

Práctica Fundamentos iOS

KeepCoding

HackerBooks

Un lector de libros en PDF para iPhone & iPad . Los libros en sí los recibirás en un json que deberás descargar de una URL.

Modelo

Un libro contiene las siguientes propiedades:

- título
- autores (una lista)
- tags (una lista)
- url a la imagen del libro
- url al pdf

Necesitarás una clase AGTBook para esto.

Una biblioteca representa un conjunto de libros. Una biblioteca permite acceder a los libros de varias maneras: por orden alfabético de título o por temática (los tags). Esto se debe de ver reflejado en una clase AGTLibrary.

Antes de seguir piensa en cómo modelar esto. Intenta crear algunas clases que lo representen y que respondan a las necesidades de una tabla. Sigue con el ejercicio después de haber dedicado al menos algún tiempo para pensar en el modelo.

Posible guía para AGTLibrary:

```
//Array de libros

var books: NSMutableArray

//Array de tags con todas las distintas temáticas en
// orden alfabético. No puede bajo ningún concepto haber
// ninguno repetido

var tags: NSMutableArray

// Número total de libros

var booksCount: Int{
    get{
        let count: Int = self.books.count
        return count
    }
}

// Cantidad de libros que hay en una temática.
// Si el tag no existe, debe de devolver cero

func bookCountForTag (tag: String?) -> Int

// Array de los libros(instancias de AGTBook) que hay en
// una temática.
// Un libro puede estar en una o más temáticas. Si no hay
// libros para una temática, ha de devolver nil

func booksForTag (tag: String?) -> [AGTBook]?

// Un AGTBook para el libro que está en la posición
// 'index' de aquellos bajo un cierto
// tag. Mira a ver si puedes usar el método anterior
// para hacer parte de tu trabajo.
// Si el índice no existe o el tag no existe, ha de devolver
// nil.

func bookAtIndex(index: Int) -> AGTBook?
```

Procesado del JSON

Recibirás un JSON con un array de libros. Cada libro tendrá esta pinta:

```
{
    "authors": "Scott Chacon, Ben Straub",
    "image_url": "http://hackershelf.com/media/cache/
b4/24/b42409de128aa7f1c9abbbfa549914de.jpg",
    "pdf_url": "https://progit2.s3.amazonaws.com/en/
2015-03-06-439c2/progit-en.376.pdf",
    "tags": "version control, git",
    "title": "Pro Git"
}
```

Deberás procesarlo para convertirlo en objetos de las clases `AGTBook` y `AGTLibrary`. Para averiguar como se convierte un JSON en un `Dictionary` o un `Array` de `Dictionary`, consulta la clase `NSJSONSerialization` en la ayuda. Necesitas uno de sus métodos de clase.

Ten en cuenta que dicho método devuelve un parámetro `AnyObject` que puede contener tanto un `Array` de `Dictionary` como un `Dictionary`. Mira en la ayuda en método `isKindOfClass:` y como usarlo para saber qué te han devuelto exactamente. ¿En qué otros modos podemos trabajar? ¿is, as?

Especificaciones de HackerBooks

Modelo

Al arrancar la App por primera vez (esto lo averiguas con `NSUserDefaults`), descarga el JSON (tienes la URL en la sección de recursos) como un `NSData` y guárdalo en la carpeta `Documents` de tu `Sandbox`. Para ello, usa el constructor/inicializador que estimes de la clase `NSData`.

Haz lo mismo para las imágenes de portada y los pdfs. ¿Donde guardarías estos datos? Cambia las urls de remotas a locales, para que la próxima vez que se cargue la App, se pillen los datos locales.

Verás que todo esto tarda un poco y bloquea la interfaz de la aplicación durante un breve momento. No te preocupes por ello a estas alturas. En otra asignatura veremos como resolver esto de forma elegante.

En las siguientes ocasiones que arranque la App (ya sabemos que tenemos el JSON y los datos en local), crea tu modelo a partir de él. Crea un inicializador en AGTLibrary que reciba el JSON como un NSData y carga los datos de ahí.

Tabla de libros

La biblioteca (AGTLibrary) se debe de mostrar en una tabla con celdas personalizadas. Piensa en qué datos deseas mostrar en dicha celda y cómo. Necesitarás un controlador AGTLibraryTableViewController.

Dicha tabla debe de tener tantas secciones como distintos tags haya en tu biblioteca y una más, la de libros favoritos (Favorites). Las secciones deben de aparecer de la siguiente manera:

- primero Favorites
- luego todos los demás en orden alfabético.

El ser o no favorito se indicará mediante una propiedad booleana de AGTBook (isFavorite) y esto se debe de guardar en el sistema de ficheros y recuperar de alguna forma. Es decir, esa información debe de persistir de alguna manera cuando se cierra la App y cuando se abre.

¿Cómo harías eso? ¿Se te ocurre más de una forma de hacerlo? Explica la decisión de diseño que hayas tomado.

Cuando cambia el valor de la propiedad isFavorite de un AGTBook, la tabla deberá reflejar ese hecho.

¿Cómo lo harías? Es decir, ¿cómo enviarías información de un AGTBook a un AGTLibraryTableViewController? ¿Se te ocurre más de una forma de hacerlo? ¿Cual te parece mejor? Explica tu elección.

Nota: para que la tabla se actualice, usa el método `reloadData` de `UITableView`. Esto hace que la tabla vuelva a pedir datos a su `dataSource`. ¿Es esto una aberración desde el punto de rendimiento (volver a cargar datos que en su mayoría ya estaban correctos)? Explica por qué no es así. ¿Hay una forma alternativa? ¿Cuándo crees que vale la pena usarlo?

Controlador de libros

Crea un controlador que muestra la información de un libro:

- el título
- la portada en grande
- los tags
- en general, todo excepto el pdf.

Debe de contar con algún elemento para marcarlo como favorito o no. Es decir, ahí es donde el usuario debe de poder marcar un libro como favorito o no.

También ha de contar con un botón para mostrar el pdf. El PDF se mostrará en otro controlador y tendrás que hacer un push.

Pon ambos controladores (el de tabla y el de libro en un `UISplitViewController`). Cuando el usuario selecciona un nuevo libro en la tabla, el controlador de libro debe de actualizarse.

Controlador de PDF: `AGTSimplePDFViewController`

Para mostrar un PDF, de momento usaremos una `UIWebView` y su método

```
func loadData(_ data: NSData,  
             mimeType: MIMEType: String,  
             textEncodingName textEncodingName: String,  
             baseURL baseURL: NSURL)
```

Busca en la documentación o en StackOverflow cómo usarlo para cargar un PDF.

Cuando el usuario cambia en la tabla el libro seleccionado, el `AGTSimplePDFViewController` debe de actualizarse. ¿Cómo lo harías?

App Universal

Haz la versión iPhone del lector.

Extras

- a ¿Qué funcionalidades le añadirías antes de subirla a la App Store?
- b Ponle otro nombre e intenta subir esta primera versión a la App Store. Como aun no tienes idea de diseño, [aquí tienes plantillas gratis y de pago](#)
- c Usando esta App como "plantilla", ¿qué otras versiones se te ocurren? ¿Algo que puedas monetizar?

Es vital que vayas teniendo un portfolio de Apps y código para mostrarlo a potenciales empleadores, así que saca tu App del armario y súbela a la App Store.

Forma de Entrega

Envío mediante [formulario de entrega de prácticas](#)

Debes de incluir tanto el proyecto como un fichero llamado README.md (lo de md significa [Markdown](#)) donde debes de responder a las preguntas que se hacen en este documento así como añadir cualquier comentario que consideres oportuno.

Recursos

Enlace al [JSON](#):