



# Desarrollo de aplicaciones móviles con herramientas del Campus UOC

Alumno:

**Pablo León Manzano**

Grado de Ingeniería Informática

Tutores:

**David Gañan Jiménez**

**Xavier Aracil Díaz**

4 de Enero de 2014



Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](https://creativecommons.org/licenses/by/3.0/es/)

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	Desarrollo de aplicaciones móviles con herramientas del campus UOC
<b>Nombre del autor:</b>	Pablo León Manzano
<b>Nombre del consultor:</b>	Xavier Aracil Díaz David Gañan Jimenez
<b>Fecha de entrega (mm/aaaa):</b>	01/2014
<b>Área del Trabajo Final:</b>	Ingeniería del Software Herramientas UOC
<b>Titulación:</b>	<i>Grado en Ingeniería Informática</i>
<b>Resumen del Trabajo (máximo 250 palabras):</b>	
<p>El objetivo de este trabajo es implementar una herramienta segura, intuitiva y útil de dispositivo electrónico móvil para que un alumno interactúe con el entorno educativo virtual creado por el Campus de Trabajo UOC.</p> <p>La herramienta ayuda al alumno a trabajar con la UOC sin necesidad de conectar directamente desde un portátil y desde cualquier lugar, siempre y cuando, se tenga conexión a Internet.</p>	
<b>Abstract (in English, 250 words or less):</b>	
<p>The goal of this project is to develop a secure, intuitive and useful tool for a mobile electronic device that will help any student to interact with the virtual learning environment created by the UOC Campus.</p> <p>The tool helps students to work with the UOC without connecting directly using a laptop, anywhere, the only requirement is the device and Internet connection.</p>	
<b>Palabras clave (entre 4 y 8):</b>	
móvil "campus UOC" "Windows Phone"	

## Índice

1	Introducción.....	1
1.1	Contexto y justificación del Trabajo .....	1
1.2	Objetivos del Trabajo.....	2
1.3	Enfoque y método seguido.....	2
1.4	Planificación del Trabajo.....	2
1.5	Breve resumen de productos obtenidos.....	3
1.6	Breve descripción de los otros capítulos de la memoria .....	3
2	Requisitos y análisis del sistema.....	4
2.1	Clasificación de requisitos.....	4
2.2	Objetivos funcionales de la aplicación.....	4
2.3	Desglose de requisitos.....	5
2.4	Cumplimiento de requisitos .....	8
3	Diseño.....	10
3.1	Diagrama de casos de uso.....	10
3.2	Diagrama de componentes .....	11
3.3	Diagrama de clases.....	12
3.4	Capa de datos.....	14
3.4.1	Uso de entidades por la capa de interfaz.....	16
4	Arquitectura y aspectos técnicos .....	17
4.1	Núcleo.....	17
4.1.1	Autenticación .....	17
4.1.1.1	Petición de la Uri de autenticación .....	18
4.1.1.2	Autenticación desde la interfaz.....	18
4.1.1.3	La petición del “token” .....	19
4.1.2	Comunicación con la API.....	19
4.1.2.1	El servicio de información de la UOC (API) .....	19
4.1.2.2	Servicios ofrecidos por la API .....	19
4.1.2.3	La librería RestSharp.....	22
4.1.2.4	Formato de respuesta de las llamadas a la API .....	22
4.2	Integración y Base de datos.....	22
4.2.1	Representación de los datos. Entidades. ....	23
4.2.2	La librería Newtonsoft JSON Deserializer.....	25
4.2.3	Llamada a la API y traducción de JSON .....	25
4.3	Interfaz.....	27
4.3.1	Vistas.....	28
4.3.1.1	Login.....	28
4.3.1.2	Página de entrada .....	28
4.3.1.3	Principal.....	29
4.3.1.4	Buzón .....	29
4.3.1.5	Calendario.....	29
4.3.1.6	Asignaturas .....	29
4.3.1.7	Configuración.....	29
4.3.2	Multilinguaje .....	30
5	Herramientas de desarrollo .....	31
6	Conclusiones.....	31
7	Bibliografía y fuentes citadas.....	33

8	Anexos.....	34
8.1	Diagrama de clases. (Ampliado).....	34
8.2	Diagrama de entidades de la capa de datos.....	35
8.3	Manual de usuario de instalación.....	36
8.3.1	Instalación manual.....	36
8.3.2	Instalación a través de la tienda (puede que no esté disponible todavía).....	37
8.4	Manual de uso de la aplicación.....	38
8.4.1	Abrir aplicación.....	38
8.4.2	Acceder al sistema.....	38
8.4.3	Menú principal. Navegación.....	39
8.4.4	Correo.....	41
8.4.5	Asignaturas.....	42

## Lista de figuras

Ilustración 1: Diagrama de casos de uso	10
Ilustración 2: Diagrama de casos de uso ampliado	11
Ilustración 3: Diagrama de componentes	12
Ilustración 4: Diagrama de clases. (Véase ampliado en Anexo 8.1)	13
Ilustración 5: Flujo de información. Camino correcto	14
Ilustración 6: Tipo de dato Usuario devuelto por la API UOC	15
Ilustración 7: Representación UML del tipo de dato Usuario en nuestro sistema.	15
Ilustración 8: Uso de un tipo de dato por una vista. Representación UML	16
Ilustración 9: Diagrama básico de la arquitectura	17
Ilustración 10: Diagrama de secuencia de autenticación y servicio correctos con la API UOC	18
Ilustración 11: Código JSON de respuesta	22
Ilustración 12: Código (C#) de la clase Classroom	24
Ilustración 13: Código (C#) de la clase Classrooms	24
Ilustración 14: Ejemplo de la llamada a ExecuteAsync	25
Ilustración 15: Método ConstruyePetición	26
Ilustración 16: Método ExecuteAsync	27
Ilustración 17: Prototipo de la aplicación. Buzón.	28
Ilustración 18: Vista de autenticación	28
Ilustración 19: Vista del prototipo de la aplicación. Página principal.	29
Ilustración 20: Interfaz multilingüaje	30
Ilustración 21: Código de inicialización genérica del idioma	30

# 1 Introducció

Esta memoria recopila y resume el Trabajo Fin de Grado del Grado de Ingeniería Informática de Pablo León Manzano. El trabajo intenta aunar parte de lo aprendido a lo largo de la carrera. Comprendiendo facetas de ingeniería del software, desarrollo e implementación en una aplicación real, en este caso, para un dispositivo móvil.

Además este trabajo supondrá el aprender a desarrollar aplicaciones en un nuevo entorno cada vez más pujante en nuestra sociedad como son las aplicaciones para dispositivos móviles, utilizando uno de estos entornos. Para enfatizar el hecho de que se trata de una nueva aplicación no existente todavía, se ha escogido un entorno en el que todavía no existe una aplicación similar.

Aunque este trabajo está relacionado con el área “.Net” se encuentra enmarcado dentro del área “Herramientas del Campus UOC”, es decir, que aprovechamos dos áreas, por un lado una que recoge el conocimiento técnico de desarrollo de una aplicación y por otro el de ofrecer una nueva propuesta para la UOC para que el alumno pudiera comunicarse con el Campus Virtual desde un dispositivo que todavía no tiene una aplicación específica para este cometido.

Para esta última faceta diseñaremos e implementaremos los métodos que nos permitan acceder a los sistemas de información de la UOC y presentando esta información de una manera práctica y amigable para el usuario final.

El trabajo ha sido tutorado por los consultores de la UOC: David Gañan Jimenez y Xavier Aracil Díaz, que han sido de inestimable ayuda para el desarrollo de este trabajo. Este documento es por tanto, la entrega final del documento que se ha ido desarrollando a lo largo del cuatrimestre corrigiéndolo y ampliándolo continuamente durante las diferentes entregas. Es por esto que el trabajo recoge partes de análisis y elicitación de requisitos, diseño, implementación y manuales. Con un cierto sentido cronológico al igual que se ha ido desarrollando durante el curso.

## 1.1 Contexto y justificación del Trabajo

Actualmente no existe una herramienta por parte de la Universitat Oberta de Catalunya (UOC), a excepción del acceso mediante navegador móvil, para los dispositivos con el sistema operativo Windows Phone [1] creado por la compañía Microsoft, que cubren una cuota de mercado del 9% [2].

Por tanto, con este trabajo se pretende ayudar a los alumnos a tener un método de acceso más para comunicarse e interactuar con el campus virtual de la UOC de una manera agradable y cómoda que desde el acceso Web, que puede suponer más costoso por el mayor volumen de datos que necesitaría y más cómodo por ser una aplicación especialmente diseñada para las interfaces de estos dispositivos electrónicos.

## 1.2 Objetivos del Trabajo

Aunque los objetivos principales de este trabajo son puramente académicos, también podríamos decir que los objetivos del TFG también serían los del desarrollo de una aplicación móvil para dispositivos electrónicos que soporten el sistema operativo Windows Phone 7 u 8 creado por la compañía Microsoft, utilizando como canal de comunicación con esta plataforma, las APIs [3] proporcionadas por la UOC [4].

La aplicación sería similar funcionalmente a la ya existente para plataforma Android [5] llamada "Campus". Pudiendo el alumno conocer los correos del buzón, la información relativa a las asignaturas y la agenda. Aunque la aplicación evidentemente no será la utilizada oficialmente por la UOC para este cometido, sí que ofrece una propuesta válida, y que puede ser libremente utilizada, para este cometido.

Las funciones para interactuar con la plataforma (enviar correos, modificar la agenda, o escribir en los foros), se implementarán en medida de lo posible, de acuerdo con el corto tiempo estimado para el desarrollo del proyecto y la disponibilidad de ellos a través de las APIs. Ya que lo que pretende el proyecto es demostrar los conocimientos adquiridos del alumno.

La aplicación soportará varios idiomas (castellano, catalán e inglés), de este modo recogemos las tres lenguas que pueden interesar más al alumno de la UOC. Ya que la UOC recoge estudios en catalán, castellano e inglés.

## 1.3 Enfoque y método seguido

La idea es desarrollar un producto nuevo similar a los ya existentes en otros soportes como Android o iOS [6], creados por Google y Apple respectivamente. Aunque la cuota de mercado de Windows Phone está muy alejada de Android e IOS, esta aplicación ayudaría a no discriminar a un soporte con miles de unidades vendidas.

Además, aunque la aplicación pueda no ser la usada finalmente, sí que exploraremos y explotaremos las herramientas necesarias para el desarrollo de la aplicación. Ayudando a otros desarrolladores a crear una herramienta similar para este mismo entorno o entornos similares.

## 1.4 Planificación del Trabajo

En la siguiente tabla presentamos una tabla que desglosa los cuatro hitos principales del trabajo.

Nombre y fecha de entrega	Objetivos de la entrega
PAC1 - 30/09/2013	Requerimientos funcionales Borrador de diseño y arquitectura
PAC2 - 28/10/2013	Diseño completo y prototipo de la aplicación
PAC3 - 16/12/2013	Aplicación para revisión final
Entrega final 04/01/2014	Versión definitiva del código fuente, memoria y documentación.



Por tanto, estos hitos se organizan en el tiempo del siguiente modo:

Id.	Nombre de tarea	Comienzo	Fin	Duración	sep. 2013			oct. 2013				nov. 2013				dic. 2013				ene. 2014				feb. 2014				
					1/9	8/9	15/9	22/9	29/9	6/10	13/10	20/10	27/10	3/11	10/11	17/11	24/11	1/12	8/12	15/12	22/12	29/12	5/1	12/1	19/1	26/1	2/2	9/2
1	Planificación	02/09/2013	30/09/2013	21d	█																							
2	Diseño y prototipo	01/10/2013	28/10/2013	20d					█																			
3	Implementación	29/10/2013	16/12/2013	35d											█													
4	Entrega final	17/12/2013	03/01/2014	14d																█								

### 1.5 Breve resumen de productos obtenidos

El resultado final será una aplicación móvil que servirá a los alumnos de la UOC para estar al día en cualquier lugar sobre las asignaturas, eventos o correos del alumno.

Esta herramienta representa una alternativa mucho más útil y práctica para acceder que mediante la página web. Al resumir los contenidos principales y ofrecer un entorno personalizado para el dispositivo electrónico.

Además, se entregará este mismo documento que contiene una memoria del trabajo y manuales de la aplicación. Junto con el mismo se entregará un pequeño vídeo de presentación de la misma y el código fuente.

### 1.6 Breve descripción de los otros capítulos de la memoria

En la siguiente memoria realizaremos un recorrido por las diferentes fases de evaluación, desarrollo e implementación de la aplicación. Hemos dividido esta memoria en tres grandes capítulos principales, estas partes comprenden el análisis de requisitos, el diseño de la aplicación y el de la arquitectura, implementación y aspectos técnicos.

Aparte dispondremos de otros capítulos que comprenden diversas facetas de la aplicación y referencias para entender mejor el grueso del documento. Además de los anexos, como diagramas ampliados, manuales de uso o de instalación.

## 2 Requisitos y análisis del sistema

En este apartado realizaremos un análisis ampliado de los requisitos que necesita la aplicación móvil enmarcada en la propuesta de TFG titulado "Desarrollo de una aplicación móvil con herramientas del campus UOC".

### 2.1 Clasificación de requisitos

Para este análisis de requisitos se han clasificado los requisitos en dos escalas distintas. Por un lado, aquella que determina la necesidad del requisito que determinaría la completitud del mismo. Teniendo tres niveles: Alta, Media y Baja. Un requisito clasificado como "Alto" debería ser un requisito básico o necesario de la aplicación, y un requisito clasificado como "Bajo" sería uno que se implementaría en el caso de que fuera sencillo o se tuviera tiempo suficiente para llevarlo a cabo sin ser una prioridad de la aplicación.

Por otro lado, se han clasificado los requisitos dependiendo de si se trata de un requisito funcional o no funcional. Entendiendo que un requisito funcional sería aquel que define una funcionalidad (cometido o utilidad) de la misma, y aquellas no funcionales, siendo aquellas que son atributos de calidad de la misma.

También, cabe indicar que hemos clasificado los requisitos con unos requisitos primarios, que abarcarían un área de funcionalidades, y que de estos dependerían diferentes requisitos que podrían ser parte del conjunto del área "padre" o una funcionalidad extra del mismo.

### 2.2 Objetivos funcionales de la aplicación

En el siguiente listado, tenemos la recopilación inicial de funcionalidades, que ahora tendremos como objetivos de la aplicación, y de los que desglosaremos en requisitos más adelante.

- Visualización del calendario de actividades
- Comunicación a través del correo de la UOC
  - Listado de los mensajes en la bandeja de entrada del alumno.
    - Filtrado por mensaje leído o no leído.
    - Ordenación por fecha o autor
  - Lectura de los mensajes
  - Posibilidad de responder los mensajes
  - Posibilidad de redactar un nuevo mensaje
- Interacción con las aulas
  - Listado de las actividades del aula
  - Listado, lectura y respuesta de los mensajes pendientes en los foros y tableros de la asignatura
  - Acceso a los materiales de las aulas
  - Grupos de trabajo y listado de compañeros del aula
    - Interacción con las actividades del aula
    - Acceso a los materiales
  - Interacción con los compañeros en caso de actividades grupales

### 2.3 Desglose de requisitos

En este apartado desglosaremos la lista de funcionalidad vista en el apartado 2.2 como requisitos de la aplicación, además de añadir los requisitos no funcionales.

Código	1
Título	Autenticación en el aula
Descripción	Proporciona al alumno la posibilidad de identificarse con su usuario y contraseña en el aula.
Prioridad	Alta
Tipo	Funcional

Código	1.1
Título	Recordar credenciales
Descripción	Proporciona al alumno la posibilidad de que la aplicación recuerde su usuario y contraseña para no tener que identificarse cada vez que se quiera utilizar la aplicación.
Prioridad	Baja
Tipo	Funcional

Código	2
Título	Visualización del calendario de actividades
Descripción	Proporciona al alumno un listado con las actividades pendientes relacionadas con el alumno.
Prioridad	Alta
Tipo	Funcional

Código	3
Título	Comunicación a través del correo de la UOC
Descripción	Proporciona al alumno la posibilidad de conocer el contenido de los directorios de correo electrónico de la UOC.
Prioridad	Alta
Tipo	Funcional

Código	3.1
Título	Listado de los mensajes en la bandeja de entrada del alumno
Descripción	Muestra al alumno los mensajes contenidos en la bandeja de entrada del correo electrónico de la UOC.
Prioridad	Alta
Tipo	Funcional

Código	3.1.1
Título	Filtrado por mensaje leído o no leído
Descripción	Ofrece la posibilidad de filtrar los mensajes por aquellos que ya han sido visitados por el alumno y los que no.
Prioridad	Media
Tipo	Funcional

Código	3.1.2
Título	Ordenación por fecha o autor
Descripción	Ofrece la posibilidad al alumno de ordenar la bandeja de entrada de correo electrónico por fecha (ascendente o descendente) o autor (remitente) del correo.
Prioridad	Media
Tipo	Funcional

Código	3.2
Título	Lectura de los mensajes
Descripción	Ofrece la posibilidad a los alumnos de leer el contenido de los mensajes de la bandeja de entrada de correo electrónico.
Prioridad	Alta
Tipo	Funcional

Código	3.3
Título	Posibilidad de responder y redactar los mensajes
Descripción	Ofrece al alumno un diálogo donde poder contestar o escribir un nuevo mensaje de correo electrónico UOC a otro miembro de la UOC.
Prioridad	Media
Tipo	Funcional

Código	4
Título	Interacción con las aulas
Descripción	Ofrece la posibilidad al alumno de poder interactuar con las aulas de las diferentes asignaturas en las que está matriculado el alumno.
Prioridad	Alta
Tipo	Funcional

Código	4.1
Título	Listado de las actividades del aula
Descripción	Ofrece al alumno el listado de las próximas y pasadas actividades (PACs, entregas, etc) de una asignatura.
Prioridad	Alta
Tipo	Funcional

Código	4.2
Título	Listado, lectura y respuesta de los mensajes pendientes en los foros y tablones de la asignatura
Descripción	Ofrece al alumno la posibilidad de poder leer los mensajes de los foros. Así, de la misma manera que se hace en los requisitos de código 3 de correo electrónico, la posibilidad de filtrarlos, contestarlos o redactar nuevos.
Prioridad	Media
Tipo	Funcional

Código	4.3
Título	Acceso a los materiales de las aulas
Descripción	Ofrece al alumno la posibilidad de acceder a los recursos del aula. Como archivos y enlaces de la asignatura.
Prioridad	Baja
Tipo	Funcional

Código	4.4
Título	Grupos de trabajo y listado de compañeros del aula
Descripción	Ofrece a los alumnos la visualización de los grupos de trabajo y de los compañeros pertenecientes a estos grupos.
Prioridad	Baja
Tipo	Funcional

Código	4.5
Título	Interacción con las actividades del aula
Descripción	Posibilidad al alumno de interactuar con las actividades del aula.
Prioridad	Baja
Tipo	Funcional

Código	4.5.1
Título	Acceso a los materiales
Descripción	Posibilidad de acceder a los acceder a los archivos y enlaces de las actividades de la asignatura.
Prioridad	Baja
Tipo	Funcional

Código	4.5.2
Título	Interacción con los compañeros en caso de actividades grupales
Descripción	Posibilidad de interactuar con los alumnos pertenecientes al mismo grupo de una actividad determinada del aula.
Prioridad	Baja
Tipo	Funcional

Código	5
Título	Globalización
Descripción	Posibilidad de interactuar con la aplicación en diferentes idiomas. Castellano, catalán e inglés.
Prioridad	Media
Tipo	No funcional

Código	6
Título	Entorno
Descripción	La aplicación debe poder soportar el sistema operativo Windows Phone 7.1 o superior.
Prioridad	Alta
Tipo	No funcional

#### 2.4 Cumplimiento de requisitos

Como anexo al apartado anterior, una vez finalizado el proyecto se entrega la siguiente tabla de cumplimiento de requisitos. Esta tabla representa en qué grado hemos cumplido los requisitos de la aplicación. Ya que algunos de ellos no han podido ser implementados por diferentes motivos, la mayoría de ellos técnicos o por falta de tiempo.

Este apartado es indicativo de un tipo de tarea habitual tras la entrega de un producto o durante del desarrollo de la misma a un cliente para analizar qué objetivos de la aplicación han acabado siendo logrados.

Como se puede observar en la tabla han sido alcanzados el 100% de los requisitos de prioridad alta, cuatro de cinco de prioridad media, aunque no todos en su totalidad y cuatro de seis objetivos de prioridad baja.

De este modo se realiza un trabajo de análisis sobre la completitud del producto o aplicación final obtenida.

Código	Prioridad	Grado de Cumplimiento	Motivo
1	Alta	100%	
1.1	Baja	0%	Técnico. La autenticación era mediante acceso web a través del navegador, lo que impedía el poder usar los controles de la web para facilitar el recuperar usuario y contraseña.
2	Alta	100%	
3	Alta	100%	
3.1	Alta	100%	
3.1.1	Media	25%	Aunque se indican los mensajes no leídos por el alumno no se permite filtrar sobre ellos.
3.1.2	Media	0%	Tiempo. Se prefirió realizar otras tareas de implementación prioritarias.
3.2	Alta	100%	
3.3	Media	75%	Aunque se pueden redactar mensajes, no se pueden responder.
4	Alta	100%	
4.1	Alta	100%	
4.2	Media	66%	Aunque se pueden leer los tabloneros de la asignatura no se permite contestar sobre ellos.
4.3	Baja	100%	
4.4	Baja	50%	Aunque se pueden visualizar los alumnos no se pueden visualizar los grupos.
4.5	Baja	100%	
4.5.1	Baja	100%	
4.5.2	Baja	0%	No se puede visualizar los grupos ni actividades grupales.
5	Media	100%	
6	Alta	100%	

Al haber sido alcanzados todos los requisitos de prioridad alta, y la mayoría de los objetivos de prioridades media y baja podemos considerar de que se trata de una herramienta funcional válida y útil, y que ha cumplido en su gran mayoría los objetivos de la misma.

## 3 Diseño

### 3.1 Diagrama de casos de uso

El siguiente diagrama de casos de uso expresa la sencillez de la aplicación a la vista de los usuarios (alumnos). El usuario que se use puede tener dos roles principales, el alumno no autenticado y el autenticado. El autenticado tendrá la opción de poder acceder a las funcionalidades del sistema previo paso por la autenticación.

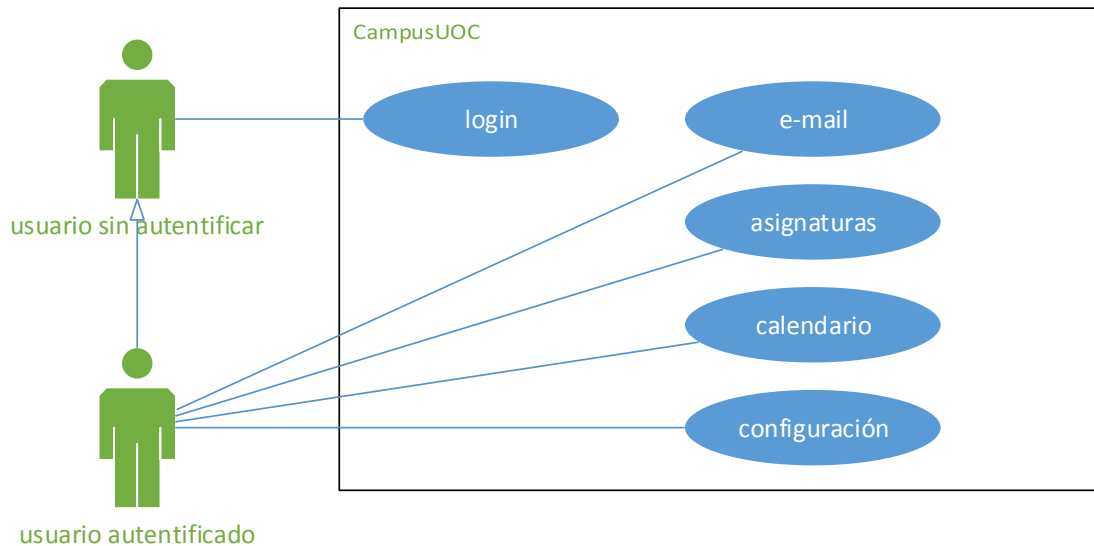


Ilustración 1: Diagrama de casos de uso

Por tanto, es importante destacar que un usuario no autenticado no debe de tener acceso a la aplicación. Además de por seguridad, porque se rompería con la lógica de la misma, al no conocer un usuario no se puede determinar cuáles son las asignaturas o recursos relativos al usuario.

La autenticación se realizará mediante OAuth, y las especificaciones técnicas se ven en el punto dedicado a la arquitectura y aspectos técnicos del sistema.



Si ampliamos el diagrama desglosando las funcionalidades descritas de un modo más detallado podríamos obtener el siguiente diagrama de casos de uso.

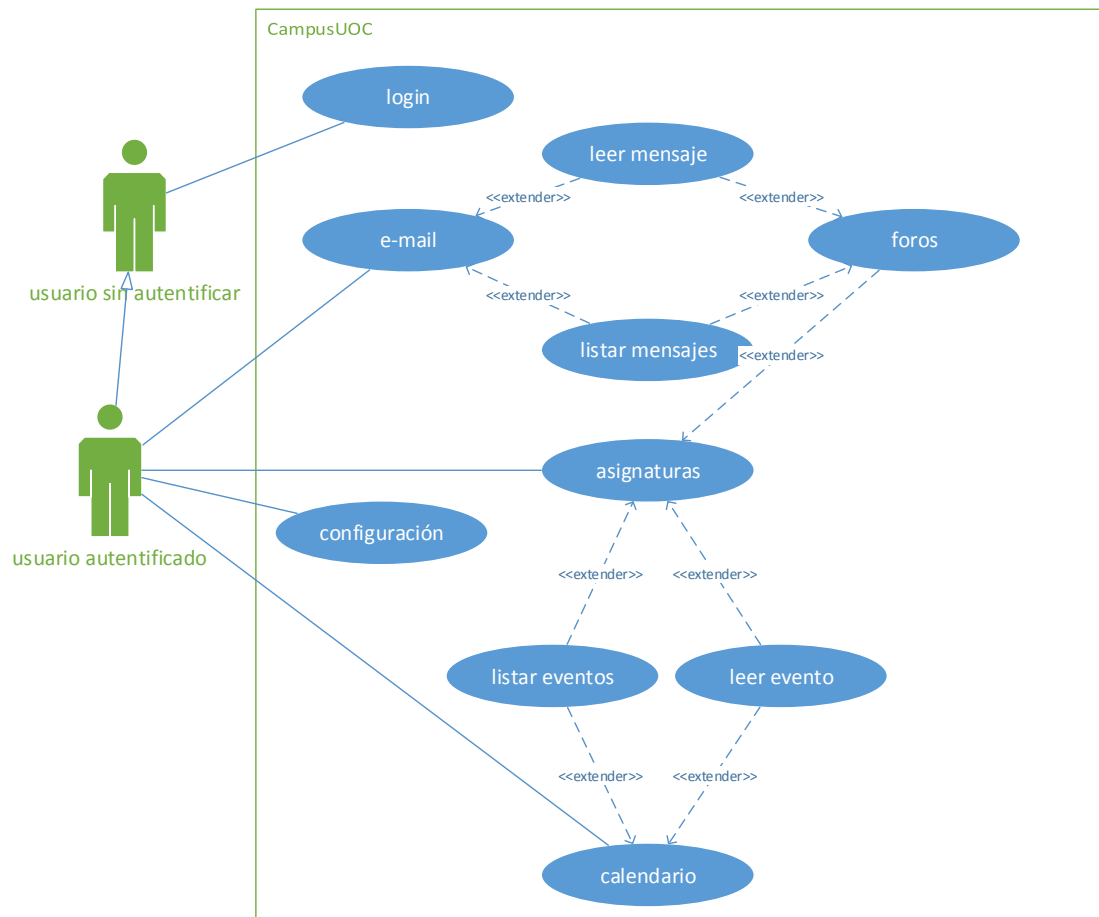


Ilustración 2: Diagrama de casos de uso ampliado

En él podemos ver las funcionalidades compartidas en las vistas de la lectura de mensajes o el listado de eventos que pueden llegar a ser funcionalidades compartidas por varios objetos del sistema.

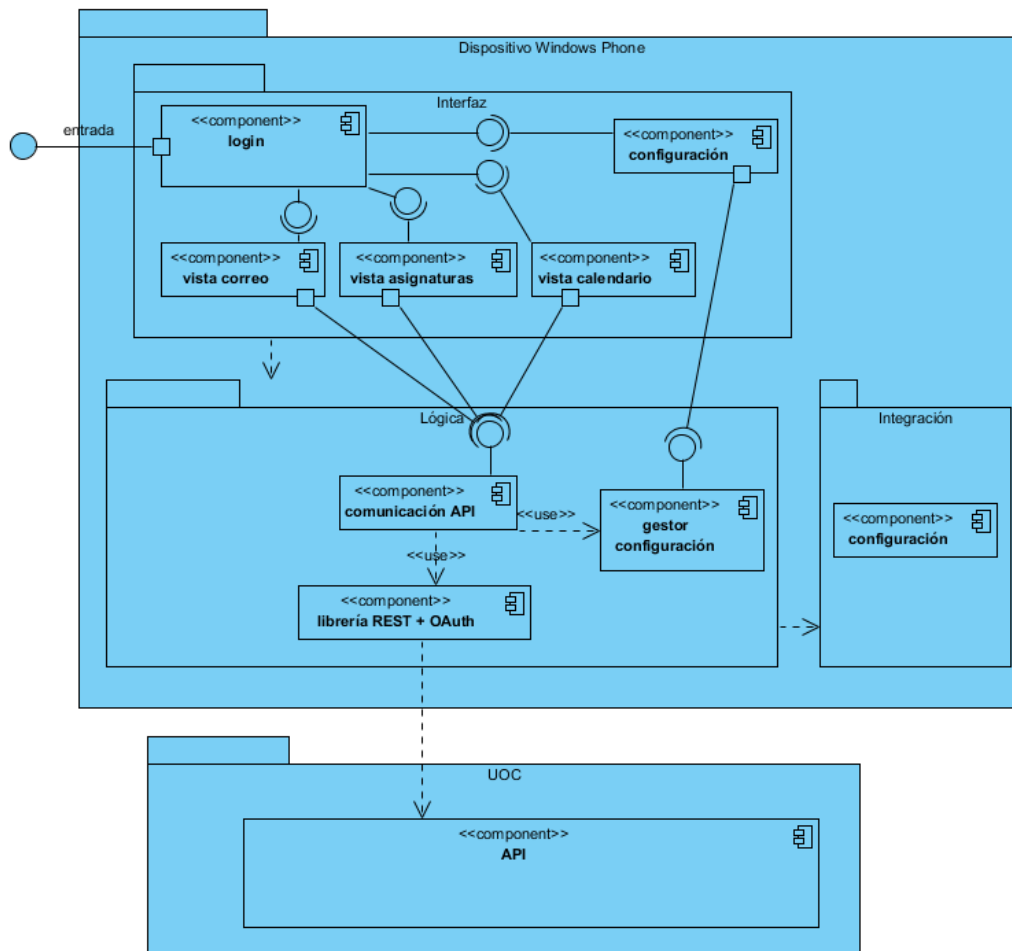
En esta vista para el diagrama de casos de usos, pasamos a relacionar de una manera más funcional y amplia los recursos y funcionalidades que nos podría ofrecer la aplicación.

### 3.2 Diagrama de componentes

Otra vista de nuestro programa es el diagrama de componentes, este nos ayudará a separar los diferentes módulos del programa a implementar. Ayudándonos en un futuro a organizar el proyecto a desarrollar.

El diagrama de componentes muestra cómo se conectan entre sí los componentes para formar componentes más grandes y representar de una sola vista la organización del sistema completo. De este modo ilustraremos la estructura básica de nuestra aplicación pudiendo organizar mejor nuestra aplicación móvil. Además también conseguimos representar componentes externos a nuestro

sistema como puede ser la API UOC, que aunque no será responsabilidad nuestra sí que será una parte esencial de nuestra aplicación puesto que es la API quién nos debe ofrecer casi toda la información con la que trabajaremos en el sistema de manera dinámica.



**Ilustración 3: Diagrama de componentes**

En el diagrama de componentes podemos apreciar la separación de las capas que integrarán nuestro proyecto. Las diferentes vistas usarán una interfaz común de comunicaciones para obtener y entregar la información necesaria a la API.

Se ha añadido un pequeño componente de integración que nos permitirá guardar aspectos de la configuración del programa. Pero la idea principal es que la aplicación no necesite salvaguardar demasiada información en el mismo sistema.

### 3.3 Diagrama de clases

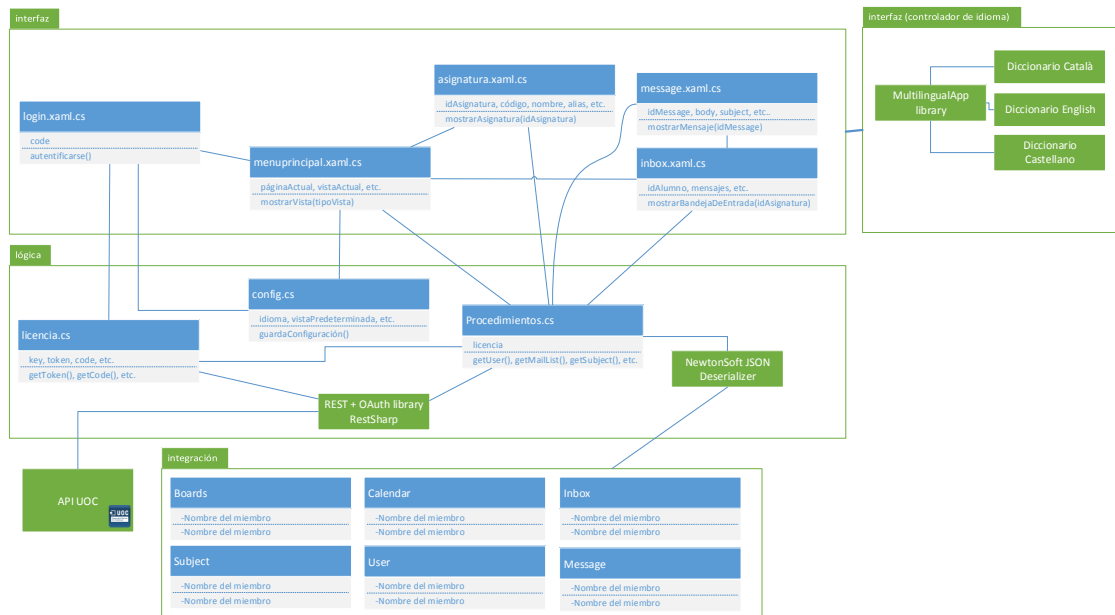
El diagrama de clases se podría resumir en el de la Ilustración 4, y que se encuentra ampliado en el anexo 8.1, que por simplificación hemos reducido en sus atributos y funcionalidades a un pequeño listado para cada clase. En él diferenciamos dos clases principales para la interfaz de usuario, éstas son las encargadas de la autenticación y del control de eventos y peticiones por parte del usuario, así como de la muestra de las diferentes vistas. Las vistas estarán diseñadas mediante WPF (XAML) [7], pero sólo necesitan una clase para gestionar el *login* y poder integrar el navegador, y otra para gestionar la página

pivotada que tendrá las diferentes vistas. Aparte, desde la página principal podremos movernos por otras vistas que comprenden las diferentes vistas dependiendo de aquello que queremos mostrar, estas vistas están dedicadas a cada uno de los objetos principales de nuestro entorno, por ejemplo, disponemos de la clase “asignatura.xaml.cs” que controla las vistas de la página “asignatura.xaml” que será donde mostremos las diferentes propiedades de una asignatura determinada.

Además, la interfaz usará una librería encargada de gestionar los diferentes idiomas lingüísticos que podrán configurarse en la aplicación final y se usará para mostrar una vista con diálogos en el idioma preferido por el usuario. Para simplificar y mejorar el entendimiento de la capa de interfaz esta librería se ha representado como una vista aparte, pero de la que dependerán todas las vistas para mostrar el idioma seleccionado por el usuario.

Mediante el control de eventos provocados por parte del sistema y del usuario estas clases realizarán llamadas a las clases encargadas de otorgar y recibir la información introducida por el usuario, como pueda ser mostrar una vista o realizar una petición del listado de correos electrónicos del usuario.

La parte encargada de la lógica del sistema dispondrá de tres clases principales, que serán las encargadas de construir las peticiones REST que la librería que explicaremos más adelante usará para comunicarse con la API de la UOC.



**Ilustración 4: Diagrama de clases. (Véase ampliado en Anexo 8.1)**

Al usar una única licencia por usuario, sólo necesitamos un objeto en memoria que podremos instanciar para realizar la autenticación y recogida de la licencia. Además deberemos de crear una clase encargada de ofrecer la funcionalidad para guardar en el dispositivo los pequeños datos de configuración de la aplicación.

La clase principal del núcleo es la clase “Procedimientos”, que es la encargada de interconectar las diferentes capas entre sí. Tras recibir una petición de una clase de la interfaz tras un evento disparado en un principio por un usuario, ésta es la encargada de formar la petición, pasarla a la librería REST, recoger la información y pasarla a la librería que transformará la codificación JSON en nuestras entidades representadas en nuestro programa como objetos instanciados de clases que tienen las mismas propiedades que la información proporcionada por la API de la UOC.

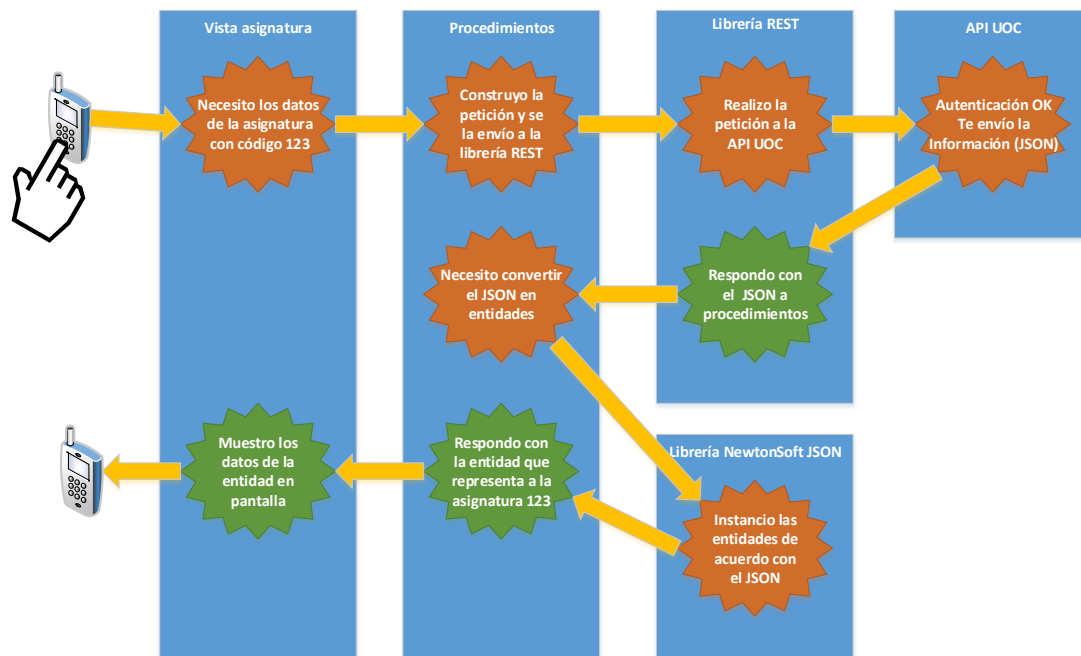


Ilustración 5: Flujo de información. Camino correcto

De esta manera, conseguimos que la información fluya desde la API hasta la interfaz. Para realizar dicha petición evidentemente el usuario debe haberse autenticado con anterioridad, ya que para realizar la petición a la API el usuario debe haberle otorgado derechos al usuario y una clave válida temporalmente que debe proporcionar en el momento de la petición el protocolo para la autenticación lo veremos más adelante.

### 3.4 Capa de datos

Tal y como hemos indicado antes, la aplicación está pensada como una ventana a los contenidos proporcionados por la API de la UOC. Por tanto, la gestión final de estos datos es responsabilidad del Campus Virtual. Aun así, la aplicación debe tener un concepto propio de estos datos, es decir, debe tener una imagen propia de esas estructuras de datos para poder interactuar con ellas, ya sea para descargar esa información como para subirla.

Este modelado de datos se realiza mediante un mapeo de la documentación proporcionada por la UOC de los tipos de datos proporcionados por las respuestas a la API a clases que representan las entidades dentro de nuestro sistema.

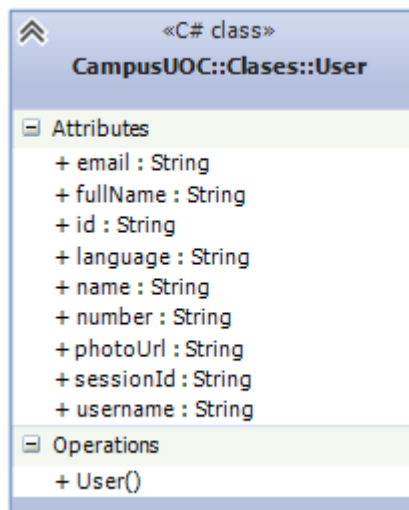
Por ejemplo, según la documentación de la API de la UOC tenemos que el ejemplo de dato devuelto para un dato tipo “Usuario” es:

Example JSON

```
{
  "id" : "130360",
  "username" : "xaracil",
  "name" : "Xavi",
  "number" : "411603",
  "fullName" : "Xavi Aracil Diaz",
  "photoUrl" : "http://cv.uoc.edu/UOC/mc-icons/fotos/xaracil.jpg",
  "language" : "ca",
  "sessionId" :
  "283ac5c62977a38daef52b5a43ec5273f371af839246fdc4e4c71aaf367972b16721a00
  fb25ca8feeb3de5012435c42218d7ae60243c7b5c3a688eb593dfbad2",
  "email" : "xaracil@uoc.edu"
}
```

**Ilustración 6:** Tipo de dato Usuario devuelto por la API UOC

Este tipo de dato será representado en nuestro sistema con una entidad que corresponda con las características y atributos proporcionados por la API. En nuestro ejemplo, la siguiente entidad:



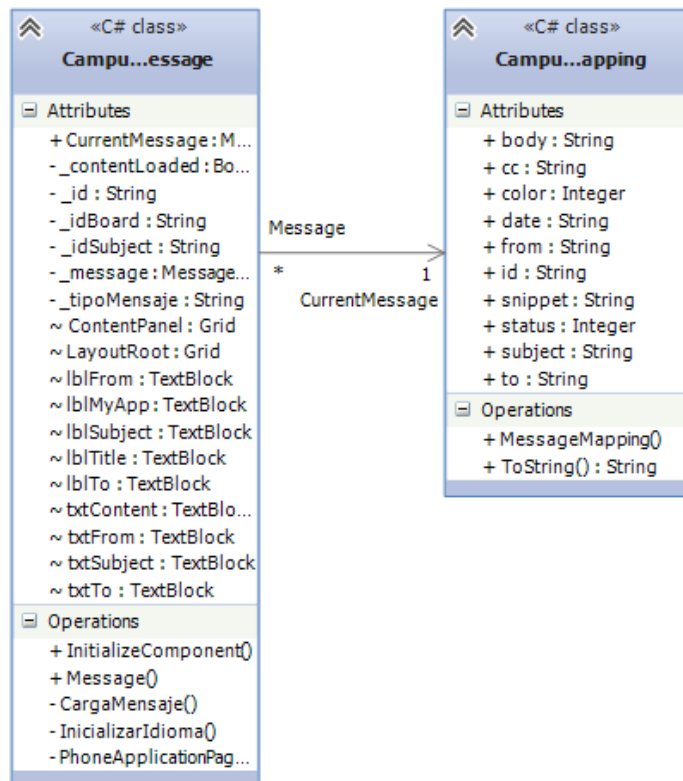
**Ilustración 7:** Representación UML del tipo de dato Usuario en nuestro sistema.

Donde asignamos un atributo correspondiente a cada atributo proporcionado por el tipo de dato que nos devolverá, o deberemos proporcionar mediante la API de la UOC en nuestras llamadas.

En el anexo “Diagrama de entidades de la capa de datos” recogido en el punto 8.2 de este documento se pueden explorar todas las entidades y sus relaciones que integrará el programa.

### 3.4.1 Uso de entidades por la capa de interfaz

En el diagrama de clases la Ilustración 4, vemos como es la comunicación entre las capas. Se puede observar como existe una separación de conceptos entre vista, lógica e integración. Aunque es la capa lógica la encargada de construir los objetos en memoria a partir de los datos obtenidos por la API, esta representación de los datos en nuestro programa es usada por las vistas, que dependiendo del aspecto que queramos dar a la información obtenida usarán dicha información para representarla a gusto del diseñador de la interfaz.



Como vemos en la Ilustración 8, la vista para la visualización de mensajes, con un gran número de atributos y métodos, la mayoría autogenerados. Dispone de una variable llamada "CurrentMessage", que será la representación en memoria que ha construido la capa lógica.

Cuando se inicialice la vista, cuando el usuario desee visualizar un mensaje determinado, la vista pedirá a la capa lógica que obtenga el mensaje deseado, sin conocer realmente cómo la capa lógica obtiene y genera ese mensaje.

**Ilustración 8:** Uso de un tipo de dato por una vista. Representación UML

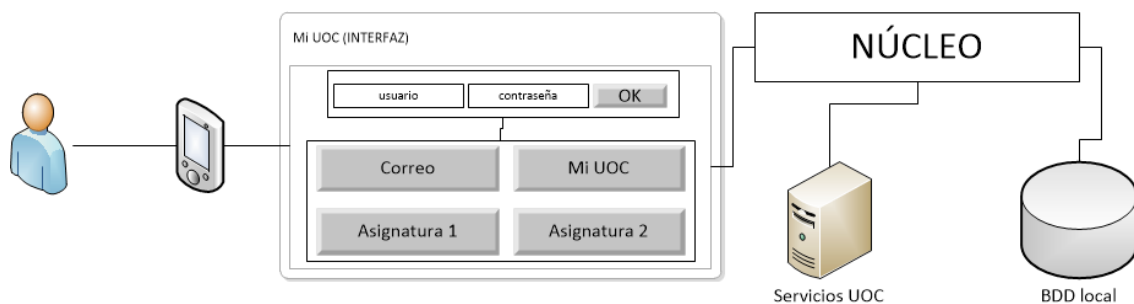
Una vez obtenido ese mensaje, utilizará de acuerdo con lo determinado por el diseñador de la interfaz esa información para mostrársela al usuario.

## 4 Arquitectura y aspectos técnicos

La aplicación estaría descrita en tres capas: interfaz, núcleo e integración.

### 4.1 Núcleo

El núcleo sería la capa encargada de realizar las operaciones lógicas y cálculos estimulados por los eventos lanzados en la capa superior (interfaz). Éste, sería el encargado por tanto de comunicar con el sistema de la UOC a través de los servicios ofrecidos por su sistema, y poder de esta manera interactuar con el sistema externo.



**Ilustración 9: Diagrama básico de la arquitectura**

#### 4.1.1 Autenticación

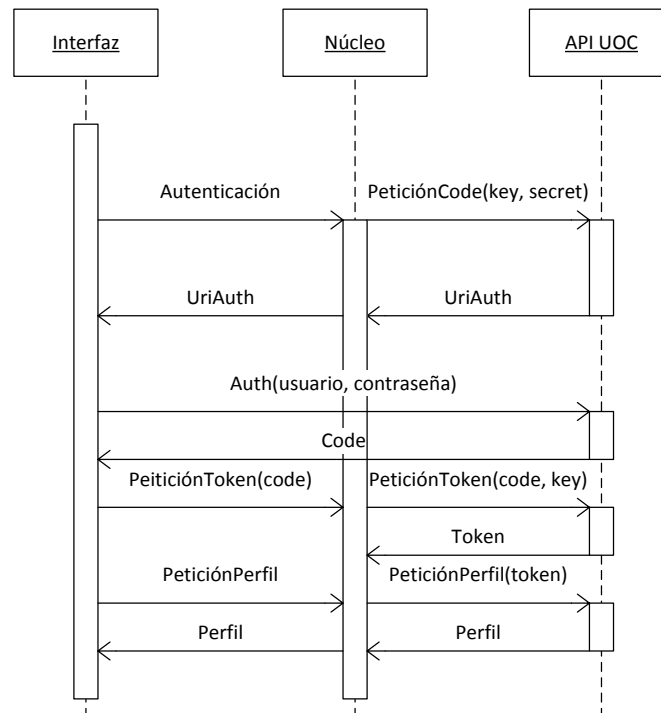
Para la autenticación del núcleo en el sistema se deberá realizar un proceso inicial en el momento de inicio de la aplicación. Se debe realizar una petición a la API de la UOC de un código llamado “code” que se hace a través de una llamada a la web de autorización de la API, donde primero se le pedirá al alumno el usuario y contraseña y después se le pedirá el nivel de permisos que le otorga a la aplicación.

La metodología de autenticación usa el protocolo OAuth [8] [9] para la identificación a través de los servicios ofrecidos por la API, pero para poder comunicar con los servicios es necesario obtener previamente un código llamado “token” que permite el acceso temporal a los servicios de la API. En el momento de la recogida de este código también se obtienen el tiempo de expiración de este código y otro código para la renovación del “token” por si fuera necesario. Pero antes, veamos el proceso de comunicación necesario para obtener el código “token”.

Si nos fijamos en la Ilustración 10, para realizar la autenticación tenemos tres pasos principales:

- La petición de la URI de autenticación
- La autenticación desde el navegador para obtener el “code”
- La petición del “token”

Para representar mejor el flujo de trabajo podemos ver la siguiente ilustración, un diagrama de secuencia que muestra los pasos a seguir para realizar una autenticación correcta en nuestro sistema. Además, realizaremos un pequeño desglose del diagrama para entenderlo mejor.



**Ilustración 10: Diagrama de secuencia de autenticación y servicio correctos con la API UOC**

#### 4.1.1.1 Petición de la Uri de autenticación

El primer paso de autenticación, que se lanzará en el inicio de la aplicación realizará una petición a través de una petición REST [10] con la que obtendremos la Uri de para presentarla en el navegador integrado en la aplicación para la autenticación en el sistema UOC.

Para realizar esta petición, el núcleo le enviará el identificador de la aplicación, llamado “key”, que se ha pedido previamente por el desarrollador de la aplicación, y una contraseña llamada “secret”, que la API utilizará para conocer cuál es la aplicación a la que debe otorgar permisos para acceder a la información.

#### 4.1.1.2 Autenticación desde la interfaz

Una vez recibida la Uri de autenticación de la UOC, ésta se lanzará al navegador web integrado en la aplicación. Donde será visualizará la interfaz de la UOC para la otorgar los permisos necesarios al sistema.

El usuario, que no percibirá la primera petición en la interfaz, en cambio, verá a través del control de navegador de Internet integrado en la interfaz que se le pide el usuario y contraseña, y que debe otorgar unos permisos a la aplicación para poder acceder a su información personal.

Cabe indicar, que esta petición se hace directamente desde el control de navegador de la interfaz y el usuario, y que es en el caso de una autenticación correcta es la interfaz la encargada de recoger el código intermedio “code” antes de la petición del código de autenticación final “token” que otorgará los permisos de acceso a la API.



Esta autenticación se realiza mediante *https* [11], lo que supone un canal seguro de comunicación entre el alumno y el servidor. Y proporciona a la aplicación un acceso confiable durante un tiempo razonable de uso.

#### 4.1.1.3 La petición del “token”

El último paso se trata la petición del “token”, un código de identificación final que tiene un periodo de vida limitado y que es el que el sistema API pedirá para poder acceder a los datos.

El token, está asociado al usuario que ha otorgado los permisos de acceso y a la aplicación desde donde se ha pedido. Para realizar la petición se deben pasar correctamente los dos puntos anteriores y haber obtenido el “code”.

### 4.1.2 Comunicación con la API

Una vez recogido el “token” [12], la comunicación con la API se realiza mediante el protocolo REST, al igual que la comunicación para obtener la autenticación a excepción de la obtención del “code”. Este protocolo permite el acceso a la información de la UOC mediante unas sencillas llamadas a una ruta relativa sobre el servidor de la UOC.

#### 4.1.2.1 El servicio de información de la UOC (API)

La UOC tiene a disposición de los alumnos un sistema de comunicación simulado del Campus Virtual, tal y como la página web de información [4] de la UOC indica:

*“Este es el entorno de desarrollo de aplicaciones para el Campus Virtual de la UOC. Utilizando los servicios de esta API, podrás construir tus aplicaciones Campus para móviles, tabletas, ordenadores o lo que quieras.”*

De esta manera, la UOC proporciona un entorno abierto y seguro que podemos usar en nuestra aplicación para poder obtener la información necesaria para alimentar nuestra aplicación móvil.

#### 4.1.2.2 Servicios ofrecidos por la API

Tal y como se ha descrito anteriormente, la API se basa en la arquitectura REST para realizar las peticiones a los servicios de la UOC. Estas peticiones incluyen la autenticación mediante OAuth como se describe en el punto 4.1.1.

La URL [13] raíz de la API es:

`http://oslo.uoc.es:8080/webapps/uocapi`

La ruta para la autenticación cuelga desde:

`http://oslo.uoc.es:8080/webapps/uocapi/oauth/authorize`

Como se puede comprobar, los recursos y servicios se organizan mediante rutas relativas. Debiendo haber obtenido una autenticación correcta previa a la consulta de los mismos.

El resto de servicios ofrecidos se ofrecen resumidos bajo el siguiente listado, todos ellos teniendo como raíz de la ruta la dirección indicada anteriormente.

Servicio	Rutas
<p>Bajo este dominio disponemos de la posibilidad de listar los eventos de calendario del alumno, así como los datos sobre los diferentes eventos.</p>	<ul style="list-style-type: none"> <li>• /calendar/</li> <li>• /calendar/events/{id}</li> </ul>
<p>Bajo este dominio, el más completo de todos, podremos obtener diversa información relativa a las asignaturas, más en concreto a sus aulas virtuales, en las que se encuentra matriculado el alumno.</p> <p>Así como los diferentes recursos y miembros de la asignatura. Materiales, foros, miembros y recursos educativos pueden ser explorados a través de este subdominio.</p>	<ul style="list-style-type: none"> <li>• /classrooms</li> <li>• /classrooms/{id}</li> <li>• /classrooms/{id}/boards</li> <li>• /classrooms/{id}/groups</li> <li>• /classrooms/{id}/materials</li> <li>• /classrooms/{id}/people</li> <li>• /classrooms/{id}/resources</li> <li>• /classrooms/{domain_id}/boards/{id}</li> <li>• /classrooms/{domain_id}/materials/{id}</li> <li>• /classrooms/{domain_id}/resources/{id}</li> <li>• /classrooms/{id}/people/students</li> <li>• /classrooms/{id}/people/teachers</li> <li>• /classrooms/{domain_id}/boards/{id}/folders</li> <li>• /classrooms/{domain_id}/boards/{id}/messages</li> <li>• /classrooms/{domain_id}/boards/{board_id}/folders/inbox</li> <li>• /classrooms/{domain_id}/boards/{board_id}/folders/{id}</li> <li>• /classrooms/{domain_id}/boards/{board_id}/messages/{id}</li> <li>• /classrooms/{domain_id}/boards/{id}/messages/unread</li> <li>• /classrooms/{domain_id}/boards/{board_id}/folders/{id}/messages</li> <li>• /classrooms/{domain_id}/boards/{board_id}/folders/{folder_id}/messages/{id}</li> <li>• /classrooms/{domain_id}/boards/{board_id}/folders/{id}/messages/unread</li> <li>• /classrooms/{domain_id}/boards/{board_id}/messages/{id}/move/{folder_id}</li> <li>• /classrooms/{domain_id}/boards/{board_id}/folders/{folder_id}/messages/{id}/body</li> <li>• /classrooms/{domain_id}/boards/{board_id}/folders/{folder_id}/messages/{id}/history</li> <li>• /classrooms/{domain_id}/boards/{board_id}/folders/{source_id}/messages/{id}/move/{destination_id}</li> </ul>

<p>Para la gestión del correo de la UOC disponemos del subdominio <i>mail</i> encargado de ofrecer los listados de carpetas, mensajes y remitentes de los correos electrónicos.</p>	<ul style="list-style-type: none"> <li>• /mail/folders</li> <li>• /mail/messages</li> <li>• /mail/folders/inbox</li> <li>• /mail/folders/{id}</li> <li>• /mail/messages/unread</li> <li>• /mail/messages/{id}</li> <li>• /mail/folders/{id}/messages</li> <li>• /mail/messages/{id}/body</li> <li>• /mail/messages/{id}/history</li> <li>• /mail/folders/{id}/messages/unread</li> <li>• /mail/messages/{id}/move/{folder_id}</li> <li>• /mail/folders/{source_id}/messages/{id}/move/{destination_id}</li> </ul>
<p>Este dominio es el destinado a la información de los miembros de la UOC.</p>	<ul style="list-style-type: none"> <li>• /people</li> <li>• /people/{id}</li> <li>• /people/{id}/profiles</li> <li>• /people/{id}/tutors</li> <li>• /people/{id}/profiles/current</li> </ul>
<p>Este dominio se encarga de ofrecer la información general de las asignaturas en las que está matriculado el alumno.</p> <p>No se debe confundir este dominio con el dominio <i>classrooms</i>, que ofrece la información del aula virtual de las asignaturas. Aunque muchos de los datos serán redundantes.</p>	<ul style="list-style-type: none"> <li>• /subjects</li> <li>• /subjects/{id}</li> <li>• /subjects/{id}/boards</li> <li>• /subjects/{id}/resources</li> <li>• /subjects/{domain_id}/boards/{id}</li> <li>• /subjects/{domain_id}/resources/{id}</li> <li>• /subjects/{domain_id}/boards/{id}/folders</li> <li>• /subjects/{domain_id}/boards/{id}/messages</li> <li>• /subjects/{domain_id}/boards/{board_id}/folders/inbox</li> <li>• /subjects/{domain_id}/boards/{board_id}/folders/{id}</li> <li>• /subjects/{domain_id}/boards/{board_id}/messages/{id}</li> <li>• /subjects/{domain_id}/boards/{id}/messages/unread</li> <li>• /subjects/{domain_id}/boards/{board_id}/folders/{id}/messages</li> <li>• /subjects/{domain_id}/boards/{board_id}/folders/{folder_id}/messages/{id}</li> <li>• /subjects/{domain_id}/boards/{board_id}/folders/{id}/messages/unread</li> <li>• /subjects/{domain_id}/boards/{board_id}/messages/{id}/move/{folder_id}</li> <li>• /subjects/{domain_id}/boards/{board_id}/folders/{folder_id}/messages/{id}/body</li> <li>• /subjects/{domain_id}/boards/{board_id}/folders/{folder_id}/messages/{id}/history</li> <li>• /subjects/{domain_id}/boards/{board_id}/folders/{source_id}/messages/{id}/move/{destination_id}</li> </ul>

Este es el dominio encargado de ofrecer información personal del alumno que se ha autenticado en la aplicación, las relaciones con el resto de miembros de la comunidad UOC y su configuración personal.	<ul style="list-style-type: none"> <li>• /user</li> <li>• /user/profiles</li> <li>• /user/settings</li> <li>• /user/tutors</li> <li>• /user/profiles/current</li> </ul>
--	---

Mediante la comunicación a estos servicios alimentaremos a la aplicación móvil, de un modo organizado y seguro. Además, gracias al uso de estándares de comunicación digital podemos interpretar y comunicarnos con la API desde varios tipos diferentes de dispositivos, y por supuesto, desde nuestra aplicación.

#### 4.1.2.3 La librería RestSharp

RestSharp [14] es una librería para .Net, y que soporta Windows Phone, que nos ayudará a comunicarnos con los servicios de la UOC. Esta librería es capaz de manejar el protocolo de comunicación REST y de devolver el contenido de las respuestas a nuestra aplicación mediante una serie de funcionalidades incluidas en la misma.

Esta librería, además, incorpora funcionalidades extra, como el soporte para la autenticación segura a través de OAuth y la deserialización de respuestas en diferentes formatos.

#### 4.1.2.4 Formato de respuesta de las llamadas a la API

La alimentación supone la constante “traducción” del formato de comunicación de los servicios de la UOC con los datos locales.

Por ejemplo, tras una petición de los datos personales de un usuario desde la ruta “/user” el servicio nos devolverá un contenido similar al siguiente:

```
{"id":"666666","name":"Harriet","number":"666666","username":"user6","fullNameName":"Harriet Upp","photoUrl":"http://cv.uoc.edu/UOC/mc-icons/fotos/desconegut.png","language":"ca","sessionId":"a72b9b63467a94d9cfb2fdc972c0e5f24b5eb8b55e6cf5e79be9981b040c3d02e68ccf27b117c79d0c9d3f115195cb705587c849ecaf035c202610dbea4ce688","email":"user6@uoc.edu"}
```

**Ilustración 11:** Código JSON de respuesta

Este formato corresponde a JSON [15] [16], es un formato ligero de intercambio de información. Es relativamente sencillo interpretarlo y generarlo y es el formato usado en las respuestas de la API.

También tenemos el formato XML [17] como posible respuesta a las llamadas de la API, otro método sencillo de transporte de la información de una manera sencilla y fácil de interpretar y generar.

## 4.2 Integración y Base de datos

Sería la capa encargada de salvaguardar la información. Aunque la mayor parte de ella provendrá del sistema UOC, parte de ella sería almacenada localmente. Como podría ser el usuario y contraseña, borradores de mensajes, caché de la

información para poder consultar cuando el dispositivo no tenga conexión a Internet, etc.

La idea de la asignatura es realizar una presentación del entorno de la UOC, donde la mayoría de la información resultará ajena a la misma, pero no todos los datos serán otorgados por la API de la UOC. Algunos de ellos podremos guardarlos en nuestro sistema. Por ejemplo, nuestra aplicación será multilingüaje, y es responsabilidad de nuestro sistema.

#### 4.2.1 Representación de los datos. Entidades.

En principio la aplicación está pensada como un *frontpage* para que el alumno interactúe con la UOC. Esto significa que el sistema de gestión de la base de datos resultará ser algo externo a la aplicación, a nuestro sistema. Nuestra responsabilidad será la de ofrecer al usuario una capa para interactuar con la API de la UOC.

Como hemos comentado, la mayor parte de los datos del entorno de la UOC serán guardados por un sistema ajeno al nuestro, pero esto no significa que no debemos modelarlo en el nuestro para poder trabajar con él. En la documentación otorgada para trabajar con la API se proporcionan las salidas JSON que ofrece cada una de las llamadas.

Mediante JSON tendremos una representación de las salidas a las llamadas de las entidades de la base de datos del entorno virtual de la UOC. Por ejemplo, para la llamada `/classrooms` definida en el punto 4.1.2.4, que es un servicio que devuelve la información relativa a las asignaturas en las que se encuentra matriculado un alumno, tenemos que la respuesta tiene el siguiente formato:

```
{
  "classrooms": [
    {
      "id": "1111111",
      "title": "Introduction to Java Programming Aula 1",
      "color": "#308961",
      "fatherId": "1111111",
      "code": "05.554",
      "shortTitle": "IJP",
      "assignments": ["ESTUDIANT"]
    },
    {
      "id": "3333331",
      "title": "Introduction to C Programming Aula 1",
      "color": "#308961",
      "fatherId": "3333333",
      "code": "05.564",
      "shortTitle": "TMS",
      "assignments": ["ESTUDIANT"]
    },
    {
      "id": "5555551",
      "title": "Test Lab subject Aula 1",
      "color": "#308961",
      "fatherId": "5555555",
      "code": "05.073",
      "shortTitle": "TLS",
      "assignments": ["ESTUDIANT"]
    }
  ]
}
```

Esta salida de ejemplo representa una lista de asignaturas, donde la primera de ellas es la asignatura "Introduction to Java Programming Aula 1" con el identificador "1111111" que es un objeto de lo que sería la clase "Classroom" de nuestro sistema. Por tanto, esta salida nos ofrece una lista donde cada uno de sus nodos contiene la información relativa a una asignatura, es decir una lista con entidades asignaturas como nodos, donde cada uno de ellos representa los datos relativos a las asignaturas a las que está matriculado el alumno. Estas entidades,

podremos usarlas virtualmente para mostrar la información relativa a las mismas en nuestro entorno. La clase que representa en nuestro entorno a esta entidad se define de la siguiente manera:

```
using System;
using System.Text;

namespace CampusUOC.Clases
{
    public class Classroom
    {
        public string id { get; set; }
        public string title { get; set; }
        public string color { get; set; }
        public string fatherId { get; set; }
        public List<string> assignments { get; set; }
        public string code { get; set; }
        public string shortTitle { get; set; }
    }
}
```

Ilustración 12: Código (C#) de la clase Classroom

Esta clase representa y tiene las mismas propiedades que nos ofrece la API, representando dichas propiedades con variables creadas a imagen del JSON que nos ofrece la API. En este caso concreto tenemos que la asignatura se define con un identificador único (id) para nuestro sistema, un título (title), un color (color), un identificador del padre (fatherId), unas responsabilidades del usuario (assignments), un código identificativo para la universidad (code) y un título corto o alias (shortTitle).

Y la clase que define la lista de asignaturas es definida de la siguiente manera:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace CampusUOC.Clases
{
    public class Classrooms
    {
        public List<Classroom> classrooms { get; set; }
    }
}
```

Ilustración 13: Código (C#) de la clase Classrooms

En este caso un conjunto de asignaturas a las que está matriculado el alumno se representan como una lista de la clase "Classroom". Una vez instanciada esta lista, tendremos una serie de asignaturas en las que el alumno podrá estar matriculado.

#### 4.2.2 La librería Newtonsoft JSON Deserializer

Para realizar la traducción de estos datos JSON en entidades hemos utilizado una librería específica para ello llamada *NewtonSoft.Json*. Esta librería rellenará los datos y generará las entidades obtenidas en la llamada.

Una llamada de esta librería para la respuesta vista anteriormente y contenida en la variable *Content* de la respuesta, nos daría una lista de sólo un elemento del objeto instanciado de la clase "Classroom" que representa dicha entidad para nuestro sistema.

Este objeto tiene varias propiedades o características propias que definen a la entidad. Por ejemplo, la clase estará definida por su nombre, alias, código de identificación, etc. Parte de los datos ofrecidos por la plataforma serán útiles para la capa de presentación, pero parte de ellos nos serán útiles para explorar el sistema. El código identificativo de la asignatura nos servirá para realizar otras llamadas a la API, donde parametrizaremos este valor para obtener otras identidades que estarán relacionadas directamente con la entidad anterior. Por ejemplo, los recursos educativos de la misma.

Esta librería nos facilitará la traducción e interpretación de las respuestas de la API mediante el protocolo JSON. Esta librería, dispone de gran cantidad de funciones para trabajar con ficheros JSON e interactuar con sistemas de desarrollo .Net. Es de amplio uso para los entornos de desarrollo y proporciona una serie de funciones eficaces y prácticas para traducir en ambos sentidos un JSON, que al fin y al cabo es una sencilla cadena de caracteres con una sintaxis basada en la separación de elementos mediante comas, entidades mediante llaves y de propiedades y valores mediante el signo de los dos puntos.

Una propiedad de JSON representa se traduce como una variable de una clase en C# [18], mediante que una entidad se representa como una clase completa. Cuando se realiza la traducción se crea dinámicamente el objeto instanciado de esa clase representando la entidad que se nos ha enviado mediante la sintaxis de JSON.

#### 4.2.3 Llamada a la API y traducción de JSON

Como hemos visto en los puntos anteriores, se realiza una llamada a la API para consultar información y posteriormente se realiza la traducción del JSON. Para simplificar las llamadas en la interfaz, se crean unos métodos públicos estáticos y asíncronos que son accesibles desde la capa de interfaz. Como ejemplo podemos ver el de la Ilustración 14.

```
public static async Task<Classrooms> GetClasses()  
{  
    return  
        (await ExecuteAsync<Classrooms>(ConstruyePetición(  
            Config.SUBJECTS_URI, new Dictionary<string, string>())));  
}
```

Ilustración 14: Ejemplo de la llamada a *ExecuteAsync*

En él simplemente ofrecemos un sencillo método que ofrecerá una tarea asíncrona, que podemos usar para obtener las clases en las que está matriculado el alumno.

```
this._clases = await Procedimientos.GetClasses();
```

Esta llamada esperará, gracias al comando `await` a que se resuelva la tarea de llamada y respuesta y se realice la conversión de clases. Para que todas las llamadas resulten sencillas hemos creado dos métodos que simplifican la tarea del programador.

El primero de ellos es el método `ConstruyePetición`, este método construye la petición que usará la librería `REST+OAuth` para realizar la llamada. Antes debemos habernos identificado y recogido el token para poder autenticarnos en el sistema y que la llamada sea válida.

```
public static RestRequest ConstruyePetición(string uriFuncion,
Dictionary<string, string> parametros, int timeOut)
{
    var peticio = new RestRequest(uriFuncion);
    if(timeOut > 0)
peticio.Timeout = timeOut;
    peticio.AddParameter("access_token",
LicenciaActual.Licencia.access_token);
    peticio.AddParameter("client_id", Config.ConsumerKey);
    peticio.AddParameter("client_secret", Config.ConsumerSecret);

    foreach (var parametro in parametros)
    {
peticio.AddParameter(parametro.Key, parametro.Value);
    }

    return peticio;
}
```

**Ilustración 15: Método ConstruyePetición**

Para construir la petición, de manera genérica, añadimos las variables necesarias a la petición, como son el identificador de la aplicación, el token y el secret. Además el método acepta poder añadir otras variables que necesitemos añadir y finalmente devuelve una petición construida, preparada para ser lanzada. Una vez creada esta petición, para la librería `REST` pasaremos a realizar la llamada. Para realizar las llamadas usamos el método `ExecuteAsync`.

```
public static async Task<T> ExecuteAsync<T>(RestRequest request)
{
    RestClient _client = ConstruyeCliente();
    _client.Timeout = request.Timeout > 0 ? request.Timeout : 5*1000;
    var tcs = new TaskCompletionSource<T>();

    Debug.WriteLine(request.Resource);
}
```



```
        _client.ExecuteAsync(request, resp =>
        {
if (resp.ResponseStatus != ResponseStatus.Completed)
{
    Debug.WriteLine(resp.ErrorMessage);
    var ex = new Exception(resp.ErrorMessage);
    tcs.TrySetException(ex);
}
else
{
#if DEBUG
    Debug.WriteLine(typeof(T) + " => " + resp.Content);
#endif

    if (resp.Content.StartsWith("<html><head><title>JBossWeb/2.0.1.GA
- Error report</title><style>"))
tcs.TrySetCanceled();

    var value = JsonConvert.DeserializeObject<T>(resp.Content);
    tcs.TrySetResult(value);
}

});

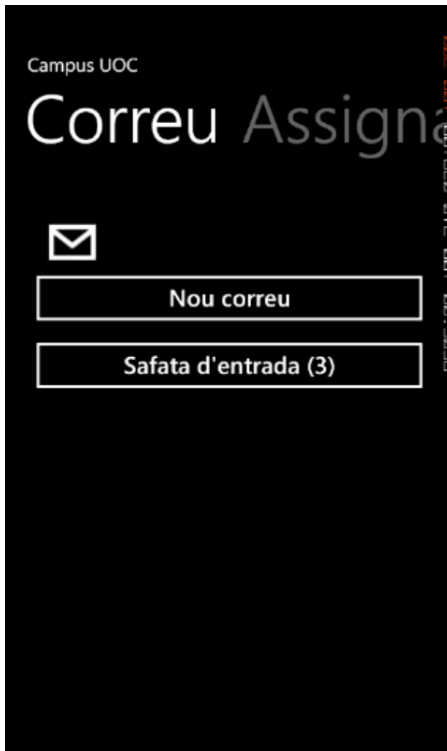
return await tcs.Task;
}
```

Ilustración 16: Método ExecuteAsync

La llamada a la API es un método complejo. Este método realiza una llamada asíncrona, aceptando un tipo genérico T [19] que se adapta a cualquier entidad representada en la aplicación móvil, siempre y cuando tenga equivalente con la entidad que devuelva la llamada. Al aceptar la librería *NewtonSoft JSON* también un tipo genérico como respuesta, obtendremos, previa comprobación a que no exista un error en la llamada o en la respuesta, una entidad del tipo genérico, en nuestro caso Classroom, de acuerdo con el tipo de respuesta que esperábamos según la llamada que hayamos hecho. Cada llamada tiene un tipo de respuesta propio que se encuentra documentado en el blog de la API.

### 4.3 Interfaz

En este apartado describiremos al aspecto visual final que tendría la aplicación, que sería la encargada de la interacción persona-ordenador. Cabe indicar que dicha interfaz se maneja de manera táctil y por tanto resultaría muy intuitiva, además usaría controles y el estilo proporcionado por las herramientas de desarrollo de Windows Phone, por lo que sería muy similar al aspecto de aplicaciones realizadas para el mismo sistema, ayudando a interactuar con el mismo.



La interfaz contaría con un primer diálogo de entrada, donde se pedirían el nombre de usuario y contraseña del alumno.

Posteriormente veríamos un menú de entrada, donde podríamos elegir entre las diferentes funcionalidades diferentes, el diseño final dependerá de las posibilidades ofrecidas por el diseñador y la usabilidad de la misma, pero tendría un aspecto similar a la Ilustración 17.

Desde este menú podríamos acceder a las funcionalidades principales de la aplicación. Destacamos en este apartado las dos principales catalogadas en los requisitos:

- El buzón: Donde listaríamos y podríamos visualizar los mensajes del buzón de la UOC.
- Asignaturas: Donde veríamos el listado o calendario de las actividades.

**Ilustración 17: Prototipo de la aplicación. Buzón.**

En las dos anteriores cabe indicar que se intentarían implementar el resto de funcionalidades descritas en el apartado de requisitos.

### 4.3.1 Vistas

En este apartado realizaremos un pequeño esbozo de lo que tendremos en la aplicación final, para ello dividiremos en los diferentes siguientes apartados las diferentes vistas.

#### 4.3.1.1 Login

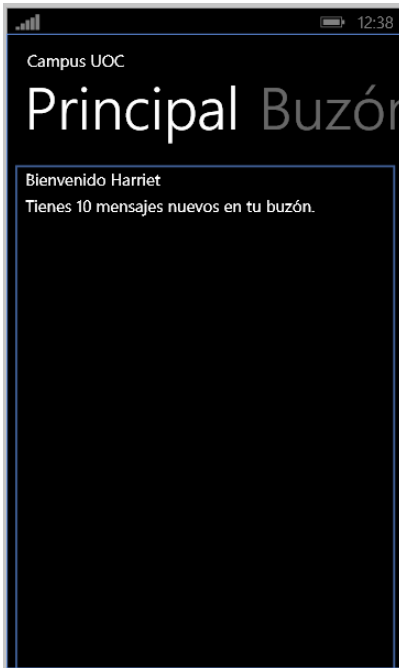
La página de entrada de la aplicación será la de autenticación en el caso que el alumno no lo haya hecho con anterioridad o haya expirado la sesión.

Para ello, debemos mostrar el navegador que otorgará el acceso a la aplicación mediante la autenticación descrita en el apartado 4.1.1.

#### 4.3.1.2 Página de entrada

Tras una autenticación correcta pasaríamos a la vista principal de la aplicación. Esta vista mostraría los diferentes enlaces para pasar a las diferentes vistas que comprenden las funcionalidades del sistema.





**Ilustración 19: Vista del prototipo de la aplicación. Página principal.**

La página de presentación inicial será una página pivotada. De este modo podremos desplazarnos fácilmente con el dedo de una sección a otra.

La página pivotada, es una primera página inicial, situada a la izquierda de una distribución horizontal de páginas. De este modo, al arrastrar el dedo de derecha a izquierda de la pantalla pasaremos a la página inmediatamente a la derecha y viceversa.

Las páginas estarán ordenadas como: Principal, Buzón (correo electrónico), Calendario, Asignaturas y Configuración. Estas serían las páginas y su organización. Cada una de ellas tendría el contenido y configuración propio para su manejo.

#### 4.3.1.3 Principal

La página principal tendrá un pequeño texto de entrada con un saludo cordial y breve información sobre las posibles novedades que nos pueda ofrecer el sistema.

#### 4.3.1.4 Buzón

El buzón es donde tendremos organizado en botones los distintos botones de con las funcionalidades de la aplicación, como la entrada a la Bandeja de entrada o los mensajes enviados.

#### 4.3.1.5 Calendario

El calendario mostrará el mes en curso y marcará los días que correspondan a un evento. Al hacer clic sobre el día podremos ver los detalles del evento en concreto.

#### 4.3.1.6 Asignaturas

Esta sección mostrará un listado de las asignaturas como botones que se agregarán programáticamente. Al hacer clic sobre la asignatura podremos ver los detalles de las asignaturas.

#### 4.3.1.7 Configuración

Esta página está enfocada a mantener las opciones y configuración de la aplicación. Por ejemplo, podremos guardar el idioma que queremos que se use para la interfaz entre los disponibles. De este modo, aunque tengamos el móvil con una interfaz en inglés, podremos elegir que se muestre en catalán.

### 4.3.2 Multilinguaje

La aplicación será multilinguaje. Para implementar esta función nos hemos servido del conjunto de herramientas “Multilingual App Toolkit” [20] proporcionado por Microsoft para integrar diferentes idiomas en una sola aplicación.



Ilustración 20: Interfaz multilinguaje

Para ello, usaremos un fichero de recursos con un diccionario genérico que referenciará a un fichero dependiente del idioma para cada término de este diccionario. De este modo, cuando demos un valor determinado a un texto de la interfaz:

```
private void InicializarIdioma()
{
    lblDebeAutenticarse.Text = AppResources.YouMustAuthenticate;
    PageTitle.Text = AppResources.Login;
    ApplicationTitle.Text = AppResources.ApplicationTitle;
}
```

Ilustración 21: Código de inicialización genérica del idioma

El término “YouMustAuthenticate” se traducirá mediante las herramientas proporcionadas por el kit al catalán, castellano e inglés. Cuando el programa se inicie por primera vez, se recogerá el idioma en el que se muestra la interfaz, siendo el idioma por defecto el inglés, para mostrar la interfaz en ese determinado idioma. El idioma será configurable, y reestablecido con las preferencias del usuario una vez inicie la aplicación.

## 5 Herramientas de desarrollo

Para desarrollar la aplicación se usaría un entorno de desarrollo Microsoft Visual Studio 2012, con licencia proporcionada por el DotNetClub de la UOC. Un ordenador personal portátil convencional equipado con un procesador Intel core i5 (necesario para la virtualización del emulador de Windows Phone 8) y para las pruebas finales un dispositivo electrónico móvil HTC 8S con un sistema operativo Windows Phone integrado.

## 6 Conclusiones

Durante este cuatrimestre hemos puesto en práctica buena parte de los conocimientos adquiridos a lo largo de nuestros estudios y carrera profesional. Donde también hemos aprendido y evaluado nuestra capacidad de aprendizaje autónoma, aprendiendo además de nuevas técnicas, sobre nosotros mismos.

Muchas de las asignaturas que se han ido realizando a lo largo del Grado y han sido de gran utilidad a la hora de realizar buena parte de este trabajo. Además, la experiencia sobre otros entornos ha dado pie a una buena adaptación al entorno de desarrollo de aplicación. Además de aprender nuevas filosofías en el diseño de aplicaciones móviles, que entrañan nuevos retos para los desarrollos del futuro al que el alumno se pueda enfrentar. Este nuevo entorno ha supuesto un reto, ya que la experiencia sobre aplicaciones móviles del alumno era mínima, y precisamente por ser un reto es lo que ha llevado al alumno a escoger este tipo de entorno de desarrollo, además del creciente auge que este tipo de aplicaciones está aconteciendo en el cada vez más mundo portátil, donde toda nuestra información acaba siendo contenida en la nube de información y nuestros dispositivos de mano son la ventana a ella.

Todo esto unido al hecho de ser una aplicación conectada con el mundo, conectada con otro sistema basado en tecnologías totalmente diferentes a las de implementación, y con las que hemos interactuado gracias a los sistemas de comunicación e interacción proporcionados por la UOC. Consiguiendo que toda esta comunicación converja en la mejora de los servicios ofrecidos por la universidad, y alimentando la idea de que cada vez el mundo está más interconectado entre sí.

Gracias a este trabajo, hemos elaborado durante todo el proceso de ingeniería del software una nueva aplicación. Esta completa elaboración ha incluido tareas de análisis, elicitación, diseño, desarrollo, implementación y corrección, y todo ello en un sistema parcialmente desconocido y nuevo para el alumno. Donde hemos abarcado desde detalles de implementación, comunicación o usabilidad de nuestra herramienta.

También hemos aprendido a autoevaluarnos, por ejemplo, analizando los logros conseguidos, como se puede dilucidar en el apartado 2.4 de este documento, evaluando punto por punto los requisitos iniciales de la aplicación y la completitud del producto obtenido. Donde, por lo general, hemos alcanzado gran parte de los requisitos propuestos, creando una herramienta llena de funciones, pero que parezca sencilla al mismo tiempo. Respetando en todo momento los

patrones de diseño de interfaces propuestos por el entorno de desarrollo de la plataforma .Net para aplicaciones Windows Phone, resultando una herramienta sencilla, intuitiva y fácilmente usable.

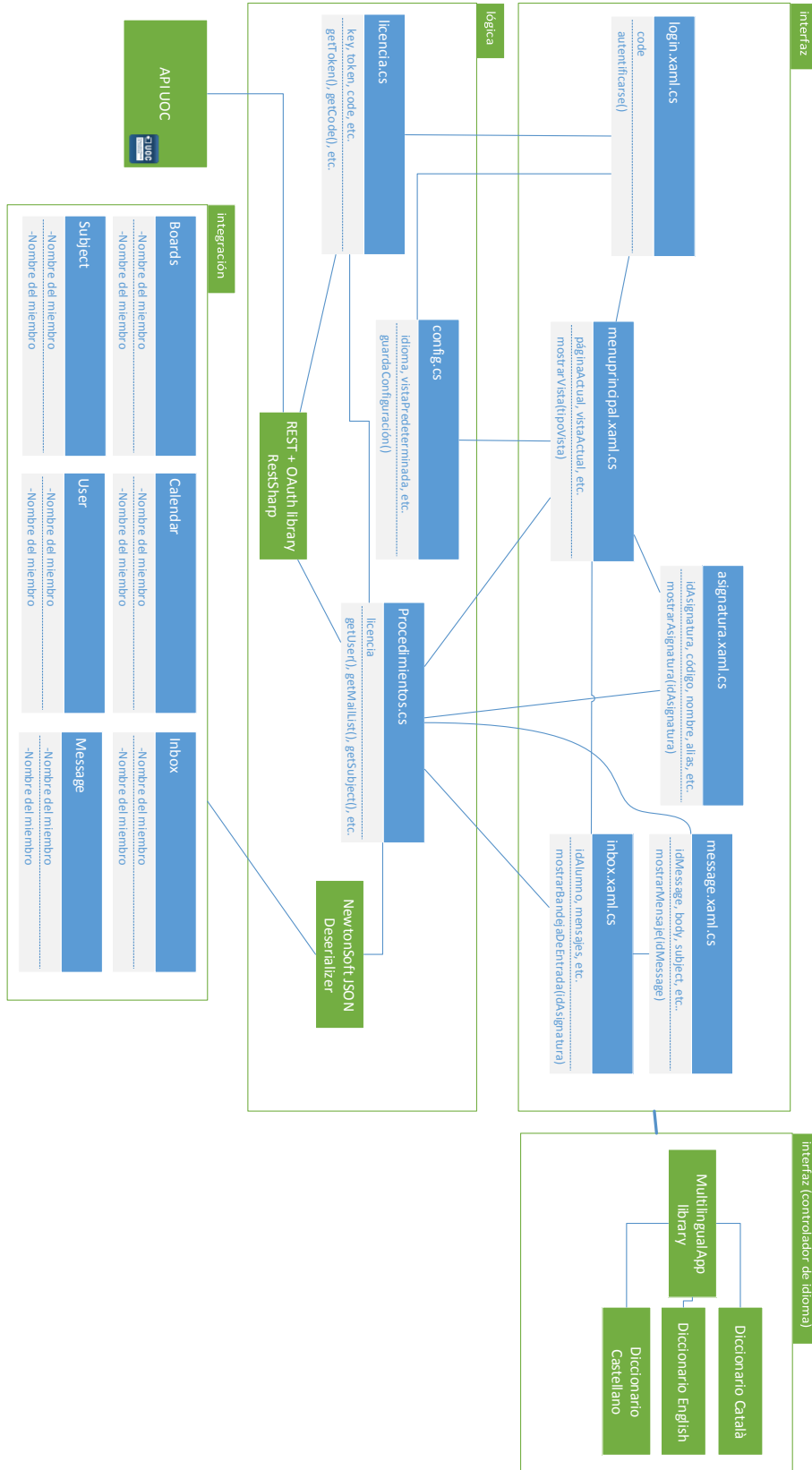
Además terminamos este trabajo con un gusto agridulce, en parte, por las líneas de trabajo que no pudieron ser terminadas en el producto final, y por otro lado, por dejar y terminar unos estudios que han supuesto un esfuerzo adicional al trabajo diario, y que ahora el alumno recuerda con añoranza. Han sido muchas horas de sobre esfuerzo y tazas de café para poder alcanzar una meta que al principio parecía muy lejana. Ahora, poniendo un ojo en el futuro, el alumno está seguro que este esfuerzo ha valido la pena y que los frutos de este fuerzo se verán recompensados a lo largo de su bagaje profesional, no por el título que se obtendrá, si no por lo aprendido en estas horas de trabajo.

## 7 Bibliografía y fuentes citadas

- [1] Microsoft Corporation, «Microsoft - Características de Windows Phone,» [En línea]. Available: <http://www.windowsphone.com/es-ES/features/all>. [Último acceso: 10 2013].
- [2] La Vanguardia Ediciones SL, «LaVanguardia.com,» 1 10 2013. [En línea]. Available: <http://www.lavanguardia.com/tecnologia/20131001/54390268104/windows-phone-cuota-mercado-europa.html>.
- [3] WikiMedia, «WikiPedia - Interfaz de Programación de Aplicaciones,» [En línea]. Available: [http://es.wikipedia.org/wiki/Interfaz\\_de\\_programaci%C3%B3n\\_de\\_aplicaciones](http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones). [Último acceso: 10 2013].
- [4] Univertitat Oberta de Catalunya, «UOC public API,» [En línea]. Available: <http://blogs1.uoc.es/developer/documentacio/uoc-public-api/>. [Último acceso: 10 2013].
- [5] Google Inc., «Android,» [En línea]. Available: <http://www.android.com/>.
- [6] Apple, «Apple - iOS,» [En línea]. Available: <http://www.apple.com/es/ios/>.
- [7] Microsoft Corporation, «XAML (WPF),» [En línea]. Available: <http://msdn.microsoft.com/es-es/library/ms752059.aspx>. [Último acceso: 10 2013].
- [8] OAuth community, «OAuth - secure authorization open protocol,» [En línea]. Available: <http://oauth.net/>. [Último acceso: 10 2013].
- [9] E. Hardt y I. Internet Engineering Task Force, «The OAuth 2.0 Authorization Framework - RFC 6749,» [En línea]. Available: <http://tools.ietf.org/html/rfc6749>. [Último acceso: 10 2013].
- [10] WikiMedia, «WikiPedia - Representational State Transfer,» [En línea]. Available: [http://es.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://es.wikipedia.org/wiki/Representational_State_Transfer). [Último acceso: 10 2013].
- [11] WikiMedia, «WikiPedia - Hypertext Transfer Protocol Secure,» [En línea]. Available: [http://es.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol\\_Secure](http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol_Secure). [Último acceso: 10 2013].
- [12] WikiMedia, «WikiPedia - Security Token,» [En línea]. Available: [http://en.wikipedia.org/wiki/Security\\_token](http://en.wikipedia.org/wiki/Security_token). [Último acceso: 10 2013].
- [13] WikiMedia, «WikiPedia - Localizador uniforme de recursos,» [En línea]. Available: <http://es.wikipedia.org/wiki/Url>. [Último acceso: 10 2013].
- [14] RestSharp project, «RestSharp - Simple REST and HTTP API Client for .NET,» [En línea]. Available: <http://restsharp.org/>. [Último acceso: 10 2013].
- [15] JSON.org, «Introducción a JSON,» [En línea]. Available: <http://www.json.org/json-es.html>. [Último acceso: 10 2013].
- [16] ECMA International, «Standad ECMA-404 - The JSON Data Interchange Format,» 10 2013. [En línea]. Available: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>. [Último acceso: 10 2013].
- [17] W3C, «Extensible Markup Language (XML),» [En línea]. Available: <http://www.w3.org/XML/>. [Último acceso: 10 2013].
- [18] WikiMedia, «WikiPedia - C Sharp (programming language),» [En línea]. Available: [http://en.wikipedia.org/wiki/C\\_Sharp\\_%28programming\\_language%29](http://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29). [Último acceso: 12 2013].
- [19] Microsoft Corporation, «Clases genéricas (Guía de programación de C#),» [En línea]. Available: <http://msdn.microsoft.com/es-es/library/sz6zd40f%28v=vs.90%29.aspx>. [Último acceso: 12 2013].
- [20] Microsoft Corporation, «Dev Center - Multilingual App Toolkit,» [En línea]. Available: <http://msdn.microsoft.com/en-us/windows/apps/bg127574.aspx>. [Último acceso: 10 2013].

# 8 Anexos

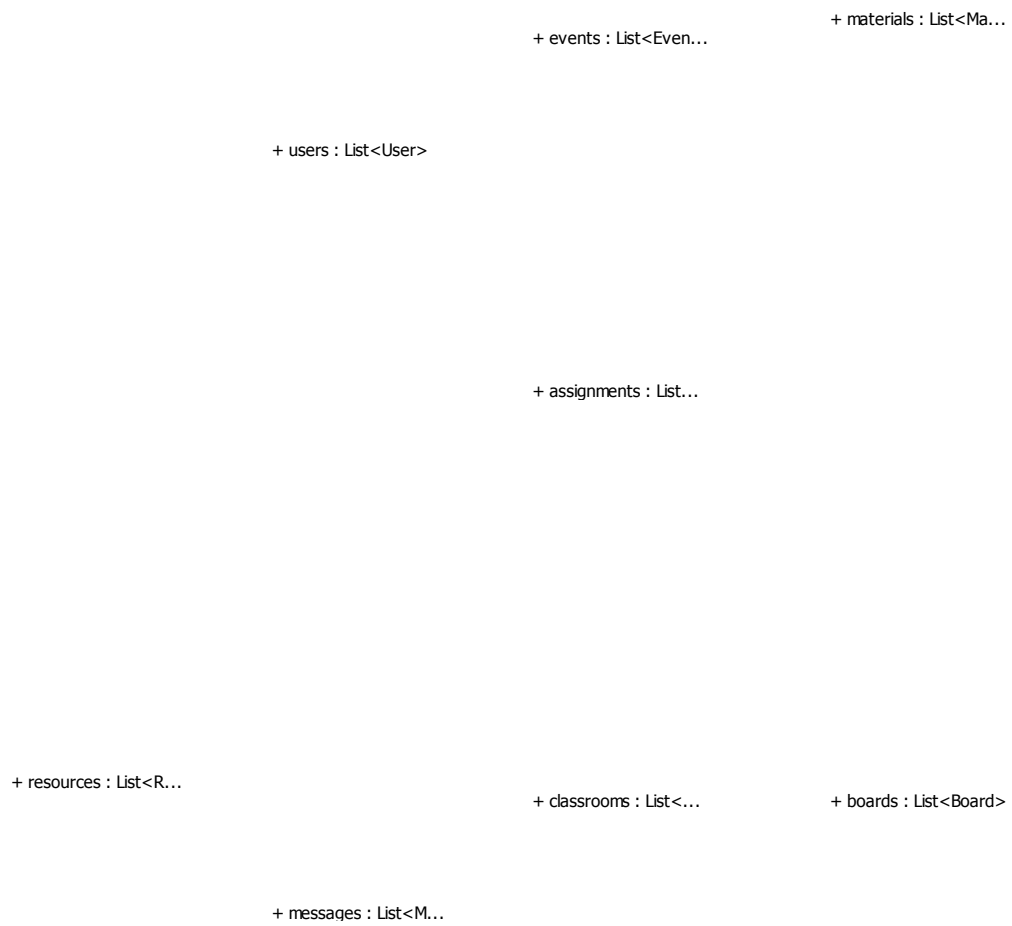
## 8.1 Diagrama de clases. (Ampliado)





## 8.2 Diagrama de entidades de la capa de datos

25



### 8.3 Manual de usuario de instalación

La aplicación es compatible con sistemas operativos Windows Phone 7.1 en adelante. La metodología de instalación comprende dos modalidades. Aunque una de ellas puede no estar disponible hasta la publicación de la aplicación en la tienda “Market” oficial de Windows Phone.

#### 8.3.1 Instalación manual

Para la instalación manual necesitaremos el fichero .XAP, nuestro caso CampusUOC.XAP, que es el paquete resultado de la compilación y compactación de la aplicación en un solo fichero de instalación. Este fichero puede ser copiado como cualquier otro fichero en la tarjeta de memoria SD del dispositivo electrónico o transferido con cualquier otra tecnología que nos permita tenerlo guardado en nuestro dispositivo de manera local.

Para realizar la instalación manual debe de seguir los siguientes pasos:

1. Descargar o transferir el archivo .XAP a la memoria interna del dispositivo.
2. En el botón Inicio, pulse en Tienda y luego pulse en Tarjeta SD.
  - a. Nota:  
Si acaba de insertar la tarjeta SD o agregar los archivos .XAP, es posible que tenga que esperar algunos minutos antes de que pueda acceder a la tarjeta SD desde la Tienda.
3. Selecciona las aplicaciones que quieras y pulsa Instalar.
  - a. Notas  
Las aplicaciones que se pueden instalar aparecen en “Aplicaciones compatibles”.  
Deberás tener la versión más reciente del archivo .XAP de la Tienda en la tarjeta SD antes de poder instalar la aplicación. Si no es la versión más reciente, aparecerá en “Aplicaciones incompatibles”.
4. Las aplicaciones instaladas aparecen en la lista de aplicaciones y los juegos aparecen en el hub de juegos. En función de la aplicación o el juego específico, podrás utilizar el elemento de esta manera:
  - a. Las aplicaciones y juegos gratuitos pueden utilizarse de forma inmediata.
  - b. Las aplicaciones y juegos de pago que tengan un período de prueba estarán disponibles como versión de prueba que podrás comprar más adelante.
  - c. Las aplicaciones de pago que no tengan un período de prueba deberán comprarse antes de poder utilizarlas.

### **8.3.2 Instalación a través de la tienda (puede que no esté disponible todavía)**

En la Tienda de Windows Phone, la tienda de entretenimiento y aplicaciones virtual de Microsoft para el teléfono, existen miles de aplicaciones y juegos. Entre ellas, en el caso de que sea publicada, puedes encontrar Campus UOC.

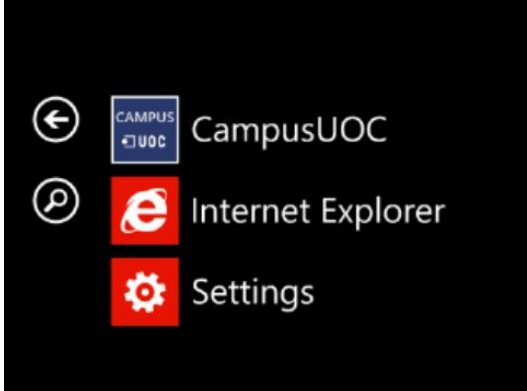
Para encontrar la aplicación pruebe a buscar la palabra “UOC” entre las aplicaciones. Y después de pulsar sobre ella podrás ver comentarios e información ampliada, ahora sólo pulsa en Instalar para descargarlo.

A fecha de la entrega de este proyecto, la aplicación se encuentra en estado de revisión por los responsables del “Market” de Microsoft para Windows Phone. Por lo que si pasa correctamente la revisión este producto será publicado en breve.

### 8.4 Manual de uso de la aplicación

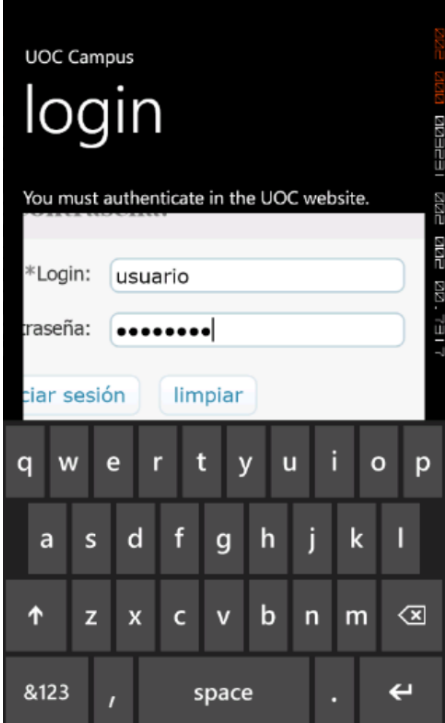
Aunque el uso de la herramienta es sencillo e intuitivo, ya que su manejo es muy similar al de aplicaciones similares en este apartado realizaremos un pequeño manual de uso.

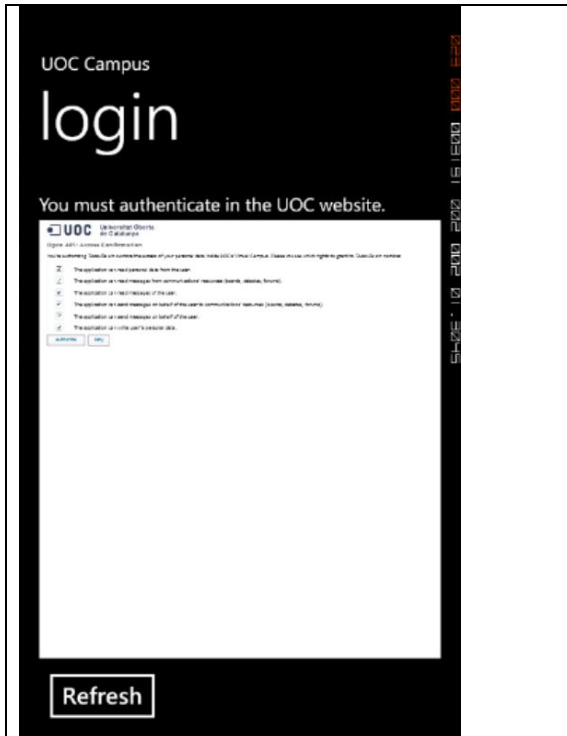
#### 8.4.1 Abrir aplicación

	<p>La aplicación una vez instalada, véase punto 8.3 de este documento, creará un acceso directo desde el menú de aplicaciones del dispositivo. Sólo necesitará ejecutar el icono asociado a la aplicación "CampusUOC".</p>
---	--

#### 8.4.2 Acceder al sistema

Los pasos para acceder son muy similares a los de la página web de la UOC añadiendo un paso para otorgar permisos.

	<p>Una vez iniciada la aplicación sólo habrá que introducir el usuario y contraseña que tenemos para todos los servicios de la UOC.</p> <p>Es posible que los casilleros de login se vean algo reducidos, pero esto es debido al paso intermedio de autenticación a través de navegador necesario para acceder al sistema.</p>
---	--

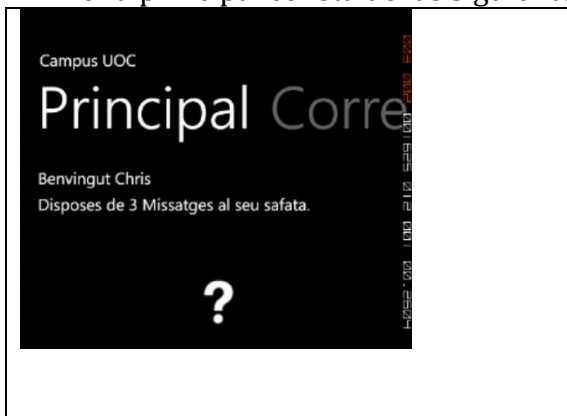
	<p>En el siguiente paso se nos preguntará por los derechos que queremos otorgar a la aplicación.</p> <p>Por defecto elegiremos todos y pulsaremos “Aceptar”.</p> <p>Como se puede ver en la parte baja, disponemos de un botón “Refresh” o actualizar para volver a cargar la página de autenticación en caso de que fuera necesario.</p>
---	---

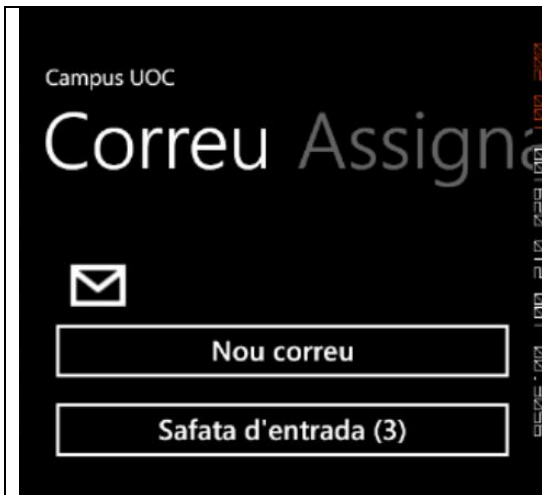
**8.4.3 Menú principal. Navegación**

La página principal de la aplicación consta de una página pivotada, esto significa que podemos movernos a través de las diferentes partes de la aplicación mediante un movimiento de arrastre de nuestro dedo en el sentido que queramos mover la página. De un modo similar al que haríamos cuando queremos pasar la página de un libro.

En el caso de entrar en una página no pivotada, que suele tratarse de una funcionalidad concreta de la aplicación, sólo tendremos que pulsar el botón del teléfono ← que representa la acción “Volver”. El uso de este botón es muy común en otras aplicaciones de este dispositivo.

El menú principal consta de las siguientes páginas:

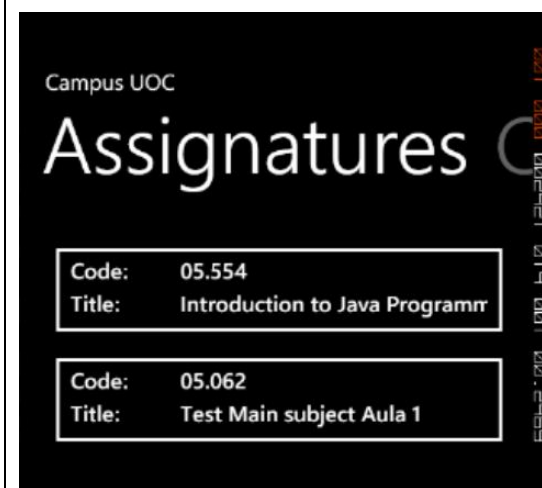
	<p>La primera página principal de entrada es un resumen. Consta de un saludo, un pequeño texto informativo de los mensajes que quedan por leer y la foto del usuario autenticado, en caso de que la tenga.</p> <p>Tal y como hemos explicado antes, deslizando el dedo por encima de la aplicación de izquierda a derecha pasaremos a la siguiente página.</p>
---	--



La siguiente página es la del correo. En ella tenemos dos botones, “Nuevo correo” o “Bandeja de entrada”.

El botón de nuevo correo nos llevará al diálogo de composición de correo que veremos más adelante.

El botón de bandeja de entrada nos llevará al diálogo de resumen de correos del usuario.

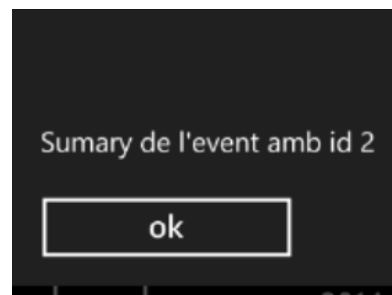



La siguiente página es la de las asignaturas. En ella nos aparecerá un listado de asignaturas, siendo cada una de ellas un botón que nos sirve para entrar al diálogo de asignatura que nos detalla más a fondo cada una de ellas.



La siguiente página representa un calendario donde se marcan con colores los días que disponen de algún evento.



También se coloreará el día actual. Al pulsar con el dedo sobre un día con evento nos mostrará información sobre el mismo.



	<p>La última página es la de configuración, donde podremos escoger el idioma de la interfaz de nuestra aplicación.</p> <p>Por defecto, el idioma seleccionado será el del sistema. Una vez cambiado el idioma éste se guardará para la próxima vez que carguemos la aplicación.</p>
---	---

#### 8.4.4 Correo


Las acciones principales que podemos realizar sobre el correo son dos, la primera de ellas es la de componer un correo y enviarlo. Y la segunda la de listarlos y visualizarlos. Se acceden a ambas opciones desde la página de correo vista anteriormente.

	<p>Para componer un correo sólo debemos añadir un correo válido, y, al menos un asunto o un contenido en el cuerpo de correo.</p> <p>Una vez pulsemos “Enviar”, y si el correo se valida correctamente, el sistema intentará enviar el correo y nos dará respuesta del resultado.</p> <div data-bbox="903 1279 1230 1496" data-label="Image">  </div>
--	---

	<p>La otra opción que tenemos sobre los correos es la de listarlos y leerlos.</p> <p>Los mensajes se listan en modo de botón con un pequeño resumen sobre el mismo. Si pulsamos sobre el correo podremos leer su contenido completo.</p> 
---	---

#### 8.4.5 Asignaturas

Desde el menú principal podemos visualizar las asignaturas relacionadas con el usuario que ha iniciado la aplicación. Este listado muestra como un botón cada una de las asignaturas. Al pulsar cualquiera de ellas llegaríamos a los detalles de la asignatura tal y como vemos en este apartado.

	<p>Al pulsar sobre una asignatura podemos ver los detalles básicos de la misma.</p> <p>Para visualizar los detalles disponemos de una página pivotada, al igual que el menú principal, pero cargada como parte de la página de la asignatura. Esta página muestra los siguientes listados relacionados con la asignatura:</p> <ul style="list-style-type: none"> <li>• Tablones</li> <li>• Asignaciones</li> <li>• Gente</li> <li>• Materiales</li> <li>• Recursos</li> </ul> <p>Dependiendo del tipo de elemento listado podremos realizar diferentes acciones sobre ellos. En el caso de asignaciones, gente o recursos no tenemos acciones.</p>
---	--



	<p>En el caso de materiales, al pulsar con el dedo sobre el recurso, nos dará opción a abrir el enlace asociado al recurso.</p> <p>Nota: será necesario tener instalado una aplicación compatible en el dispositivo con el tipo de fichero asociado al recurso para poder abrirlo.</p>
	<p>En el caso de tableros, se nos abrirá automáticamente la vista preparada para listarlos, y de manera similar a la bandeja de entrada podremos pulsar sobre cualquier mensaje del tablón para poder visualizarlo.</p> 

Como hemos indicado anteriormente, para volver desde cualquiera de los submenús de la aplicación sólo deberemos ir pulsando el botón ← del dispositivo el mismo número de veces que acciones de apertura de submenús hemos haciendo.