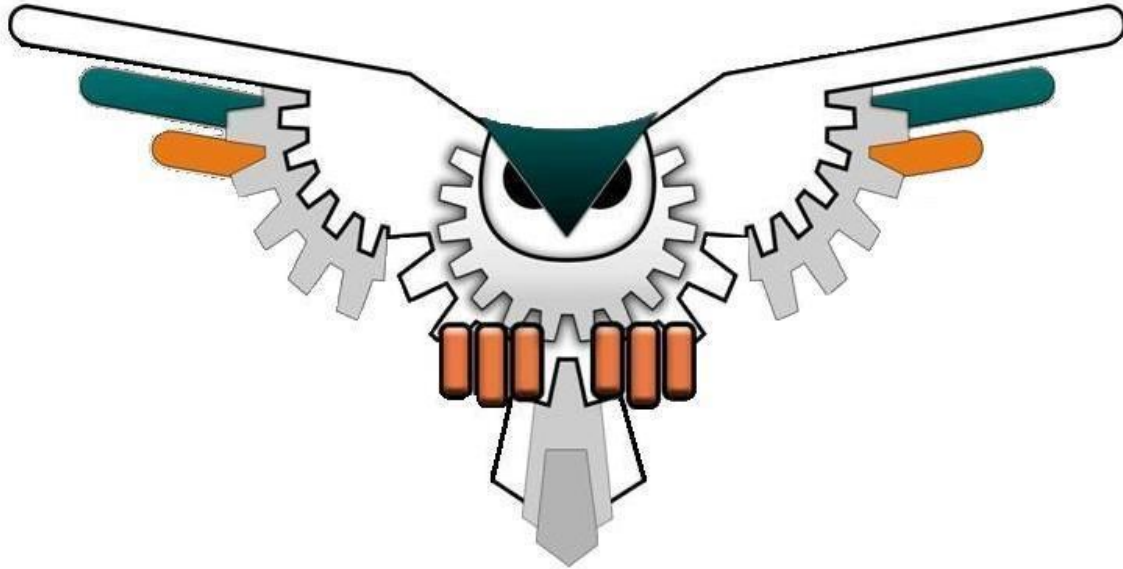


Instituto Tecnológico Superior de Irapuato.



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 específico.

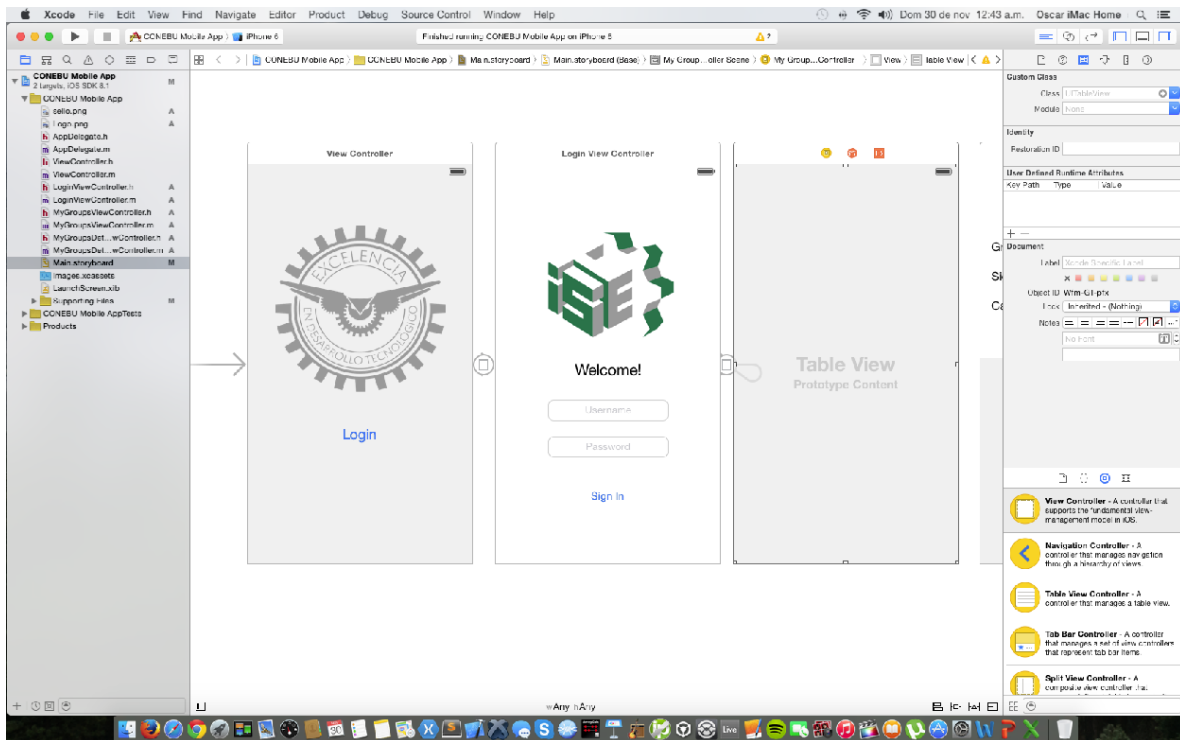
El aplicativo consumirá los servicios web y la base de datos disponibles en el servidor para la validación de Login y también para el consumo de los datos que se mostraran en el table view de *Grupos*, *Detalles de Grupos* y *Horarios de Grupos*.

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.



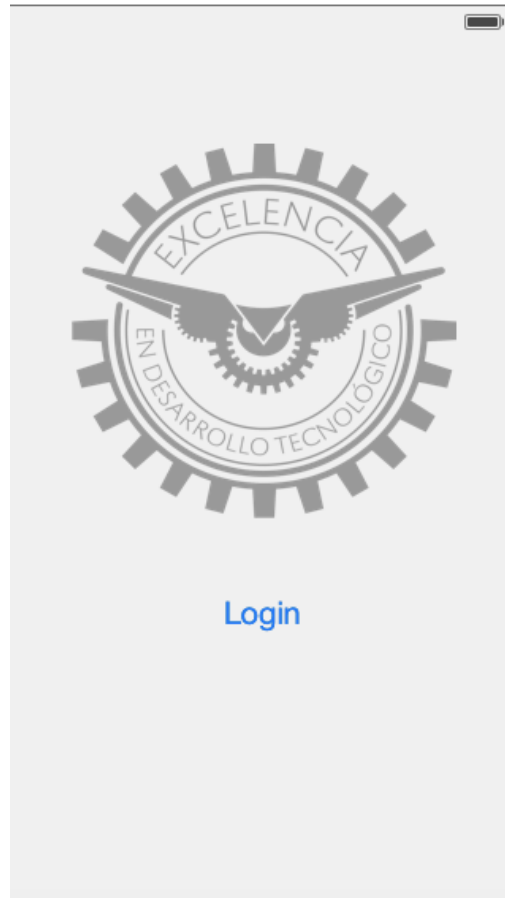
El lenguaje utilizado en la programación es Objective-C, lenguaje nativo de Apple para el desarrollo de aplicaciones para dispositivos móviles, así como también para aplicativos de escritorio.

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*.

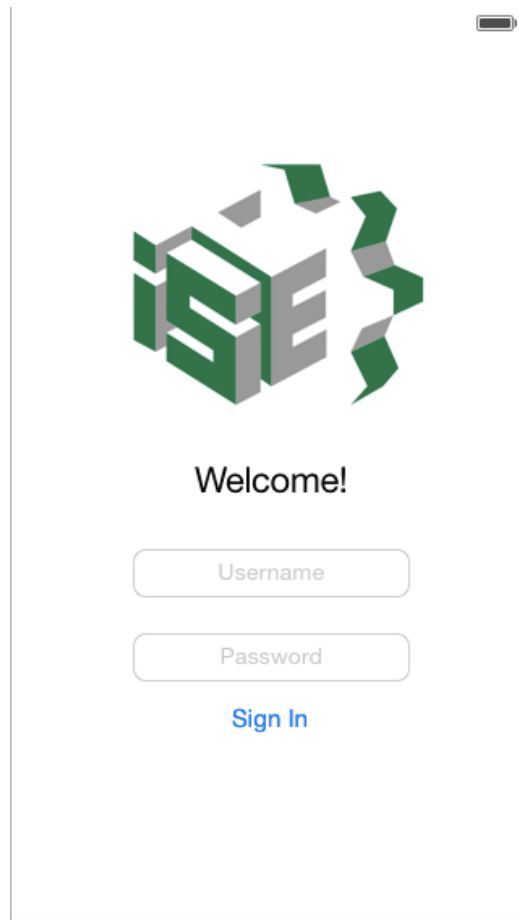


- **Interfaz de Bienvenida.** Esta interfaz sólo contiene un botón con el cual se accede a la ventana de Login. No contiene ninguna clase asociada con el manejo de datos en la aplicación para su funcionamiento .

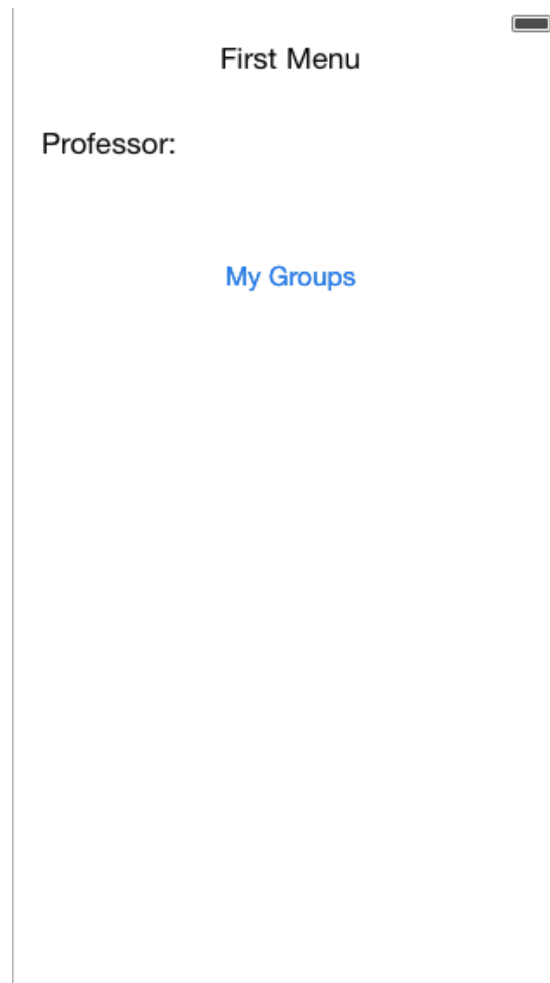


- **Interfaz de Login.** Esta interfaz tiene dos cajas de texto, en estas el usuario ingresará su nombre y su contraseña. Esta sección está codificada hacia la base de datos de la aplicación, esto con el fin de obtener y validar los datos del usuario ,como nombre y contraseña.

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



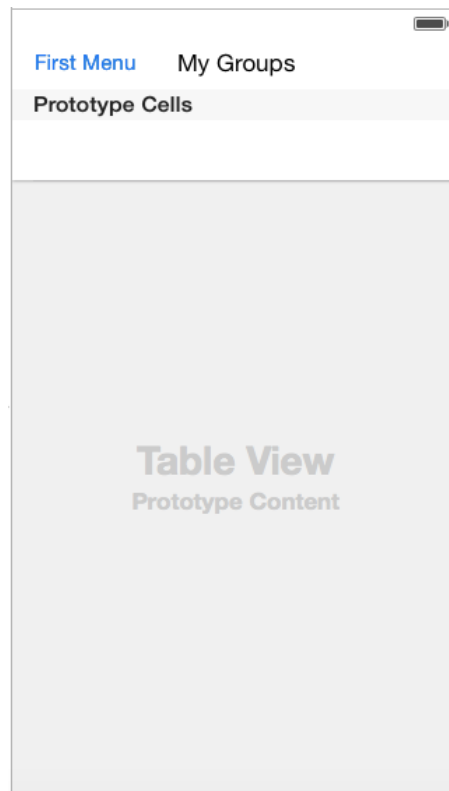
- ***Interfaz Main Menu.*** Esta interfaz solo consta de botones, los cuales, ayudaran al usuario a escoger alguna de las opciones que muestre la aplicación.



- **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.

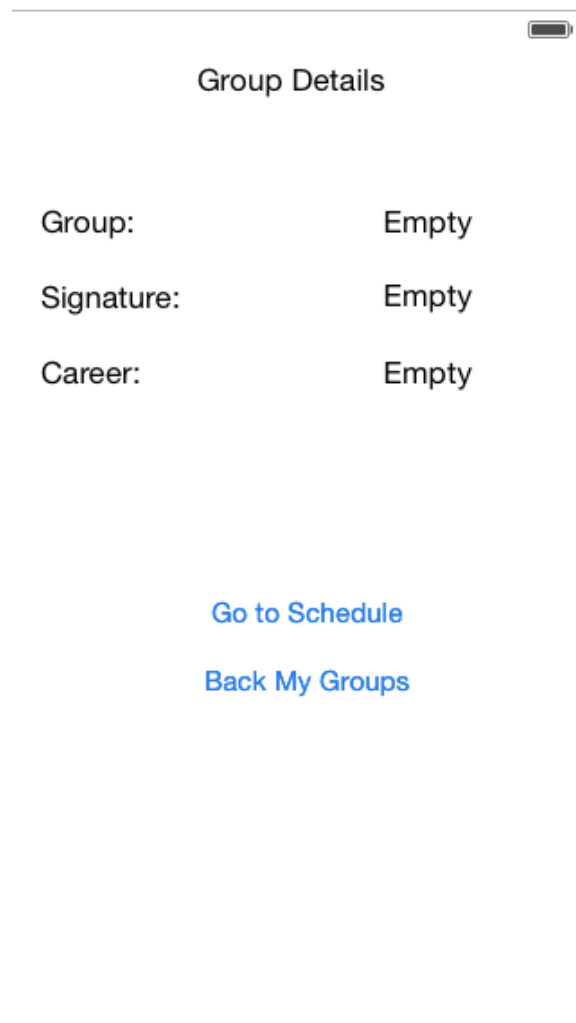
La interfaz de usuario de Grupos es sencilla y fácil de usar. Consta de una lista mostrada en un objeto o contenedor llamado Table View, el cual, ordena los elementos en forma de columna. Esta interfaz tiene código hacia el servicio web y base de datos; los cuales, son los únicos recursos con los que 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.

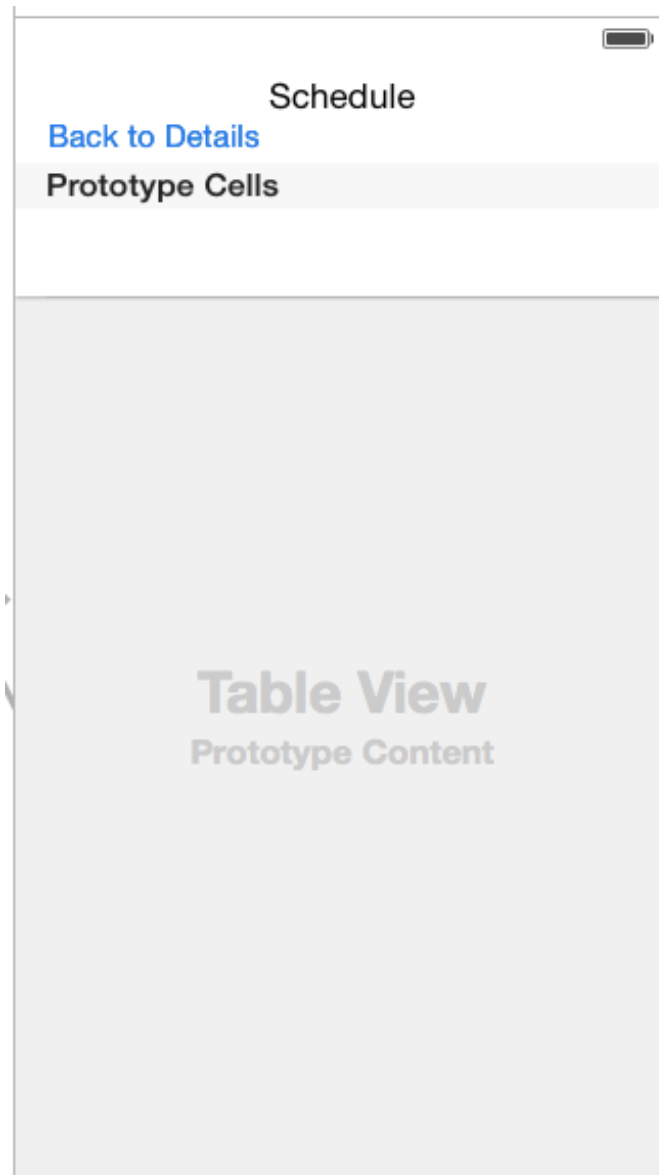
Esta interfaz, al igual que la anterior, esta codificada con clases en Objective-C ligadas a la base de datos y también lleva como único consumible la base datos y el servicio.



The screenshot displays a mobile application interface titled "Group Details". At the top right, there is a battery status icon. Below the title, there are three rows of labels and values: "Group:" followed by "Empty", "Signature:" followed by "Empty", and "Career:" followed by "Empty". At the bottom of the interface, there are two blue text buttons: "Go to Schedule" and "Back My Groups".



- **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.Material];
    //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:@"%I@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.Salón=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 respuesta cada 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 respuesta el servicio web deberá serializar y mandar en JSON un valor entero 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 respuesta 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

