Universidad Zaragoza
1542

Trabajo Fin de Grado

CampusTeruel: Aplicación de ayuda para los alumnos del Campus de Teruel

Autor

Roberto Gimeno Callejas

Directores

Francisco José Martínez Domínguez
Jesús Gallardo Casero

Escuela Universitaria Politécnica

2015

Resumen

En el entorno universitario existe una gran cantidad de información que los alumnos necesitan prácticamente a diario: información sobre los horarios de clases o exámenes de la titulación, horarios de servicios como la biblioteca o reprografía o información de contacto como el correo electrónico de un profesor. Actualmente la información descrita está colgada en diferentes puntos de la red, que para el alumno son difíciles de encontrar o directamente decide no intentar buscar por el coste que le pueda suponer.

Actualmente, con el auge de las tecnologías móviles, prácticamente todo el mundo posee un teléfono inteligente. El presente proyecto plantea la posibilidad de solucionar la situación anterior, en la medida de lo posible, con el desarrollo de una aplicación que permita el acceso a toda la información relacionada con el campus de Teruel de la Universidad de Zaragoza desde la palma de la mano de una forma fácil y sencilla.

El sistema está compuesto por dos aplicaciones diferenciadas, pero relacionadas entre sí: una aplicación del lado del cliente para teléfonos inteligentes, y una aplicación web para la administración de los datos que se mostrarán en el cliente. El propósito fundamental es permitir a los alumnos tener acceso de forma inmediata, y en cualquier momento, al conjunto de información relacionada con el campus.

El sistema implementado cumple con todos los objetivos marcados inicialmente, con un correcto funcionamiento de la aplicación, y el consiguiente ahorro de tiempo y comodidad del usuario a la hora de buscar la información necesaria.

Palabras clave: Android, centros, titulaciones, asignaturas, servicios, profesorado.

Contenido

Índice de ilustraciones

Índice de tablas

1. Introducción

En el presente capítulo se presentarán tanto la motivación como los objetivos del proyecto a realizar. Además, se presentarán brevemente los diferentes apartados del presente documento, y los agradecimientos del autor.

1.1. Motivación

La entrada en el mundo universitario puede suponer un cambio significativo para un nuevo alumno, donde se tiene una libertad más amplia comparándola con la educación secundaria o bachillerato. Esto puede hacer que el alumno encuentre muchas dificultades a la hora de querer encontrar un servicio de la universidad como puede ser la secretaría, ya que es algo a lo que no está acostumbrado.

Por otro lado, se puede dar el caso en que el alumno necesite un acceso rápido a la información sobre horarios, correos electrónicos o teléfonos, ya sea de un servicio, de las clases o para ponerse en contacto con un profesor.

Actualmente existe el acceso a este tipo de información desde las páginas web de los centros del campus, pero el acceder a estas suele ser algo que al usuario normal, por lo general, le supone un coste de tiempo y más si a ésta no se accede directamente desde la página principal y debe realizar una búsqueda en la web.

El principal motivo para realizar este proyecto es solucionar todo esto en la medida de lo posible, implementando un sistema que permita a los alumnos nuevos moverse de una forma más sencilla en un entorno nuevo, y que cualquier alumno pueda tener un acceso fácil y rápido a información que necesita prácticamente a diario.

Actualmente cualquier persona dispone ya de un dispositivo inteligente; esto unido al auge de estas tecnologías ha favorecido que muchas universidades ya tengan su propia aplicación que permita acceder a sus servicios. Sin embargo, el campus de Teruel de la Universidad de Zaragoza todavía no ofrece este tipo de servicio al alumnado. Aquí es donde comienza este proyecto que intentará facilitar la vida universitaria en el campus.

Finalmente, dado que es el último trabajo de la titulación, existe una gran disposición en familiarizarme con una tecnología de máxima actualidad como la programación Android, pues es un sistema que está creciendo día a día debido al aumento de usuarios a nivel mundial.

1.2. Objetivos

Los objetivos que se pretenden conseguir durante la realización de este son los siguientes:

1. Explorar las posibilidades de desarrollar una aplicación móvil que sirva de ayuda a los alumnos del campus de Teruel de la Universidad de Zaragoza.
2. Realizar un estudio de las herramientas necesarias para el desarrollo del proyecto, para una mejor comprensión y utilización de las mismas.
3. Revisar el estado del arte actual de las aplicaciones de una temática similar a la que se presenta en este proyecto, mostrando las principales soluciones adoptadas.
4. Realizar un análisis y diseño exhaustivo de la estructura que deberá seguir el sistema para desarrollar una aplicación Android que sea capaz de mostrar la información nombrada en los siguientes puntos:
    - Acceso a la actualidad del campus.
    - Acceso al calendario académico.
    - Acceso rápido al horario del alumno.
    - Información de los centros como teléfonos o dirección, titulaciones impartidas y horarios.
    - El profesorado con información como las tutorías o correo.
    - Información como la ubicación, horarios o teléfonos de servicios como Universa o la biblioteca.
    - Plano con los centros del campus.
5. Indicar también el análisis y diseño para el desarrollo de una interfaz web que permita mantener toda la información nombrada lo más actualizada posible.
6. Realizar la implementación del sistema descrito en el documento, con una correcta sincronización entre el cliente Android y el servidor que almacena los datos que muestra la interfaz web de administración.
7. Realizar una encuesta con alumnos del campus de Teruel de la Universidad de Zaragoza, permitiéndoles probar la aplicación, para identificar posibles mejoras, errores y la acogida que podría tener este proyecto en el alumnado.

1.3. Visión general del documento

El presente documento consta de los apartados a continuación:

- **Conceptos previos:** Apartado en el que se contextualiza la aplicación y se exponen conceptos básicos, necesarios para entender el funcionamiento de la misma.
- **Estado del arte:** Se exponen diferentes aplicaciones presentes en el mercado que ofrecen el mismo servicio o similares al trabajo fin de carrera que estamos tratando.
- **Análisis:** Contiene todas las condiciones que debe satisfacer la implementación del sistema, basándose en las directrices que se marcan en el estándar del IEEE *Recommended Practice for Software Requirements Specifications* IEEE 830-1998.

- **Planificación y estimación de costes:** Aquí se encuentra tanto la planificación prevista como la planificación real del proyecto, además de una estimación de los costes que se ocasionarán durante el desarrollo del proyecto.
- **Diseño:** Sección en la que se expone y explica detalladamente toda la fase de diseño de la aplicación.
- **Validación:** Contiene los resultados obtenidos en una encuesta realizada por alumnos del campus que han probado la aplicación.
- **Conclusiones:** Apartado para la reflexión y el análisis sobre la realización del proyecto y los resultados obtenidos.
- **Ampliaciones futuras:** Se especifican las diferentes mejoras que en un futuro se pueden realizar sobre este proyecto.
- **Bibliografía:** Se incluyen las referencias bibliográficas utilizadas a lo largo de la redacción del presente proyecto.
- **Anexos:** Incluye información complementaria sobre los diferentes apartados del proyecto que no han tenido cabida en el documento principal.

1.4. Agradecimientos

En primer lugar agradecer a mi familia el apoyo constante recibido, no solo durante la realización del proyecto, sino durante toda la carrera, y en especial a mi hermana Mirian que me ha acompañado durante buena parte de mi vida universitaria en Teruel.

Agradecer también a mis directores de proyecto Francisco J. Martínez y Jesús Gallardo la ayuda recibida y el haber estado siempre disponibles para resolver cualquier duda.

Y por último, agradecer a toda lo comunidad de desarrolladores que hay detrás de Android por la cantidad de tutoriales y foros de ayuda que existen, ya que, me han permitido avanzar en momentos de dificultad en el desarrollo del proyecto.

2. Conceptos previos

Para el uso de cualquier tecnología es fundamental conocer dicha tecnología, así como la historia y el futuro de la misma para así poder optimizar y resolver problemas de manera más simple. Esta sección pretende explicar los conceptos importantes de las partes que forman el proyecto. En concreto, se presentarán las características del sistema operativo Android, entorno sobre el cual gira el proyecto, se expondrán los servicios *Google Cloud Messaging* o *Google Maps* utilizados en la aplicación, se presentará la herramienta *Automatic Api Rest* para la obtención de datos desde el dispositivo, y por último se expondrán unos conocimientos básicos de PHP y MySQL, que han sido utilizados para el desarrollo de la interfaz web de administración.

2.1. Android

Android [1] es un sistema operativo basado en el núcleo Linux, diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o *tablets*. También está siendo implementado en otros sistemas como relojes inteligentes, televisores y automóviles.

Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró. Android fue presentado en 2007 junto la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles.

Android es multitarea y permite a los desarrolladores acceder a las funcionalidades principales del dispositivo mediante aplicaciones, que pueden ser reemplazadas por otras oficiales o de terceros desarrolladas a través de las herramientas proporcionadas por Google. El sistema consta de 12 millones de líneas de código escritas en XML, C, C++ y Java. La mayoría de dicho código se libera bajo la licencia Apache, lo que permite a la comunidad de desarrolladores realizar cambios y mejoras en el sistema operativo.

2.1.1. Arquitectura

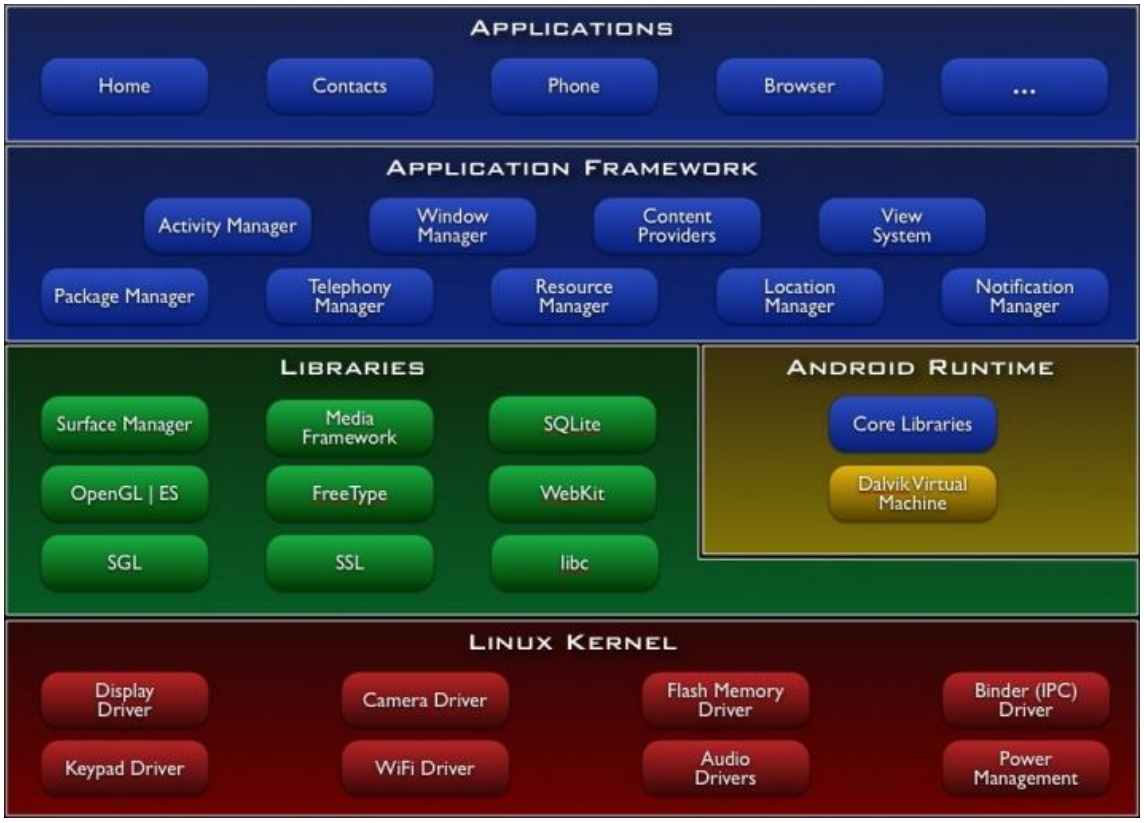


Ilustración 1. Arquitectura Android [2]

La Ilustración 1 muestra cómo está estructurado el sistema operativo Android. Como se puede observar, los diferentes componentes están agrupados por capas. Cada una de las capas utiliza elementos de la capa inferior para realizar sus funciones. Es por ello, que a este tipo de arquitectura se le conoce también como pila. A continuación, presentaremos cada una de las capas de Android.

Kernel

El núcleo del sistema operativo Android está basado en un *kernel* de Linux, similar al que puede incluir cualquier distribución de Linux, como Ubuntu, solo que adaptado a las características del hardware en el que se ejecutará Android, es decir, para dispositivos móviles.

Dicho núcleo actúa como una capa de abstracción entre el hardware y el resto de las capas de la arquitectura, lo que permite acceder a los componentes del sistema sin necesidad de conocer las características precisas de cada teléfono.

Librerías

Se sitúan justo sobre el *kernel*, están escritas en C o C++ y son compiladas para el hardware específico del dispositivo. Su objetivo principal es proporcionar funcionalidad a las aplicaciones para las tareas que se repiten con frecuencia, permitiendo que se ejecuten con mayor eficiencia.

Entorno de ejecución

Como podemos apreciar en la Ilustración 1, el entorno de ejecución de Android no se considera una capa en sí mismo, dado que también está formado por librerías. El componente principal del entorno de ejecución de Android es la máquina virtual Dalvik, que es la encargada de ejecutar todas las instrucciones no nativas de Android.

Desde la versión 4.4 Kitkat se introduce la nueva máquina virtual ART permitiendo al usuario elegir entre ambas, siendo la principal diferencia el tipo de compilación. Dalvik utiliza una compilación llamada JIT (just-in-time), esto quiere decir que cada vez que iniciamos una aplicación, la máquina virtual dinámicamente traduce el código a código máquina durante la ejecución de ésta. En el caso de ART, utiliza AOT (ahead-of-time), las aplicaciones se compilan desde que se instalan en el dispositivo, y ya se deja instalado el código máquina listo para ejecutarse y sin necesidad de otra compilación [3].

Framework de aplicaciones

La siguiente capa está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas anteriores a través de la máquina virtual.

Aplicaciones

Es la última capa; en ella se encuentran instaladas todas las aplicaciones del dispositivo, las nativas (programadas en C o C++) y las instaladas (normalmente programadas en Java).

2.1.2. Fragmentación

Dentro de Android existen numerosas versiones y cada una tiene un API distinta, para la cual se han podido desarrollar más librerías, actualizar otras o descartar librerías que están en uso en versiones anteriores. Esto ocasiona el problema de la fragmentación y por ello, cuando se desarrolla una aplicación, no tiene por qué funcionar en todas las versiones. Por esta razón cobra vital importancia la elección de la API con la que se va a desarrollar la aplicación.

En la Tabla 1 se muestran las versiones que están en uso y el porcentaje de dispositivos que lo utilizan a nivel mundial. Estos datos están actualizados a marzo de 2015, recogidos en la página web de Android Developers [4].

Versión	Nombre	API	Distribución
2.2	Froyo	8	0.4%
2.3.3 – 2.3.7	Gingerbread	10	6.9%
4.0.3 – 4.0.4	Ice Cream Sandwich	15	5.9%
4.1.x	JellyBean	16	17.3%
4.2.x		17	19.4%
4.3		18	5.9%
4.4	KitKat	19	40.9%
5.0	Lollipop	21	3.3%

Tabla 1. Distribución según la versión

Como se puede observar, más del 90% de los dispositivos Android utilizan la versión 4.0.3 Ice Cream Sandwich o superior.

2.1.3. Mercado

A medida que los teléfonos móviles crecen en popularidad, los sistemas operativos con los que funcionan adquieren mayor importancia. Dentro de éstos el más destacado es Android, como se puede ver en la Tabla 2.

Sistema Operativo	Volumen de unidades 2014	Cuota de mercado 2014	Volumen de unidades 2013	Cuota de mercado 2013	Crecimiento
Android	1.059,3	81,5%	802,2	78,7%	32,0%
iOS	192,7	14,8%	153,4	15,1%	25,6%
Windows Phone	34,9	2,7%	33,5	3,3%	4,2%
BlackBerry	5,8	0,4%	19,2	1,9%	-69,8%
Otros	7,7	0,6%	2,3	0,2%	234,8%
Total	1.300,4	100,0%	1.018,7	100,0%	27,7%

Tabla 2. Distribución de los SO móviles en el mercado [5]

Los datos mostrados en la anterior tabla corresponden al año 2014, siendo Android el Sistema Operativo más usado con un 81.5% frente al 14.8% del sistema de Apple iOS.

En la Ilustración 2 se puede ver el crecimiento ascendente que ha tenido Android desde 2010.

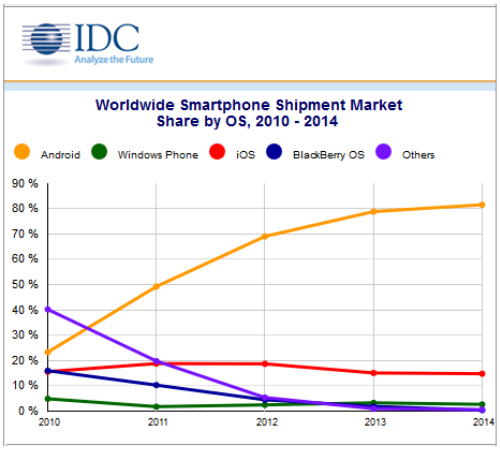


Ilustración 2. Cuota de mercado de los SO (2010-2014) [5]

2.2. Servicios de Google

Google ofrece una gran variedad de servicios que se pueden utilizar entre otras cosas en el desarrollo de aplicaciones Android. Durante el desarrollo del proyecto se utilizarán los servicios detallados en este apartado.

2.2.1. Google Cloud Messaging (GCM)

Es un servicio que permite a los desarrolladores enviar datos desde los servidores a las aplicaciones Android o del navegador Chrome [6] [7].

Ofrece un mecanismo sencillo y ligero que los servidores pueden utilizar para contactar con las aplicaciones móviles para tareas como actualizar datos de la aplicación. El servicio se encarga de todos los aspectos de gestión de colas de mensajes y la entrega de éstos a la aplicación destino.

GCM es un servicio gratuito que tiene la capacidad de enviar un mensaje ligero informando a la aplicación Android de nuevos datos en el servidor. También puede enviar mensajes más grandes con hasta 4KB de datos útiles. Además de un tamaño máximo en los mensajes, Google limita también el número de mensajes que un remitente envía en conjunto, y el número de mensajes que un remitente envía a un dispositivo específico.

La aplicación no tiene por qué estar en ejecución para la recepción de los mensajes. El sistema despertará la aplicación a través de un mecanismo llamado Broadcast cuando llegue el mensaje, siempre y cuando la aplicación tenga definido el sistema Broadcast apropiado y los permisos adecuados.

2.2.2. Google Maps

Es un servidor de aplicaciones de mapas en la web [8]. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle con Google Street View.

Google Maps fue anunciado por primera vez en Google Blog el 8 de febrero de 2005. Originalmente soportaría solo a los usuarios de Internet Explorer y Mozilla Firefox, pero el soporte para Opera y Safari fue agregado el 25 de febrero de 2005. El software estuvo en su fase beta durante seis meses, antes de convertirse en parte de Google Local, el 6 de octubre de 2005.

Permite aumentar y reducir la escala del mapa para ver en mayor o menor detalle el mismo, buscar servicios a través del buscador y posicionarlos en el mapa, así como obtener las coordenadas de una posición, entre otras muchas características.

2.3. Automatic Api Rest

Api Rest [9] es una herramienta de código abierto que permite conectar los datos de una base de datos de un servidor con un servicio externo como un sistema Android o iOS. Esto facilita al programador el acceso sencillo a los datos del servidor desde cualquier dispositivo.

Rest [10] (Representational State Transfer) es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP. Permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP.

La herramienta Automatic Api Rest se basa en este tipo de servicio Rest al que le hacen peticiones para recibir los datos, pero únicamente tiene que ser configurado para darle acceso a la base de datos a la que se desea conectar, no es necesario nada más para tener el servicio en funcionamiento.

2.4. PHP

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico [11]. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. Puede ser usado en la mayoría de los servidores web, al igual que en casi todos los sistemas operativos y plataformas sin ningún coste.

Se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy, lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico como Facebook, para optar por PHP como tecnología de servidor.

Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP y forma parte del software libre publicado bajo la licencia PHP.

2.5. MySQL

MySQL [12] es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB, desde enero de 2008, una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009, desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

MySQL es usado por muchos sitios web grandes y populares, como Wikipedia, Google (aunque no para búsquedas), Facebook, Twitter, Flickr y YouTube.

3. Estado del arte

En lo referente al estado del arte se han encontrado las aplicaciones que se comentarán a continuación, las cuales tienen similitudes con la aplicación del proyecto.

Las universidades, gracias al auge de las tecnologías y dispositivos móviles, comienzan a hacer uso de éstas para facilitar a los alumnos toda la información que tienen disponible sobre los servicios que ofertan en sus instalaciones. En el siguiente apartado se mostrarán algunas de las aplicaciones de este tipo que existen actualmente.

3.1. Aplicaciones similares disponibles en el mercado

3.1.1. EINA 2015

Es la aplicación Android de la escuela de ingeniería y arquitectura (EINA) de la Universidad de Zaragoza [13]. Está disponible para terminales con Android 1.5 o superior.

La aplicación consiste en una serie de botones que funcionan como enlaces, desde los cuales se puede acceder rápidamente a:

- La web de la EINA.
- Las plataformas del ADD.
- Al email de UNIZAR.
- Horarios y calendarios del curso académico 2014-2015.

Ilustración 3. Aplicación Eina 2015

3.1.2. UPV

Esta aplicación ha sido desarrollada por alumnos, titulados y técnicos de la Universidad Politécnica de Valencia (UPV) [14]. Está disponible para Android, iOS y Windows Phone, y permite tener acceso a una gran cantidad de información relacionada con la universidad:

- Integra todas las agendas disponibles en la UPV: exámenes, clases, actividades deportivas, eventos culturales y científicos…

- Acceso a la plataforma donde profesores y alumnos comparten toda la información sobre las asignaturas.
- Permite consultar una gran parte de la intranet: datos personales, información reservada, pagos pendientes, herramientas, solicitudes y notificaciones…
- Posiciona las escuelas y facultades, así como el resto de entidades de la UPV sobre los mapas de los tres campus: campus de Vera, de Gandía y de Alcoy.
- Información sobre la próxima llegada de los autobuses urbanos a las paradas del campus y la cantidad de bicicletas y bornetas que hay disponibles en las estaciones de Valenbisi.
- Tiene acceso a todas las noticias publicadas en la portada de la página principal de la UPV.
- Puede reproducir el informativo diario de la televisión de la TV.

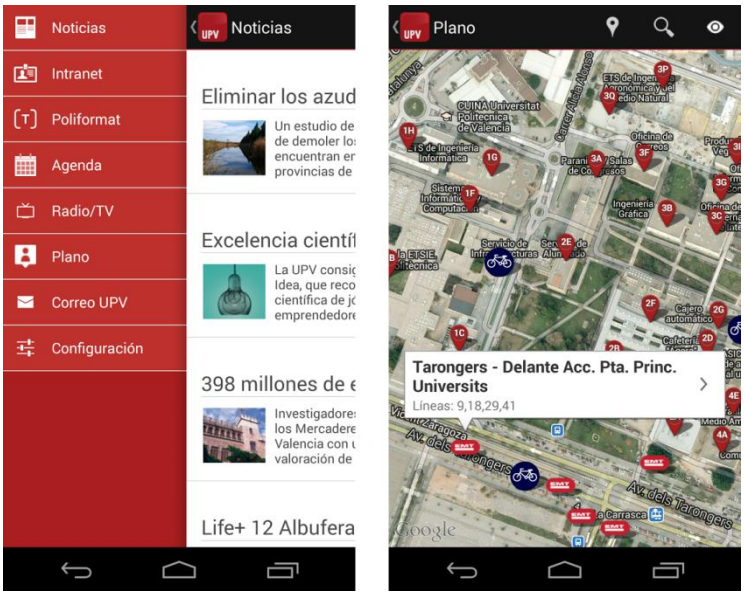


Ilustración 4. Aplicación UPV

3.1.3. UGR

Aplicación desarrollada por la Universidad de Granada (UGR) [15] para Android e iOS. Provee información útil sobre la universidad, sus órganos de gobierno actuales, los centros que la componen y las titulaciones que en ella se imparten.

Dispone de información para los nuevos estudiantes como el alojamiento en Granada, el calendario académico, el uso de la tarjeta universitaria o el proceso para solicitar una cuenta de correo electrónico y además da un acceso rápido a las redes sociales de la universidad. También incluye acceso al directorio de la universidad, con el que podrás localizar la información sobre los miembros de esta comunidad, permite localizar geográficamente los centros y servicios de la universidad e incluye un apartado de noticias.

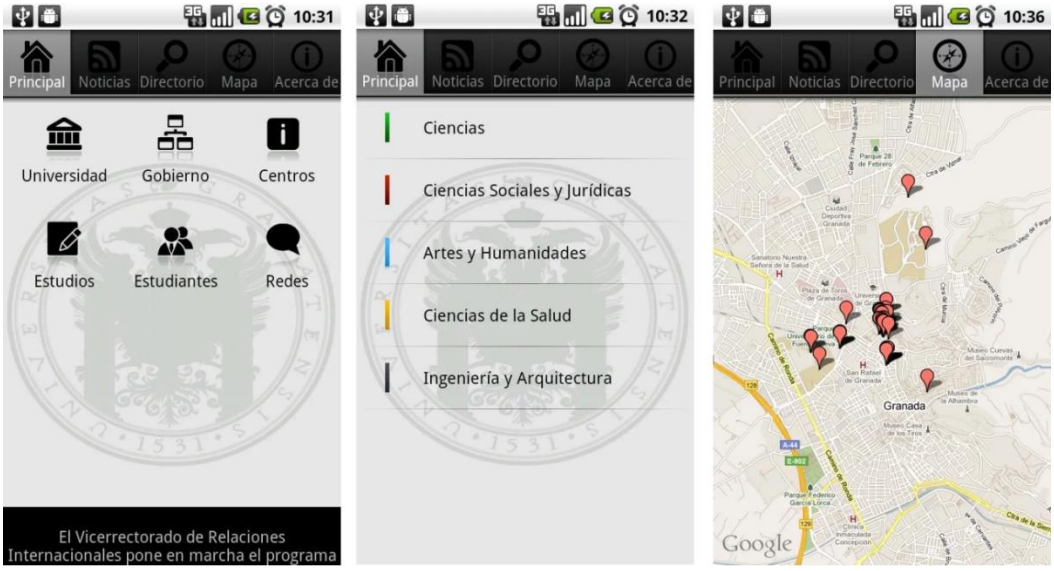


Ilustración 5. Aplicación UGR

3.1.4. UCMovil

Es la aplicación desarrollada por la Universidad Complutense de Madrid (UCM) [16], está desarrollada tanto para Android como para iOS. Da acceso a la información de interés de forma cómoda. Algunas de sus funcionalidades son:

- Información general sobre la universidad y las titulaciones.
- Información de localización y contacto de todos los centros y departamentos.
- Directorio de profesores y personal.
- Últimas noticias de la universidad.
- Catálogo, información de contacto y horarios de las bibliotecas.
- Acceso a las redes sociales de la universidad.

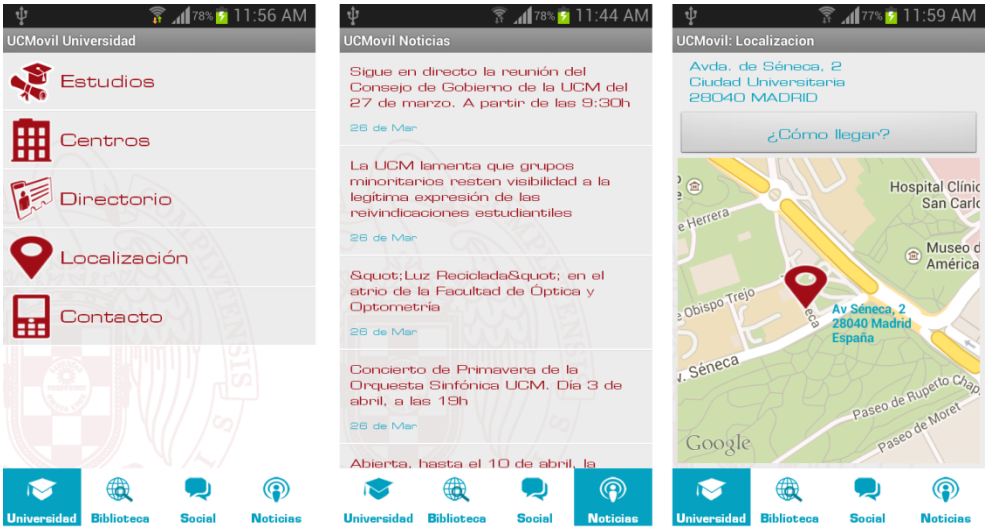


Ilustración 6. Aplicación UCMovil

3.1.5. Universidad de Murcia App

Aplicación que facilita la información de la Universidad de Murcia [17], está disponible tanto para Android como para iOS.

Además de permitir el acceso a la información básica y localización de la universidad, permite realizar llamadas y mensajes entre los usuarios. De esta forma permite una fácil y rápida comunicación entre los miembros de la universidad.

Informa también sobre los eventos que se producen y el lugar de celebración, además de incluir acceso al canal de noticias.

Ilustración 7. Aplicación de la Universidad de Murcia

3.1.6. UviGO!

Esta App da acceso a la información relacionada con la Universidad de Vigo [18], disponible para Android e iOS.

Permite el acceso a la siguiente información:

- Información institucional como el saludo del rector y la organización.
- Información de contacto, como el teléfono o el correo, de cada facultad y cada campus.
- Acceso a la información de servicios como las bibliotecas, centrándose en la información de contacto.
- Cómo llegar a cada uno de los campus y las distintas posibilidades de hacerlo a través del transporte público.
- Estudios que ofrecen en los diferentes campus.

También contiene una sección que permite al alumno ingresar en el sistema y acceder a su expediente académico. Por último, permite definir diferentes avisos sobre las actividades que se realizan en el campus.

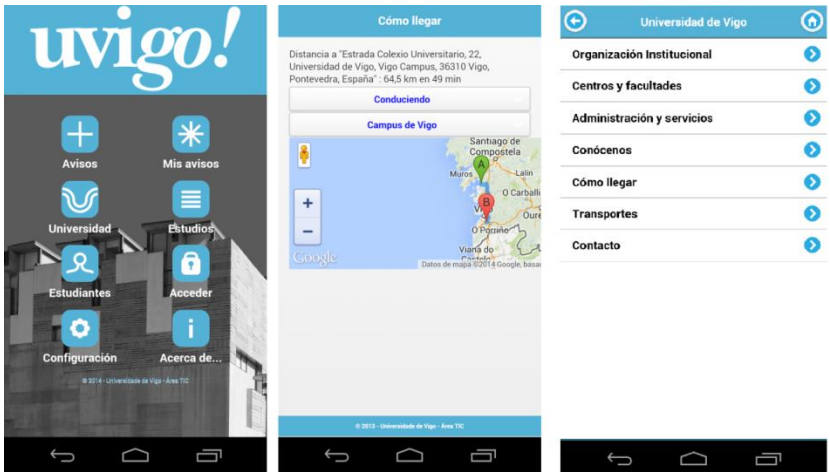


Ilustración 8. Aplicación UviGO!

3.1.7. UniOviApp

Se trata de la aplicación desarrollada por la Universidad de Oviedo [19]. Disponible para Android, permite un acceso rápido a muchos servicios y datos de interés de la universidad.

Contiene las siguientes funcionalidades:

- Búsquedas en el directorio de profesores.
- Incorporación del calendario académico al calendario del dispositivo.
- Posicionamiento en el mapa de las principales sedes universitarias.
- Enlace a los canales oficiales en redes sociales.
- Entrada al Campus Virtual y enlace al correo corporativo.
- Datos principales del expediente académico (asignaturas, convocatorias, notas...).
- Consulta de notas de la PAU.
- Agenda de eventos, noticias de la Universidad y dUO.

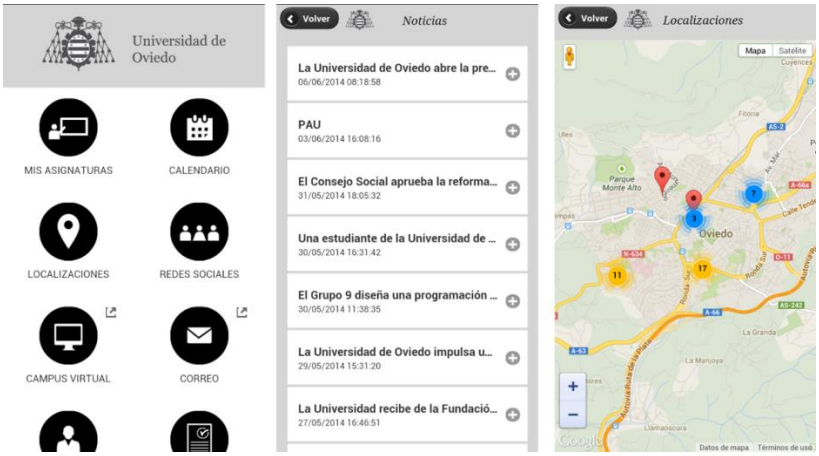


Ilustración 9. UniOvi App

3.2. Comparativa

Tras haber expuesto las aplicaciones más relevantes desarrolladas por algunas de las universidades de España, en este apartado se presenta una comparativa de dichas aplicaciones, comparando sus principales características.

Característica	Eina 2015	UPV	UGR	UCMovil	Universidad de Murcia App	UviGO!	UniOvi App
Dispositivo	Android	Android, iOS, Windows pone	Android e IOS	Android e iOS	Android e iOS	Android e iOS	Android
Noticias	No	Sí	Sí	Sí	Sí	No	Sí
Info. Centros	Sí	Sí	Sí	Sí	No	Sí	No
Info. Titulaciones	Sí	Sí	Sí	Sí	No	Sí	No
Info. Servicios	No	Sí	Sí	Sí	No	Sí	No
Info. Profesorado	No	No	Sí	Sí	No	No	Sí
Localización	No	Sí	Sí	Sí	Sí	Sí	Sí
Acceso a Redes sociales	Sí	No	Sí	Sí	Sí	No	Sí
Acceso al ADD	Sí	Sí	No	No	No	Sí	Sí
Contenido Multimedia	No	Sí	Sí	No	No	No	No

Tabla 3. Comparativa de las aplicaciones de cada universidad

Como se observa en la Tabla 3, las dos aplicaciones más destacadas serían UGR y UCMovil, ya que incluyen gran parte de los servicios que son relevantes en este tipo de aplicaciones.

No obstante también hay que tener en cuenta el aspecto y usabilidad que tienen éstas ya que tienen una presentación más bien pobre. Teniendo en cuenta esto se puede destacar la aplicación UPV, la cual tiene una interfaz agradable y un rápido acceso al menú desde cualquier pantalla.

Dos características a tener en cuenta son el apartado de noticias, que lo incluyen todas las aplicaciones analizadas salvo una y las plataformas en las que se podrá ejecutar la aplicación.

Como se puede ver en la tabla, la mayor parte de ítems definidos se corresponden con la información que se indica en el apartado 1.2 de los objetivos del proyecto, exceptuando el acceso directo a las redes sociales de la universidad ya que no es algo que aporte un gran valor añadido y el acceso a contenidos multimedia ya que en la Universidad de Zaragoza son inexistentes.

4. Análisis

Este apartado del documento trata sobre la fase de análisis de la aplicación. El contenido del mismo consiste en un documento de especificación de requisitos. A lo largo del desarrollo del apartado se hará referencia a él como “documento”.

4.1. Introducción

El presente documento contiene una Especificación de Requisitos de Software (ERS) de la aplicación informática *CampusTeruel*, incluyendo la aplicación cliente y la interfaz web de administración. La estructuración de dicha Especificación de Requisitos de Software ha sido realizada siguiendo las directrices que se marcan en el estándar IEEE Recommended Practice for Software Requirements Specifications IEEE 830-1998 [20].

4.1.1. Propósito

El principal propósito del documento es establecer funcionalidades, objetivos y restricciones del producto a desarrollar así como ofrecer una descripción detallada de toda la información relevante para entender cómo funciona la aplicación y cómo va a ser desarrollada. Este documento va dirigido tanto a desarrolladores como a los clientes del producto.

4.1.2. Alcance

El objetivo es obtener un sistema software que permita a los alumnos del campus de Teruel de la Universidad de Zaragoza conocer de una forma rápida todos los servicios que ofrece la universidad. De este modo, los usuarios tendrán un rápido acceso a sus horarios, correos del profesorado o la localización de los centros, entre otras cosas.

Además del software, se realizará también la especificación del hardware necesario para su ejecución con unos tiempos de respuesta que se consideren aceptables.

Se requerirá que toda la documentación referente al proceso de diseño sea conforme al estándar UML.

4.1.3. Definiciones, acrónimos y abreviaturas

En este apartado se proporcionan las definiciones de todos los términos, acrónimos y abreviaturas necesarios para interpretar adecuadamente el presente Documento de Especificación de Requisitos de software. De este modo, tenemos los siguientes términos:

- **Android SDK:** Android Software Development Kit, es un conjunto de herramientas de desarrollo de software que le permiten a un programador crear aplicaciones para dispositivos Android.
- **SQLite:** Sistema de gestión de bases de datos relacional, de pequeño tamaño y libre.
- **RSS:** Really Simple Syndication, un formato XML para sindicar o compartir contenido en la web sin necesidad de un navegador.
- **JSON:** JavaScript Object Notation, es un formato ligero para el intercambio de datos.

- **GSON:** Es una librería de java que se puede utilizar para convertir objetos JAVA en formato JSON y viceversa.
- **ADD:** Anillo Digital Docente, plataforma de la Universidad de Zaragoza para el acceso a recursos
- **PDF:** Portable Document Format, es un formato de almacenamiento para documentos digitales.
- **CU:** Caso de uso, representa una de las acciones que puede realizar el usuario.
- **Noticia:** Noticia accesible en el blog de Teruel.unizar.es
- **Centro:** Edificio que forma parte del campus de Teruel.
- **Profesorado:** Profesores pertenecientes a uno de los centros del campus.

4.1.4. Referencias

Para la elaboración de este documento se han seguido los parámetros indicados en el documento: "IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications", IEEE Computer Society, 1998 [20].

4.1.5. Visión general del documento

El resto del documento está estructurado de la siguiente manera:

- **Descripción general:** Apartado en el que se proporciona una visión general acerca de las funcionalidades de la aplicación móvil y web.
- **Características del Producto:** Descripción general del sistema a desarrollar, con el fin de indicar los factores que afectan al producto y sus requerimientos. Proporciona las principales características a cumplir por el producto.
- **Requisitos específicos:** En este apartado se describen las funcionalidades de la aplicación de un modo más técnico que en el primer apartado, ya que va dirigido principalmente a los desarrolladores.

4.2. Descripción general

4.2.1. Perspectiva del producto

Para que la aplicación lleve a cabo su cometido, es necesario el uso, por parte de ésta, de una serie de componentes software que dan como resultado una aplicación cliente, una aplicación web y una aplicación servidor. La estructura general del sistema es la siguiente:

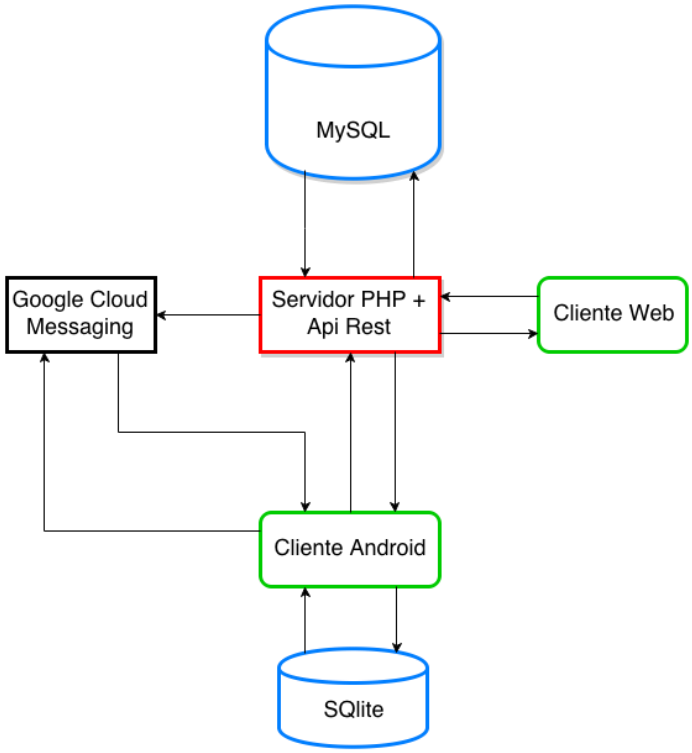


Ilustración 10. Estructura general del sistema

Como se puede ver en la Ilustración 10, el cliente Android se conectará por un lado con el servidor de Google Cloud Messaging (GCM) para activar el servicio de mensajería Push y con el servidor del sistema para recibir los datos de la aplicación a través de la Automatic Api Rest en formato JSON.

La aplicación Android consta de una base de datos SQLite en la cual se almacenarán los datos recibidos por el servidor para evitar la conexión continua con el mismo y permitir un ahorro considerable de datos y energía del dispositivo. Para mantener actualizada esta base de datos local se hará uso de la mensajería Push de Google.

El cliente web será el encargado de realizar una actualización de los datos a través de una interfaz de administrador. Cada vez que realice un cambio el servidor se lo comunicará al GCM y éste se lo comunicará a todos los dispositivos registrados, que deberán comprobar los cambios que se han producido en el servidor y añadirlos a la base de datos local.

Por último, el servidor será el único que interactúe directamente con la base de datos MySQL y será el encargado de mandar los datos necesarios cuando le sean solicitados.

Interfaces del sistema

El sistema a desarrollar consta de dos interfaces:

- Aplicación para dispositivos móviles: Es la aplicación destinada al alumno desde la cual podrá acceder a todos los servicios del campus.
- Aplicación web: Está destinada a la administración de los datos que va a mostrar la aplicación móvil para conseguir que estén lo más actualizados posible.

Interfaces de usuario

Las interfaces de usuario deberán ser adecuadas y accesibles para todo el mundo, teniendo en cuenta posibles deficiencias en la vista o limitaciones de los usuarios. Siguiendo criterios del libro de la Asociación de Interacción Persona-Ordenador (AIPO) [21] en el que está basado la asignatura *Interacción Persona-Ordenador* de la titulación, criterios de usabilidad tales como fuentes adecuadas, colores con suficiente contraste entre ellos y que no sean pesados para la vista.

En todos los casos en los que se realice una acción, existirá una realimentación para que el usuario sepa que el sistema está en proceso de realizarla, hasta que se finalice y se muestren los datos.

Por otra parte, si una acción puede resultar conflictiva o es de gran importancia, se mostrarán los mensajes correspondientes con opciones para la confirmación o cancelación de dichas acciones.

Para la aplicación móvil se consideraran los siguientes aspectos:

- La aplicación está diseñada para ser ejecutada en dispositivos móviles con un tamaño de pantalla medio o grande, y una densidad de píxeles media o alta, para los cuales se garantiza una correcta visualización.
- Se requerirá Android 4.0 o superior.

Para la aplicación web se consideraran los siguientes aspectos:

- Se deberá mostrar adecuadamente en diferentes resoluciones y en diferentes navegadores.

Interfaces hardware

El sistema deberá tener en cuenta las diferentes interfaces hardware que se utilizarán, tal y como se refleja en el siguiente diagrama de despliegue de la Ilustración 11.

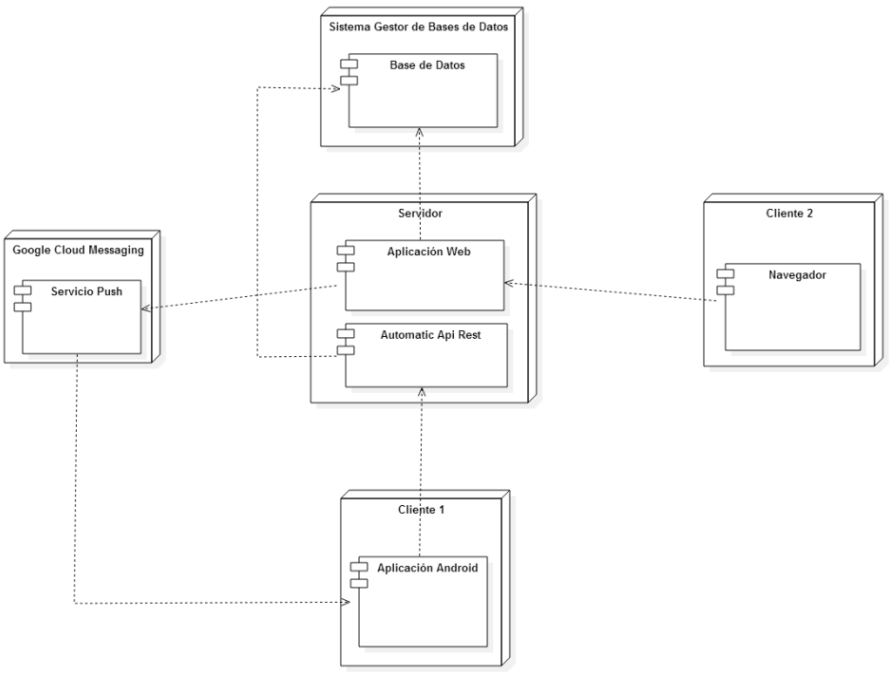


Ilustración 11. Diagrama de despliegue

Los clientes de la aplicación Android se conectarán al servidor a través de la Api Rest, mediante una URL, y éste se conectará a la base de datos para mandarles los datos en formato JSON. Tras esto, la aplicación cliente solo se conectará al servidor cuando reciba un mensaje Push a través del servidor GCM, indicándole que hay cambios nuevos en el servidor.

El administrador que deba acceder a los datos, lo hará a través de su navegador, conectándose a través de la IP o nombre del dominio que tenga la aplicación web en el servidor. Esta accederá a la base de datos para mostrarlos en la interfaz web y cuando se modifiquen los guardará y mandará el mensaje al servidor GCM.

Aunque en la ilustración la base de datos se muestre como un componente independiente del servidor podrá estar alojada en el mismo servidor sin ningún problema.

Será un sistema de alta disponibilidad que contará con todos los servicios duplicados en dos servidores distintos, con una tarjeta de red cada uno conectados a dos redes TCP/IP distintas y un SAI (Sistema de Alimentación ininterrumpida) que permitirá ejecutar las aplicaciones aunque se produzcan cortes de tensión.

Interfaces software

Las interfaces software dependerán de si hablamos del cliente Android, web o el servidor:

- Cliente Android:
  - Sistema operativo Android 4.0 o superior.
- Cliente web:
  - Firefox 8.0.1, Chrome 15.0.874, Internet Explorer 9.
  - Soporte de JavaScript.

- Aplicación servidor:
  - Apache2.
  - Módulo de PHP.
  - Automatic Api Rest.
  - Conectividad con la base de datos MySQL, que puede estar en la misma o en distinta máquina que el servidor.

Interfaces de comunicación

Existirán varios tipos de comunicación:

- El cliente Android se comunicará con el servidor gracias al Api Rest (TCP/IP).
- El cliente Web a través de TCP/IP.
- El servidor deberá poder comunicarse con la base de datos, la cual podrá encontrarse en el mismo equipo que el servidor o no, por lo que se podrá acceder a ella de forma remota.
- La comunicación con GCM se realizará mediante la API de Google.

Restricciones de memoria

El equipo donde se ejecute la aplicación servidor (teniendo la base de datos en otra máquina), deberá tener al menos 4 GB de memoria RAM y 100 MB de espacio libre en el disco duro como mínimo. En el caso de que la base de datos se encuentre en el mismo servidor, el espacio en disco necesario variará dependiendo del tamaño de dicha base de datos.

En el caso de la aplicación cliente Android, los terminales no tienen restricción de memoria RAM, ya que, con cualquier terminal de hoy en día, incluso con los de gama baja, se podría ejecutar la aplicación de forma fluida. En cuanto a almacenamiento interno se refiere, serán necesarios 25 MB libres, como mínimo.

Casos de uso

Aplicación Android

En la Ilustración 12 se muestran los casos de uso de los usuarios de la aplicación:

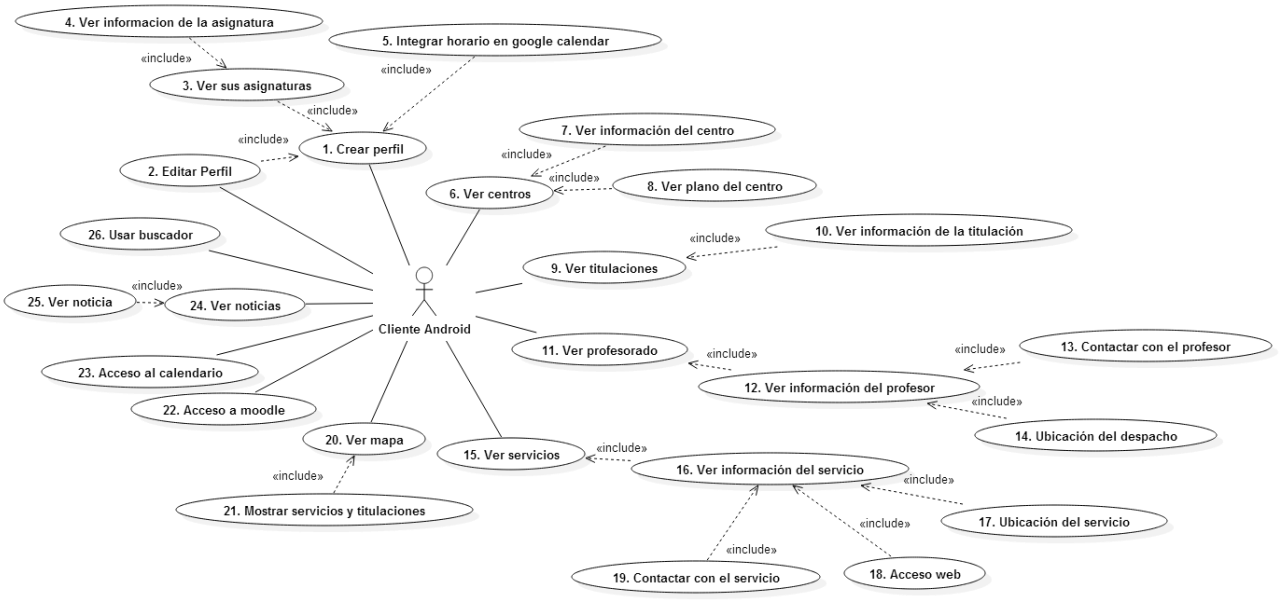


Ilustración 12. Diagrama de casos de uso cliente Android

CU 1. Crear perfil

La interfaz proporcionará al usuario la posibilidad de crear un perfil con su carrera y las asignaturas de las que está matriculado.

CU 2. Editar perfil

El usuario debe tener la posibilidad de modificar los datos que haya introducido en su perfil, ya que son datos que pueden cambiar como mínimo con el cambio de curso.

CU 3. Ver sus asignaturas

El usuario tendrá un acceso rápido a las asignaturas que haya seleccionado en su perfil.

CU 4. Ver información de la asignatura

Al seleccionar una de las asignaturas del perfil, el usuario verá toda la información relacionada con esta asignatura: horarios de clase y exámenes, aulas donde se imparte, profesores que la imparten, etc.

CU 5. Integrar horario en Google Calendar

El usuario podrá integrar en Google Calendar el horario de las asignaturas que tenga seleccionadas en su perfil, teniendo la posibilidad de incluir una notificación si desea recibir una notificación antes de que comience el evento.

CU 6. Ver centros

Se mostrará un listado con los centros del campus.

CU 7. Ver información del centro

Al seleccionar uno de los centros del listado, el usuario podrá ver diferentes opciones, entre ellas la información disponible del centro.

CU 8. Ver plano del centro

Si está disponible, se podrá acceder al plano del centro con indicaciones de dónde se encuentran los servicios, aulas y despachos del profesorado.

CU 9. Ver titulaciones

Se podrá acceder a un listado de las titulaciones relacionadas con el centro o también se podrán encontrar desde el buscador de la aplicación.

CU 10. Ver información de la titulación

Al seleccionar una de las titulaciones, se podrá acceder a la página web de titulaciones de la Universidad de Zaragoza para acceder a la información relacionada con ésta.

También se tendrá acceso a los documentos con los horarios de clases y de los exámenes de la titulación.

CU 11. Ver profesorado

Como con las titulaciones se podrá ver un listado de los profesores relacionados con un centro o se podrá usar el buscador para buscar profesores.

CU 12. Ver información del profesor

Cuando se seleccione un profesor, se podrá ver la siguiente información relacionada con éste: teléfono, correo, ubicación del despacho y horario de tutorías.

CU 13. Contactar con el profesor

Desde la propia pantalla de información del profesor, el usuario se podrá poner en contacto con el profesor pulsando sobre el teléfono o el correo.

CU 14. Ubicación del despacho

Pulsando sobre la ubicación del despacho, el usuario podrá ver el plano del centro donde está ubicado el despacho para que identifique dónde se ubica, podrá ir al mapa de los centros para que vea el centro donde se ubica el despacho y por último tendrá la opción de que le indique cómo llegar al centro.

CU 15. Ver servicios

Se mostrará un listado con los servicios que dispone el campus de Teruel de la Universidad de Zaragoza.

CU 16. Ver información del servicio

Al seleccionar un servicio, el usuario tendrá acceso a la información disponible del servicio como los horarios, la ubicación, acceso web si dispone de él, teléfono o correo, entre otras cosas.

CU 17. Ubicación del servicio

Al pulsar sobre la ubicación, se mostrarán las mismas opciones que el CU 14.

CU 18. Acceso web

En caso de que el servicio disponga de página web, se podrá acceder a ella pulsando sobre la opción. Para el servicio de cafetería, esto se modifica para que muestre los menús de la semana.

CU 19. Contactar con el servicio

Al igual que el CU 13, el usuario tendrá la posibilidad de contactar con el servicio directamente desde la aplicación.

CU 20. Ver mapa

Desde la aplicación se tendrá acceso a un mapa de Google Maps en el que estarán señalizados todos los centros del campus.

CU 21. Mostrar servicios y titulaciones

Al pulsar sobre uno de los centros, se verá de cuántos servicios y titulaciones dispone, y al pulsar sobre la ventana que detalla lo anterior se tendrá acceso a una ventana donde se mostrará qué servicios y titulaciones son, y se tendrá acceso a su información.

CU 22. Acceso a Moodle

Desde la aplicación, el usuario tendrá la opción de acceder a la página web del ADD.

CU 23. Acceso al calendario académico

Se tendrá acceso al PDF del calendario académico, donde se indica el periodo lectivo del curso, fechas de matriculación o las festividades.

CU 24. Ver noticias

El usuario tendrá acceso al listado de noticias que hay en el blog del campus.

CU 25. Ver noticia

Al pulsar sobre una de las noticias, el usuario verá el contenido de la misma.

CU 26. Usar buscador

Desde el buscador, el usuario podrá tener acceso a la información disponible en la aplicación, todos los centros, titulaciones, profesores, servicios y noticias.

Aplicación Web

En la Ilustración 13 se muestran los casos de uso de la interfaz web de administración:

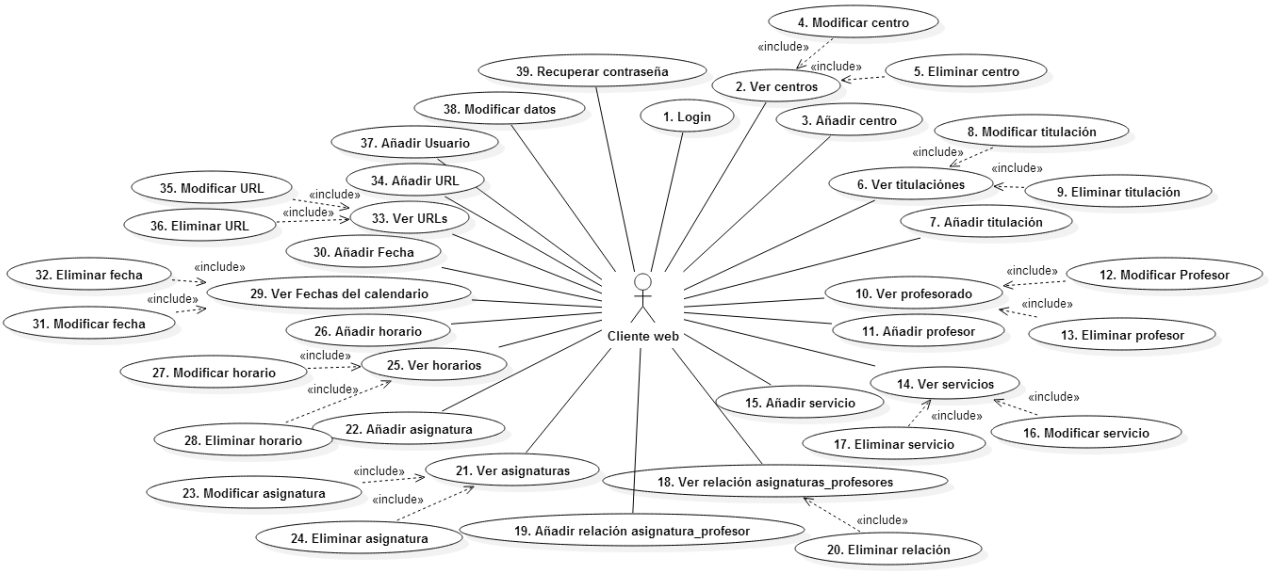


Ilustración 13. Diagrama de casos de uso cliente Web

CU 1. Login

Para acceder a la interfaz es necesario tener un usuario en el sistema y efectuar el ingreso con él.

CU Ver

Los CU 2, 6, 10, 14, 18, 21, 25, 29 y 33 son idénticos. Permiten ver en forma de tabla los datos de los centros, titulaciones, profesores, servicios, relaciones asignaturas-profesor, asignaturas, horarios de las asignaturas, fechas representativas del calendario académico y direcciones Web que utiliza la aplicación como la dirección del calendario en PDF.

CU Añadir

Los CU 3, 7, 11, 15, 19, 22, 26, 30 y 34 permiten acceder a un formulario que permitirá añadir un centro, una titulación, un profesor, un servicio, una relación asignatura-profesor, una asignatura, un horario, una nueva fecha del calendario o una nueva URL al sistema.

CU Modificar

Los CU 4, 8, 12, 16,23, 27, 31 y 35 permiten modificar el centro, titulación, profesor, servicio, asignatura, horario, fecha o URL seleccionada.

CU Eliminar

Los CU 5, 9, 13, 17, 20, 24, 28, 32 y 36 permiten eliminar el centro, titulación, profesor, servicio, relación asignatura-profesor, asignatura, horario, fecha o URL seleccionada.

CU 37. Añadir Usuario

Permite añadir un nuevo usuario de administración al sistema.

CU 38. Modificar datos

Muestra un formulario con los datos del usuario, y permite editarlos y guardarlos.

CU 39. Recuperar contraseña

Esta acción mandará un correo a la dirección de correo del usuario administrador con un enlace, en el cual podrá restablecer su contraseña.

4.2.2. Funciones del Producto

Funciones generales

FUN_GEN_01: El sistema deberá poder manejar un número indeterminado de usuarios.

FUN_GEN_02: El sistema dispondrá de una base de datos para almacenar la información de los servicios y centros del campus.

FUN_GEN_03: La base de datos estará alojada en un servidor central.

FUN_GEN_04: Las aplicaciones clientes accederán a los datos a través de una aplicación servidor.

FUN_GEN_05: Los clientes dispondrán de una base de datos local donde almacenarán los datos del servidor.

FUN_GEN_06: La aplicación deberá estar disponible las 24 horas del día, 365 días al año.

FUN_GEN_07: El sistema deberá estar siempre conectado a Internet.

FUN_GEN_08: El usuario de la aplicación web deberá acceder al sistema identificándose mediante su nombre de usuario y contraseña.

Funciones de la aplicación Android

FUN_AND_01: Crear un perfil del usuario con la carrera y asignaturas.

FUN_AND_02: Ver el horario personalizado en función del perfil.

FUN_AND_03: Integrar el horario con Google Calendar.

FUN_AND_04: Acceso al documento del calendario académico.

FUN_AND_05: Acceso a Moodle.

FUN_AND_06: Acceso a las noticias del campus.

FUN_AND_07: Ver información de cada centro. Esto incluye información sobre:

- Titulaciones.
- Profesorado.
- Contacto.
- Plano de distribución de cada planta.

FUN_AND_08: Ver información de cada servicio. Esto incluye, dependiendo de cada servicio ya que no todos tienen la misma información, información sobre:

- Horarios.
- Contacto.
- Ubicación.
- Acceso a la web del servicio.

FUN_AND_09: Ver en un mapa los centros del campus.

FUN_AND_10: Buscador para encontrar la información rápidamente.

Funciones de la aplicación web

FUN_WEB_01: Añadir, modificar o eliminar centros.

FUN_WEB_02: Añadir, modificar o eliminar titulaciones.

FUN_WEB_03: Añadir, modificar o eliminar asignaturas.

FUN_WEB_04: Añadir, modificar o eliminar profesorado.

FUN_WEB_05: Añadir, modificar o eliminar servicios.

FUN_WEB_06: Añadir o eliminar relaciones de profesores con sus asignaturas.

FUN_WEB_07: Añadir, modificar o eliminar horarios de un asignatura.

FUN_WEB_08: Añadir, modificar o eliminar fechas del calendario académico.

FUN_WEB_09: Añadir, modificar o eliminar URLs que se utilizan en la aplicación.

FUN_WEB_10: Añadir un nuevo usuario.

FUN_WEB_11: Modificar los datos del usuario.

Funciones del servidor

FUN_SER_01: Será la única aplicación que tendrá acceso a la base de datos del sistema.

FUN_SER_02: Deberá servir los datos a la aplicación cliente, a través del API Rest, cuando sean solicitados.

FUN_SER_03: Deberá ser capaz de registrar los dispositivos para el uso de GCM.

FUN_SER_04: Dará acceso a la aplicación web.

FUN_SER_05: Validará el acceso a la aplicación web.

4.2.3. Características del usuario

Se pueden distinguir dos tipos de usuarios:

- **Usuario estándar:** Cualquier tipo de usuario que posea un dispositivo Android (teléfono inteligente o *tablet*) y desee utilizar la aplicación. Estos usuarios no tienen por qué tener conocimientos avanzados de informática, por lo que el uso de la aplicación deberá simplificarse todo lo posible, proporcionando una interfaz de usuario agradable y sencilla.
- **Usuario administrador:** Accederá desde la aplicación web. Se encargará del mantenimiento de la información del servidor, teniendo plenos privilegios para añadir, modificar o eliminar cualquier elemento. No tendrá por qué tener unos conocimientos avanzados, por lo que se deberá desarrollar una interfaz intuitiva y usable.

4.2.4. Restricciones

De diseño

1. Uso de la notación de diagramas de diseño UML.
2. Seguimiento del documento "IEEE Recommended Practice for Software Requirements Specifications" [20] para la documentación.

De implementación

1. La aplicación cliente deberá estar escrita con Java utilizando el SDK de Android, concretamente la API nivel 14.
2. La aplicación web deberá estar escrita en PHP.
3. Se contará con un servidor Apache y módulo de PHP.
4. Se deberá usar MySQL para la creación de la base de datos.

Del sistema

1. Se requerirá Android 4.0 o superior en los terminales que ejecuten la aplicación.
2. El dispositivo deberá disponer de conexión a Internet, en el inicio de la aplicación.
3. Se almacenarán datos en una base de datos interna en el dispositivo.
4. Se requerirá almacenar datos en la memoria externa del dispositivo.
5. El servidor deberá estar alojado en una máquina accesible a través de Internet para que tanto el cliente web como el cliente Android tengan acceso desde cualquier lugar.
6. La base de datos deberá estar alojada en un equipo al que tenga acceso el servidor.
7. Se utilizará la API de Google Maps, por lo que deberán cumplirse las limitaciones de ésta.
8. Se utilizará la API de GCM, por lo que deberán cumplirse las limitaciones de ésta.

De usabilidad

1. La interfaz deberá ser clara, sencilla y no sobrecargada, tanto para la aplicación Android como para la aplicación Web.
2. La interfaz deberá estar adaptada para que se visualice correctamente en distintos dispositivos móviles con distintas resoluciones.

4.2.5. Supuestos y dependencias

Para el correcto funcionamiento de la aplicación, se da por supuesto que los componentes indicados en el apartado Interfaces Software están instalados y configurados correctamente. También se presupone que tanto el cliente como el servidor tendrán acceso a Internet.

4.3. Especificación de requisitos

La especificación de requisitos del sistema impone restricciones en el diseño o la implementación para la creación del sistema. Para realizar la generación de éstos se han tenido en cuenta los aspectos descritos en los apartados anteriores donde se describe la estructura del sistema.

En el Anexo A: Requisitos se presentan en detalle los requisitos del sistema desarrollado en el presente proyecto.

5. Planificación y cálculo de costes

En el presente apartado se va a desarrollar la planificación realizada antes de acometer el desarrollo del proyecto y el cálculo de los costes que ocasionaría dicho desarrollo.

5.1. Planificación

Para realizar la planificación del presente proyecto, se ha tenido en cuenta cada una de las fases necesarias para finalizar exitosamente cualquier desarrollo software de cierta envergadura. Dichas fases son la de análisis, diseño, implementación y pruebas.

5.1.1. Planificación inicial

La planificación inicial está pensada para que ocupe todo el primer cuatrimestre del curso, pensando en presentar el proyecto en febrero de 2015. De esta forma la duración estimada de proyecto será de 131 días.

En el apartado B.1. Planificación inicial, que se incluye en el Anexo B: Planificación, se presentan los diagramas de Gantt de la planificación inicial.

5.1.2. Planificación final

Finalmente la duración total del proyecto ha costado 217 días. La principal causa de la demora de días con respecto a la planificación inicial se debe a mi incorporación al mundo laboral desde mediados de Octubre del 2014, además del trabajo ocasionado por dos asignaturas que me faltaban por cursar.

Otra de las causas han sido las dificultades encontradas a la hora de desarrollar algunos de los ítems de la planificación, que se pensaba que iban a ser más sencillos pero que a la hora de realizar el desarrollo se ha visto su mayor dificultad, destacando la correcta sincronización de la base de datos del servidor con la base de datos local.

En el apartado B.2. Planificación final, que se incluye en el Anexo B: Planificación, se presentan los diagramas de Gantt de la planificación final.

5.2. Estimación de costes

En el desarrollo de aplicaciones informáticas, al igual que en el resto de actividades comerciales, es importante conocer los costes para obtener la rentabilidad del producto. Por ello es necesario realizar una estimación de costes, para evitar tener pérdidas y sobre todo, para obtener beneficios económicos.

He planificado un calendario laboral que implica media jornada laboral, es decir, 4 horas diarias. Se ha supuesto un esfuerzo continuo aunque la línea de carga no ha sido completamente regular a lo largo de todo el proceso, por lo que la solución adoptada refleja una media real de las horas invertidas en el desarrollo del proyecto completo.

A partir de la planificación es posible calcular fácilmente el número de horas totales dedicadas al proyecto, ascendiendo el total de horas invertidas en el proyecto a 518 durante poco más de 4 meses de trabajo con la distribución de horas que figura en la Tabla 4.

Distribución de horas entre fases		
Fase	**Cálculo**	**Total**
Análisis	21 días x 3 horas/día	63 horas
Diseño	33 días x 3 horas/día	99 horas
Implementación	66 días x 3 horas/día	198 horas
Pruebas	9 días x 3 horas/día	27 horas
Documentación	131 días x 1 horas/día	131 horas
Total		**518 horas**

Tabla 4. Distribución de horas de trabajo – Planificación inicial

5.2.1. Coste estimado

Asumiendo que el sueldo básico neto diario de un Ingeniero, según el convenio colectivo en vigor [22], es de 61.03€/día, el coste estimado de la mano de obra es el siguiente:

61,03€ / 8 horas de jornada de trabajo = 7,63€/hora

518 horas * 7,63€/hora = **3.952,34€**

A este coste se le debe añadir el coste del software y hardware necesario para el desarrollo del proyecto:

Equipo	Coste
Ordenador portátil Dell Latitude 14 5000 series	799€
Smartphone Motorola Moto G	169€
Microsoft Office 2010 Home and Student	109€
Licencia Balsamiq Mockups	80€
Total	**1157€**

Tabla 5. Coste del equipamiento

El coste total del proyecto por lo tanto es el siguiente:

3.952,34 + 1157 = **5.109,34€**

5.2.2. Coste real

A continuación calcularemos el coste real de la aplicación para ver la diferencia existente con el coste estimado. Con los datos obtenidos del diagrama de Gantt, disponible en el apartado B.2. Planificación final del Anexo B: Planificación, tenemos:

Distribución de horas entre fases		
Fase	**Cálculo**	**Total**
Análisis	29 días x 3 horas/día	87 horas
Diseño	47 días x 3 horas/día	141 horas
Implementación	124 días x 3 horas/día	372 horas
Pruebas	14 días x 3 horas/día	42 horas
Documentación	217 días x 1 horas/día	217 horas
Total		**859 horas**

Tabla 6. Distribución de horas de trabajo – Planificación final

Realizando los cálculos anteriores con el cumulo de horas obtenido de la planificación final, se obtienen los siguientes resultados:

859 horas * 7,63€/hora = **6.554,17€**

6.554,17 + 1157 = **7.711,17€**

5.2.3. Conclusiones

Tras haber calculado los costes estimado y real, observamos que se ha producido una desviación de 33,74% debido al incremento del número de días empleados en el desarrollo del proyecto.

6. Diseño

Este apartado se centra en la fase de diseño de la aplicación. Dicho diseño debe ser exhaustivo y claro, de manera que no se generen dudas o confusiones durante el proceso de implementación.

6.1. Base de datos

Para desarrollar la aplicación es necesario contar con una base de datos central donde se almacene toda la información que mostrará la aplicación, los usuarios que tendrán acceso a la interfaz de administración, o los identificadores de los dispositivos que tengan la aplicación para poder utilizar el servicio GCM. Por otro lado, se contará con una base de datos local en cada dispositivo que permitirá evitar abusar de una conexión de datos constante.

6.1.1. Servidor-Web

Durante este apartado se expondrá la estructura de la base de datos central del sistema, con una explicación del diseño conceptual de la misma. En el apartado C.1. Servidor Web del Anexo C: Base de datos se presenta en detalle el diseño lógico y físico de la base de datos.

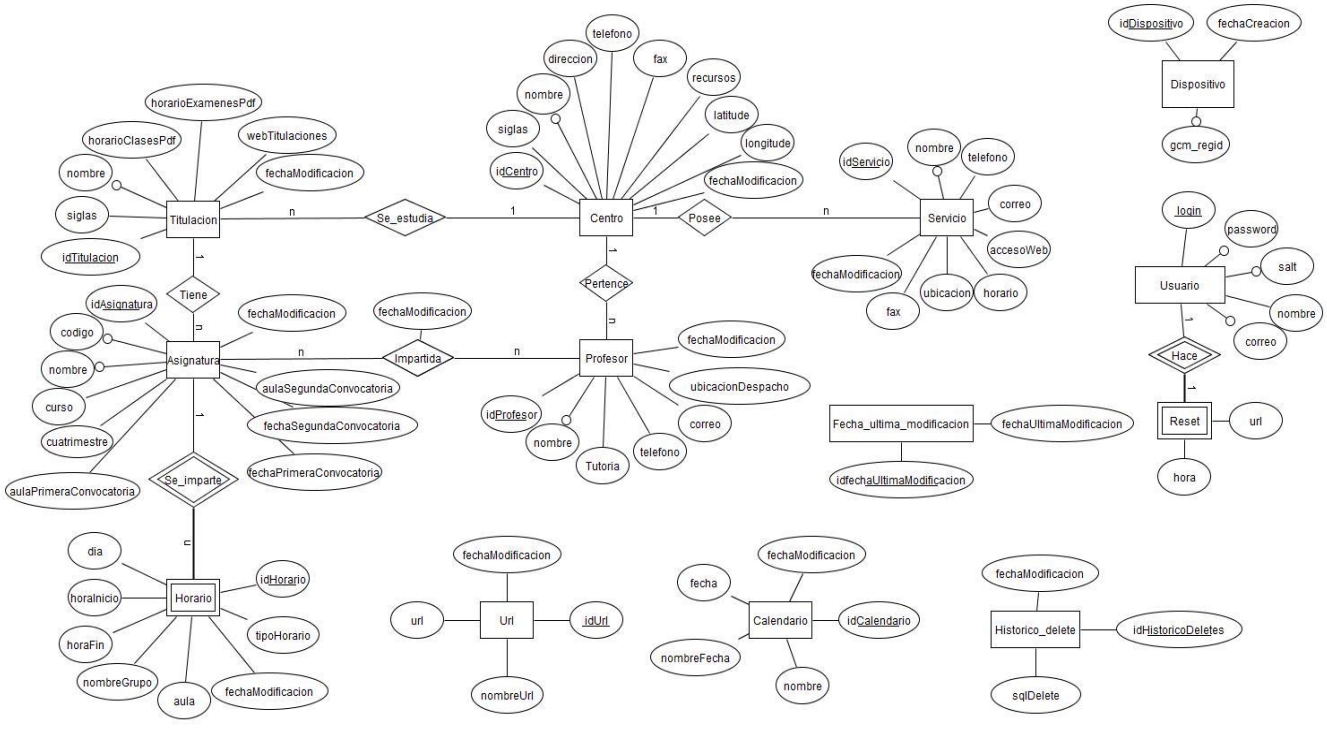


Ilustración 14. Esquema Entidad-Relación de la base de datos del servidor

Como se puede observar en la Ilustración 14, la base de datos se compone de 12 entidades con un número significativo de atributos. Tiene un nivel medio de complejidad ya que la mayor parte de sus relaciones son simples. A continuación, se realiza una explicación más detallada de las entidades:

- Usuario: En ella se especifica todos los datos necesarios y relativos al usuario que va a hacer uso de la interfaz web de administración.
- Reset: Representa el enlace que permite al usuario recuperar su contraseña, cada usuario solo puede tener únicamente un reset, o sea, una dirección que le permita modificar su contraseña desde fuera.
- Dispositivo: Guarda los identificadores del GCM de los dispositivos que instalan la aplicación.
- Fecha_ultima_modificacion: Contiene la fecha en la que se realizaron los últimos cambios relacionados con la información que muestra la aplicación.
- Historico_delete: Permite almacenar las consultas SQL que eliminan información de la base de datos en el servidor, para después ejecutarlas en la aplicación y mantener así la base de datos local completamente actualizada.
- Calendario: Almacena la información relacionada con las fechas festivas del calendario académico, vacaciones, inicio de los semestres, etc. Esto permite tener en un sitio las fechas que la aplicación debe consultar para hacer una importación correcta del horario del usuario a Google Calendar.
- Url: Contienen URLs que son utilizadas por la aplicación como la dirección del calendario académico en PDF o la web de Moodle.
- Centro: Entidad que representa un edificio del campus, con su dirección, información de contacto, etc.
- Titulación: Almacena la información relativa a una titulación del campus, como los horarios en PDF o el enlace a la web de titulaciones.
- Asignatura: Representa la asignatura de una titulación, y contiene información como el curso al que pertenece, cuatrimestre o fechas de los exámenes.
- Horario: Contiene la información horaria de una asignatura, indicando el día, el intervalo de horas en que se produce y además incluye información sobre el grupo al que pertenece en caso de que así sea, o el aula.
- Profesor: Representa a un miembro docente del campus y guarda información de contacto como el teléfono, correo e información de la ubicación de su despacho.
- Servicio: Entidad que almacena información relativa a los servicios del campus como reprografía o Universa, con información sobre la ubicación y teléfono de contacto.

6.1.2. Cliente

Durante este apartado se expondrá la estructura de la base de datos local de la aplicación cliente, con una explicación del diseño conceptual de la misma. En el apartado C.2. Cliente del Anexo C: Base de datos se presenta en detalle el diseño lógico y físico de la base de datos.

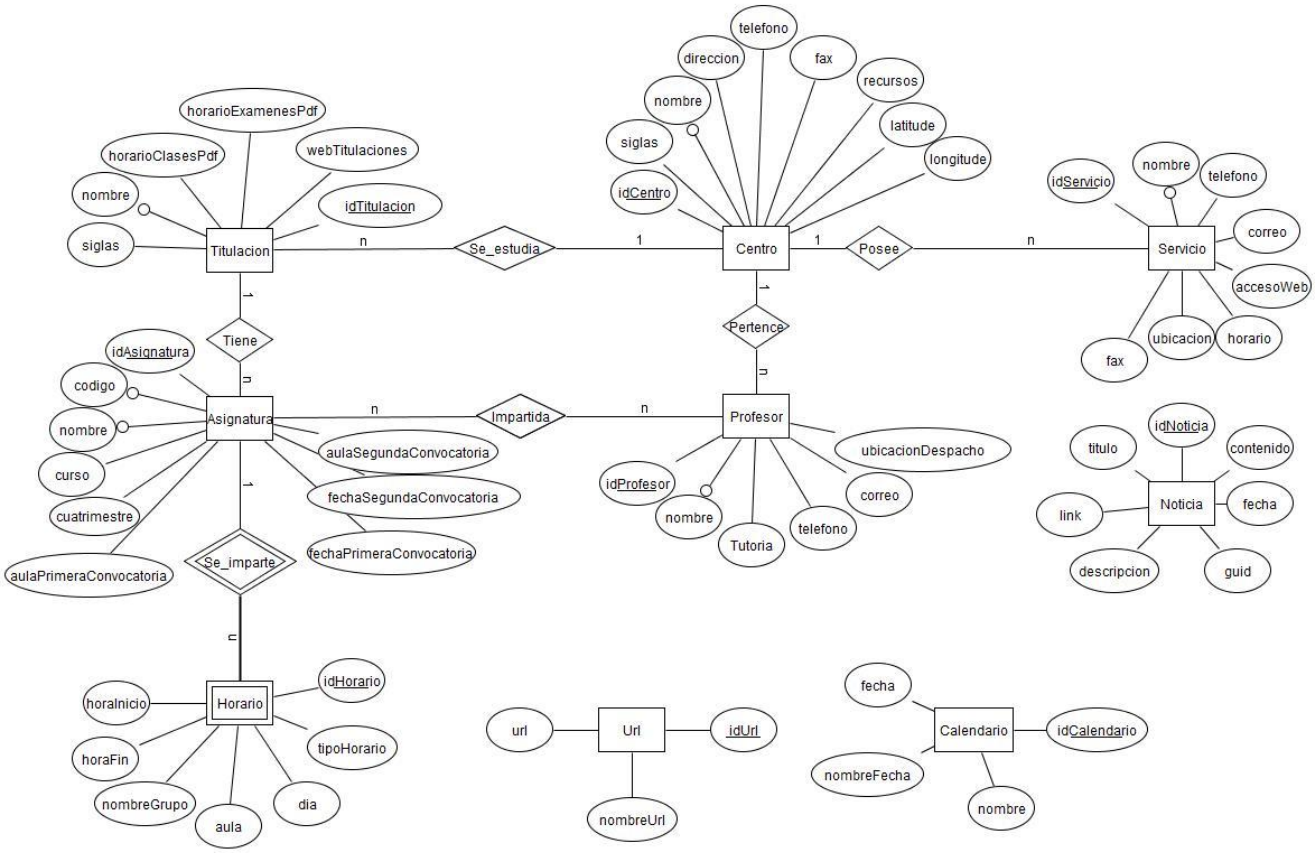


Ilustración 15. Diagrama Entidad-Relación de la base de datos de la aplicación cliente

La Ilustración 15 representa la base de datos que contendrá la aplicación de manera local. Gracias a esta base de datos se agilizará el acceso a los datos para visualizarlos, además del ahorro en el consumo de recursos del dispositivo que supone.

Es una base de datos más sencilla pero que tiene el mismo esqueleto que la base de datos del servidor. Concretamente, contiene las tablas que representan la información que se mostrará en la aplicación.

6.2. Interfaz de usuario

La interfaz de usuario es un aspecto importante en la mayoría de aplicaciones informáticas. Además, en el caso de la aplicación que estamos desarrollando, adquiere especial importancia al necesitar una visualización clara y sencilla de toda la información del sistema. En este apartado, se desarrollan todos los aspectos relacionados con dicha interfaz.

6.2.1. Interfaz "cliente"

Árbol de pantallas

En la siguiente figura, se muestra el árbol de pantallas de la presente aplicación. Se muestra cada una de las diferentes pantallas, además de las diferentes transiciones entre las mismas.

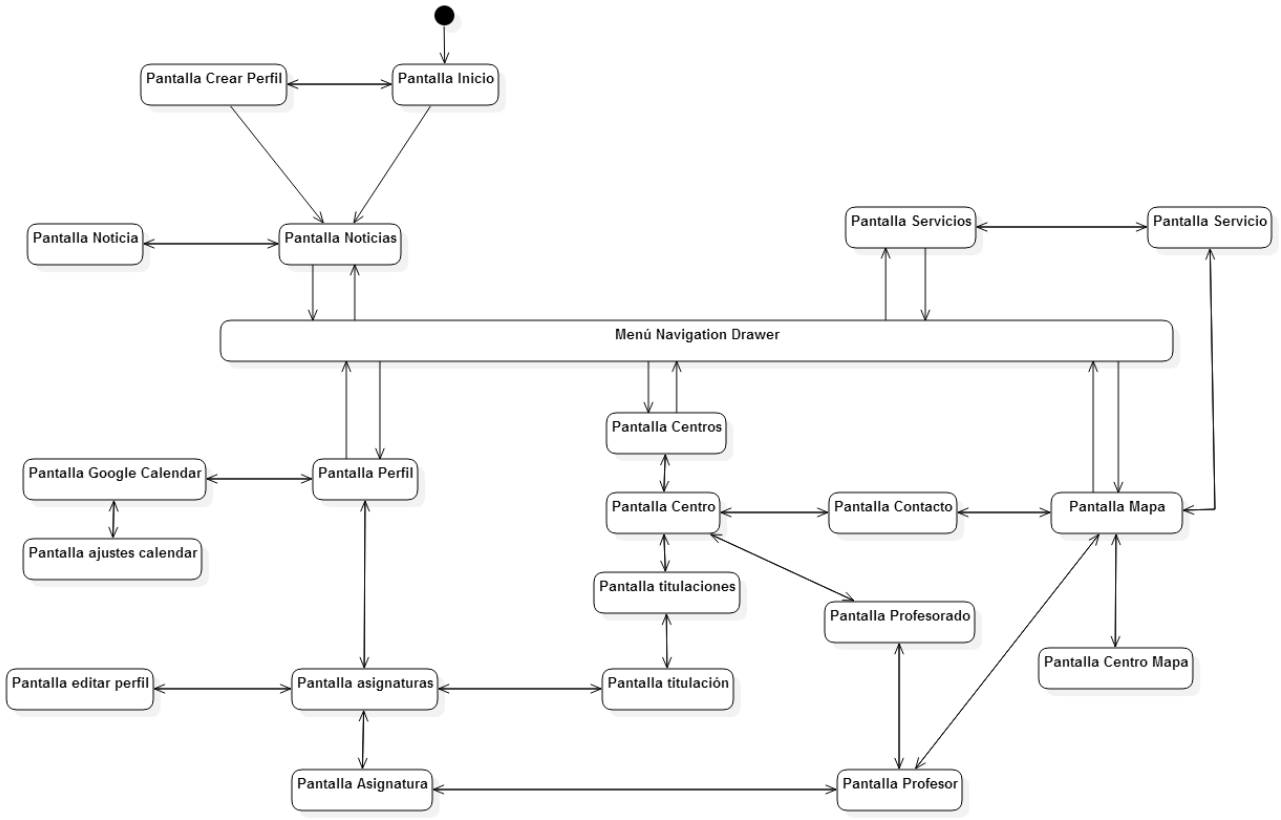


Ilustración 16. Árbol de pantallas de la aplicación cliente

Como se puede observar en la Ilustración 16, cada ente en forma de burbuja, corresponde con una pantalla distinta de la aplicación cliente.

La pantalla de entrada cuando abrimos la aplicación es la *Pantalla Inicio*, esta pantalla se mostrará únicamente la primera vez que se inicie la aplicación para dar la opción al usuario de elegir entre crear o no un perfil. Desde ella se puede acceder a la *Pantalla Crear Perfil* para crear el perfil o acceder a la *Pantalla Noticias*, que será la primera pantalla a mostrar una vez el usuario haya elegido o no entre crear un perfil.

Desde la *Pantalla Noticias*, podemos acceder a una noticia en la *Pantalla Noticia* o con el *Menú NavigationDrawer* movernos por la aplicación accediendo al resto de pantallas, como la *Pantalla Perfil,* que da acceso a la *Pantalla asignaturas* que muestra las asignaturas del perfil.

Otras pantallas importantes serían *Pantalla Centros, Pantalla Mapa* o *Pantalla Servicios* que dan acceso a todas las funcionalidades descritas en los requisitos de la aplicación.

Prototipo de interfaz

Previamente a la implementación de la interfaz gráfica de usuario, se han realizado los prototipos (*mockups*) de cada una de las pantallas que forman la aplicación. Esto se hace para tener una idea clara de cómo deben ser cada una de las pantallas, así como sus funcionalidades. En el apartado D.1. Prototipos Android del Anexo D: Prototipos se presentan las imágenes de los prototipos con sus descripciones.

6.2.2. Interfaz "web"

Árbol de pantallas

En la siguiente ilustración se puede observar el árbol de pantallas de la interfaz web de administración, con sus diferentes transacciones.

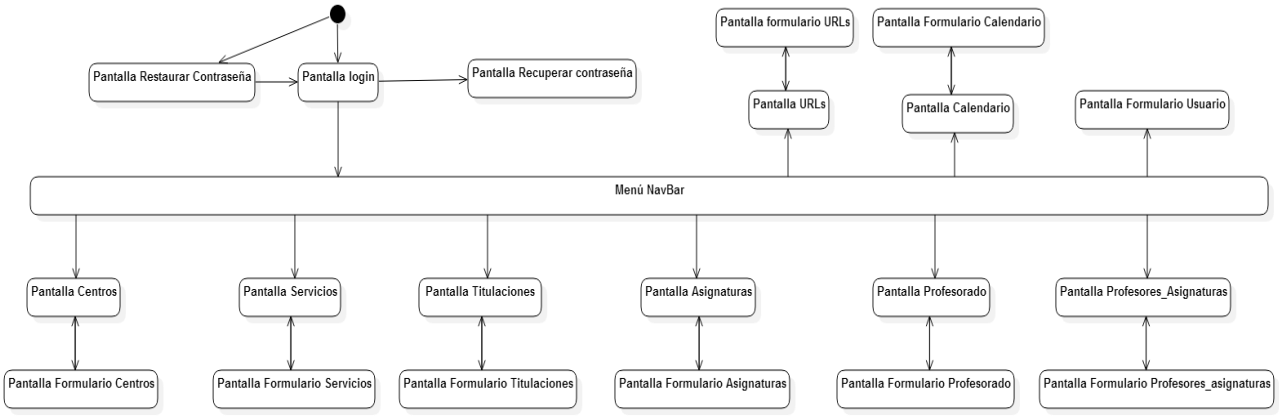


Ilustración 17. Árbol de pantallas web

Como se puede ver, de inicio se accede a la *Pantalla Login,* que permite al usuario introducir sus datos para el acceso. Desde esa página se podrá acceder a la *Pantalla Recuperar Contraseña* que permitirá solicitar la recuperación de la contraseña. En el caso de haber solicitado restaurar la contraseña, se podrá acceder directamente a esta página desde la URL que se manda en el correo de recuperación de la contraseña.

Una vez estamos dentro de la aplicación, desde el *Menú NavBar* el usuario se moverá por toda la aplicación. El menú estará siempre visible para que el usuario pueda acceder a cualquier parte desde cualquier sitio. Desde él se accederá a las diferentes pantallas para ver los datos que hay en el sistema y los formularios que permitirán editarlos

Prototipo de interfaz

Como en el caso del cliente móvil se han desarrollado unos prototipos para orientar la implementación de la interfaz web. En el apartado D.2. Prototipos web del Anexo D: Prototipos, se presentan todos los prototipos desarrollados.

6.3. Diagramas de clases

En este apartado se van a exponer los diagramas de clases de nuestro proyecto, tanto de la aplicación web como de la aplicación cliente. En ambos casos se mostrara el diagrama de clases completo con sus relaciones y se indicara donde se pueden visualizar las clases en detalle.

Aplicación cliente

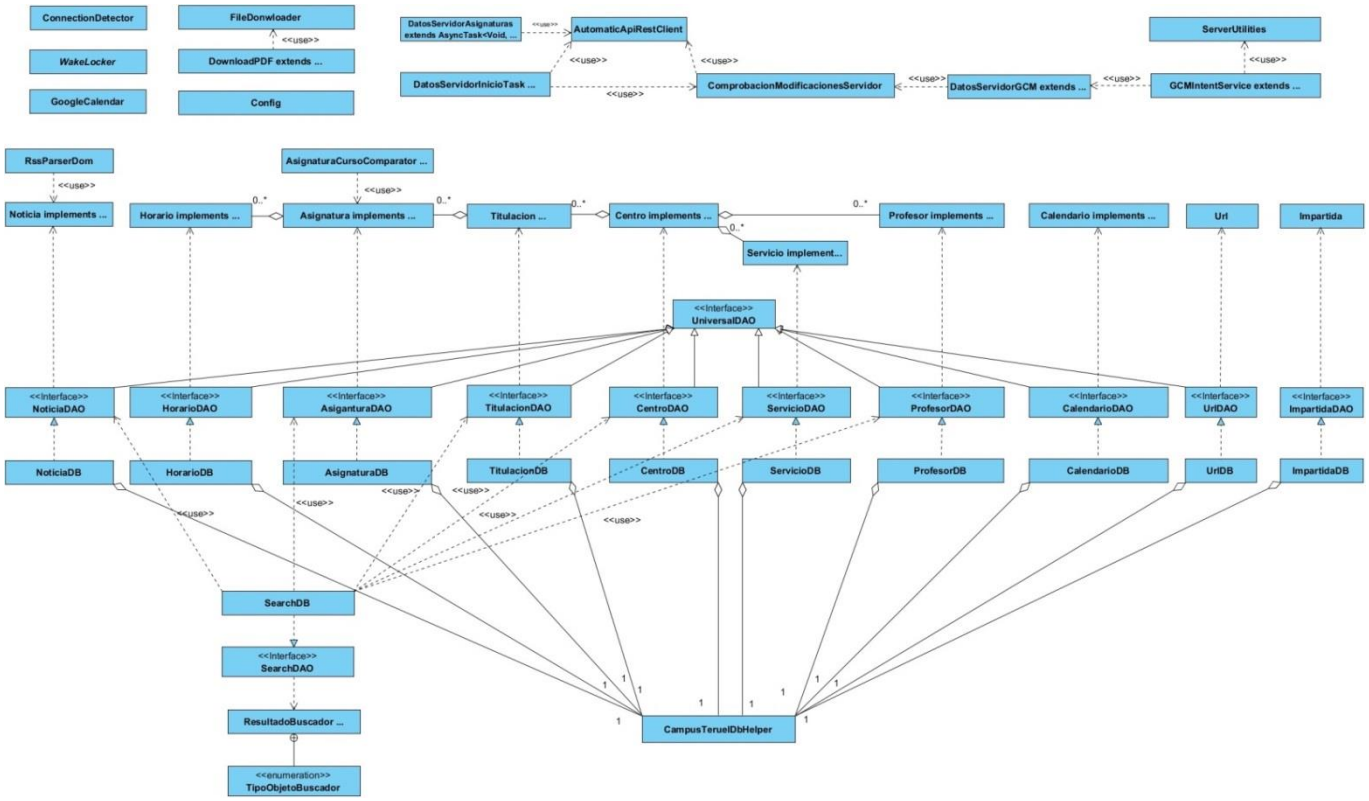


Ilustración 18. Diagrama de clases de la aplicación Android

En la anterior ilustración, se puede ver el diagrama de clases completo de la aplicación Android. Las clases que aparecen en primera línea de la imagen sirven para interactuar con el exterior de la aplicación, por ejemplo, con Google Calendar o el servidor del sistema. La siguiente línea de clases se corresponden a los objetos de los datos, y el último grupo son las interfaces y clases que las implementan para tener acceso a la base de datos del dispositivo.

En el apartado E.1. Aplicación cliente del Anexo E: Diagramas de clases se detallan todas las clases de la aplicación.

Aplicación web

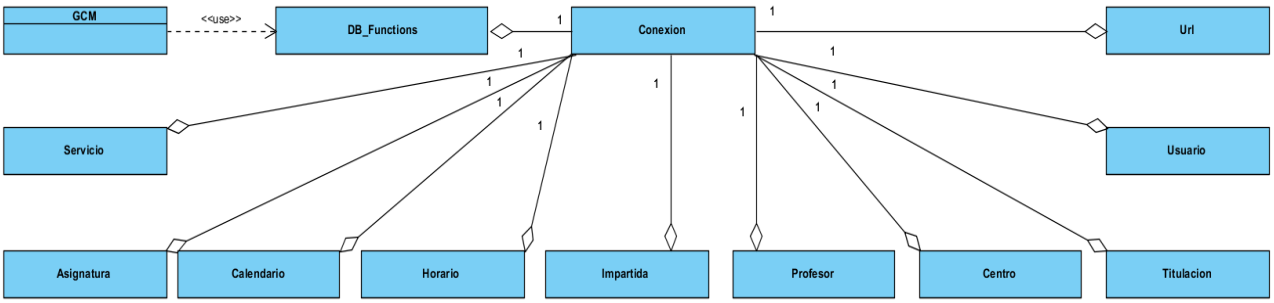


Ilustración 19. Diagrama de clases de la aplicación web

El diagrama de clases de la aplicación web básicamente se basa en las clases que contienen los métodos que interactúan con la base de datos, para poder mostrarlos y editarlos.

En el apartado E.2. Aplicación web del Anexo E: Diagramas de clases se pueden ver en detalle las clases del diagrama de clases del servidor.

6.4. Diagramas de actividad

En el Anexo F: Diagramas de Actividad se encuentran los diagramas de actividad de la aplicación cliente y la aplicación web, que permiten ver mediante una serie de acciones los procesos que se llevan a cabo en ambas.

6.5. Diagramas de secuencia

En el Anexo G: Diagramas de secuencia se encuentran los diagramas de secuencia de la aplicación cliente y la aplicación web, que representan la secuencia de mensajes entre las instancias de clases, interfaz y actores para poder visualizar los procesos en detalle.

7. Validación

Uno de los aspectos importantes una vez implementada la aplicación es realizar una correcta validación de la misma, incidiendo en la aceptación que pueda tener por parte de sus usuarios potenciales, en este caso los alumnos del campus de Teruel.

Dado que se trata de un proyecto dirigido a éstos, se ha realizado una encuesta a través de Google Drive, que pretende conocer la opinión de los alumnos en los siguientes aspectos:

- Qué les parece el diseño de la aplicación
- La dificultad de manejo.
- Posibles errores durante la ejecución.
- Posibles mejoras.
- Qué secciones son más interesantes.
- La acogida que podría tener en caso de distribuirla a través de Google Play.

La encuesta la han realizado un total de 20 alumnos de diferentes cursos y carreras del campus de Teruel de la Universidad de Zaragoza. Previamente, se les pidió que instalaran la aplicación en sus dispositivos para probarla antes de contestar las preguntas. En el Anexo H: Encuesta se presenta la encuesta en detalle.

Los resultados obtenidos de la encuesta se muestran en los siguientes apartados.

7.1. Diseño de la aplicación

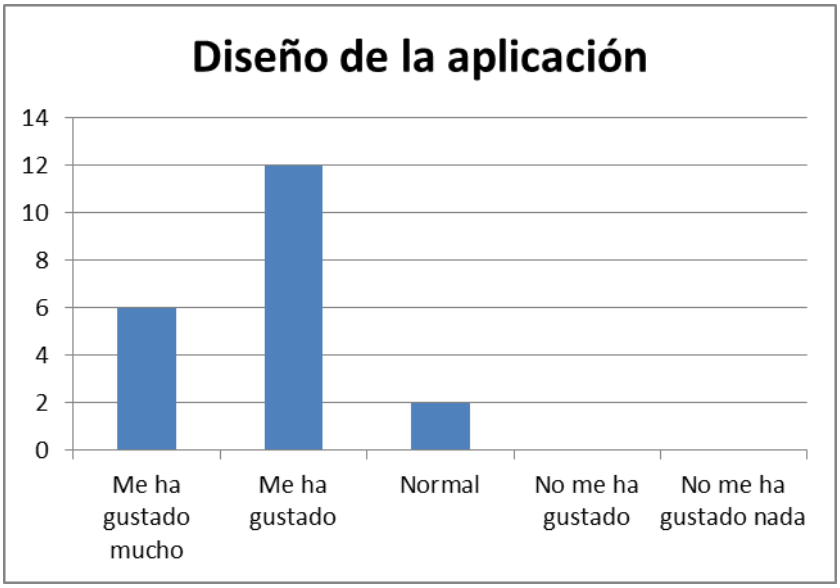


Ilustración 20. Diseño de la aplicación

Según los resultados mostrados en la ilustración anterior, el diseño de la aplicación ha resultado ha resultado más que aceptable. Únicamente a dos personas les ha parecido un diseño normal, al resto les ha gustado o les ha gustado mucho.

7.2. Dificultad de manejo

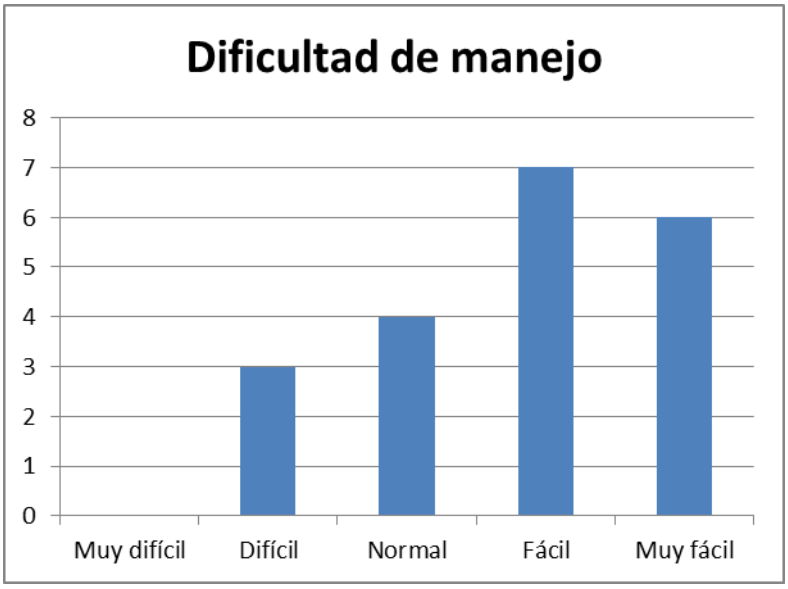


Ilustración 21. Dificultad de manejo

Tres personas son las únicas que consideran que tiene un manejo difícil, el resto les parece que tiene un manejo normal o fácil.

7.3. Errores durante la ejecución

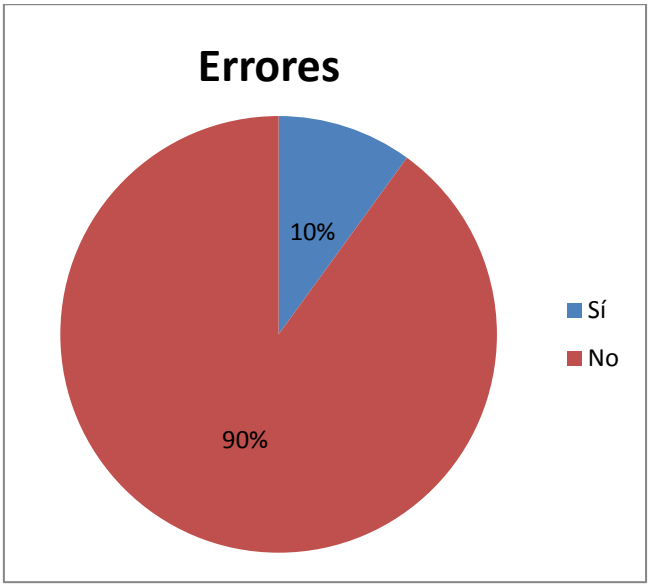


Ilustración 22. Errores en la ejecución

Únicamente dos personas tuvieron problemas durante la ejecución de la aplicación. Una de ellas indico en el apartado para introducir el error que tuvo un problema al cargar el mapa, desgraciadamente la carga del mapa de Google Maps consume muchos recursos y eso no es algo que se pueda cambiar desde el código y la otra perdió la conexión a la red mientras se descargaba datos de la aplicación.

7.4. Secciones de la aplicación

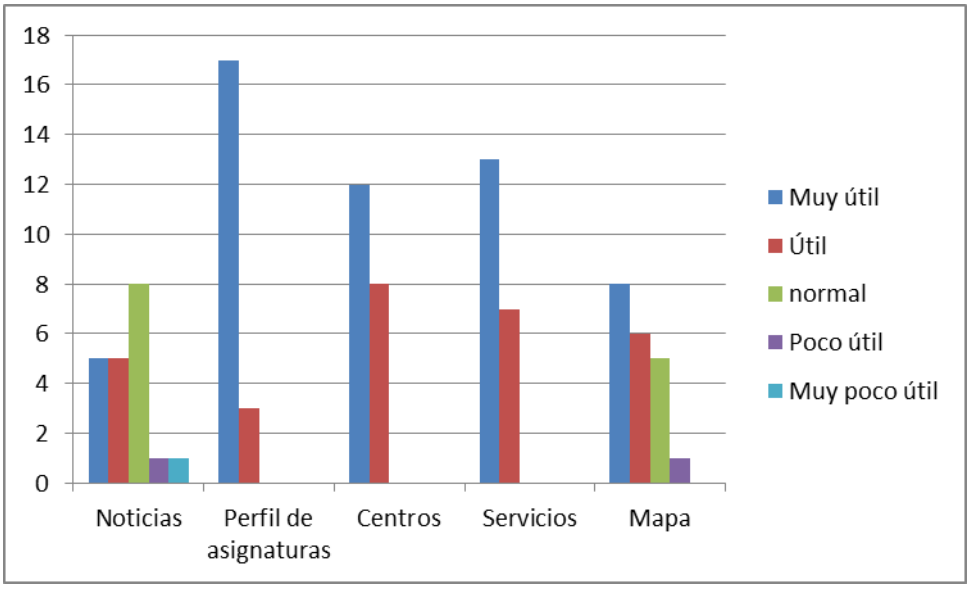


Ilustración 23. Secciones de la aplicación

En la anterior ilustración se puede observar la opinión de los usuarios respecto a las secciones que contiene la aplicación, como se puede observar las secciones “Perfil de asignaturas”, “Centros” y “Servicios” has tenido una gran acogida y para los usuarios les parecen de gran utilidad. Las secciones “Mapa” y “Noticias”, les parecen más secundarias sobre todo “Noticias”.

7.5. Google Play

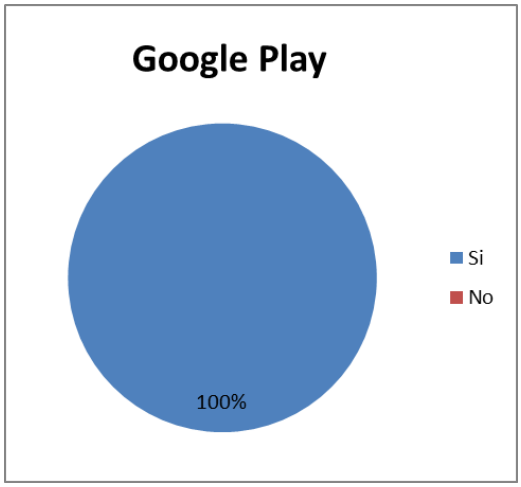


Ilustración 24. Google Play

Para los usuarios que han probado la aplicación y han realizado la encuesta la respuesta a si debería subirse la aplicación a Google Play es clara, un sí rotundo. Esto es muy probable que no sea solo porque les haya podido gustar más o menos el diseño o las funcionalidades que contiene, sino porque es un servicio del que el campus no dispone, el cual permite suplir una carencia importante y es el acceso rápido a toda la información disponible.

8. Conclusiones

El principal objetivo que se ha pretendido conseguir con la realización de este proyecto ha sido diseñar e implementar una aplicación que ayude a los alumnos del campus de Teruel de la Universidad de Zaragoza a la hora de buscar información sobre el mismo, con el consiguiente ahorro de tiempo.

Más concretamente, se ha desarrollado un sistema integral que consta de dos aplicaciones diferentes (aplicación cliente y aplicación web) desarrolladas para Android, que hoy en día, es un sistema operativo para móviles que está mayoritariamente extendido tanto en nuestro país como a nivel mundial, y utilizando PHP, el lenguaje web por excelencia. Además, el sistema conecta dos sistemas gestores de base de datos relacionales: MySQL, para el sistema del servidor y SQLite, para la aplicación Android.

Cabe destacar que la aplicación *CampusTeruel* cumple con la totalidad de los objetivos definidos inicialmente: se ha creado una aplicación distribuida (cliente-servidor) en la que el servidor central es capaz de manejar y comunicarse con varios clientes de forma simultánea; se muestra toda la información del campus; y finalmente, ha servido para ampliar mis conocimientos de Android.

Durante la creación de todo este complejo sistema, además de poner en práctica multitud de conceptos aprendidos durante la carrera, he tenido que desarrollar otra serie de habilidades, como enfrentarme a retos de mayor entidad o el uso de tecnologías que no había utilizado hasta ahora. La experiencia ha sido muy satisfactoria y ya ha comenzado a ser de utilidad durante la vida laboral que he comenzado este año.

Por último, me gustaría destacar los resultados de la encuesta, ya que aunque sean a pequeña escala, parece claro que la aplicación podría tener una gran acogida entre el alumnado.

9. Líneas futuras

Algunas de las líneas futuras de trabajo que se pueden destacar son las siguientes:

- **Nuevos contenidos:** Añadir información nueva a la aplicación como información sobre los Trabajos Fin de Grado o las directivas de los centros.
- **Conectar con la biblioteca:** Tener la posibilidad de ver los préstamos que tiene el usuario y poder aumentar el plazo del tiempo, o ver el catálogo de libros disponibles.
- **Mejoras en el diseño:** Realizar posibles mejoras en la interfaz, optimizar las pantallas para *tablets* que se consideren y tener en cuenta la opinión de los usuarios para hacer una interfaz más intuitiva.
- **Expansión a otras plataformas:** Se podrá crear la aplicación cliente para otras plataformas móviles de actualidad para ampliar el número de usuarios de nuestro servicio, como iOS o Windows Phone.
- **Internacionalización de la aplicación:** La posibilidad de que la aplicación esté traducida a otros idiomas haría que también fuera accesible para los alumnos de Erasmus del campus, a los cuales les podría ayudar enormemente.
- **Mantenimiento:** Si la aplicación es puesta en funcionamiento será necesario realizar un mantenimiento de la información que contiene. Una posibilidad sería que el campus tuviera un sistema centralizado para almacenar la información y recoger de ahí los datos en lugar de almacenarlos en una base de datos propia para el sistema.
- **Mejorar la aplicación web:** Se pueden aprovechar la creación de usuarios con distintos permisos que actualicen únicamente los datos a los que tienen acceso con sus permisos y así repartir la carga de trabajo que supone la tarea de mantener los datos actualizados.
- **Noticias:** Dado que desde la aplicación se accede rápidamente a las noticias, un mejor contenido de estas ayudaría a una subida de popularidad de las mismas.

Bibliografía

[1] Android. (2015, Marzo) Wikipedia. [Online]. http://es.wikipedia.org/wiki/Android

[2] Arquitectura Android. (2015, Marzo) Androideity. [Online]. http://androideity.com/2011/07/04/arquitectura-de-android/

[3] ART. (2015, Marzo) El android libre. [Online]. http://www.elandroidelibre.com/2014/08/entendiendo-el-impacto-de-art-la-nueva-maquina-virtual-de-android.html

[4] Dashboards. (2015, Marzo) Developer Android. [Online]. https://developer.android.com/about/dashboards/index.html?utm_source=ausdroid

[5] Cuota de mercado. (2015, Marzo) Xatakaandroid. [Online]. http://www.xatakandroid.com/sistema-operativo/android-consolido-su-liderazgo-en-2014-alcanzando-el-81-5-de-cuota-de-mercado

[6] GCM. (2015, Marzo) Android developers. [Online]. https://developer.android.com/google/gcm/gcm.html

[7] GCM. (2015, Marzo) Wikipedia. [Online]. http://en.wikipedia.org/wiki/Google_Cloud_Messaging

[8] Google Maps. (2015, Marzo) Wikipedia. [Online]. http://es.wikipedia.org/wiki/Google_Maps

[9] Automatic Api Rest. (2015, Marzo) GeekyTheory. [Online]. https://geekytheory.com/automatic-api-rest/

[10] REST. (2015, Abril) Asiermarques. [Online]. http://asiermarques.com/2013/conceptos-sobre-apis-rest/

[11] PHP. (2015, Marzo) Wikipedia. [Online]. http://es.wikipedia.org/wiki/PHP

[12] MySQL. (2015, Marzo) Wikipedia. [Online]. http://es.wikipedia.org/wiki/MySQL

[13] EINA. (2014, Septiembre) Google Play. [Online]. https://play.google.com/store/apps/details?id=appinventor.ai_aitorgar14.eina

[14] UPV. (2014, Septiembre) Google Play. [Online]. https://play.google.com/store/apps/details?id=es.upv

[15] UGR. (2014, Septiembre) Google Play. [Online]. https://play.google.com/store/apps/details?id=com.proyectosUGR.ugr

[16] UCM. (2014, Septiembre) Google Play. [Online]. https://play.google.com/store/apps/details?id=es.ucm.ucmovil

[17] Universidad de Murcia. (2014, Septiembre) Google Play. [Online]. https://play.google.com/store/apps/details?id=com.um

[18] UviGO. (2014, Octubre) Google Play. [Online]. https://play.google.com/store/apps/details?id=es.uvigo.app

[19] UniOvi App. (2014, Octubre) Google Play. [Online]. https://play.google.com/store/apps/details?id=es.uniovi.innova.unioviapp

[20] IEEE, "IEEE Recommended Practice for Software Requirements Specifications," IEEE Computer Society, Estandar 830, 1998.

[21] Libro AIPO. (2015, Abril) AIPO. [Online]. http://aipo.es/content/libro-aipo

[22] Convenio colectivo estatal. (2015, Abril) BOE. [Online]. http://www.boe.es/boe/dias/2009/04/04/pdfs/BOE-A-2009-5688.pdf

Anexos

Anexo A: Requisitos

A.1. Requisitos funcionales de la Aplicación Android

Requisitos de interfaz

RQ_AND_01: La aplicación se desarrollará para dispositivos con pantalla de 4,8 pulgadas, con una resolución de 720 x 1280 píxeles. En el caso de dispositivos con pantalla más grande o más pequeña y una resolución menor, no se garantiza su correcta visualización.

RQ_AND_02: El icono que hace referencia a la aplicación en el menú del dispositivo Android será el logo de la Universidad de Zaragoza que se muestra en la Ilustración 25.

Ilustración 25. Icono de la aplicación

RQ_AND_03: Todas las pantallas de la aplicación tendrán el mismo "Look and Feel", al igual que una estructura homogénea. En la medida de lo posible, el usuario deberá poder efectuar su operación en 3 o menos movimientos.

RQ_AND_04: La navegación por la aplicación será sencilla y eficiente.

RQ_AND_05: El usuario deberá saber en todo momento, en qué sección de la aplicación se encuentra.

RQ_AND_06: La primera vez que se inicie la aplicación deberá aparecer una pantalla que pregunte al usuario si desea o no crear un perfil.

RQ_AND_07: La pantalla de crear perfil será un formulario que solicitará titulación y asignaturas.

RQ_AND_08: La pantalla de crear perfil tendrá un desplegable para buscar por curso y una caja de texto para poder buscar por el nombre.

RQ_AND_09: Habrá un botón que permitirá la cancelación de la creación del perfil.

RQ_AND_10: La aplicación mostrará las mismas opciones con o sin perfil.

RQ_AND_11: En la mayor parte de las pantallas deberá aparecer el botón del buscador en la barra del menú, excepto en las pantallas que visualicen una web, documentos, el mapa con los centros señalizados y la pantalla de creación del perfil, que no mostrarán ninguna opción.

RQ_AND_12: Para el desplazamiento por la aplicación, se contará con un menú desplegable, con las opciones:

- Noticias.
- Perfil.
- Centros.
- Servicios.
- Mapa.

RQ_AND_13: La opción "Perfil" tendrá las siguientes opciones:

- Moodle: Dará acceso a la web de Moodle.
- Calendario: Mostrará el calendario académico de la universidad.
- Asignaturas: Mostrará las asignaturas elegidas en el perfil
- Google Calendar: Dará acceso a las opciones a realizar con el calendario de Google.

RQ_AND_14: En la pantalla "Asignaturas" del perfil se visualizará en la barra del menú, además del buscador, un acceso rápido a la titulación elegida en el perfil y un botón para la modificación de las asignaturas.

RQ_AND_15: En caso de que no se haya creado el perfil, cuando se seleccione la opción "Asignaturas" aparecerá la pantalla de crear perfil.

RQ_AND_16: La opción "Google Calendar" del perfil permitirá las siguientes opciones:

- Importar: Añadirá los horarios de clases y exámenes de las asignaturas seleccionadas en el perfil, además de las fechas del calendario académico como el inicio de las vacaciones o los festivos.
- Eliminar: Eliminará los eventos añadidos por la aplicación en el calendario.
- Ajustes: Permitirá establecer si se desea recibir una notificación o no antes de los eventos.

RQ_AND_17: En el caso de que no exista el perfil, no se permitirá el acceso a la pantalla de "Google Calendar" al usuario.

RQ_AND_18: La pantalla principal de la aplicación será el listado de noticias del campus. Los centros y servicios serán también un listado.

RQ_AND_19: Para ver el contenido de alguno de los elementos listados, únicamente se debe pulsar sobre el mismo.

RQ_AND_20: Cada centro tendrá las siguientes opciones:

- Titulaciones.
- Contacto.
- Profesorado.
- Plano del centro.

RQ_AND_21: En caso de que no haya datos de alguna de las opciones, se mostrará un mensaje al usuario indicando que no hay datos disponibles.

RQ_AND_22: Las titulaciones mostrarán las siguientes opciones:

- Web titulaciones.
- Horario de clases.
- Horario de exámenes.

RQ_AND_23: La opción "Contacto" de un centro mostrará:

- Dirección.
- Teléfono.
- Fax.

RQ_AND_24: La opción "Profesorado" de un centro mostrará un listado con todos los profesores que estén relacionados con ese centro.

RQ_AND_25: En la información de un profesor se verá:

- Nombre.
- Teléfono.
- Correo.
- Ubicación del despacho.
- Tutorías.

RQ_AND_26: El mapa mostrará todos los centros del campus.

RQ_AND_27: Al entrar a la información de uno de los centros a través del mapa, se verá una foto del centro, los servicios y titulaciones que contiene.

RQ_AND_28: La pantalla descrita en el RQ_AND_27 deberá contener las siguientes opciones en la barra del menú:

- Información de contacto del centro.
- Como llegar al centro.
- Ver plano del centro.

RQ_AND_29: Al pulsar sobre un teléfono, aparecerá una ventana emergente preguntando si desea llamar.

RQ_AND_30: Al pulsar sobre un correo, permitirá redactar un correo a esa dirección.

RQ_AND_31: Al pulsar sobre una dirección o ubicación, aparecerá una ventana emergente con opciones para ubicar el centro, servicio, etc.

RQ_AND_32: En ningún momento habrá un desplazamiento en *scroll* horizontal.

RQ_AND_33: Será necesario pedir confirmación cuando se desee salir de la aplicación.

Requisitos de inicio de la aplicación

RQ_AND_34: Al iniciar comprobará si hay datos en la base de datos. Si no hay, y tampoco se dispone de conexión a Internet, la aplicación no dejará continuar.

RQ_AND_35: Si hay conexión hay Internet y no hay datos, la aplicación se comunicará con el servidor para obtener los datos.

RQ_AND_36: Para agilizar la descarga de datos, los datos referentes a las asignaturas de las aplicaciones se obtendrán del servidor cuando se acceda a la pantalla que permita crear el perfil de usuario, y no en el inicio con el resto de datos.

RQ_AND_37: Una vez recogidos los datos se guardará la fecha de la última modificación de la base de datos en las preferencias de la aplicación.

RQ_AND_38: Si no está registrado en GCM, se deberá obtener el ID del servidor GCM.

RQ_AND_39: Si se obtiene un ID, se mandará al servidor para que lo guarde en la base de datos.

RQ_AND_40: Se comprobara si hay nuevas noticias en el RSS del blog del campus.

Requisitos de Noticias

RQ_AND_41: Se mostrarán las noticias por orden cronológico, de la más reciente a la más antigua.

RQ_AND_42: Al ver el contenido de una noticia, se verá el título, la fecha y el cuerpo de la noticia.

Requisitos de Centros

RQ_AND_43: Se mostrarán los centros que posean al menos una titulación, es decir, el Vicerrectorado se omitirá.

Requisitos de comunicación con el servidor

RQ_AND_44: Se conectará con el servidor para indicarle el ID de GCM.

RQ_AND_45: La primera vez que se inicie la aplicación, se conectará con el servidor para recibir todos los datos.

RQ_AND_46: Cuando la aplicación reciba un mensaje desde el servidor GCM, deberá conectar con el servidor y comprobar si hay datos nuevos, y si los hay, actualizarlos en la base de datos local.

RQ_AND_47: Cuando la aplicación reciba un mensaje, indicando que ha habido cambios en el servidor, ésta obtendrá todos los datos posteriores a la última fecha de modificaciones que tendrá almacenada en las preferencias (RQ_AND_37). Cuando finalice la actualización, deberá actualizar la fecha de última modificación local.

Otros requisitos

RQ_AND_48: El buscador deberá buscar entre las noticias, titulaciones, centros, servicios, profesores y asignaturas del perfil, en caso de que se haya creado.

RQ_AND_49: Todos los datos que se muestran en forma de listado se deben sacar de su correspondiente tabla en la base de datos.

RQ_AND_50: La base de datos local deberá tener la misma estructura de tablas y nombres de atributos que la alojada en el servidor.

RQ_AND_51: El plano del centro estará incluido en la aplicación en formato PDF, al igual que la imagen de cada centro en formato JPG.

RQ_AND_52: Para acceder a los recursos mencionados en el requisito RQ_AND_51, deberá existir un campo en la base de datos en la tabla que guarde los datos de los centros llamado recursos y los nombres de los recursos de RQ_AND_51 serán los que se introduzcan en ese campo.

A.2. Requisitos no funcionales de la Aplicación Android

Requisitos de rendimiento

RQ_AND_53: Deberá poder gestionar una gran cantidad de información.

RQ_AND_54: La ejecución de la aplicación debe hacerse de una manera fluida.

Requisitos de entrega

RQ_AND_55: El plazo de entrega de la aplicación previsto es antes de finalizar el curso 2014-2015.

Requisitos de usabilidad

RQ_AND_56: La interfaz debe seguir los estándares de usabilidad.

RQ_AND_57: La aplicación del cliente debe ser clara y sencilla para el uso.

RQ_AND_58: Se deberán seguir los patrones de diseño de Android.

A.3. Requisitos funcionales de la Aplicación Web

Requisitos de interfaz

RQ_WEB_01: La navegación será eficiente y sencilla.

RQ_WEB_02: La estructura de la página será homogénea.

RQ_WEB_03: El usuario debe tener información de dónde se encuentra en todo momento.

RQ_WEB_04: Todas las páginas deben contener un enlace a la página de inicio.

RQ_WEB_05: Desde cualquier página debe salir la opción de abandonar la sesión.

RQ_WEB_06: La interfaz web poseerá una estructura balanceada.

RQ_WEB_07: Cuando algún tipo de información se muestre en forma de tabla deberá aparecer en la cabecera la zona en la que se encuentra.

RQ_WEB_08: A excepción de las páginas de autenticación y recuperar contraseña, no se podrá acceder a ninguna página de la aplicación sin haber superado con éxito la autenticación en la aplicación.

RQ_WEB_09: El menú principal de la aplicación estará en todo momento visible a excepción de las pantallas de autenticación y recuperar contraseña.

RQ_WEB_10: El menú principal constará de los siguientes elementos:

- Centros
- Servicios
- Enseñanzas
    - Titulaciones
    - Asignaturas
- Profesorado
    - Profesores
    - Asignaturas
- App
    - URLs
    - Calendario
- Usuarios
    - Añadir usuario
    - Modificar datos

RQ_WEB_11: Todas las opciones descritas en el requisito RQ_WEB_10, excepto las del apartado *Usuarios*, deberán cargar una página con una tabla que liste todos los datos del elemento correspondiente y botones que permitan la modificación o eliminación de una tupla de datos. También contendrá un botón para añadir un nuevo elemento.

Las opciones del apartado *Usuarios* deberán mostrar un formulario que permita añadir un nuevo usuario o modificar los datos del usuario actual.

RQ_WEB_12: Cuando se pulse el botón añadir, se cargará una página con un formulario para meter los datos correspondientes.

RQ_WEB_13: Los campos de los formularios tendrán una breve descripción o ayuda, para que el usuario sepa lo que debe introducir en ese campo.

RQ_WEB_14: Se necesitará confirmación cuando se modifique o elimine algún elemento.

Requisitos de autenticación

RQ_WEB_15: Para el acceso a la aplicación se deberá autenticar el usuario mediante usuario y contraseña.

RQ_WEB_16: Existirá un apartado que permita la modificación de los datos del usuario.

RQ_WEB_17: Existirá un formulario para añadir nuevos usuarios.

RQ_WEB_18: En todo momento se permitirá desconectar la sesión.

RQ_WEB_19: La aplicación se desconectará automáticamente en caso de no interactuar con ella en un tiempo de 10 minutos.

RQ_WEB_20: El usuario tendrá disponible un enlace en la ventana de *login* que le permita recuperar su contraseña. Para ello, deberá introducir su *login* y deberá recibir un correo, en la dirección de correo que se rellenó al añadirlo, con un enlace generado de forma aleatoria para modificar su contraseña.

RQ_WEB_21: El enlace para recuperar la contraseña solo estará disponible durante una hora. En caso de que pase este tiempo esa dirección pasará a ser una URL invalida.

A.4. Requisitos no funcionales de la Aplicación Web

Requisitos de seguridad

RQ_WEB_22: Únicamente accederán los usuarios dados de alta en el sistema.

A.5. Requisitos funcionales de la Aplicación Servidor

Requisitos generales

RQ_SER_01: El servidor deberá estar disponible las 24 horas del día 365 días al año.

RQ_SER_02: El sistema deberá poder manejar un número indeterminado de usuarios.

RQ_SER_03: El servidor se encargará de gestionar el almacenamiento de los identificadores de los dispositivos para poder mandar los mensajes a través de GCM.

Requisitos de la base de datos

RQ_SER_04: La base de datos a la que accederá la aplicación podrá estar en la misma máquina en la que está se esté ejecutando o en otra diferente.

RQ_SER_05: Se deberá usar MySQL.

RQ_SER_06: La base de datos estará construida siguiendo el esquema entidad-relación indicado en la documentación.

RQ_SER_07: Deberá contener toda la información que se mostrará en la base de datos.

RQ_SER_08: Las tablas que contengan información que deba estar en la aplicación contendrán un campo llamado “fechaModificacion”. Este campo contendrá la fecha de creación de la tupla o la de la última modificación.

RQ_SER_09: Existirá una tabla “Fecha_ultima_modificacion” que contendrá una única tupla con la última fecha en la que ha habido una modificación.

Requisitos de la comunicación con la aplicación cliente

RQ_SER_10: La aplicación del servidor deberá atender las peticiones realizadas por las aplicaciones cliente, a través de Automatic Api Rest.

A.6. Requisitos no funcionales de la Aplicación Servidor

Requisitos de rendimiento

RQ_SER_11: La aplicación permitirá gestionar un número indeterminado de peticiones.

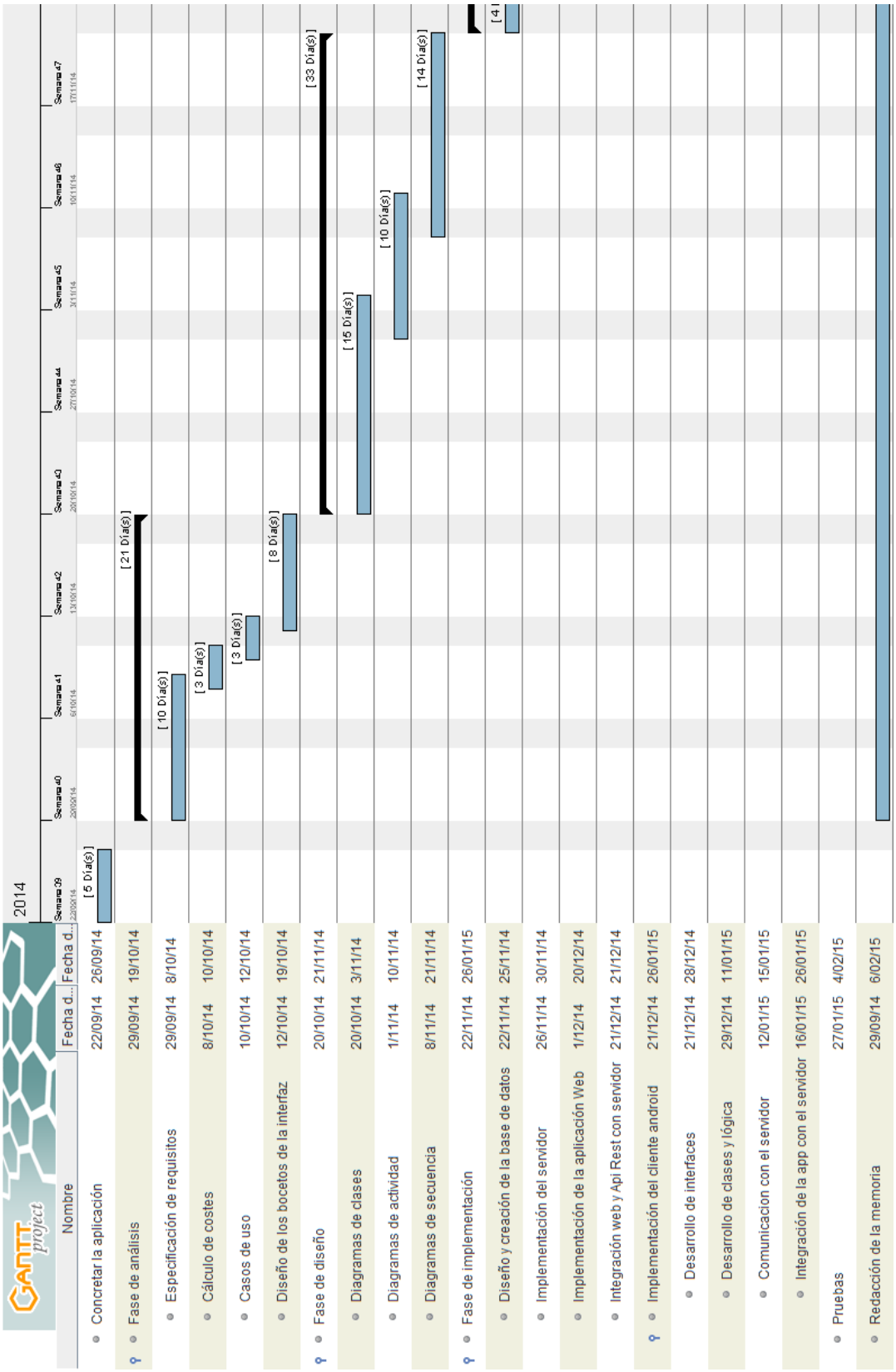
Requisitos de seguridad

RQ_SER_12: El acceso a la base de datos estará protegido mediante un usuario y contraseña, que no será conocida más que por los programadores.

RQ_SER_13: Las contraseñas que contenga la base de datos estarán cifradas. Únicamente son necesarias en la aplicación web.

Anexo B: Planificación

B.1. Planificación inicial



Nombre	Fecha d...	Fecha d...
Concretar la aplicación	22/09/14	26/09/14
Fase de análisis	29/09/14	19/10/14
Especificación de requisitos	29/09/14	8/10/14
Cálculo de costes	8/10/14	10/10/14
Casos de uso	10/10/14	12/10/14
Diseño de los bocetos de la interfaz	12/10/14	19/10/14
Fase de diseño	20/10/14	21/11/14
Diagramas de clases	20/10/14	3/11/14
Diagramas de actividad	1/11/14	10/11/14
Diagramas de secuencia	8/11/14	21/11/14
Fase de implementación	22/11/14	26/01/15
Diseño y creación de la base de datos	22/11/14	25/11/14
Implementación del servidor	26/11/14	30/11/14
Implementación de la aplicación Web	1/12/14	20/12/14
Integración web y Api Rest con servidor	21/12/14	21/12/14
Implementación del cliente android	21/12/14	26/01/15
Desarrollo de interfaces	21/12/14	28/12/14
Desarrollo de clases y lógica	29/12/14	11/01/15
Comunicacion con el servidor	12/01/15	15/01/15
Integración de la app con el servidor	16/01/15	26/01/15
Pruebas	27/01/15	4/02/15
Redacción de la memoria	29/09/14	6/02/15

Ilustración 26. Planificación inicial. Parte 1

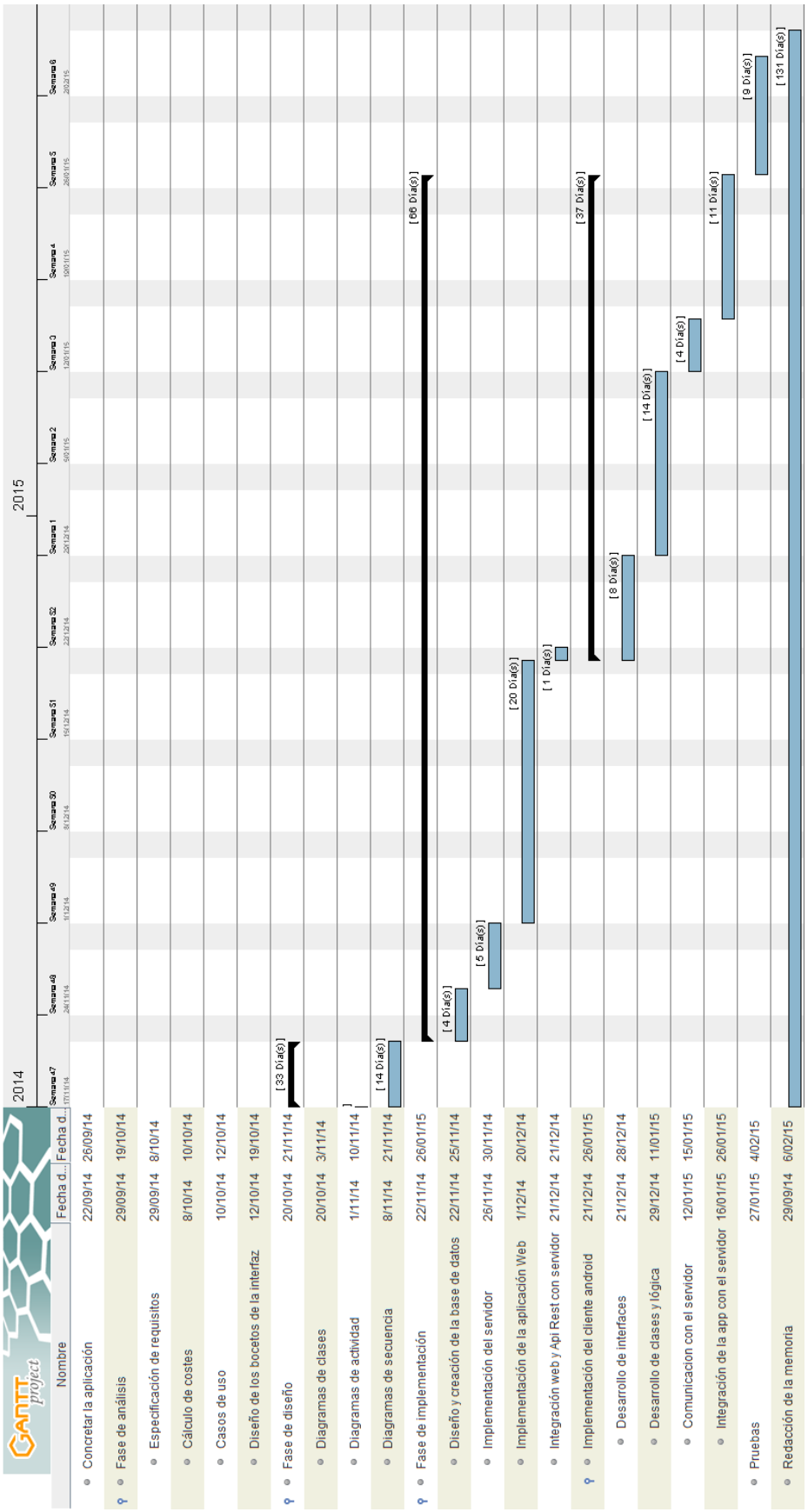


Ilustración 27. Planificación inicial. Parte 2

B.2. Planificación final

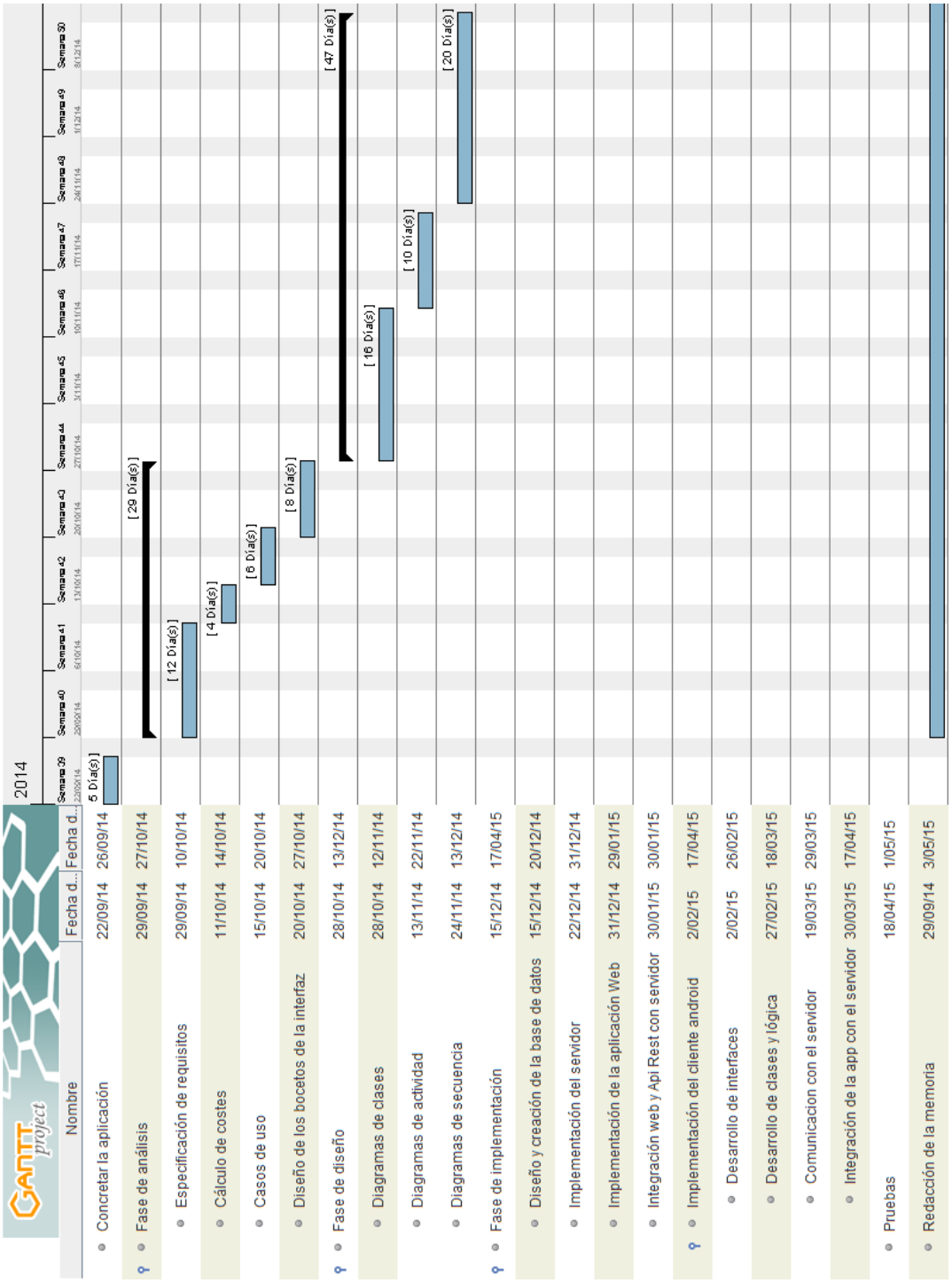


Ilustración 28. Planificación final. Parte 1.

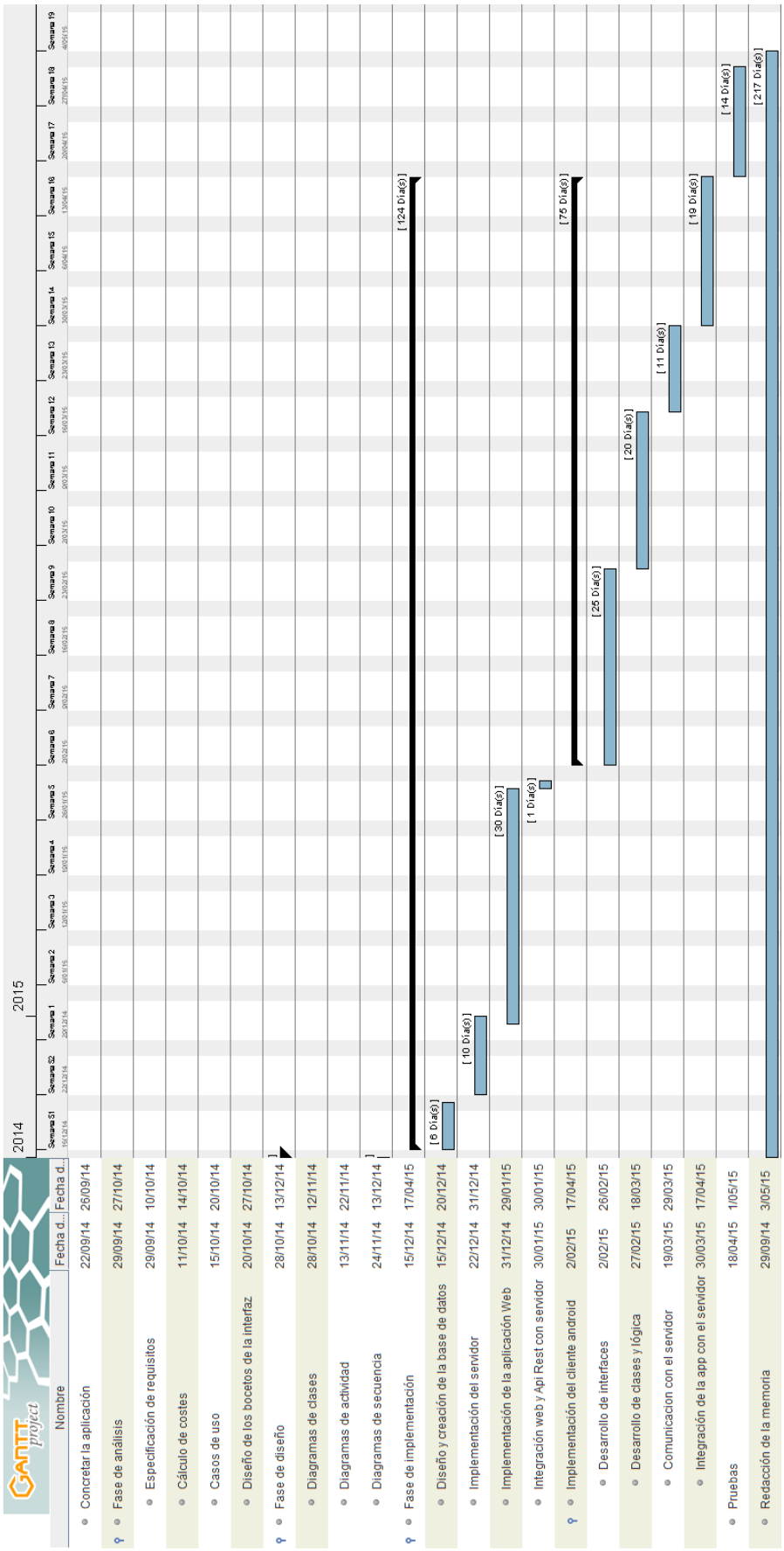


Ilustración 29. Planificación final. Parte 2.

Anexo C: Base de datos

C.1. Servidor-Web

Diseño lógico

```
Profesor (idProfesorinteger, nombre:varchar, telefono:integer,
ubicacionDespacho:varchar, Tutoria:varchar, correo:varchar,
fechaModificacion:datetime, fkCentro:integer )

     Clave Primaria { idProfesor }

     Valor No Nulo { nombre }

     Clave Ajena { fkCentro } hace referencia a Centro


Asignatura (idAsignaturainteger, nombre:varchar, curso:integer,
cuatrimestre:integer, fechaPrimeraConvocatoria:datetime,
aulaPrimeraConvocatoria:varchar,
fechaSegundaConvocatoria:datetime,
aulaSegundaConvocatoria:varchar, fechaModificacion:datetime,
codigo:integer, fkTitulacion:integer )

     Clave Primaria { idAsignatura }

     Valor No Nulo { nombre }

     Valor No Nulo { codigo }

     Clave Ajena { fkTitulacion } hace referencia a Titulacion


Titulacion (idTitulacioninteger, nombre:varchar,
horarioClasesPdf:varchar, horarioExamenesPdf:varchar,
siglas:varchar, webTitulaciones:varchar,
fechaModificacion:datetime, fkCentro:integer )

     Clave Primaria { idTitulacion }

     Valor No Nulo { nombre }

     Clave Ajena { fkCentro } hace referencia a Centro


Impartida (fkProfesorinteger, fkAsignaturainteger,
fechaModificacion:datetime )

     Clave Primaria { fkProfesor, fkAsignatura }

     Clave Ajena { fkProfesor } hace referencia a Profesor
```

Clave Ajena { fkAsignatura } hace referencia a Asignatura

Servicio (idServiciointeger, accesoWeb:varchar, correo:varchar, telefono:varchar, horario:varchar, ubicacion:varchar, nombre:varchar, fax:integer, fechaModificacion:datetime, fkCentro:integer)

Clave Primaria { idServicio }

Valor No Nulo { nombre }

Clave Ajena { fkCentro } hace referencia a Centro

Centro (idCentrointeger, nombre:varchar, direccion:varchar, telefono:integer, siglas:varchar, fax:integer, recursos:varchar, latitude:double, longitude:double, fechaModificacion:datetime)

Clave Primaria { idCentro }

Valor No Nulo { nombre }

Dispositivo (idDispositivointeger, gcm_regid:varchar, fechaCreacion:datetime)

Clave Primaria { idDispositivo }

Valor No Nulo { gcm_regid }

Usuario (loginvarchar, nombre:varchar, salt:varchar, password:varchar, correo:varchar)

Clave Primaria { login }

Valor No Nulo { salt }

Valor No Nulo { password }

Valor No Nulo { correo }

Horario (idHorariointeger, fkAsignaturainteger, tipoHorario:integer, aula:varchar, nombreGrupo:varchar, dia:integer, horaInicio:varchar, horaFin:varchar, fechaModificacion:datetime)

Clave Primaria { idHorario, fkAsignatura }

Clave Ajena { fkAsignatura } hace referencia a Asignatura

Reset (loginvarchar, url:varchar, hora:datetime)

Clave Primaria { login }

Clave Ajena { login } hace referencia a Usuario

Fecha_ultima_modificacion (idfechaUltimaModificacioninteger, fechaUltimaModificacion:datetime)

Clave Primaria { idfechaUltimaModificacion }

Historico_delete (idHistoricoDeletesinteger, sqlDelete:varchar, fechaModificacion:datetime)

Clave Primaria { idHistoricoDeletes }

Url (idUrlinteger, nombreUrl:varchar, url:varchar, fechaModificacion:datetime)

Clave Primaria { idUrl }

Calendario (idCalendariointeger, nombre:varchar, nombreFecha:varchar, fecha:date, fechaModificacion:datetime)

Clave Primaria { idCalendario }

Diseño físico

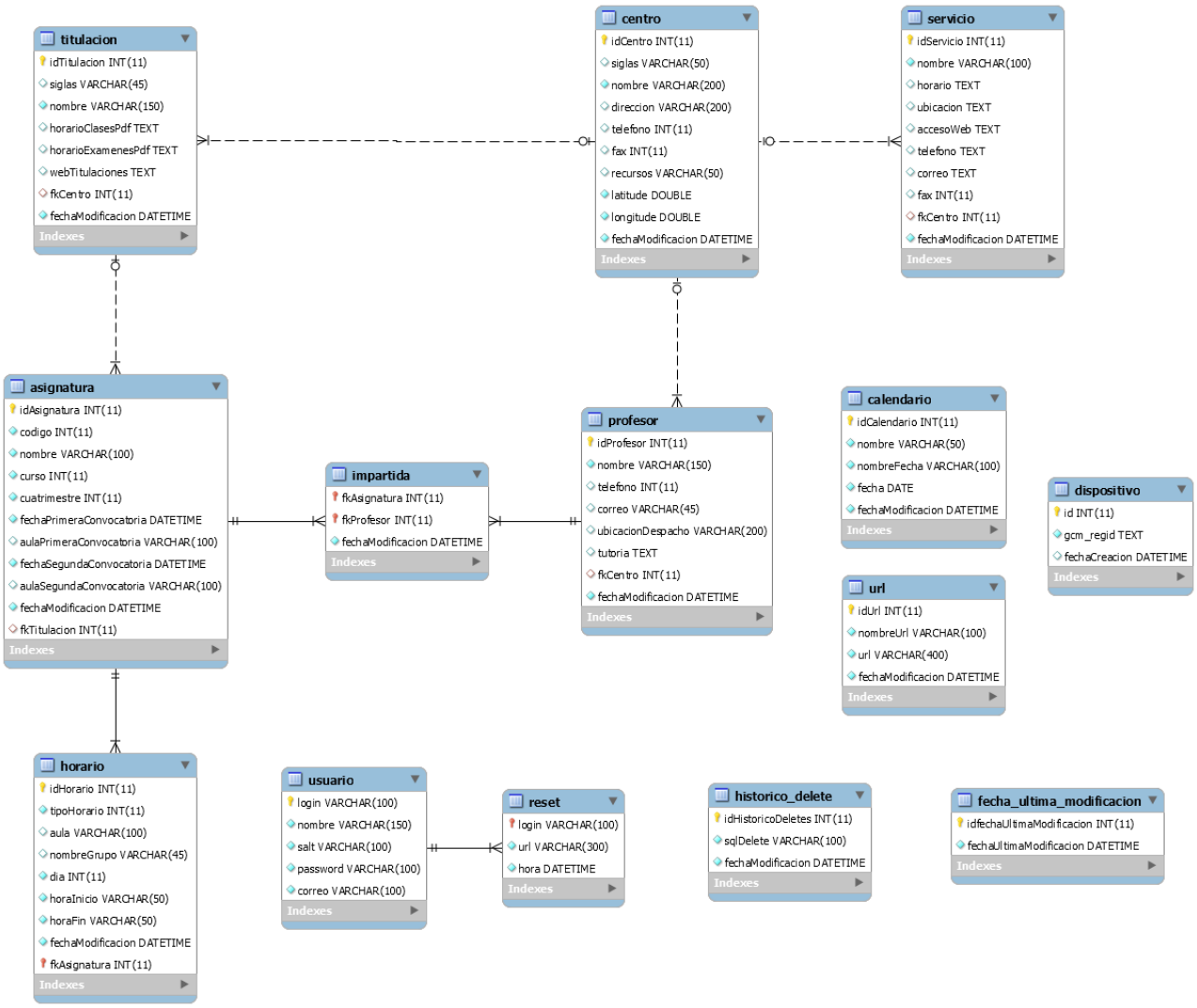


Ilustración 30. Tablas de la base de datos del servidor

Para la implementación de la base de datos se ha utilizado MySQL, y para facilitar la creación de la misma se ha utilizado el cliente MySQLWorkbench. La Ilustración 30 muestra el diseño final de las tablas, tal y como deberán estar en el servidor.

C.2. Cliente

Diseño lógico

Profesor (idProfesorinteger, nombre:varchar, telefono:integer, ubicacionDespacho:varchar, Tutoria:varchar, correo:varchar, fkCentro:integer)

Clave Primaria { idProfesor }

Valor No Nulo { nombre }

Clave Ajena { fkCentro } hace referencia a Centro

Asignatura (idAsignaturainteger, nombre:varchar, curso:integer, cuatrimestre:integer, fechaPrimeraConvocatoria:date, aulaPrimeraConvocatoria:varchar, fechaSegundaConvocatoria:date, aulaSegundaConvocatoria:varchar, codigo:integer, fkTitulacion:integer)

Clave Primaria { idAsignatura }

Valor No Nulo { nombre }

Valor No Nulo { codigo }

Clave Ajena { fkTitulacion } hace referencia a Titulacion

Titulacion (idTitulacioninteger, nombre:varchar, horarioClasesPdf:varchar, horarioExamenesPdf:varchar, siglas:varchar, webTitulaciones:varchar, fkCentro:integer)

Clave Primaria { idTitulacion }

Valor No Nulo { nombre }

Clave Ajena { fkCentro } hace referencia a Centro

Impartida (fkProfesorinteger, fkAsignaturainteger)

Clave Primaria { fkProfesor, fkAsignatura}

Clave Ajena { fkProfesor} hace referencia a Profesor

Clave Ajena { fkAsignatura} hace referencia a Asignatura

Servicio (idServiciointeger, accesoWeb:varchar, correo:varchar, telefono:varchar, horario:integer, ubicacion:varchar, nombre:varchar, fax:integer, fkCentro:integer)

Clave Primaria { idServicio }

Valor No Nulo { nombre }

Clave Ajena { fkCentro } hace referencia a Centro

Centro (idCentrointeger, nombre:varchar, direccion:varchar, telefono:integer, siglas:varchar, fax:integer, recursos:varchar, latitude:double, longitude:double)

Clave Primaria { idCentro }

Valor No Nulo { nombre }

Horario (idHorariointeger, fkAsignaturainteger, tipoHorario:integer, aula:varchar, nombreGrupo:varchar, dia:integer, horaInicio:varchar, horaFin:varchar)

Clave Primaria { idHorario, fkAsignatura}

Clave Ajena { fkAsignatura} hace referencia a Se_imparte

Url (idUrlinteger, nombreUrl:varchar, url:varchar)

Clave Primaria { idUrl }

Calendario (idCalendariointeger, nombre:varchar, nombreFecha:varchar, fecha:date)

Clave Primaria { idCalendario }

Noticia (idNoticiainteger, titulo:varchar, link:varchar, descripcion:varchar, guid:varchar, fecha:varchar, contenido:varchar)

Clave Primaria { idNoticia }

Diseño físico

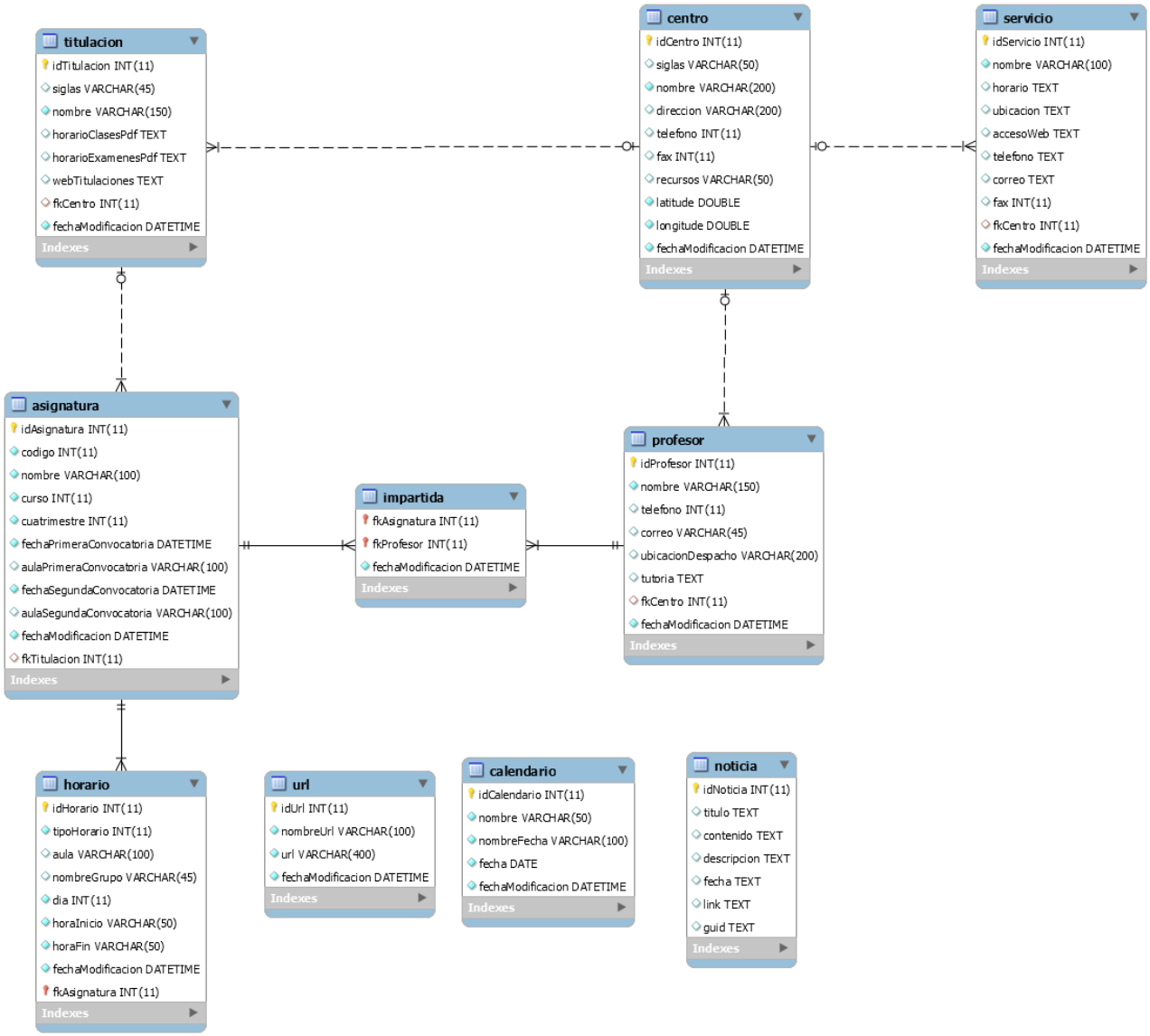


Ilustración 31. Tablas de la base de datos de la aplicación cliente

Para la implementación de la base de datos de la aplicación cliente, se ha utilizado SQLite. Para la creación de la misma se ha utilizado código Java mediante la clase `CampusTeruelDbHelper`. En la Ilustración 31 se pueden ver las tablas finales de la base de datos.

Anexo D: Prototipos

D.1. Prototipos Android

Pantalla Inicio

Ilustración 32. *Mockup* pantalla inicio

La Ilustración 32 muestra el prototipo de la pantalla de inicio. En ella aparecerá el icono de la aplicación, un texto breve para indicar al usuario si desea o no crear el perfil, y dos botones, uno más resaltado que otro para crear o no el perfil.

Pantalla Crear Perfil

Ilustración 33. *Mockup* pantalla crear perfil

Esta pantalla posibilita el crear un perfil con las asignaturas que el usuario seleccione, deberá elegir la carrera, y después podrá filtrar las asignaturas por curso o buscarlas por el nombre. Tendrá la opción de crear el perfil pulsando sobre el botón *Guardar,* o cancelar la creación del mismo pulsando sobre el botón *Cancelar.*

Pantalla Noticias

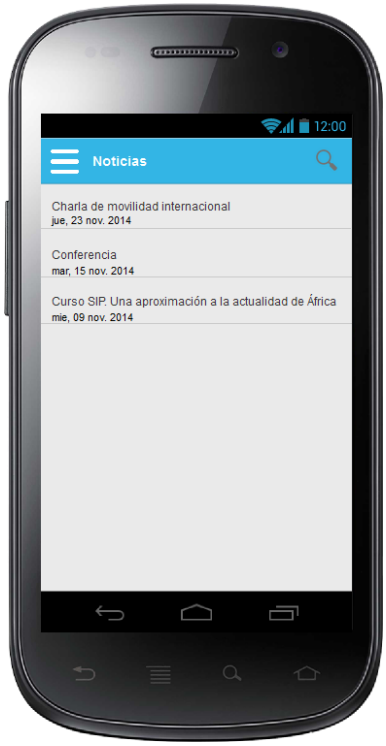


Ilustración 34. *Mockup* pantalla noticias

La pantalla *Noticias* (Ilustración 34) mostrará el listado de las noticias disponibles, además de la fecha de la noticia. También deberá estar ya disponible el buscador de la aplicación y el botón *burguericon* que indica que existe un menú *navigationdrawer* o menú lateral.

Pantalla Noticia

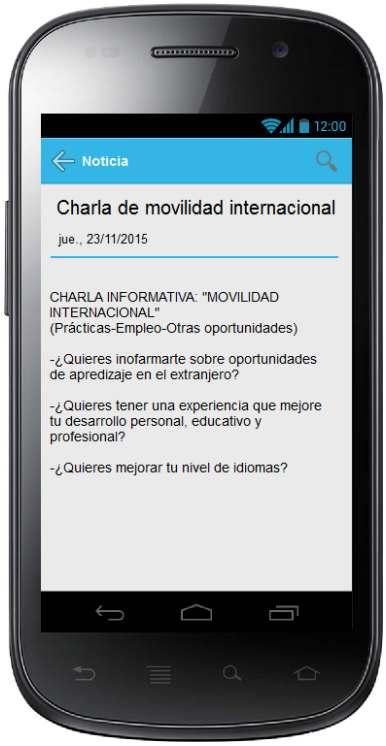


Ilustración 35. *Mockup* pantalla noticia

La pantalla que muestra el contenido de la noticia, deberá contener en la barra del menú, además del buscador de la aplicación, una flecha que indique al usuario que puede volver hacia atrás. La noticia tendrá tres partes, por un lado el título en grande y en el comienzo de la pantalla, la fecha de la noticia justo debajo del título, y por último el contenido de la noticia separado de la fecha y el título por una línea divisoria.

Menú NavigationDrawer

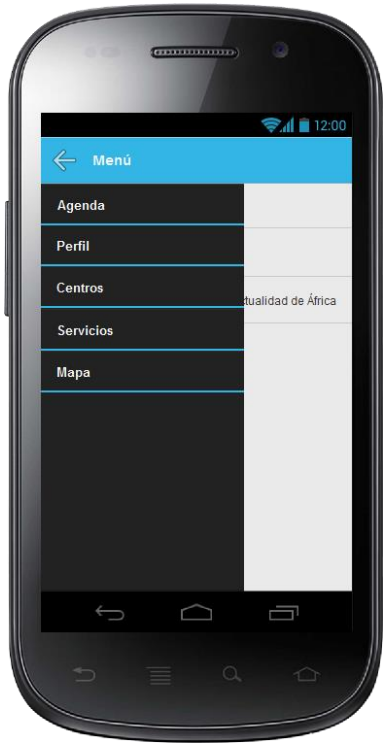


Ilustración 36. *Mockup* menú Navigation Drawer

El menú de la aplicación será como el mostrado en la ilustración, se creará con el elemento *navigationdrawer* de Android, de ahí el nombre elegido, y mostrará las opciones disponibles para moverse por la aplicación, indicadas en los requisitos.

Pantalla Perfil

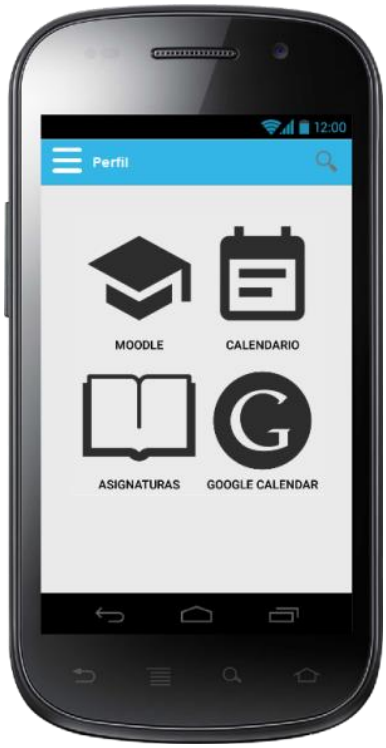


Ilustración 37. *Mockup* pantalla perfil

Este prototipo hace referencia a la pantalla que deberá verse cuando el usuario pulse sobre la opción *Perfil* del menú. Este menú mostrará cuatro opciones:

- Moodle: permite acceder a la web del anillo digital docente.
- Calendario: permite visualizar el calendario académico de la universidad de Zaragoza.
- Asignaturas: dará acceso a la *Pantalla Asignaturas* del perfil.
- Google Calendar: dará acceso a la *Pantalla Google Calendar*.

Pantalla Asignaturas

Ilustración 38. *Mockup* pantalla asignaturas

Este prototipo representa a la pantalla que listará las asignaturas del perfil, además del listado, y de tener el buscador. También tendrá un acceso para editar el perfil que irá a una pantalla idéntica a la de crear el perfil donde podrá volver a elegir la titulación y sus asignaturas, así como un botón que permita acceder de forma rápida a la titulación seleccionada en el perfil.

Pantalla Google Calendar

Ilustración 39. *Mockup* pantalla Google Calendar

En esta pantalla se pueden visualizar las diferentes opciones que hay para sincronizar el horario de las asignaturas del perfil con *Google Calendar*. Además de importar o eliminar el calendario, se puede acceder a la *Pantalla de ajustes* que permite añadir alertas a los eventos de Google.

Pantalla Ajustes de Google Calendar

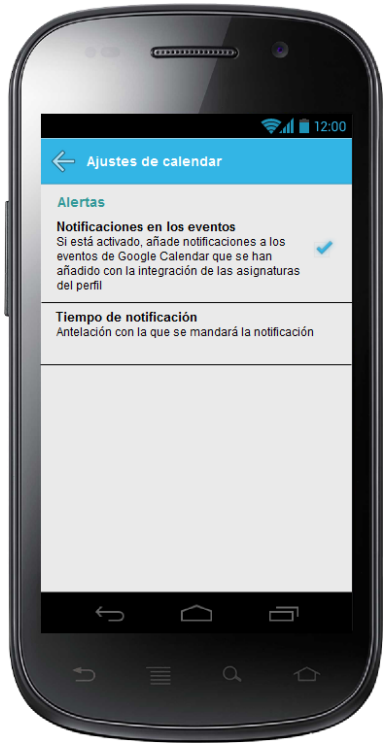


Ilustración 40. *Mockup* pantalla ajustes de Google Calendar

Como se puede ver en la ilustración, el usuario tendrá la opción de activar las alertas de los eventos y en la opción *tiempo de notificación* deberá elegir la antelación que desea para que le llegue la notificación.

Pantalla Asignatura

Ilustración 41. *Mockup* pantalla asignatura

La Ilustración 41 representa la pantalla de información de una asignatura. En ella se puede ver el nombre de la asignatura con su código, los horarios que tiene, las fechas y aulas de los exámenes y por último los profesores encargados de impartirla.

En el caso de los profesores, al pulsar sobre uno de ellos se accederá a la pantalla de información de ese profesor.

Pantalla Centros y Servicios

Ilustración 42. *Mockup* pantalla centros

El prototipo de la Ilustración 42 muestra el listado de los centros, al que se accede desde la opción del menú. Pulsando sobre uno de los centros se accederá a la *Pantalla Centro.*

La pantalla de servicios tendrá exactamente el mismo formato que la ilustración mostrada. Únicamente variará el título de la pantalla que será "Servicios" y el contenido, donde se mostrarán todos los servicios del campus. Al seleccionar sobre uno se irá a la *Pantalla Servicio* donde se podrá ver toda la información del servicio.

Pantalla Centro

Ilustración 43. *Mockup* pantalla centro

La pantalla de un centro deberá mostrar el menú que se ve en la Ilustración 43, para así tener acceso a las titulaciones del centro, información de contacto, profesorado y un plano en PDF donde se podrán ver los servicios, aulas y despachos de los profesores del centro.

Pantalla Contacto

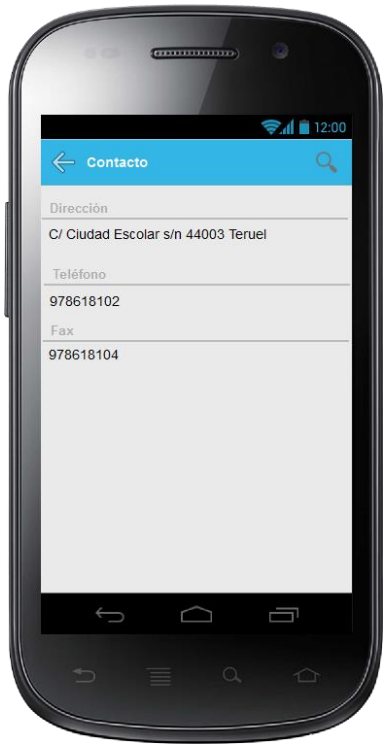


Ilustración 44. *Mockup* pantalla contacto

En esta pantalla se verá la información de contacto del centro que se había seleccionado, es decir, la dirección, el teléfono y el fax.

Pantalla Titulaciones y Profesorado

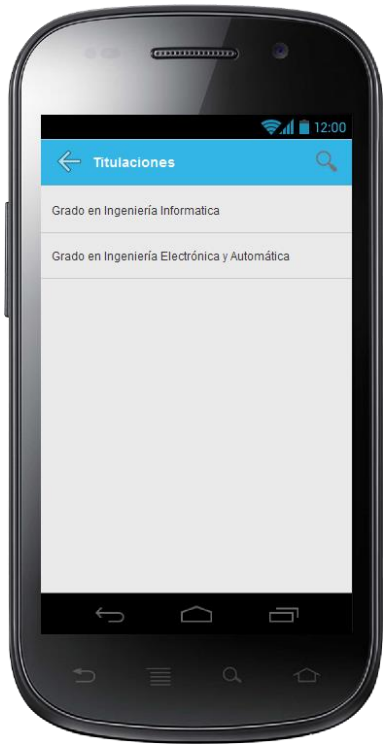


Ilustración 45. *Mockup* pantalla titulaciones

En la pantalla de la Ilustración 45 se ven en un listado todas las titulaciones pertenecientes a un centro. Al pulsar en una de ellas se va a la *Pantalla Titulación* donde se puede acceder a la información del centro.

De forma similar a los centros y los servicios, la pantalla del profesorado del centro debe ser exactamente igual a la de titulaciones variando el contenido y el título a "Profesores". El botón de volver, el buscador y el formato de letra será exactamente lo mismo.

Pantalla Titulación

Ilustración 46. *Mockup* pantalla titulación

Este prototipo muestra el menú que deberá ver el usuario cuando pulse sobre una titulación. Con esos tres botones podrá acceder a ver la Web de titulaciones, así como a los documentos PDF con los horarios de clases y exámenes.

Pantalla Profesor

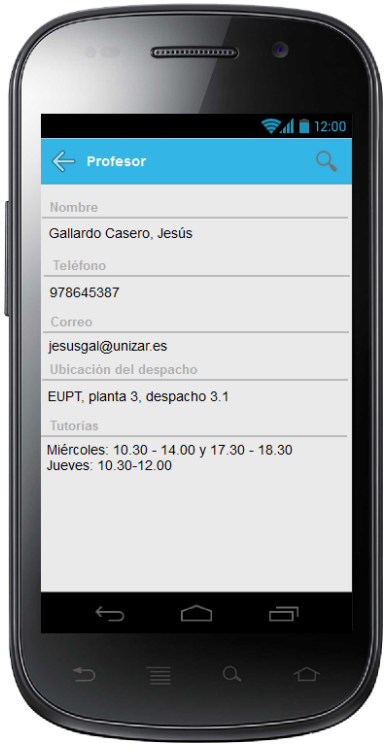


Ilustración 47. *Mockup* pantalla profesor

La Ilustración 47 muestra cómo se visualizará toda la información de un profesor; se mostrará información de contacto como su teléfono o correo, la ubicación de su despacho y el horario de tutorías que tiene.

Pantalla Servicio

Ilustración 48. *Mockup* pantalla servicio

Esta pantalla es similar a la anterior, solo que en ésta se muestran los campos de información referentes a un servicio del campus.

Pantalla Mapa

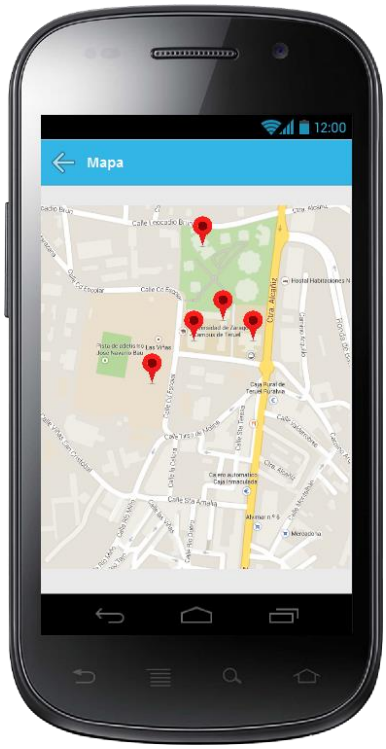


Ilustración 49. *Mockup* pantalla mapa

La Ilustración muestra cómo será la pantalla mapa, que señalará la localización de los centros del campus. Además, si se pulsa sobre uno de los centros se irá a la *Pantalla Centro Mapa* para ver la información que incluyen.

Pantalla Centro Mapa

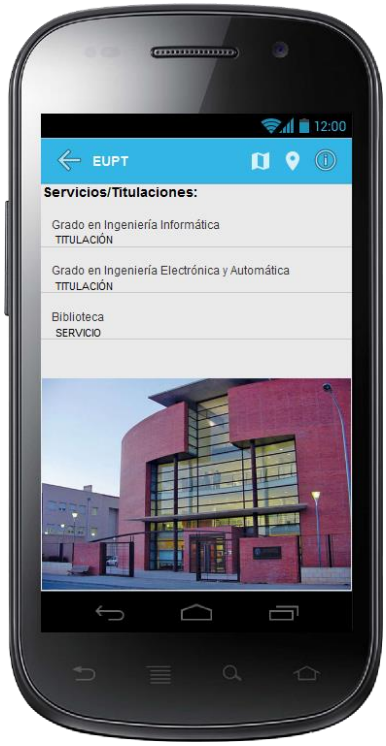


Ilustración 50. *Mockup* pantalla centro mapa

Esta pantalla sirve para ver de un solo vistazo los contenidos, en cuanto a servicios y titulaciones se refiere, de un centro. Para su identificación se mostrará una imagen del centro y se podría acceder a la información de contacto, al plano del centro, o se podría ver cómo llegar a través de *Google Maps*. Estas acciones se realizarán con los botones que se muestran en la barra del menú.

D.2. Prototipos web

Debido a que la mayor parte de las pantallas de la aplicación web son similares y únicamente varían en el contenido que muestran y en los campos de los formularios, que pueden ser más o menos dependiendo del tipo de datos que se vaya a introducir, no se han realizado los prototipos de todas las pantallas.

Pantalla Login

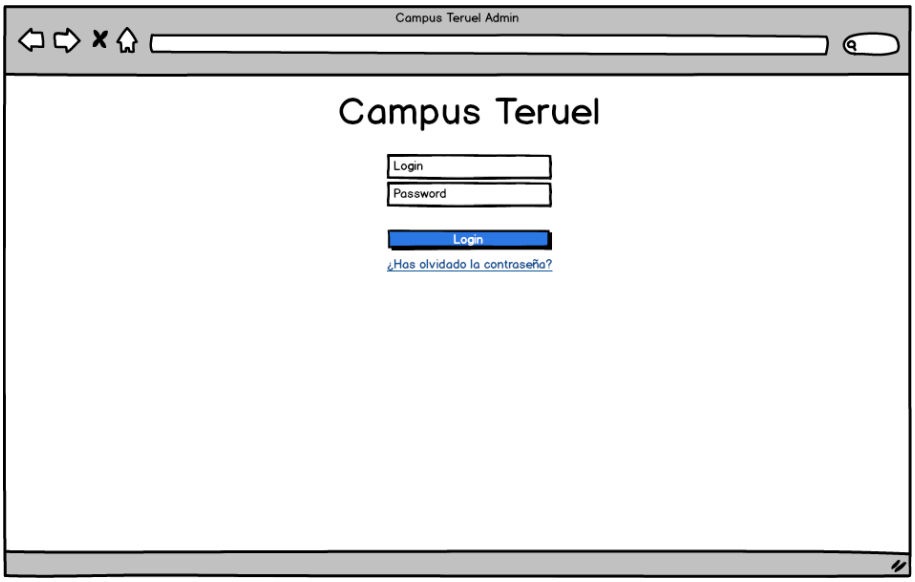


Ilustración 51. *Mockup* pantalla *login*

La Ilustración 51 muestra la *Pantalla Login* que básicamente será un formulario con dos campos, un botón y un enlace web. En los campos se deberán introducir el *login* del usuario y la contraseña que están almacenados en la base de datos del sistema. El enlace se muestra para que en caso de que el usuario no recordara su contraseña tener la posibilidad de restaurarla y así poder entrar.

Pantalla Recuperar Contraseña

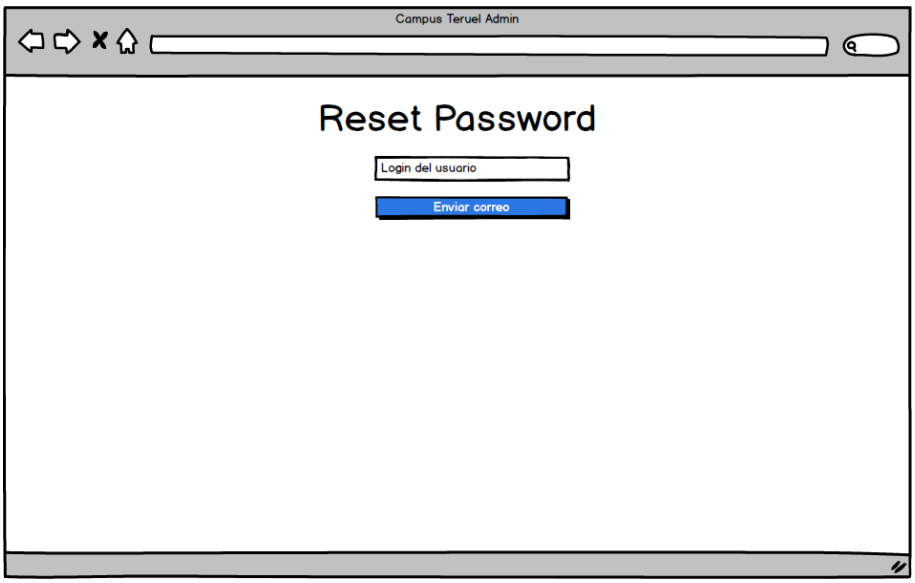


Ilustración 52. *Mockup* pantalla recuperar contraseña

La Ilustración 52 muestra la pantalla que el usuario verá en caso de necesitar recuperar su contraseña. Únicamente se le solicitará su *login* y se le enviará un correo a la dirección de correo perteneciente a esa cuenta, si es que existe.

Pantalla Restaurar Contraseña

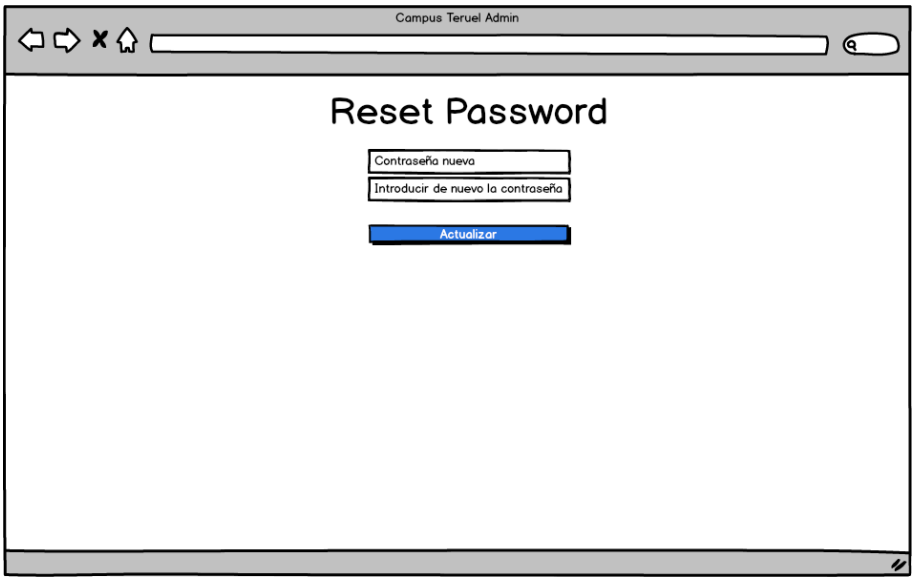


Ilustración 53. *Mockup* pantalla restaurar contraseña

El prototipo de la Ilustración 53 muestra lo que verá el usuario cuando pulse sobre el enlace de recuperación del correo. Únicamente deberá introducir la nueva contraseña dos veces.

Pantalla Centros

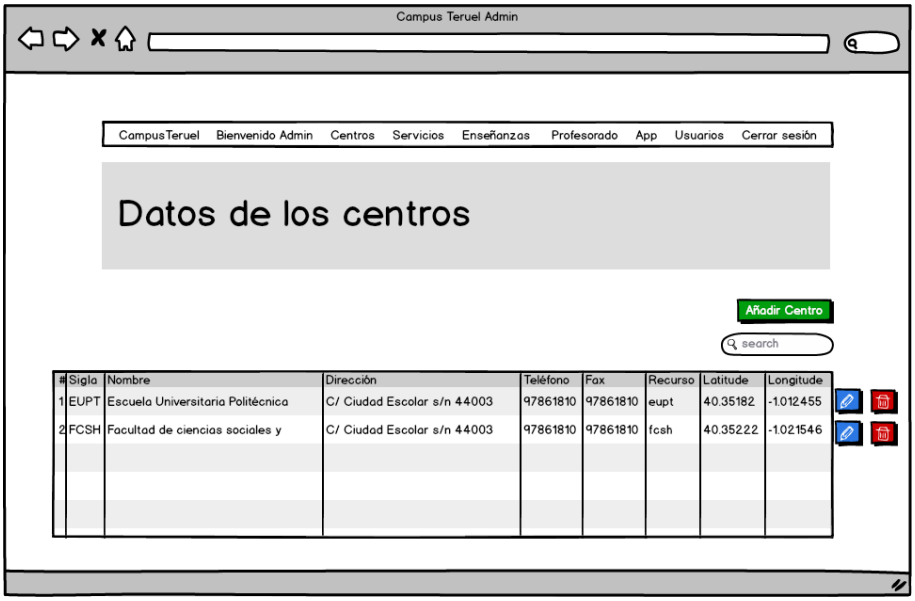


Ilustración 54. *Mockup* pantalla centros

En la Ilustración 54 se ve cómo se organizará la información para que la visualice el usuario. En el centro tendrá una tabla con los datos que haya solicitado ver, en cada fila tendrá un botón de editar y de eliminar, y en cada pantalla habrá un botón de añadir que llevará al usuario al formulario para añadir los datos. Ese mismo formulario será utilizado para editar los datos. Habrá un buscador para facilitar al administrador el encontrar los datos y justo debajo del menú habrá un texto que indique al usuario lo que está visualizando.

Todas las pantallas en las que se visualicen datos tendrán este mismo formato. Únicamente variará su contenido, exceptuando la pantalla donde se verán las relaciones entre profesores y asignaturas, que solo permitirá añadir y borrar relaciones, no editarlas.

Pantalla Formulario Centros

Ilustración 55. *Mockup* pantalla formulario centros

La pantalla de la Ilustración 55 muestra el formato que tendrán los formularios para añadir o editar nuevos datos. Cada formulario podrá tener más o menos campos dependiendo del tipo de datos que se vaya a añadir, pero todos tendrán este formato, salvo para añadir nuevas relaciones entre profesores y asignaturas, que se muestra en el siguiente prototipo.

Pantalla Formulario Profesores_Asignaturas

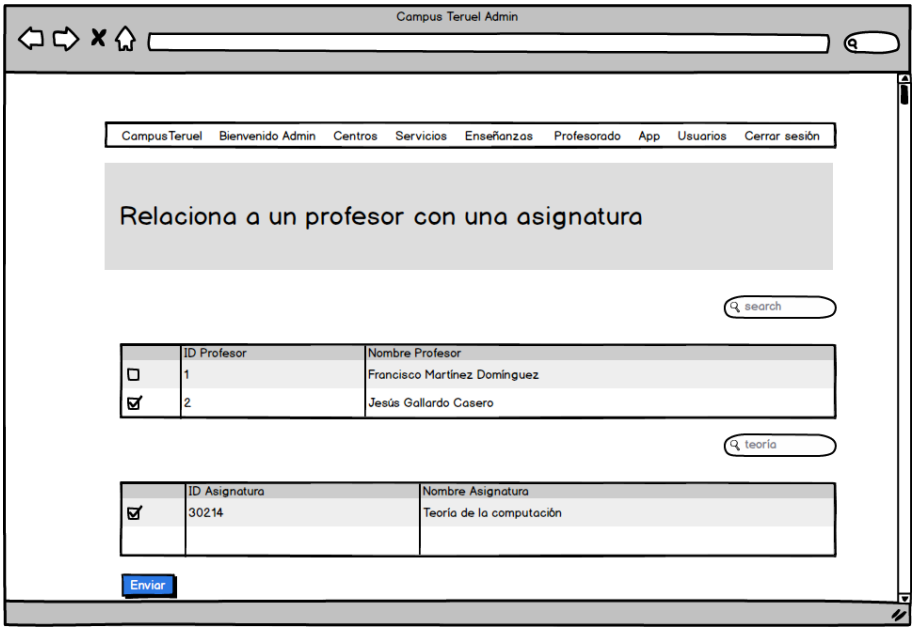


Ilustración 56. *Mockup* pantalla profesores_asignaturas

El prototipo de la Ilustración 56 muestra cómo se añade una nueva relación entre profesores y asignaturas. El usuario deberá elegir un profesor y una asignatura de las tablas y después enviar los datos. Para mayor rapidez se incluirán buscadores para ambas tablas.

Anexo E: Diagramas de clases

E.1. Aplicación cliente

Noticia

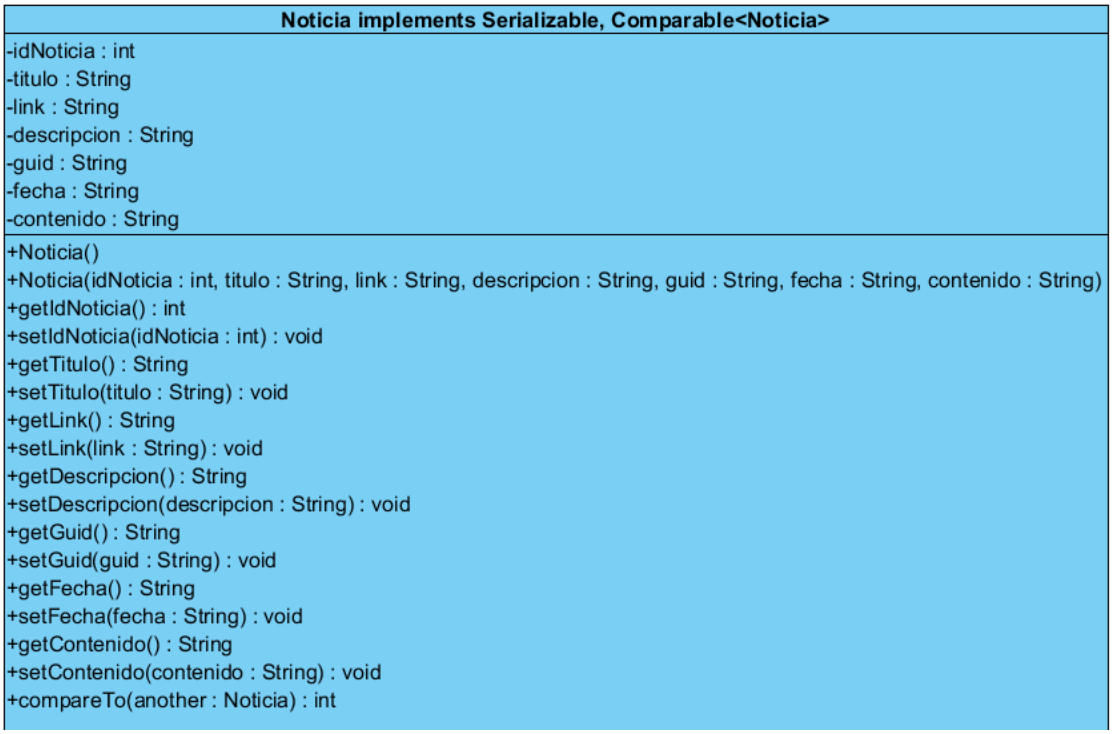


Ilustración 57. Clase Noticia

Esta clase permitirá manejar los datos relativos a las noticias del campus. En concreto, contiene los atributos relacionados con las noticias, dos constructores, los métodos `get/set` para los atributos y el método `compareTo` que implementa la interfaz `Comparable<Noticia>` y que está redefinido para que ordene las noticias por fecha. Además, implementa `Serializable` para poder pasar el objeto entre *activities.*

Horario

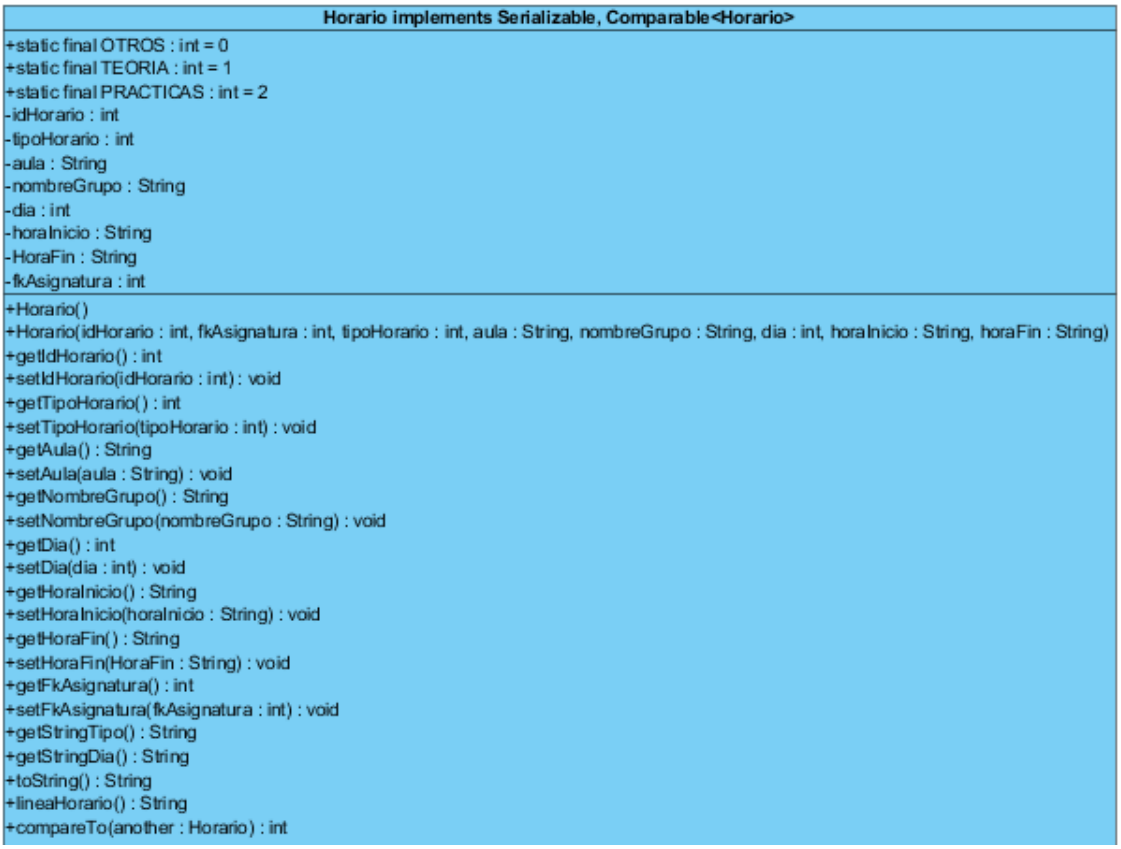


Ilustración 58. Clase Horario

Esta clase representa un horario de una asignatura y además de contener las variables relativas a los horarios, incluyendo tres constantes para distinguir el tipo de horario (`OTROS`, `TEORIA` o `PRACTICAS`).

Aparte de los constructores, los métodos `get/set`, `toString` y `compareTo`, existen tres métodos que añaden funcionalidad a la clase:

- `getStringTipo`: devuelve un `String` con el tipo de texto que es.
- `getStringDia`: devuelve un `String` con el nombre del día al que pertenece el horario.
- `lineaHorario`: devuelve un `String` con la información del horario lista para mostrarlo en la aplicación.

Como en la clase anterior, implementa `Serializable` y `Comparable`, permitiendo mandar objetos entre *activities* y ordenar en este caso los horarios.

Asignatura

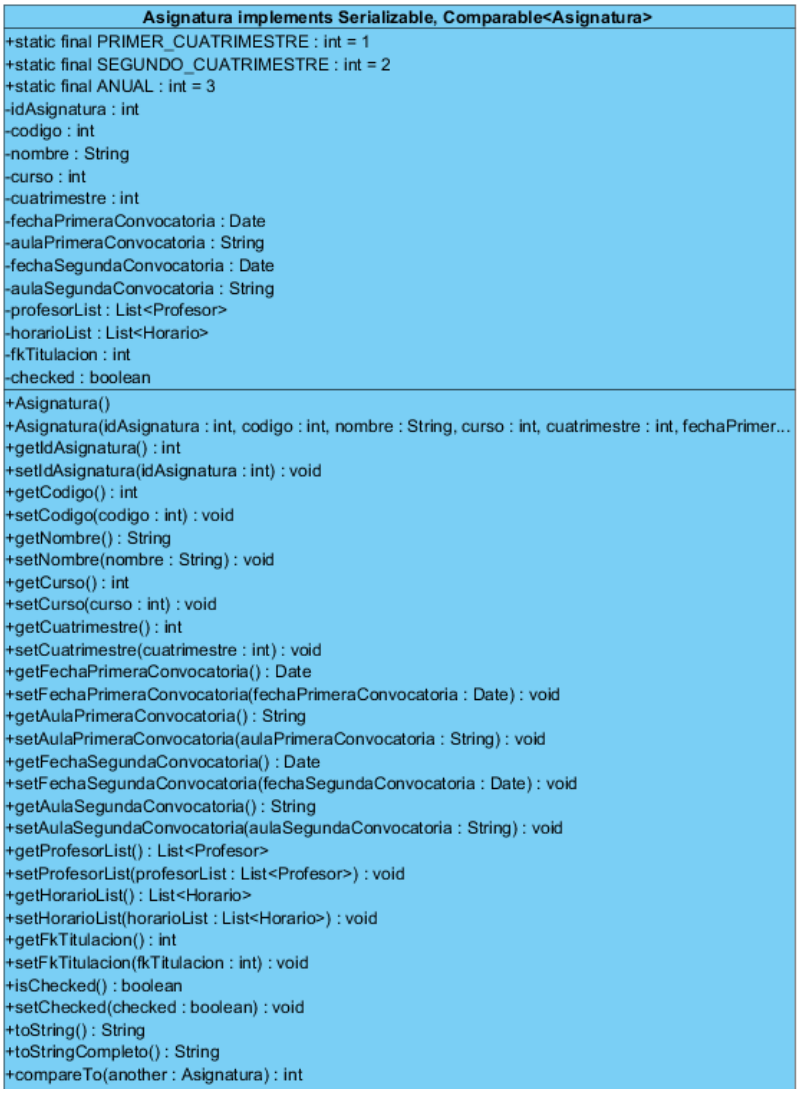


Ilustración 59. Clase Asignatura

Clase que contiene todos los datos relativos a las asignaturas. Contiene las constantes `PRIMER_CUATRIMESTRE`, `SEGUNDO_CUATRIMESTRE` y `ANUAL` para identificar los diferentes periodos en que se puede impartir una asignatura.

Contiene el método `toString` que retorna el nombre de la asignatura y `toStringCompleto` que retorna en una cadena todos los datos del objeto.

El segundo constructor no llega a mostrarse entero en la ilustración. La información completa es la siguiente:

```
Asignatura(idAsignatura : int, codigo : int, nombre : String, curso : int, cuatrimestre : int, fechaPrimeraConvocatoria : Date, aulaPrimeraConvocatoria : String, fechaSegundaConvocatoria : Date, aulaSegundaConvocatoria : String, fkTitulacion : int)
```

AsignaturaCursoComparator

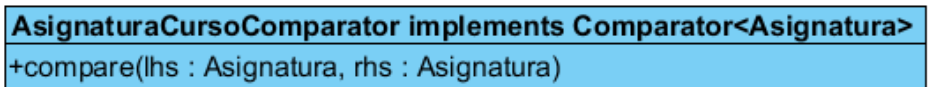


Ilustración 60. Clase AsignaturaCursoComparator

Esta clase implementa la interfaz `Comparator`, por lo que implementa el método `compare` que está sobrescrito para que ordene las asignaturas por el curso, dentro del curso por cuatrimestre, y dentro del cuatrimestre por el nombre.

Titulación

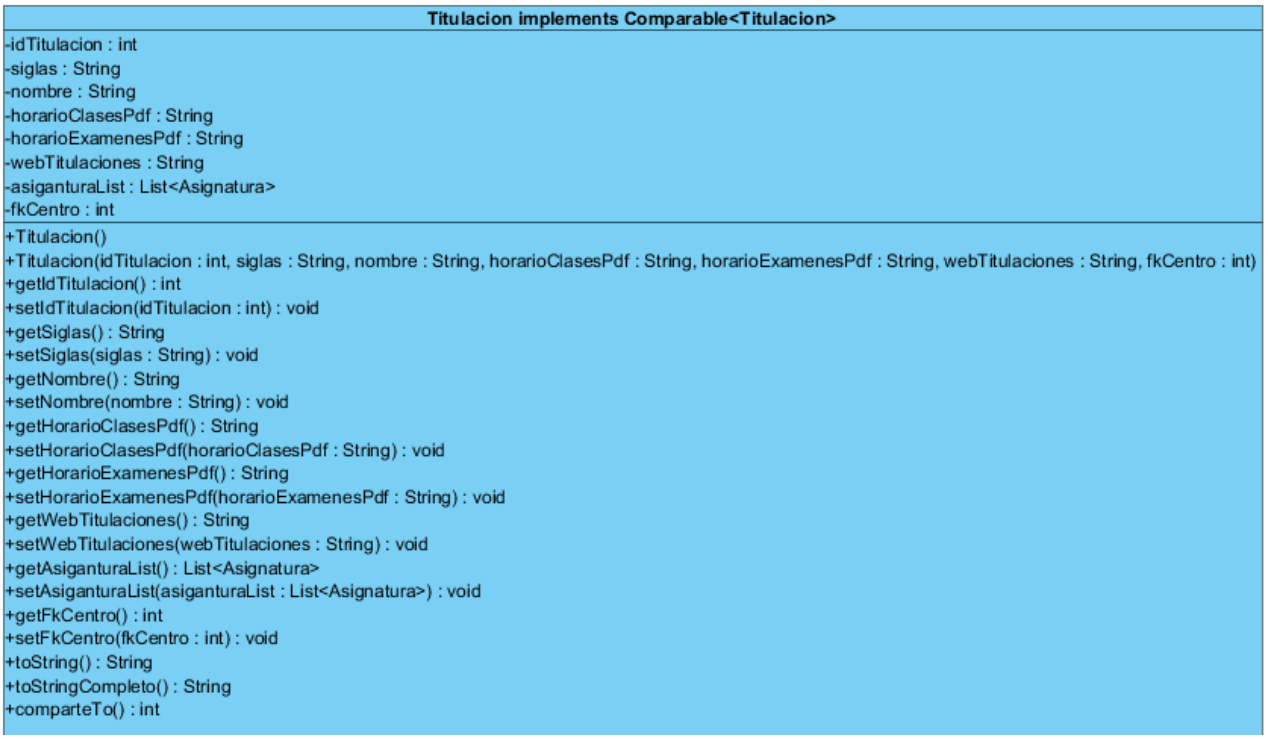


Ilustración 61. Clase Titulación

Clase que contiene todos los datos relativos a las titulaciones, con los métodos `get/set, toString y compareTo.`

Centro

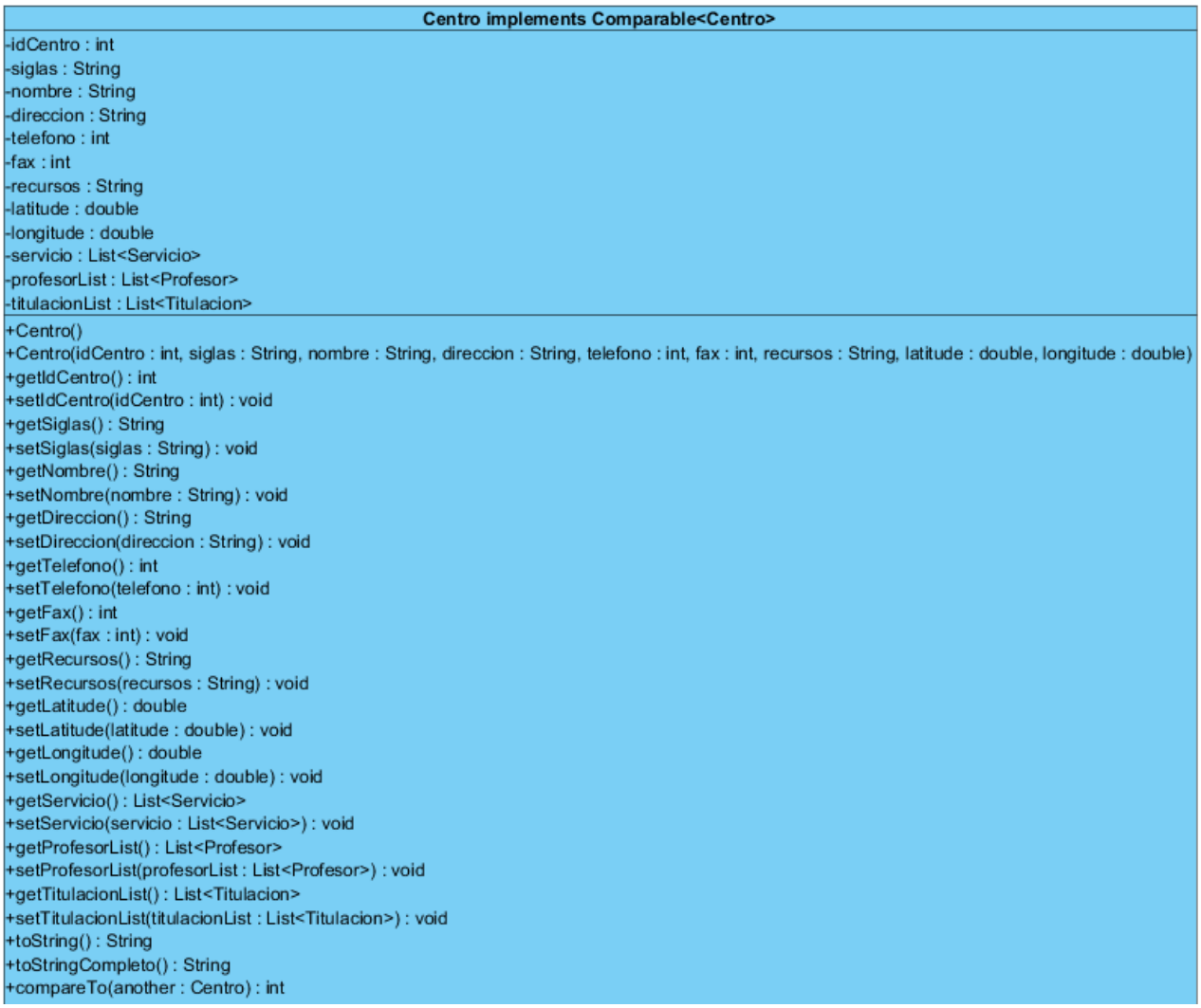


Ilustración 62. Clase Centro

Clase que contiene todos los datos relativos a los centros, con los métodos `get/set`, `toString` y `compareTo`.

Servicio

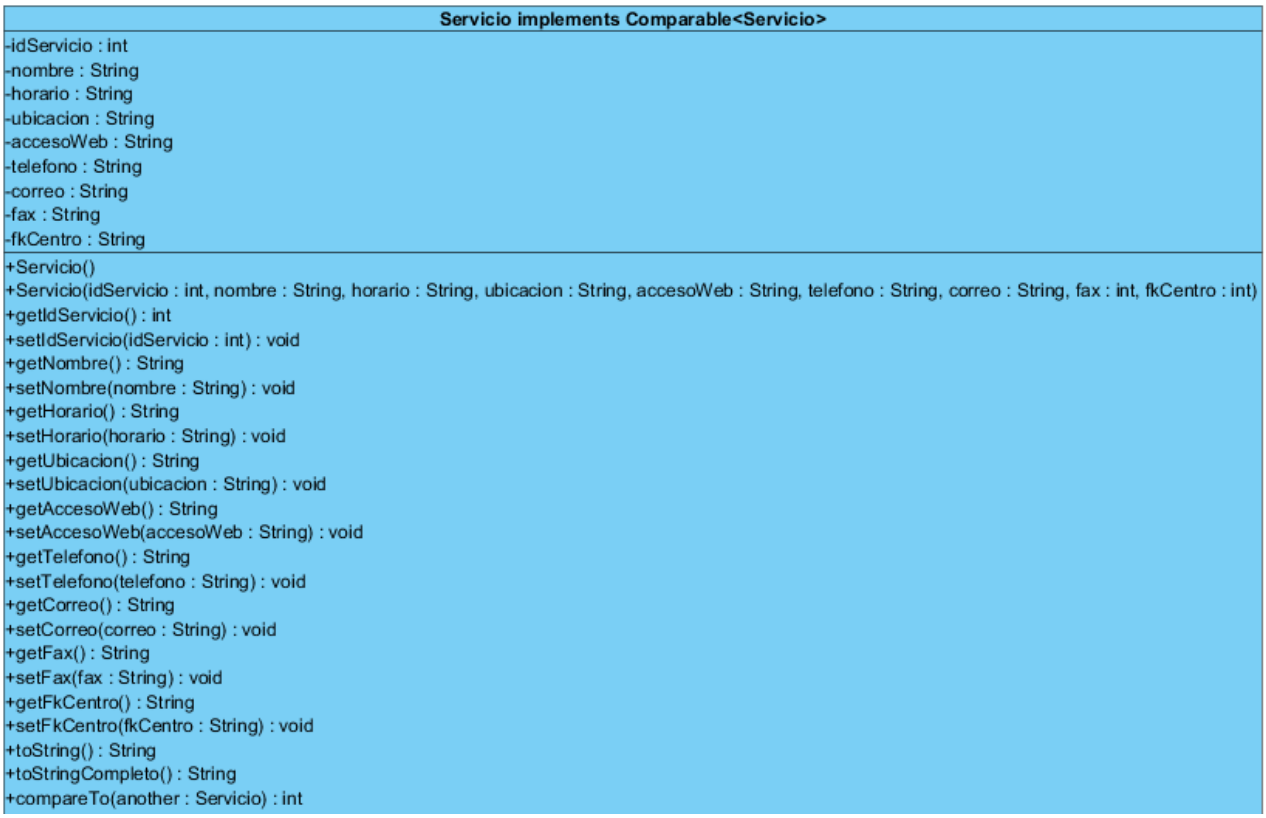


Ilustración 63. Clase Servicio

Clase que contiene todos los datos relativos a los servicios, con los métodos `get/set, toString y compareTo.`

Profesor

Profesor implements Serializable, Comparable<Profesor>
-idProfesor : int -nombre : String -telefono : String -correo : String -ubicacionDespacho : String -tutoria : String -asignaturaList : List<Asignatura> -fkCentro : int
+Profesor() +Profesor(idProfesor : int, nombre : String, telefono : int, correo : String, ubicacionDespacho : String, tutoria : String, fkCentro : int) +getIdProfesor() : int +setIdProfesor(idProfesor : int) : void +getNombre() : String +setNombre(nombre : String) : void +getTelefono() : String +setTelefono(telefono : String) : void +getCorreo() : String +setCorreo(correo : String) : void +getUbicacionDespacho() : String +setUbicacionDespacho(ubicacionDespacho : String) : void +getTutoria() : String +setTutoria(tutoria : String) : void +getAsignaturaList() : List<Asignatura> +setAsignaturaList(asignaturaList : List<Asignatura>) : void +getFkCentro() : int +setFkCentro(fkCentro : int) : void +toString() : String +toStringCompleto() : String +compareTo(another : Profesor) : int

Ilustración 64. Clase Profesor

Clase que contiene todos los datos relativos a los profesores, con los métodos `get/set, toString y compareTo.`

Calendario

Calendario implements Serializable, Comparable<Calendario>
-idCalendario : int -nombre : String -nombreFecha : String -fecha : Date
+Calendario() +Calendario(idCalendario : int, nombre : String, nombreFecha : String, fecha : Date) +getIdCalendario() : int +setIdCalendario(idCalendario : int) : void +getNombre() : String +setNombre(nombre : String) : void +getNombreFecha() : String +setNombreFecha(nombreFecha : String) : void +getFecha() : Date +setFecha(fecha : Date) : void +toString() : String +compareTo(another : Calendario)

Ilustración 65. Clase Calendario

Clase que contiene todos los datos relativos al calendario, con los métodos `get/set, toString y compareTo.`

Url

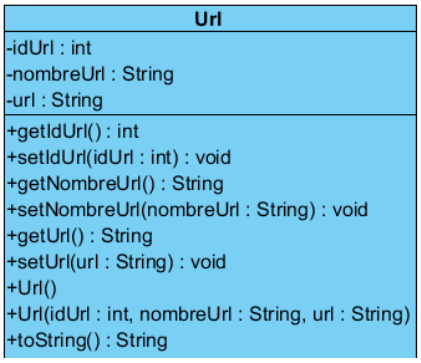


Ilustración 66. Clase Url

Clase que contiene todos los datos relativos a las URL utilizadas por la aplicación, con los métodos `get/set y toString`.

Impartida

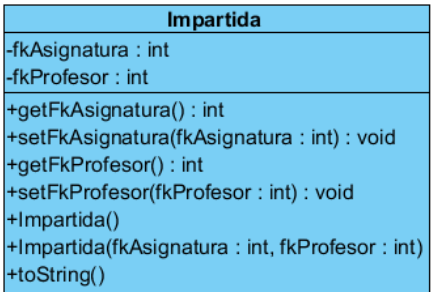


Ilustración 67. Clase Impartida

Clase que contiene todos los datos relativos a las relaciones entre una asignatura y el profesor que la imparte, con los métodos `get/set y toString`.

ResultadoBuscador

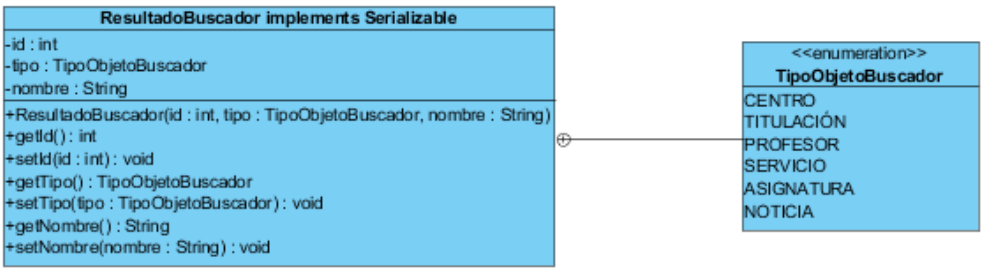


Ilustración 68. Clase ResultadoBuscador

La ilustración anterior muestra la clase que se utiliza en el buscador para recoger los identificadores y nombres de los elementos se encuentran en una búsqueda. Esta clase contiene además un tipo enumerado con los diferentes tipos que se pueden buscar en la aplicación.

UniversalDAO

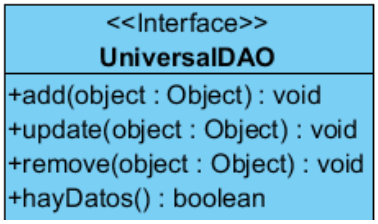


Ilustración 69. Interface UniversalDAO

Es la interfaz que heredan todas las interfaces que controlan los métodos que permiten tener acceso a la base de datos, contiene los métodos genéricos para añadir, modificar o eliminar cualquier objeto de la base de datos y también incluye el método hay datos para saber si existen datos en la interfaz que la implemente.

NoticiaDAO

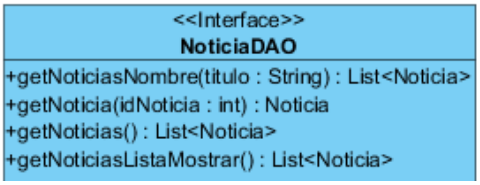


Ilustración 70. Interface NoticiaDAO

La interfaz `NoticiaDAO` contiene los métodos que deberá definir la clase encargada de la gestión de las noticias. Los métodos definidos se encargan de devolver una lista de noticias o una noticia concreta dependiendo del método. La causa de que el método `getNoticiasNombre` devuelva una lista es que busca que los títulos de las noticias contengan la cadena introducida por lo que puede haber más de una coincidencia y por tanto más de una noticia, y el método `getNoticiasListaMostrar` devuelve la lista de noticias que se mostraran en la pantalla que muestra el listado de noticias.

HorarioDAO

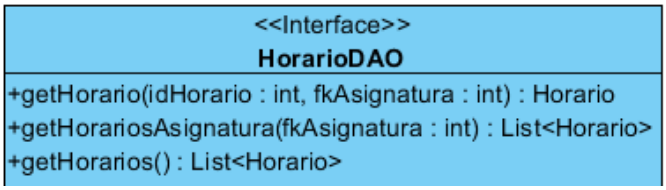


Ilustración 71. Interface HorarioDAO

La interfaz `HorarioDAO` contiene los métodos que deberá definir la clase encargada de la gestión de los horarios de las asignaturas. Los métodos se encargan de retornar una lista de horarios o un horario concreto.

AsignaturaDAO

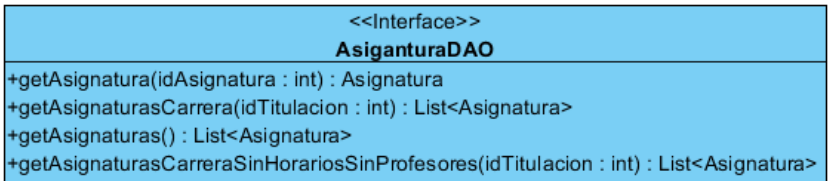


Ilustración 72. Interface AsignaturaDAO

La interfaz `AsignaturaDAO` contiene los métodos que deberá definir la clase encargada de la gestión de las asignaturas. Los métodos se encargan de retornar una lista de asignaturas o una asignatura concreta, el caso del método `getAsignaturasCarreraSinHorariosSinProfesores` devuelve una lista de la clase `Asignatura` pero con los atributos pero con los atributos `horarioList` y `profesorList` nulos, de esta forma se obtendrán las asignaturas de una forma más rápida cuando se necesite.

TitulacionDAO

Ilustración 73. Interface TitulacionDAO

La interfaz `TitulacionDAO` contiene los métodos que deberá definir la clase encargada de la gestión de las titulaciones. Los métodos definidos se encargan de devolver una lista de titulaciones o una titulación concreta dependiendo del método. El método `getTitulacionNombre` devuelve una lista ya que busca en los nombres de las titulaciones coincidencias con la cadena introducida por lo que puede haber más de una coincidencia y por tanto más de una titulación, y el método `getTitulacionesCentroSinAsignaturas` devuelve la lista de titulaciones de un centro, pero sin las listas de asignaturas en los objetos de la lista de titulaciones, es básicamente para tener más rendimiento a la hora de obtener solo las titulaciones de un centro. El método `getTitulacionesSoloAsignaturas` permite obtener las titulaciones y sus asignaturas sin los horarios de estas.

CentroDAO

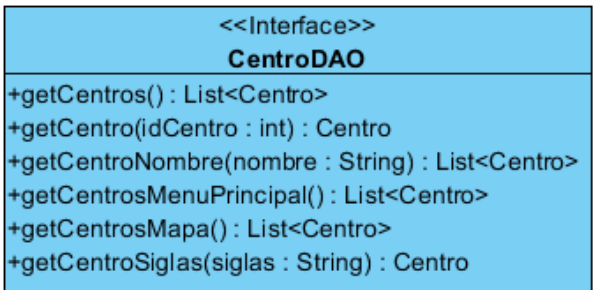


Ilustración 74. Interface CentroDAO

La interfaz `CentroDAO` contiene los métodos que deberá definir la clase encargada de la gestión de los centros. Los métodos definidos se encargan de devolver una lista de centros o un centro concreto dependiendo del método. La causa de que el método `getCentroNombre` devuelva una lista es que busca que los nombres de los centros contengan la cadena introducida por lo que puede haber más de una coincidencia y por tanto más de un centro, el método `getCentrosMenuPrincipal` devuelve la lista de centros que se mostraran en el listado de centro, evita recoger centros que no contengan titulaciones como el Vicerrectorado o el colegio mayor, además tampoco recoge los servicios y profesores asociados al centro para un mayor rendimiento en la ejecución, el método `getCentrosMapa` devuelve la lista de centros que se mostrarán en el mapa sin la lista de profesores asociada a cada centro., y el método `getCentrosSiglas` que permite obtener un centro a partir de sus siglas.

ServicioDAO

Ilustración 75. Interface ServicioDAO

La interfaz `ServicioDAO` contiene los métodos que deberá definir la clase encargada de la gestión de los servicios. Los métodos definidos se encargan de devolver una lista de servicios o un servicio concreto dependiendo del método. El caso del método `getServicioNombre` es similar al de los comentados en otros diagramas, buscando coincidencias de la cadena introducida por parámetros en los nombres de los servicios.

ProfesorDAO

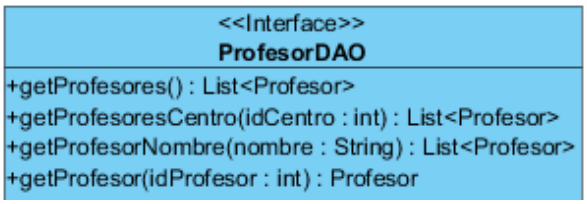


Ilustración 76. Interface ProfesorDAO

La interfaz `ProfesorDAO` contiene los métodos que deberá definir la clase encargada de la gestión de los profesores. Los métodos definidos se encargan de devolver una lista de profesores o un profesor concreto dependiendo del método. Como otras clases contiene un método que permite buscar los profesores por su nombre.

CalendarioDAO

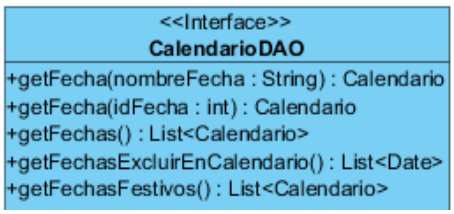


Ilustración 77. Interface CalendarioDAO

La interfaz `CalendarioDAO` contiene los métodos que deberá definir la clase encargada de la gestión de las fechas del calendario. Los métodos definidos se encargan de devolver una lista o una fecha concreta dependiendo del método. El método `getFechasExcluirEnCalendario` retorna todas las fechas del calendario que son no lectivas como periodos de vacaciones o puentes, y el método `getFechasFestivos` retorna los días que son festivos para incluirlos en el calendario de Google al realizar la importación.

UrlDAO

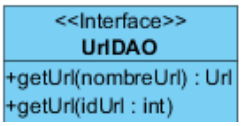


Ilustración 78. Interface UrlDAO

La interfaz `UrlDAO` contiene los métodos que deberá definir la clase encargada de la gestión de las URLs de la aplicación. Ambos métodos permiten buscar una URL por el nombre que tenga asignado o por el identificador.

ImpartidaDAO

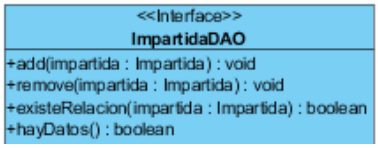


Ilustración 79. Interface ImpartidaDAO

La interfaz `ImpartidaDAO` no extiende de `UniversalDAO` ya que controla la relación en la base de datos entre los profesores y las asignaturas que imparten, por lo que únicamente permite añadir y eliminar relaciones, además el método `existeRelacion` permite averiguar si existe una relación entre un profesor y una asignatura.

SearchDAO

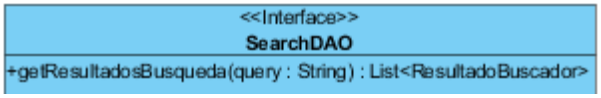


Ilustración 80. Interface SearchDAO

La interfaz que muestra la ilustración no extiende de `UniversalDAO`, esta interfaz contiene el método que utilizara el buscador de la aplicación para buscar datos y los devuelve en una lista con objetos del tipo `ResultadoBuscador`.

NoticiaDB

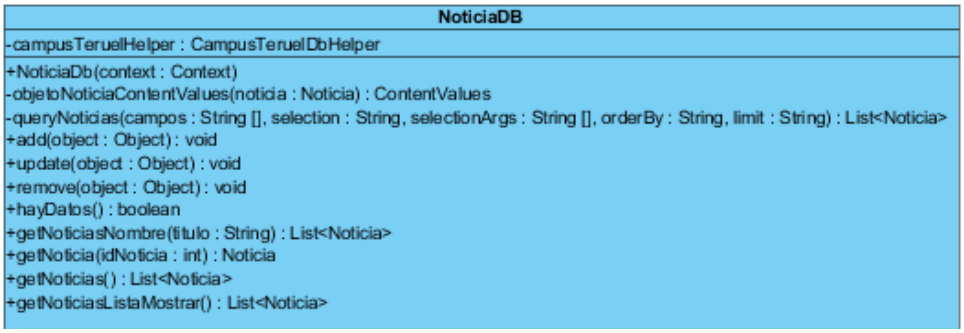


Ilustración 81. Clase NoticiaDB

La clase `NoticiaDB` implementa la interfaz `NoticiaDAO`, definiendo todos los métodos de esta para el acceso a los datos de las noticias en la base de datos. Incluye el método `objetoNoticiaContentValues` para crear un objeto `ContentValues` que permita añadir o actualizar una noticia en la base de datos y `queryNoticias` que permite realizar una consulta a la base de datos y devuelve los datos en una lista de noticias o en el caso de no haber encontrado ningún dato devuelve una lista vacía.

HorarioDB

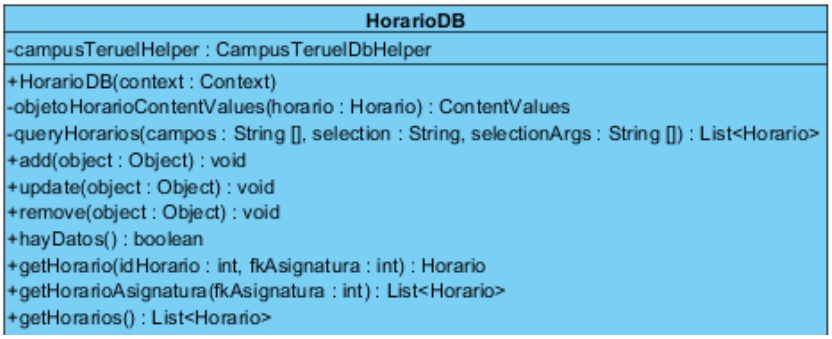


Ilustración 82. Clase HorarioDB

La clase `HorarioDB` implementa la interfaz `HorarioDAO`, definiendo todos los métodos de esta para el acceso a los datos de los horarios en la base de datos. Incluye el método `objetoHorarioContentValues` para crear un objeto `ContentValues` que permita añadir o actualizar un horario en la base de datos y `queryHorarios` que permite realizar una consulta a la base de datos y devuelve los datos en una lista de horarios, en el caso de no haber encontrado ningún dato devuelve una lista vacía.

AsignaturaDB

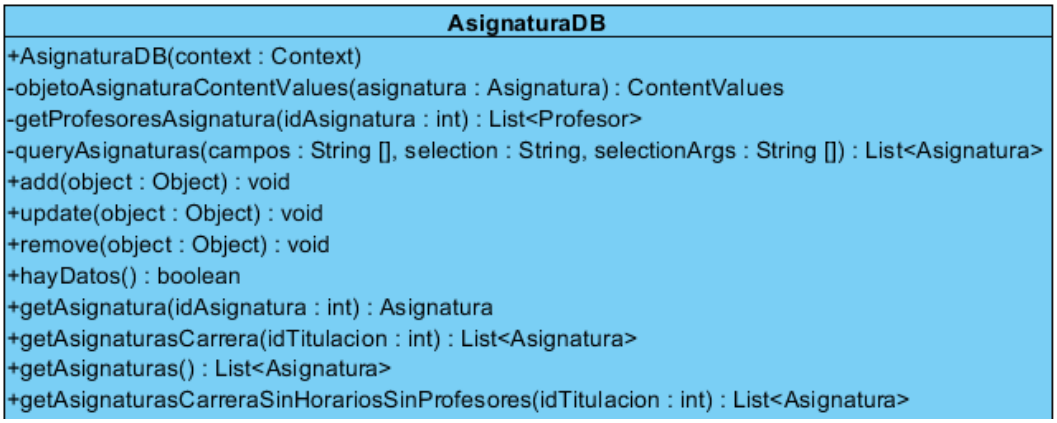


Ilustración 83. Clase AsignaturaDB

La clase `AsignaturaDB` implementa la interfaz `AsignaturaDAO`, definiendo todos los métodos de esta para el acceso a los datos de las asignaturas en la base de datos. Incluye el método `objetoAsignaturaContentValues` para crear un objeto `ContentValues` que permita añadir o actualizar una asignatura en la base de datos, el método `getProfesoresAsignaturas` que permite buscar todos los profesores de una asignatura, `queryAsignaturas` permite realizar una consulta a la base de datos y devuelve los datos en una lista de asignaturas, en el caso de no haber encontrado ningún dato devuelve una lista vacía.

TitulacionDB

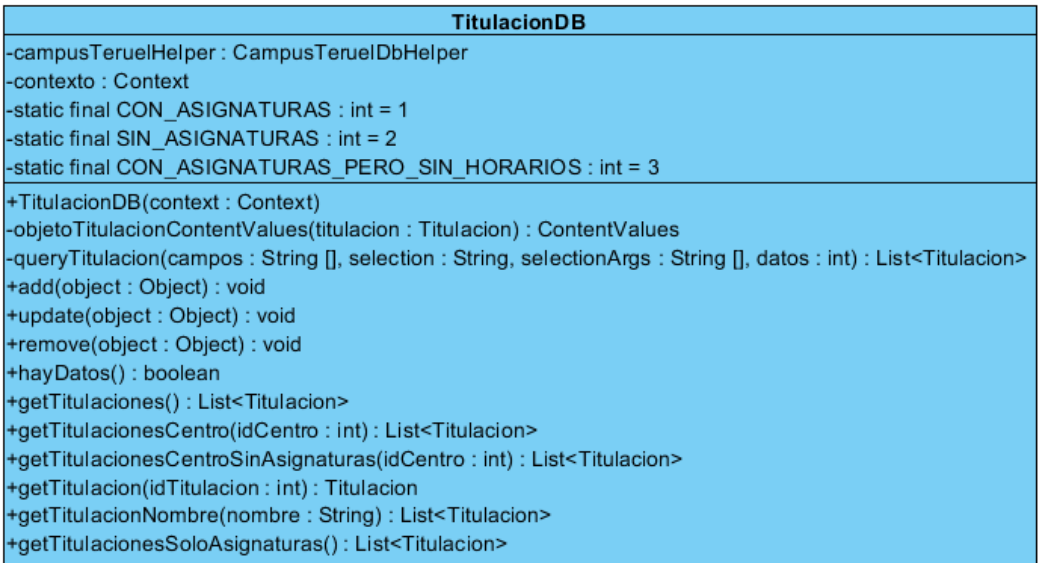


Ilustración 84. Clase TitulacionDB

La clase `TitulacionDB` implementa la interfaz `TitulacionDAO`, definiendo todos los métodos de esta para el acceso a los datos de las titulaciones en la base de datos. Incluye el método `objetoTitulacionContentValues` para crear un objeto `ContentValues` que permita añadir o actualizar una titulación en la base de datos y `queryTitulacion` que permite realizar una consulta a la base de datos y devuelve los datos en una lista de titulaciones, en el caso de no haber encontrado ningún dato devuelve una lista vacía.

CentroDB

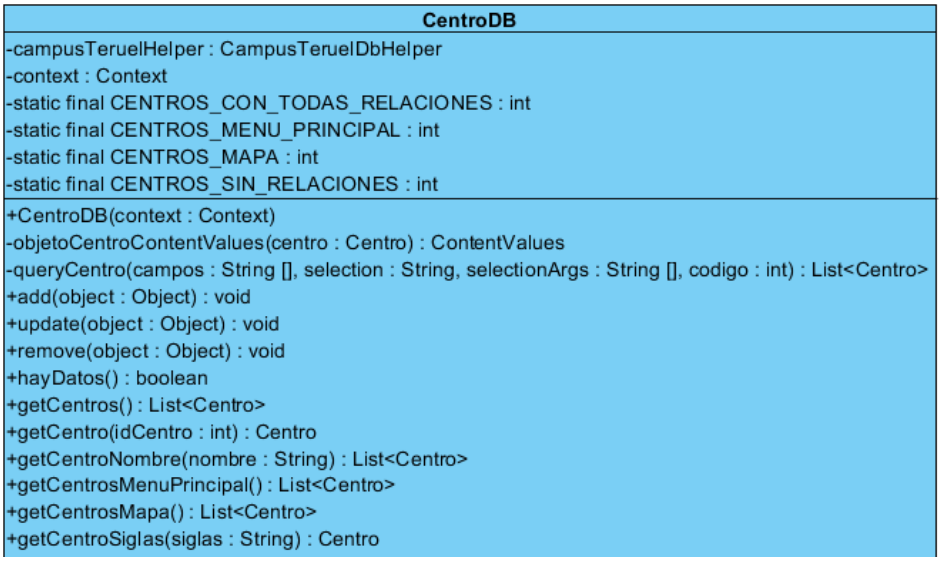


Ilustración 85. Clase CentroDB

La clase `CentroDB` implementa la interfaz `CentroDAO`, definiendo todos los métodos de esta para el acceso a los datos de los centros en la base de datos. Incluye el método `objetoCentroContentValues` para crear un objeto `ContentValues` que permita añadir o actualizar un centro en la base de datos y `queryCentro` que permite realizar una consulta a la base de datos y devuelve los datos en una lista de centros, en el caso de no haber encontrado ningún dato devuelve una lista vacía y por parámetros recibe no solo los campos para realizar la consulta sino también un código de tipo `int` que está definido en las constantes, de esta forma se puede reutilizar este método en la diferentes métodos `get`.

ServicioDAO

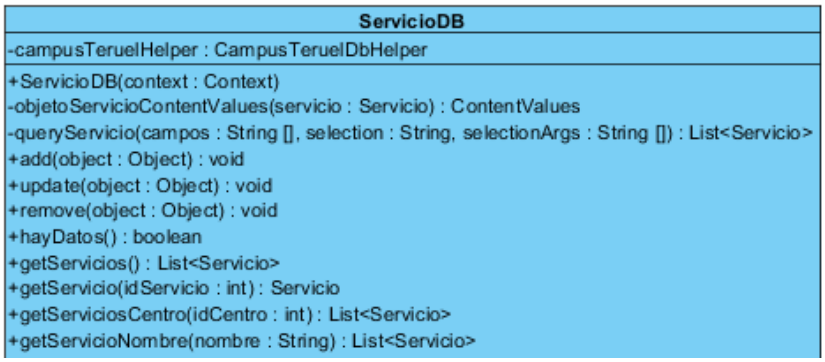


Ilustración 86. Clase ServicioDB

La clase `ServicioDB` implementa la interfaz `ServicioDAO`, definiendo todos los métodos de esta para el acceso a los datos de los servicios en la base de datos. Incluye el método `objetoServicioContentValues` para crear un objeto `ContentValues` que permita añadir o actualizar un servicio en la base de datos y `queryServicio` que permite realizar una consulta a la base de datos y devuelve los datos en una lista de servicios, en el caso de no haber encontrado ningún dato devuelve una lista vacía.

ProfesorDAO

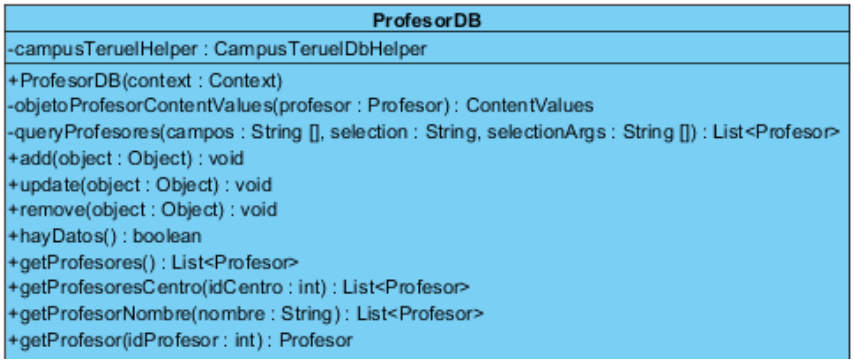


Ilustración 87. Clase ProfesorDB

La clase `ProfesorDB` implementa la interfaz `ProfesorDAO`, definiendo todos los métodos de esta para el acceso a los datos de los profesores en la base de datos. Incluye el método `objetoProfesorContentValues` para crear un objeto `ContentValues` que permita añadir o actualizar un profesor en la base de datos y `queryProfesores` que permite realizar una consulta a la base de datos y devuelve los datos en una lista de profesores, en el caso de no haber encontrado ningún dato devuelve una lista vacía.

CalendarioDB

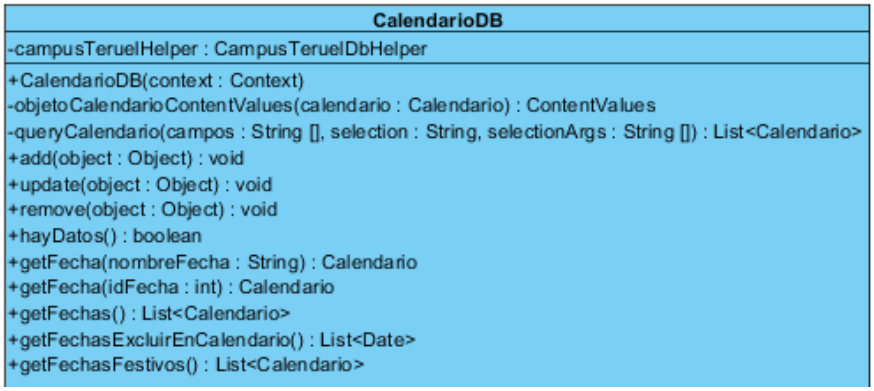


Ilustración 88. Clase CalendarioDB

La clase `CalendarioDB` implementa la interfaz `CalendarioDAO`, definiendo todos los métodos de esta para el acceso a los datos de las fechas del calendario en la base de datos. Incluye el método `objetoCalendarioContentValues` para crear un objeto `ContentValues` que permita añadir o actualizar una fecha en la base de datos y `queryCalendario` que permite realizar una consulta a la base de datos y devuelve los datos en una lista del objeto calendario, en el caso de no haber encontrado ningún dato devuelve una lista vacía.

UrlDB

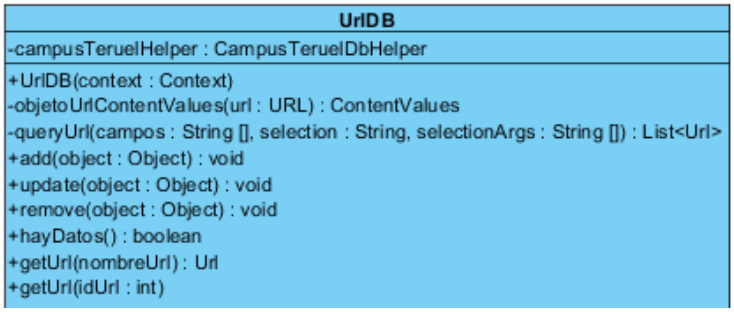


Ilustración 89. Clase UrlDB

La clase `UrlDB` implementa la interfaz `UrlDAO`, definiendo todos los métodos de esta para el acceso a los datos de las URLs en la base de datos. Incluye el método `objetoUrlContentValues` para crear un objeto `ContentValues` que permita añadir o actualizar una URL en la base de datos y `queryUrl` que permite realizar una consulta a la base de datos y devuelve los datos en una lista del objeto URL, en el caso de no haber encontrado ningún dato devuelve una lista vacía.

ImpartidaDB

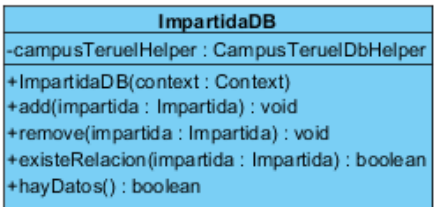


Ilustración 90. Clase ImpartidaDB

La clase `ImpartidaDB` implementa la interfaz `ImpartidaDAO`, definiendo todos los métodos de esta para el acceso a las relaciones entre profesores y las asignaturas que imparten en la base de datos.

SearchDB

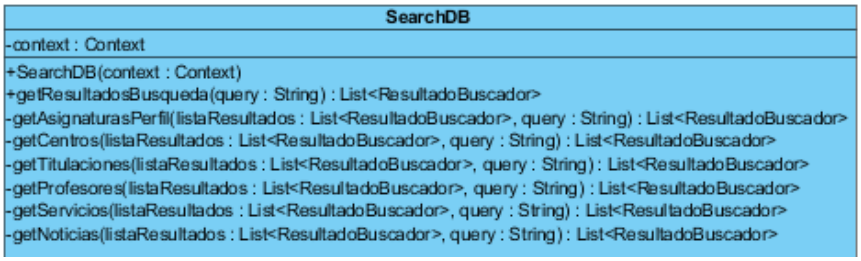


Ilustración 91. Clase SearchDB

La clase `SearchDB` implementa la interfaz `SearchDAO`, define el método `getResultadosBusqueda` para realizar búsquedas en la base de datos desde el buscador, para completar esta tarea contiene métodos privados para buscar en cada tabla de la base de datos y añadirlos en la lista de resultados.

CampusTeruelDbHelper

Ilustración 92. Clase CampusTeruelDbHelper

La clase `CampusTeruelDbHelper` mostrada en la ilustración anterior, contiene los nombres de los campos y las tablas de la base de datos, además también contiene las sentencias `SQL` para crear y eliminar las tablas. Extiende de `SQLiteOpenHelper` por lo que sobrescribe los métodos `onCreate` y `onUpgrade` para la creación de la base de datos y posibles actualizaciones de la estructura.

RssParserDom

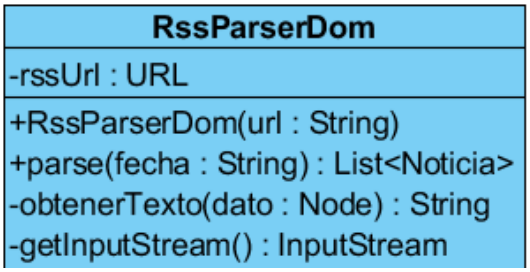


Ilustración 93. Clase RssParserDom

La clase que muestra la ilustración anterior permite procesar el RSS de las noticias del campus, con el método `parse` se recogen las noticias y se introducen en una lista.

ConnectionDetector

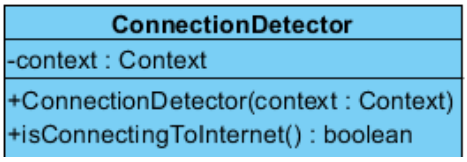


Ilustración 94. Clase ConnectionDetector

Esta clase permite saber si el usuario tiene conexión a la red con el método `isConnectingToInternet.`

WakeLocker

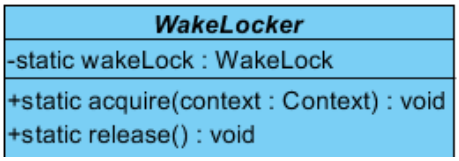


Ilustración 95. Clase WakeLocker

La clase abstracta `WakeLocker` permite “despertar” y “dormir” el teléfono cuando recibe las notificaciones del GCM, de esta forma cuando se reciba una notificación GCM el dispositivo despertará, realizará una ejecución en segundo plano y finalmente volver a dormir el dispositivo dejando la CPU libre.

GoogleCalendar

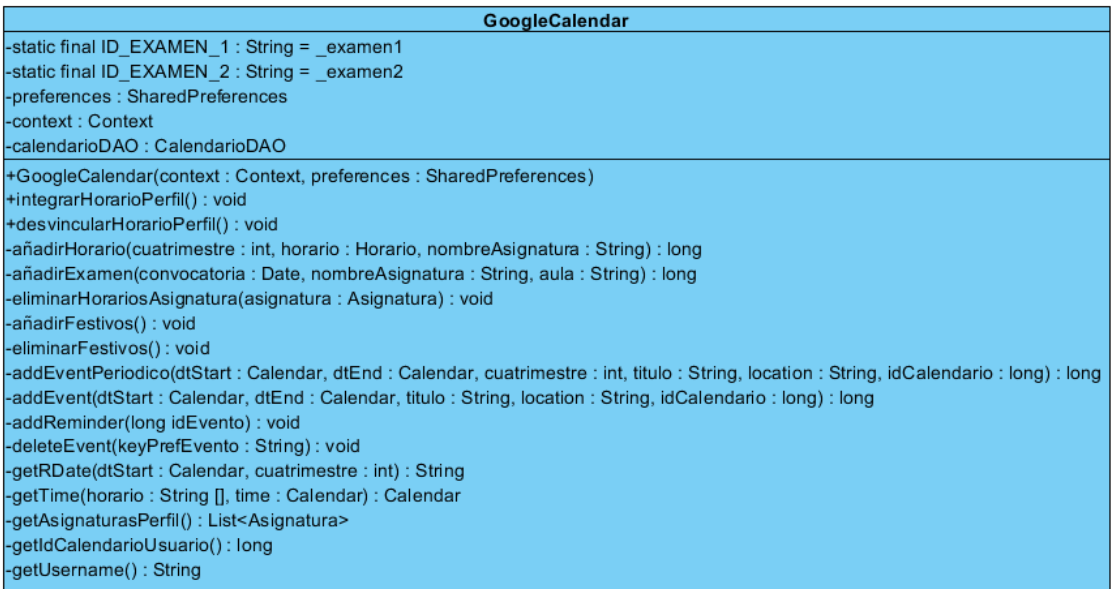


Ilustración 96. Clase GoogleCalendar

La clase `GoogleCalendar` contiene los métodos necesarios para realizar la importación y desvinculación en el calendario de Google del usuario de las asignaturas elegidas por el usuario en el perfil. Los métodos públicos `integrarHorarioPerfil` y `desvincularHorarioPerfil` son los utilizados para la tas tareas comentadas e internamente hacen uso de todos los métodos privados de la clase para realizar amabas operaciones. Los métodos de añadir y eliminar se encargan de preparar la información de horarios y exámenes de las asignaturas para después pasárselos a los métodos `add` y `delete` para realizar las acciones en el calendario de Google, por último los métodos `get` que devuelven información necesaria como `getRDate` que indica las fechas que debe repetirse un horario, evitando las fechas festivas o vacaciones.

FileDownloader

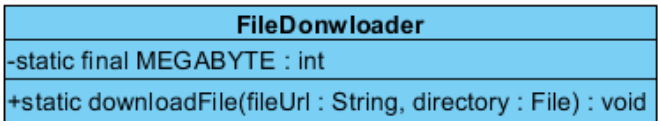


Ilustración 97. Clase FileDownloader

Esta clase permite descargarse un fichero con el método estático `downloadFile`.

DownloadPDF

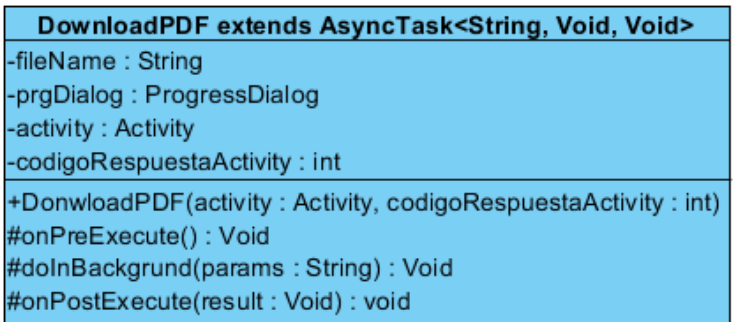


Ilustración 98. Clase DownloadPDF

La clase `DownloadPDF` es un hilo que permite hacer uso de la clase `FileDownloader` y conectar con una URL para descargar un fichero y abrirlo al finalizar la descarga del fichero.

AutomaticApiRestClient

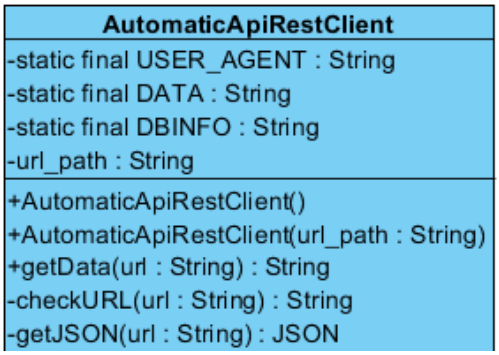


Ilustración 99. Clase AutomaticApiRestClient

Esta clase permite conectar, a través de una URL, con Automatic Api Rest en el servidor y recoger los datos que se indiquen en la URL. El método principal es `getData` que devuelve la respuesta JSON del servidor en un `String`, el método `checkURL` se encarga de comprobar que es una dirección válida y el método `getJSON` es el que realmente lanza la petición al servidor y recibe la respuesta.

DatosServidorInicioTask

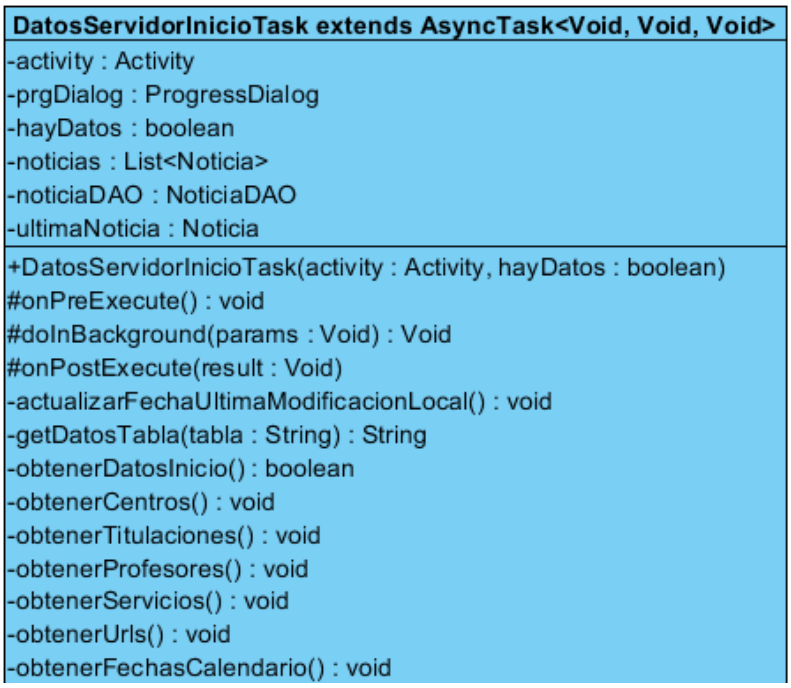


Ilustración 100. Clase DatosServidorInicioTask

La clase `DatosServidorInicioTask` es el hilo que se lanza en el inicio de la aplicación para recoger los datos del servidor y obtener las noticias o comprobar si hay noticias nuevas.

DatosServidorGCM

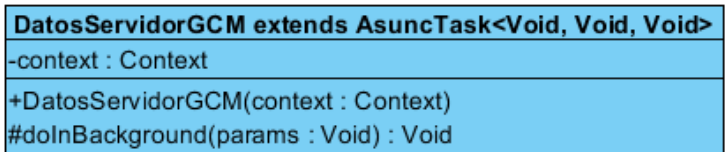


Ilustración 101. Clase DatosServidorGCM

La clase que se muestra en la ilustración, es el hilo que se ejecuta cuando la aplicación recibe una notificación GCM, este internamente ejecuta el método `comprobacionesModificacionesEnElServidor` de la clase `ComprobacionModificacionesServidor`.

ComprobacionModificacionesServidor

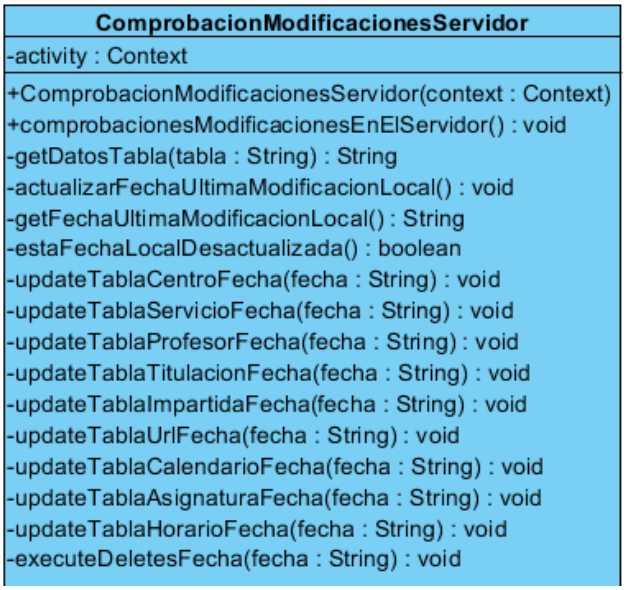


Ilustración 102. Clase ComprobacionModificacionesServidor

Esta clase contiene los métodos necesarios para comprobar si se han realizado modificaciones en la base de datos del servidor, a partir de la fecha indicada.

DatosServidorAsignaturasTask

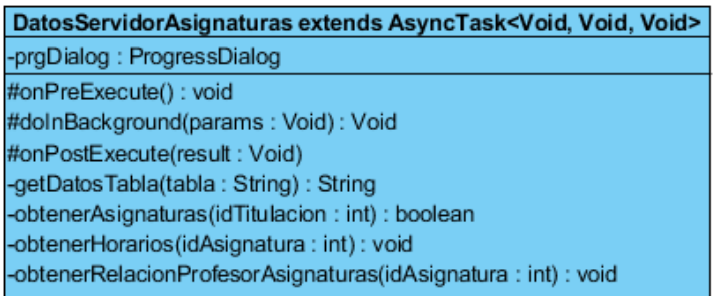


Ilustración 103. Clase DatosServidorAsignaturasTask

La clase `DatosServidorAsignaturasTask` es el hilo que se lanza, cuando se accede a la creación del perfil y no existen asignaturas, para que se encargue de mandar la petición al servidor y recoger los datos y almacenarlos en el servidor.

ServerUtilies

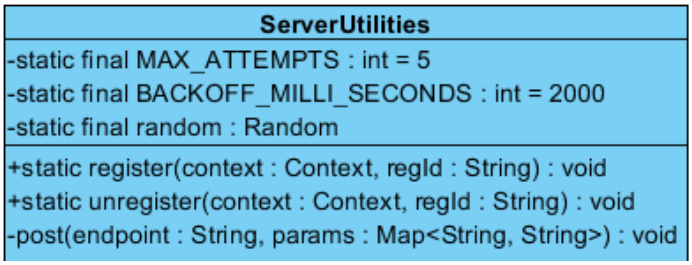


Ilustración 104. Clase ServerUtilies

La clase `ServerUtilies` permite registrar o eliminar el id del dispositivo recibido por el GCM en la base de datos del servidor del sistema.

GCMIntentService

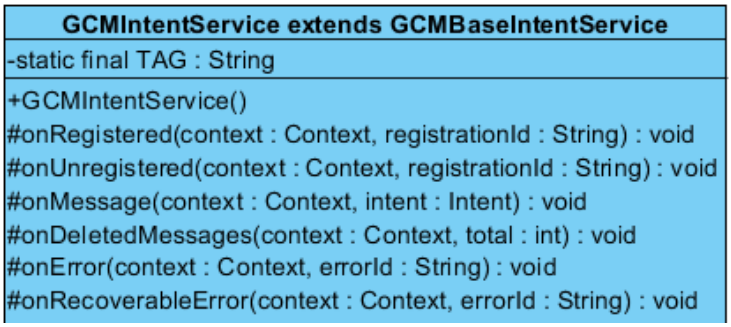


Ilustración 105. Clase GCMIntentService

Esta clase es necesaria para implementar las notificaciones GCM, en los métodos `onRegistered y onUnregistered` se hace uso de la clase `ServerUtilies` para registrar o borrar el id del dispositivo, otro método importante es `onMessage` método que se ejecuta cuando la aplicación recibe un mensaje.

Config

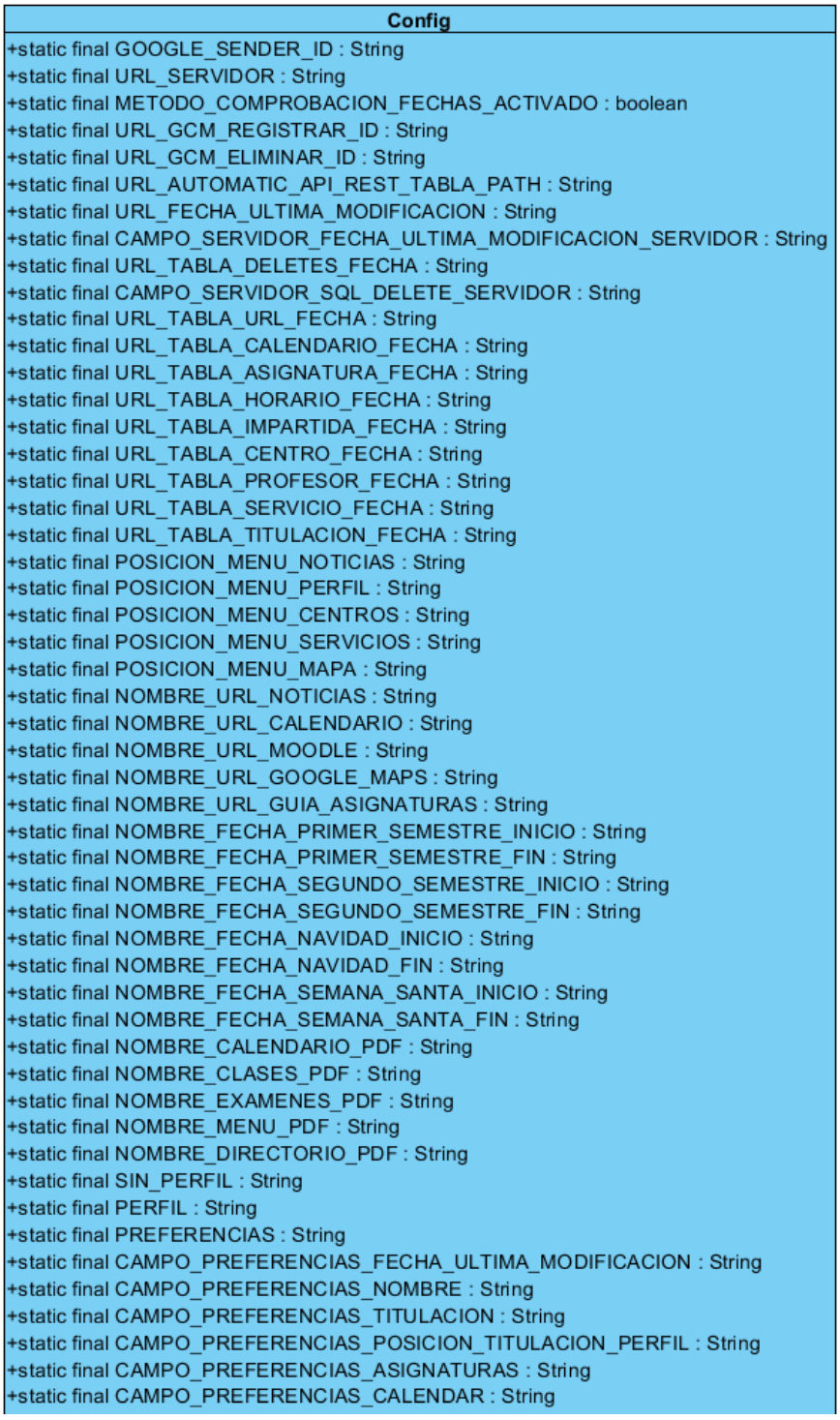


Ilustración 106. Clase Config

Esta clase contiene las variables que se utilizan durante la ejecución de la aplicación, principalmente son las claves para tener acceso a un dato que puede variar pero que se obtiene a partir de esa clave. Algunas variables destacables son `GOOGLE_SENDER_ID` que contiene el id para hacer uso de Google Play Services, necesario para Google Maps y Google Cloud Messaging, o las variables URL que son utilizadas para lanzar las peticiones al servidor.

E.2. Aplicación web

GCM

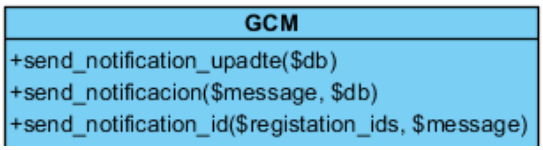


Ilustración 107. Clase GCM

La clase de la ilustración contiene los métodos que permiten mandar mensajes a los dispositivos a través de GCM, con `send_notificaction_update` se manda el mensaje "update" a la aplicación para que esta compruebe las actualizaciones de las base de datos del servidor y el método `send_notification` permite recibir un mensaje personalizado, y ambos métodos hacen uso de `send_notification_id` para mandar el mensaje al servidor GCM con los identificadores de los dispositivos.

Servicio

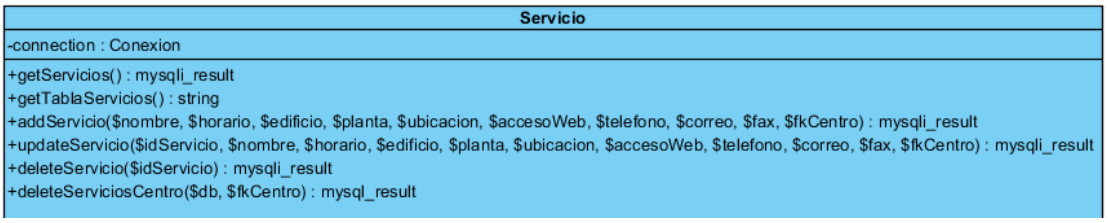


Ilustración 108. Clase Servicio

La clase `Servicio` contiene los métodos necesarios para interactuar con los servicios que hay en la base de datos.

Asignatura

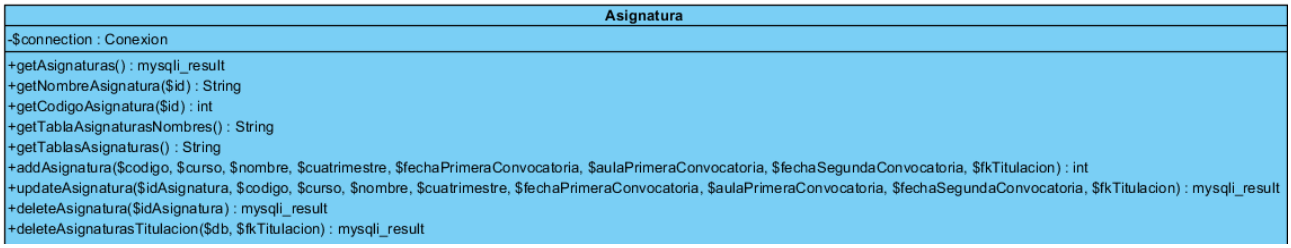


Ilustración 109. Clase Asignatura

La clase `Asignatura` contiene los métodos necesarios para interactuar con las asignaturas que hay en la base de datos.

Calendario

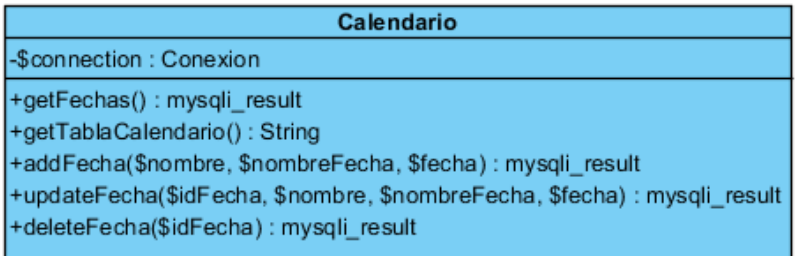


Ilustración 110. Clase Calendario

La clase `Calendario` contiene los métodos necesarios para interactuar con las fechas del calendario que hay en la base de datos.

Horario

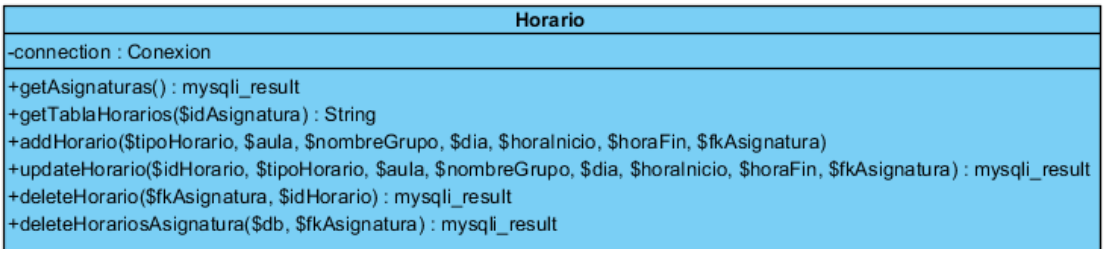


Ilustración 111. Clase Horario

La clase `Horario` contiene los métodos necesarios para interactuar con los horarios de las asignaturas que hay en la base de datos.

Impartida

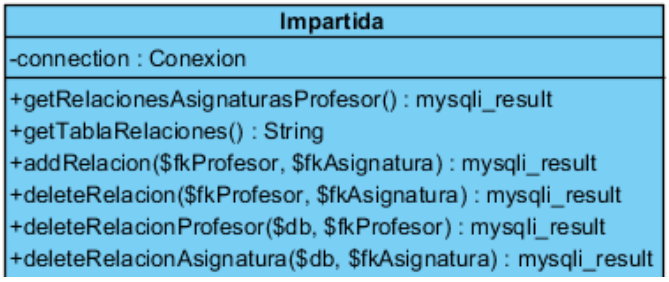


Ilustración 112. Clase Impartida

La clase `Impartida` contiene los métodos necesarios para interactuar con la tabla que relaciona profesores y sus asignaturas en la base de datos.

Profesor

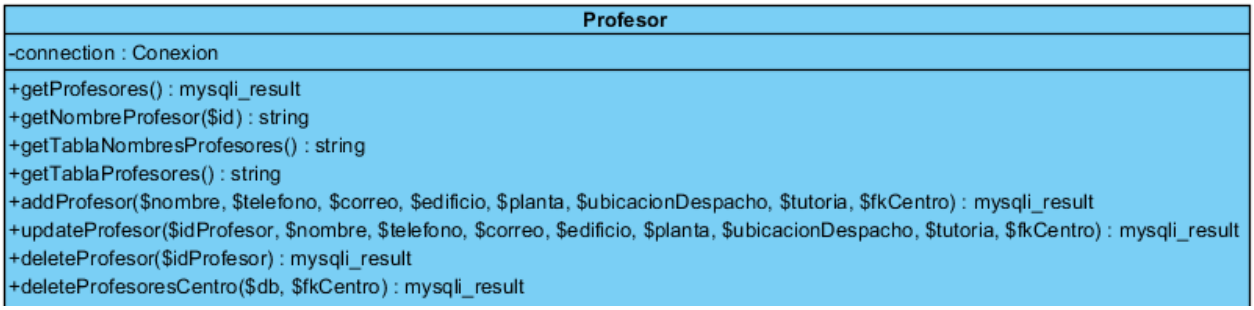


Ilustración 113. Clase Profesor

La clase `Profesor` contiene los métodos necesarios para interactuar con los profesores que hay en la base de datos.

Centro

Ilustración 114. Clase Centro

La clase `Centro` contiene los métodos necesarios para interactuar con los profesores que hay en la base de datos.

Titulacion

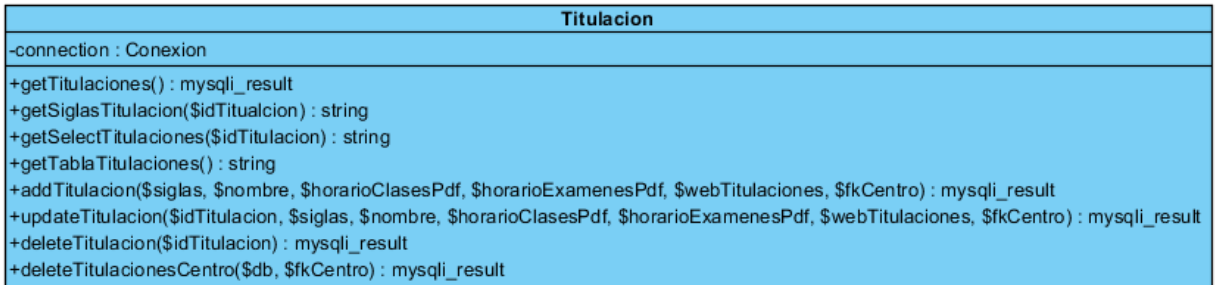


Ilustración 115. Clase Titulacion

La clase `Titulación` contiene los métodos necesarios para interactuar con las titulaciones que hay en la base de datos.

Url

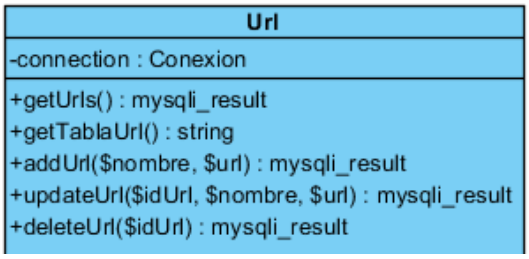
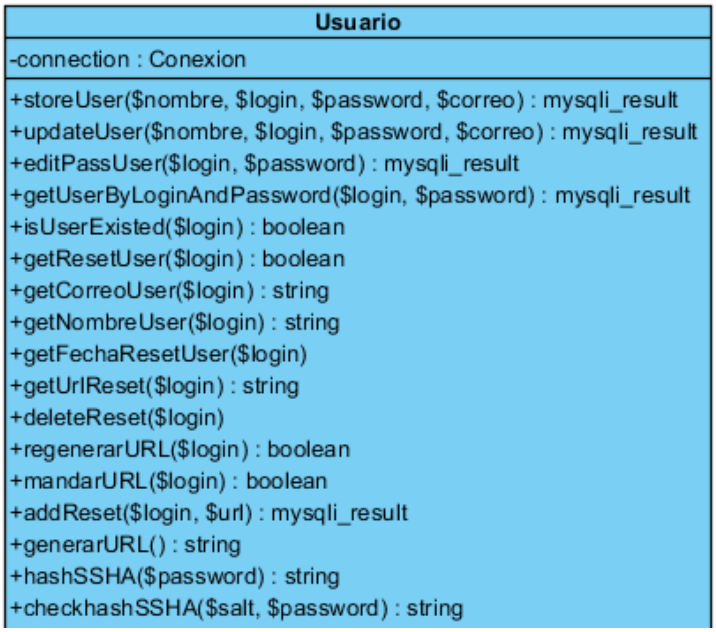


Ilustración 116. Clase Url

La clase `Url` contiene los métodos necesarios para interactuar con las URLs que hay en la base de datos.

Usuario



La clase `Usuario` contiene los métodos necesarios para interactuar con los usuarios que hay en la base de datos.

DB_Functions

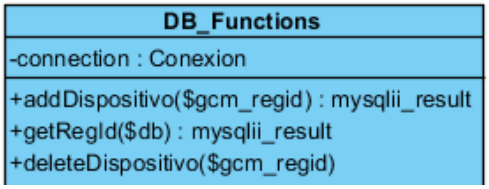


Ilustración 117. Clase DB_Funtions

La clase `DB_Functions` contiene los métodos necesarios para interactuar con los dispositivos que hay en la base de datos.

Conexion

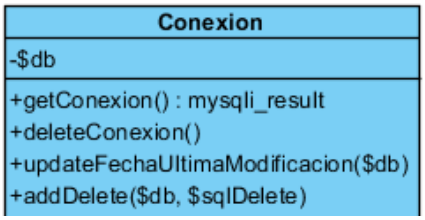


Ilustración 118. Clase Conexión

Esta clase contiene los métodos para realizar la conexión con la base de datos para poder ejecutar consultas, también contiene el método para actualizar el campo que indica la fecha de la última actualización de la base de datos y el método para añadir código SQL en el histórico de borrados.

Anexo F: Diagramas de Actividad

F.1. Aplicación cliente

Inicio

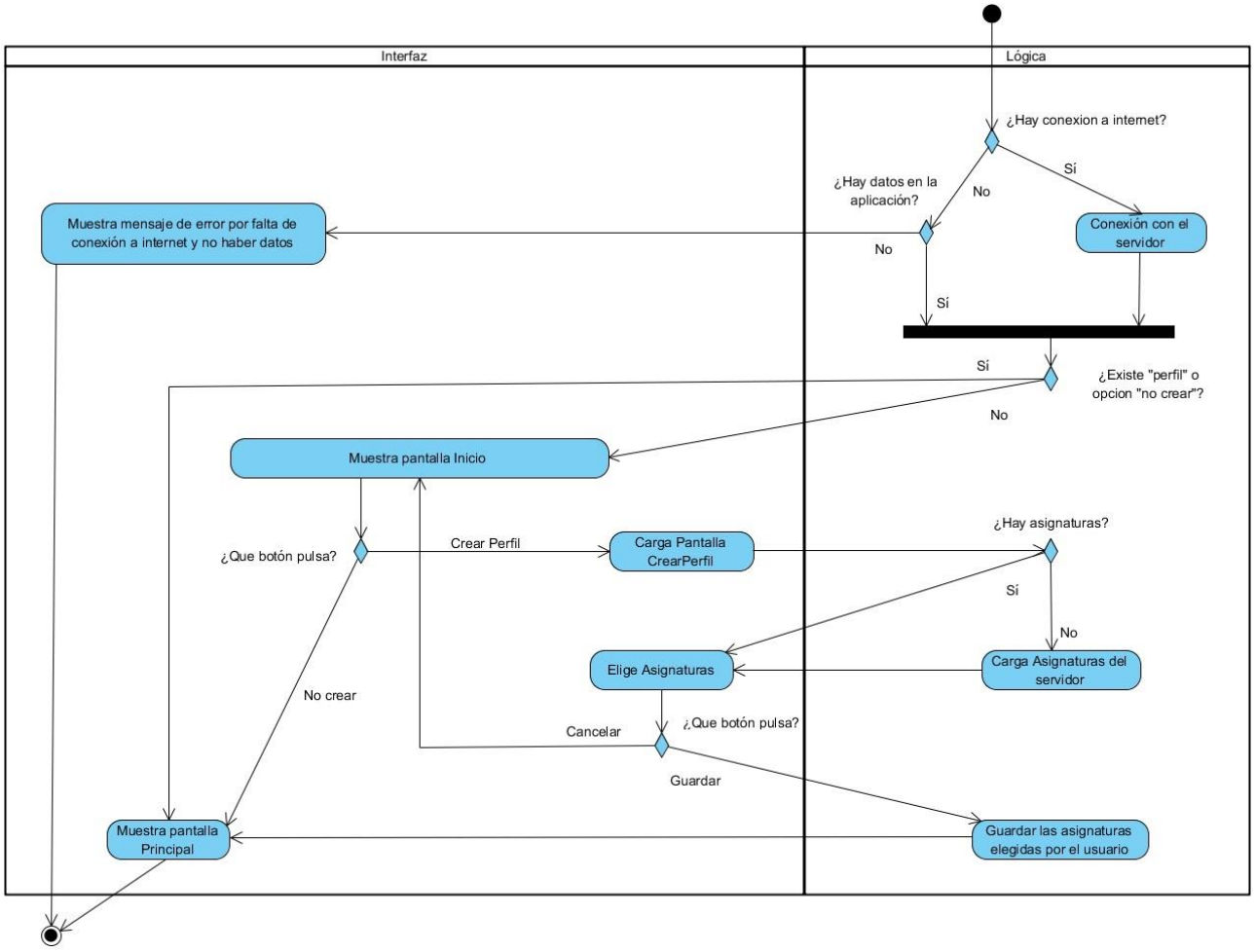


Ilustración 119. Diagrama de actividad - Inicio

El diagrama que muestra la Ilustración 119 representa el diagrama de actividad que se produce en el inicio de la aplicación. Primero realiza una comprobación de la conexión a Internet, en caso de que no haya únicamente dejara continuar si hay datos en la aplicación, si hay conexión se realizara una comunicación con el servidor, para una mejor visualización del diagrama las acciones de conexión que el servidor de este diagrama se detallaran en diagramas posteriores. Tras esto se realiza una comprobación para saber si el usuario ya ha elegido entre la opción de crear o no el perfil, si el usuario ya ha elegido la opción ira a la pantalla principal, en caso contrario se le dejara seleccionar la opción. Dependiendo de la opción elegida el usuario irá directamente a la pantalla principal o pasará por el proceso de seleccionar las asignaturas para su perfil y finalmente acceder a la pantalla principal.

Conexión Servidor Inicio

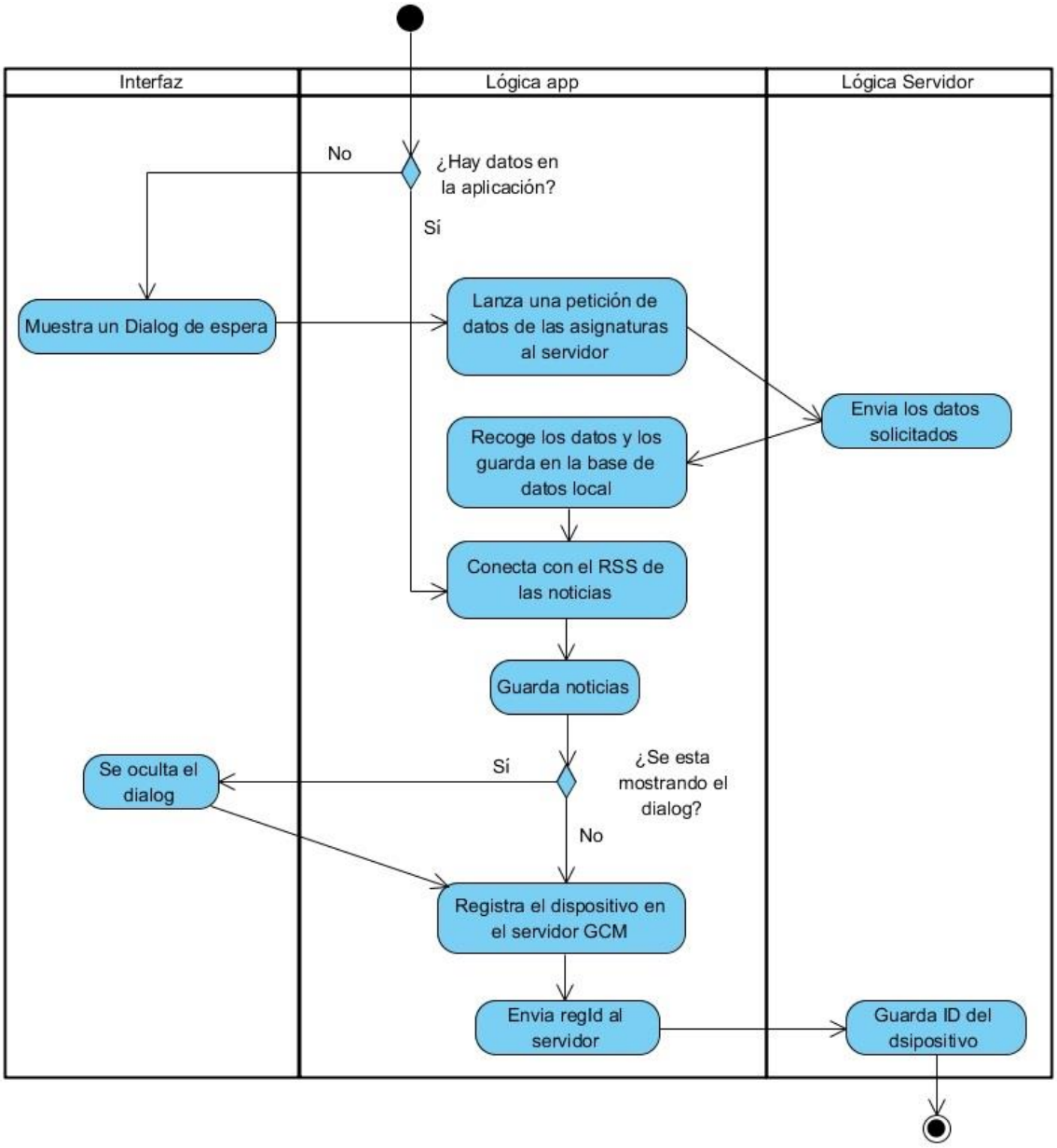


Ilustración 120. Diagrama de actividad - Conexión Servidor Inicio

La anterior ilustración muestra la acción que se desarrolla cuando la aplicación conecta con el servidor en el inicio. Primero realiza la comprobación de si hay datos o no en la aplicación, en el caso de haya salta directamente a comprobar las noticias y a registrar el dispositivo en el servidor GCM, en el caso de que no haya datos lanza una petición al servidor para recibir los datos y los guarda en la base de datos local.

Carga Asignaturas del Servidor

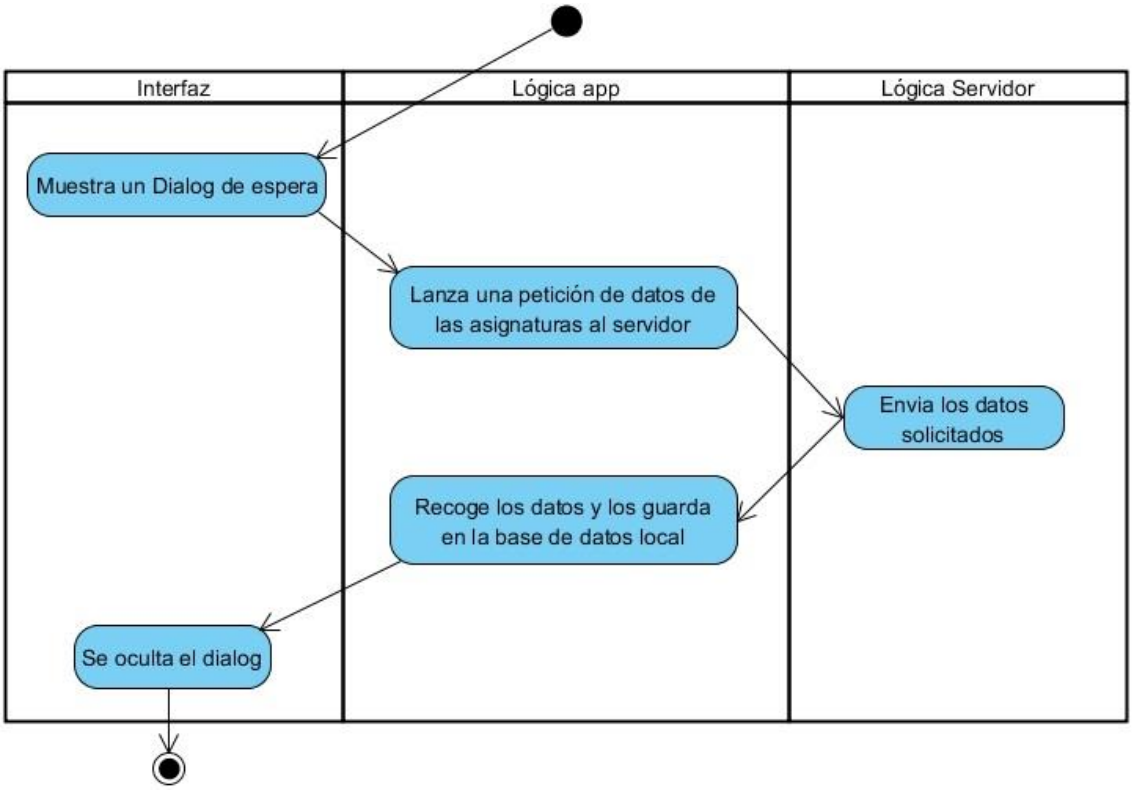


Ilustración 121. Diagrama de actividad - Carga asignaturas del servidor

Este diagrama muestra la acción que se realiza cuando el usuario va a realizar la elección de las asignaturas de su perfil y no hay datos de las asignaturas en el perfil. Primero muestra un ProgressDialog para indicar al usuario que esté realizando una descarga de datos y después lanzara una petición al servidor para descargar los datos y guardarlos en la base de datos local.

Editar Perfil

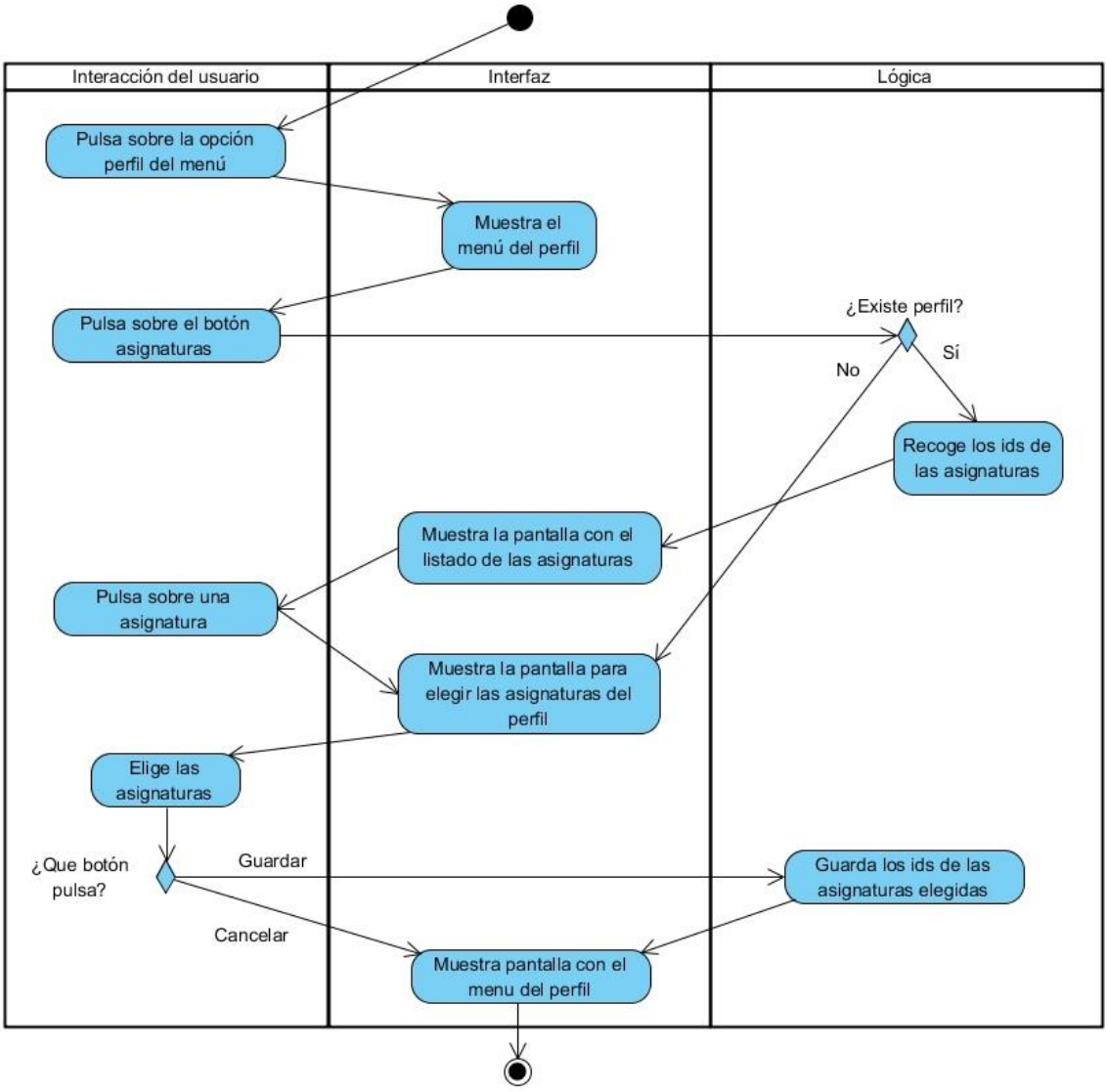


Ilustración 122. Diagrama de actividad - Editar perfil

Esta ilustración muestra el proceso de actividades que debe seguir el usuario para llegar a editar las asignaturas elegidas en el perfil. Primero debe llegar al menú de la pantalla perfil a través del menú principal de la aplicación y pulsar sobre la opción de asignaturas, en el caso de que no haya elegido asignaturas anteriormente la aplicación directamente lo mandará a la pantalla para elegir asignaturas, si ha elegido asignaturas anteriormente primero lanzará la pantalla con el listado de asignaturas y tendrá la posibilidad de pulsar el botón de editar que lo mandará a la pantalla de elegir asignaturas, una vez en esa pantalla podrá decidir entre guardar las asignaturas que elija o cancelar la operación.

Google Calendar

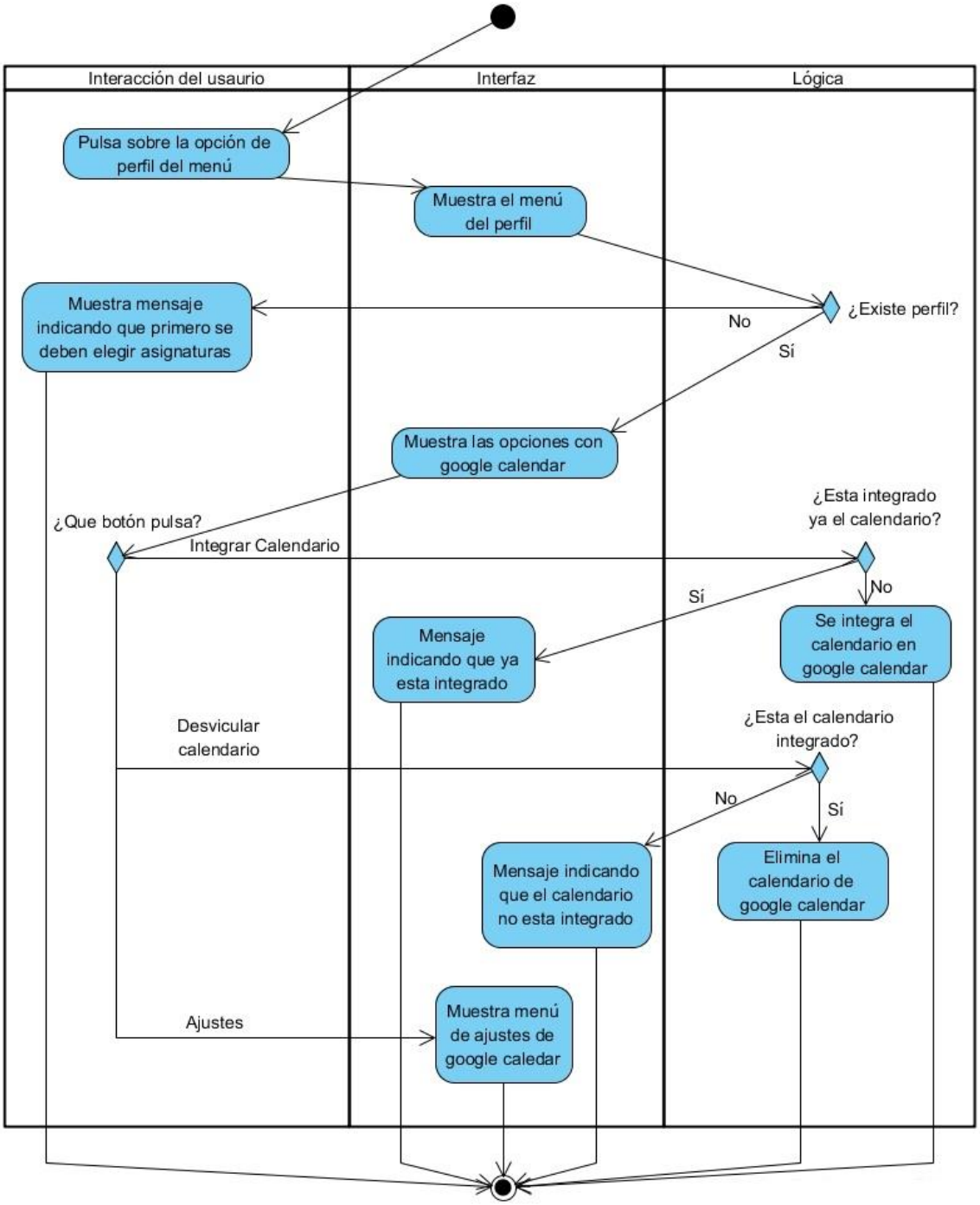


Ilustración 123. Diagrama de actividad - Google Calendar

Este diagrama muestra el proceso de acciones, que puede realizar el usuario, relacionados con Google Calendar. Desde el menú del perfil podrá acceder a la opción Google Calendar si tiene un perfil creado, en caso contrario recibirá un mensaje informándole que debe crear un perfil. Una vez dentro, el usuario tiene tres posibilidades: integrar calendario, desvincular calendario y ajustes, cuando el usuario pulse alguno de esos botones se realizara su acción correspondiente.

Ver asignatura

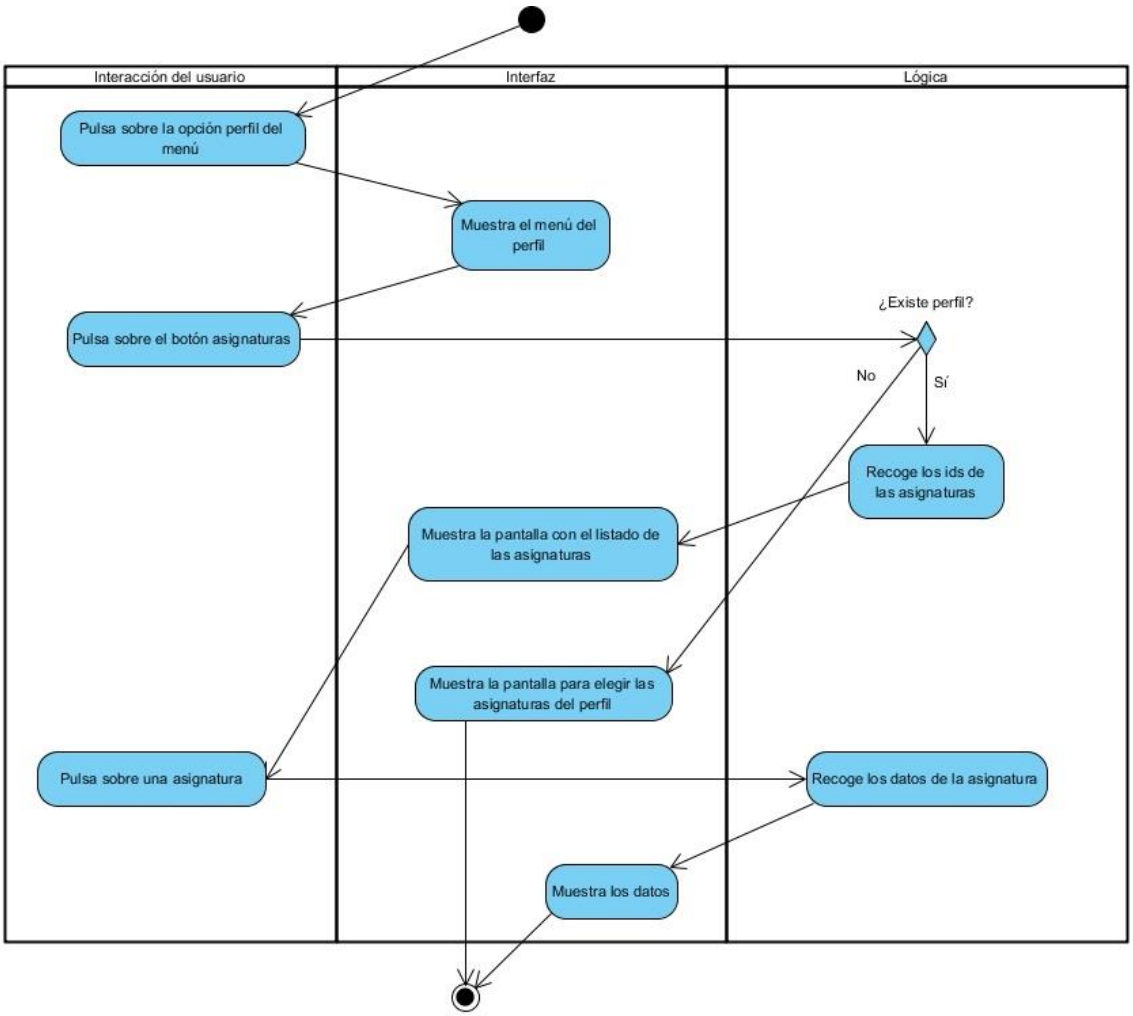


Ilustración 124. Diagrama de actividad - Ver asignatura

El diagrama de la ilustración muestra el proceso que sigue el usuario hasta que ve la información de una asignatura concreta. El proceso es similar al de editar perfil, pero en este caso en la pantalla que muestra la lista de asignaturas debe seleccionar una en concreto para recoger los datos desde la base de datos local y mostrarlos.

Ver datos de un centro

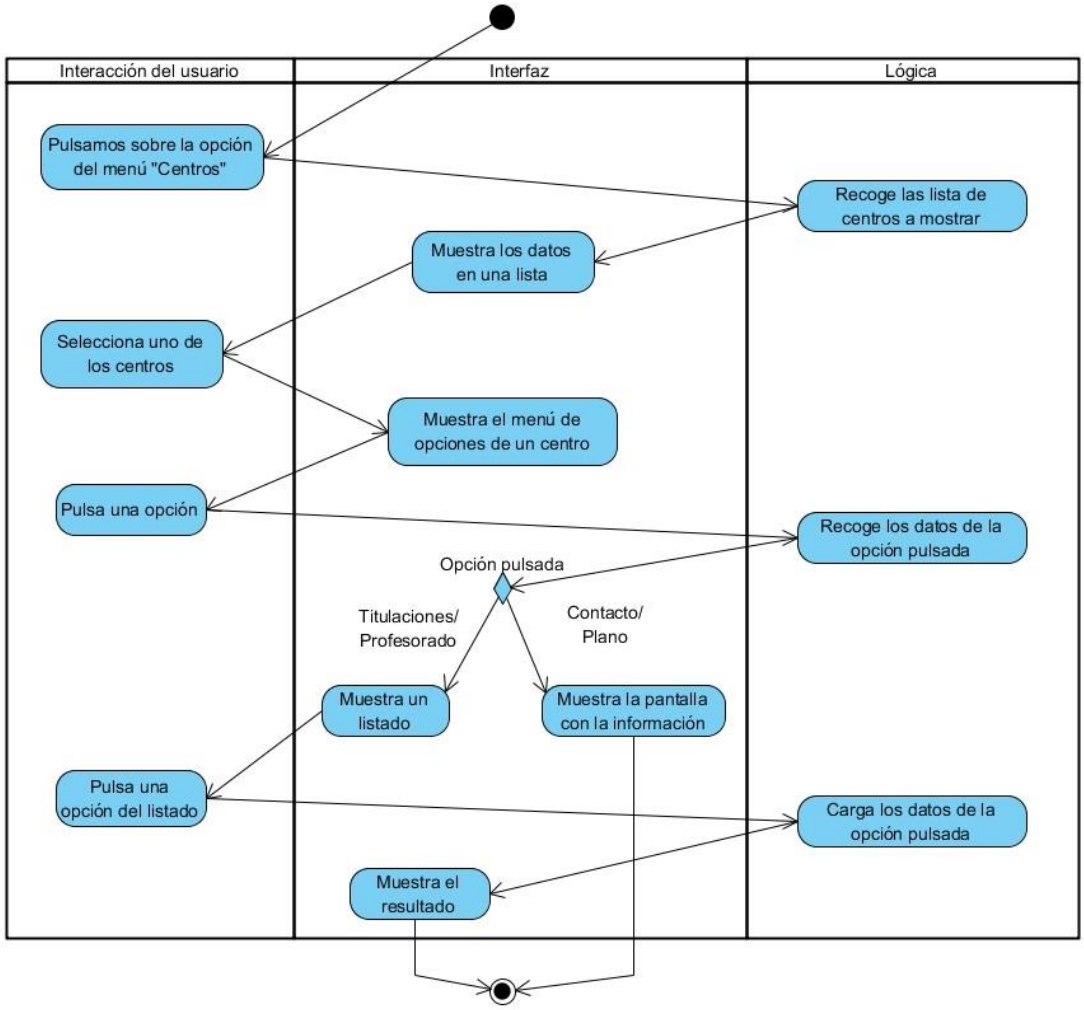


Ilustración 125. Diagrama de actividad - Ver datos de un centro

En este diagrama se ve que el usuario tiene diferentes opciones a la hora de pulsar sobre un centro, el caso de pulsar sobre el botón *titulaciones* o *profesorado*, se verá un listado con todos los datos y podrá pulsar una de las filas para ver sus datos, en el caso de ser *contacto* o *plano* el botón pulsado se mostrara directamente la pantalla con la información.

Ver datos

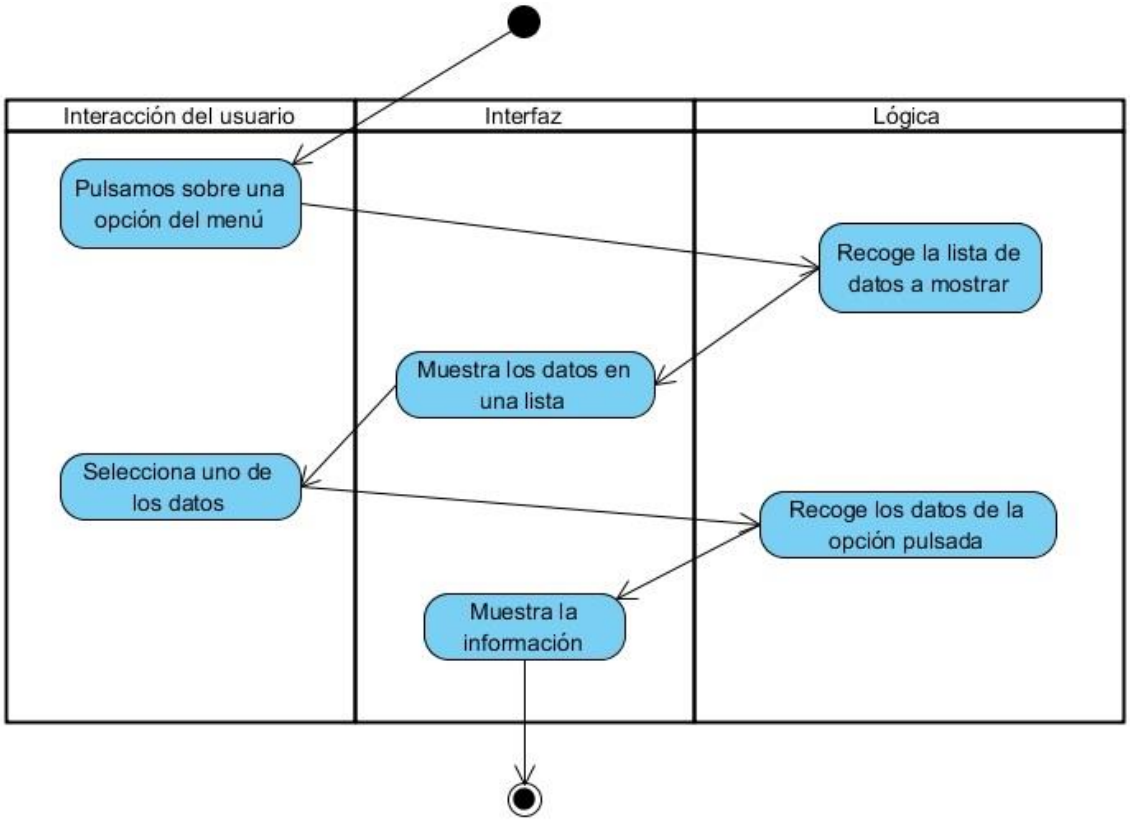


Ilustración 126. Diagrama de actividad - Ver datos

El diagrama de actividad que se muestra en esta ilustración vale para los casos de los servicios y las noticias, ya que ambos se muestran en un listado y después el usuario tiene la opción de pulsar una fila en concreto para ver la información del servicio o la noticia al completo.

Buscador

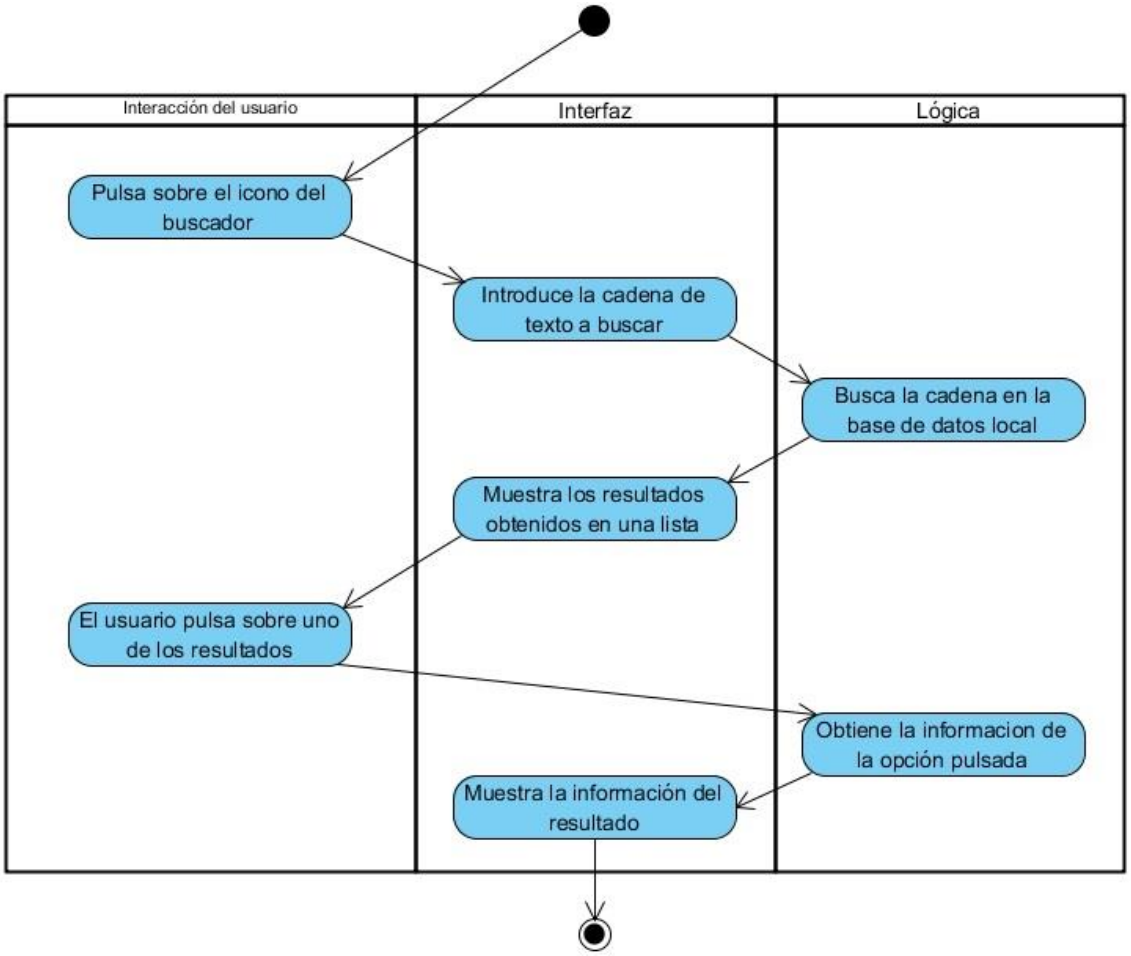


Ilustración 127. Diagrama de actividad - Buscador

Cuando el usuario pulse sobre el icono del buscador se desplegara una caja de texto en la que introducirá la cadena de texto a buscar, la cadena se buscara en los datos indicados en los requisitos y una vez finalizada la búsqueda mostrara los datos encontrados en un listado con la posibilidad de tener acceso a la información de uno de esos datos pulsando sobre su fila.

Mapa

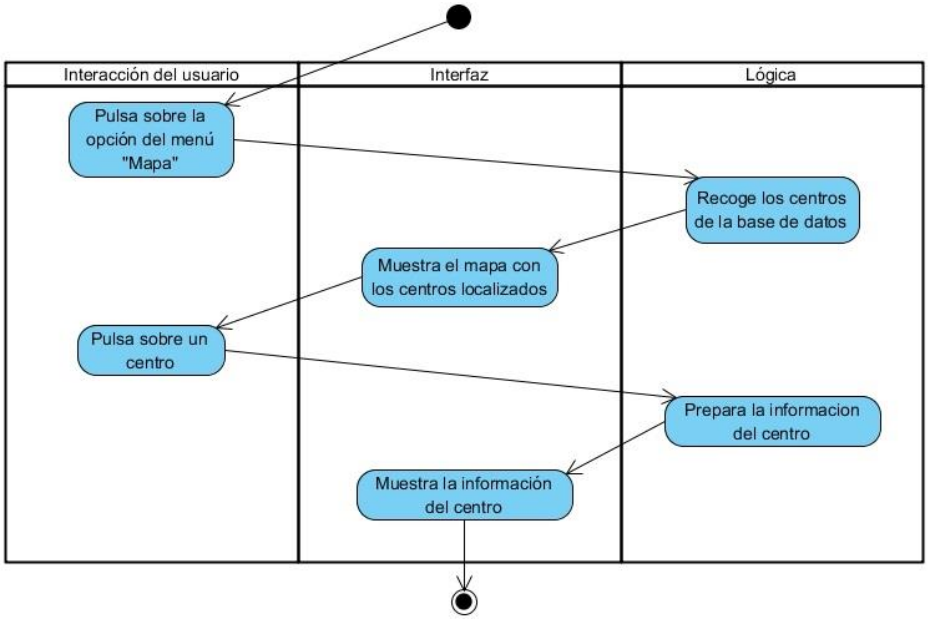


Ilustración 128. Diagrama de actividad – Mapa

Cuando el usuario pulse sobre la opción del menú “Mapa”, la aplicación deberá recoger todos los centros que existan en la base de datos y mostrárselos al usuario en un mapa de Google Maps con los centros posicionados sobre sus coordenadas y si el usuario pulsa sobre uno de los centros la aplicación deberá mostrarle la información del centro.

Cargar PDF

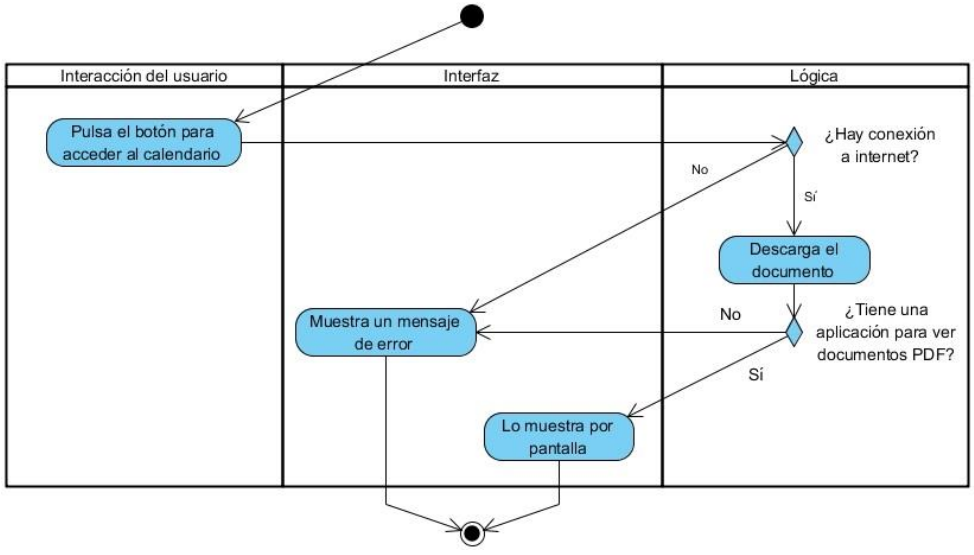


Ilustración 129. Diagrama de actividad - Cargar PDF

Desde la aplicación se puede realizar el proceso de visualizar archivos PDF colgados en la red. En el anterior diagrama se muestra el proceso que se debe seguir para tener acceso al documento y lo muestra en caso de ser posible. Este es un proceso idéntico cada vez que se accede a un archivo PDF.

Cargar web

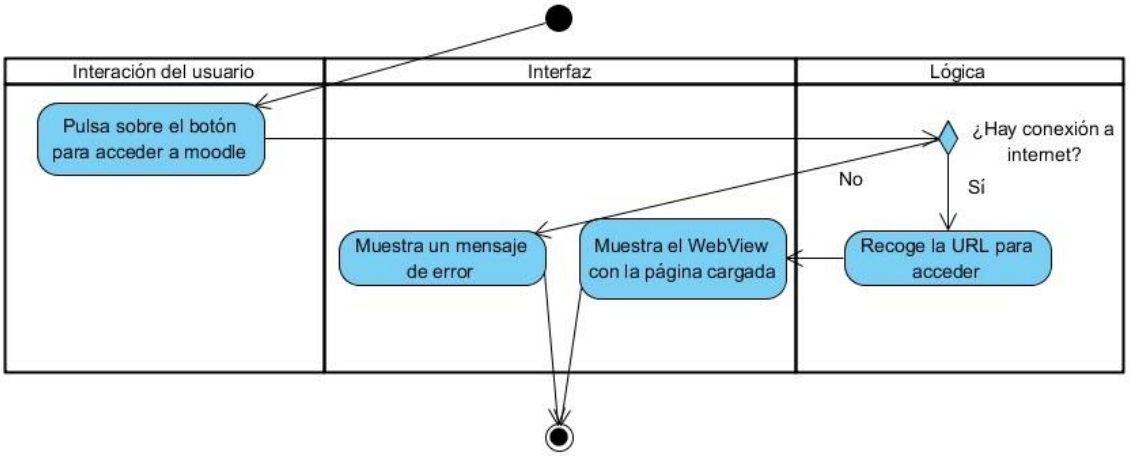


Ilustración 130. Diagrama de actividad - Cargar web

Como el caso de cargar archivos PDF, en la aplicación se pueden visualizar páginas web. En todos los casos el sistema es el mismo, comprueba la conexión a la red para ver si puede acceder a la página y recoge la URL de la página que se debe cargar.

Actualizar Datos

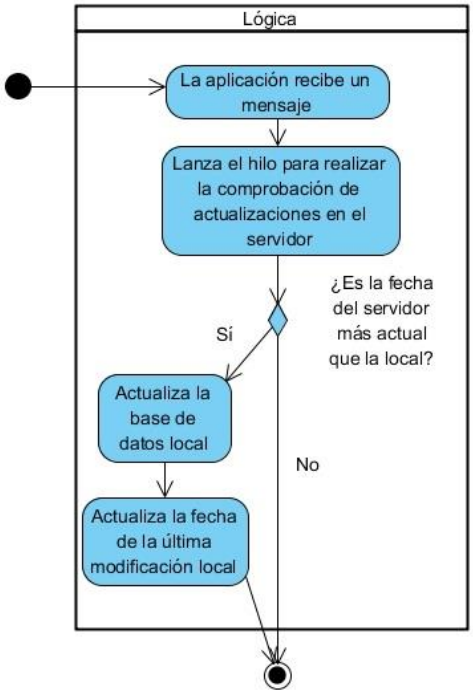


Ilustración 131. Diagrama de actividad - Actualizar datos

Este diagrama trata de explicar cuáles son las acciones a realizar para mantener la aplicación actualizada con la información que hay en el servidor. Con los mensajes que recibirá desde el servidor la aplicación comprobará que la última fecha de modificación local es anterior a la fecha que hay en el servidor lo que indicará que deberá actualizar la base de datos y finalmente actualizar la fecha de la última modificación local.

F.2. Aplicación web

Login

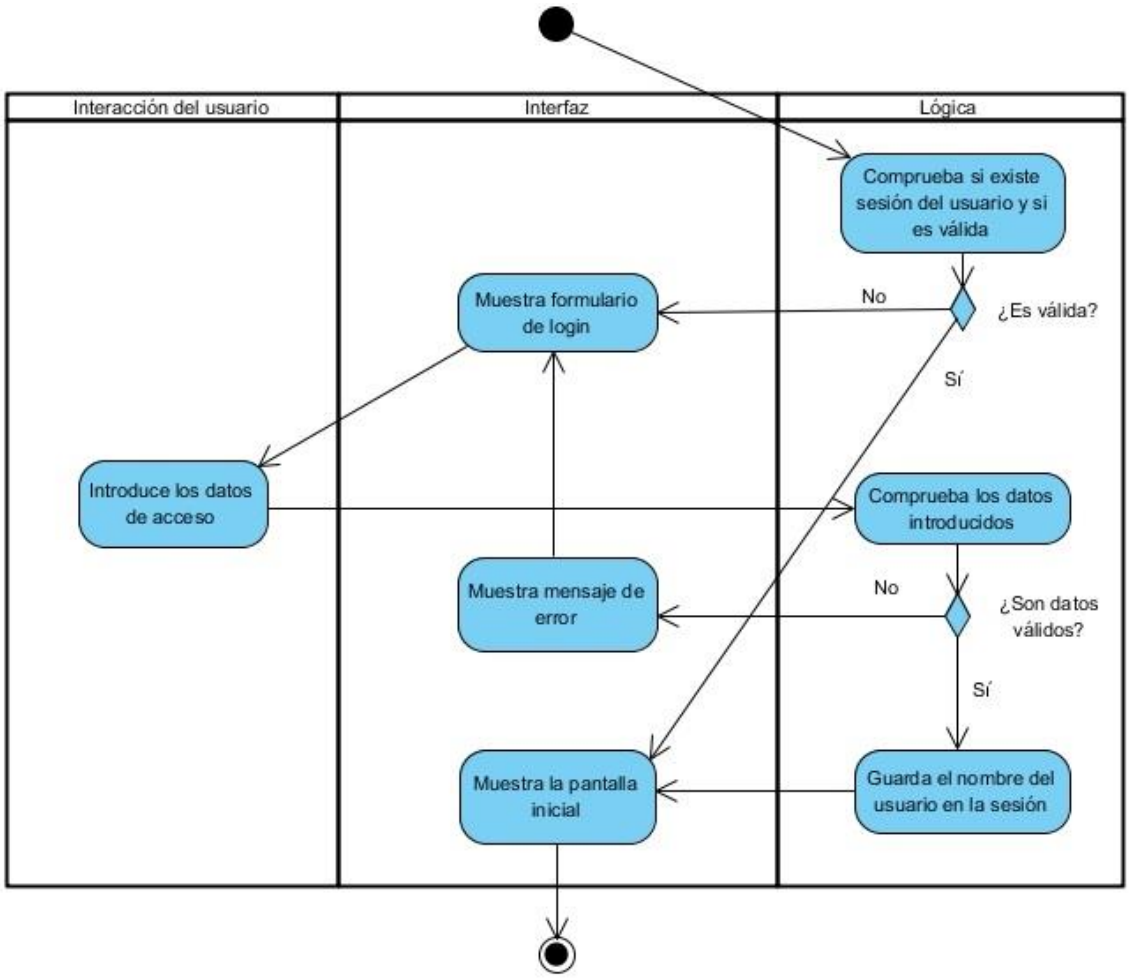


Ilustración 132. Diagrama de actividad – Login

Este diagrama muestra el proceso que se sigue para entrar al sistema de administración web de la aplicación. Primero comprueba si ya existía una sesión y si es válida, en caso de serlo va directamente a la pantalla principal, en caso de no serlo obliga al usuario a realizar el *login*. El usuario introducirá los datos para el inicio de sesión y el servidor se encargará de comprobar si esos datos son válidos. Si los datos son válidos el *login* se guardará en la sesión para validar al usuario en las diferentes páginas. En caso de que los datos no fueran válidos, el usuario vería un mensaje de error y no podrá avanzar a la pantalla de inicio de la aplicación.

Recuperar contraseña

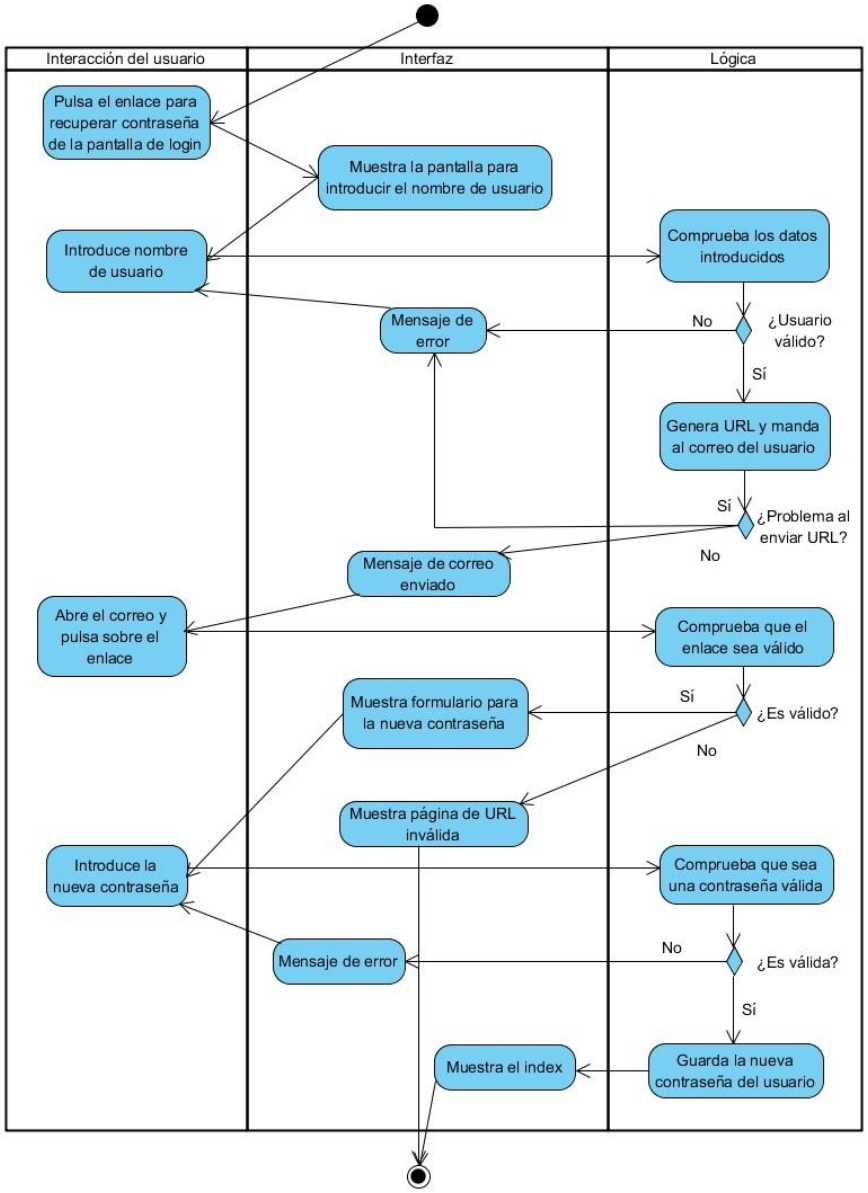


Ilustración 133. Diagrama de actividad - Recuperar contraseña

La ilustración anterior muestra el diagrama de actividad que sigue el usuario para poder restaurar su contraseña en caso de no recordarla. El usuario primero accede a la pantalla para solicitar el correo donde debe de introducir su nombre de usuario. Tras introducirlo, el sistema comprueba que es un nombre válido, y si lo es genera una URL aleatoria y la manda al correo que tiene el usuario asignado en el sistema. Si el usuario ha introducido un nombre de usuario erróneo o hay algún problema cuando se envía el correo el usuario verá un mensaje en la pantalla.

Continuando el proceso de la restauración, el usuario deberá acceder al correo y pulsar sobre la URL que se le ha mandado, en el caso de que haya pasado una hora o se modifique algún carácter de la URL se mandará al usuario a la pantalla de URL inválida. Tras esto podrá introducir una nueva contraseña en el sistema.

Añadir usuario

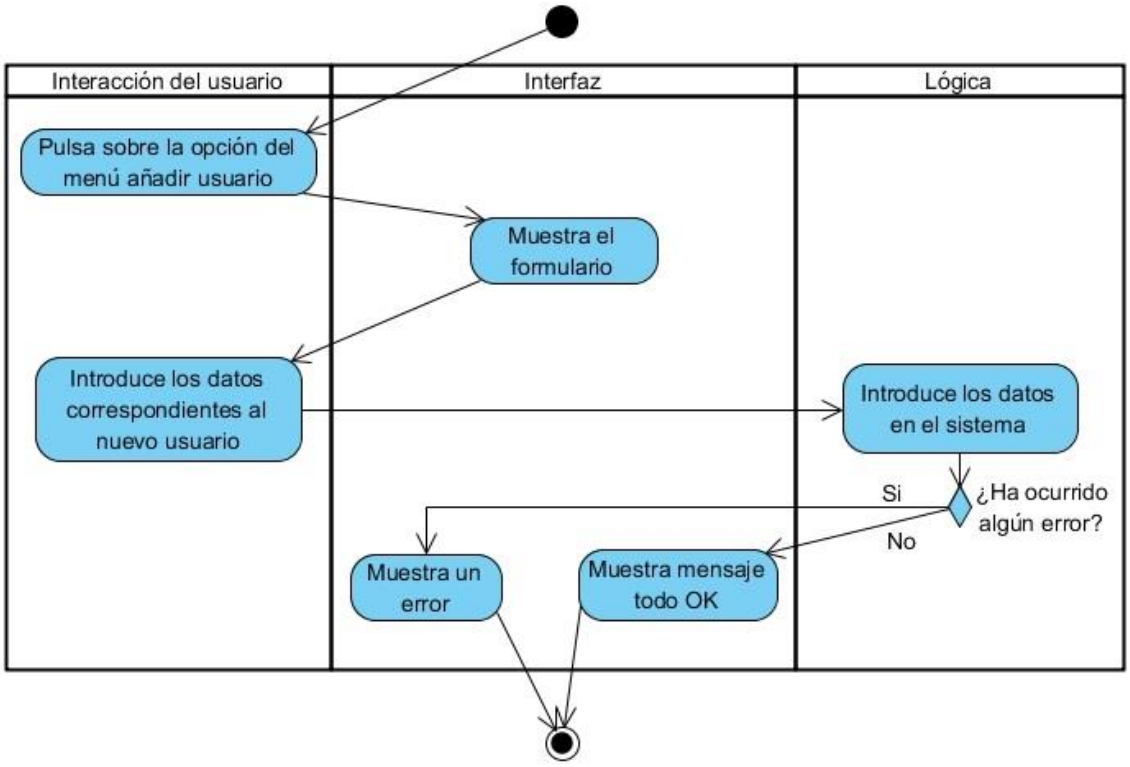


Ilustración 134. Diagrama de actividad - Añadir usuario

En este diagrama se ven los pasos a seguir que debe realizar un usuario para añadir un nuevo usuario que tenga acceso a la web. Básicamente desde el menú podrá pulsar la opción *añadir usuario* e irá directo al formulario para introducir los datos del nuevo usuario, cuando se los mande al servidor los introducirá en la base de datos y dependiendo del resultado mostrará un mensaje de error o de que todo ha ido bien.

Editar usuario

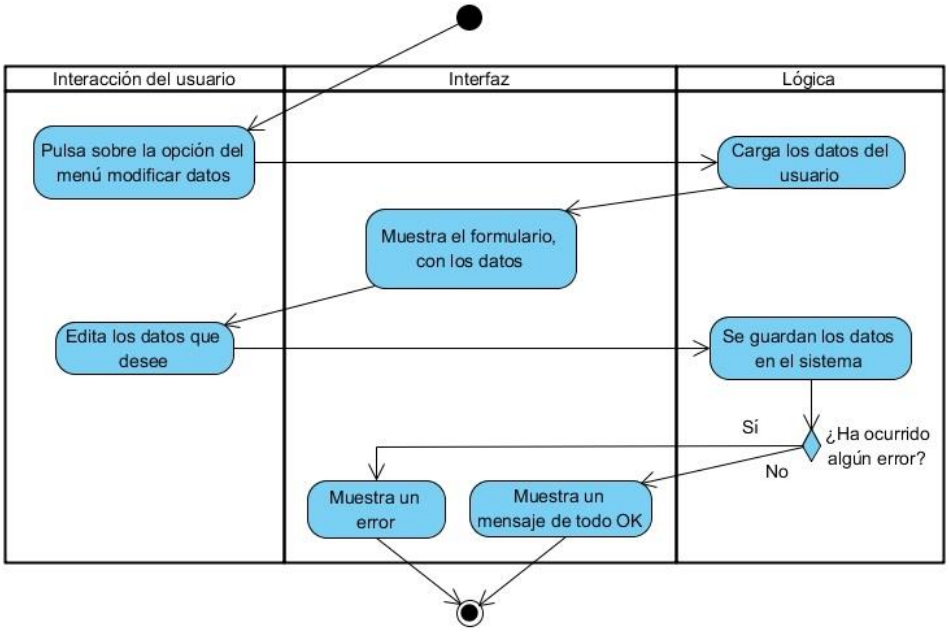


Ilustración 135. Diagrama de actividad - Editar usuario

El diagrama de actividad de editar usuario es muy similar al de añadir usuario, únicamente al mostrar el formulario añade los datos del usuario para que el usuario los edite.

Ver datos

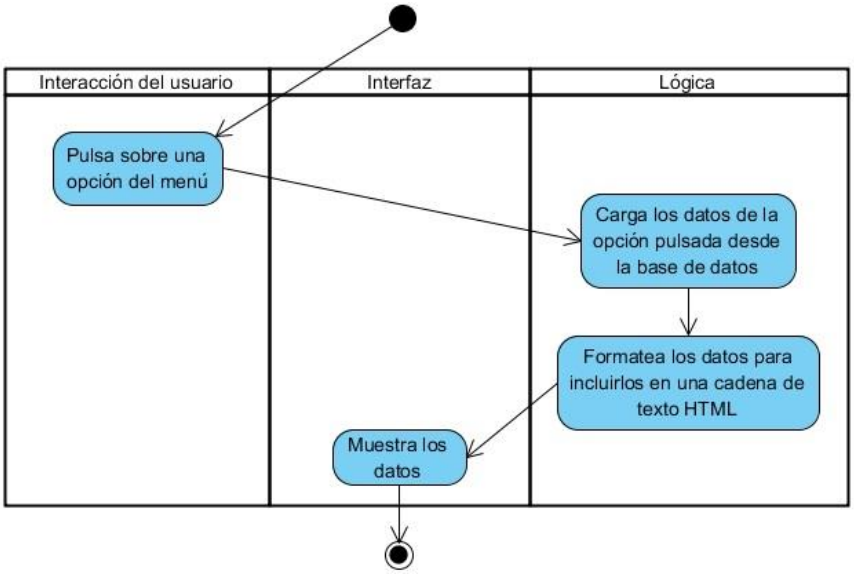


Ilustración 136. Diagrama de actividad - Ver datos

Ya que el proceso de mostrar los datos es el mismo para cada tabla de la base de datos, este diagrama muestra de forma genérica los pasos a seguir para mostrar los datos en la aplicación. El usuario accederá a los datos desde el menú, el sistema los cargara y prepara para mostrarlos en la interfaz.

Añadir datos

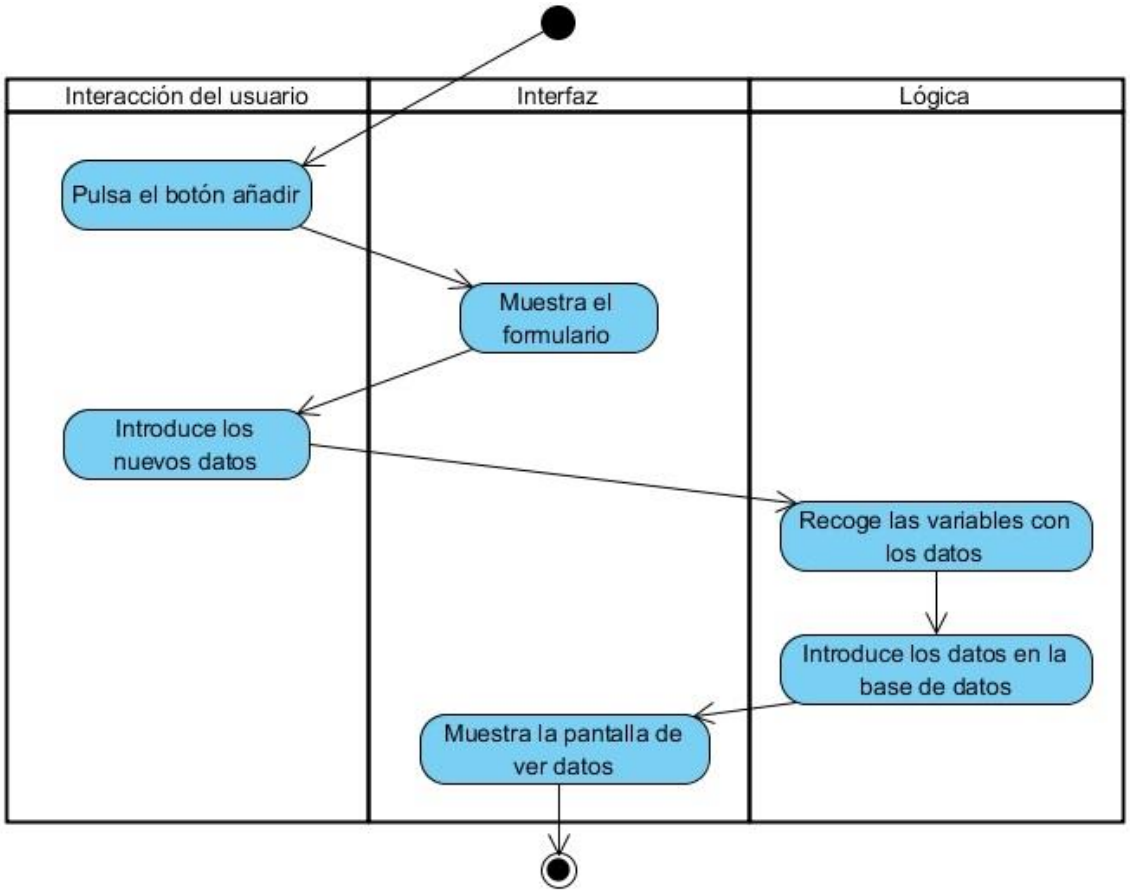


Ilustración 137. Diagrama de actividad - Añadir datos

Como en el apartado anterior, este diagrama muestra de forma genérica como se pueden añadir nuevos datos al sistema. Primero el usuario deberá pulsar el botón añadir, presente en las páginas de visualización de datos. Tras esto se mostrara el formulario donde el usuario introducirá los datos y los enviara al servidor, este recogerá los datos y los introducirá en la base de datos, tras todo esto finalmente volverá a la pantalla de visualización de los datos.

Editar datos

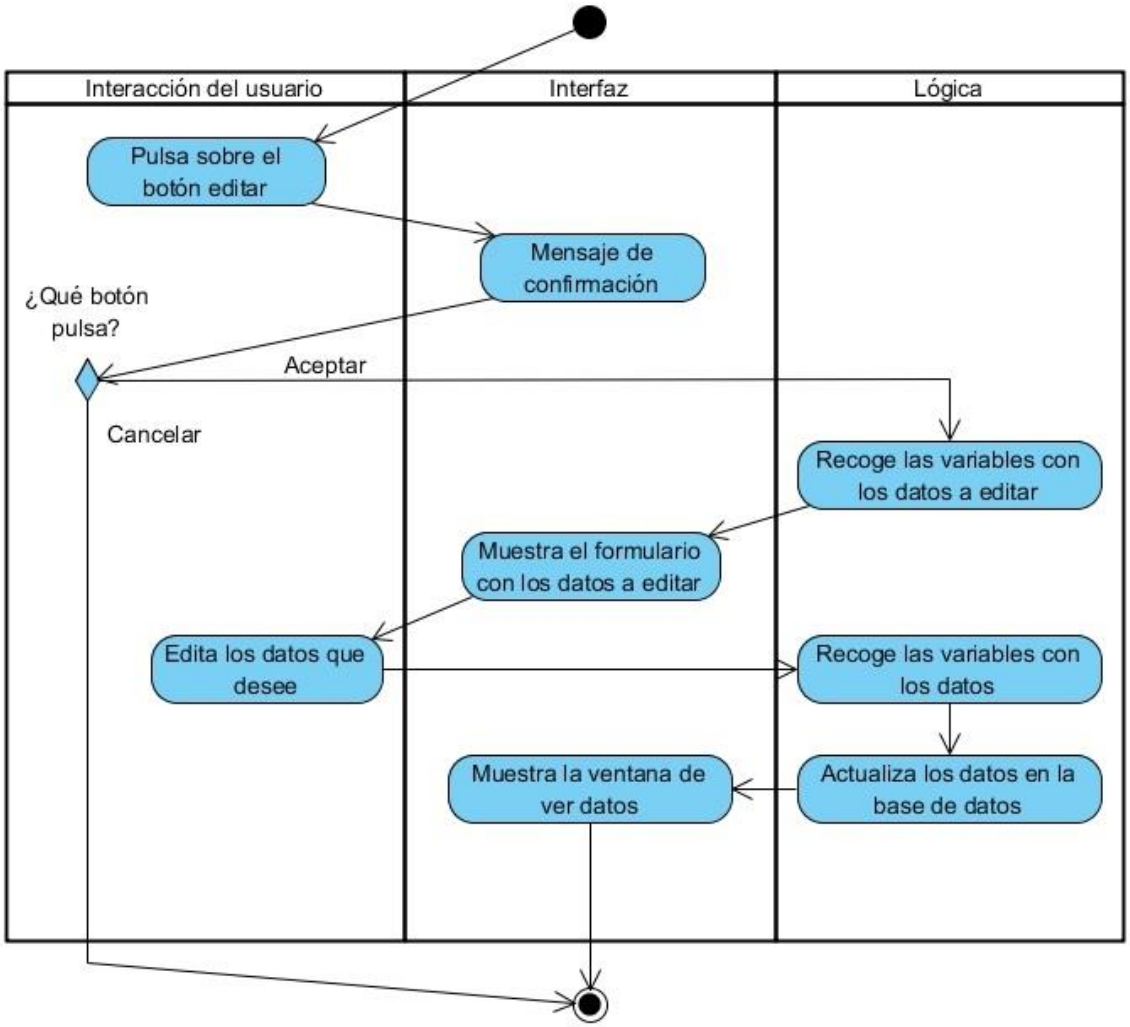


Ilustración 138. Diagrama de actividad - Editar datos

Para editar datos el usuario deberá pulsar sobre el botón editar de la fila correspondiente de datos que desea editar y confirmar que desea realizar una modificación en la ventana emergente que le aparecerá, tras esto el servidor recogerá los datos que se van editar y los mostrará en un formulario donde el usuario podrá modificarlos y luego volver a mandarlos al servidor para que los guarde.

Eliminar datos

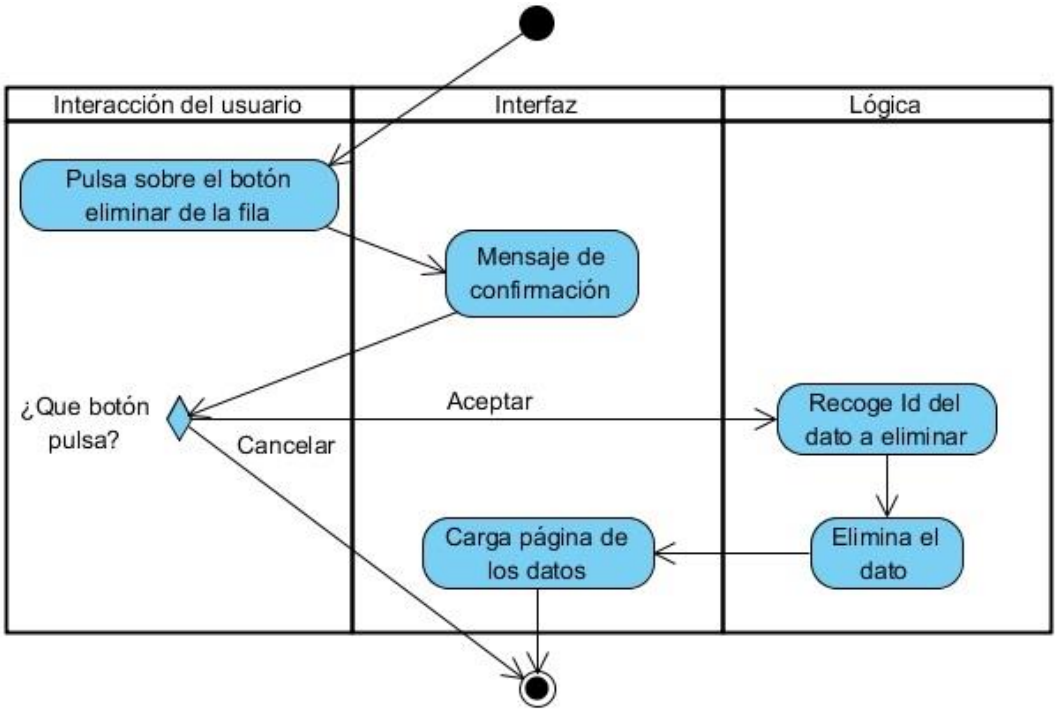


Ilustración 139. Diagrama de actividad - Eliminar datos

La eliminación de datos se realiza de una forma sencilla, el usuario simplemente pulsa sobre el botón eliminar de la fila de datos que desea borrar, acto seguido el usuario verá una ventana emergente donde deberá confirmar que desea borrar los datos, si confirma el servidor recibirá el identificador del dato a borrar y lo eliminará del sistema.

Anexo G: Diagramas de secuencia

G.1. Aplicación cliente

Inicio

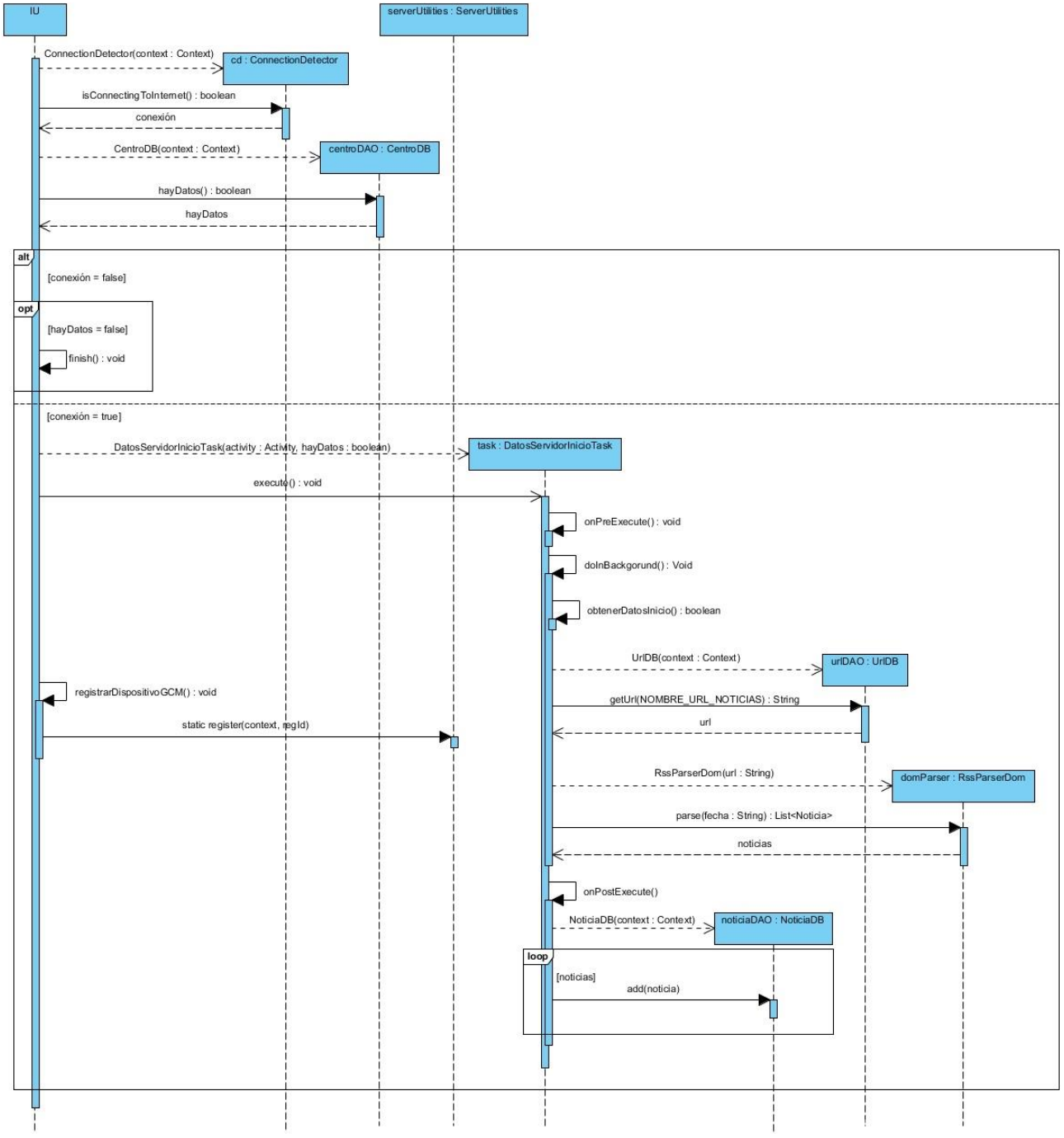


Ilustración 140. Diagrama de secuencia – Inicio

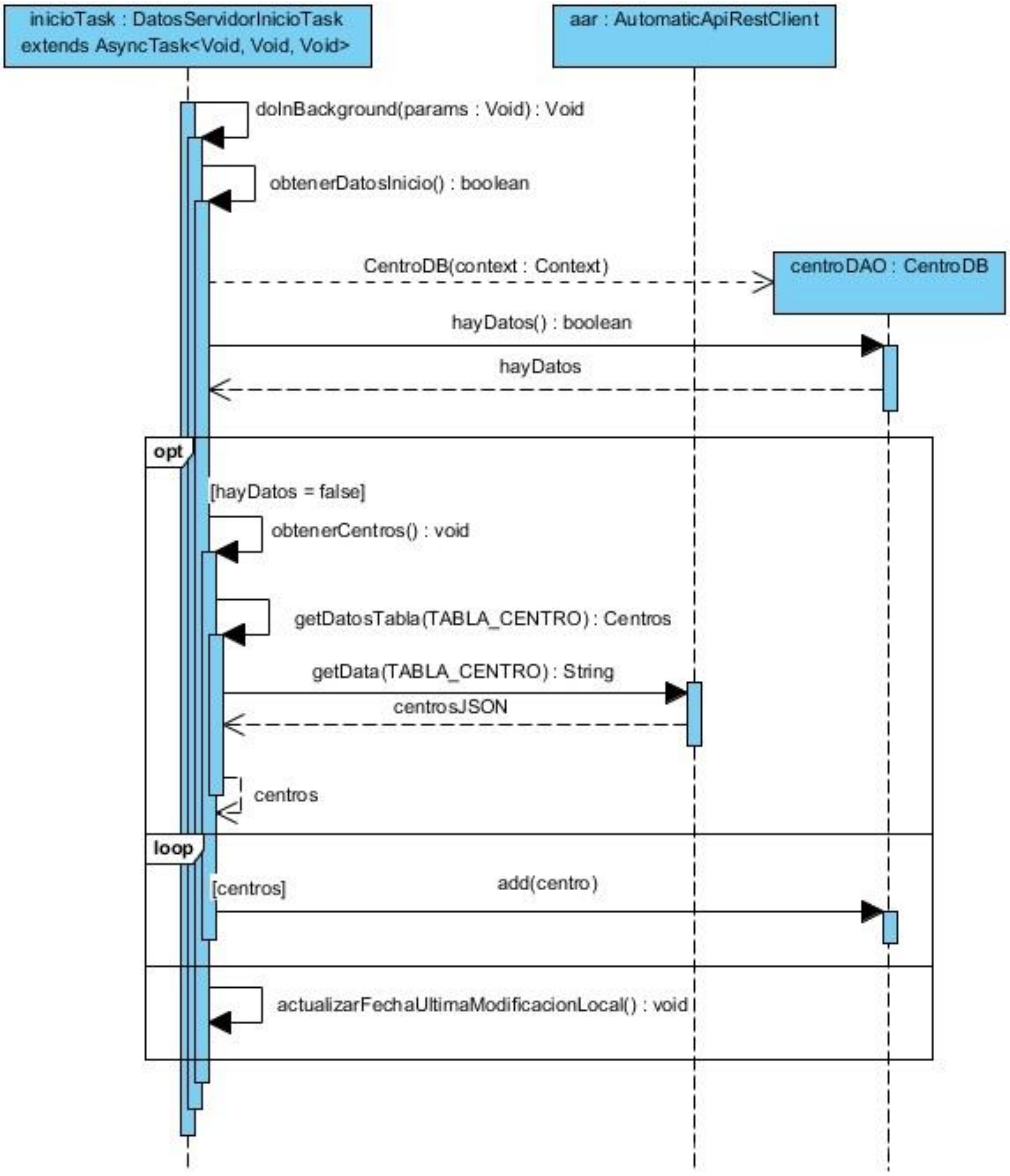


Ilustración 141. Diagrama de secuencia - Obtener datos inicio

El inicio de la aplicación será la primera secuencia de operaciones que se realizará, como se puede ver en este proceso no interviene el usuario. Este diagrama contiene los métodos de las clases que se llaman en el proceso, y como se puede ver el proceso depende de si hay conexión a internet y si hay datos en la base de datos local, una vez finalizado acabará mostrándose la pantalla principal.

Para poder realizar un diagrama más visible, se ha separado el proceso en dos. El primer diagrama muestra el proceso general y el segundo entra en detalle en el método `obtenerDatosInicio` de la clase `DatosServidorInicioTask`. El segundo diagrama muestra solo una parte de todo el método ya que el mismo proceso que se ha realizado para los centros también se realiza para las titulaciones, servicios, profesores, URLs y fechas del calendario, para realizar esto únicamente es necesario cambiar la clase `CentroDB` y el método `obtenerCentros` por los correspondientes, en el caso de las titulaciones por ejemplo `TitulacionDB` y `obtenerTitulaciones`.

Carga Asignaturas del Servidor

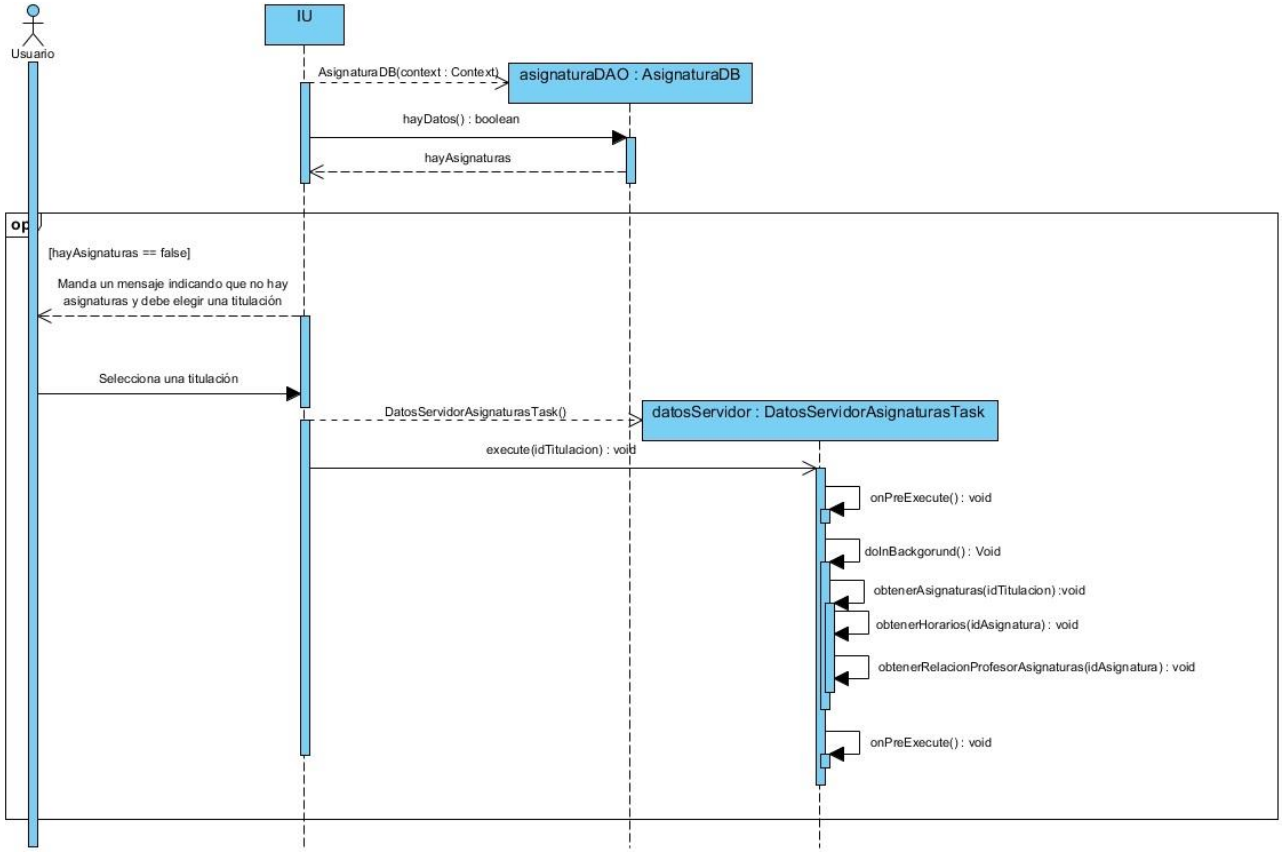


Ilustración 142. Diagrama de secuencia - Carga asignaturas del servidor

En este diagrama se puede ver como se obtienen las asignaturas desde el servidor, primero se comprobará si hay datos en la aplicación sobre estas, y en caso de que no haya preguntará al usuario la titulación que desee, para obtener sus asignaturas lanzando un hilo de ejecución con el identificador de la titulación que conectara con el servidor. Esto se realizará durante la carga de la interfaz para elegir las asignaturas del perfil.

El método `obtenerAsignaturas` es un proceso muy similar al método `obtenerDatosInicio`, salvo que en este caso además de obtener los datos de las asignaturas, se van recogiendo los datos de los horarios y las relaciones entre profesores y asignaturas en el propio método.

El diagrama mostrado se ejecutará antes de mostrar la pantalla para la elección de las asignaturas por lo que una vez lanzado el hilo de ejecución para conectar con el servidor la ejecución continuará mostrando la pantalla para la creación o edición del perfil.

Crear/Editar perfil

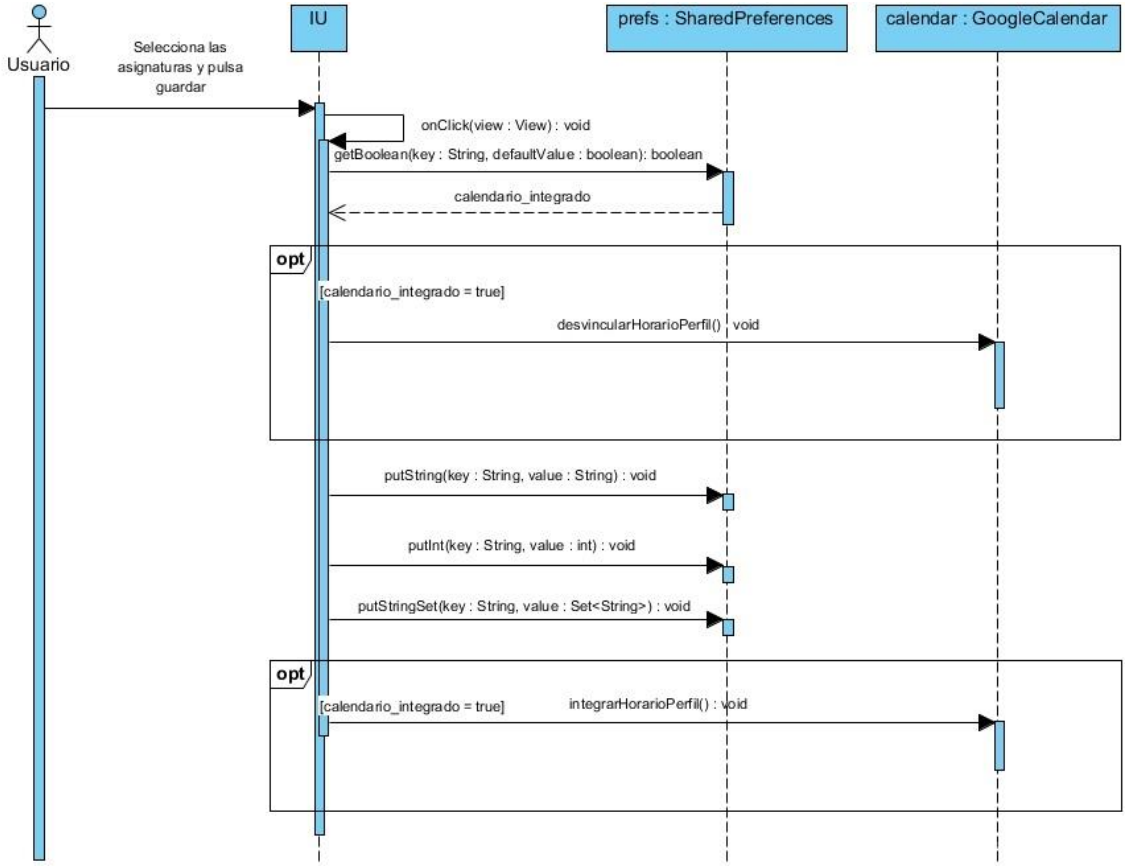


Ilustración 143. Diagrama de secuencia - Crear/Editar perfil

Los pasos a seguir para la creación o la edición del perfil son los mismos, el usuario deberá seleccionar las asignaturas que desee elegir para tener acceso rápido a ellas y pulsará sobre el botón guardar, a partir de aquí se comprobara si el usuario tenía integrado el horario en Google Calendar, para que en caso de ser así se borre y al final del proceso se vuelva a importar el calendario con las asignaturas elegidas, estos métodos se desarrollarán en diagramas posteriores. Entre los procesos realizados con Google Calendar, si se deben realizar, se guardaran las variables necesarias para guardar que el usuario tiene un perfil, el identificador de la titulación y la lista de identificadores de las asignaturas seleccionadas.

Google Calendar

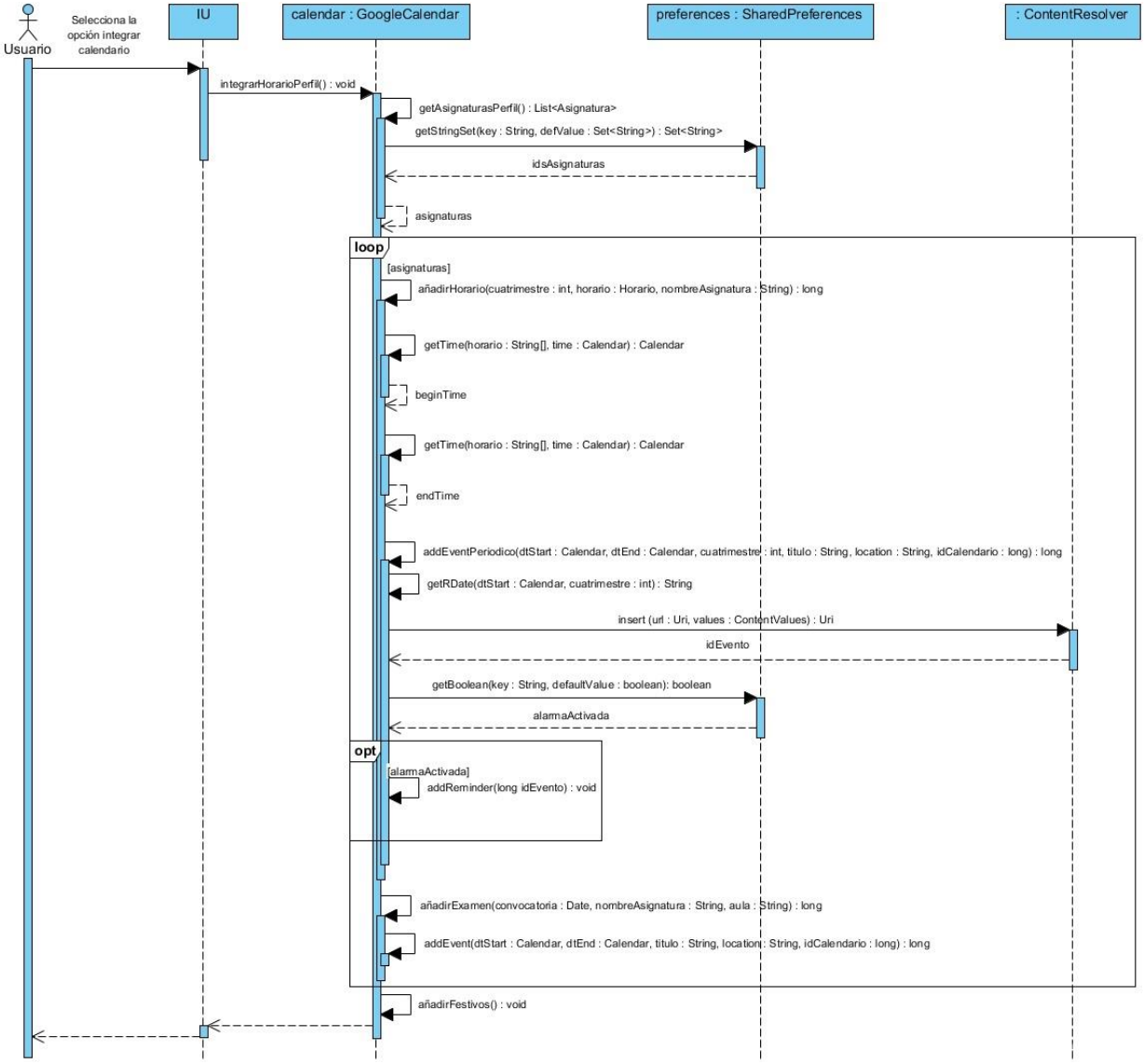


Ilustración 144. Diagrama de secuencia - Integrar calendario

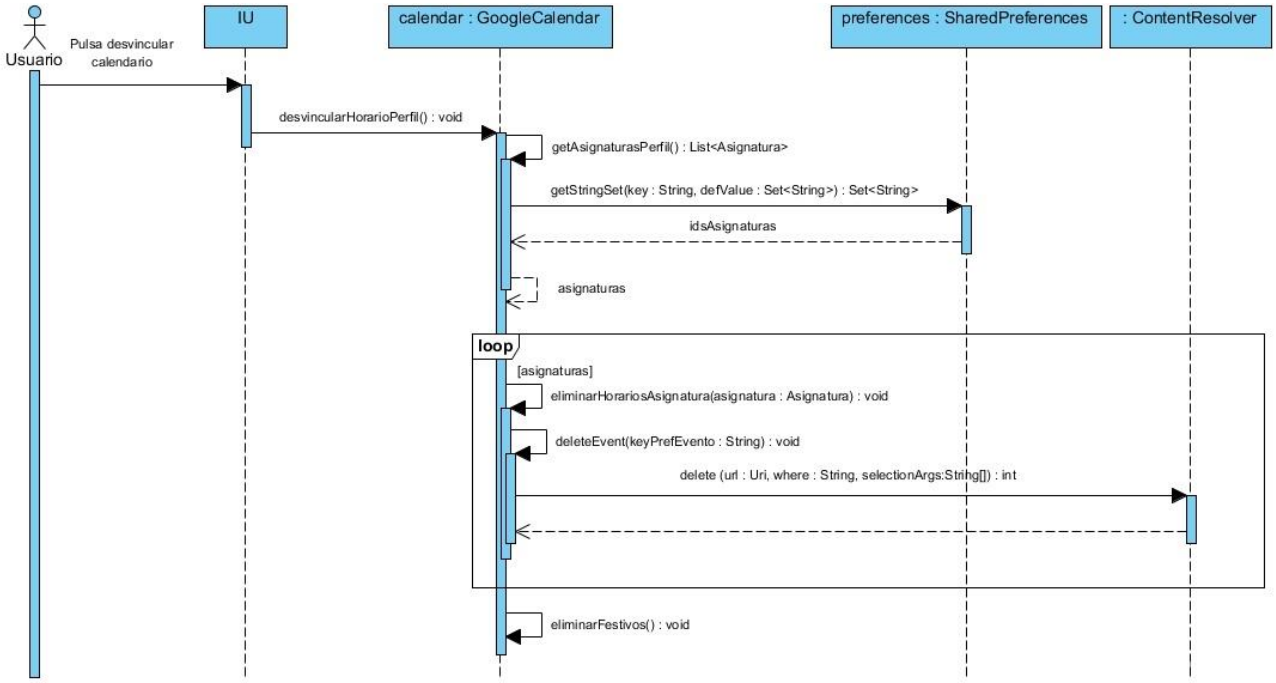


Ilustración 145. Diagrama de secuencia - Desvincular calendario

Los diagramas que contiene este apartado corresponden a las acciones realizadas con Google Calendar, en ambos se observa que una vez el usuario interactúa con la interfaz, la clase `GoogleCalendar` contiene los métodos para realizar las acciones pertinentes con los métodos internos que hay implementados en esa clase.

Ver asignatura

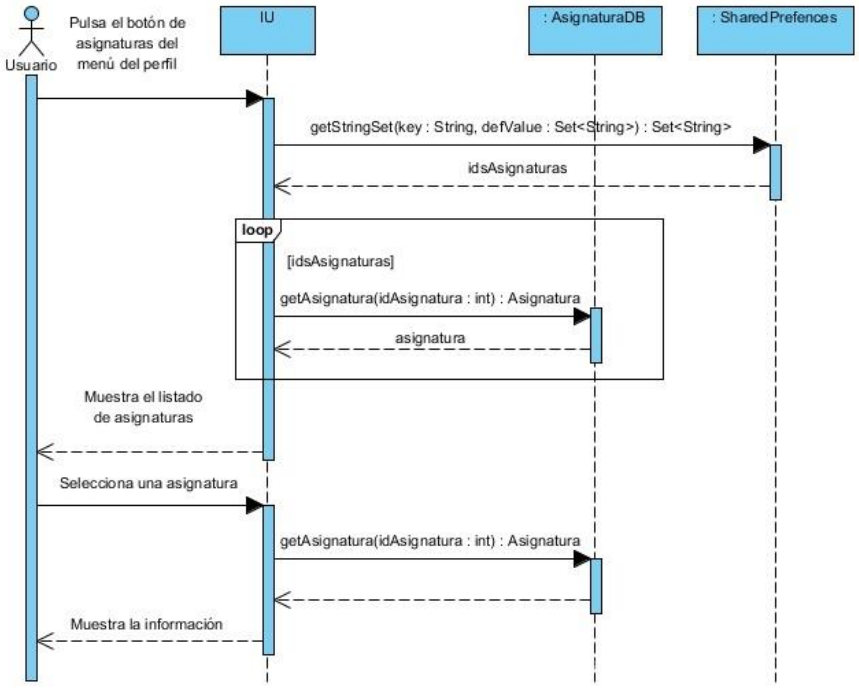


Ilustración 146. Diagrama de secuencia - Ver asignatura

Para que el usuario pueda ver las asignaturas que haya seleccionado en su perfil, se recogerán de las preferencias los identificadores de estas y se irán recogiendo las mismas en un listado para imprimirlas en la pantalla del usuario. Cuando el usuario vea el listado podrá elegir una y esta se recogerá de la base de datos y se imprimirá su información.

Ver noticia

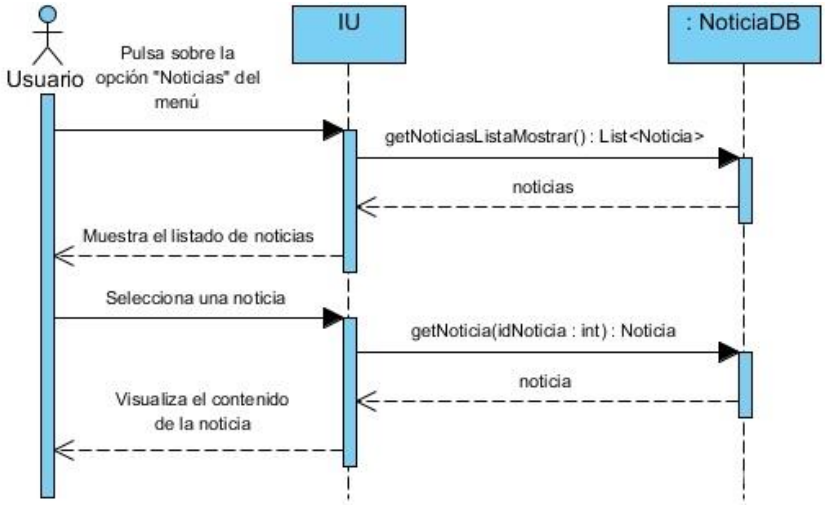


Ilustración 147. Diagrama de secuencia - Ver noticia

Cuando el usuario pulse sobre la opción “Noticias” en la interfaz, esta se encargara de llamar al método `getNoticiasListaMostrar` para recoger las noticias que debe mostrar por pantalla, desde ahí el usuario tendrá la posibilidad de seleccionar una de ellas y ver su contenido en otra pantalla.

Ver Servicio

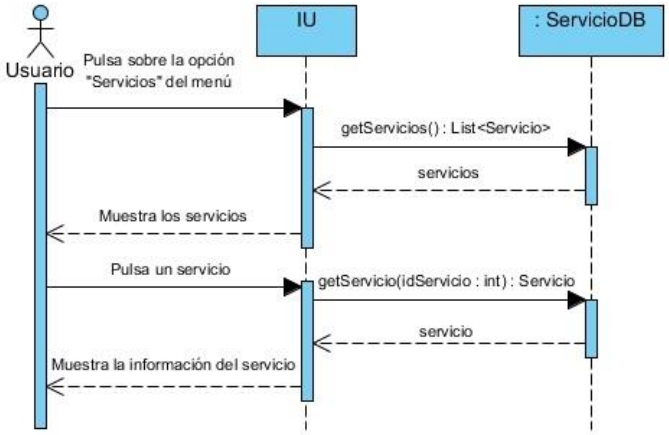


Ilustración 148. Diagrama de secuencia - Ver servicio

Este diagrama es muy similar al caso de ver noticia, por eso en los diagramas de actividad ambos están expuestos en un único diagrama. Aquí la diferencia se sitúa en el acceso a los datos, que se accede a través de la clase `ServicioDB` y luego la interfaz se encargara de mostrar la información.

Ver titulación

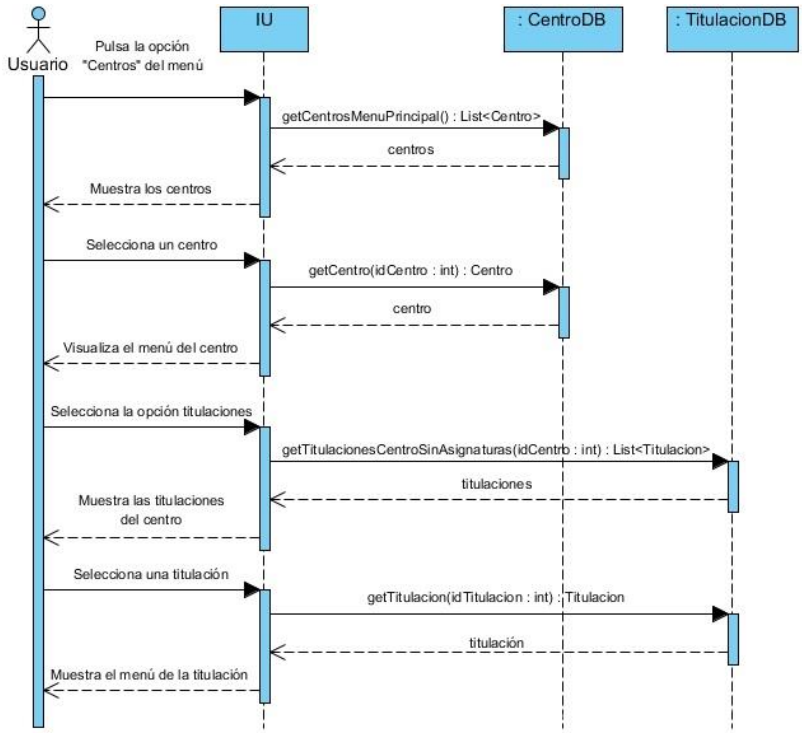


Ilustración 149. Diagrama de secuencia - Ver titulación

Para tener acceso a la información de una titulación existen dos formas, el buscador que se expondrá más adelante y la que muestra este diagrama. Primero se visualizarán los centros para poder seleccionar uno de ellos, tras seleccionar uno seleccionará ver las titulaciones del centro y tendrá acceso a las titulaciones pertenecientes al centro y por último pulsará una titulación para poder acceder a su información.

Ver profesor

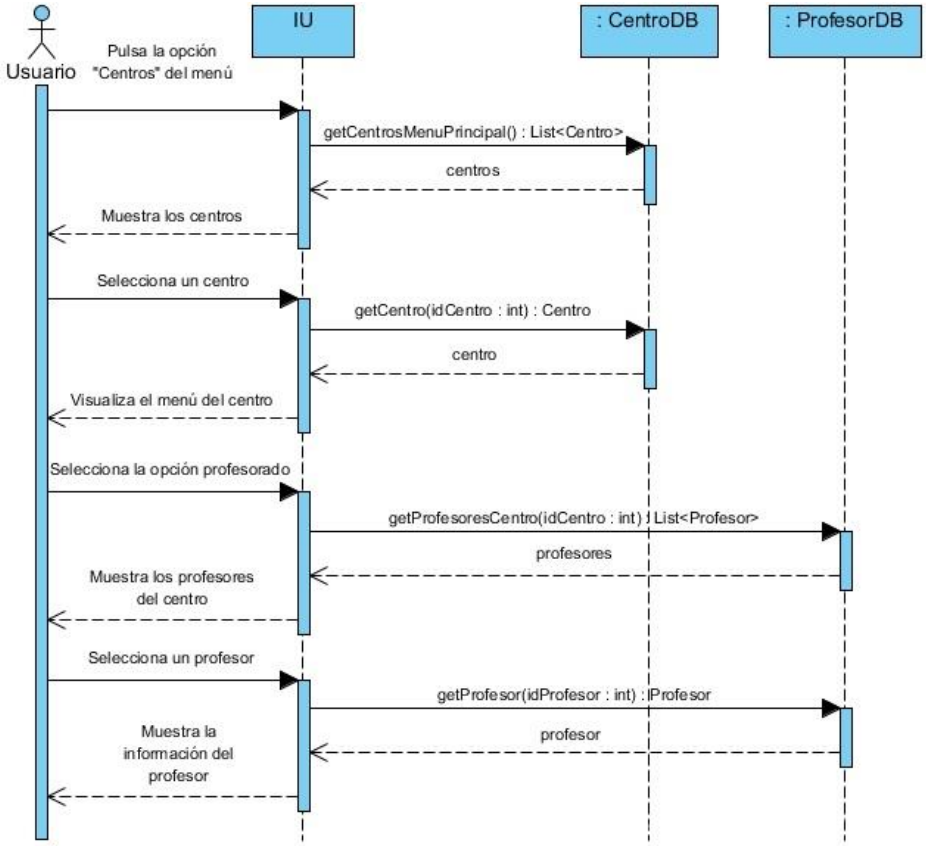


Ilustración 150. Diagrama de secuencia - Ver profesor

Como se puede ver este diagrama es similar al diagrama para ver una titulación. En este caso se accede a la información de los profesores desde `ProfesorDB` y de la misma forma que en el caso de la titulación, la interfaz se encarga de presentar la información al usuario.

Ver información de contacto de un centro

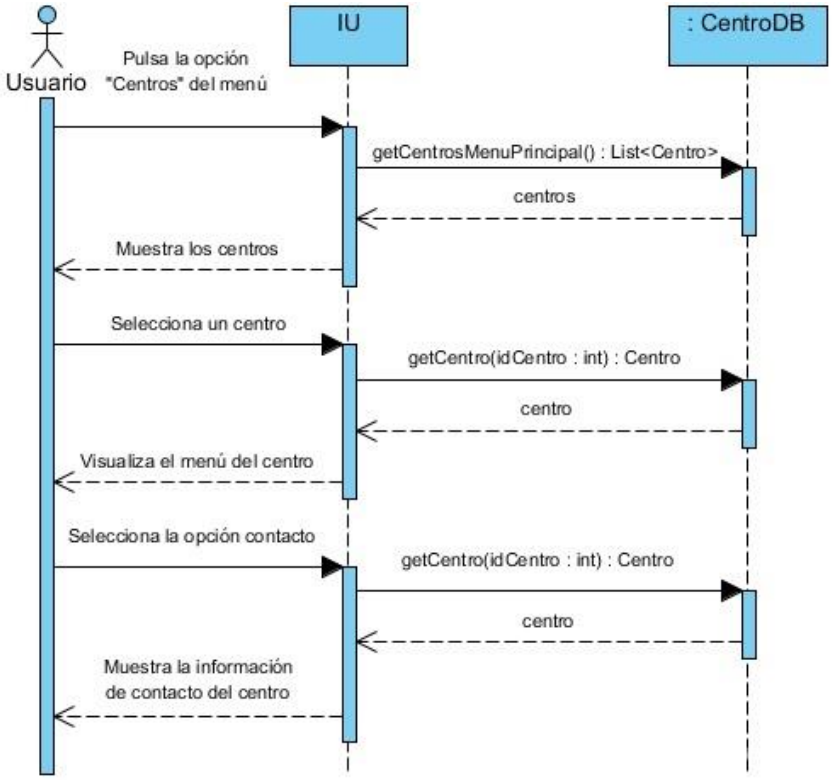


Ilustración 151. Diagrama de secuencia - Ver información de contacto

Como en los casos anteriores, primero se obtiene la información de los centros para mostrarlos por pantalla y finalmente el usuario podrá seleccionar ver los datos de contacto del centro. El proceso de recoger el centro dos veces se realizara principalmente por que la interfaz son dos pantallas distintas.

Ver plano de un centro

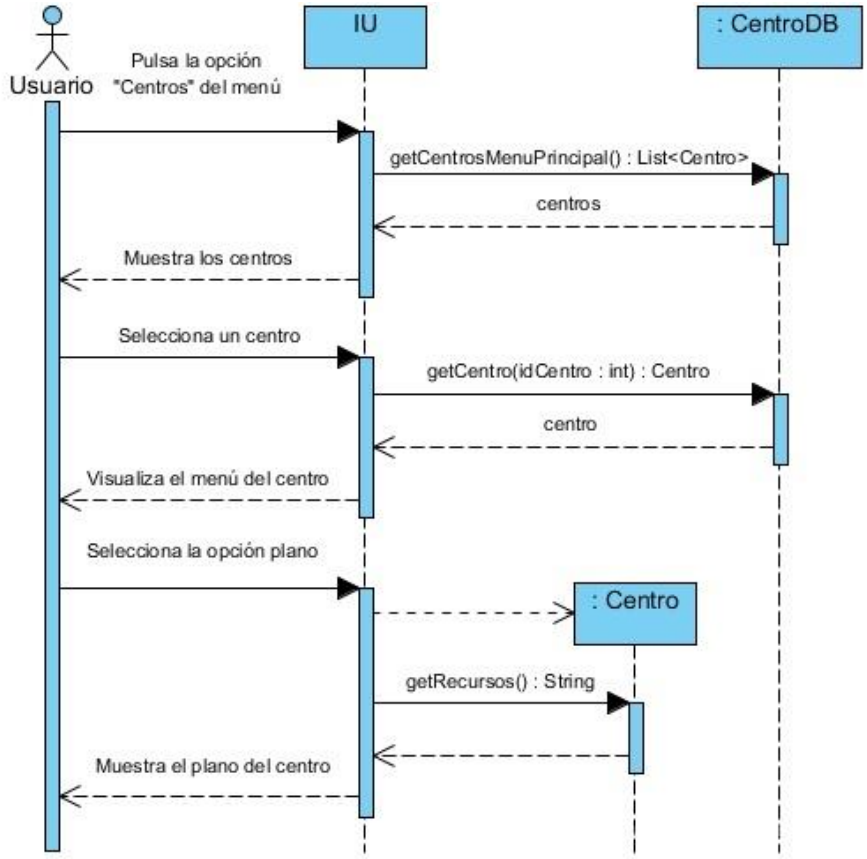


Ilustración 152. Diagrama de secuencia - Ver plano

En este caso cuando el usuario llega a la opción de visualizar el plano del centro, para acceder a él usa el método `getRecursos` de la clase `Centro`, que contiene el nombre que se usa para acceder a los recursos desde la aplicación, tras esto la interfaz se encargará de mostrar el plano en pantalla para el usuario.

Buscador

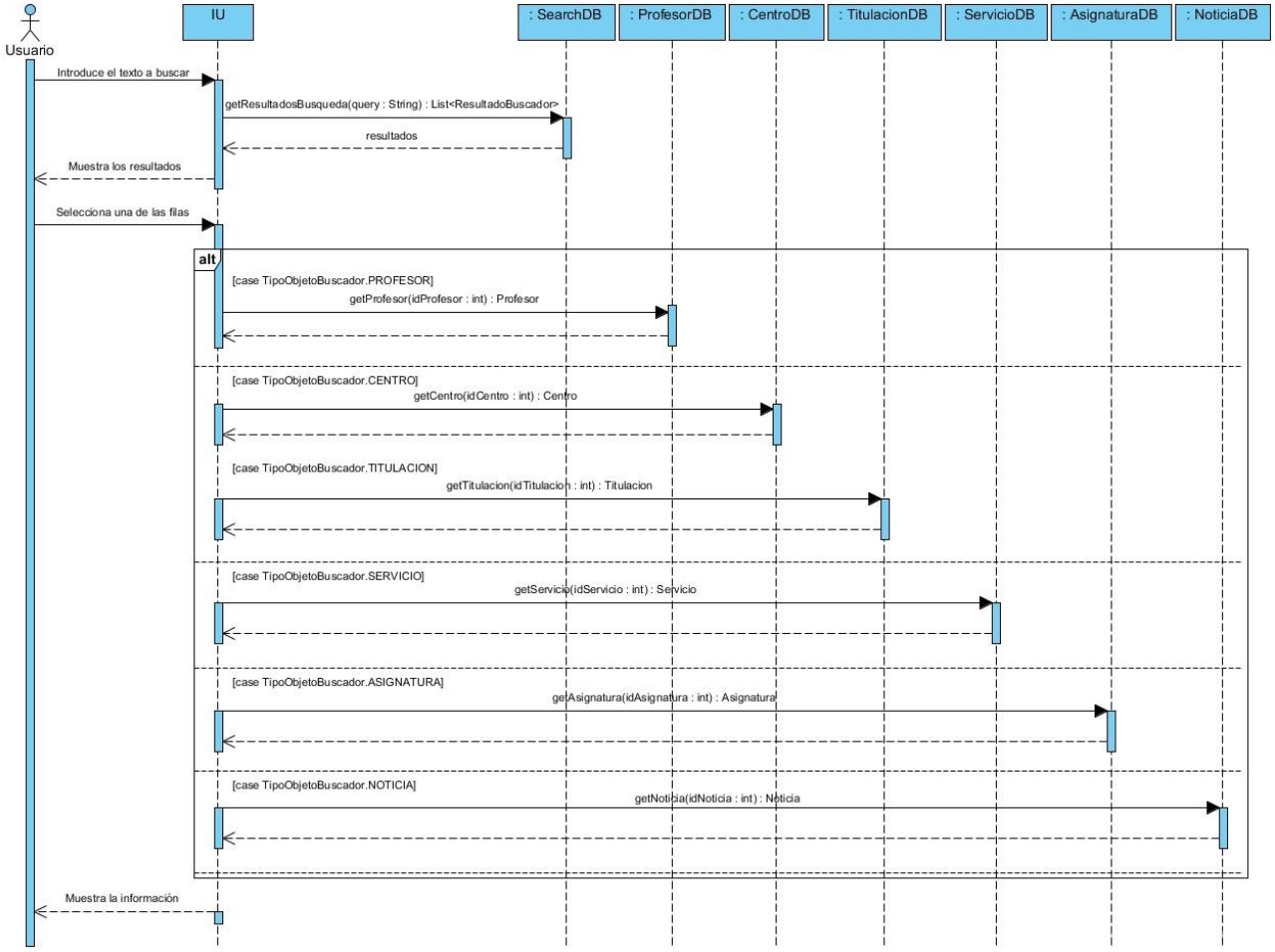


Ilustración 153. Diagrama de secuencia – Buscador

El buscador hace uso de la clase `SearchDB` para realizar la búsqueda de la cadena introducida por el usuario, los resultados los devuelve en una lista del objeto `ResultadoBuscador` que contiene de manera interna el tipo enumerado `TipoObjetoBuscador`, el usuario podrá seleccionar una de las opciones y con el identificador del objeto y dependiendo del tipo que sea realiza una llamada u otra para mostrar la información.

Mapa

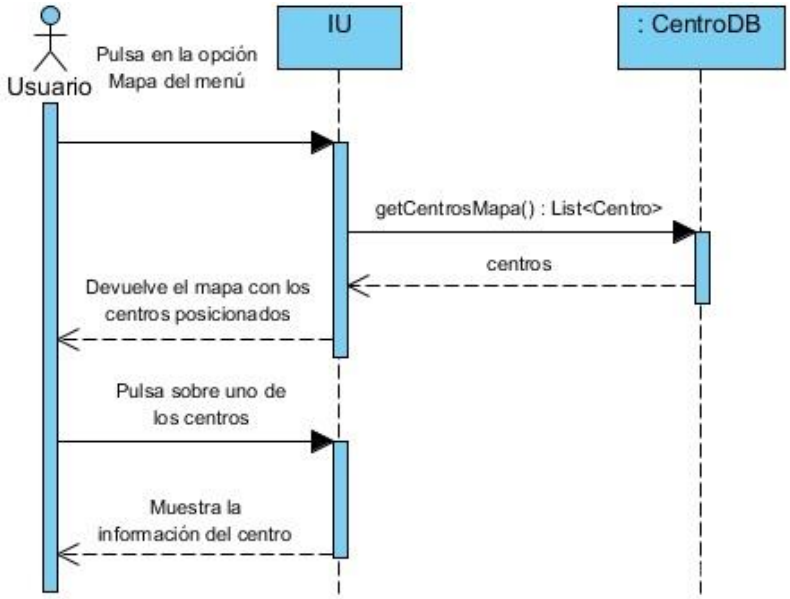


Ilustración 154. Diagrama de secuencia – Mapa

Para mostrar los centros en el mapa, el sistema hará uso del método `getCentrosMapa`, una vez se cargue la vista con los centros que haya devuelto el método el usuario podrá pulsar sobre uno de los centros y el sistema deberá coger uno de los centros de la lista devuelta por el método y cargar su información en la vista para que lo vea el usuario.

Cargar PDF

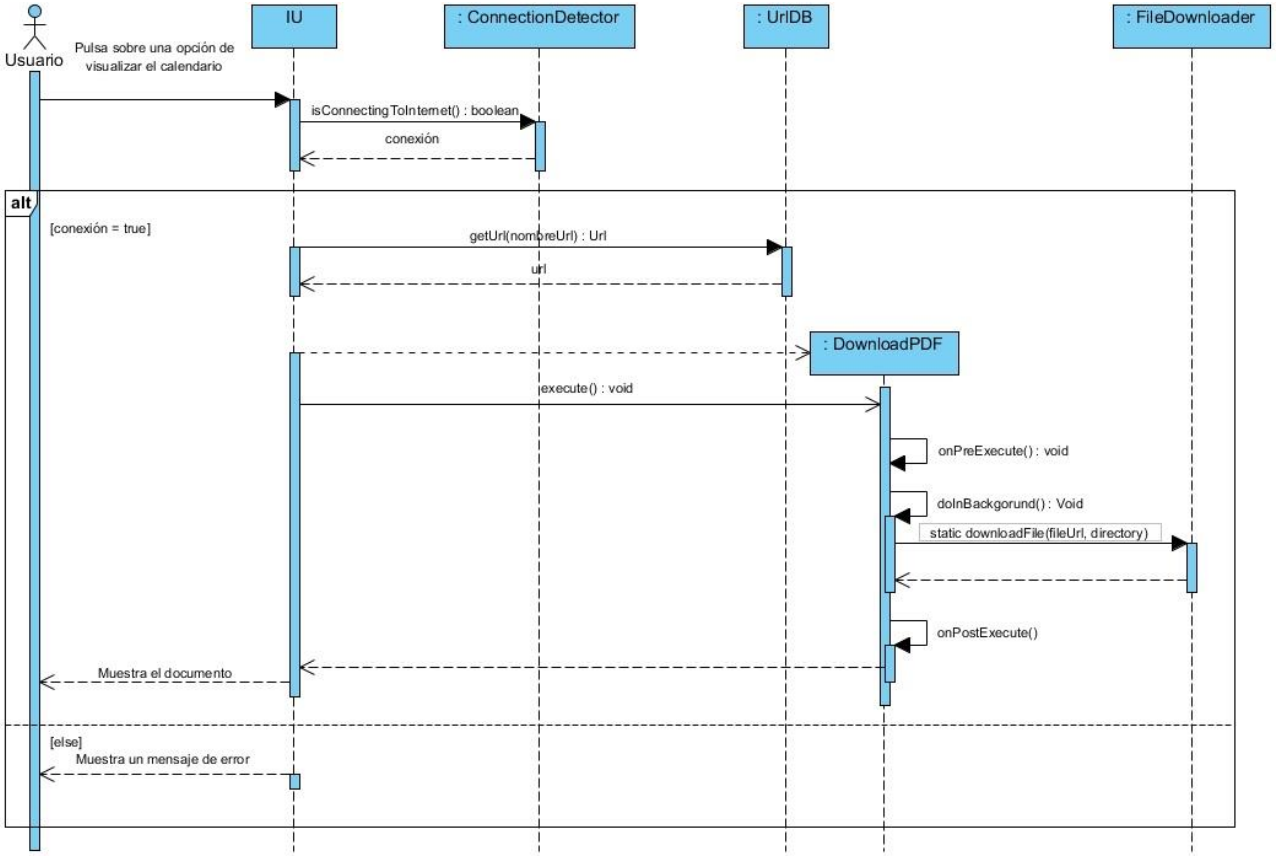


Ilustración 155. Diagrama de secuencia - Cargar PDF

Para realizar la carga de un fichero PDF, el sistema primero deberá comprobar que el usuario dispone de conexión a internet, en caso de no tenerla mostrará un mensaje de error al usuario. Para tener acceso al fichero primero deberá obtener la URL en este caso del calendario, una vez conseguido esto simplemente se lanza el hilo con la URL y cuando este finaliza se abre el fichero para que lo visualice el usuario.

Cargar web

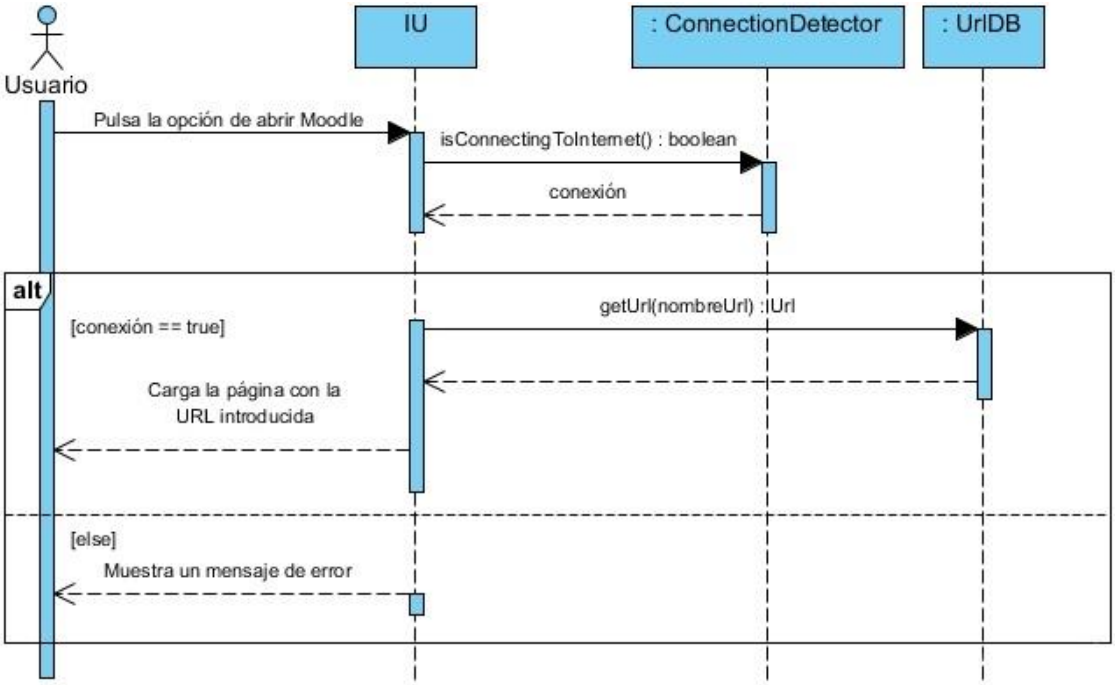


Ilustración 156. Diagrama de secuencia - Cargar web

Para cargar una página web, primero comprueba la conexión y después recoge la URL que desea cargar, para mostrar la página al usuario.

Actualizar datos

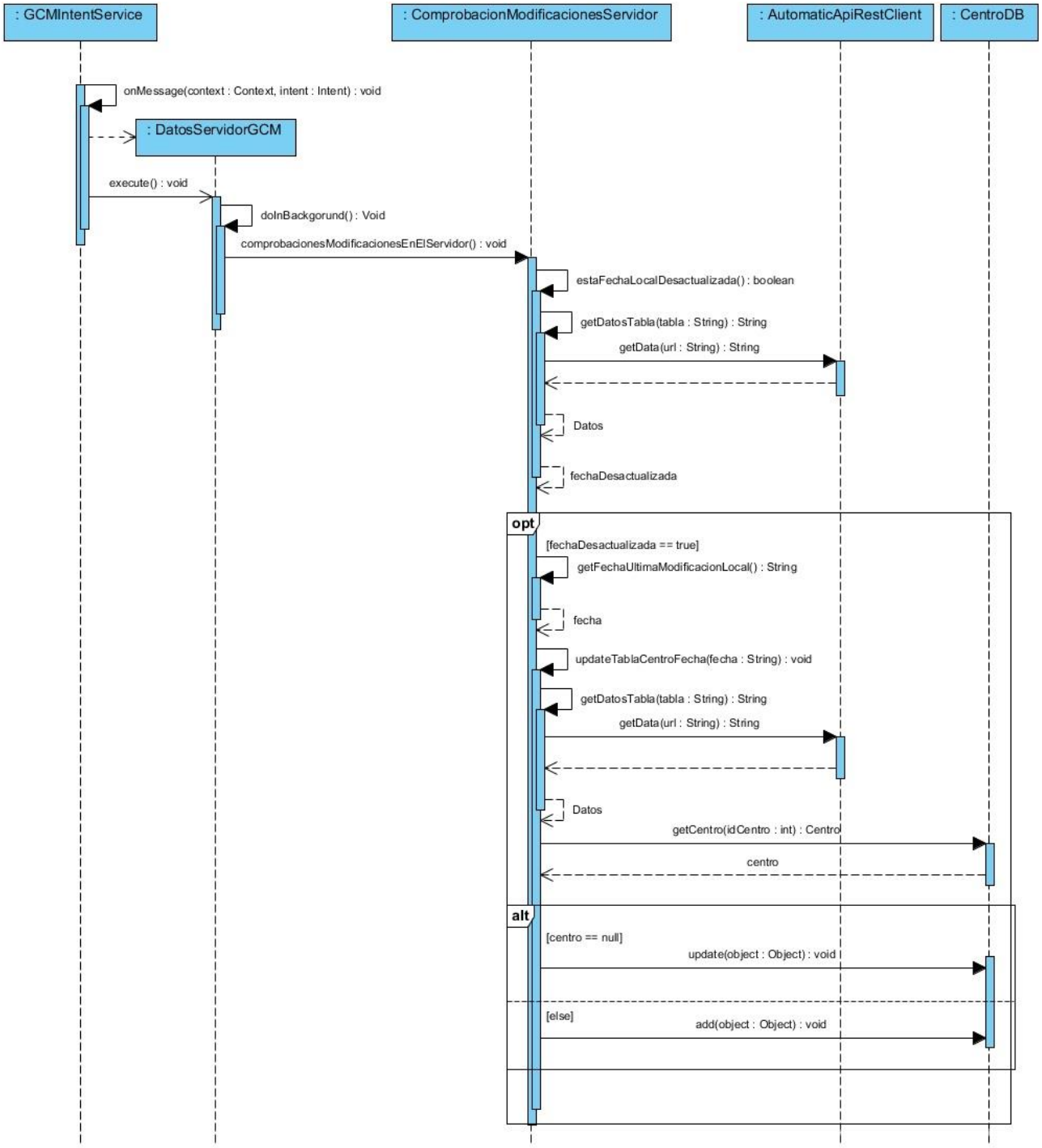


Ilustración 157. Diagrama de secuencia - Actualizar datos

Cuando la aplicación recibe un mensaje desde el servidor GCM de google, indica que se ha realizado una modificación en el servidor del sistema. Cuando esto ocurre se llama al método `onMessage` y desde ahí se ejecuta un hilo que se encarga de realizar las comprobaciones, primero comprueba que la fecha de la última modificación local es anterior a la del servidor y entonces es cuando comienza a llamar a los métodos para realizar la actualización, para realizar esto recoge los datos del servidor y comprueba si ya estaban en la aplicación, y si fuera así realiza una actualización y en caso contrario añade el datos.

En el diagrama que se muestra únicamente se ve el proceso de actualización de la tabla centro, para el resto de tablas se llamaría al resto de métodos privados que contiene la clase `ComprobacionModificacionesServidor` que conectan con el servidor y añaden o actualizan los datos de la misma forma que se realiza con los centros en esta ilustración.

G.2. Aplicación web

Login

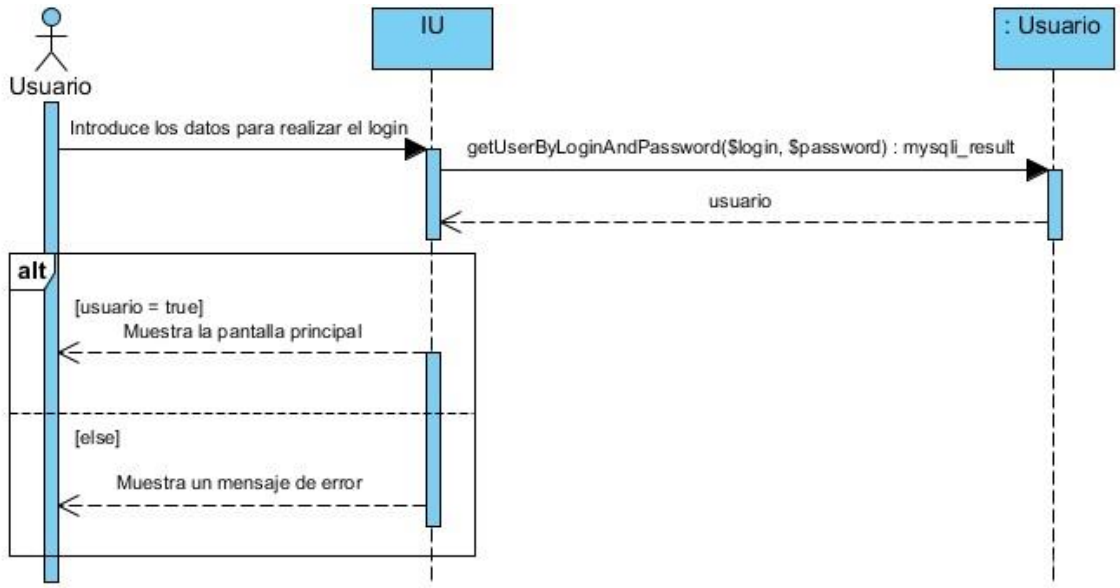


Ilustración 158. Diagrama de secuencia – Login

Este diagrama muestra cómo, cuando el usuario introduce sus datos de *login*, se comprueban los datos del usuario y dependiendo del resultado se realiza una acción u otra.

Recuperar contraseña

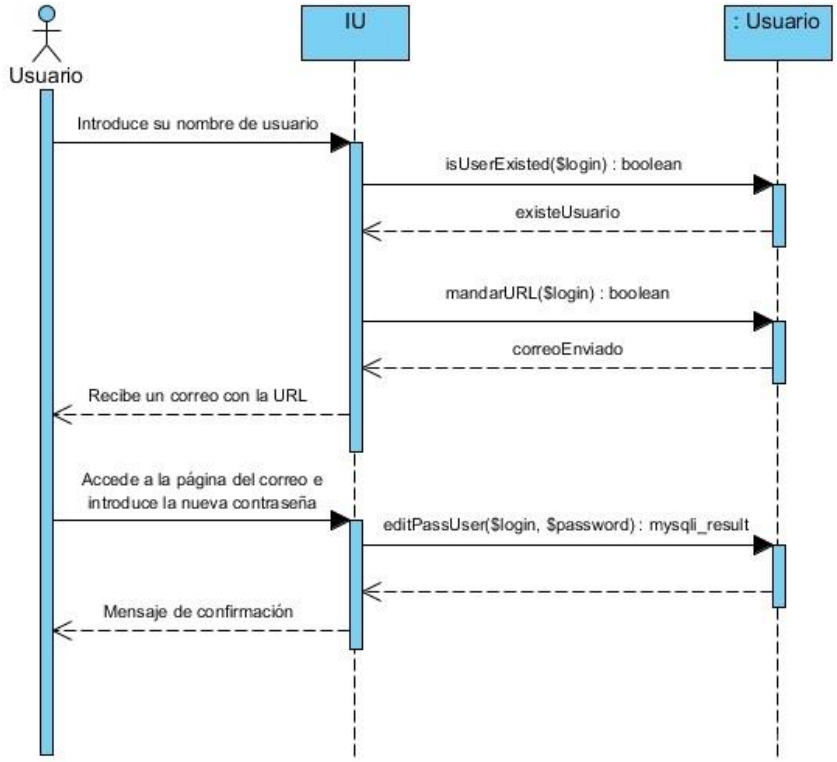


Ilustración 159. Diagrama de secuencia - Recuperar contraseña

Para recuperar la contraseña el usuario deberá introducir su nombre de usuario y a partir de ahí si es válido el sistema con los métodos que contiene el diagrama será capaz de generar una URL que recibirá el usuario y con la que podrá acceder a la pantalla para modificar su contraseña.

Añadir usuario

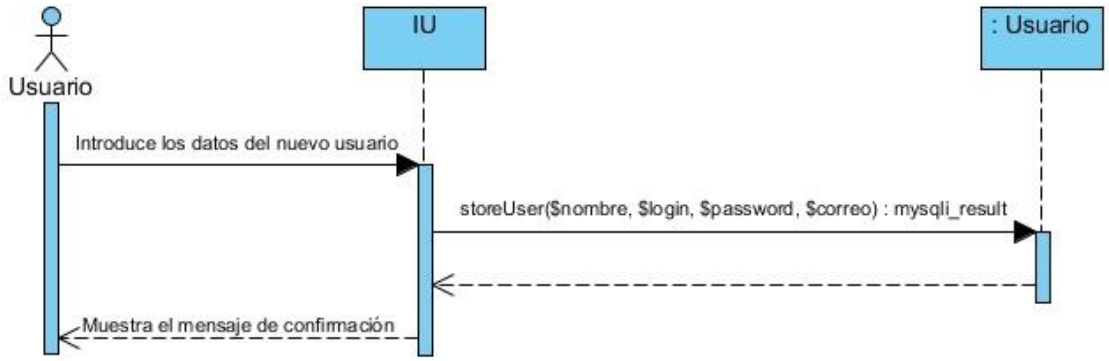


Ilustración 160. Diagrama de secuencia - Añadir usuario

El usuario se encarga de introducir los datos del usuario en la interfaz web y al enviar los datos el sistema deberá llamar al método `storeUser` que se encarga de almacenar un usuario en la base de datos.

Modificar usuario

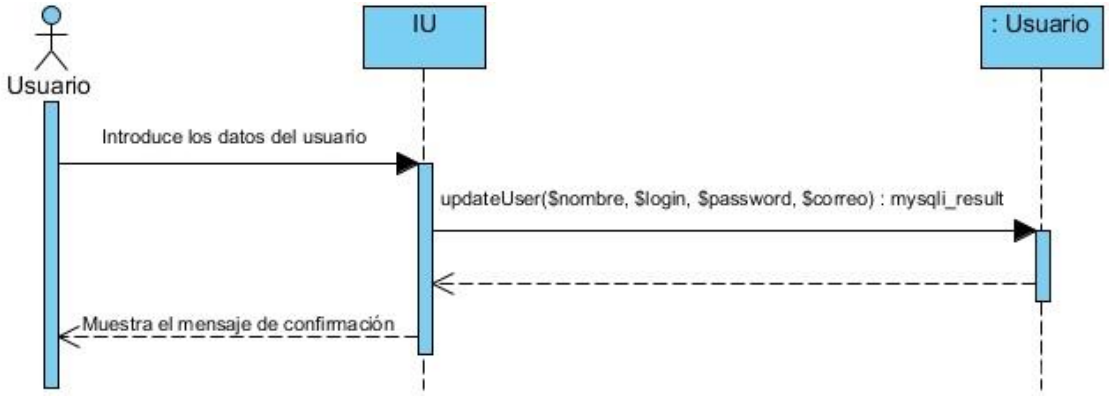


Ilustración 161. Diagrama de secuencia - Modificar usuario

Al igual que el diagrama para añadir un nuevo usuario, en este el usuario también debe introducir los nuevos datos del usuario existente, y se actualizarán con el método `updateUser`.

Ver datos

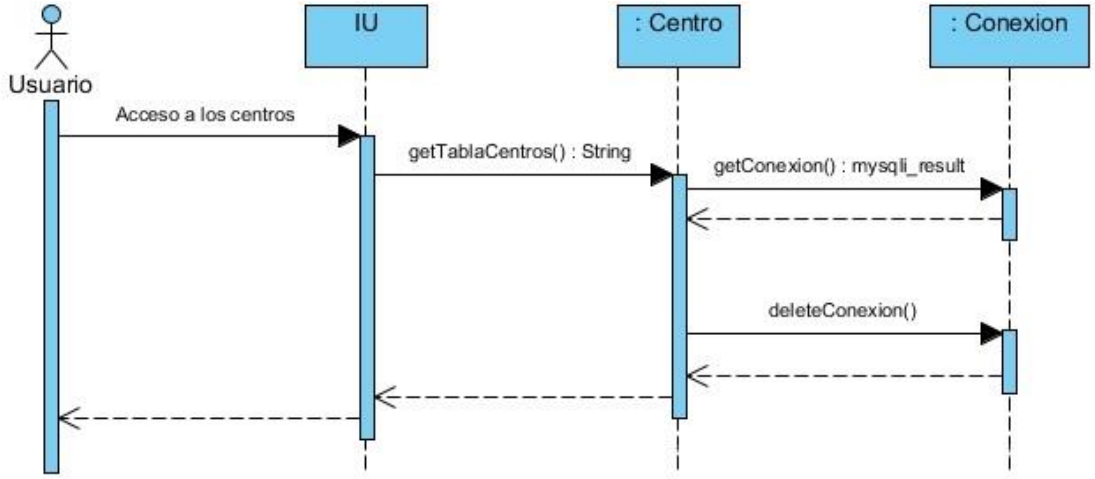


Ilustración 162. Diagrama de secuencia - Ver datos

El diagrama anterior muestra el proceso para visualizar datos de la aplicación, concretamente los centros, básicamente con el método `getTablaCentros` la interfaz recibe una cadena de texto con código HTML y los datos de los centros para imprimirlos por pantalla. Para realizar la consulta deberá ejecutar el método `getConexion` para conectar con la base de datos y `deleteConexion` para finalizar la conexión.

Este mismo proceso es el que deberán seguir el resto de datos del sistema para visualizarse. Cada clase contiene su correspondiente método `getTabla` para la obtención y visualización de los métodos.

Añadir datos

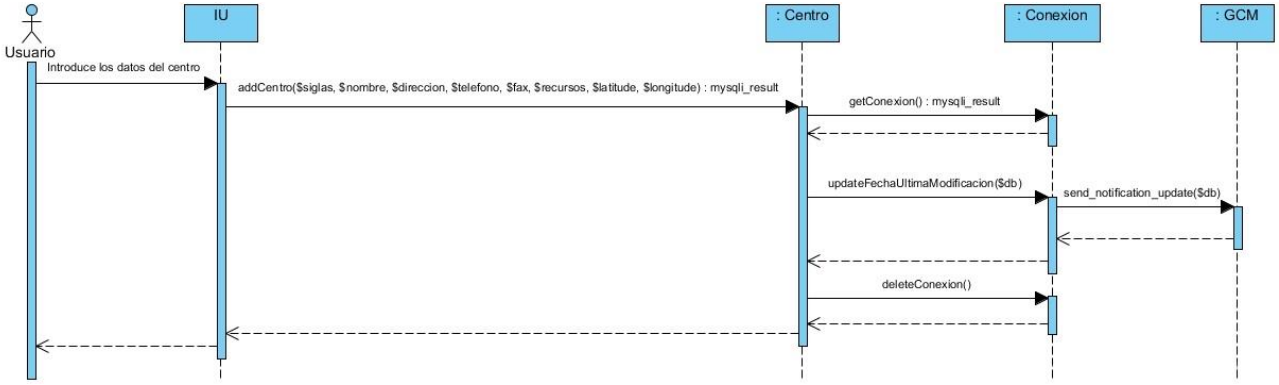


Ilustración 163. Diagrama de secuencia - Añadir datos

Como en el diagrama anterior, como ejemplo de añadir datos, se muestra el proceso de añadir un nuevo centro, para ello una vez el usuario ha introducido los datos y el servidor los recoge y utiliza el método `addCentro`. En este caso, igual que en todos los que se realiza un acceso a la base de datos, se llama a los métodos `getConexion` y `deleteConexion`, la diferencia principal con el caso anterior es que además de realizar la tarea correspondiente en la base de datos también se realiza una actualización del campo que contiene la última fecha de actualización del servidor, para esto llama al método `updateFechaUltimaModificacion` y este internamente llama al método `send_notificaction_update` para enviar la notificación de actualización a los dispositivos registrados en el servidor.

Editar datos

Ilustración 164. Diagrama de secuencia - Editar datos

Este diagrama muestra cómo se editan los datos en el servidor, la única diferencia con el caso de añadir datos es que se llama al método `updateCentro`. Internamente este método también realiza la conexión con la base de datos, actualiza la fecha de la última modificación local y manda un mensaje de notificación a los dispositivos a través de GCM.

Eliminar datos

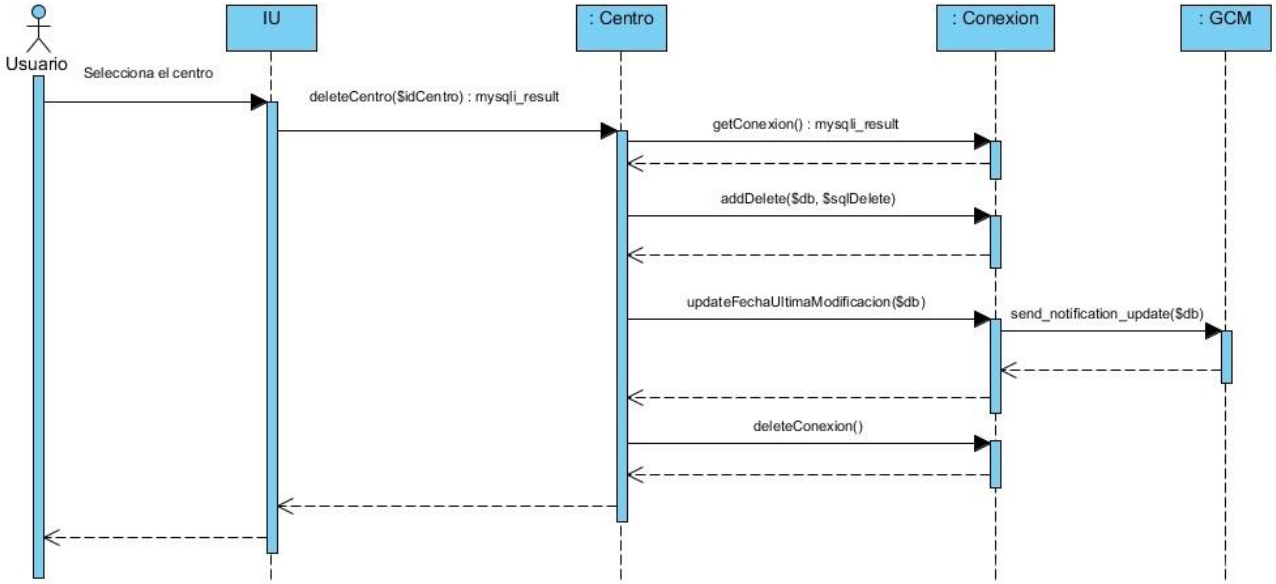
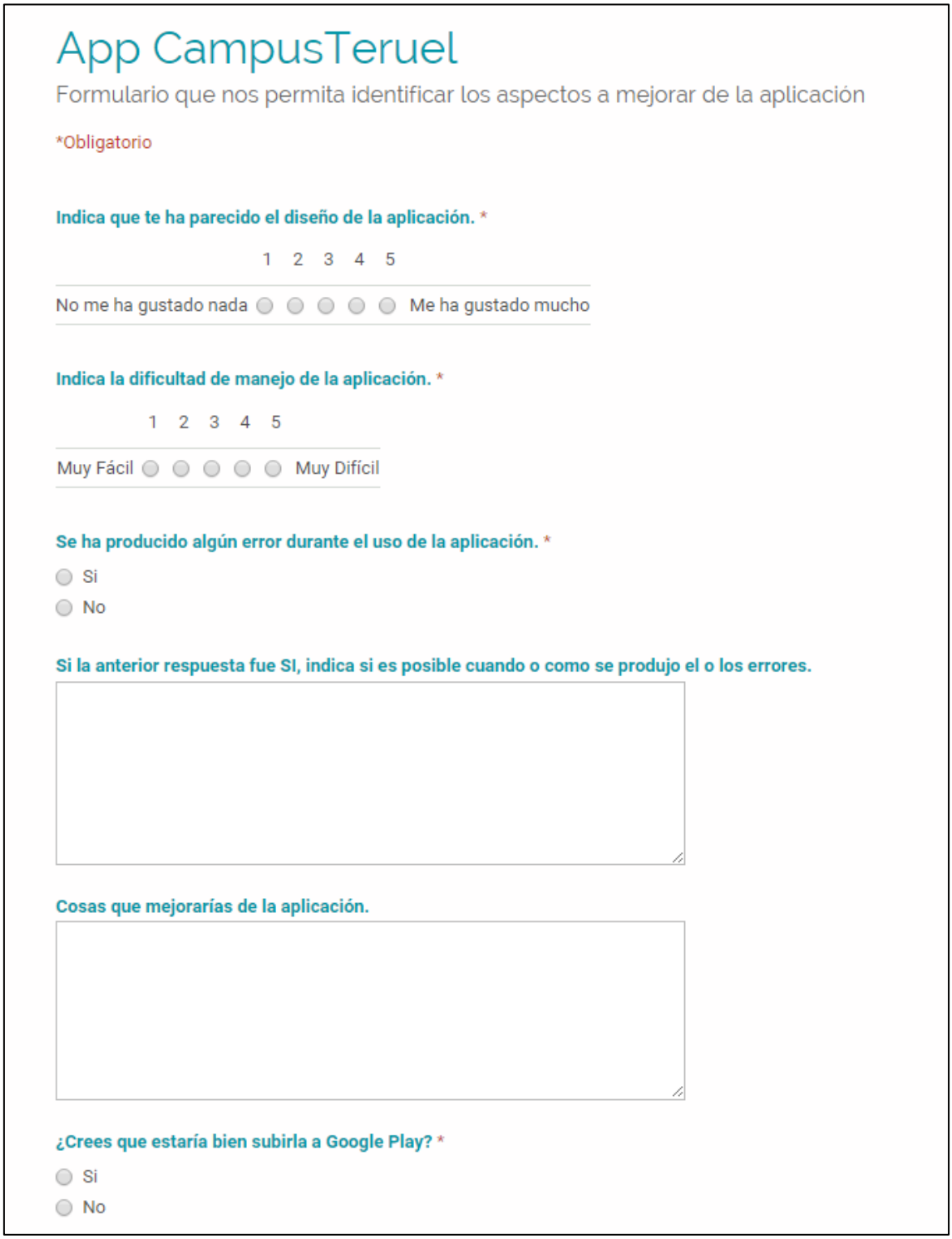


Ilustración 165. Diagrama de secuencia - Eliminar datos

Cuando un usuario selecciona un centro para ser eliminado se llamara al método `deleteCentro` para realizar la acción, este método realizara acciones similares a las mostradas en los diagramas anteriores pero también llamara al método `addDelete` que se encarga de añadir la sentencia SQL utilizada para eliminar el centro y poder así ejecutarla en la base de datos local para realizar la eliminación del centro.

Anexo H: Encuesta



App CampusTeruel

Formulario que nos permita identificar los aspectos a mejorar de la aplicación

*Obligatorio

Indica que te ha parecido el diseño de la aplicación. *

	1	2	3	4	5	
No me ha gustado nada	○	○	○	○	○	Me ha gustado mucho

Indica la dificultad de manejo de la aplicación. *

	1	2	3	4	5	
Muy Fácil	○	○	○	○	○	Muy Difícil

Se ha producido algún error durante el uso de la aplicación. *

- ○ Si
- ○ No

Si la anterior respuesta fue SI, indica si es posible cuando o como se produjo el o los errores.

Cosas que mejorarías de la aplicación.

¿Crees que estaría bien subirla a Google Play? *

- ○ Si
- ○ No

Ilustración 166. Encuesta - Parte 1

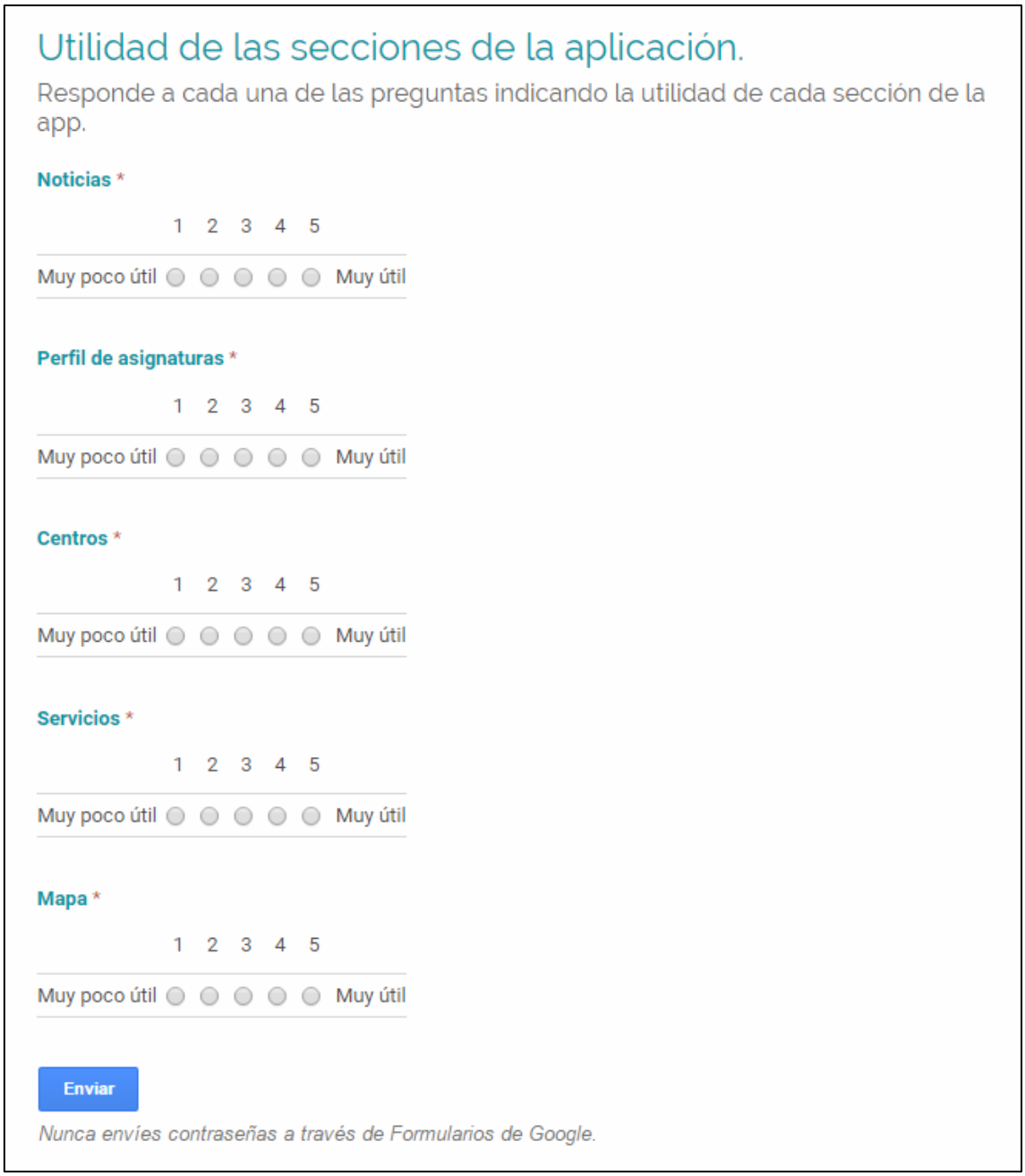


Ilustración 167. Encuesta - Parte 2

Anexo I: Manual de instalación

A continuación, se muestran los pasos a seguir para la correcta instalación del sistema; antes de ejecutar nada, se debe importar la base de datos y desplegar la aplicación web. Para ello, se deberá disponer de un servidor web, que contenga el modulo PHP y MySQL instalado si se desea alojar la base de datos en el mismo servidor.

I.1. Importar la base de datos

Para realizar la importación de las tablas y los registros a la base de datos del servidor, únicamente hará falta ejecutar el script SQL que se incluye en los archivos del proyecto. Simplemente habrá que indicar en qué base de datos se va ejecutar, una tarea fácil desde phpMyAdmin o MySQLWorkbench, y se añadirán a esta base de datos todas las tablas con los datos que contienen.

I.2. Desplegar la web de administración

Lo primero será desplegar la carpeta del proyecto web en la raíz del servidor web o en la carpeta donde se deban incluir las webs. Tras tener la web desplegada en la carpeta es necesario realizar algunas modificaciones en el fichero *config* en la ruta /inc/config.php. Las modificaciones a realizar son las siguientes:

```
define("SERVER","localhost");
define("USER","root");
define("PASS","1234");
define("DB","campusteruel");

define("GOOGLE_API_KEY", "AIzaSyCg89XOIARWOANMrXTkwcADvV9p910vCvo");

define("ENVIO_PHP_MAILER", true);
define("URL_SERVIDOR_RESET", "http://localhost/CampusTeruelAdmin/modificarPass.php?url=");
define("MAIL", "campusteruel@gmail.com");
define("PASSWORD_MAIL", "proyectoteruel");
```

Ilustración 168. Variables de la aplicación web

Las cuatro primeras variables corresponden a la conexión con la base de datos, con SERVER se le indicara el servidor donde está alojada la base de datos, USER y PASS son un usuario y su contraseña que tendrá acceso a la base de datos y DB es el nombre de la base de datos.

GOOGLE_API_KEY es la variable que contiene la clave para poder con la API de google GCM y poder mandar los mensajes de actualización a los dispositivos. En caso de que se desee utilizar otra cuenta de Google para realizar esta tarea habrá que generar una nueva clave y modificar la variable.

El último grupo de cuatro variables corresponden al método utilizado para enviar el correo con la URL para modificar la contraseña. La variable ENVIO_PHP_MAILER permite usar un método alternativo al que viene por defecto en PHP para enviar correos, de esta forma se le podrá indicar la dirección que se desea usar; si no se usará la configuración que tenga por defecto el servidor para enviar correos electrónicos. URL_SERVIDOR_RESET sirve para definir la URL que se le mandará al usuario que variará en función del nombre del dominio que tenga el servidor donde esté desplegada la web.

I.3. Desplegar AutomaticApiRest

Es un paso muy similar al anterior ya que se trata de desplegar igualmente una página web. Se coloca la carpeta de la API en la carpeta raíz del servidor o donde se desplieguen los proyectos web como el caso anterior, y al igual que antes se debe realizar una modificación en un fichero de configuración en la ruta /config.php.

```php
<?php
define("SERVER","localhost");
define("USER","root");
define("PASS","1234");
define("DB","campusteruel");
```

Ilustración 169. Configuración de Automatic Api Rest

Como se puede ver son las variables para la conexión a la base de datos, por lo que se deberán introducir los mismos parámetros que en el fichero anterior.

I.4. Instalación de la aplicación

Antes de realizar la instalación de la aplicación, como en los casos anteriores, se debe realizar una configuración y acto seguido generar el APK para proceder con la instalación. El fichero que se debe modificar es la clase `Config` en el paquete com.utility.campusteruel.

```
public static final String GOOGLE_SENDER_ID = "679310250757";

public static final String URL_SERVIDOR = "http://campusteruel.esy.es/";

public static final String URL_AUTOMATIC_API_REST_TABLA_PATH =   "http://apirest.campusteruel.esy.es/getData.php?t=";
```

Ilustración 170. Configuración de la aplicación.

La variable GOOGLE_SENDER_ID corresponde con el ID del proyecto abierto en la consola de APIs de Google; esto es necesario para poder conectar el dispositivo con el servidor GCM de Google.

Las variables URL_SERVIDOR y URL_AUTOMATIC_API_REST_TABLA_PATH son las variables necesarias para conectar con el servidor y con la API. Deben tener el valor de las URLs donde estén alojadas.

Por último, en caso de que se modificara el proyecto de la consola de APIs de Google, también sería necesario ir al archivo Manifest de la aplicación y modificar la API KEY que utiliza para la conexión con Google Maps.

Tras esto, el proyecto ya puede ser compilado y generar el APK para realizar la instalación en los dispositivos, recordar que si se desea instalar una aplicación fuera de Google Play es necesario tener activada la opción “Orígenes desconocidos” en los ajustes del dispositivo para poder instalar aplicaciones de fuentes que no vienen de Google Play.

Anexo J: Manual de usuario

J.1. Inicio

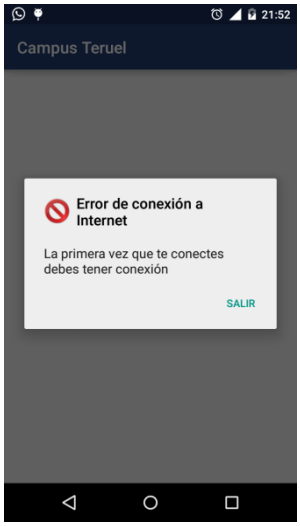


Ilustración 171. Alerta de conexión a Internet

En el inicio de la aplicación, al no existir datos, el usuario deberá tener conexión a Internet. En caso contrario, recibirá un mensaje de alerta, como el que muestra la Ilustración 171, y no podrá entrar en la aplicación. El usuario únicamente tendrá la opción de salir. De esta forma podrá ir a los ajustes del dispositivo y activar la conexión a la red mediante Wi-Fi o datos móviles.

Una vez esté conectado a la red, el usuario podrá acceder a la aplicación.

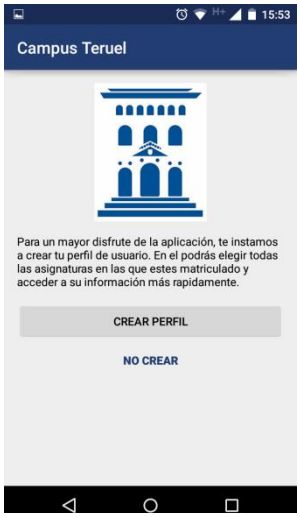


Ilustración 172. Pantalla inicio de la aplicación

Tras la descarga inicial de datos, el usuario tendrá la posibilidad de elegir entre crear su perfil seleccionando las asignaturas que desee, o acceder a la aplicación directamente.

Para realizar estas acciones tiene disponible en la pantalla de inicio los botones *crear perfil* o *no crear*, como se muestra en la Ilustración 172.

J.2. Crear perfil

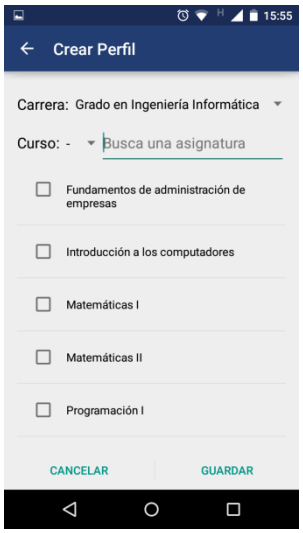


Ilustración 173. Pantalla crear perfil

La acción de crear el perfil del usuario es una de las más importantes, ya que permite al alumno seleccionar su titulación y las asignaturas en las que está matriculado para poder observar rápidamente la información disponible de ellas.

Para realizar esto, el usuario podrá seleccionar su carrera en el desplegable que aparece en la parte superior de la pantalla que muestra la Ilustración 173; acto seguido tendrá la posibilidad de filtrar las asignaturas por curso o buscarlas en la caja de texto, y así seleccionar las asignaturas que desee. Para finalizar el proceso, deberá pulsar sobre el botón *guardar* o si desea volver atrás, deberá pulsar *cancelar*.

J.3. Menú

Ilustración 174. Menú de la aplicación

Para abrir el menú de la aplicación basta con deslizar el dedo desde la izquierda hacia la derecha o bien pulsar el icono de tres rayitas que se sitúa en la pantalla inicial en la esquina superior derecha. El icono solo aparece en la pantalla inicial, ya que para seguir con el patrón de navegación de Android, a medida que se avanza por las pantallas se incluye el icono de una flecha que indica al usuario que puede volver.

Para cerrarlo basta con pulsar el icono de la fecha en la esquina superior izquierda, el botón de atrás o pulsar sobre la parte de la pantalla que no es ocupada por el menú.

Como se puede ver, a partir de este menú se puede acceder a las diferentes secciones de la aplicación, y la sección en la que nos encontremos está remarcada para que el usuario sepa dónde está ubicado.

J.4. Ver noticias

Ilustración 175. Pantalla de noticias

Ilustración 176. Pantalla de una noticia

Esta será la primera pantalla que el usuario visualizara. En ella tendrá acceso a las noticias que se van colgando en la web del campus. Para visualizar una en concreto basta con pulsar sobre ella. La Ilustración 176 muestra la visualización de una noticia, con su título, fecha en la que está publicada y el contenido de la misma.

El estilo de las pantallas donde se listan los datos es muy similar, la única diferencia es que no tienen una segunda línea, que en este caso se utiliza para mostrar la fecha en la que se publicó la noticia.

Al igual que con el listado de noticias, para acceder a la información de cualquiera de ellos basta con pulsar sobre una fila.

Como se ha explicado en el apartado anterior cuando aparece una flecha en la esquina superior izquierda, indica que se puede volver hacia atrás bien pulsando la flecha o el botón de atrás del dispositivo.

En estas dos ilustraciones también se puede observar el icono de una lupa, es el buscador de la aplicación que una vez se pulsa permite introducir el texto que se desee buscar. Más adelante veremos cómo muestra la información.

J.5. Perfil

Ilustración 177. Pantalla perfil

La Ilustración 177 muestra la pantalla de perfil del usuario. Tiene el menú donde podrá acceder a la web de Moodle desde la aplicación, el documento PDF del calendario académico, el listado de asignaturas que ha seleccionado a la hora de crear el perfil de usuario y el acceso a las opciones Google Calendar para importar y desvincular el horario al calendario de Google del usuario.

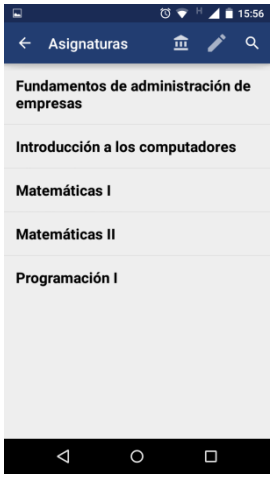


Ilustración 178. Pantalla asignaturas del perfil

La Ilustración 178 muestra lo que verá el usuario tras pulsar el botón *asignaturas*. Desde aquí podrá visualizar la información de una asignatura de las seleccionadas en el perfil, como el horario de exámenes o de clase, profesores que la imparten o el acceso a la web de la asignatura en la página de titulaciones de la Universidad de Zaragoza.

Con los iconos que aparecen en la barra del menú podrá tener un rápido acceso a la titulación seleccionada en el perfil y con el otro botón tendrá la opción de editar la carrera y asignaturas seleccionadas.

J.6. Ver centro

Ilustración 179. Pantalla menú de un centro

Cuando un usuario pulsa sobre un centro, tendrá acceso a las titulaciones que pertenecen a ese centro, la información de contacto disponible, el profesorado, y un plano ilustrativo del centro con las aulas, despachos de profesores y servicios señalizados.

Con el menú que aparece en la Ilustración 179, el usuario podrá acceder a la información comentada en el párrafo anterior.

J.7. Ver servicio

Ilustración 180. Pantalla de información de un

La Ilustración 180 muestra la pantalla de información de un servicio seleccionado por un usuario. Este formato es el seguido por la aplicación para mostrar la mayor parte de la información, por ejemplo de las asignaturas o de los profesores.

En caso de que aparezca una flecha en el contenido de la información, eso significa que se puede realizar una acción con esa información, por ejemplo con la dirección de correo la aplicación hará que el dispositivo habrá la aplicación por defecto de correos electrónicos y permite redactar un correo con la dirección pulsada.

J.8. Mapa

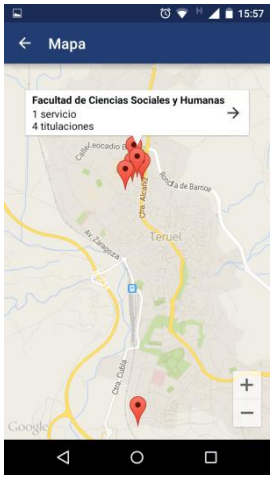


Ilustración 181. Mapa de la aplicación

Cuando el usuario pulsa sobre la opción *mapa* del menú, accederá a un mapa de Google Maps con los centros señalizados.

Cuando pulsa sobre uno de los *puntos* del mapa, se ve el nombre del centro y las titulaciones y servicios que contiene ese centro. Y como se puede ver muestra una flecha, este icono indica que al pulsar se puede acceder a la información del centro, tal y como muestra la Ilustración 182.

Ilustración 182. Información de un centro

Cuando el usuario accede a la información de un centro, verá un listado con las titulaciones y servicios de éste, y podrá seleccionar una de las filas para acceder a la información concreta de la titulación o del centro.

Con los botones del menú de la barra se puede acceder al plano y la información de contacto del centro. Además, se puede acceder a la localización del centro en Google Maps para poder aprovechar la opción de cómo llegar.

Por último, también visualizará una imagen del centro.

J.9. Ver titulación

Ilustración 183. Menú de una titulación

Cuando un usuario accede a la información de una titulación, aparecerá la pantalla de la Ilustración 183, desde donde podrá acceder a la información concreta de la titulación.

Con el botón *web titulaciones* accede a la web de titulaciones de la titulación. Con los botones de *horario de clases* y *calendario de exámenes* se descargan los documentos PDF correspondientes a la titulación y los muestra en la aplicación por defecto del dispositivo para visualizar PDFs.

Para mostrar realizar búsquedas, se puede ver que en la parte superior aparece un campo que permite introducir datos los datos que el usuario desee. En el siguiente apartado se verá cómo muestra los resultados.

J.10. Buscador

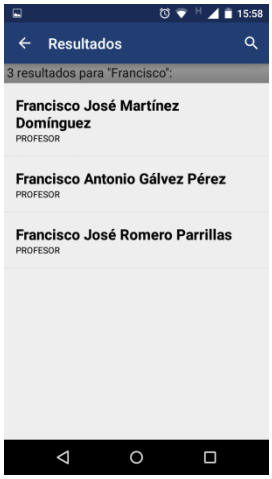


Ilustración 184. Buscador

La Ilustración 184 muestra cómo se ve el resultado de una búsqueda. Desde aquí el usuario tiene la posibilidad de acceder a la información concreta de cualquiera de los resultados simplemente pulsando sobre uno de ellos

J.11. Ver profesor

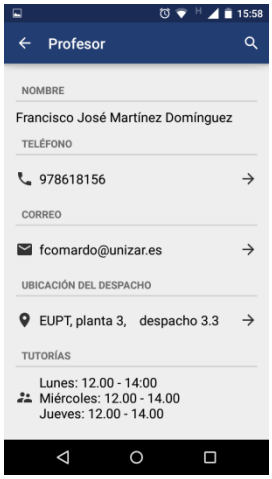


Ilustración 185. Profesor

Como se indicó en el apartado J.7., el formato para mostrar la información es muy similar en todos los casos.

Cuando el usuario pulsa sobre un profesor, la información que verá será la presentada en la Ilustración 185. Como en la gran mayoría de las pantallas, podrá utilizar el buscador y realizar acciones con la información mostrada.