

# Curso IOS

## Práctica 11 - Aplicación Final de Curso

Daniel Prado Rodríguez  
Enero 2012



# Práctica 11 - Aplicación Final de Curso

## ¿Jugamos una Porra?

Daniel Prado Rodríguez  
Enero 2012

### Idea

Para la realización de la práctica 11 como trabajo final del Curso IOS, no tenía una idea predefinida de Aplicación con vistas a su posible publicación en la AppStore, etc., así que he buscado una idea sencilla pero que permitiera repasar los conceptos principales que hemos ido viendo a lo largo del curso.

La idea se trata de una aplicación para gestionar las típicas apuestas informales sobre partidos de fútbol, con amigos o compañeros de trabajo, y en las que se suele jugar una pequeña cantidad (1€, 2€...). Cada jugador apuesta por un resultado del partido, y el acertante se lleva la cantidad recaudada.

### Borrador

Antes de empezar a trabajar con Xcode, le dediqué bastante tiempo a imaginar el funcionamiento de la aplicación y plasmarlo sobre el papel realizando unos bocetos de cómo serían las pantallas principales de la aplicación. Encontré que este proceso fue tremendamente útil para perfilar y corregir ideas pre-concebidas de funcionamiento.

También dediqué un tiempo a examinar el funcionamiento y diseño de algunas apps instaladas en mi propio Iphone, como la aplicación de Calendario, o 'Remember the Milk' para la gestión de tareas.

### Diseño

He intentado que el diseño de la aplicación se basara en los siguientes factores:

1. Paradigma Modelo-Vista-Controlador (MVC).
2. Utilización de soluciones comunes a otras aplicaciones, como por ejemplo:
  - a. Navigation Controller combinado con Tab Bar.
  - b. Selector de contenido (filtro) de tablas mediante un UISegmentController.
3. Simplicidad visual y funcionamiento intuitivo.

El esqueleto de la Aplicación se ha creado en XCode 3.2, partiendo de la plantilla de aplicación **Window-Based Application con CoreData**.

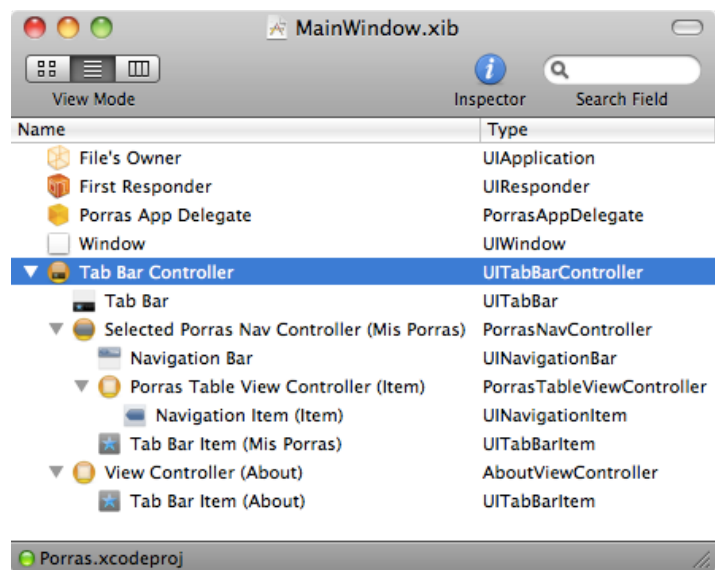
La clase AppDelegate, a parte de inicializar la propia aplicación y los objetos de CoreData, también carga un Plist que he diseñado para configurar los nombres de los equipos de fútbol y los nombres de los correspondientes imágenes PNG con los escudos.

A partir de ahí, en Interface Builder, he definido la jerarquía objetos controladores y vistas principales de la aplicación en el fichero MainWindow.xib.

1. En el primer nivel de la jerarquía se sitúa un UITabBarController.
2. A su vez, este contiene un Controlador de vista para cada 'tab'. Para esta primera versión de la app, solo he definido 2 tabs, uno con la funcionalidad de la aplicación y otro con una pantalla simple de 'About' e instrucciones de uso.  
El uso de TabBar permite ampliar fácilmente la aplicación en el futuro de forma modular.

Aunque para este app no era estrictamente necesario (se podría haber implementado con un único Navigation

Controller), también me motivaba aprender a como implementar esta combinación de TabBar + Navigation Controller, ya que es muy típica en muchas apps.



*Estructura de MainWindow.xib*

3. Del primer Tab, cuelga un UINavigationController.
4. El controlador de la vista principal de la aplicación lo he llamado "PorrasTableViewController", (en IB PorrasSegmentTableViewController.xib) que está compuesto por un UITableView y un selector de tipo segmento. Obsérvese que no es subclase de UITableViewController, aunque como contiene una tabla, sí implementa los correspondientes delegados de UITableView.

```
@interface PorrasTableViewController : UIViewController <PorraNewFormControllerDelegate,
                                                         NSFetchResultsControllerDelegate,
                                                         UITableViewDelegate,
                                                         UITableViewDataSource> {
    IBOutlet UITableView *porrasTable;
    IBOutlet UISegmentedControl *selTable;
}
```

*Cabecera del interface PorrasTableViewController con los outlets a UITableView y el control segmentado*

Toda esta definición de objetos en Interface Builder se ha ido realizando a la vez en Xcode... Los controladores de una vista superior tienen IBOutlets a los controladores de nivel inferior. Esto facilita el acceso a los datos y la propagación del managedObjectContext de CoreData.



*Vista Principal de la Aplicación.*

La implementación de la tabla de la vista principal se ha realizado mediante celdas customizadas. Cada celda representa un objeto Porra (PorraCD en CoreData) que contiene los escudos de los dos equipos, el importe de la apuesta, y el resultado final del partido.

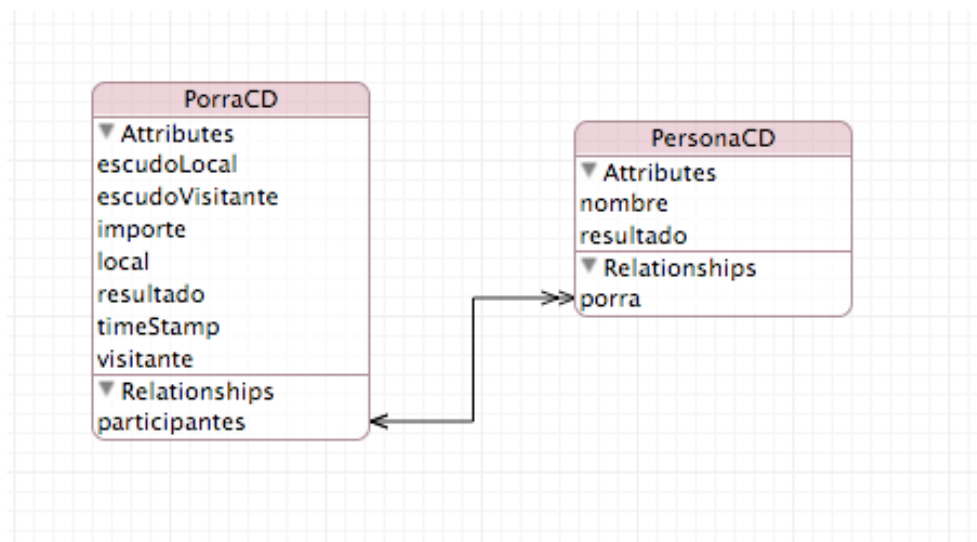
Para poder ver más información de la porra, el usuario debe seleccionar dicha celda para acceder a la vista de detalle.

En todo caso, las diferentes vistas de detalle se explican en el apartado de Manual de Usuario de este documento.

## CoreData

El modelo de datos de la aplicación se ha implementado en CoreData.

A continuación se presenta el modelo diseñado en Xcode:



*Modelo Core-Data*

El modelo consta simplemente de dos Entidades:

- Una entidad “PorraCD”, que contiene los atributos que pueden visualizarse en las pantallas y celdas correspondientes. También se incluye un timestamp (NSDate) que se utilizará como clave única.
- Una entidad “PersonaCD”, que contiene a la vez el nombre de una persona y su correspondiente apuesta sobre el resultado.

Vemos que una Porra puede contener varios participantes (PersonaCD), pero un participante solo puede pertenecer a una porra.

En la pantalla Principal de la aplicación, se ha implementado, de forma similar a como hicimos en la aplicación de Productos, un NSFetchedResultsController de CoreData que alimenta la tabla.

La novedad en mi aplicación es la posibilidad de filtrar los objetos de dicho NSFetchedResultsController, mediante la utilización de un NSPredicate. El usuario, mediante el UISegmentedControl, selecciona entre porras Abiertas y Cerradas, invocando la siguiente IBAction donde se puede observar el establecimiento del NSPredicate y el correspondiente refresco de NSFetchedResultsController:

```

-(IBAction) cambiaSegmento:(id)sender {
    [self saveContext];

    NSPredicate *predicate;
    switch (self.selTable.selectedSegmentIndex){
        case 0:
            [self.navigationItem.rightBarButtonItem setEnabled:TRUE];
            //Mostramos las Porras Abiertas (cuyo resultado es nil)
            predicate = [NSPredicate predicateWithFormat:@"resultado == nil"];
            break;
        case 1:
            [self.navigationItem.rightBarButtonItem setEnabled:FALSE];
            //Mostramos las Porras Cerradas (cuyo resultado es != nil)
            predicate = [NSPredicate predicateWithFormat:@"resultado != nil"];
            break;
        default:
            break;
    }
    [[self.fetchedResultsController fetchRequest] setPredicate:predicate];
    [NSFetchedResultsController deleteCacheWithName:@"Root"];

    NSError * error = nil;
    [self.fetchedResultsController performFetch:&error];
    if (error) {
        // report error
        NSLog(@"Catastrófico!");
    }
    [self.porrasTable reloadData];
}

```

*Uso de NSPredicate y refresco del NSFetchedResultsController*

# Manual de Usuario

IZDA: Primera pantalla que ve el usuario al arrancar la aplicación

DCHA: al pulsar + aparece la pantalla para añadir una nueva porra (vista modal)



Los campos de la pantalla Nueva Porra se introduce seleccionado el campo correspondiente, apareciendo vistas modales auxiliares:

IZDA: vista modal de introducción de importe.

DCHA: vista modal de selección de equipo.



Al salvar los datos, se graba la porra en coredata y se visualiza en la pantalla principal, en la lista de porras abiertas.

IZDA: Vista de porras abiertas con la porra recién creada.

DCHA: al seleccionar la porra, navegamos a la vista de detalle, desde donde podemos añadir las apuestas de los participantes.



Desde la vista de detalle, añadimos participantes.

IZDA: Pantalla de introducción de nombre y apuesta del participante.

DCHA: participante añadido a la sección tercera de la vista de detalle.





IZDA: Mientras la porra esté abierta, se puede acceder al detalle de la apuesta de un participante, para modificarla o eliminarla.

DCHA: Para cerrar la porra, hay que introducir el resultado del partido. Esto se hace seleccionando la celda de resultado de la vista de detalle.

Una vez cerrada la porra, no se permite editar ninguno de sus datos.



IZDA: Vista de detalle de la porra cerrada (parte superior)

DCHA: Vista de detalle de la porra cerrada (parte inferior). Nótese que los acertantes de la porra se marcan de color verde, y los perdedores en rojo.

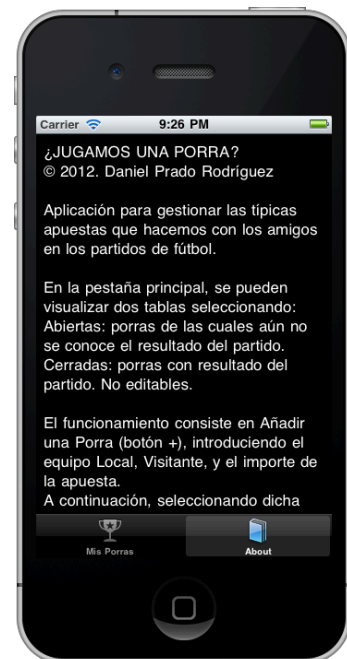


IZDA: Volvemos a la lista de porras. Para ver la porra que acabamos de cerrar tenemos que pulsar en "Cerradas".

DCHA: La porra puede eliminarse completamente (junto con las apuestas de sus participantes) mediante el botón edit o deslizando el dedo por la celda.



Pantalla del Tab 'About'.



## Posibilidades futuras

Aunque el funcionamiento de esta primera versión de la aplicación es muy sencillo y no deja de ser un entorno donde apuntar y consultar las apuestas de forma estructurada, creo que la app podría tener algunas posibilidades de publicarse añadiendo ciertas funcionalidades para hacerla más atractiva. Por ejemplo:

- Pre-carga automática desde un servidor web de los partidos de las diferentes competiciones.
- Posibilidad de compartir la información con los participantes (email, facebook, etc.)

- Posibilidad de convertirla en una App 'social' 2.0.

## Conclusión personal

La realización de esta práctica, como primera aplicación de diseño 'libre' a la que me enfrentaba, ha supuesto un esfuerzo en horas bastante considerable, más de lo que esperaba, supongo que fruto de la inexperiencia en este entorno y de las características propias de IOS. Encuentro que desarrollar para dispositivos móviles es más laborioso que para aplicaciones de PC (entorno Visual C o .NET a los que estaba más acostumbrado).

Sin embargo, está claro que enfrentándose a los problemas inesperados que han ido apareciendo en esta práctica he aprendido mucho más sobre desarrollo IOS que con las anteriores, lo cual me hace estar bastante satisfecho con el resultado.