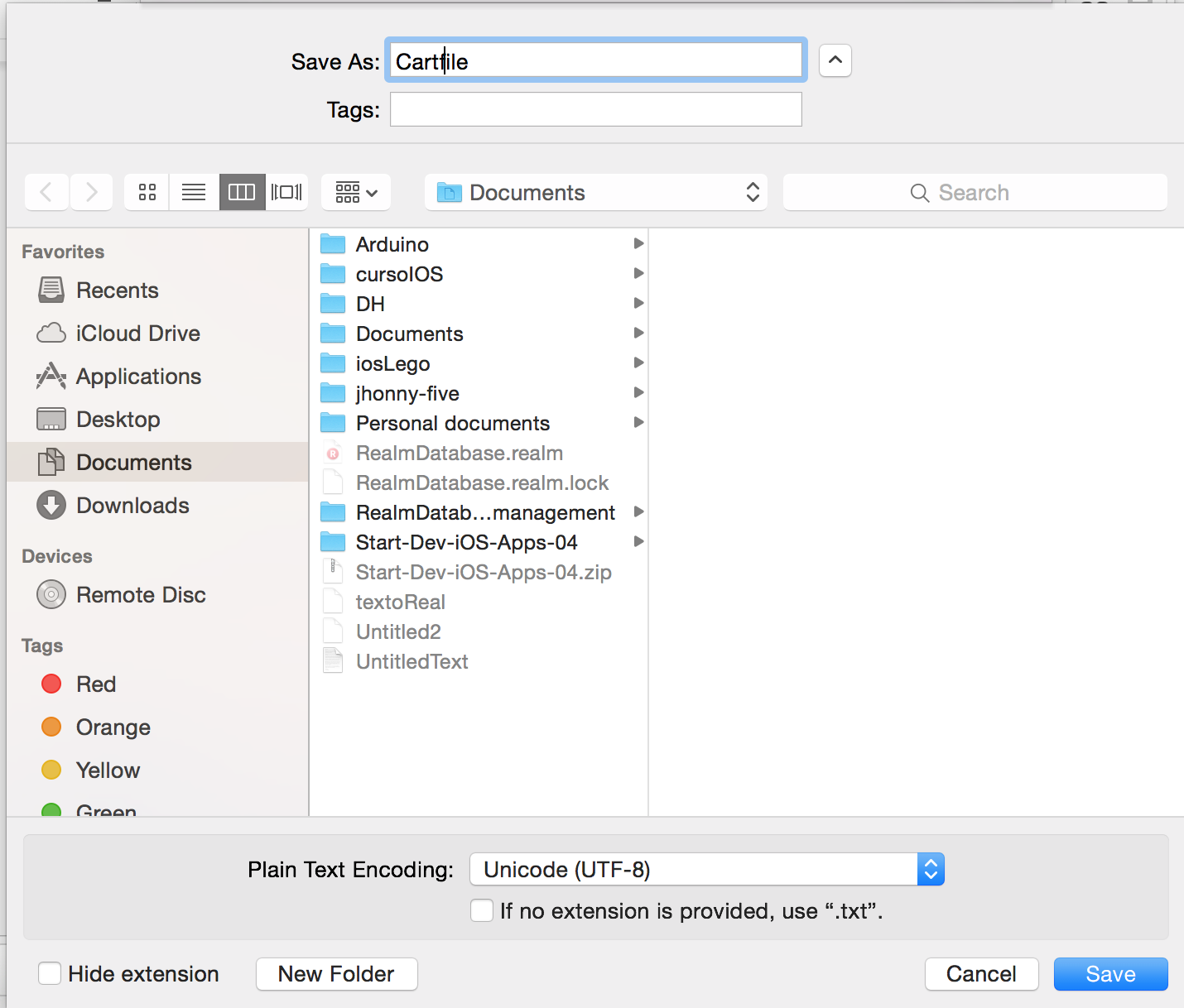
En el Editor de escribir la siguiente informacion:

github "Alamofire/Alamofire" == 3.1.3

github "SwiftyJSON/SwiftyJSON" ~> 2.3.0

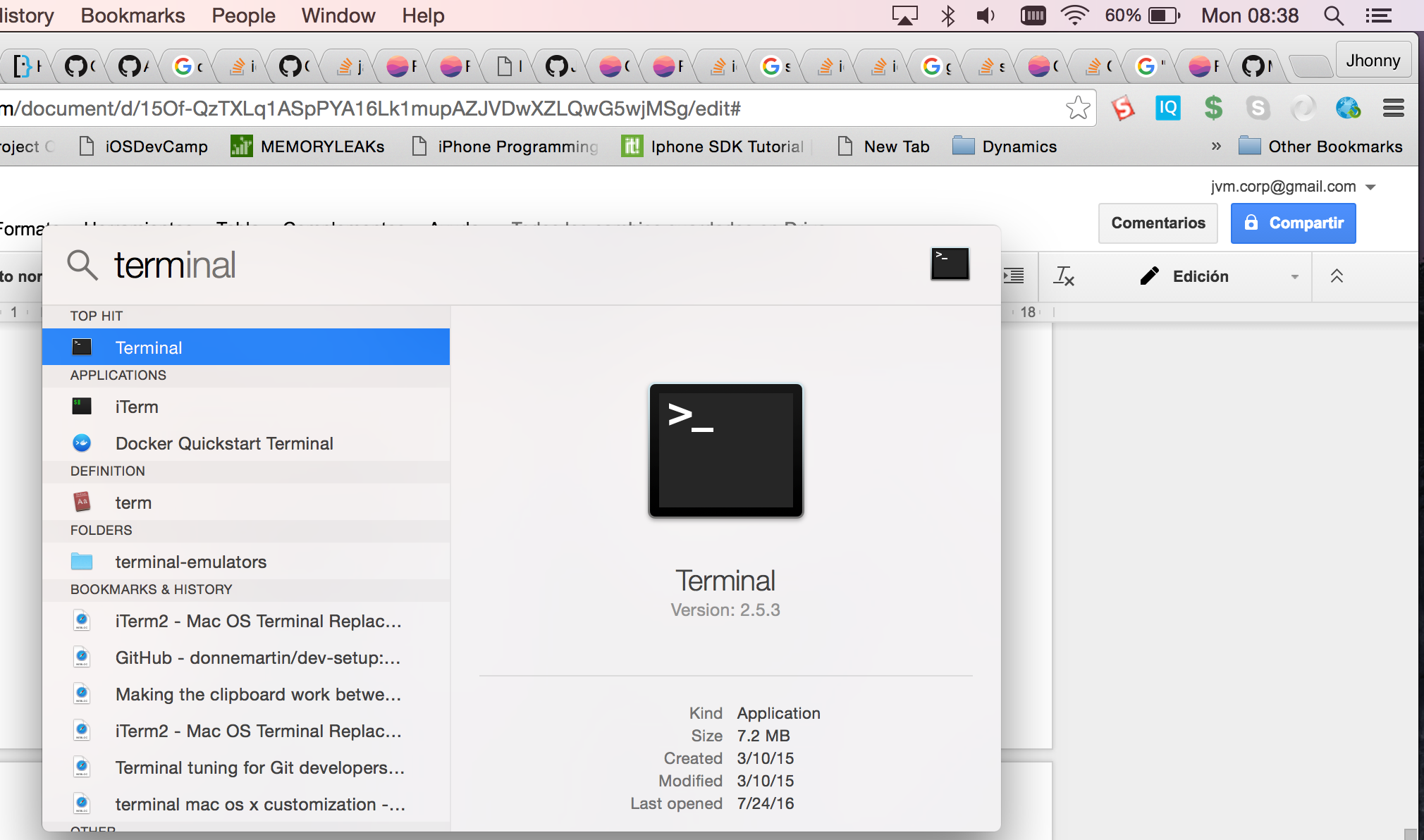


Guardar el archivo con el nombre “Cartfile” y destiquear la opción: “If no extension is provided use .“txt”



Abrir la consola en la raíz del proyecto

* Abrir el programa de consola, click en búsqueda (esquina superior derecha) escribir terminal en la barra de búsqueda:



direccionar la consola a la raíz de su proyecto:

$ cd <path of your projecto>

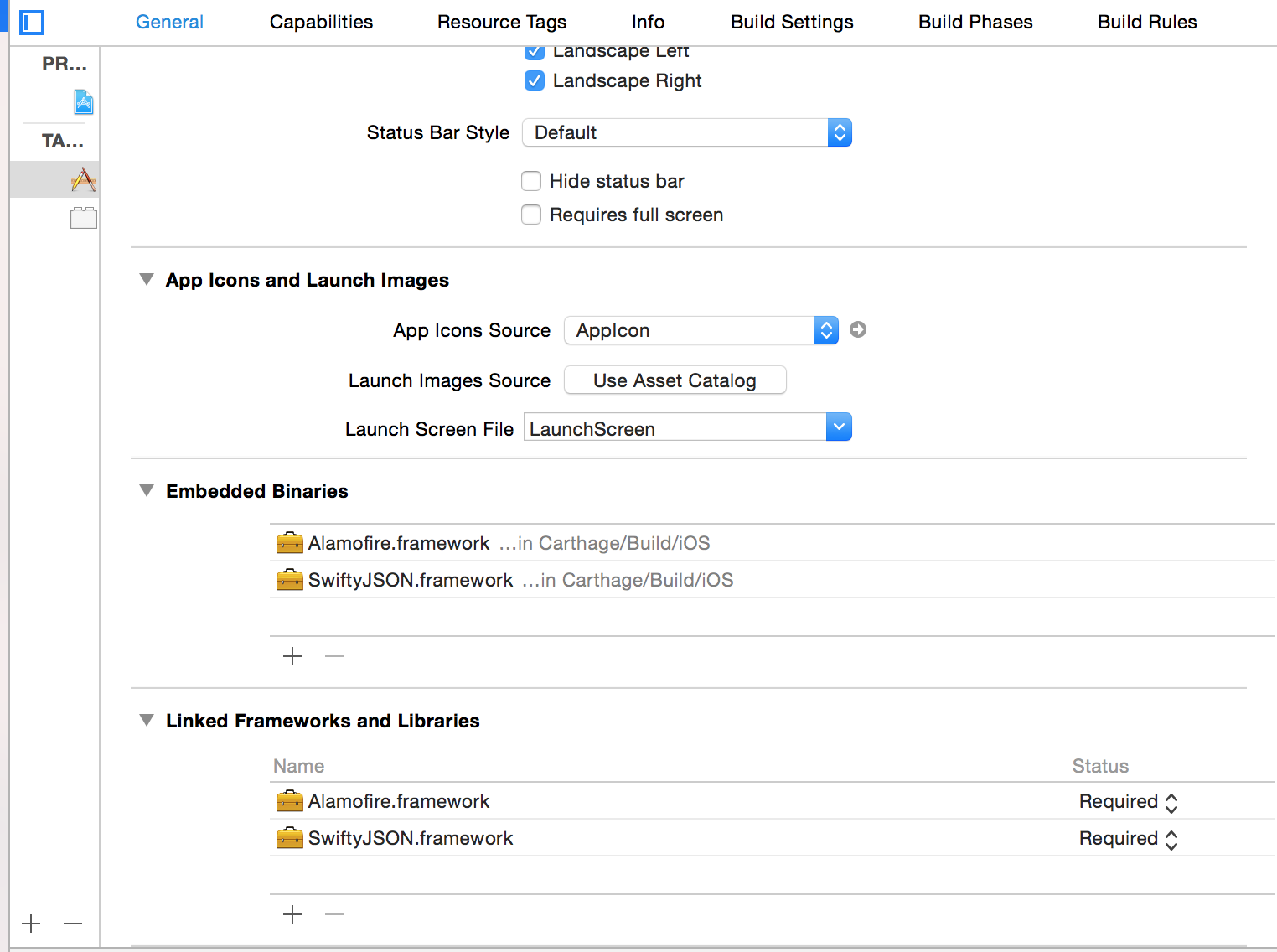
## Build Cartfile

Una vez guardado su **Cartfile** escribir en la consola:

$ carthage update --platform iOS

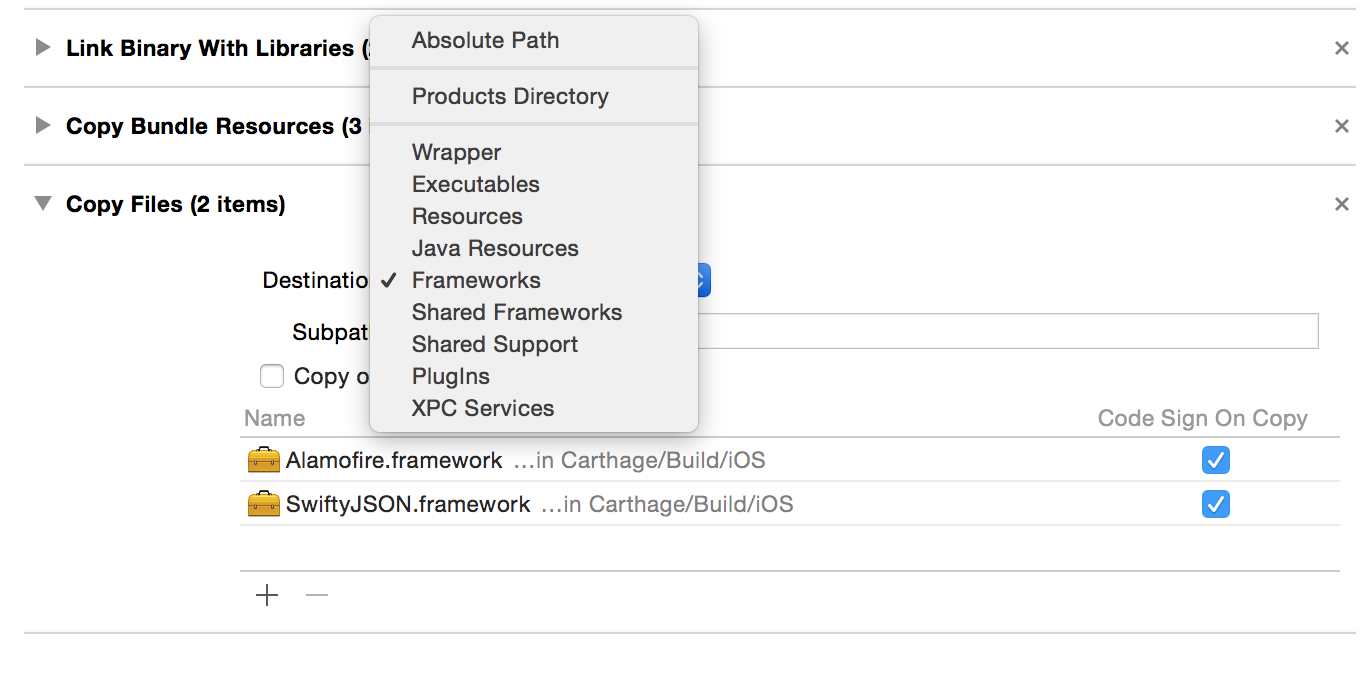
## Incluir librerías de Carthage al proyecto

Click en la propiedades del proyecto, “General” ahi agregar en **Linkend Framerworks and Libraries,** en el diálogo seleccione Other y buscar la librerías de **Alomafire y swiftJSON** en la carpeta **IOS**:



Agregar la librerías en Build Phases

* Seleccionar el tab “Build Phases” presionar en el botón (+)
* Seleccionar la opción “New Copy Files Phase”
* En el combo Destination seleccionar la opción Frameworks y agregar las dos librerias “Alomfire”, “SwiftyJSON”



## Test de coneccion a REST/API

En el metodo “loadSampleMeals” de la clase “MealTableViewController”

// Import the library

import Alamofire

// populate using Rest

Alamofire.request(.GET, "http://localhost:3000/meals")

.responseJSON { response in

print(response.request) // original URL request

print(response.response) // URL response

print(response.data) // server data

print(response.result) // result of response serialization

if let JSON = response.result.value {

print("JSON: \(JSON)")

}

}

Agregar la siguiente directiva para consumir rest que no tienen HTTPS configurado

* Abrir info.plist como codigo y agregar las siguientes lineas

<key>NSAppTransportSecurity</key>

<dict> <!--Include to allow all connections (DANGER)--> <key>NSAllowsArbitraryLoads</key>

<true/>

</dict>

Cargar Meals de un rest Services

func loadSampleMeals() {

// populate using Rest

Alamofire.request(.GET, "http://localhost:3000/meals")

.responseJSON { response in

print(response.request) // original URL request

print(response.response) // URL response

print(response.data) // server data

//response.data

print(response.result) // result of response serialization

self.data = response.data

if let JSON = response.result.value {

print("JSON: \(JSON)")

}

let json = JSON(data: response.data!)

for meal in json {

print("the name of meal is \(meal)")

}

//If json is .Dictionary

for (key,subJson):(String, JSON) in json {

// add new Meal

let newMeal = Meal(name: subJson["name"].string!,

photo: nil,

rating: subJson["rating"].int!)

// Retrieve Image

Alamofire.request(.GET, subJson["photo"].string!)

.responseJSON {img in

// Create imagen with data

let imagen = UIImage(data: img.data!)

newMeal?.photo = imagen

self.meals.append(newMeal!)

// Reload data Table

self.tableView.reloadData()

}

}

}

}