

# iAd, AdMob e In Apps - Ejercicios

## Índice

1 Incorporando iAd a una aplicación iOS.....	2
2 (*) Gestionando las orientaciones del dispositivo con iAd.....	3
3 Incorporando AdMob a una aplicación iOS.....	3
4 Integrando in-apps: desbloquear pantalla secreta.....	3

## 1. Incorporando iAd a una aplicación iOS

En este ejercicio vamos a crear una aplicación muy sencilla para iPhone que contenga la plataforma de publicidad **iAd** como modelo de negocio. La aplicación estará compuesta por una sola ventana que tendrá un objeto de tipo `UILabel` el cual informará del estado de la publicidad (visible o no visible). En el caso de que esta no esté visible mostraremos una imagen de publicidad nuestra. De este modo siempre se mostrará publicidad en la aplicación. Para completar el ejercicio deberemos de realizar los siguientes pasos:

1) Creamos el proyecto en XCode con los siguientes datos:

- Product name: `ejercicio_iad`
- Company Identifier: `es.ua.jtech`
- Class prefix: `UA`
- Device family: `iPhone`
- Marcar sólo la opción `Use Automatic Reference Counting`. El resto dejarlas desmarcadas.

2) Modificamos el fichero `UAViewController.xib` añadiendo un objeto de tipo `Label` y el objeto de tipo `Ad BannerView` (el banner iAd) en la parte superior de la pantalla y un objeto de tipo `Image View` que será nuestro propio banner. Definimos los outlets correspondientes en el fichero `UAViewController.h`, los `synthesize` necesarios en el fichero `UAViewController.m` y establecemos las relaciones desde la vista.



Mi banner

3) Importamos el framework de iAd (`iAd.framework`) al proyecto.

4) Modificamos la definición de la controladora (fichero `UAViewController.h`) para que esta implemente el protocolo `ADBannerViewDelegate`.

5) Añadimos los dos métodos necesarios del protocolo `ADBannerViewDelegate` en la clase `UAViewController`:

```
//metodos de la clase delegada de ADBannerView
#pragma mark ADBannerViewDelegate

- (void)bannerViewDidLoadAd:(ADBannerView *)banner {
    // Completar: banner visible y actualizar label
}

- (void)bannerView:(ADBannerView *)banner
didFailToReceiveAdWithError:(NSError *)error
{
    // Completar: iAd no visible, mostrar nuestro banner y actualizar
    label
}
```

6) Comprobar que funciona todo bien. ¿Qué pasa si rotamos el dispositivo?

## 2. (\*) Gestionando las orientaciones del dispositivo con iAd

En este ejercicio vamos a tener en cuenta las distintas posiciones que puede adoptar el dispositivo (portrait y landscape). Cuando giremos el dispositivo el banner de iAd debe de girar también, para ello modificaremos el ejercicio anterior implementando el método `willRotateToInterfaceOrientation` de manera que cuando este se ejecute se compruebe la orientación del dispositivo y el banner gire según esta.

## 3. Incorporando AdMob a una aplicación iOS

En este ejercicio, al igual que hemos hecho en el primero, vamos a incorporar **AdMob** a una aplicación de iPhone. En este caso, para evitar todo el proceso de registro en el sitio de AdMob (<http://es.admob.com/>) comenzaremos el ejercicio desde una plantilla que podemos descargar desde [aquí](#).

Completar el archivo `UAViewController.m` con el código necesario para mostrar el banner:

- a) En la parte inferior de la pantalla.
- b) En la parte superior de la pantalla.
- c) Código para hacer las peticiones del banner en modo testing.

¿Qué ventajas y desventajas encuentras al usar la plataforma AdMob con respecto a iAd de Apple?

## 4. Integrando in-apps: desbloquear pantalla secreta

En este ejercicio vamos a usar los in-apps para desbloquear una pantalla secreta de una aplicación. El ejercicio consistirá en programar parte de la controladora de una aplicación para conseguir que funcione el in-app que desbloquea un botón que da acceso a una pantalla "secreta".

Ya que para usar in-apps necesitamos disponer de una cuenta de desarrollador Apple y nosotros no la tenemos, usaremos una plantilla con todo el proceso previo de preparación de la aplicación en iTunes Connect, certificados, etc ya implementado. Lo único que deberemos hacer será completar una serie de ejercicios necesarios para el funcionamiento del in-app.

- a) Importar el framework `StoreKit.framework` necesario para el uso de la API de in-apps.

- b) Completar el método `restoreTransaction`
- c) Completar el método `completeTransaction`
- d) Completar el método `desbloqueaVentanaSecreta`

Podemos probar si funciona el in-app usando el siguiente usuario de test:

- Nombre de usuario: `usuariotest@ua.es`
- Contraseña: `usuariotest1A`

La plantilla del ejercicio se puede descargar desde [aquí](#).

#### Ayuda

Para almacenar en memoria el estado de in-app comprado o no comprado podemos hacerlo usando el método de persistencia `NSUserDefaults` de la siguiente forma:

```
[[NSUserDefaults standardUserDefaults] setBool:YES  
forKey:@"inappComprado"];
```

