



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Una aplicación móvil para mejorar la integración de los estudiantes de intercambio en la ETSINF

PROYECTO FINAL DE CARRERA

Ingeniería Informática

*Autor:* Alberto Sáez Requena

*Director:* Prof. Juan Carlos Ruiz García

6 de septiembre de 2013

## Resumen

En este proyecto final de carrera se desarrolla una aplicación para la plataforma *iOS* con el fin de contribuir a la integración de los estudiantes de intercambio que van a estudiar a la ETS. de Ingeniería Informática de la Universitat Politècnica de València.

Para ello se estudiaron los requisitos que una aplicación de ese tipo podía tener y se desarrolló mediante Xcode una aplicación para iOS que se basa principalmente en la Guía de Intercambio proporcionada por la UPV a dichos alumnos.

Los resultados obtenidos son muy satisfactorios, teniendo una aplicación preparada y lista para su distribución en la tienda de aplicaciones móviles *AppStore* de Apple.

*Palabras clave:* iOS, aplicación, erasmus, intercambio, etsinf, upv

# Índice

Índice de figuras . . . . .	I
<b>1 Introducción . . . . .</b>	<b>1</b>
1.1 Motivación . . . . .	1
1.1.1 Motivación personal . . . . .	2
1.2 Objetivos . . . . .	2
1.3 Organización . . . . .	2
1.3.1 iOS . . . . .	3
<b>2 Estado del arte . . . . .</b>	<b>5</b>
<b>3 Análisis de requisitos . . . . .</b>	<b>7</b>
3.1 Requisitos funcionales . . . . .	7
3.2 Requisitos no funcionales . . . . .	10
<b>4 Desarrollo . . . . .</b>	<b>11</b>
4.1 Home . . . . .	13
4.1.1 Guía de Intercambio . . . . .	14
4.1.2 Otra información . . . . .	16
4.2 Mapa . . . . .	17
4.3 Información académica . . . . .	20
4.4 ETSINF . . . . .	21
4.5 Contacto . . . . .	27
4.6 Otros aspectos . . . . .	30
4.6.1 Localización . . . . .	30
4.6.2 NSLocalizedString . . . . .	33
4.6.3 NSUserDefaults . . . . .	35
4.6.4 Ocultar y mostrar barra de navegación . . . . .	37
4.6.5 Botones, iconos y diseño corporativo . . . . .	38
<b>5 Pruebas . . . . .</b>	<b>39</b>
5.1 'Static Analyzer' . . . . .	40
5.2 'Instruments' . . . . .	41
<b>6 Conclusiones . . . . .</b>	<b>42</b>
6.1 Trabajo futuro . . . . .	43
<b>Bibliografía . . . . .</b>	<b>45</b>
<b>Glosario . . . . .</b>	<b>47</b>

## Índice de figuras

1	Capas de <i>iOS</i> . . . . .	3
2	App's de universidades . . . . .	6
3	Imagen de carga de 'ETSINF-UPV' . . . . .	10
4	UITabBar de 'ETSINF-UPV' . . . . .	12
5	'Navigation Bar' . . . . .	12
6	Vista inicial de la aplicación, 'Home' . . . . .	13
7	Título de una sección de <b>UITableView</b> . . . . .	14
8	Secciones de la Guía de Intercambio . . . . .	15
9	Botón para mostrar información sobre 'Valenbisi' . . . . .	16
10	Vista 'Valenbisi' . . . . .	16
11	Vista 'Mapa' . . . . .	17
12	Puntos de interés . . . . .	18
13	'Segmented Control' . . . . .	18
14	Mapas ETSINF . . . . .	19
15	Mapas Valencia . . . . .	19
16	Vista 'Información académica' . . . . .	20
17	Vista 'ETSINF' . . . . .	21
18	Vista 'Entidades' . . . . .	22
19	Vista 'Bares' . . . . .	23
20	Vista 'Imprimir' . . . . .	24
21	Vista 'Asignaturas' . . . . .	25
22	Vista 'Asignaturas en Inglés' . . . . .	26
23	Vista 'Asignaturas - Léeme' . . . . .	26
24	Vista 'Contacto' . . . . .	27
25	<b>UIWebView</b> y <b>UITextView</b> de la vista 'Contacto' . . . . .	28
26	Mensaje de error cuando no se puede acceder a la información del servidor . . . . .	30
27	Preferencias del sistema donde se puede cambiar el idioma del dispositivo . . . . .	31
28	Forma en la que visualiza <i>Xcode</i> los archivos localizados . . . . .	32
29	Resultado de localizar <i>Localizable.strings</i> . . . . .	34
30	'Home' de la aplicación en inglés . . . . .	35
31	Icono principal de ETSINF-UPV . . . . .	38
32	Ejemplo de botón para volver a la vista anterior . . . . .	38
33	Icono de la pestaña 'ETSINF' . . . . .	38
34	' <i>iPhone Simulator</i> ' versión 6.0 . . . . .	39
35	<i>Plugins</i> de <i>Instruments</i> . . . . .	41



# 1. Introducción

## 1.1. Motivación

El imparable crecimiento del uso de los *smartphones* y las tabletas está cambiando radicalmente la forma en la que se consumen los contenidos. Desde la prensa escrita hasta las grandes editoriales de libros difunden desde hace tiempo una versión online de todas sus publicaciones.

El rápido acceso a la información, el bajo coste y la facilidad de mantenimiento se postulan como los principales factores de éxito de los dispositivos móviles en la sociedad.

En este contexto, en el que las publicaciones en papel comienzan a estar obsoletas, no sólo las editoriales se han subido al carro de las difusiones online, sino que distintas empresas y organizaciones han aprovechado para, ya sea en forma de *ebook* o de aplicación, lanzar al público sus contenidos y publicaciones.

Si nos centramos en el área de las organizaciones, en concreto en la UPV, encontramos un claro ejemplo de como todas las instituciones tienen como mínimo una aplicación móvil con información sobre los servicios que ofrece.

Con ese fin, se ha visto interesante la realización de una aplicación para la plataforma *iOS* que sirva de complemento a la tradicional *Guía de Intercambio* que entrega la universidad a los estudiantes de intercambio que vienen a estudiar a la UPV, concretamente a los que vienen a la ETS. de Ingeniería Informática (ETSINF).

Las ventajas son obvias: el coste de impresión de la *Guía de Intercambio* pasará a ser cero, mientras que los estudiantes sacarán un mayor provecho de dicha información, pues podrán llevarla siempre encima.

Uno de los mayores retos a la hora de realizar la aplicación, más allá de los requisitos técnicos, consiste en cómo adaptar la inmensa cantidad de información disponible en la *Guía de Intercambio* para que sea consumible en un *smartphone*, de manera que resulte clara y accesible para el usuario.

### 1.1.1. Motivación personal

Debido a que el autor ha trabajado como becario en la Oficina de Relaciones Internacionales de la ETSINF durante más de un año, conoce las dudas más frecuentes entre los estudiantes de intercambio. La motivación personal a la hora de elegir y desarrollar este proyecto es la de ayudar al estudiante a sumergirse no sólo en un nuevo ambiente universitario, sino también en una ciudad y una cultura a veces muy diferente a la de procedencia.

## 1.2. Objetivos

El proyecto tiene como objetivo el desarrollo de una aplicación móvil para la plataforma *iOS*, que se distribuirá en la tienda de aplicaciones *AppStore* de manera gratuita. Dicha aplicación está enfocada directamente a los estudiantes de intercambio de la ETSINF, tanto a los que vienen de fuera de España (*ERASMUS* y *PROMOE*) como los que vienen de dentro (*SICUE*). Los objetivos específicos que debe cumplir la aplicación son los siguientes:

- Contener de manera clara y legible la información disponible actualmente en la *Guía de Intercambio*.
- Localizar al usuario mediante el dispositivo GPS del *smartphone* y ubicarlo en el mapa.
- Disponer de información académica específica de la ETSINF.
- Contener planos e información útil tanto de la UPV como de servicios de transporte públicos de Valencia.
- Proporcionar un contacto directo con la Oficina de Relaciones Internacionales de la ETSINF.
- Optimizar los recursos del dispositivo durante la ejecución de la misma.
- Cumplir con los estándares de accesibilidad y diseño establecidos por Apple.

## 1.3. Organización

Para el desarrollo del proyecto se ha elegido la plataforma *iOS* frente a otras alternativas como *Android* o *Windows Phone*. La principal razón de esta elección es la motivación del autor por aprender el lenguaje *Objective-C* (lenguaje utilizado tanto en *iOS* como en *Mac OS X*), así como la creciente demanda por parte de las empresas de desarrolladores de dicha plataforma.

Existen en el mercado multitud de opciones para desarrollar para *iOS*. Sin embargo, se ha optado por la que resulta más cómoda, fácil, accesible y rápida: utilizar el *IDE* Xcode ofrecido por Apple. Este solo puede ser ejecutado en un ordenador Mac con *Mac OS X*. Por tanto, las herramientas utilizadas han sido:

- *iPhone 4S con iOS 6.1.3*
- *MacBook Pro mediados 2010 con Mac OS X 10.8.4 y Xcode 4.6.1*

Para el diseño y adaptación de iconos, imágenes de carga y logotipos se han usado los programas *Adobe Photoshop CS6*, *Adobe Illustrator CS6* y *Gimp 2.8*. Con el fin de poder trabajar de una manera mas cómoda y productiva en Xcode, se ha añadido al *IDE* el *plugin XVim* [17], que emula el comportamiento del editor *Vim* en Xcode.

### 1.3.1. iOS

El sistema operativo *iOS* [3] [4] es el sistema operativo móvil desarrollado por Apple para su gama de dispositivos móviles. Es un sistema Unix, basado en *Mac OS X* pero optimizado para interactuar con multitud de gestos táctiles. Tiene un núcleo híbrido llamado *XNU*, adquirido y usado por Apple para sus sistemas operativos. Como puede apreciarse en la figura 1, su arquitectura se divide en capas.

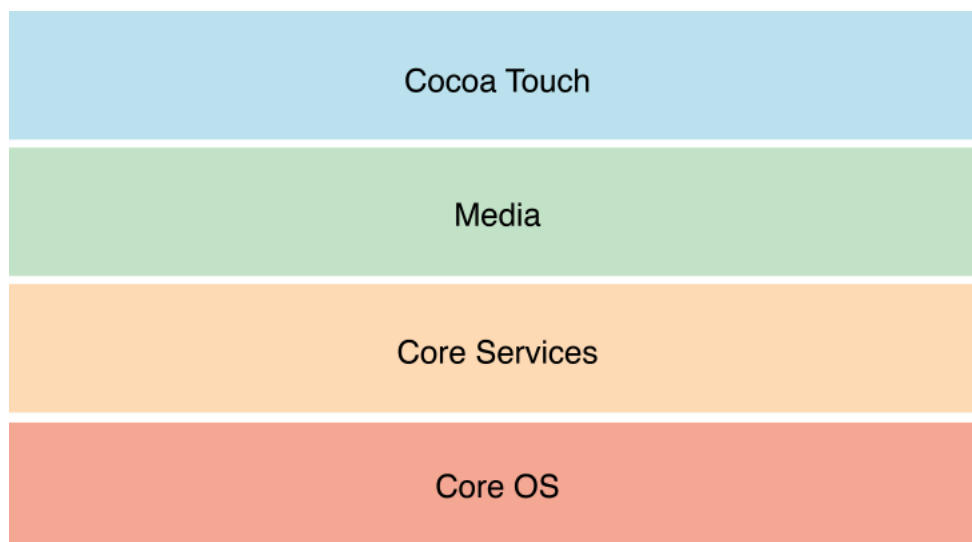


Figura 1: Capas de *iOS*

**Core OS (Núcleo del SO) :** La capa inferior de *iOS* contiene el núcleo del sistema operativo, así como el sistema de ficheros, mecanismos de seguridad y diversos controladores del dispositivo.

**Core Services (Servicios del Núcleo) :** Los frameworks de este nivel contienen los servicios fundamentales que usan todas las aplicaciones. Proporcionan servicios tales como bases de datos, servicios de red, protección de datos, localización, teléfono y preferencias del sistema entre otros.

**Media :** Esta capa contiene todas las tecnologías multimedia relacionadas con los gráficos, el audio y el video. Depende directamente de la capa inferior *Core Services* y proporciona diversos servicios multimedia a la capa superior *Cocoa Touch*.

**Cocoa Touch :** Es el nivel más alto del sistema operativo, donde se encuentran los frameworks de alto nivel entre los que destacan *Foundation Kit*, *UIKit*, *Game Kit*, *iAd*, *MapKit*. Estos soportan directamente las aplicaciones basadas en *iOS*. En *Cocoa Touch* encontramos también tecnologías clave como la multitarea, el soporte táctil, las notificaciones *push* y otros servicios de alto nivel.

Al desarrollar para *iOS*, lo normal es investigar los frameworks ofrecidos en este nivel para ver si cumplen las expectativas. Conforme se vaya requiriendo un mayor control sobre el sistema, se irán bajando capas del sistema operativo para ver si los frameworks ofrecidos en cada una de ellas cumplen con los requisitos deseados.

En *iOS* existen fundamentalmente dos tipos de aplicaciones: las nativas, desarrolladas en *Objective-C* y ejecutadas directamente sobre *iOS* y las aplicaciones web (*web apps*), desarrolladas mediante *HTML* y *CSS*, ejecutadas mediante el navegador web y solo disponibles con conexión a internet. Se ha decidido desarrollar la aplicación 'ETSINF-UPV' de manera nativa, para poder sacar partido a los frameworks disponibles en Cocoa, así como obtener una mayor velocidad de ejecución y un menor consumo de datos en la aplicación.

Una característica importante a destacar de las aplicaciones desarrolladas para *iOS*, es lo que Apple denomina "*Sandbox*" (Caja de Arena) [6]. Por razones de seguridad, *iOS* aísla cada aplicación en su propia 'caja' y no le permite comunicarse con otras aplicaciones, concediéndole un acceso limitado a los recursos del sistema. Esta caja es realmente una carpeta, de estructura

conocida, donde el sistema operativo guarda todos los datos relacionados con la aplicación.

## 2. Estado del arte

Tal y como se ha comentado anteriormente, cada vez más instituciones cuentan con una o varias aplicaciones móviles que permiten a los usuarios interactuar con ellas y ver los servicios que ofrece, tal y como si se accediese desde un ordenador. Basta con una simple búsqueda en la *AppStore* para encontrar cientos de aplicaciones de distintas organizaciones, muchas de ellas universidades.

Con el fin de poder analizar dichas aplicaciones se ha procedido a descargar del *AppStore* distintas aplicaciones de universidades españolas así como algunas de las universidades con mas prestigio del planeta (Figura 2):

- **UPV** - Universitat Politècnica de València.
- **Cambridge** - University of Cambridge .
- **Texas** - University of Texas at Austin.
- **UGR** - Universidad de Granada.
- **Univ.Navarra** - Universidad de Navarra.
- **UJaen** - Universidad de Jaén.
- **iStanford** - Standford University.
- **Harvard** - Harvard University.



Figura 2: App's de universidades

Después de probar e investigar cada una de ellas, se llegó a la conclusión de que la mayoría cuentan con los mismos aspectos en común: noticias, eventos, historia, intranet y otros servicios específicos de cada universidad. Cabe destacar que la UPV ya cuenta con una aplicación propia 'UPV' [18], en la que se puede encontrar noticias, lugares, eventos y localizar edificios mediante realidad aumentada. Debido a esto, en la aplicación a desarrollar no se va a repetir ninguna funcionalidad ya existente en la aplicación 'UPV'.

Si bien es cierto que existen multitud de aplicaciones de universidades con multitud de servicios, no se ha encontrado una aplicación enfocada directamente a la ayuda e integración de los estudiantes de intercambio. Por ello, aunque para diseñar la aplicación se han tenido en cuenta algunos aspectos de las aplicaciones previamente descargadas, la mayoría de la estructura y los contenidos de la aplicación son fruto del trabajo del autor y de su experiencia como becario en la Oficina de Relaciones Internacionales de la ETSINF.

Aunque en la gran mayoría de las aplicaciones se pueden encontrar apartados para acceder a la intranet del alumno, consultar el correo personal, etc., estos requieren de un trabajo de forma interna con la universidad. Al realizar

este proyecto de forma externa, no se cuenta con muchos de los recursos con los que cuentan instituciones como las mencionadas anteriormente.

Aun así, se va a tratar de desarrollar una aplicación estable, rápida y con contenido accesible para intentar de ayudar lo máximo posible a los futuros alumnos de intercambio de la ETS. de Ingeniería Informática de la Universitat Politècnica de València.

### 3. Análisis de requisitos

En esta sección se van a desarrollar en profundidad los objetivos explicados en la sección 1.2, a partir de los cuales comenzará el desarrollo de la aplicación.

#### 3.1. Requisitos funcionales

Los requisitos funcionales en los que se debe basar la aplicación, agrupados por bloques, son los siguientes:

- **Guía de intercambio** :

- **Contener la 'Guía de Intercambio'** : La aplicación deberá contener toda la información recogida en la 'Guía de Intercambio' entregada por la UPV a los estudiantes. Dicha guía tiene los siguientes contenidos:

1. *La ciudad de Valencia.*
2. *La Universitat Politècnica de València*
3. *Preparando tu estancia en la Universitat Politècnica de València*
4. *Servicios en la Universitat Politècnica de València*
5. *Información de utilidad*

- **Mapas** :

- **Mostrar un mapa de la ETSINF** : La aplicación debe mostrar un mapa con los lugares mas importantes de la ETSINF, tales como entidades, oficinas, biblioteca, etc.
- **Mostrar un mapa de Valencia** : Junto con el mapa de la ETSINF, se deberá dar la posibilidad de mostrar un mapa de Valencia, para ubicar al estudiante en los primeros días de la estancia.

- **Localización** : Si el dispositivo dispone de los requisitos técnicos disponibles, la aplicación debe ser capaz de localizar la posición del usuario en el mapa.
- **Posibilidad de cambiar entre diferentes tipos de mapa** : Se deberá dar la posibilidad al usuario de elegir entre los tres diferentes tipos de mapas ofrecidos en el framework *MapKit* de Apple: Estándar, Satélite y mezcla de ambos.
- **Información académica** :
  - **Disponer del calendario académico** : En la aplicación se deberá tener acceso al documento proporcionado por la ETSINF donde está marcado el calendario académico.
  - **Disponer del calendario de exámenes** : También deberá ser posible visualizar el documento donde se muestran todos los exámenes de todas las titulaciones de la ETSINF.
- **Planos** :
  - **Contener planos e información útil** : Se deberá incluir planos tanto de la propia UPV como de diferentes transportes públicos de Valencia (*Metro, Valenbisi, etc.*), para facilitar lo más posible al estudiante la movilidad en los primeros días.
  - **Explicar los servicios** : Debido a que hay algunos servicios disponibles en Valencia que, al contrario que el metro por ejemplo, no están disponibles en todas las ciudades del mundo, se deberá buscar la forma de explicar al usuario en que consiste ese servicio, proporcionándole acceso a la página web del mismo si es necesario.
- **ETSINF** :
  - **Disponer de información sobre entidades** : Se podrá visualizar una lista de las entidades más significativas que hay en la ETSINF, con una explicación de cada una, localización y horarios o estudios que imparte según sea el caso.
  - **Mostrar información sobre puntos de interés** : También se deberá mostrar diferentes puntos que no figuran tan explícitamente en el resto de información disponible de la escuela, como cafeterías, reprografía, bares o delegaciones de alumnos.
  - **Explicar distintos servicios de la ETSINF** : Entrar en detalle sobre los servicios que ofrece la ETSINF a los estudiantes, como el aula libre, tarjeta de impresión o secretaría entre otros.



- **Contacto** :

- **Proporcionar contacto con la oficina internacional** : Se deberá poder contactar directamente con la oficina internacional vía e-mail. En el asunto quedará reflejado que el mail se envía desde la aplicación.
- **Mostrar un horario actualizado de la oficina internacional** : La aplicación se conectará a un servidor web donde el personal de la oficina internacional de la ETSINF podrá subir el horario actualizado de la oficina, para evitar malentendidos en los horarios especiales de vacaciones y ahorrar tiempos de espera a los estudiantes. Si no se dispone de conexión a internet, la aplicación guardará el último horario descargado para mostrarlo sin conexión. En el caso de que nunca se haya descargado ningún horario, la aplicación mostrará el horario por defecto.
- **Mostrar redes sociales** : Las principales redes sociales en las que aparece la ETSINF deberán mostrarse en la aplicación, para fomentar que los alumnos estén en contacto con las noticias de la escuela. Estas serán inicialmente Twitter y Facebook.
- **Enlace al blog** : También se deberá disponer de un enlace al blog de la Oficina de Relaciones Internacionales, ya que es ahí donde aparecen las noticias relacionadas con los estudiantes de intercambio.

- **Otros** :

- **Imagen de carga corporativa** : Se denomina imagen de carga (Figura 3) a la imagen que utiliza *iOS* para mostrar en la pantalla mientras se cargan los recursos de la aplicación, dando *feedback* al usuario de que su orden de abrir la aplicación seleccionada ha tenido lugar. De otra manera, el usuario podría pensar que nada está sucediendo o que la aplicación no funciona y desinstalarla.



Figura 3: Imagen de carga de 'ETSINF-UPV'

- **Guardar preferencias del usuario** : La aplicación deberá ser capaz de guardar determinadas preferencias seleccionadas por el usuario en el uso de la aplicación, por ejemplo, el tipo de mapa que ha seleccionado.
- **Localización** : Puesto que la aplicación está enfocada a estudiantes de otras universidades, el lenguaje primario será el inglés. La aplicación deberá ser capaz de detectar el idioma del dispositivo. Si el idioma es español, la aplicación cargará todos los contenidos en español. Si no lo es, cargará los contenidos en inglés.

### 3.2. Requisitos no funcionales

Los requisitos no funcionales para la aplicación, es decir, los que no especifican el comportamiento del sistema, son:

- **Rendimiento** : La aplicación debe comportarse de manera fluida, sin retardos que entorpezca la experiencia del usuario. Se deberá buscar el equilibrio entre rendimiento y consumo de manera que la experiencia resulte lo más positiva posible para el usuario.
- **Disponibilidad** : La aplicación debe estar disponible mediante la tienda de aplicaciones *AppStore* de manera gratuita.
- **Accesibilidad** : Debe cumplir con los estándares de accesibilidad y diseño establecidos por Apple. Para ello, la aplicación se asemejará gráficamente a otras muchas ya disponibles en la *AppStore*.

- **Usabilidad** : Cualquier usuario sin formación previa debe ser capaz de navegar por la aplicación y acceder a todos sus contenidos sin restricciones. Para ello se usarán iconos claros, tamaños de letra adecuados e imágenes para mejorar la calidad visual de la aplicación.
- **Estabilidad** : La aplicación debe ser capaz de gestionar los errores que se produzcan durante su ejecución, avisando al usuario del error producido y la causa.
- **Mantenibilidad** : La aplicación deberá poder ser actualizada con mejoras de estabilidad, corrección de errores y nuevas prestaciones.
- **Interfaz** : Debe ser clara, sencilla e intuitiva.
- **Integración** : La aplicación deberá integrarse perfectamente en la plataforma *iOS*, tanto a nivel gráfico como en entornos de multitarea, segundo plano, etc. Si se necesita, hará uso de los distintos frameworks ofrecidos por el sistema, como enviar correos, abrir enlaces web, etc.
- **Optimización** : Tanto el consumo de batería como el de datos deberán ser tenidos en cuenta a la hora de desarrollar la aplicación. Para ello, se tendrá muy en cuenta la gestión de memoria en tiempo de ejecución así como las conexiones que realice la aplicación a través de internet.

## 4. Desarrollo

Para el desarrollo de la aplicación, la primera tarea a realizar ha sido agrupar toda la información que debe contener en bloques. Estos bloques pasarán posteriormente a ser las secciones (pestañas o *tabs*) de la aplicación, y son:

1. Home
2. Mapa
3. Información Académica
4. ETSINF
5. Contacto

Para una mayor comodidad y facilidad de uso, se ha optado por estructurar la aplicación gráficamente mediante el elemento **UITabBar** (figura 4). Este elemento se puede encontrar en aplicaciones del sistema como 'Teléfono' o 'Música'. Consiste en una barra en la parte inferior de la pantalla con iconos y texto que permite la navegación entre distintas vistas de manera fluida.



Figura 4: UITabBar de 'ETSINF-UPV'

Dentro de cada una de las vistas que nos proporciona la **UITabBar**, podemos insertar cualquier tipo de elemento visual que queramos que queramos: tablas, mapas, imágenes o incluso otras vistas. Por ello, hemos adaptado cada una de las *tabs* según las necesidades de la información que contienen. Las pestañas 'Home', 'Información académica' y 'ETSINF' contienen cada una el elemento llamado **UINavigationController** (figura 5)). Este proporciona una barra en la parte superior de la pantalla que permite navegar y manejar las distintas vistas que contiene, volver atrás, etc.

*iOS* crea una jerarquía de vistas conforme el usuario va interactuando con la aplicación. Es una especie de pila, de la cual selecciona la vista que se encuentra mas arriba. Para el manejo de la misma, se utilizan los métodos *pushViewController:animated:* y *popViewController:animated:* de la clase **UINavigationController**. Estos van 'pusheando' o 'popeando' vistas de la pila y estableciendo conexiones entre ellas para poder volver atrás.



Figura 5: 'Navigation Bar'

## 4.1. Home

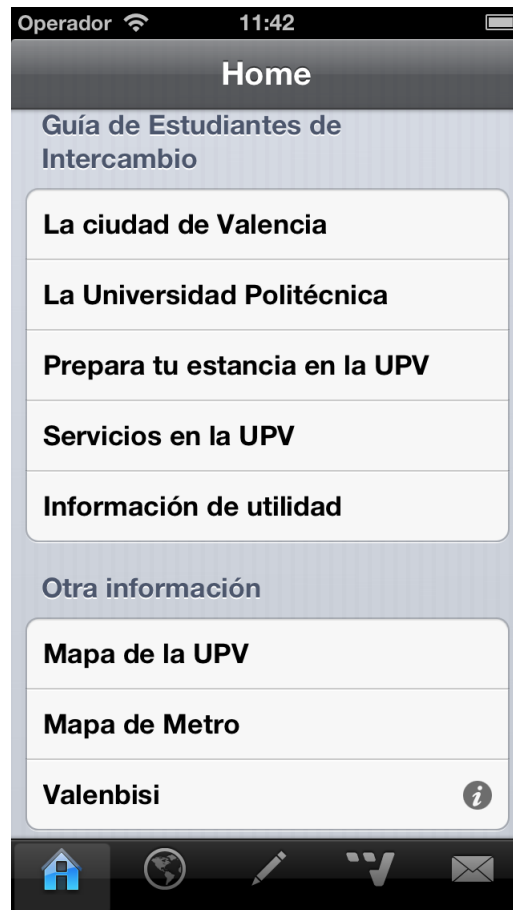


Figura 6: Vista inicial de la aplicación, 'Home'

El apartado *Home* (figura 6) es la vista principal de la aplicación. Es la pantalla que se muestra cuando la aplicación se inicia. En ella, podemos encontrar dos grandes secciones: 'Guía de Intercambio' y 'Otra información'. Estas constituyen dos *secciones* del elemento **UITableView**. Cada una de las secciones cuenta con un título informativo sobre la información que contienen (*tableView: titleForHeaderInSection:*) (figura 7).

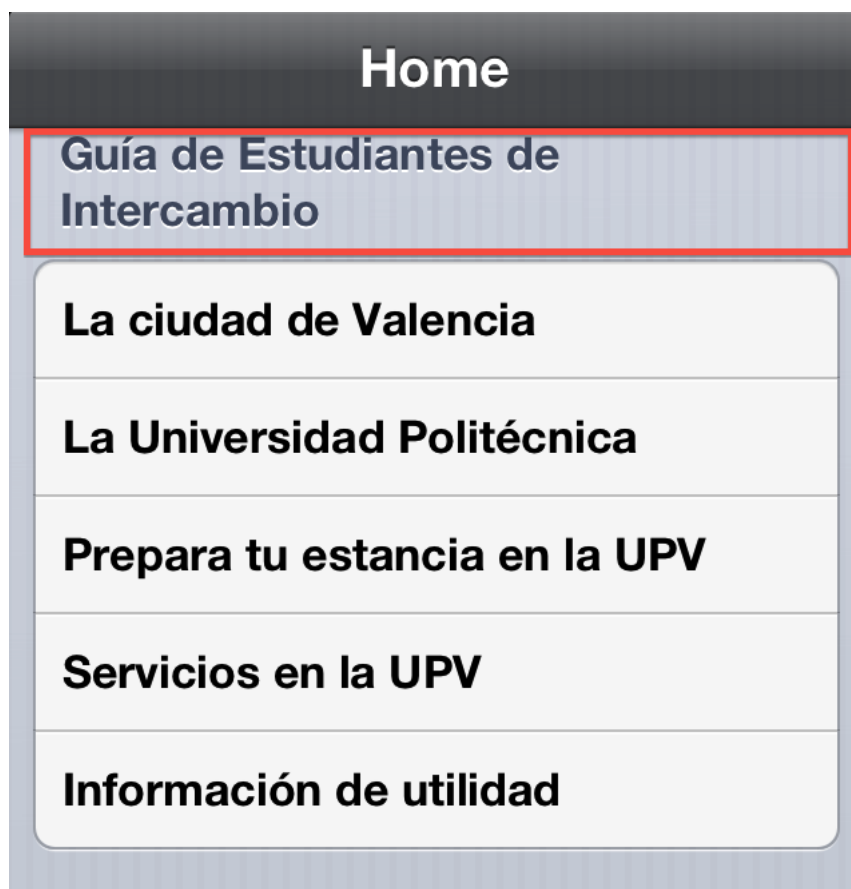


Figura 7: Título de una sección de `UITableView`

#### 4.1.1. Guía de Intercambio

En esta sección de la tabla podremos encontrar la Guía de Intercambio del actual curso académico. Cada celda (`UITableViewCell`) de la tabla se corresponde con cada sección de la Guía de Intercambio original, editada y repartida por la UPV, la cual consta de las siguientes partes:

1. *La ciudad de Valencia*
2. *La Universitat Politècnica de València*
3. *Preparando tu estancia en la Universitat Politècnica de València*
4. *Servicios en la Universitat Politècnica de València*
5. *Información de utilidad*

En la figura 8 se muestran los contenidos correspondientes a las diferentes secciones de la guía. Con el fin de que visualizar la guía de intercambio resulte lo más cómodo y rápido posible, se ha decidido hacer uso de **UITableView's** para expandir las diferentes partes de la misma. De esta forma, el estudiante puede navegar por los apartados de manera más fluida e ir directo al contenido que desea ver.

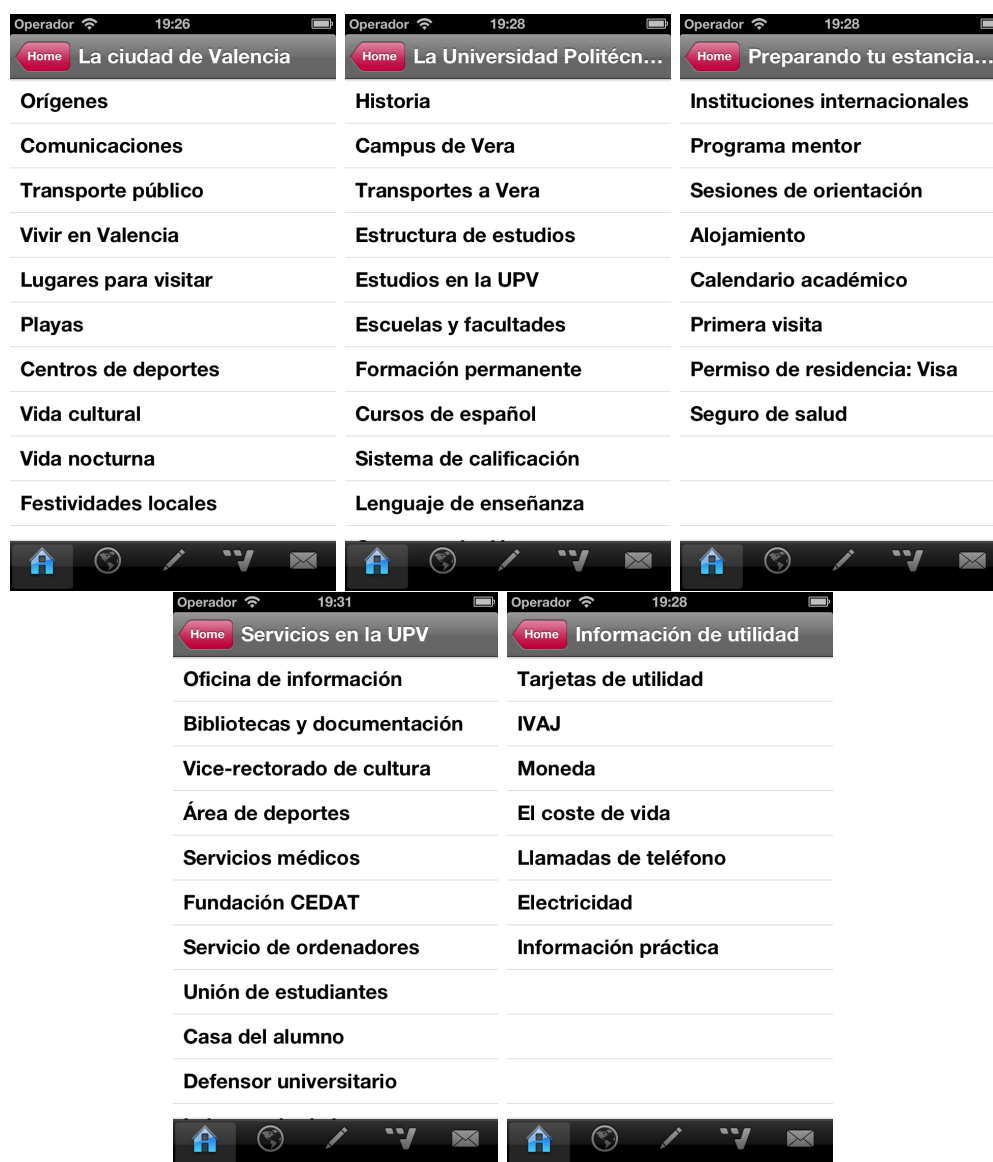


Figura 8: Secciones de la Guía de Intercambio

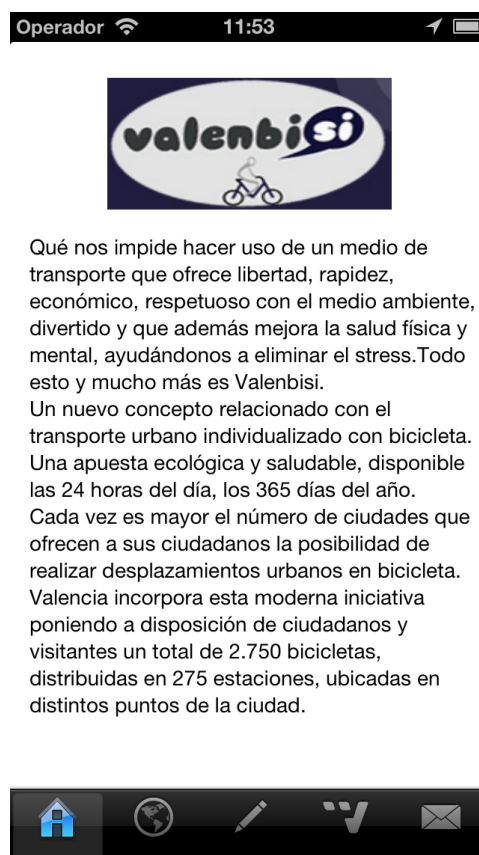
#### 4.1.2. Otra información

Aquí podremos encontrar mapas tanto de la propia UPV como de diferentes transportes públicos de Valencia. Si bien es cierto que la *app* 'UPV' ya cuenta con un mapa de la universidad, se ha visto útil y necesario tenerlo siempre a mano, por lo que también forma parte de esta aplicación.

En cuanto al transporte público, se han proporcionado mapas de los dos principales transportes de la ciudad: el metro y *Valenbisi*. Además, y debido a que *Valenbisi* no es un servicio tan presente como lo puede ser el metro en otras ciudades, se ha añadido un botón (figura 9) que abre una nueva vista con información sobre el servicio y un enlace a su página web (figura 10).



Figura 9: Botón para mostrar información sobre 'Valenbisi'



Qué nos impide hacer uso de un medio de transporte que ofrece libertad, rapidez, económico, respetuoso con el medio ambiente, divertido y que además mejora la salud física y mental, ayudándonos a eliminar el stress. Todo esto y mucho más es Valenbisi. Un nuevo concepto relacionado con el transporte urbano individualizado con bicicleta. Una apuesta ecológica y saludable, disponible las 24 horas del día, los 365 días del año. Cada vez es mayor el número de ciudades que ofrecen a sus ciudadanos la posibilidad de realizar desplazamientos urbanos en bicicleta. Valencia incorpora esta moderna iniciativa poniendo a disposición de ciudadanos y visitantes un total de 2.750 bicicletas, distribuidas en 275 estaciones, ubicadas en distintos puntos de la ciudad.

Figura 10: Vista 'Valenbisi'



## 4.2. Mapa



Figura 11: Vista 'Mapa'

En esta pestaña, se ha hecho uso de los frameworks *MapKit* y *CoreLocation*. Estos proporcionan todas las clases y métodos necesarios para mostrar un mapa en la vista actual, visualizar puntos de interés y anotaciones (figura 12), cambiar entre los distintos tipos de mapa, etc..



Figura 12: Puntos de interés

Como se puede observar en la figura 11, el mapa permite cambiar entre los tres tipos de mapa por defecto que ofrece *iOS*: Estándar, Satélite y una mezcla de ambos. Para ello, y para poder cambiar la vista del mapa desde la ETSINF a Valencia o viceversa, se ha utilizado el elemento **UISegmentedControl** (figura 13). Este elemento envía un mensaje con el número de segmento seleccionado (propiedad *selectedSegmentIndex*), para que el desarrollador actúe en consecuencia.



Figura 13: 'Segmented Control'

En definitiva, como se puede apreciar en las figuras 14 y 15 tenemos la posibilidad de visualizar 6 combinaciones distintas de mapas y vistas:

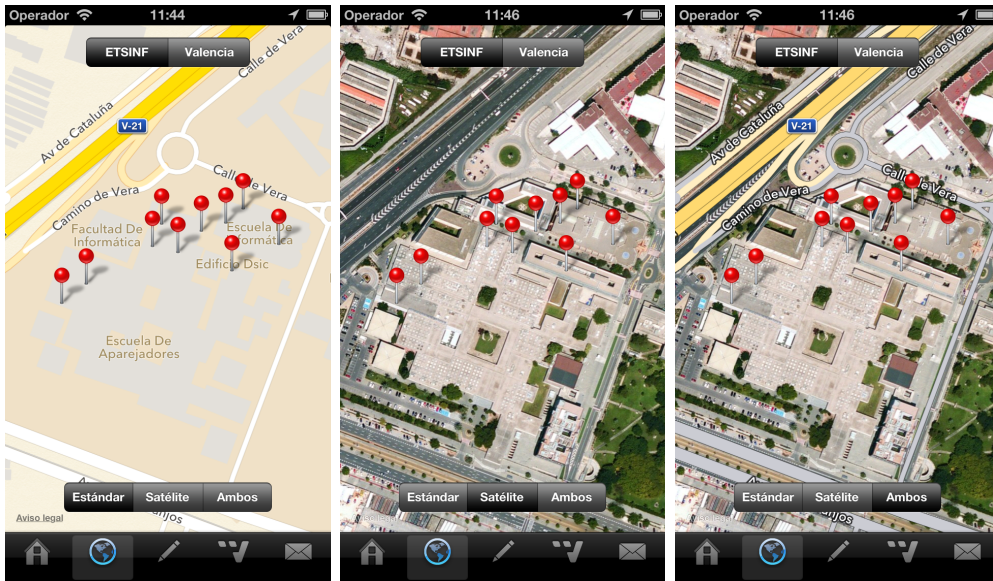


Figura 14: Mapas ETSINF

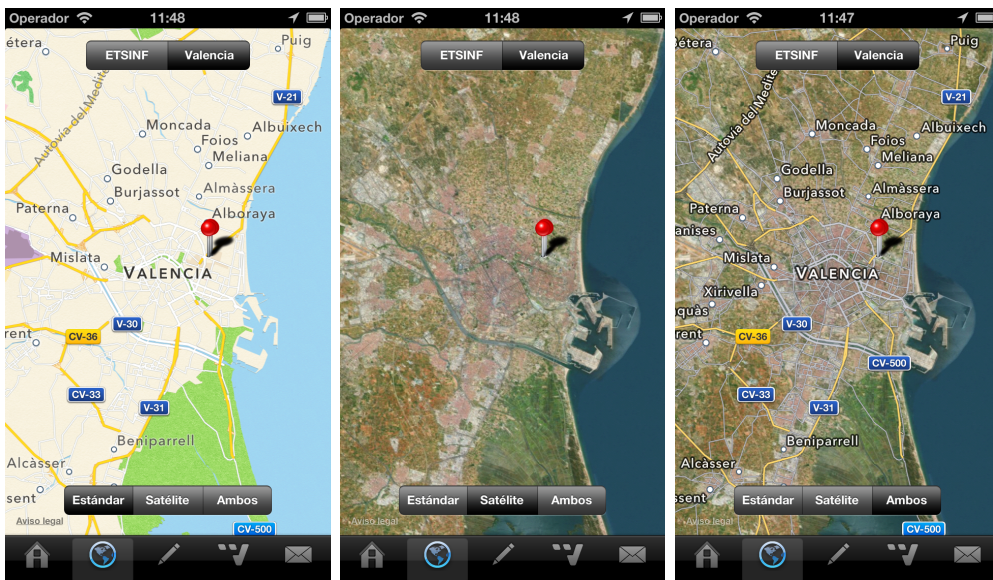


Figura 15: Mapas Valencia

### 4.3. Información académica

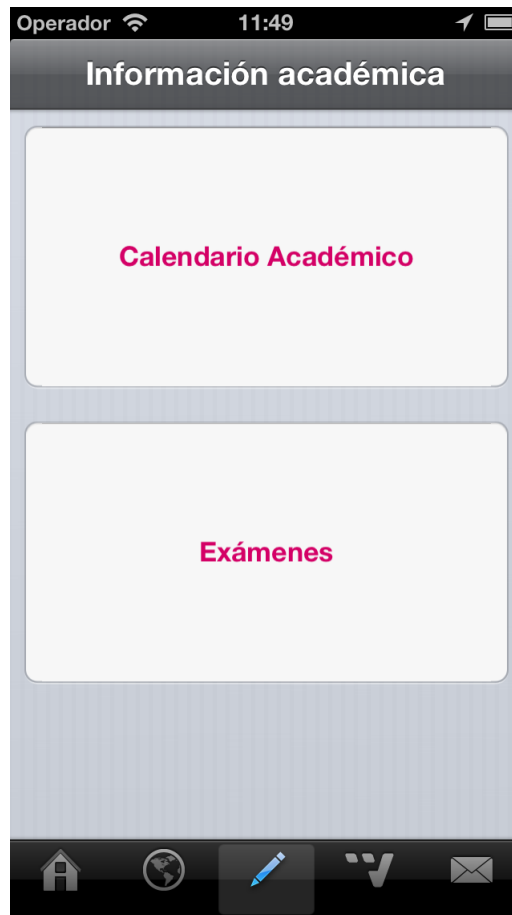


Figura 16: Vista 'Información académica'

En esta *tab* (figura 16) podemos encontrar información académica de la ETS. de Ingeniería Informática. Consta de dos secciones, 'Calendario Académico' y 'Exámenes'. La primera de ellas contiene un PDF del actual curso académico con toda la información disponible sobre:

- Matrícula
- Principio y fin de curso
- Periodos de evaluación
- Plazo de presentación de actas
- Evaluación por currículum

- Días festivos
- Vacaciones
- Calendario ilustrado que refleja las fechas importantes, cambios docentes, etc.

Por otro lado, si pulsamos sobre el botón exámenes, la aplicación mostrará un archivo PDF proporcionado por la ETSINF en el que se muestran todos los exámenes de todas las convocatorias de todas las titulaciones. Dado que a fecha de realización de este proyecto dicho archivo no se encuentra disponible, en la aplicación figura la frase: 'Proximamente'.

#### 4.4. ETSINF

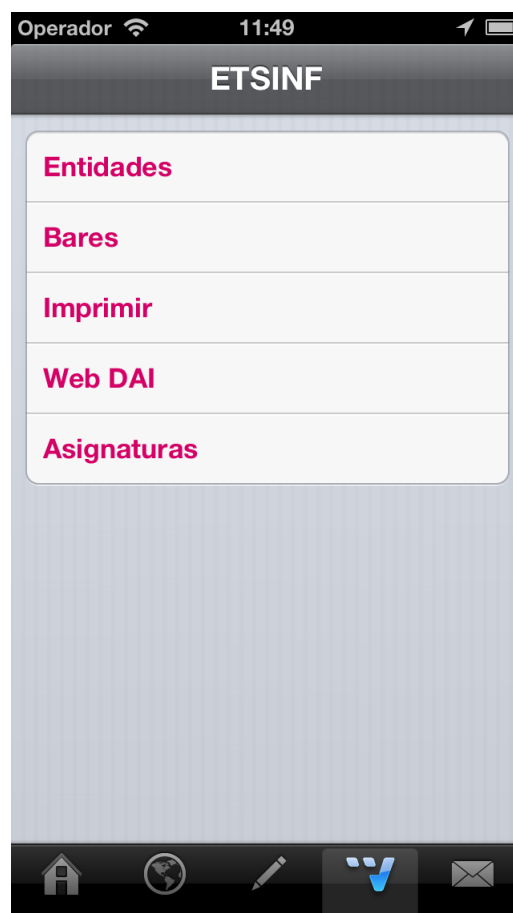


Figura 17: Vista 'ETSINF'

En este apartado encontramos información relacionada con la escuela. Consta de los siguientes puntos:

1. **Entidades**, figura 18 : Donde podremos encontrar información sobre las diferentes entidades de la Universitat Politècnica de València que están estrechamente relacionadas con la ETSINF. De momento, encontramos el *Departamento de Sistemas Informáticos y Computación (DSIC)*, el *Departamento de Informática de Sistemas y Computadores (DISCA)* y la *Biblioteca de computación y documentación 'Enric Valor'*.

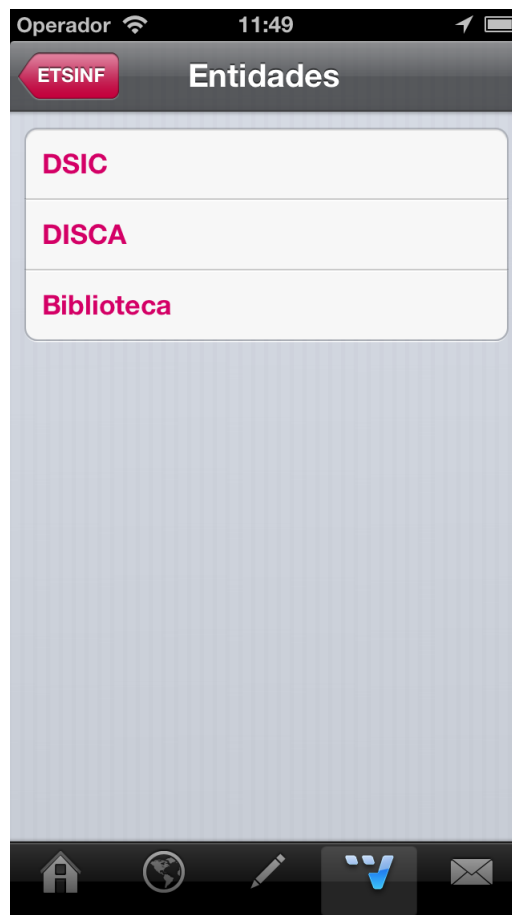


Figura 18: Vista 'Entidades'

2. **Bares**, figura 19 : Vista en la que se puede ver una explicación de la cafetería principal de la ETSINF, '*La Vella*'. Además del horario de la misma, encontramos dos botones en la parte inferior, 'Menú del

día' y 'Bocadillo del día'. Al pulsar cualquiera de ellos, se crea una instancia de **UIWebView** en la que se carga el archivo PDF disponible en <http://www.lavella.es> , con el bocadillo o menú de ese día.



Figura 19: Vista 'Bares'

3. **Imprimir**, figura 20 : Aquí podremos encontrar información sobre las dos opciones principales de las que dispone la ETSINF para imprimir trabajos, documentos y demás. Estas son 'Reprografía' y 'Tarjetas de impresión'.

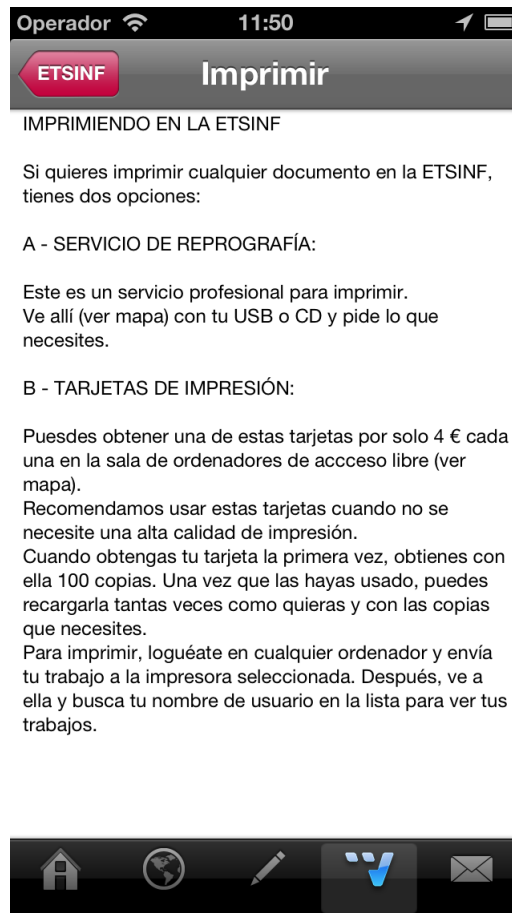


Figura 20: Vista 'Imprimir'

4. **Web DAI** : Al pulsar sobre esta entrada de la tabla, la aplicación utiliza la clase **UIApplication** para abrir en el navegador *Safari* la página web de la Delegación de Alumnos de Informática. En ella, los alumnos pueden encontrar toda la información relacionada con actividades, eventos, semana cultural y demás que organiza su delegación.

La clase **UIApplication** es un punto centralizado que permite controlar como se relacionan las distintas aplicaciones. Basta con obtener la instancia *sharedApplication* e invocar al método *openURL:*, tal y como muestra el siguiente fragmento de código:

```

1 NSURL *url = [NSURL URLWithString:@"http://www.dai.upv.es
   "];
2
3 if (![UIApplication sharedApplication] openURL:url)

```



```
4 NSLog(@"%@%@", @"Error al abrir la URL: ", [url description]);
```

5. **Asignaturas**, figura 21 : En esta vista se puede encontrar información sobre todas las asignaturas de todas las titulaciones que se imparten en la ETSINF, tanto las troncales y obligatorias como las que se ofertan de libre elección. La estructura es una **UITableView** en la que cada celda corresponde con las asignaturas de una titulación.



Figura 21: Vista 'Asignaturas'

Existen dos celdas especiales, 'POR FAVOR LÉEME' y 'Asignaturas en inglés'. Al pulsar en la primera de ellas, aparece una vista (figura 22) en la que se explica brevemente las diferentes titulaciones que existen en la ETSINF. Esto se ha visto necesario ya que el cambio docente que

está teniendo lugar en la universidad al pasar de las antiguas titulaciones a grado, genera normalmente confusión en muchos alumnos de intercambio. Si se pulsa en 'Asignaturas en Inglés', la aplicación muestra un archivo PDF (figura 23) en el que se ven todas las asignaturas con docencia en inglés. Esto facilita la tarea sobretodo a los alumnos que vienen con un nivel bajo de español y buscan materias en inglés para poder seguir bien las clases.

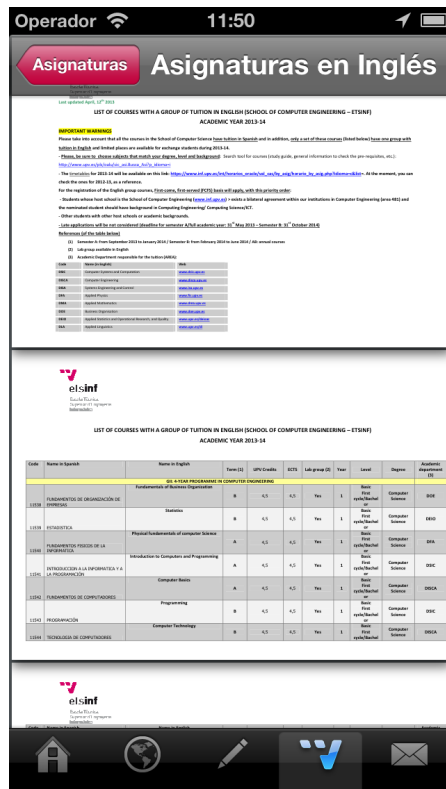


Figura 22: Vista 'Asignaturas en Inglés'

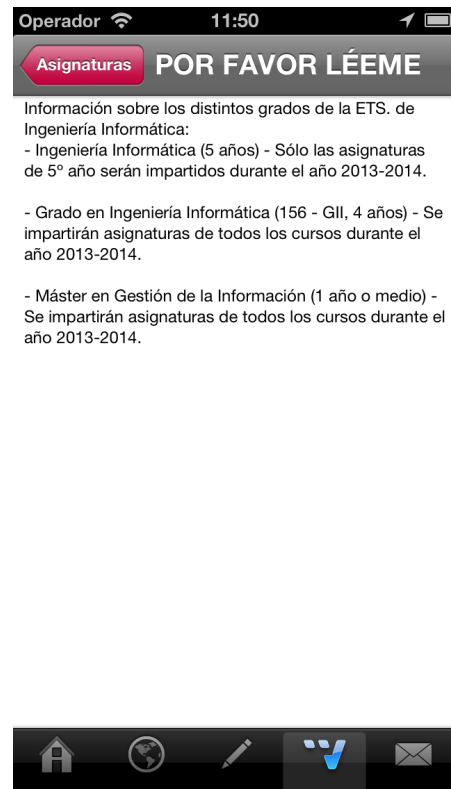


Figura 23: Vista 'Asignaturas - Léeme'

## 4.5. Contacto

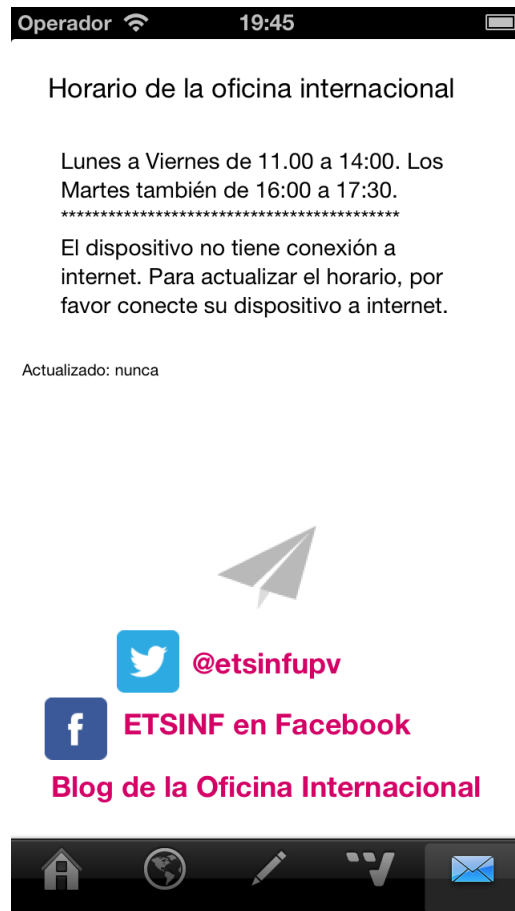


Figura 24: Vista 'Contacto'

En la pestaña de 'Contacto' (figura 24) encontramos diversas formas de contacto con la ETSINF y la oficina de relaciones internacionales de la misma.

Lo principal de esta vista es el horario actualizado de la oficina de relaciones internacionales de la ETSINF. Justo debajo de la frase 'Horario de la oficina internacional' se encuentran dos elementos fundamentales : una **UIWebView** y un **UITextView**. Tal y como se puede ver en la figura 25, estos se encuentran superpuestos.

## Horario de la oficina internacional

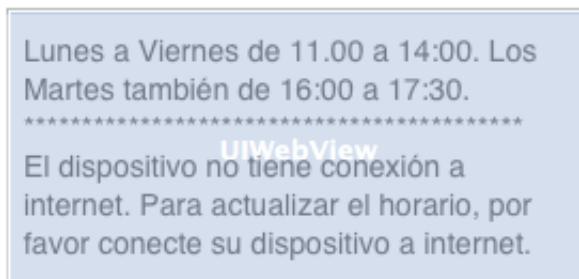


Figura 25: **UIWebView** y **UITextView** de la vista 'Contacto'

A continuación se detalla el proceso que realiza la aplicación internamente:

Cuando el usuario pulsa sobre el *tab* 'Contacto', la aplicación ejecuta como siempre el código del método *(void) viewDidLoad:* y *(void) viewWillAppear:(BOOL) animated* del controlador que se encarga de la vista seleccionada. Estos se ejecutan previamente a que se cargue y previamente a que aparezca la vista respectivamente.

```
1 - (void) viewDidLoad
2 {
3     [super viewDidLoad];
4     [scheduleWebView setDelegate:self];
5     [scheduleWebView setUserInteractionEnabled:NO];
6     [scheduleWebView autoresizingMask=
7     UIViewAutoresizingFlexibleHeight |
8     UIViewAutoresizingFlexibleWidth];
9 }
10 - (void) viewWillAppear:(BOOL) animated
11 {
12     [super viewWillAppear:animated];
13     [scheduleOfflineText setHidden:YES];
14     [scheduleWebView loadRequest:[NSURLRequest requestWithURL
15     :[NSURL URLWithString:@"http://172.20.10.2"]]];
16 }
```

En *iOS*, estos métodos son usados normalmente para inicializar todo tipo de elementos, objetos, etc. que tengan que estar preparados cuando el usuario desea cargar una determinada vista.

En esta ocasión, el método *(void)viewDidLoad:* realiza lo siguiente:

1. Llama al mismo método de la clase superior.
2. Inhabilita la posibilidad de interacción del usuario con la vista web. Esto evita que el usuario pueda desplazar ese recuadro a su antojo, lo que daría una sensación rara en la vista al tapar el resto de elementos.
3. Deja que la propia vista web se encargue de redimensionar automáticamente las subvistas que pudiera contener.

Seguidamente, en *(void)viewWillAppear:(BOOL)animated :*

1. Al igual que en el método anterior, llama al mismo método de la clase superior.
2. Oculta el recuadro **UITextView** (*scheduleOfflineText*) para dejar a la vista la **UIWebView** (*scheduleWebView*) que debe mostrar el horario actualizado.
3. Carga mediante el método *loadRequest:requestWithURL:* el horario de la oficina desde internet <sup>1</sup>

Cada vez que se carga la vista, esta comprueba si la información está disponible (el servidor está *online*). Si es el caso, actualiza la vista de la **UIWebView** y pone la fecha de ese día. En caso contrario, la aplicación muestra un mensaje de error (figura 26), oculta la **UIWebView** y muestra el texto por defecto que contiene **UITextView** que se puede observar en la figura 25.

Esto sólo ocurre la primera vez que se inicia la aplicación, ya que una vez consigue contactar con el servidor, se guarda en las preferencias del usuario la información recibida, junto con la fecha en la que se recibió. A partir de entonces, cuando la información no esté disponible en el servidor, la aplicación carga la última información recibida junto con la fecha en la que se guardó. Este procedimiento se detalla más adelante en la sección 4.6.3.

---

<sup>1</sup>“Aunque en el código figura una dirección IP, esta corresponde a una IP local asignada al ordenador de desarrollo para realizar distintas pruebas sobre el funcionamiento de los diferentes métodos empleados en esta vista. La dirección web final será una acordada con los responsables de la oficina de relaciones internacionales de la ETSINF en la que ellos mismo podrán ir actualizando el horario.”



Figura 26: Mensaje de error cuando no se puede acceder a la información del servidor

En esta pestaña también se dispone de acceso directo tanto a las principales redes sociales con las que cuenta la ETSINF (Facebook y Twitter) como al blog de la oficina de relaciones internacionales [19]. Además existe de un botón con forma de avión para mandar un correo electrónico directamente a la oficina de relaciones internacionales de la ETSINF.

## 4.6. Otros aspectos

### 4.6.1. Localización

Uno de los aspectos importantes del desarrollo a destacar es lo que se conoce en la programación de dispositivos móviles como 'Localización'. Este consiste en hacer accesible una aplicación para diferentes idiomas. Para ello, se generan varias copias de los recursos que se necesite localizar (tantas como

idiomas se deseen añadir) y se le especifica a la aplicación que recurso debe escoger según el idioma del dispositivo en el que se ejecute.

*iOS* permite realizar este proceso de forma extremadamente sencilla. Una vez que se le indica a *Xcode* que se desea localizar una aplicación, este se encarga de crear automáticamente una carpeta dentro del directorio de la aplicación por cada idioma que añadamos. Dichas carpetas, conocidas como directorios *lproj*, son nombradas según el estándar de códigos y regiones establecido en todas las plataformas. El formato del nombre de las carpetas es *código-del-lenguaje\_Región.lproj*, aunque muchas veces se omite la región.

El funcionamiento del sistema operativo a la hora de buscar los recursos es muy simple: Empieza recorriendo el sistema de ficheros de la aplicación desde la raíz buscando el archivo que necesita. Si no lo encuentra en ese directorio, busca en la carpeta *lproj* que se corresponde con el idioma del dispositivo. Este idioma está especificado dentro de las preferencias generales de *iOS*, y puede ser cambiado por el usuario en cualquier momento (figura 27).



Figura 27: Preferencias del sistema donde se puede cambiar el idioma del dispositivo

Por consiguiente, cuando se quiera localizar cualquier archivo, por ejemplo un *.xib* (archivos de interfaz), *Xcode* moverá automáticamente una copia de este dentro de cada directorio *lproj*. Sólo habrá que sustituir en el archivo correspondiente los objetos a localizar (texto, imágenes, etc.) por los objetos traducidos. Una vez que un archivo ha sido localizado, *Xcode* se encarga de que se pueda acceder fácilmente a todas sus versiones en todos los idiomas desglosando el archivo principal, tal y como se puede apreciar en la figura 28.

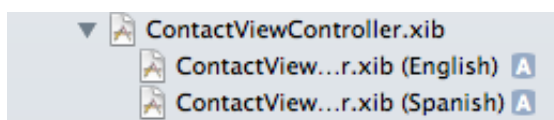


Figura 28: Forma en la que visualiza *Xcode* los archivos localizados

Pero la ayuda proporcionada por *iOS* no acaba ahí. Para que sea más rápido y fácil traducir textos (desde vistas de texto hasta etiquetas o el texto de los botones), el sistema operativo proporciona una herramienta de consola para extraer e importar cadenas de texto de los archivos de interfaz. De esta manera, el proceso consiste en sólo tres pasos realizados desde la misma consola: exportar del archivo del idioma original, traducir al nuevo idioma e importar al archivo de interfaz nuevo.

Para ello se hace uso de la herramienta de consola *ibtool* (Interface Builder Tool, Herramienta Constructora de Interfaces). Mediante el siguiente comando, se exportan todas las cadenas de texto del archivo de interfaz *ejemplo.xib*:

```
>> ibtool --export-strings-file  
/ruta_donde_se_dejara_el_archivo_con_las_cadenas ejemplo.xib
```

Esta orden genera un archivo *.strings* que contiene en cada línea la descripción del objeto al que hace referencia ese texto, el signo 'igual' y el valor de la cadena en el idioma original. Sólo se deberá abrir dicho archivo con un editor y cambiar las cadenas del idioma original al deseado. Entonces, mediante el siguiente comando se importan las traducciones al correspondiente archivo *.xib*:



```
>> ibtool --import-strings-file
/ruta_donde_se_encuentra_el_archivo_con_las_cadenas
--write
/ruta_donde_se_encuentra_el_archivo_del_nuevo_idioma.xib
/ruta_donde_se_encuentra_el_archivo_del_idioma_original.xib
```

Con este simple proceso, los archivos de interfaz de una aplicación podrán ser localizados fácilmente. Si lo único que hace falta en una aplicación es localizar cadenas de texto, se puede hacer uso de la macro **NSLocalizedString**, la cual se explica en la siguiente sección.

#### 4.6.2. NSLocalizedString

Cuando se pretenda localizar cadenas de texto, ya sean estáticas o dinámicas, se deberá hacer uso de la macro **NSLocalizedString** [5]. Esta función coge dos argumentos, la cadena que se desea localizar y el comentario que la describe, cuya utilidad se explicará posteriormente. Para ello se deberán seguir dos sencillos pasos: poner la macro en todas las cadenas que se quieran localizar y generar el archivo *Localizable.strings* mediante la herramienta de consola *genstrings*.

El uso de la macro es muy fácil, basta con llamarla:

```
1 NSString *string = NSLocalizedString(@"Test_string", @"Sample_
comment")
```

Una vez que se ha acoplado a todas las cadenas que se quieran localizar, se deberá hacer uso de la herramienta *genstrings*. Con el comando

```
>> genstrings ExampleViewController.m
```

se genera el archivo *Localizable.strings*. Siguiendo con el ejemplo anterior, dicho archivo, que se debe añadir de manera manual al proyecto, contendría lo siguiente:

```
/*Sample comment*/
"Test string"="Test string"
```

Como se puede observar, el campo del comentario de la macro sirve para poder identificar la cadena que se está traduciendo. Una vez añadido al proyecto, bastaría con localizarlo para obtener las diferentes versiones (figura 29)

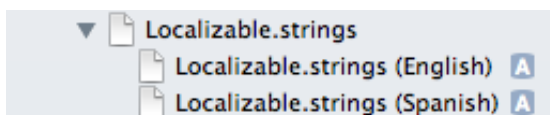


Figura 29: Resultado de localizar *Localizable.strings*

Sólo habría que seleccionar la versión, en este caso española, y traducir el campo de la derecha por la frase que deba figurar cuando el idioma del dispositivo sea el español. Cada vez que *iOS* encuentra la macro **NSLocalizedString**, busca en el fichero correspondiente la traducción. El contenido del fichero *Localizable.strings* siguiendo con el ejemplo anterior sería el siguiente:

```
/*Sample comment*/  
"Test string"="Cadena de prueba"
```

Por último, destacar que la aplicación ETSINF-UPV se encuentra totalmente localizada, disponiendo de todos los recursos (excepto los archivos PDF's) en inglés y español. Como ejemplo, en la figura 30 se puede ver la pestaña 'Home' de un dispositivo que tuviese seleccionado como idioma el inglés.

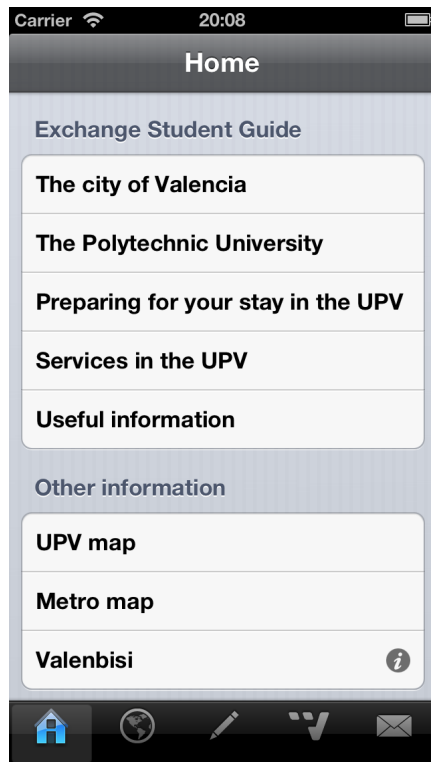


Figura 30: 'Home' de la aplicación en inglés

#### 4.6.3. NSUserDefaults

Muchas aplicaciones tienen distintas preferencias que los usuarios o usuarias pueden seleccionar. La clase **NSUserDefaults** se usa para guardar y cargar esas preferencias y que sean consistentes en cada inicio de la aplicación. En el caso de ETSINF-UPV, se ha implementado en tres lugares:

1. Para guardar el tipo de mapa seleccionado en la pestaña 'Mapa'.
2. Para guardar el mapa en la vista de 'Valencia' o de 'ETSINF' en la pestaña 'Mapa'.
3. Para guardar la información y la fecha del horario de la oficina de relaciones internacionales recibida del servidor en la pestaña 'Contacto'.

En los dos primeros puntos, por ejemplo, sólo se necesita almacenar el valor del segmento seleccionado del elemento **UISegmentedControl** de la figura 13. Para ello, deberemos acceder a la instancia *standardUserDefaults*. Esta se utiliza como si fuese un diccionario, con parejas de *clave-valor*. Una

vez se defina una constante global que será la *clave*, se podrá utilizar la clase `NSUserDefaults` para almacenar y cargar la información deseada:

```
1 NSString * const etsinfUpvMapTypePrefKey = @"
    etsinfUpvMapTypePrefKey";
2 NSString * const etsinfUpvMapRegionPrefKey = @"
    etsinfUpvMapRegionPrefKey";
3
4 ...
5
6 - (void) viewDidLoad
7 {
8     [super viewDidLoad];
9
10    ...
11
12    NSInteger mapTypeValue = [[NSUserDefaults
        standardUserDefaults] integerValueForKey:
        etsinfUpvMapTypePrefKey];
13    [mapTypeControl setSelectedSegmentIndex:mapTypeValue];
14    [self changeMapType:mapTypeControl];
15
16    NSInteger mapRegionValue = [[NSUserDefaults
        standardUserDefaults] integerValueForKey:
        etsinfUpvMapRegionPrefKey];
17    [mapRegionControl setSelectedSegmentIndex:mapRegionValue];
18    [self changeMapRegion:mapRegionControl];
19 }
20
21 ...
22
23 - (IBAction) changeMapType:(id) sender
24 {
25     [[NSUserDefaults standardUserDefaults] setInteger:[sender
        selectedSegmentIndex] forKey:etsinfUpvMapTypePrefKey];
26
27     ...
28 }
29
30 - (IBAction) changeMapRegion:(id) sender
31 {
32     [[NSUserDefaults standardUserDefaults] setInteger:[sender
        selectedSegmentIndex] forKey:etsinfUpvMapRegionPrefKey
        ];
33
34     ...
35 }
```

Para el punto 3, bastaría con hacer lo mismo almacenando un objeto (la información recibida del servidor y la fecha) en vez de un entero. En este caso, el método sería - *(void)setObject:(id)value forKey:(NSString \*)defaultName*.

#### 4.6.4. Ocultar y mostrar barra de navegación

En todas las vistas de ETSINF-UPV donde se pueden encontrar tablas, se ha añadido la posibilidad de ocultar la barra de navegación (figura 5). Se ha visto una mejora importante ya que en las vistas donde hay texto resulta mucho más cómodo leer sin tener que lidiar con la barra de navegación, que carece de utilidad en ese momento. Para ello, basta con pulsar sobre cualquier lugar de la vista y la aplicación ocultará la barra de navegación. Si se vuelve a pulsar en cualquier punto, la barra de vuelve a mostrar, permitiendo al usuario volver atrás y seguir navegando por las diferentes vistas de la aplicación.

La forma de hacer esto es añadir a los controladores de las vistas el siguiente código:

```
1 - (void) viewDidLoad
2 {
3     [super viewDidLoad];
4     UITapGestureRecognizer *tapGesture = [[
5         UITapGestureRecognizer alloc] initWithTarget:self
6         action:@selector(changeNavigationBar)];
7     [self.view addGestureRecognizer:tapGesture];
8 }
9 - (void) changeNavigationBar
10 {
11     if ([[self navigationController] navigationBar]
12         isHidden) {
13         [[self navigationController] navigationBar]
14         setHidden:NO;
15     } else {
16         [[self navigationController] navigationBar]
17         setHidden:YES;
18     }
19 }
```

Cuando se llame al método *viewDidLoad*., este crea una instancia de la clase **UITapGestureRecognizer**, la cual se encarga de lidiar con los eventos táctiles que se produzcan. Entonces le añade como *target* (objetivo) la misma clase, para que **UITapGestureRecognizer** le envíe los distintos eventos

relacionados con ella y le especifica como *action* (acción) el método *changeNavigationBar*;, definido por el desarrollador para implementar la lógica comentada antes para ocultar la barra de navegación.

#### 4.6.5. Botones, iconos y diseño corporativo

La aplicación se ha adaptado al máximo posible para que refleje la imagen corporativa de la ETS. de Ingeniería Informática. El icono principal, que es el que se ve en la pantalla 'Home' del dispositivo, consiste en una simplificación del logotipo original de la ETSINF (figura 31).



Figura 31: Icono principal de ETSINF-UPV

Desde los botones de la barra de navegación para navegar entre las vistas (figura 32) hasta el icono correspondiente a la pestaña ETSINF (figura 33) se han obtenido directamente de la imagen original. El color utilizado en tanto en los botones como en las tablas corresponde al original del logotipo de la ETSINF:  $RGB(212,0,102)$ .



Figura 32: Ejemplo de botón para volver a la vista anterior



Figura 33: Icono de la pestaña 'ETSINF'

El resto de iconos se han obtenido de diversas fuentes de iconos [9] [10] [11] [12] con licencias de libre uso.

## 5. Pruebas

Durante el desarrollo, se han ido probando y corrigiendo los avances obtenidos principalmente en dos dispositivos: el *iPhone 4S* comentado al principio y el simulador 'iPhone Simulator' versión 6.0, figura 34, proporcionado por Apple junto con el *Xcode*.

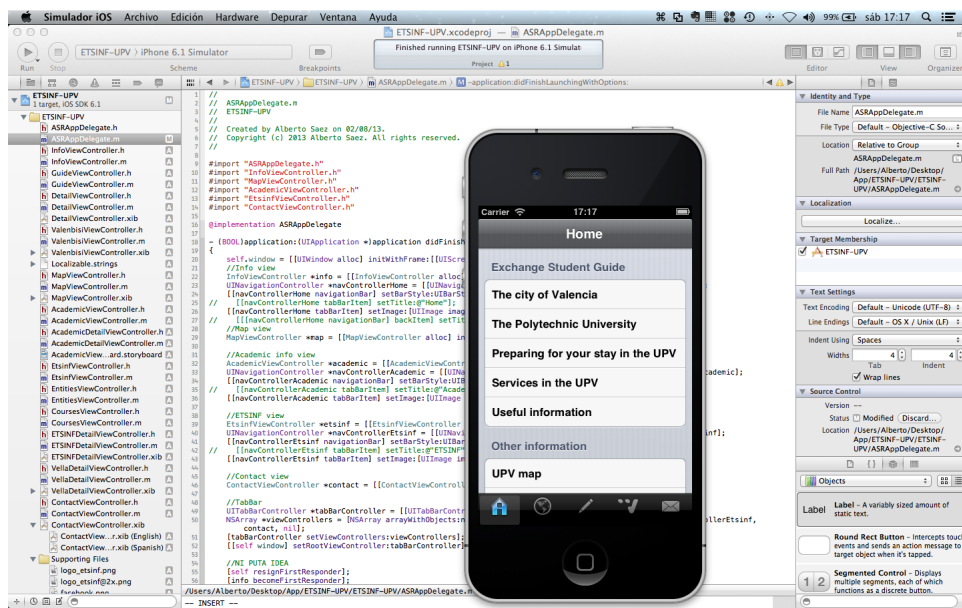


Figura 34: 'iPhone Simulator' versión 6.0

Las ventajas de utilizar el simulador durante el periodo de desarrollo son varias:

1. Mayor velocidad de carga de la aplicación que en el dispositivo físico
2. Capacidad de probar en dispositivos que no se tienen físicamente (por ejemplo, el *iPhone 5*)
3. Mayor facilidad para obtener capturas de pantalla (todas las capturas de pantalla que aparecen en este proyecto han sido tomadas con el simulador)
4. Aprovechar distintas opciones que posee el simulador para detectar varios tipos de fallos (imágenes fuera de pantalla o animaciones mal hechas entre otras)

Debido a que un simulador continúa siendo un simulador, en ocasiones se ejecutaba la aplicación en el dispositivo físico disponible (*iPhone 4S*). A veces determinadas vistas o imágenes se mostraban un poco diferente, descuadradas o movidas, por lo que se ha visto muy necesario probarlo en un dispositivo físico real.

Por este motivo, y debido a que la última versión del iPhone, el *iPhone 5*, dispone de una pantalla mas grande (tiene una pantalla de 4" mientras que las versiones anteriores cuentan con una pantalla de 3,5"), también se ha pedido prestado un *iPhone 5* real para poder corregir las posibles deficiencias en los dispositivos más actuales.

Al finalizar el desarrollo, se ejecutó una exhaustiva prueba de la aplicación en ambos dispositivos pero, al haberlos utilizado durante el desarrollo junto con el simulador, ambos ejecutaron la aplicación sin problemas, sin elementos fuera de lugar y sin retardos de ningún tipo.

## 5.1. 'Static Analyzer'

Un interesante instrumento proporcionado por Apple e integrado en *Xcode* es el '*Static Analyzer*' (Analizador estático). Este se encarga de analizar de manera estática e individual cada función presente en el código e intenta averiguar lo que pasaría si la ejecutáramos. Por ejemplo, si el código tuviera el siguiente método:

```
1 //Supongamos definidas 'x' e 'y'.
2
3 - (int)numberOfRows
4 {
5     int count;
6
7     if (x && y)
8         count = x + y;
9     return count;
10 }
```

el analizador estático nos daría un *error lógico*, ya que si la condición del *if* no se cumple, se devolvería un valor nulo en la variable *count*, ya que no está inicializada. Con este error, podríamos seguir compilando el programa, pero el analizador está avisando que en ese método es posible que la aplicación se cierre inesperadamente. Una solución simple para este ejemplo sería inicializar la variable *count*.



## 5.2. 'Instruments'

Otro de los instrumentos que proporciona Apple para los desarrolladores es la aplicación *Instruments* (Instrumentos). Esta consiste en conjunto de *plugins* (figura 35) diseñados para detectar problemas en tiempo de ejecución.

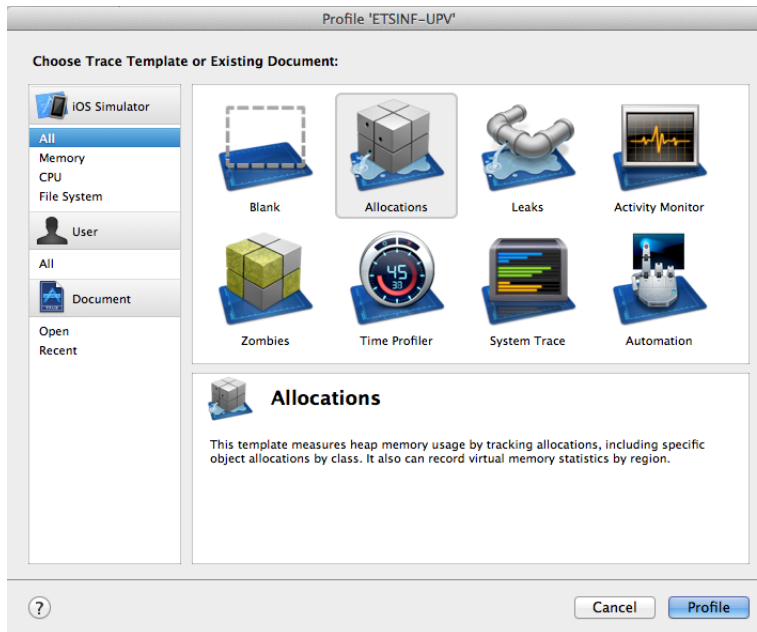


Figura 35: *Plugins* de *Instruments*

En el mundo *iOS*, al procedimiento de utilizar los *plugins* de *Instruments* para analizar una aplicación se le conoce como hacer un '*profiling*' (perfil) de la aplicación. En este caso, se ha decidido perfilar la aplicación con los tres *plugins* explicados en el libro que ha servido al autor para sumergirse en el mundo de la programación en *iOS* [1]:

1. **Allocations Instrument** (*Instrumento de asignaciones*): Este *plugin* se usa para obtener información sobre todos los objetos creados por la aplicación, así como la memoria que ocupan.
2. **Time Profiler Instrument** (*Instrumento perfilador de tiempos*): Este instrumento proporciona información acerca de los ciclos de CPU que gasta la aplicación. Es capaz de obtener el porcentaje de tiempo que usa cada método de la misma; de esta manera el desarrollador puede observar si un método que se supone debería consumir poco tiempo, gasta más del que necesita y de esa forma poder corregir el posible error.

3. **Leaks Instrument** (*Instrumento de fugas*): Por último, este *plugin* se ocupa de la detección de trozos de memoria en los que se encuentran objetos que se han quedado huérfanos, es decir, que ya no tienen padre, nadie apunta a ellos y por tanto nadie los va a poder liberar y desocupar ese trozo de memoria.

En ninguno de los tres *plugins* empleados se encontraron evidencias de trozos de código, métodos o procedimientos que estuviesen afectando de manera inesperada o dañina al sistema, arrojando todos ellos el resultado esperado de acuerdo con la lógica de la aplicación.

## 6. Conclusiones

Se ha desarrollado desde cero una aplicación móvil para la plataforma *iOS* que servirá como ayuda para la integración de los nuevos estudiantes de intercambio que reciba la ETS. de Ingeniería Informática de la Universitat Politècnica de València. Para ello, se han utilizado todas las herramientas proporcionadas por Apple para el desarrollo de aplicaciones, el *SDK*, los frameworks y las distintas *APIS*'s de las que dispone la plataforma.

Una vez finalizado el proyecto, se puede decir que los objetivos marcados en la sección 1.2 se han cumplido en su totalidad. Durante el desarrollo se han ido incorporando una serie de mejoras que complementan lo inicialmente propuesto. Ha sido en la etapa de desarrollo donde, por motivos de usabilidad, velocidad o simplemente por mejorar lo existente, se ha visto interesante el incorporar una serie de mejoras, la mayoría de fácil implementación en relación con la utilidad que tendrá posteriormente para el usuario.

Uno de los objetivos más importantes y a raíz del cual surgió la idea de desarrollar la presente aplicación, el de adaptar la 'Guía de Intercambio' para que sea fácilmente consumible en un dispositivo móvil, se considera cumplido. El sistema empleado de dividir las secciones de dicha guía en tablas, proporciona la flexibilidad y velocidad que se buscaban en un principio.

Se debe comentar que el objetivo de 'Disponer de información académica específica de la ETISNF' no se ha podido desarrollar en profundidad, ya que el apartado 'Exámenes' con el que cuenta la aplicación en la pestaña de 'Información Académica' (figura 16) se encuentra vacío al no estar disponibles a fecha de realización del proyecto los exámenes en la página web de la ETSINF.

Uno de los mayores problemas o desventajas que se han visto ha sido el hecho de que los distintos archivos PDF's disponibles en la aplicación no tienen URL's fijas, por tanto ha sido necesario descargarlos de la página web de la escuela e incorporarlos en la aplicación. El problema no es tanto de espacio (ocupan bastante poco) sino que cada año se deberán actualizar esos archivos directamente desde el código de la aplicación. Por contra, los archivos PDF's de la sección 'Bares' de la pestaña 'ETSINF', aunque se actualizan cada día, conservan siempre el mismo nombre, lo que hace posible que se carguen actualizados directamente desde internet.

Otro de los problemas observados es la falta de un diseñador gráfico, o alguien con experiencia en interfaces y usabilidad. Desde el adaptar las distintas imágenes o iconos hasta la distribución de los contenidos de la interfaz, se han hecho siguiendo la opinión y gusto del autor, sin ayuda de un profesional del sector.

Destacar también que el presente documento desarrollado en  $\text{\LaTeX}$  ha servido para cumplir uno de los objetivos personales del autor: aprender y utilizar dicho lenguaje para el desarrollo de un documento profesional.

También se debe hacer una mención especial a las páginas web de StackOverflow [7] y StackExchange [8]. Ambas han servido para resolver multitud de problemas técnicos, tanto de programación en *iOS* como de  $\text{\LaTeX}$ .

A todos los efectos, el autor considera el trabajo como una experiencia muy fructífera, de la que ha aprendido no sólo a desarrollar en *iOS*, sino a gestionar un proyecto por si mismo, con todo lo que ello implica, a nivel de recursos y organización.

## 6.1. Trabajo futuro

Uno de los primeros pasos a seguir, y dado las fechas en las que se ha acabado el proyecto, es el de enviar la aplicación para su publicación en el *AppStore*. De esta manera, los estudiantes de intercambio del actual año académico podrán sacar provecho de ella durante el curso.

Una vez la aplicación se encuentre aprobada y publicada, se dispone de una serie de ampliaciones o mejoras que han ido surgiendo durante el desarrollo, y que por motivos de tiempo o dificultad, no han podido ser implementadas todavía:

1. Completar la sección de 'Información Académica' con los exámenes
2. Añadir fotos a las secciones de la 'Guía de Intercambio'
3. Incorporar más bares en la pestaña 'ETSINF'
4. Empotrar dentro de la pestaña 'Mapas' (figura 11) el mapa en 3D de la UPV desarrollado como proyecto final de carrera de unos antiguos alumnos, y que sería de gran ayuda para que los estudiantes se orientasen dentro de la UPV
5. Añadir una sección que mediante *RSS* muestre las noticias más recientes de la ETSINF
6. Adaptar la aplicación para que sea consumible en los dispositivos en modo horizontal
7. Adaptar la aplicación para los dispositivos *iPad* y *iPad Mini*, aprovechando el espacio que ofrecen sus grandes pantallas

## Referencias

### iOS

- [1] 'iOS Programming: The Big Nerd Ranch Guide' 3ed
- [2] Apple Organizer Documentation provided with *Xcode*'.
- [3] 'iOS'. Wikipedia. Último acceso: 14/08/2013. Url: <http://en.wikipedia.org/wiki/iOS>
- [4] 'About the iOS technologies'. iOS Developer Center. Último acceso: 14/08/2013. Url: <https://developer.apple.com/library/ios/documentation/miscellaneous/conceptual/iphoneostechoverview/Introduction/Introduction.html>
- [5] 'Localizing String Resources'. iOS Developer Center. Último acceso: 19/08/2013. Url: <https://developer.apple.com/library/ios/DOCUMENTATION/MacOSX/Conceptual/BPInternational/Articles/StringsFiles.html>
- [6] 'The iOS environment'. iOS Developer Center. Último acceso: 16/08/2013. Url: <https://developer.apple.com/library/ios/documentation/iphone/conceptual/iphonesprogrammingguide/TheiOSEnvironment/TheiOSEnvironment.html>
- [7] 'StackOverflow'. Aclaración de muchas dudas y problemas. Url: <http://www.stackoverflow.com>
- [8] 'StackExchange'. Aclaración de muchas dudas y problemas. Url: <http://www.stackexchange.com>

### Recursos

- [9] 'Glyph icons'. Iconos de licencia de libre uso. Url: <http://glyphicons.com/>
- [10] 'TWG'. Iconos de licencia de libre uso. Url: <http://blog.twg.ca/2010/11/retina-display-icon-set/>
- [11] 'Iconshock'. Iconos de licencia de libre uso. Url: <http://www.iconshock.com/>
- [12] 'Obox social icon pack'. Iconos de licencia de libre uso. Url: [http://www.obox-design.com/view\\_item.cfm/title/free-social-icon-pack](http://www.obox-design.com/view_item.cfm/title/free-social-icon-pack)
- [13] 'Corporate visual identity'. Logotipo UPV. Url: <http://www.upv.es/perfiles/pas-pdi/identidad-corporativa-upv-en.html>
- [14] 'Mapa de estaciones Metrovalencia'. Último acceso: 7/08/2013. Url: <http://www.metrovalencia.es/>
- [15] 'Mapa de estaciones Valenbisi'. Último acceso: 7/08/2013. Url: [http://www.valenbisi.es/content/download/28017/149457/version/1/file/Borna\\_Valenbisi\\_2012.pdf](http://www.valenbisi.es/content/download/28017/149457/version/1/file/Borna_Valenbisi_2012.pdf)

- [16] 'Recursos ETSINF'. Información académica. Último acceso: 15/08/2013. Url: <http://www.etsinf.upv.es>

## Otros

- [17] 'XVim'. GitHub. Último acceso: 02/07/2013 Url: <https://github.com/JugglerShu/XVim>
- [18] 'UPV'. Appstore. Url: <https://itunes.apple.com/es/app/upv-universitat-politecnica/id561951828?mt=8>
- [19] 'Blog de la Oficina de Relaciones Internacionales de la ETSINF'. Url: <http://www.intacadetsinf.blogs.upv.es>
- [20]  $\text{\LaTeX}$ . Aclaración de dudas y problemas. Url: <http://en.wikibooks.org/wiki/LaTeX/>
- [21] 'StackExchange - TeX'. Aclaración de dudas y problemas sobre  $\text{\LaTeX}$ . Url: <http://www.tex.stackexchange.com>

## Glosario

**ETISNF** - Escuela Técnica Superior de Ingeniería Informática.

**UPV** - Universitat Politècnica de València.

**Appstore** - Tienda de aplicaciones de Apple donde los desarrolladores pueden distribuir sus aplicaciones para la plataforma *iOS*.

**ERASMUS** - Estudiantes que obtienen una beca para estudiar en el extranjero dentro de la Unión Europea.

**PROMOE** - Estudiantes que obtienen una beca para estudiar en el extranjero en cualquier lugar del mundo fuera de la Unión Europea.

**SICUE** - Estudiantes que obtienen una beca para estudiar en otras universidades de España.

**IDE** - *Integrated Development Environment* (Entorno de Desarrollo Integrado) es una aplicación informática formada por distintas herramientas que facilita el desarrollo de software.

**Mac OS X** - Sistema operativo para PC's de Apple.

**Xcode** - IDE proporcionado por Apple para desarrollar principalmente aplicaciones para la plataforma *iOS* o *Mac OS X*.

**XVim** - *Plugin* de *Xcode* para emular el entorno del editor *Vim*.

**SDK** - *Software Development Kit* (Kit de desarrollo de Software) es un conjunto de herramientas que permiten crear aplicaciones para cierto sistema.

**Framework** - Conjunto de tecnologías concretas que sirven de base para el desarrollo de software en un sistema.

**API** - *Application Programming Interface* (Interfaz de Programación de Aplicaciones) es un conjunto de métodos, funciones, etc. que ofrecen generalmente las bibliotecas para ser usadas.

**HTML** - *HyperText Markup Language* (Lenguaje de Marcado Hipertextual) es el lenguaje estándar para el desarrollo de páginas web.

**CSS** - *Cascading Style Sheets* (Hojas de Estilo en Cascada) es un lenguaje usado en HTML para describir hojas de estilo.

**Safari** - Navegador web de Apple, presente tanto en *iOS* como en *Mac OS X*.

**CPU** - *Central Process Unit* (Unidad Central de Procesamiento) es el componente principal de un sistema informático que se encarga del procesamiento de las instrucciones.

**L<sup>A</sup>T<sub>E</sub>X** - Sistema de composición de textos para el lenguaje de composición tipográfica T<sub>E</sub>X.

**RSS** - *Really Simple Syndication* (Distribución Realmente Simple) es un formato *XML* para compartir contenido en internet.

**URL** - *Uniform Resource Locator* (Localizador de Recursos Uniforme) secuencia de caracteres usados para localizar recursos (generalmente servidores web) en internet.