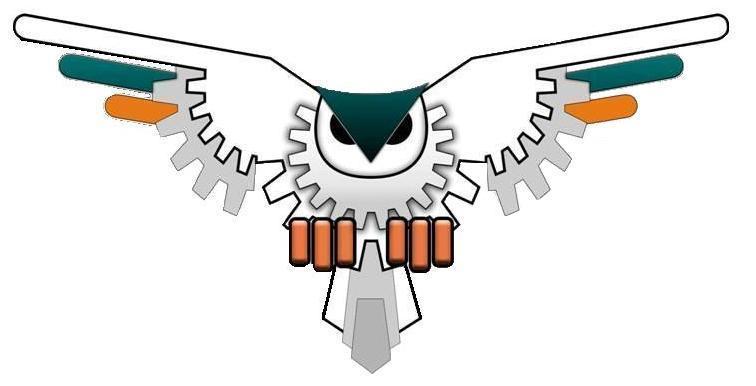
**Instituto Tecnológico Superior de Irapuato.**



**CONEBU iOS Application.**

**Integrantes del Equipo.**

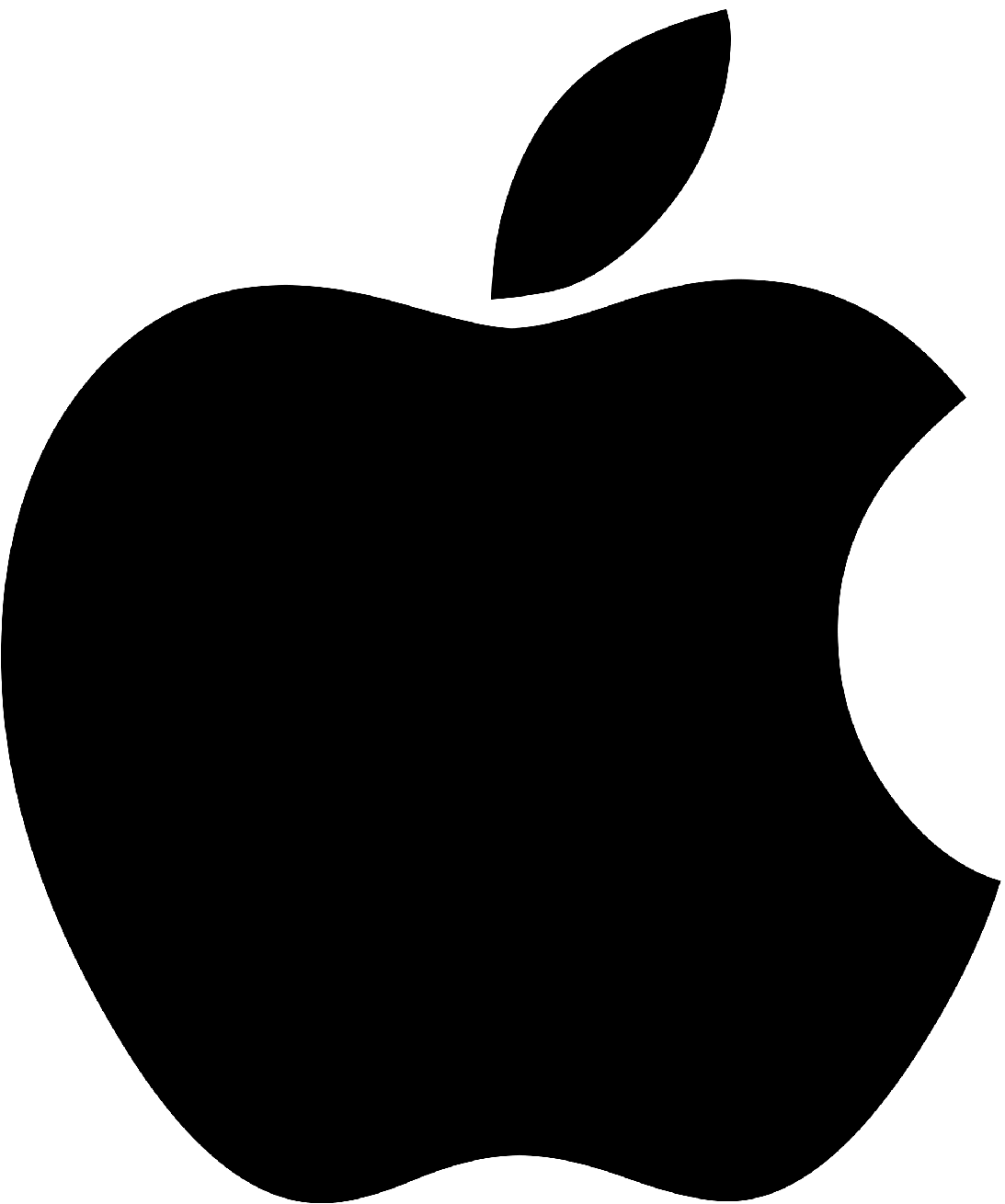
Cano Segura Antonio de Jesús.

Ramos Cabrera Francisco Alberto.

Razo Jaramillo Ángel.

Razo López Oscar Paolo.

**Ing. Néstor León Vega.**



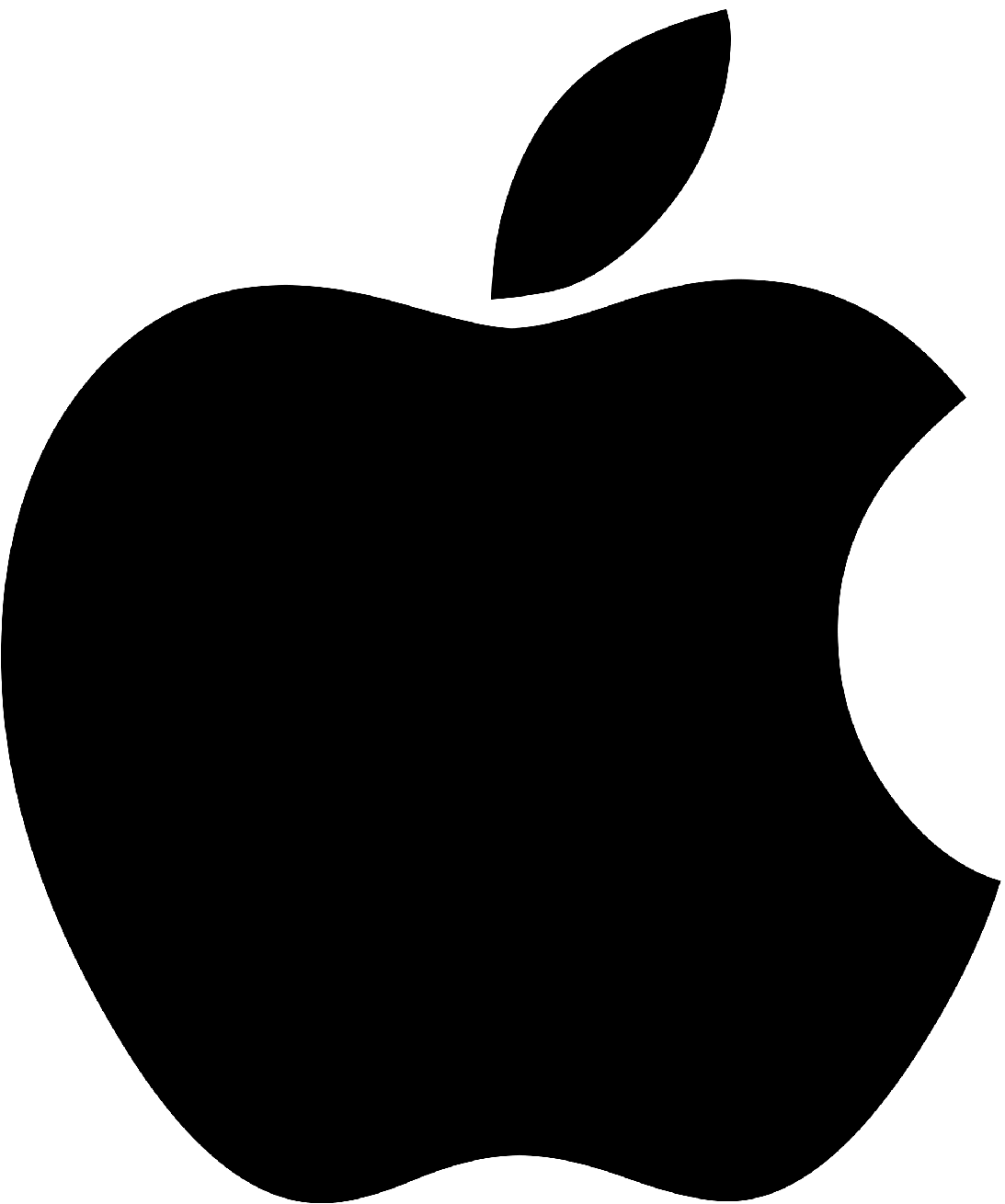
**Descripción de la Aplicación.**

La aplicación móvil esta desarrollada pensando en el usuario promedio creando interfaces intuitivas y fáciles de comprender. El lenguaje de programación en el que esta creada es Objective-C, lenguaje nativo de Apple para el desarrollo de aplicaciones móviles y de escritorio.

Al principio se pensaba en crear la aplicación en el nuevo lenguaje de programación desarrollado por Apple denominado Swift debido a su fácil entendimiento e implementación mas rápida y sencilla. Sin embargo, cambiamos al lenguaje antiguo ya que muchas de las clases y funciones que se manejan en Swift aun no son estables y no existe mucha información acerca de la solución de ciertos problemas en especifico.

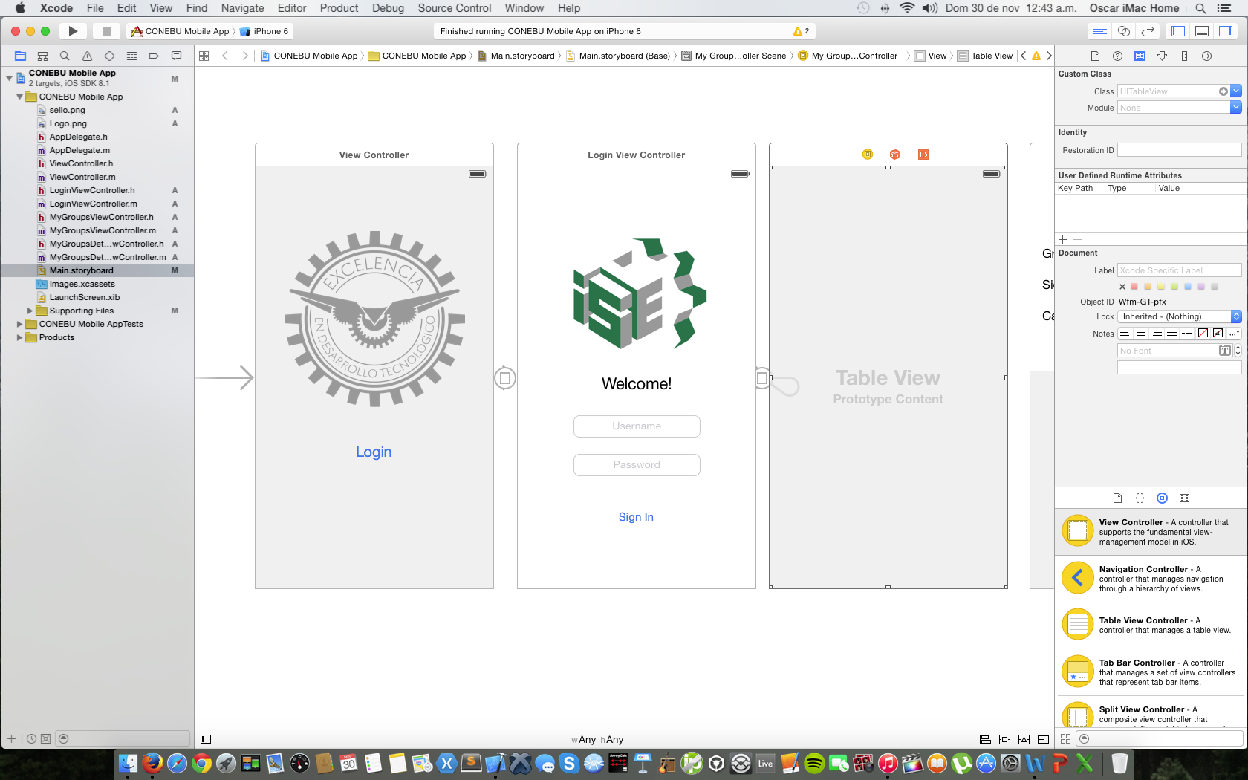
El aplicativo consumirá los servicios web y la base de datos disponibles en el servidor para la validación de Login y también para el consumo de los datos que se mostraran en el table view de *Grupos, Detalles de Grupos y Horarios de Grupos.*

El código para el consumo de los servicios web y de la base de datos vienen en un apartado al final, se implementan 17 clases distintas para la llamada de los recursos.



**Interfaces.**

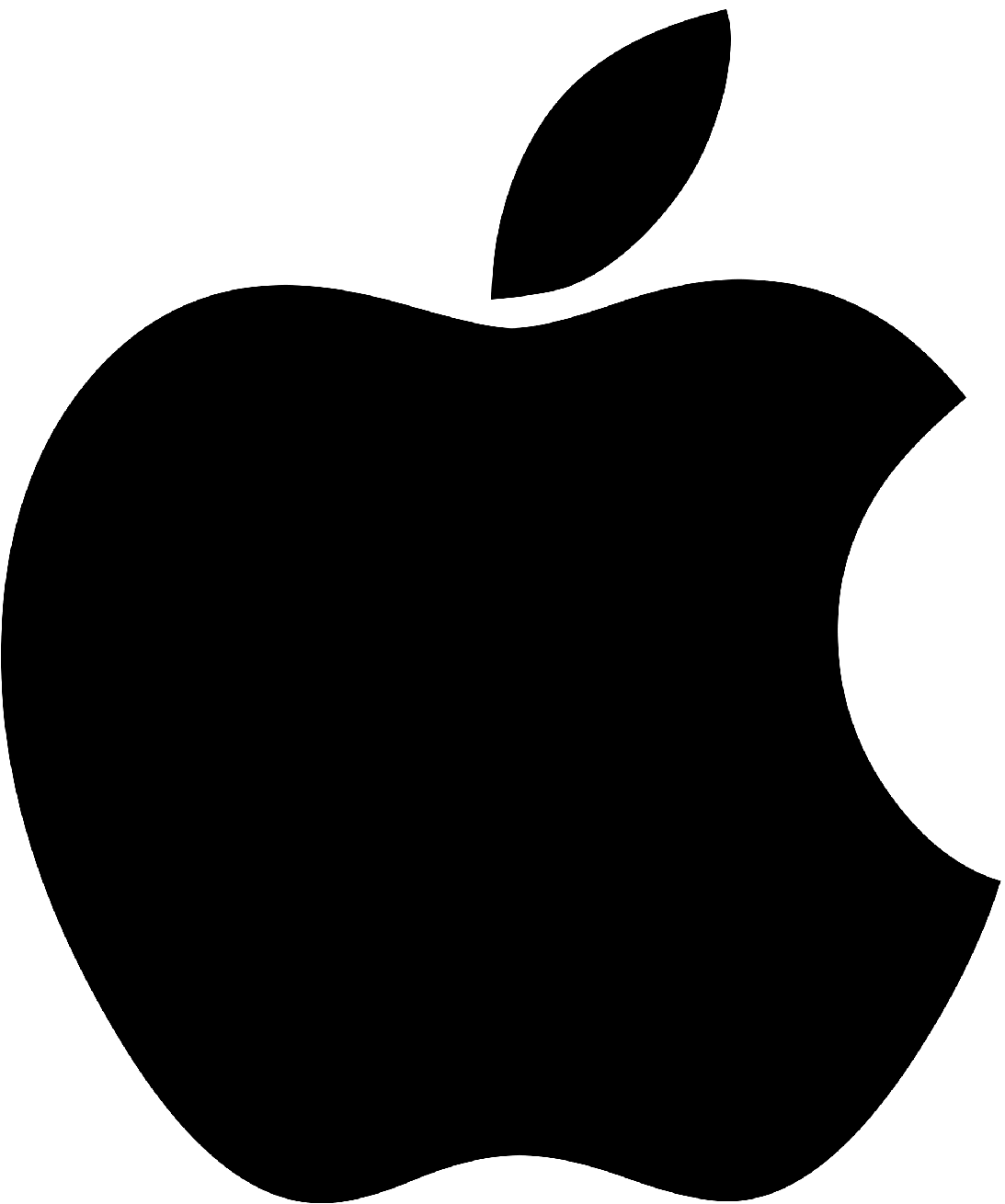
En el siguiente apartado veremos las interfaces de usuario que tiene la aplicación y también una descripción de cada interfaz.



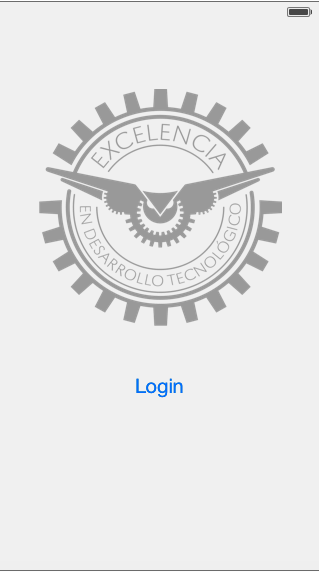
El lenguaje utilizado en la programación es Objective-C, lenguaje nativo de Apple para el desarrollo de aplicaciones para dispositivos móviles, así como también para aplicativos de escritorio.

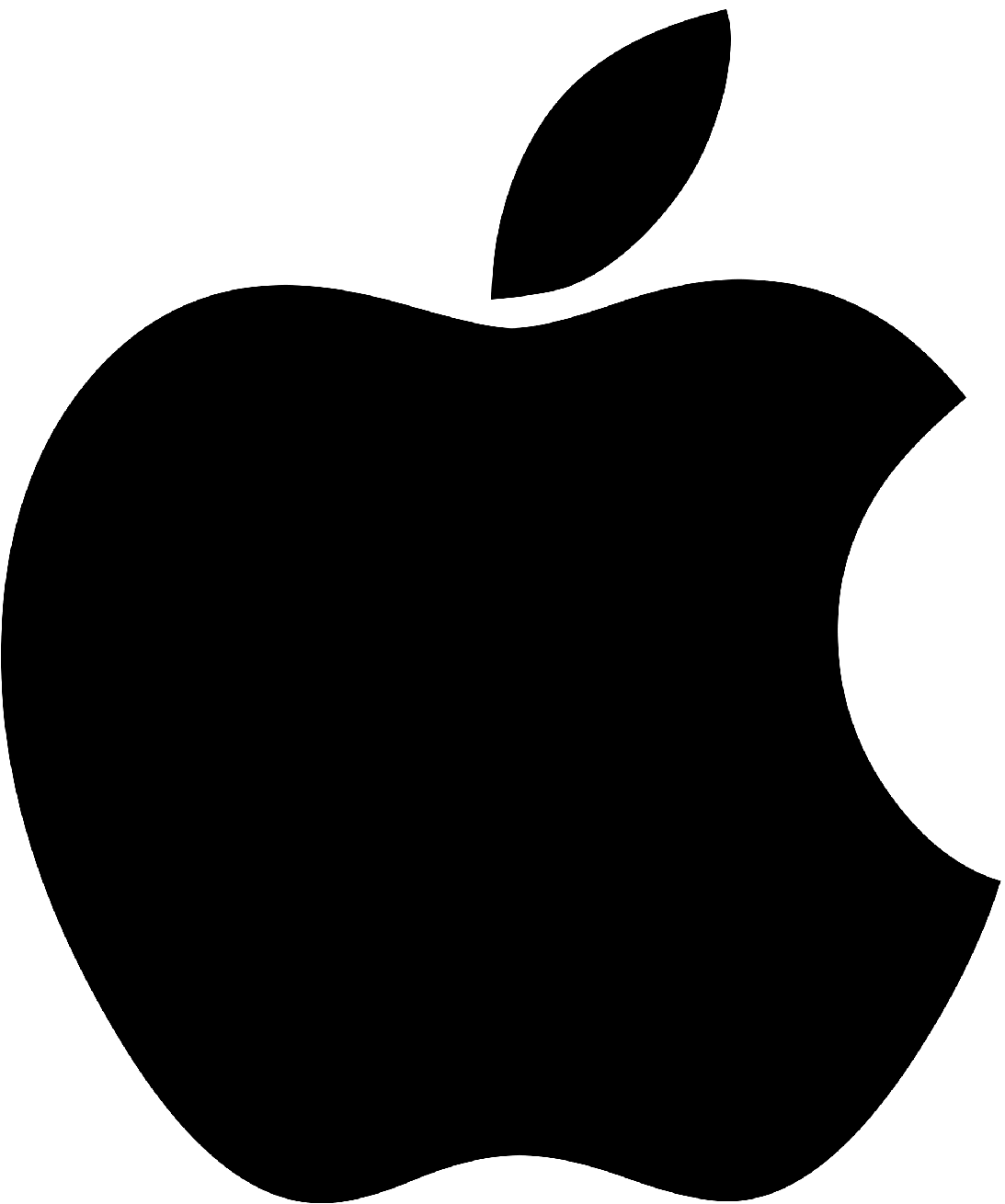
Se planeaba crear la aplicación con el nuevo lenguaje de Apple llamado Swift, pero las funciones para serialización y deserialización en Swift aún tienen un nivel importante de inestabilidad, así que optamos por el lenguaje antiguo de Apple.

Son 6 el numero total de interfaces que se manejaran en la aplicación, dentro de las cuales se encuentran las interfaces de *Bienvenida, Login, Menu, Grupos, Detalle de Grupos y Horarios.*



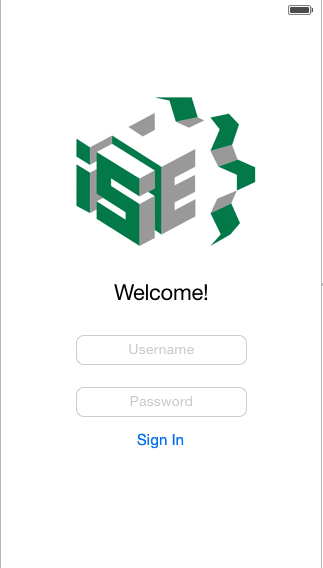
* ***Interfaz de Bienvenida.*** Esta interfaz sólo contiene un botón con el cual se accede a la ventana de Login. No contiene ninguna clase asociada con el manejo de datos en la aplicación para su funcionamiento .

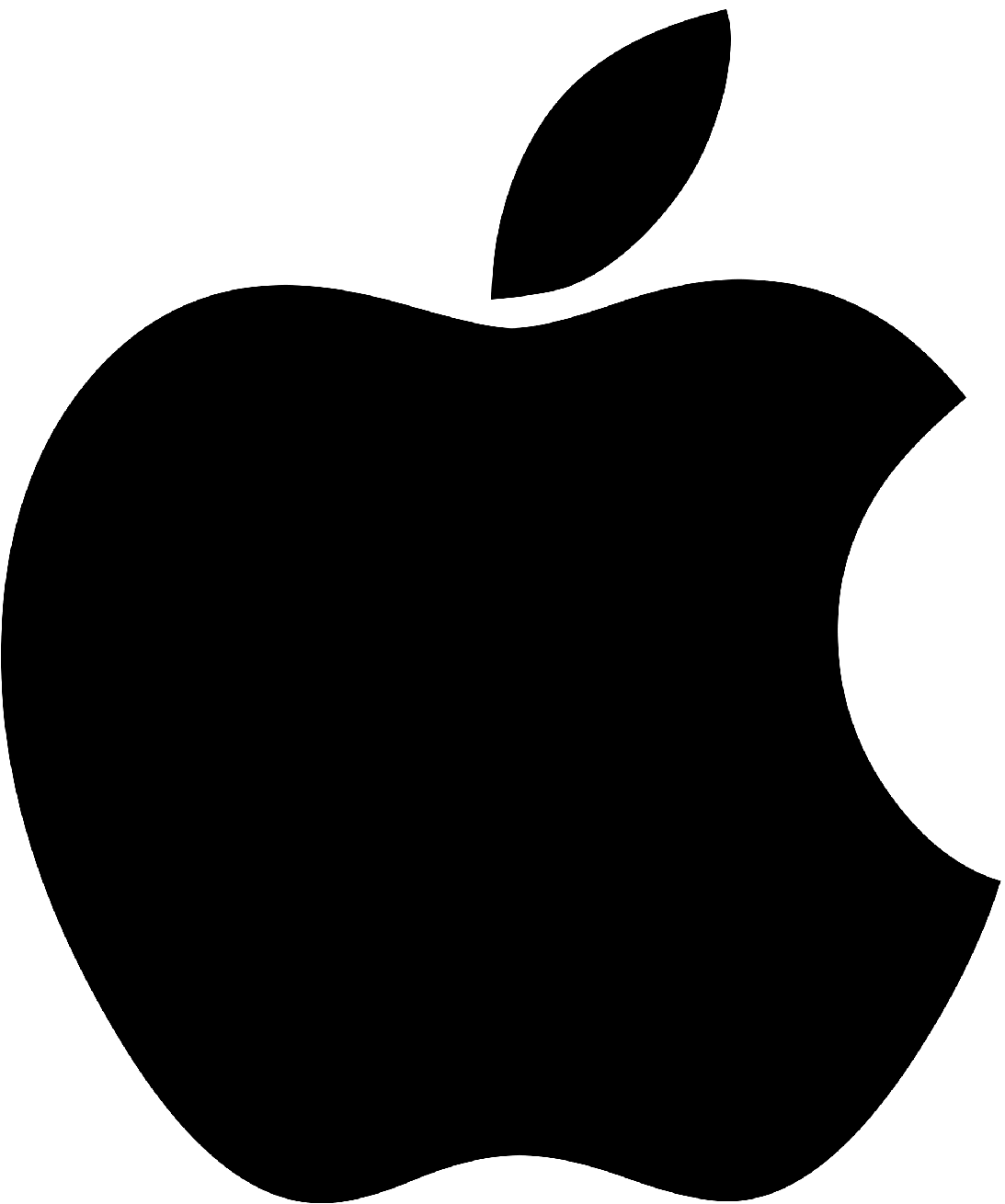




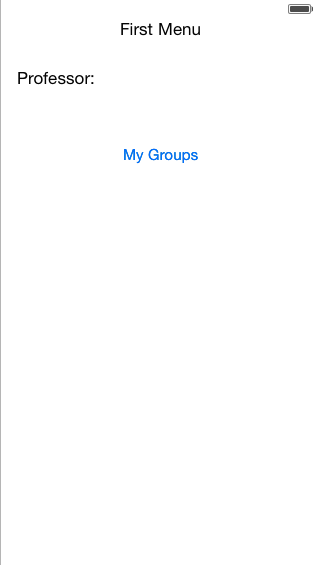
* ***Interfaz de Login.*** Esta interfaz tiene dos cajas de texto, en estas el usuario ingresará su nombre y su contraseña. Esta sección está codificada hacia la base de datos de la aplicación, esto con el fin de obtener y validar los datos del usuario ,como nombre y contraseña.

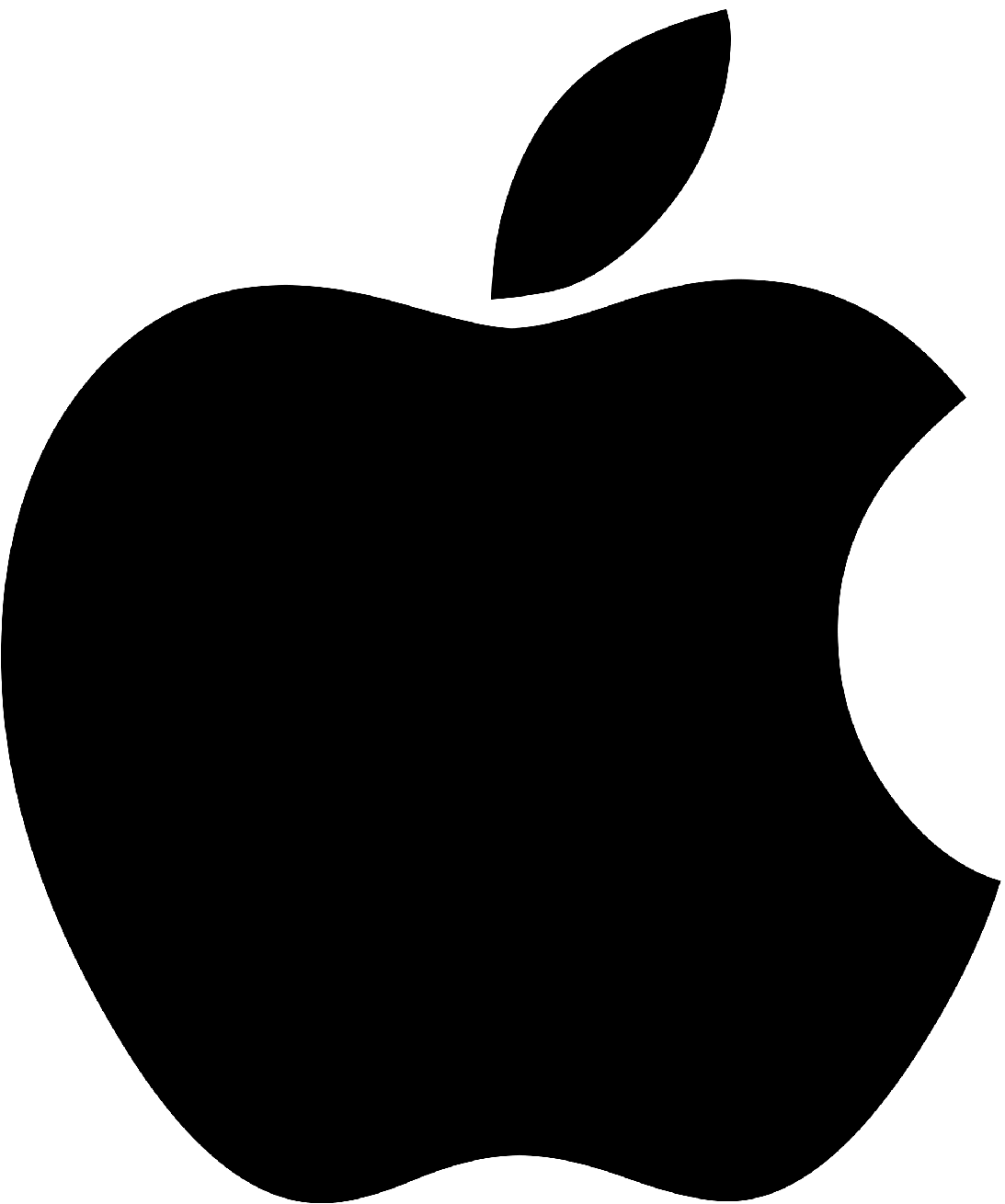
Esta interface es la que nos permitirá entrar hacia los datos de la aplicación que se mostrarán en las siguientes interfaces.





* ***Interfaz Main Menu.*** Esta interfaz solo consta de botones, los cuales, ayudaran al usuario a escoger alguna de las opciones que muestre la aplicación.

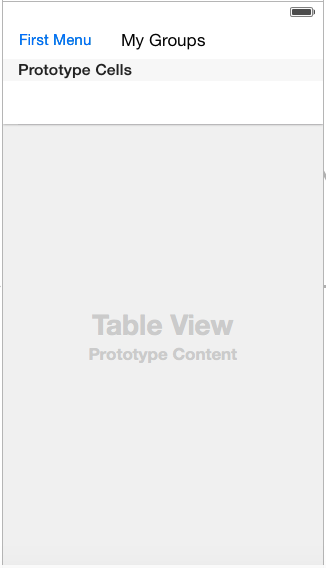


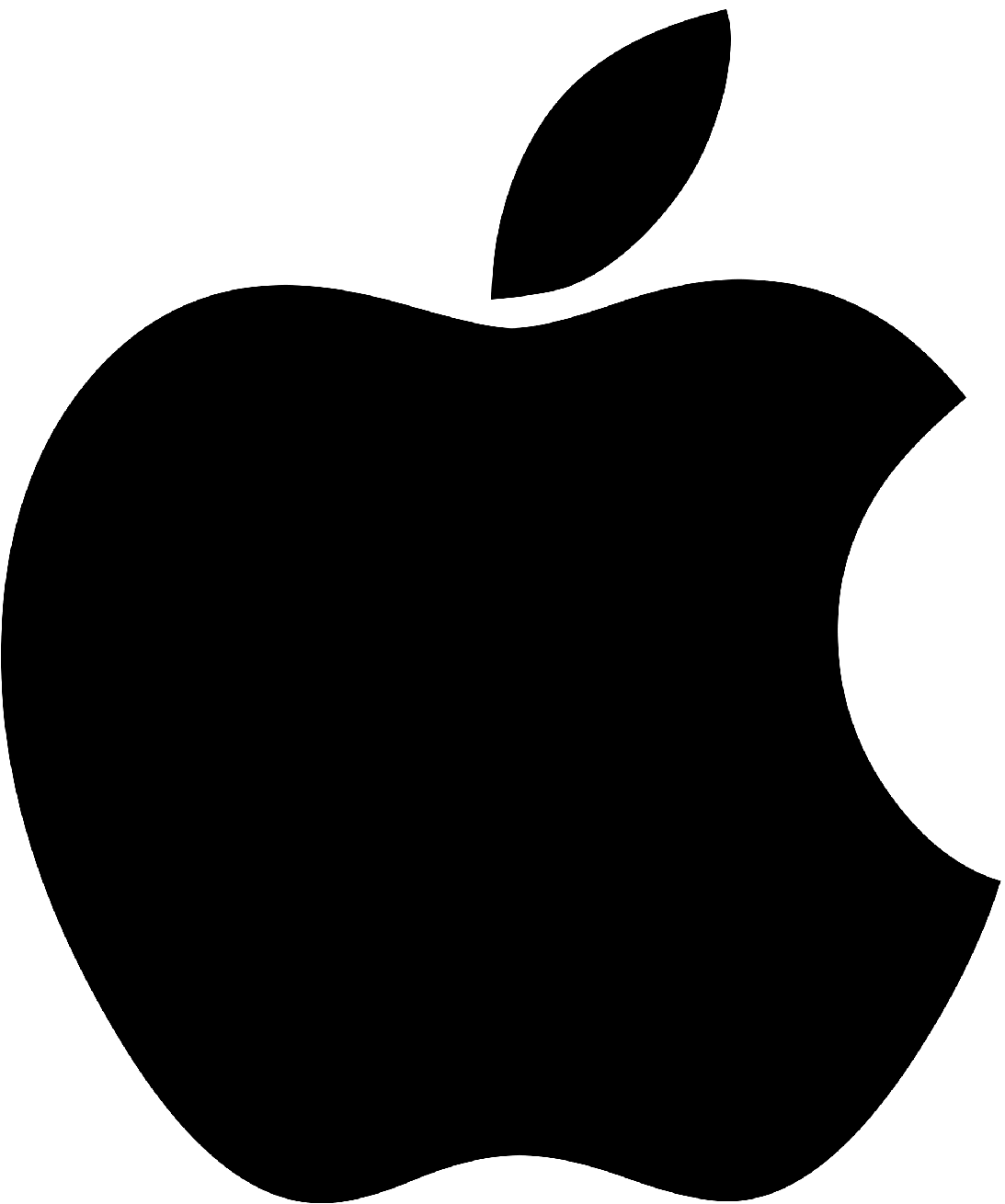


* ***Interfaz de Grupos.*** Esta interfaz muestra cada uno de los grupos asignados al usuario por medio de una lista.

En esta lista, el usuario podrá seleccionar alguno de los grupos que tenga para ver los detalles del grupo. Estos detalles se mostrarán en otra interfaz de usuario que está diseñada para mostrar al usuario cada uno de los detalles de una forma entendible y eficaz.

La interfaz de usuario de Grupos es sencilla y fácil de usar. Consta de una lista mostrada en un objeto o contenedor llamado Table View, el cual, ordena los elementos en forma de columna. Esta interfaz tiene código hacia el servicio web y base de datos; los cuales, son los únicos recursos con los que cuenta para su funcionamiento. Aunque el listado es sencillo, la codificación para la interfaz no lo es tanto ya que tiene un grado de complejidad de bastante interés para el desarrollador.

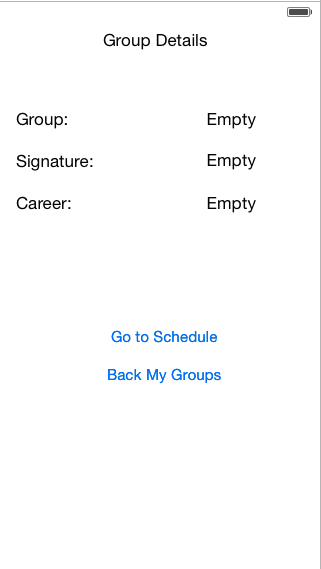


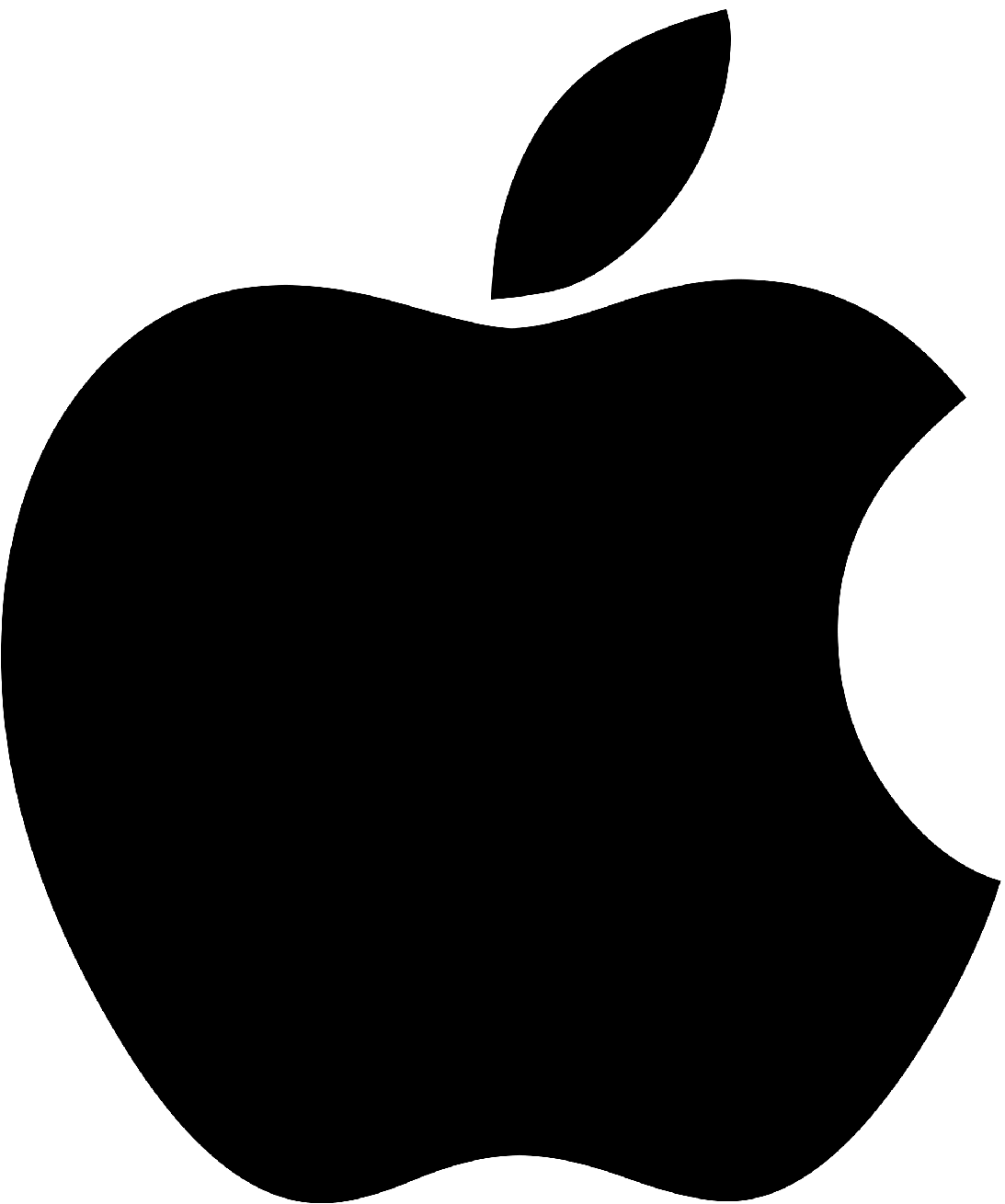


* ***Interfaz Detalles de Grupos.*** Esta interfaz se despliega posteriormente tras haber seleccionado alguno de los grupos de la lista que se mostró con anterioridad.

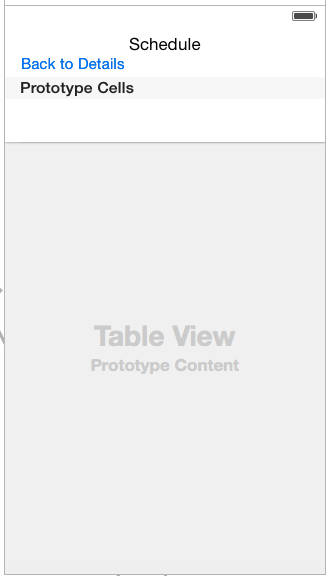
Al seleccionar alguno de estos grupos, en la interfaz ***Detalles de grupos*** , se mostrarán los datos de ese grupo que se tengan almacenados en la base de datos. Estos datos se verán reflejados en los Labels correspondientes así como también se mostrará en el table view de la siguiente interfaz la información de dicho grupo ,como el horario con sus días y horas.

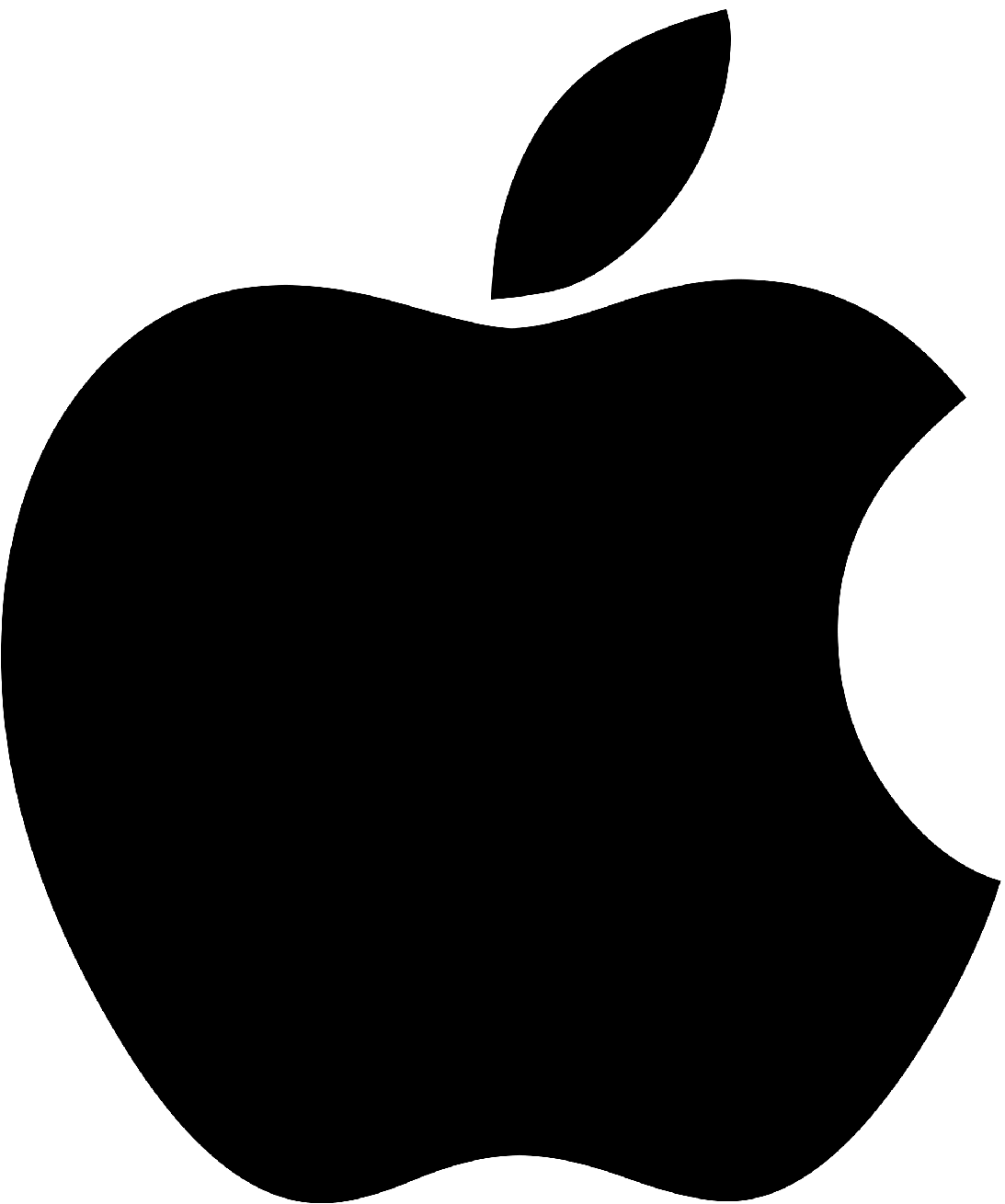
Esta interfaz, al igual que la anterior, esta codificada con clases en Objective-C ligadas a la base de datos y también lleva como único consumible la base datos y el servicio.

******



* ***Interfaz Schedule.***  Esta interfaz solo contiene un table view en el que se mostrara el horario del grupo que se haya seleccionado. Es importante decir que esta interfaz también esta codificada hacia el servicio web y la base de datos de la aplicación en el servidor.





**Código de la aplicación.**

El código de la aplicación viene separado en dos secciones distintas, aunque dependientes una de la otra, en cada interface. En Objective-C, cada interfaz cuenta con dos tipos de documentos para codificación; el **.m**  y el **.h**.

A continuación se muestra el código de cada sección con sus respectivos comentarios acerca de su funcionamiento dentro de la aplicación.

**Login.**

* ***LoginViewController.h***

//

// Login\_ViewController.h

// CONEBU Mobile App

//

// Created by Alberto Ramos on 30/11/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import <UIKit/UIKit.h>

@interface Login\_ViewController : UIViewController

//Objetos del VIEW Controller

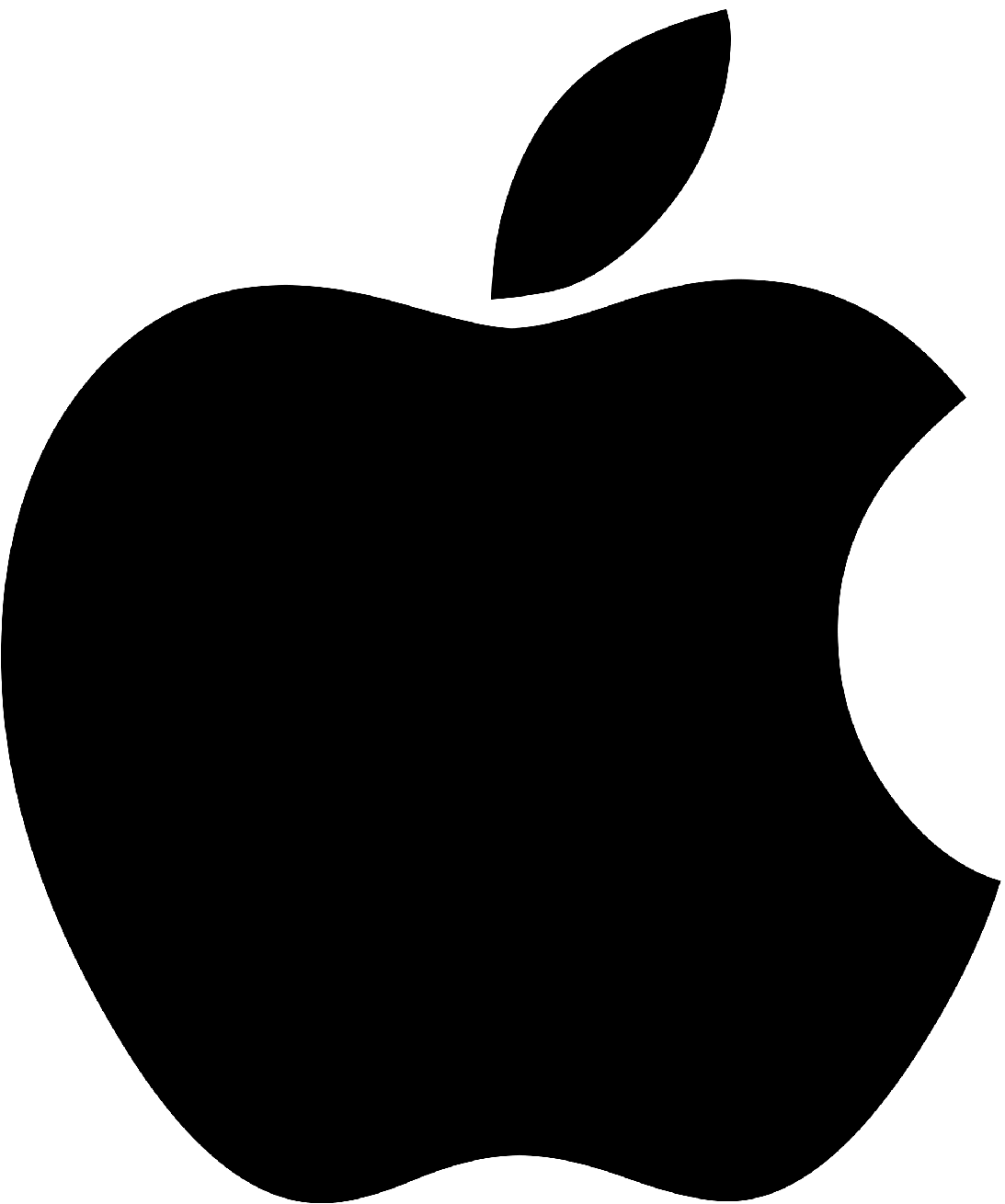
@property (unsafe\_unretained, nonatomic) IBOutlet UITextField \*UserName;

@property (unsafe\_unretained, nonatomic) IBOutlet UITextField \*Password;

@property (unsafe\_unretained, nonatomic) IBOutlet UIButton \*btn\_login;

@property (nonatomic, retain) IBOutlet UIView \*mrView;

@end



* ***LoginViewController.m***

//

// Login\_ViewController.m

// CONEBU Mobile App

//

// Created by Alberto Ramos on 30/11/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import "Login\_ViewController.h"

#import "PrincipalMenu\_ViewController.h"

@implementation Login\_ViewController

- (IBAction)login:(id)sender {

//limpiado variable global de url

[[NSUserDefaults standardUserDefaults]setObject:@"" forKey:@"url"];

[[NSUserDefaults standardUserDefaults] synchronize];

//asignando direccion de webService

[[NSUserDefaults standardUserDefaults]setObject:@"http://localhost/webservice/" forKey:@"url"];

[[NSUserDefaults standardUserDefaults] synchronize];

NSURL \*url = [NSURL URLWithString:@"http://localhost/webservice/wbs\_login.php"];

NSError \*error = nil;

NSData \*data = [NSData dataWithContentsOfURL:url options:0 error:&error];

if (error)

{

NSLog(@"%s: dataWithContentsOfURL error: %@", \_\_FUNCTION\_\_, error);

return;

}

NSArray \*array = [NSJSONSerialization JSONObjectWithData:data options:0 error:&error];

if (error)

{

NSLog(@"%s: JSONObjectWithData error: %@", \_\_FUNCTION\_\_, error);

return;

}

NSString \*matricula=self.UserName.text;

NSString \*json\_resp;

for (int i=0;i<array.count;i++)

{

NSArray \*firstItemArray = array[i];

json\_resp=firstItemArray[0];

}

[[NSUserDefaults standardUserDefaults]setObject:@"" forKey:@"matricula"];

[[NSUserDefaults standardUserDefaults] synchronize];

// etc.

if([json\_resp isEqual:@"1"])

{

[[NSUserDefaults standardUserDefaults]setObject:matricula forKey:@"matricula"];

[[NSUserDefaults standardUserDefaults] synchronize];

NSLog(@"username saved = %@", matricula);

[self performSegueWithIdentifier:@"login" sender:self];

}

}

-(void)prepareForSegue:(UIStoryboardSegue \*)segue sender:(id)sender {

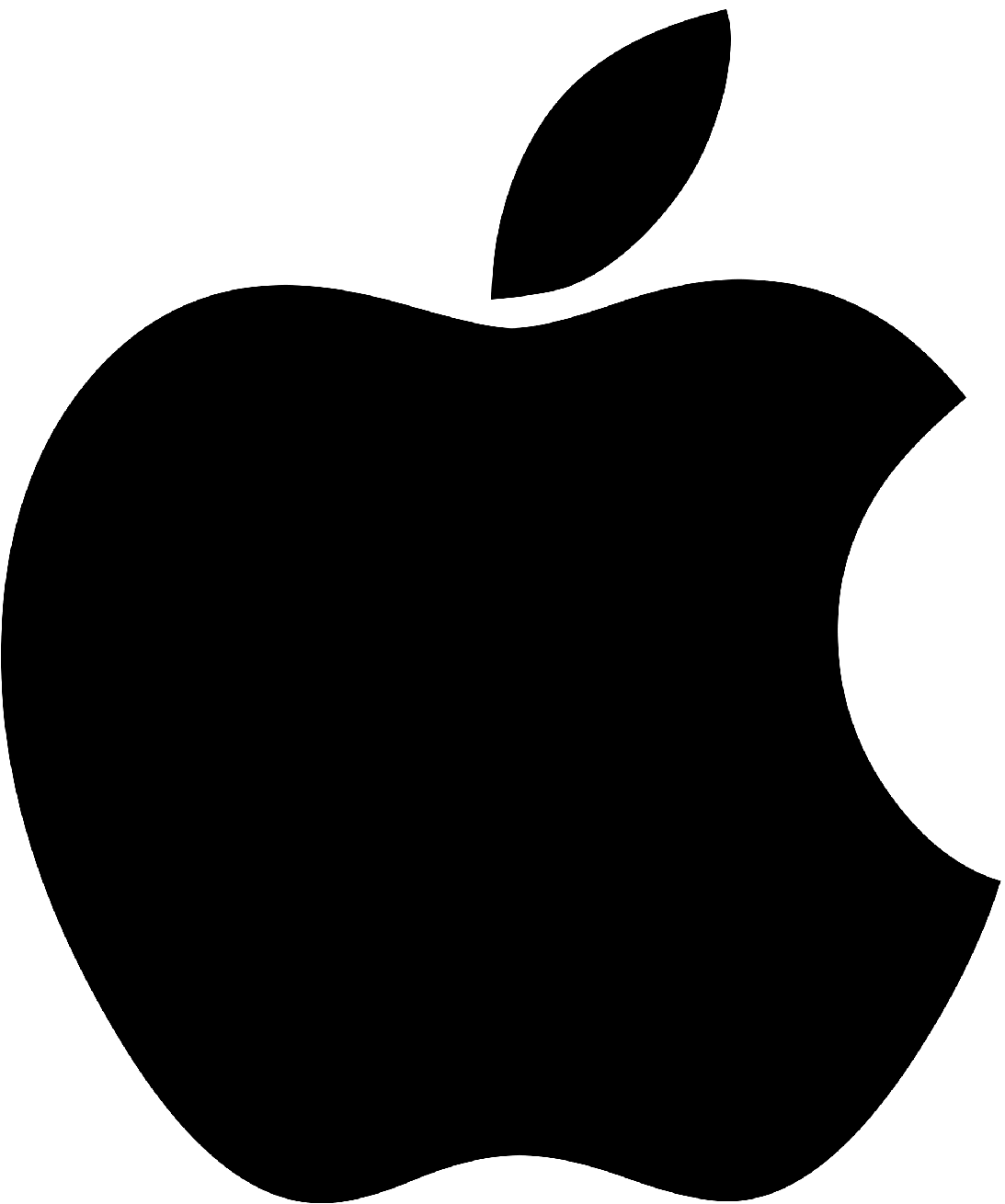
if ([[segue identifier] isEqualToString:@"login"]) {

PrincipalMenu\_ViewController \*purchaseContr = [segue

destinationViewController]; }

}

@end



**Principal Menu View Controller.**

* ***PrincipalMenuViewController.h***

//

// PrincipalMenu\_ViewController.h

// CONEBU Mobile App

//

// Created by Alberto Ramos on 01/12/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import <UIKit/UIKit.h>

@interface PrincipalMenu\_ViewController : UIViewController

@property (weak, nonatomic) IBOutlet UILabel \*lbl\_profesor;

@end

* ***PrincipalMenuViewController.m***

//

// PrincipalMenu\_ViewController.m

// CONEBU Mobile App

//

// Created by Alberto Ramos on 01/12/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import "PrincipalMenu\_ViewController.h"

@implementation PrincipalMenu\_ViewController

- (void)viewDidLoad

{

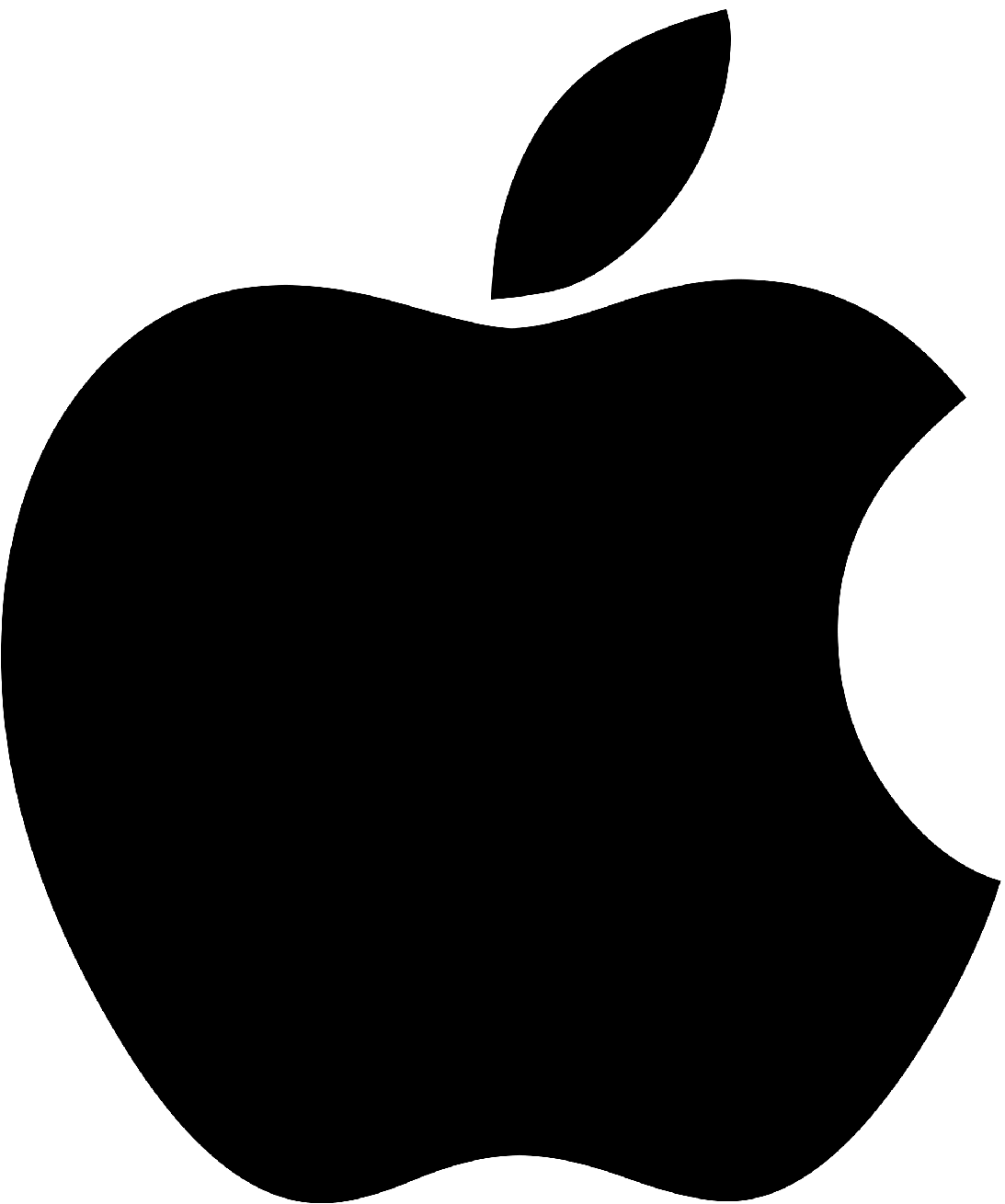
[super viewDidLoad];

NSString \* formato\_etiqueta = [NSString stringWithFormat:@"Professor: %@",[[NSUserDefaults standardUserDefaults] objectForKey:@"matricula"]];

self.lbl\_profesor.text=formato\_etiqueta;

}

@end



**Detail Group View Controller.**

* ***Detail\_Grupo\_View\_Controller.h***

//

// Detail\_Grupo\_View\_Controller.h

// CONEBU Mobile App

//

// Created by Alberto Ramos on 30/11/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import <UIKit/UIKit.h>

#import "Grupos.h"

@interface Detail\_Grupo\_View\_Controller : UIViewController

//Objetos de la interfaz grafica enlazados

@property (strong,nonatomic) Grupos \*Grupo\_Seleccionado;

@property (weak, nonatomic) IBOutlet UILabel \*Grupo;

@property (weak, nonatomic) IBOutlet UILabel \*Materia;

@property (weak, nonatomic) IBOutlet UILabel \*Carrera;

@property (weak, nonatomic) IBOutlet UITableView \*listViewHorario;

@end

* ***Detail\_Grupo\_View\_Controller.m***

//

// Detail\_Grupo\_View\_Controller.m

// CONEBU Mobile App

//

// Created by Alberto Ramos on 30/11/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import "Detail\_Grupo\_View\_Controller.h"

@interface Detail\_Grupo\_View\_Controller()

{

}

@end

@implementation Detail\_Grupo\_View\_Controller

- (void)viewDidLoad

{

[super viewDidLoad];

//Asignando Variables pasadas mediante el Segue de enlace

self.Grupo.text=self.Grupo\_Seleccionado.Clave;

self.Materia.text=self.Grupo\_Seleccionado.Materia;

self.Carrera.text=self.Grupo\_Seleccionado.Carrera;

}

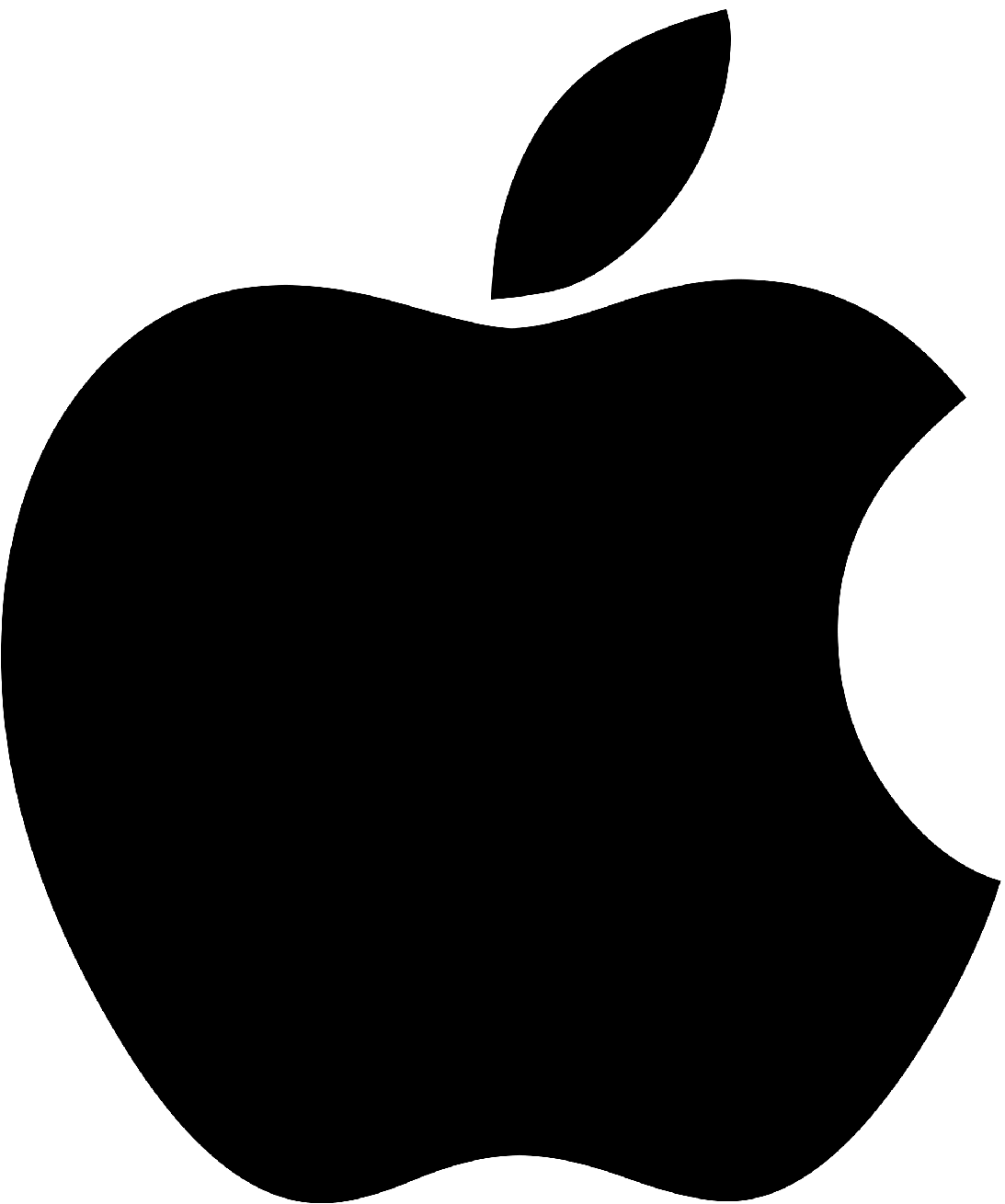
- (void)didReceiveMemoryWarning

{

[super didReceiveMemoryWarning];

}

@end



**Grupos View Controller.**

* ***Grupos\_View\_Controller.h***

//

// Grupos\_VIew\_Controller.h

// CONEBU Mobile App

//

// Created by Alberto Ramos on 30/11/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import <UIKit/UIKit.h>

#import "HomeModel.h" //importando el modelo de datos

@interface Grupos\_VIew\_Controller : UIViewController <UITableViewDataSource, UITableViewDelegate, HomeModelProtocol> //Asignando los metodos del protocolo de Datos

@property (weak, nonatomic) IBOutlet UITableView \*listTableView; //objeto Table View Controller de grupos

@end

* ***Grupos\_View\_Controller.m***

//

// Grupos\_VIew\_Controller.m

// CONEBU Mobile App

//

// Created by Alberto Ramos on 30/11/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import "Grupos\_VIew\_Controller.h"

#import "Grupos.h"

#import "Detail\_Grupo\_View\_Controller.h"

@interface Grupos\_VIew\_Controller ()

{

HomeModel \*\_homeModel;

NSArray \*\_feedItems;

Grupos \*\_GrupoSeleccionado;

}

@end

@implementation Grupos\_VIew\_Controller

- (void)viewDidLoad

{

[super viewDidLoad];

// Asignar del tableView el delegado y el data source al view controller

self.listTableView.delegate = self;

self.listTableView.dataSource = self;

// Creando un arreglo de objetos y asignandolo a la variable \_feedItems

\_feedItems = [[NSArray alloc] init];

// creando un nuevo objeto HomeModel y asignandolo a la variable \_homeModel

\_homeModel = [[HomeModel alloc] init];

// Asignando a este ViewController el objeto que es el delegado del HomeModel \_homeModel.delegate = self;

// Llamar el metodo de descarga de objetos de la clase HomeModel

[\_homeModel downloadItems];

}

- (void)didReceiveMemoryWarning

{

[super didReceiveMemoryWarning];

// Dispose of any resources that can be recreated.

}

-(void)itemsDownloaded:(NSArray \*)items

{

// This delegate method will get called when the items are finished downloading

// Este metodo de delegado es llamado cuando los objetos terminan de ser descargados

\_feedItems = items;

// Recargando el table view

[self.listTableView reloadData];

}

#pragma mark Table View Delegate Methods

- (NSInteger)tableView:(UITableView \*)tableView numberOfRowsInSection:(NSInteger)section

{

// Regresando el numero de objetos descargados (Inicia en 0)

return \_feedItems.count;

}

- (UITableViewCell \*)tableView:(UITableView \*)tableView cellForRowAtIndexPath:(NSIndexPath \*)indexPath

{

// Generando la celda

NSString \*cellIdentifier = @"BasicCell";

UITableViewCell \*myCell = [tableView dequeueReusableCellWithIdentifier:cellIdentifier];

// Tomando el item a mostrar

Grupos \*item = \_feedItems[indexPath.row];

// Generando las referencias de las etiquetas de la celda

//Formando el texto a introducir

NSString \* formato\_celda = [NSString stringWithFormat:@"%@ - %@",item.Clave,item.Materia];

//Asignando valor a la celda

myCell.textLabel.text = formato\_celda;

return myCell; // Devolviendo Celda

}

- (void)tableView:(UITableView \*)tableView didSelectRowAtIndexPath:(NSIndexPath \*)indexPath

{

// Asignando grupo a la variable

\_GrupoSeleccionado = \_feedItems[indexPath.row];

// Llamando manual mente al segue que lleva al view controller de detalles

[self performSegueWithIdentifier:@"detailSegue" sender:self];

}

#pragma mark Segue

- (void)prepareForSegue:(UIStoryboardSegue \*)segue sender:(id)sender

{

// Tomando la referencia del view controller de destino

Detail\_Grupo\_View\_Controller \*detailVC = segue.destinationViewController;

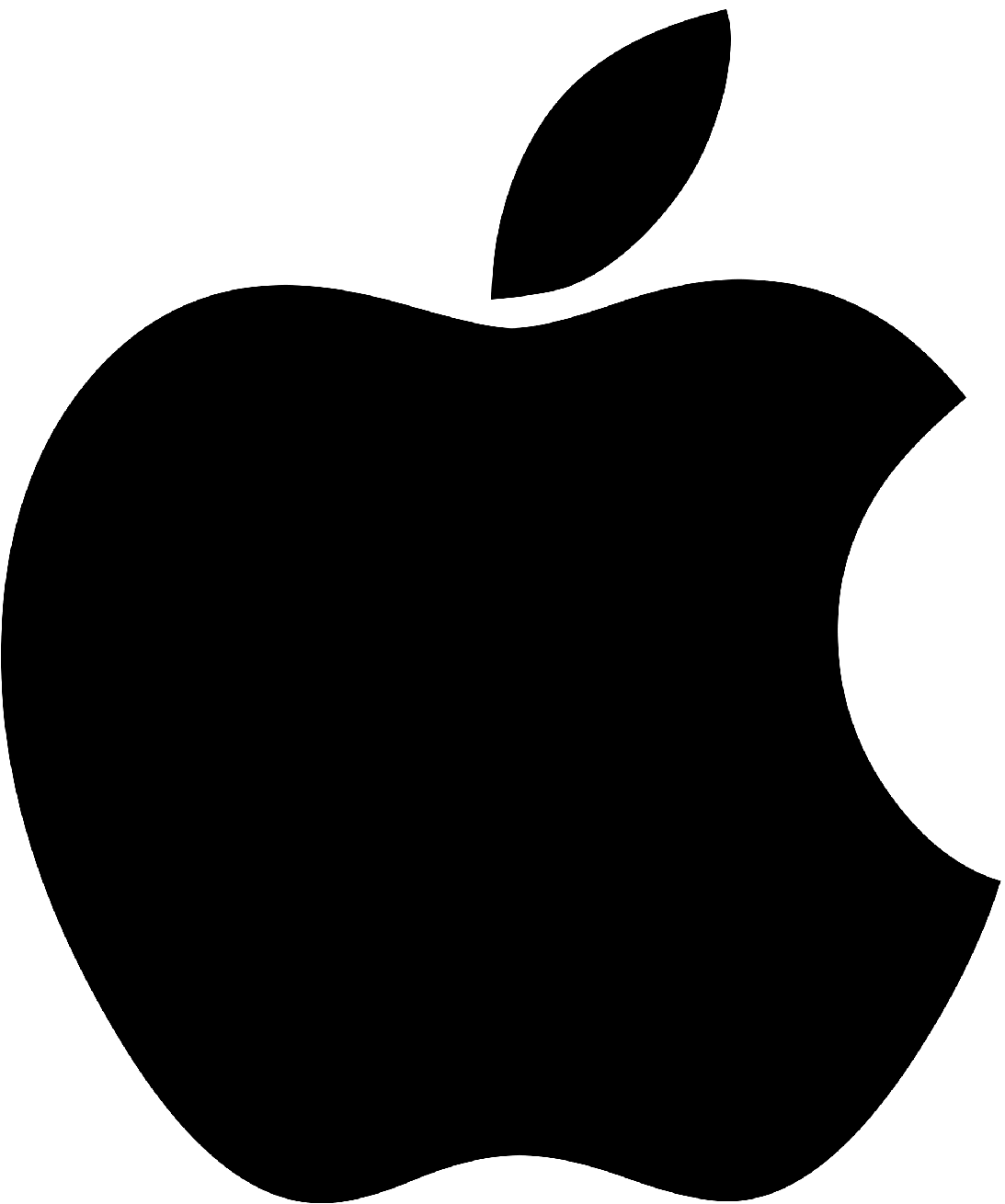
// Asginando la propiedad del grupo seleccionado para cargar datos en el

// view controller de detalles carga, y poder acceder a los objetos descargados

// del webService

detailVC.Grupo\_Seleccionado = \_GrupoSeleccionado;

}

@end

**Grupos.**

**(Código Adicional Dependiente)**

* ***Grupos.h***

//

// Grupos.h

// CONEBU Mobile App

//

// Created by Alberto Ramos on 29/11/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import <Foundation/Foundation.h>

@interface Grupos : NSObject

//Propiedades del objeto Grupos

@property (nonatomic, strong) NSString \*Clave;

@property (nonatomic, strong) NSString \*Materia;

@property (nonatomic,strong) NSString \*Carrera;

@end

* ***Grupos.m***

//

// Grupos.m

// CONEBU Mobile App

//

// Created by Alberto Ramos on 29/11/14.

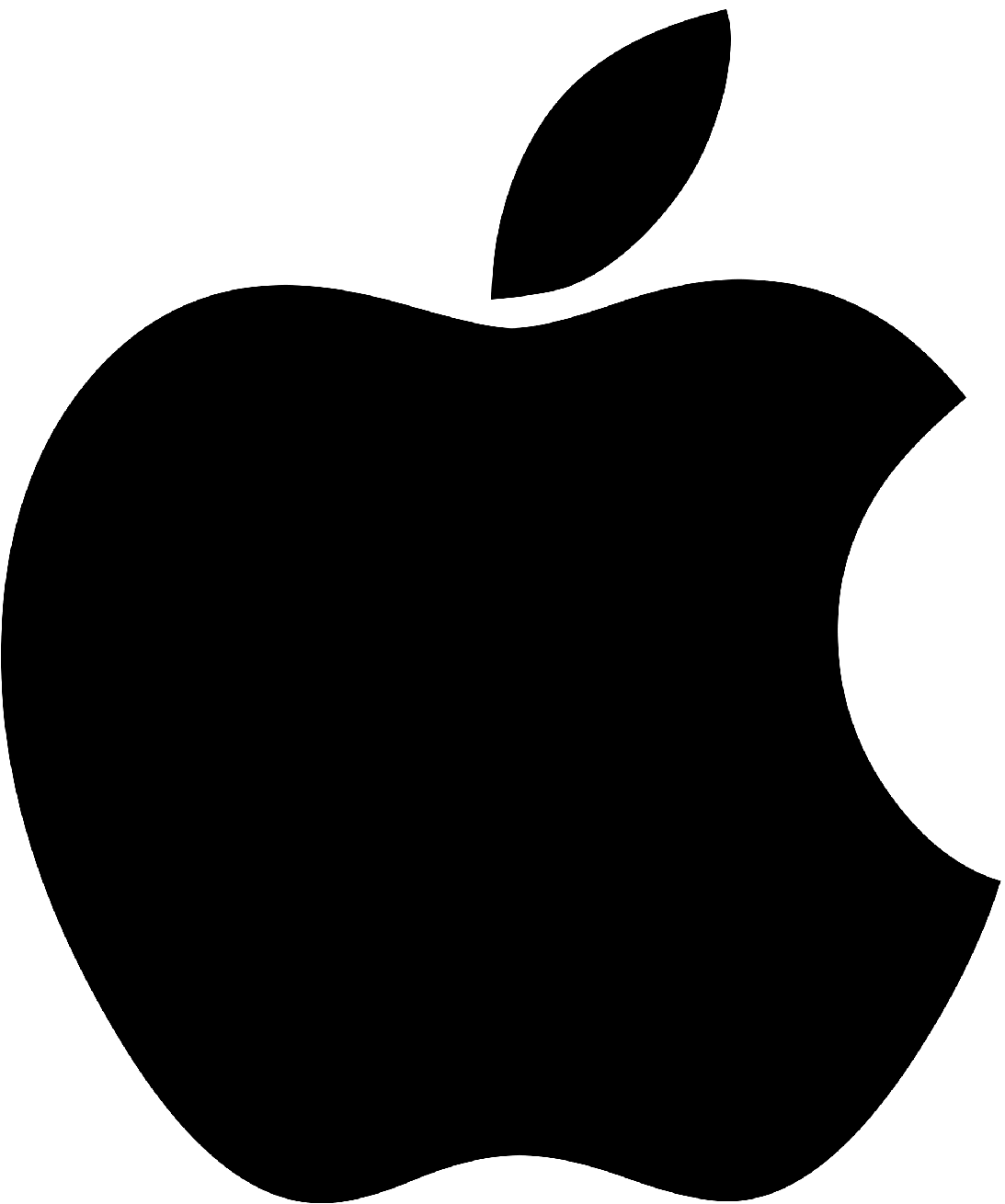
// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import "Grupos.h"

@implementation Grupos

@end



**Home Model.**

**(Codigo Adicional Dependiente)**

* ***HomeModel.h***

//

// HomeModel.h

// CONEBU Mobile App

//

// Created by Alberto Ramos on 29/11/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import <Foundation/Foundation.h>

@protocol HomeModelProtocol <NSObject> //Protocolo de acceso a datos

-(void)itemsDownloaded:(NSArray \*)items; //Arreglo de acceso a datos

@end

@interface HomeModel : NSObject <NSURLConnectionDataDelegate> //Delegando el objeto de conexion

@property (nonatomic, weak) id<HomeModelProtocol> delegate; //Delegando el protocolo de conexion a id

- (void)downloadItems; //Metodo de descarga de Datos

@end

* ***HomeModel.m***

//

// HomeModel.m

// CONEBU Mobile App

//

// Created by Alberto Ramos on 29/11/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import "HomeModel.h"

#import "Grupos.h"

@interface HomeModel()

{

NSMutableData \*\_downloadedData;

}

@end

@implementation HomeModel

- (void)downloadItems

{

//recuperando matricula guardada en el metodo global NSUserDefaults

NSString \*matricula = [[NSUserDefaults standardUserDefaults] objectForKey:@"matricula"];

NSLog(@"username retrieved = %@", matricula); //mandando al log para debugeo

//generando string de conexion a URL de servicio web

NSString \* formato\_url = [NSString stringWithFormat:@"%@IOS\_Grupos.php?Matricula\_Docente=%@", [[NSUserDefaults standardUserDefaults] valueForKey:@"url"],matricula];

// Descarga de datos JSON

NSURL \*jsonFileUrl = [NSURL URLWithString:formato\_url];

// Creando la peticion

NSURLRequest \*urlRequest = [[NSURLRequest alloc] initWithURL:jsonFileUrl];

// creando la conexion NSURLConnection

[NSURLConnection connectionWithRequest:urlRequest delegate:self];

}

#pragma mark NSURLConnectionDataProtocol Methods

- (void)connection:(NSURLConnection \*)connection didReceiveResponse:(NSURLResponse \*)response

{

// Inicializando el objeto de Datos

\_downloadedData = [[NSMutableData alloc] init];

}

- (void)connection:(NSURLConnection \*)connection didReceiveData:(NSData \*)data

{

// Mapeando los nuevos datos descargados

[\_downloadedData appendData:data];

}

- (void)connectionDidFinishLoading:(NSURLConnection \*)connection

{

// Creando el Arreglo para alojar datos

NSMutableArray \*\_Grupos = [[NSMutableArray alloc] init];

// Parseando el JSON que se descargo

NSError \*error;

NSArray \*jsonArray = [NSJSONSerialization JSONObjectWithData:\_downloadedData options:NSJSONReadingAllowFragments error:&error];

// Ciclo para llenar el arreglo de datos con el JSON mapeado

for (int i = 0; i < jsonArray.count; i++)

{

//Primera fila del arreglo mapeado

NSArray \*firstItemArray = jsonArray[i];

// creando nuevo objeto para alojar datos y propiedades del objeto JSON

Grupos \*newGrupo = [[Grupos alloc] init];

newGrupo.Clave=firstItemArray[0];

newGrupo.Materia=firstItemArray[1];

newGrupo.Carrera=firstItemArray[2];

// Añadiendo el nuevo Objeto generado al arreglo de Datos

[\_Grupos addObject:newGrupo];

}

// Notificando al delegado principal que los datos han sido decodificados

if (self.delegate)

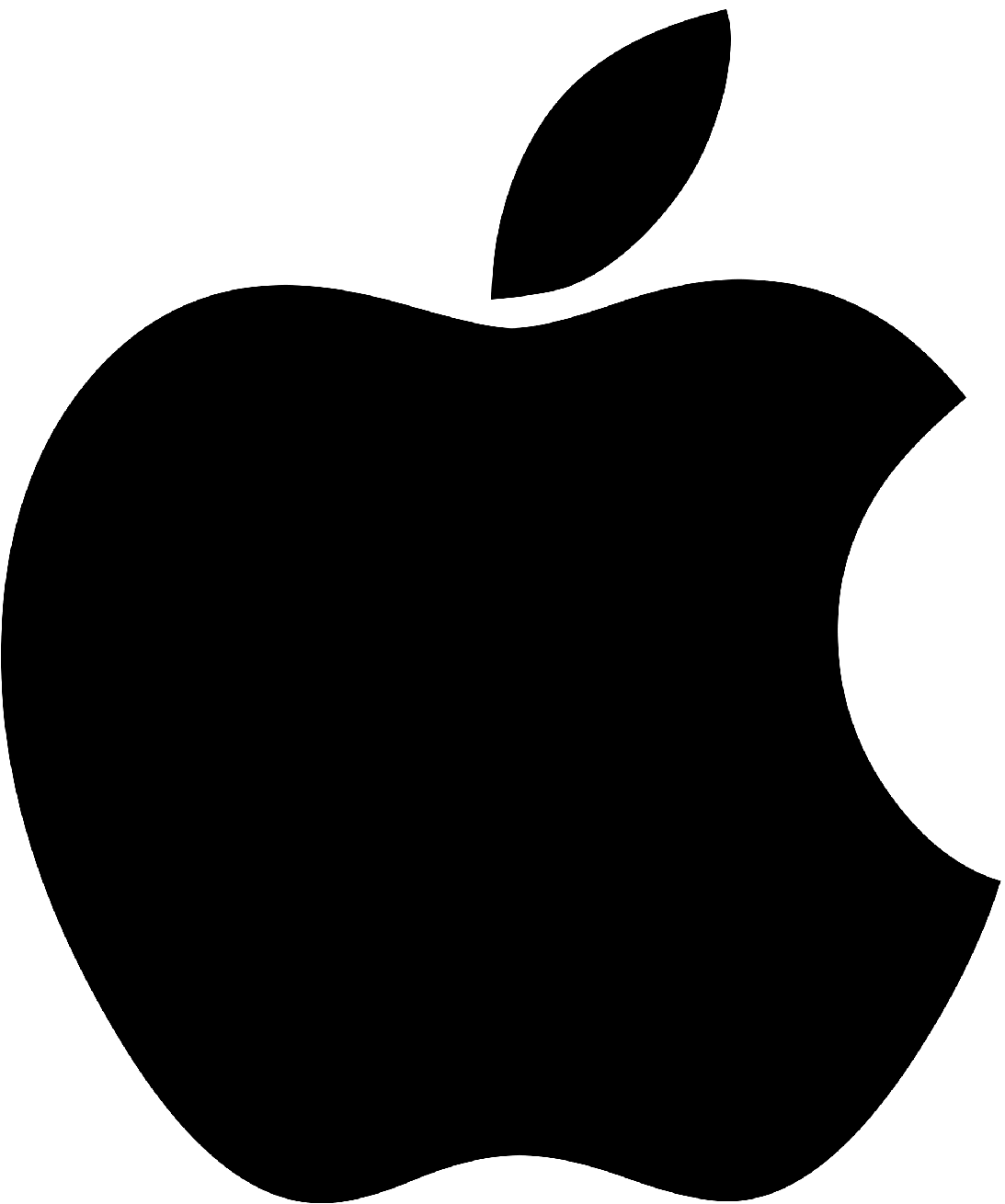
{

[self.delegate itemsDownloaded:\_Grupos];

}

}

@end



**Horarios.**

* ***Horarios.h***

//

// Horarios.h

// CONEBU Mobile App

//

// Created by Alberto Ramos on 01/12/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import <Foundation/Foundation.h>

@interface Horarios : NSObject

@property (nonatomic, strong) NSString \*id\_dia;

@property (nonatomic, strong) NSString \*hr\_inicio;

@property (nonatomic, strong) NSString \*hr\_fin;

@property (nonatomic, strong) NSString \*Salon;

@end

* ***Horarios.m***

//

// Horarios.m

// CONEBU Mobile App

//

// Created by Alberto Ramos on 01/12/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import "Horarios.h"

@implementation Horarios

@end

* HomeModelH.h

//

// HomeModelH.h

// CONEBU Mobile App

//

// Created by Alberto Ramos on 01/12/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import <Foundation/Foundation.h>

@protocol HomeModelProtocolH <NSObject> //Protocolo de acceso a datos

-(void)itemsDownloaded:(NSArray \*)items; //Arreglo de acceso a datos

@end

@interface HomeModelH : NSObject <NSURLConnectionDataDelegate> //Delegando el objeto de conexion

@property (nonatomic, weak) id<HomeModelProtocolH> delegate; //Delegando el protocolo de conexion a id

- (void)downloadItems; //Metodo de descarga de Datos

@end

* ***HomeModelH.m***

// HomeModelH.m

// CONEBU Mobile App

//

// Created by Alberto Ramos on 01/12/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import "HomeModelH.h"

#import "Horarios.h"

@interface HomeModelH()

{

NSMutableData \*\_downloadedData;

}

@end

@implementation HomeModelH

- (void)downloadItems

{

//mandando al log para debugeo

//generando string de conexion a URL de servicio web

NSString \* formato\_url = [NSString stringWithFormat:@"%@IOS\_DetalleGrupo.php?Clave=%@", [[NSUserDefaults standardUserDefaults] valueForKey:@"url"],[[NSUserDefaults standardUserDefaults] valueForKey:@"claveSalon"]];

// Descarga de datos JSON

NSURL \*jsonFileUrl = [NSURL URLWithString:formato\_url];

// Creando la peticion

NSURLRequest \*urlRequest = [[NSURLRequest alloc] initWithURL:jsonFileUrl];

// creando la conexion NSURLConnection

[NSURLConnection connectionWithRequest:urlRequest delegate:self];

}

#pragma mark NSURLConnectionDataProtocol Methods

- (void)connection:(NSURLConnection \*)connection didReceiveResponse:(NSURLResponse \*)response

{

// Inicializando el objeto de Datos

\_downloadedData = [[NSMutableData alloc] init];

}

- (void)connection:(NSURLConnection \*)connection didReceiveData:(NSData \*)data

{

// Mapeando los nuevos datos descargados

[\_downloadedData appendData:data];

}

- (void)connectionDidFinishLoading:(NSURLConnection \*)connection

{

// Creando el Arreglo para alojar datos

NSMutableArray \*\_Horarios = [[NSMutableArray alloc] init];

// Parseando el JSON que se descargo

NSError \*error;

NSArray \*jsonArray = [NSJSONSerialization JSONObjectWithData:\_downloadedData options:NSJSONReadingAllowFragments error:&error];

// Ciclo para llenar el arreglo de datos con el JSON mapeado

for (int i = 0; i < jsonArray.count; i++)

{

//Primera fila del arreglo mapeado

NSArray \*firstItemArray = jsonArray[i];

// creando nuevo objeto para alojar datos y propiedades del objeto JSON

Horarios \*newHorario = [[Horarios alloc] init];

NSString \*id\_dia=firstItemArray[0];

if([id\_dia isEqual:@"1"])

newHorario.id\_dia=@"Lunes";

else if([id\_dia isEqual:@"2"])

newHorario.id\_dia=@"Martes";

else if([id\_dia isEqual:@"3"])

newHorario.id\_dia=@"Miercoles";

else if([id\_dia isEqual:@"4"])

newHorario.id\_dia=@"Jueves";

else if([id\_dia isEqual:@"5"])

newHorario.id\_dia=@"Viernes";

newHorario.hr\_inicio=firstItemArray[1];

newHorario.hr\_fin=firstItemArray[2];

newHorario.Salon=firstItemArray[3];

// Añadiendo el nuevo Objeto generado al arreglo de Datos

[\_Horarios addObject:newHorario];

}

// Notificando al delegado principal que los datos han sido decodificados

if (self.delegate)

{

[self.delegate itemsDownloaded:\_Horarios];

}

}

@end

* ***Horarios\_View\_Controller.h***

//

// Horarios\_View\_Controller.h

// CONEBU Mobile App

//

// Created by Alberto Ramos on 01/12/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import <UIKit/UIKit.h>

#import "HomeModelH.h"

@interface Horarios\_View\_Controller : UIViewController <UITableViewDataSource, UITableViewDelegate, HomeModelProtocolH> // //Asignando los metodos del protocolo de Datos

@property (weak, nonatomic) IBOutlet UITableView \*listTableView;

@end

Horario\_View\_Controller.m

//

// Horarios\_View\_Controller.m

// CONEBU Mobile App

//

// Created by Alberto Ramos on 01/12/14.

// Copyright (c) 2014 Oscar Razo. All rights reserved.

//

#import "Horarios\_View\_Controller.h"

#import "Horarios.h"

@interface Horarios\_View\_Controller ()

{

HomeModelH \*\_homeModel;

NSArray \*\_feedItems;

Horarios \*\_HorarioSeleccionado;

}

@end

@implementation Horarios\_View\_Controller

- (void)viewDidLoad

{

[super viewDidLoad];

// Asignar del tableView el delegado y el data source al view controller

self.listTableView.delegate = self;

self.listTableView.dataSource = self;

// Creando un arreglo de objetos y asignandolo a la variable \_feedItems

\_feedItems = [[NSArray alloc] init];

// creando un nuevo objeto HomeModel y asignandolo a la variable \_homeModel

\_homeModel = [[HomeModelH alloc] init];

// Asignando a este ViewController el objeto que es el delegado del HomeModel

\_homeModel.delegate = self;

// Llamar el metodo de descarga de objetos de la clase HomeModel

[\_homeModel downloadItems];

}

- (void)didReceiveMemoryWarning

{

[super didReceiveMemoryWarning];

// Dispose of any resources that can be recreated.

}

-(void)itemsDownloaded:(NSArray \*)items

{

// This delegate method will get called when the items are finished downloading

// Este metodo de delegado es llamado cuando los objetos terminan de ser descargados

\_feedItems = items;

// Recargando el table view

[self.listTableView reloadData];

}

#pragma mark Table View Delegate Methods

- (NSInteger)tableView:(UITableView \*)tableView numberOfRowsInSection:(NSInteger)section

{

// Regresando el numero de objetos descargados (Inicia en 0)

return \_feedItems.count;

}

- (UITableViewCell \*)tableView:(UITableView \*)tableView cellForRowAtIndexPath:(NSIndexPath \*)indexPath

{

// Generando la celda

NSString \*cellIdentifier = @"BasicCell";

UITableViewCell \*myCell = [tableView dequeueReusableCellWithIdentifier:cellIdentifier];

// Tomando el item a mostrar

Horarios \*item = \_feedItems[indexPath.row];

// Generando las referencias de las etiquetas de la celda

//Formando el texto a introducir

NSString \* formato\_celda = [NSString stringWithFormat:@"%@ / %@ - %@ / Salon: %@",item.id\_dia,item.hr\_inicio,item.hr\_fin,item.Salon];

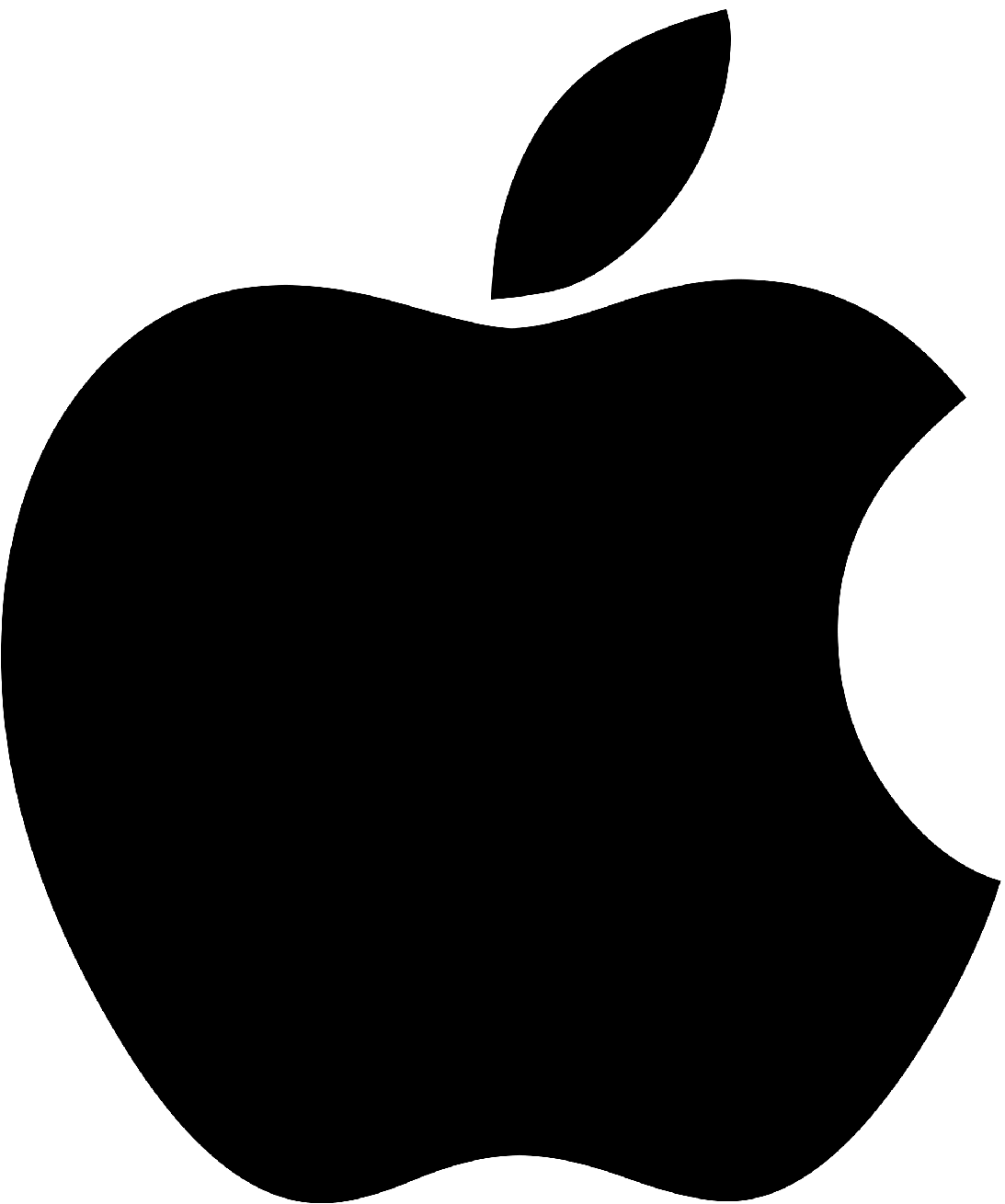
//Asignando valor a la celda

myCell.textLabel.text = formato\_celda;

return myCell; // Devolviendo Celda

}

@end



**Descripción de Servicios.**

# Requisición de servicio web para aplicación móvil – Detalle de Grupo

El nombre del método para obtener Detalle de grupos se divide en 2 servicios web:

* IOS\_Detalles\_grupo
* IOS\_Horarios\_de\_grupo

Ambos servicios web deberán recibir 1 parámetro serializado en formato JSON

* Clave de Grupo – String

Y como repuesta cada servicio servicio web deberá serializar y mandar en JSON 1 arreglo con la siguiente información:

Arreglo IOS\_Detalles\_grupo

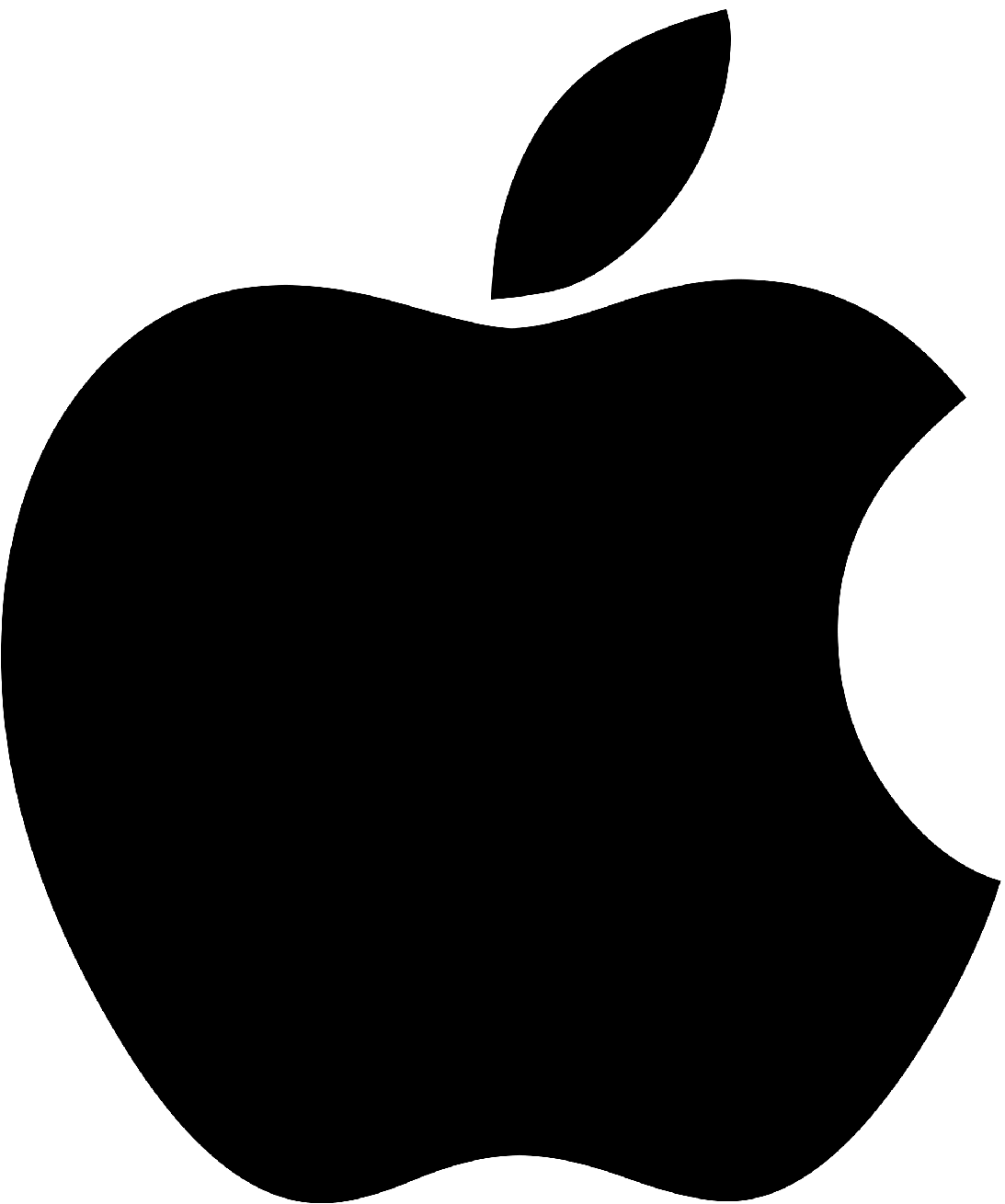
* Grupo
* Materia
* Grado/Nivel

De forma que el arreglo tenga la forma de -> Grupo\Materia\Grado

Arreglo IOS\_Horarios\_de\_grupo

* Dia
* Hora
* Salón

De forma que el arreglo tenga la forma de -> Dia\Hora\Salón



# Requisición de servicio web para aplicación móvil – Login

El nombre del método de login, deberá ser nombrado como “IOS\_Login”.

El servicio web, deberá recibir 2 parámetros serializados en formato JSON

* Usuario
* Contraseña

Y como repuesta el servicio web deberá serializar y mandar en JSON un valor entro acorde a:

* 1 si el usuario y contraseña es correcto
* 0 si el usuario y contraseña es incorrecto

# Requisición de servicio web para aplicación móvil – Grupos

El nombre del método para obtener grupos, deberá ser nombrado como “IOS\_Grupos”.

El servicio web, deberá recibir 1 parámetro serializado en formato JSON

* Clave del docente – String

Y como repuesta el servicio web deberá serializar y mandar en JSON un arreglo con la información de los grupos asignados al usuario.

El arreglo deberá contener:

* Clave del Grupo
* Materia del Grupo
* Carrera del Grupo

