Escuela Politécnica Superior (Jaén)
Ingeniería en Informática

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Proyecto Fin de Carrera

DISEÑO Y DESARROLLO DE UNA APLICACIÓN DE ESCRITORIO BAJO CLOUD COMPUTING

Alumno: José María Cuadros Higueras

Tutores: Prof. Dr. D. Luis Martínez López
Prof. D. Iván Palomares Carrascosa
Dpto: Informática

Octubre, 2013

Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Dr. D. Luis Martínez López y D. Iván Palomares Carrascosa, tutores del Proyecto Fin de Carrera titulado: Diseño y desarrolo de una aplicación de escritorio bajo Cloud Computing, que presenta D. José María Cuadros Higueras, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Octubre de 2013

El alumno: Los Tutores:

D. José María Cuadros Higueras

Prof. Dr. D. Luis Martínez López
Prof. D. Iván Palomares Carrascosa

Agradecimientos

A mis padres, por convertirme en quien soy y enseñarme desde el principio los valores que me habéis inculcado, así como que el trabajo, la constancia y la voluntad son la clave para conseguir todo lo que uno se proponga en la vida.

A mi hermano, por aportarme la diversión en esos ratos de desconexión que tanto necesitaba y por llenar mi vida desde que llegaste.

A mis abuelos y tío, por los momentos que me habéis dado y por todo lo que me habéis enseñado. Sé que siempre, y algunos desde el cielo, me seguís y lo seguiréis haciendo.

A Charlotte, porque a pesar de que aún no llevas mucho tiempo en mi vida te has hecho un hueco muy grande en ella y por tu apoyo incondicional.

A mis amigos, en especial a Yeyu, Jose, Salva, Isa, Carlos, Antonio, Kiko, Elena, Jan, Dito, Miguel, Pia, Paco y a todos los que en algún momento habéis formado parte de mi vida. Por estar siempre ahí, por los momentos juntos, y por completarme tal y como lo hacéis... Gracias.

A Macarena, por tus consejos y los buenos momentos de conversación... y porque precisamente gracias a ti empezó el interés por el Cloud Computing, el tema de este proyecto.

A Miguel Ángel, por tener siempre la puerta abierta de tu despacho para resolver mis dudas y cuestiones de informática durante los últimos años.

A Fran Estrella, por transmitirme el amor por lo que haces, por tu ayuda y tus conocimientos.

Y por supuesto, a Iván, por tus conocimientos, disponibilidad y cercanía siempre, y que espero que mantengamos así como a Luis por aprender todo lo que he aprendido contigo y darme la oportunidad de poder llevar a cabo este proyecto.

Índice general

Lista de figuras

2. Cloud Computing

3. Desarrollo del proyecto

A. Manual de Instalación

B. Manual de Usuario

CAPÍTULO 1:

Introducción

1.1 Introducción al Proyecto

Hace cien años, a comienzos del s. XX, nadie podía imaginarse que cien años después íbamos a poder realizar algunas tareas tan complejas como las siguientes: tener reuniones con personas que están a miles de kilómetros, estar en continuo contacto con familiares y amigos a pesar de encontrarnos en otros lugares, disponer de cualquier información sobre un evento que en ese mismo momento sucede en el otro extremo del mundo, y seguiríamos con ejemplos innumerables. Con esto, queremos poner en relieve que una de las características que más destacan de los seres humanos de principios del s. XXI, es la utilización de Internet para llevar a cabo cada vez más acciones. Nos hemos dado cuenta de que Internet nos hace la vida más cómoda, más accesible y, siempre dependiendo del uso que se le dé, puede ofrecer una calidad de vida más alta.

Las tecnologías asociadas a Internet van avanzando prácticamente día a día. Algunas de ellas surgen y unos meses después se popularizan y se extienden hasta convertirse en totalmente imprescindibles para la humanidad. Sin embargo, algunas otras pueden caer en el olvido años después y quedarse obsoletas. Este último no parece ser el caso de la tecnología sobre la que versa este proyecto, el Cloud Computing, que ha venido para quedarse, y para proporcionar al ser humano del s. XXI medios que cualquier generación anterior no podría ni imaginarse.

El Cloud Computing [1] es actualmente un concepto que se utiliza de forma habitual en las TICs (Tecnologías de Información y Comunicación), y que a medio plazo será la base de todas las aplicaciones y servicios que utilizamos. Este término se refiere a aquellas aplicaciones y servicios que se ejecutan en una red distribuida utilizando recursos virtualizados y accediendo mediante protocolos comunes de Internet y estándares de redes. Se identifica por la noción de que los recursos son virtuales, por lo general con menos límites que los recursos físicos actuales, y que las características de los sistemas en los que se ejecuta el software se abstraen del usuario.

Igualmente, cabe destacar que el paradigma del Cloud Computing toma protocolos y estándares similares a los que se usan en Internet y los convierte en una herramienta de autoservicio [2]. El uso de la palabra "nube" hace referencia a dos conceptos esenciales en Cloud Computing: virtualización y abstracción.

Algunos de los ejemplos de herramientas que pueden servir como muestra de la utilidad e importancia de este nuevo paradigma computacional son:

- Dropbox [3]: Servicio de almacenamiento en la nube que permite almacenar documentos, fotos, imágenes, etc. y acceder a ellos desde cualquier dispositivo. Además, permite también compartir esta información con otros usuarios.
- Salesforce.com [4]: Solución en la nube para sistemas de gestión de clientes, empresas etc, sin limitación de hardware ni software.
- Google App [5]: Herramienta ofimática en la nube para crear y gestionar nuestros documentos sin necesidad de instalación de software ni actualizaciones.

Al referirnos al Cloud Computing, podemos distinguir fundamentalmente tres modelos de servicio: plataformas, protocolos y servicios:

- "Infrastructure as a Service" (Iaas) [1][28]: Es el modelo más básico, y en él se ofrece hardware para poder ser utilizado a través de la nube por los clientes. La entidad que proporciona el hardware es en este caso el proveedor de la infraestructura.

- "Platform as a Service" (PaaS) [1][30]: en este modelo se proporciona ya la infraestructura con funciones software añadidas, como pueden ser el sistema operativo, una base de datos, un servidor web o entornos cerrados de programación, siempre dependiendo de la plataforma que se adquiera y de los servicios que ésta ofrezca.

- "Software as a Service" (SaaS) [1][24]: software completo almacenado en la nube para que el usuario pueda utilizarlo sin importar dónde se encuentre. Ofrece también normalmente la posibilidad de pagar por su uso, y no por licencia. Twitter, Facebook y Flickr son algunos ejemplos de SaaS.

Dada la creciente importancia que está adquiriendo el paradigma de Cloud Computing en multitud de aplicaciones software, en este proyecto se desarrollará una aplicación orientada a la administración y gestión de tareas en el ámbito académico que permita a los estudiantes organizar sus asignaturas, tareas, prácticas y exámenes.

Por tanto, crearemos una aplicación de escritorio desarrollada mediante el lenguaje de programación Java, y que mediante la plataforma de desarrollo en la nube de Google (Google App Engine), nos permitirá acceder a nuestros datos, es decir, nuestras asignaturas, prácticas, tareas y exámenes desde cualquier lugar. Nuestro software será un servicio Cloud del tipo SaaS, que accederá a la plataforma Google App Engine. Dicho SaaS mantendrá una base de datos con las cuentas de los usuarios y su información, a la que cada usuario tendrá acceso mediante la aplicación de escritorio construida en Java [6].

1.2. Propósito

Diseño y desarrollo de una aplicación de escritorio para la gestión de tareas académicas basada en el paradigma Cloud-Computing.

1.3. Objetivos

- Búsqueda y revisión bibliográfica.
- Estudio de los diferentes servicios, plataformas y aplicaciones existentes en Cloud Computing y elección de los recursos que se utilizarán para desarrollar la aplicación.
- Estudio y obtención del espacio de almacenamiento utilizado para la aplicación.
- Diseño de una interfaz para aplicación de escritorio y de la funcionalidad de ésta.
- Implementación de la aplicación.
- Prueba y evaluación del funcionamiento del sistema.
- Redacción de la memoria

1.4 Estructura y Planificación del Proyecto

A continuación, haremos una breve introducción que explicará la estructura general de esta memoria.

En el capítulo 2, se muestra una visión general del Cloud Computing. En primer lugar, explicaremos qué es el Cloud Computing, conceptos previos necesarios para su entendimiento y algunas de sus características más importantes. A continuación estudiaremos las diferentes clasificaciones, arquitecturas y plataformas de desarrollo en la nube que existen, además de mostrar diversos ejemplos de cada uno de éstos, haciendo especial hincapié en Google App

Engine por ser la plataforma en la que se basa este proyecto. Por último se hablará de las limitaciones y retos para el futuro del Cloud Computing.

El capítulo 3 está dedicado a los procesos de Ingeniería del Software desarrollados en este proyecto. Al tratarse de un proyecto software, a lo largo del capítulo se desarrollan las diferentes etapas de la Ingeniería de Software. Así, definiremos los requerimientos funcionales y no funcionales del sistema, abordaremos la etapa de análisis, mediante el modelo de casos de uso y los escenarios de la aplicación. Además, realizaremos la fase de diseño, definiendo la estructura de la base de datos y el diseño de la interfaz de la aplicación. Para terminar, se realiza la fase de implementación, pruebas y se explica cada una de las tecnologías que se han utilizado para llevar a cabo el proyecto.

Una vez expuesto el desarrollo del proyecto, el cuarto capítulo está dedicado a las conclusiones derivadas de la realización del mismo, y a propuestas de futuros proyectos que sirvan como continuación de éste.

Esta memoria termina con los anexos dedicados a la instalación de la aplicación, el manual de usuario de la misma y el código fuente correspondiente al algoritmo de sincronización que se ha implementado en la aplicación.

CAPÍTULO 2:

Cloud Computing

2.1. Introducción al Cloud Computing

Como ya se ha dicho en el primer capítulo, el Cloud Computing se refiere a aquellas aplicaciones y servicios que se ejecutan en una red distribuida utilizando recursos virtualizados y accediendo mediante protocolos comunes de Internet y estándares de redes [1].

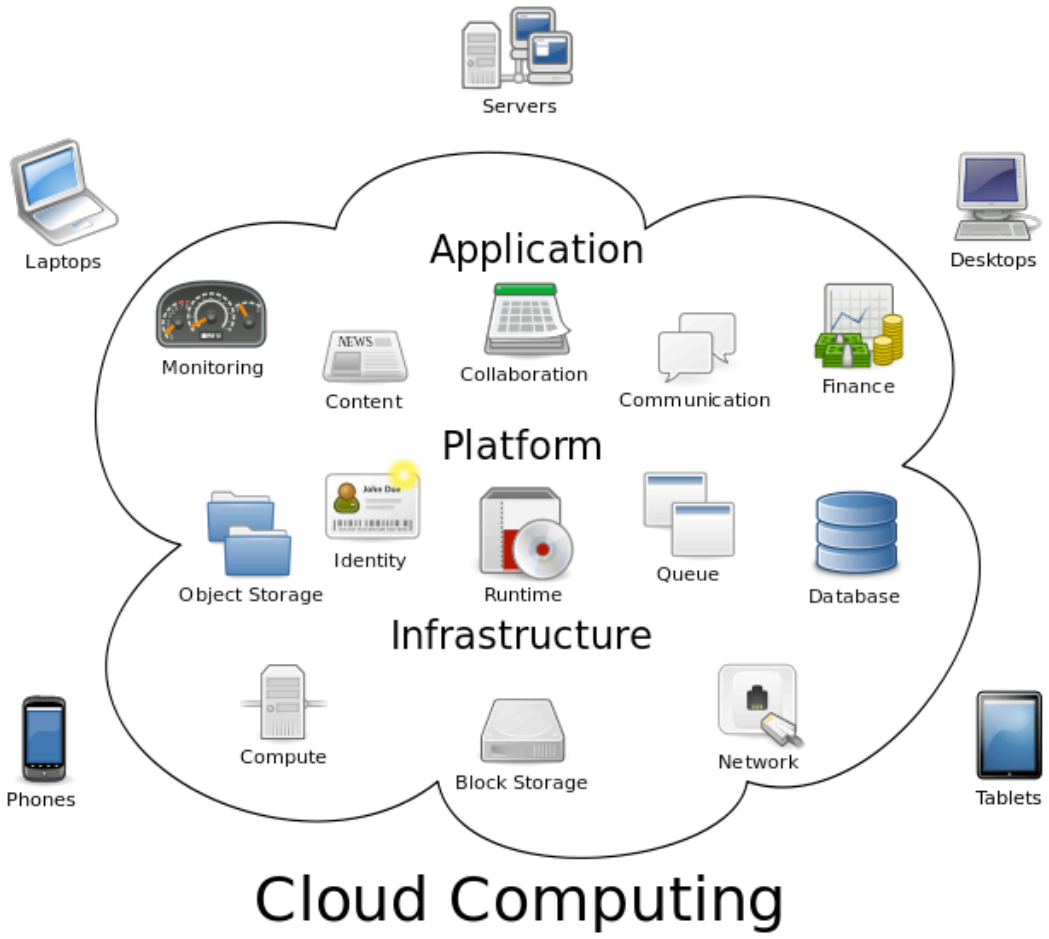

Figura 2.1: Estructura general del Cloud Computing.

En principio, el Cloud Computing, cuya estructura podemos observar en las figura 2.1 y 2.2, nos ofrece una serie de ventajas que nos hace mucha más cómoda la relación que los seres humanos establecemos con los ordenadores. Algunas de ellas son:

- Nos permite sincronizar información entre diferentes dispositivos del usuario, evitando que se generen versiones de datos sin sentido y sin actualizar.

- Nos da la posibilidad de acceder a nuestros documentos, música, imágenes, etc. desde cualquier lugar, sin lamentar que hemos olvidado almacenar los documentos en un pendrive o en un CD.

- Con el Cloud Computing, podemos compartir nuestros documentos con amigos, trabajar de forma paralela sobre los mismos proyectos que nuestros compañeros de trabajo, y combinar posteriormente los progresos que cada uno haya llevado a cabo.

- Nos evita tener que depender de un ordenador físico con nuestra información, e incluso puede suponer un ahorro importante en hardware, ya que no necesitamos ningún dispositivo externo (ya sea CD, disco duro o pendrive) para transportar nuestra información, con las consecuentes ventajas que esto ofrece, tanto económicas, como por comodidad, así como para el medio ambiente.

En conclusión, la idea clave del Cloud Computing es que las TIC (Tecnologías de la información y la comunicación) se conviertan en un servicio, de modo que "las aplicaciones del software no tienen por qué existir en un lugar concreto sino que pueden estar compuestas de múltiples elementos procedentes de múltiples sitios" [7].

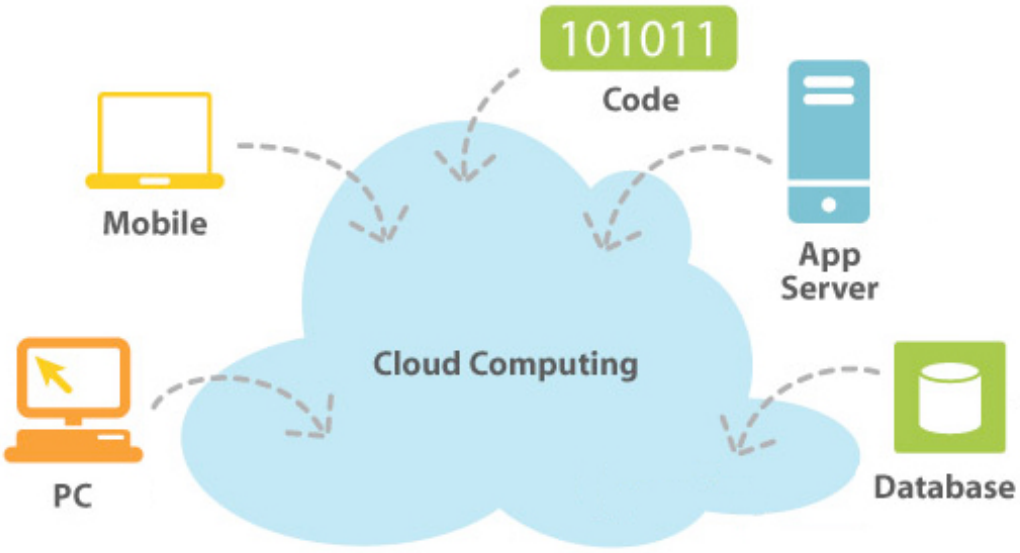

Figura 2.2: Estructura externa del Cloud Computing.

En los epígrafes mostrados a lo largo de este capítulo se explicará:

- Conceptos básicos necesarios para entender en qué consiste el Cloud Computing. En él se incluyen también algunas de las características fundamentales de éste.
- Una breve historia del Cloud Computing, describiendo desde su origen e inicios hasta lo que hoy en día es.
- Las características principales del Cloud Computing.
- Clasificación atendiendo a los modelos de implementación y a los modelos de servicio de la nube.
- Se explicarán con más detalle los modelos de servicio, y se mostrarán varios ejemplos de cada uno de ellos.
- Finalmente, hablaremos de las limitaciones del Cloud Computing, así como la dirección en la que este paradigma se mueve actualmente y el futuro que queda por explorar.

2.2. Conceptos básicos para entender el Cloud Computing

Antes de profundizar sobre el Cloud Computing, vamos a explicar algunos de los conceptos clave para entenderlo. El Cloud Computing basa su funcionamiento en dos procedimientos: virtualización y servicios web. Por tanto, cada uno de ellos tendrá un apartado en esta sección. De la misma manera, se dedica un apartado a dos propiedades que caracterizan al Cloud Computing, y cuyo entendimiento nos facilitará su comprensión cuando hablemos con más detalle sobre éste. Estas propiedas son: la escalabilidad elástica y la capacidad "multi-inquilino" (multi-tenant).

2.2.1. Virtualización

La virtualización es la abstracción de algún componente físico a través de un objeto lógico. Mediante la virtualización de un recurso, se puede obtener mayor utilidad que la proporcionada por el propio recurso. La virtualización puede implementarse en tres niveles diferentes: nivel de aplicaciones, nivel de sistema operativo y nivel de hardware [8]. Cada uno de estos tres niveles de virtualización tiene representación en el Cloud Computing, tal y como se verá en los siguientes epígrafes. A modo de ejemplo, podemos ver un diagrama sobre virtualización a nivel

de hardware en la figura 2.3, en la que es posible la ejecución simultánea de múltiples Sistemas Operativos por parte de una única computadora.

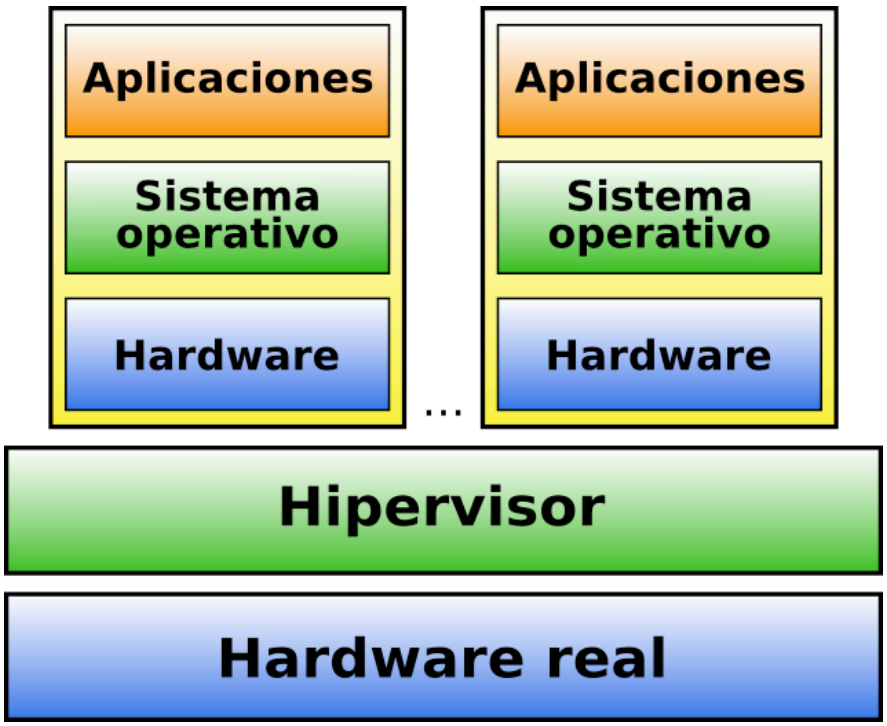

Figura 2.3: Estructura de la virtualización a nivel de hardware.

2.2.2. Servicios Web

A continuación, para hablar de los servicios Web, se incluyen dos definiciones tomadas de diferentes citas para comprender mejor el término:

"Un servicio Web es una interfaz accesible a la Red construida para permitir la funcionalidad de una aplicación mediante tecnologías web estándar" [9].

"Un servicio Web es una aplicación del software que permite acceder y ser accedida remotamente usando diferentes lenguajes estándar estructurados. Normalmente se identifica mediante una dirección URL o mediante otro sitio web. Lo que diferencia los servicios Web de los sitios Web convencionales es el tipo de interacción que proporcionan" [10].

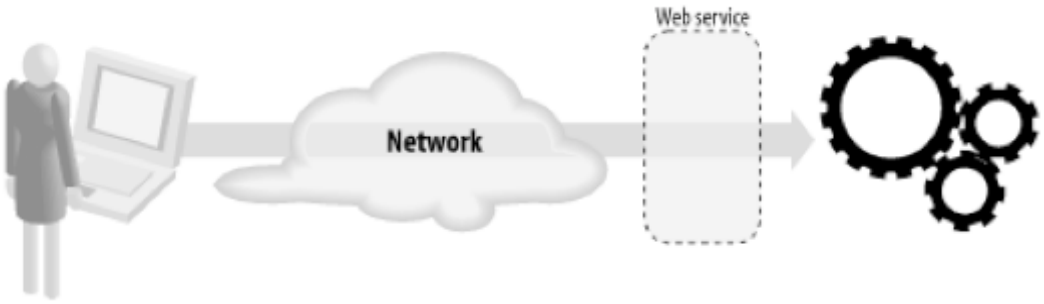

Figura 2.4: Esquema de un Servicio Web.

En la figura 2.4, se encuentra el esquema que mantiene un servicio Web. Mientras que la mayoría de los sitios Webs proporcionan respuestas a peticiones desde el cliente directamente, en los servicios Web se envían las peticiones desde el cliente mediante el servicio, esto es, mediante un documento con un formato especificado tanto en el servidor como en el cliente, de tal manera que puedan comunicarse de manera bidireccional mediante el protocolo establecido [10].

Algunos de los protocolos más utilizados para llevar a cabo la comunicación entre cliente y servidor son SOAP y REST [9,10,11].

SOAP (Simple Object Access Protocol) es un protocolo estandarizado que permite compartir mensajes en aplicaciones cliente - servidor [10]. La especificación se define simplemente como un mensaje sencillo basado en XML que transfiere la información que se requiere, y un conjunto de reglas para la traducción de aplicaciones y tipos de datos específicos de la plataforma en representaciones XML [9], tal y como se muestra en la figura 2.5.

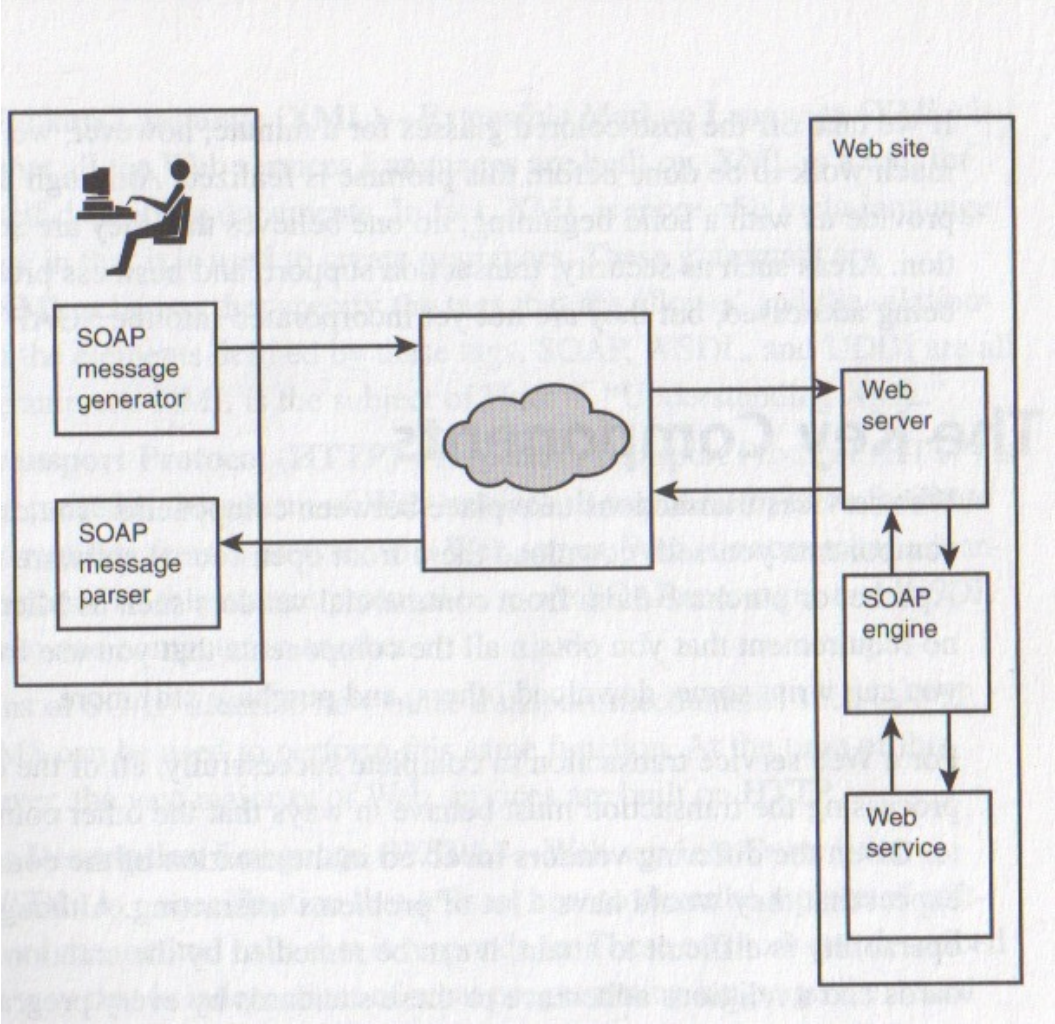

Figura 2.5: Ejemplo de Servicio Web mediante mensajes SOAP [10].

Por su parte, REST (Representational State Transfer) es un protocolo para comunicaciones entre cliente y servidor caracterizado por su utilización de peticiones HTTP. REST es un protocolo para el diseño de aplicaciones de red. La idea es que, en lugar de utilizar mecanismos complejos, como CORBA, RPC o SOAP la conexión se realice de manea simple mediante HTTP para lograr la comunicación entre hosts. En muchos sentidos, la Web en sí misma, basada en HTTP, puede ser vista como una arquitectura basada en REST [11].

2.2.3 Propiedades del Cloud Computing

A) Multi-Tenant

Una de las propiedas más importantes del Cloud Computing es su propiedad "Multi-Tenant" (multiinquilino). Esta propiedad se refiere a un principio de arquitectura de software, que consiste en utilizar múltiples instancias del software, cada una de las cuales se ejecutará mediante virtualización, y servirá a un usuario diferente (entendiendo usuarios no sólo como particulares sino también como organizaciones o empresas clientes). Con una arquitectura clásica, una aplicación software es diseñada para trabajar prácticamente con particiones de los datos de los clientes y su configuración, sin embargo en Cloud Computing cada organización cliente trabajaría con una instancia de la aplicación virtual personalizada, esto es, solamente con sus datos y configuración sin tener acceso a los de los demás. Con ello, cada recurso real es utilizado concurrentemente por varios «tenant» (siendo cada uno de ellos un usuario a nivel individual o corporativo) [12].

Además, cabe tener en cuenta que la seguridad, la privacidad y la protección de los datos de cada usuario debe ser una prioridad para los proveedores de Cloud Computing.

B) Escalabilidad elástica

Puede darse el caso de que en algunos momentos una empresa necesite realizar algunos cálculos con alto coste computacional. Estas empresas pueden utilizar la nube para estos cálculos, ya que les permitiría beneficiarse de la escalabilidad elástica de la virtualización. Se puede agregar capacidad adicional cuando se necesite en un momento dado, y fácilmente retirar el uso de estos recursos cuando no se necesite. La mayoría de los proveedores de Cloud ofrecen costes de "pago por uso", es decir, pagar solamente por el tiempo y el recurso usado, ya que esto les puede ofrecer un beneficio significativo en su balance final, reduciendo sus costes en épocas de menor utilización. También aporta a los usuarios la sensación de que los recursos son ilimitados [13].

2.3. Breve historia del Cloud Computing

A continuación, se comenta brevemente la historia del Cloud Computing, que si bien es bastante reciente, ya podemos encontrar algunos documentos sobre sus sucesos más destacados.

El origen del Cloud Computing data de la década de los 50, donde ya las grandes empresas muestran su tendencia a tener acceso a grandes cantidades de información desde distintos terminales que estén localizados en diferentes puntos del planeta. Sin embargo, el alto coste de toda la infraestructura necesaria propició que no se pusiera en marcha en ese momento. Al menos, es en ese momento cuando surge el concepto de Cloud teóricamente.

El concepto básico del "Cloud Computing" se le atribuye a John McCarthy - que también fue el responsable de introducir el término "inteligencia artificial". En 1961, durante un discurso en el centenario del MIT (Massachusetts Institute of Technology), sugirió públicamente que la tecnología de tiempo compartido (Time-Sharing) de los ordenadores podría conducir a un futuro donde el poder de computación e incluso aplicaciones específicas podrían venderse como un servicio (tal como el agua o la electricidad). Un año más tarde, el informático estadounidense Joseph Carl Robnett Licklider escribió sobre la idea de una red informática mundial, una especie de "redes intergalácticas de computación", proponiendo teóricamente muchos conceptos de los que hoy nos ofrece el Cloud Computing [14].

El primer sector en experimentar el Cloud Computing fue el bancario, pues se popularizaron los cajeros automáticos que permitían acceder al dinero y servicios desde cualquier terminal. Ya en estos sistemas, se disponía de la infraestructura necesaria para acceder a los datos de los usuarios desde distintos terminales.

Sin embargo, no fue hasta la década de los noventa cuando se pudo hacer realidad la nube, mediante los protocolos TCP/IP. A partir de este momento, el uso del Cloud Computing se disparó, primero mediante el apoyo de grandes empresas, y después, mediante el impulso necesario que han llevado a cabo empresas más pequeñas, que empiezan a utilizar éste a través de los servicios virtuales que ofrecen las grandes compañías.

En 2002, Amazon creó los Servicios Web con AWS (AMAZON WEB SERVICE), que permitían a los usuarios disponer de un sistema de almacenamiento en la nube, algo así como un pendrive en la red, virtual, y que siempre esté disponible al usuario simplemente utilizando una conexión a Internet y su cuenta de usuario Amazon [15]. Encontramos la interfaz de AWS en la figura 2.6.

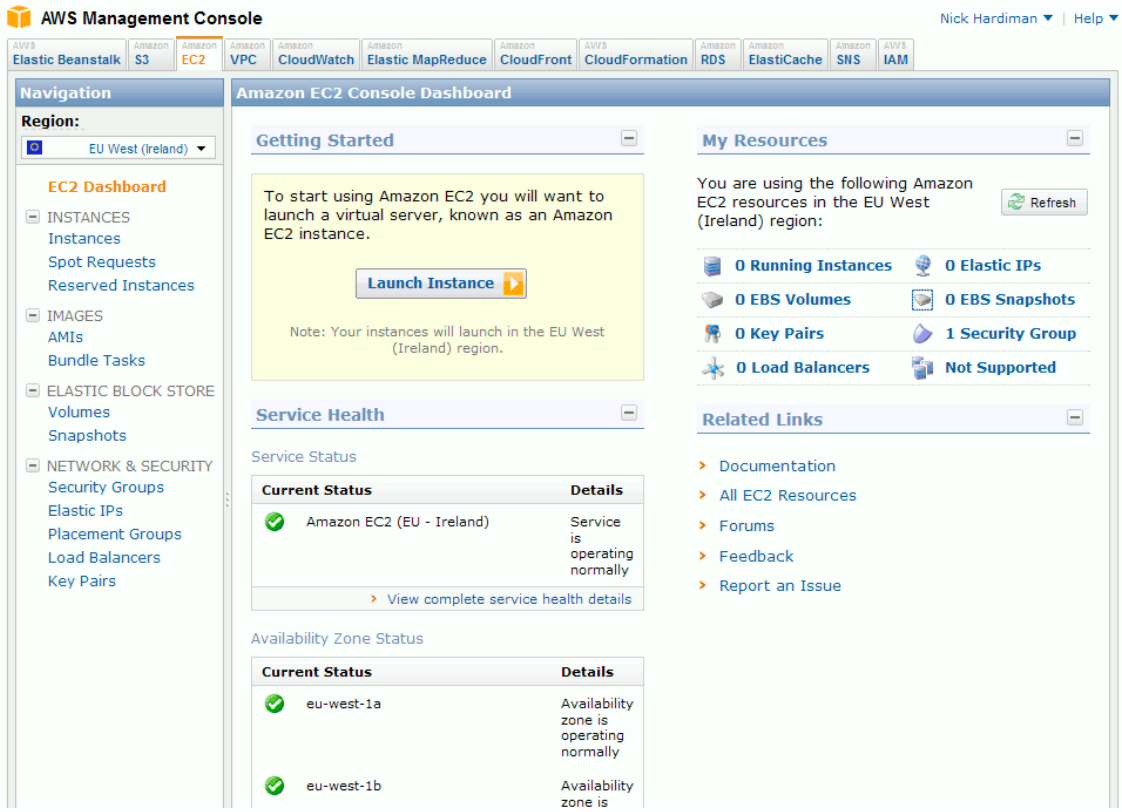

Figura 2.6: Interfaz de Amazon AWS.

A partir de 2006, nuevamente Amazon creó Elastic Compute Cloud (Ec2), esto es, un servicio comercial que permite a pequeñas y medianas empresas alquilar servidores para hacer funcionar sus aplicaciones (figura 2.7). Incluso empresas grandes que necesiten grandes cálculos, ya no necesitan grandes y costosos equipos, pues simplemente con el alquiler de éstos en servicios como el que ofrece Amazon, permiten utilizar estos servicios durante el tiempo necesario, ahorrando en costes de infraestructura, medio ambiente y energía [15].

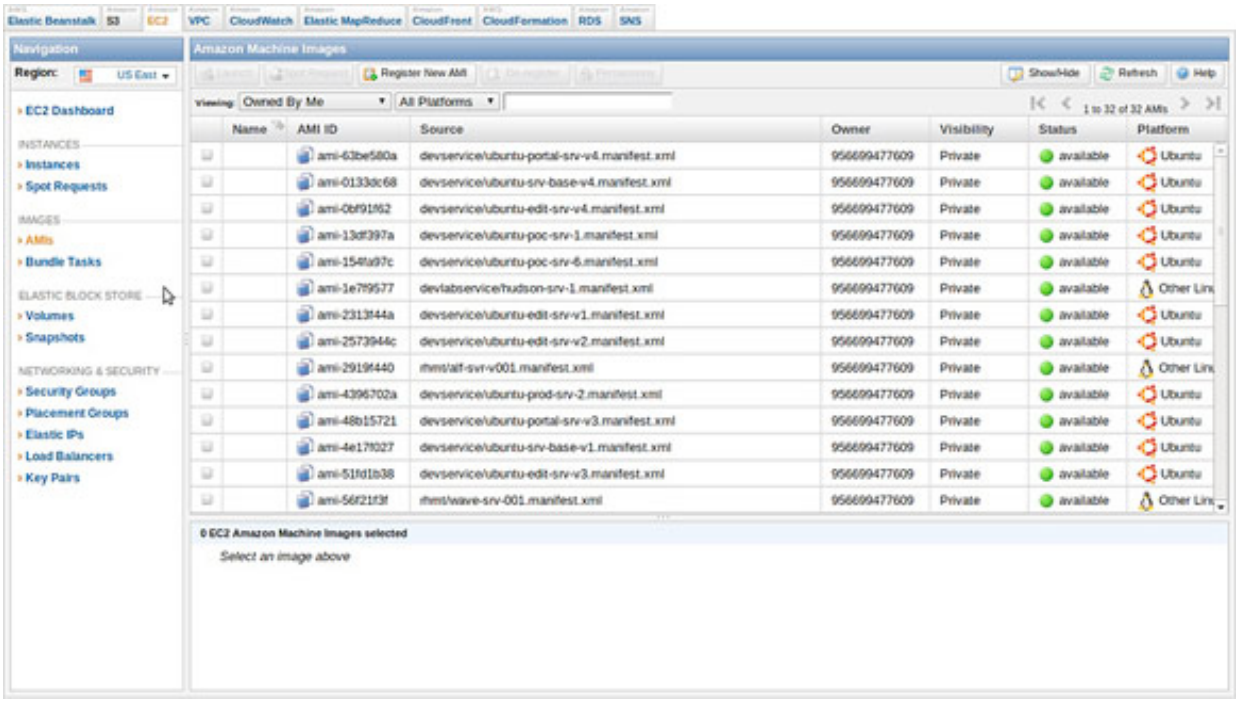

Figura 2.7: Interfaz de Amazon EC2.

Otra gran empresa que ha contribuido enormemente en la llegada del Cloud Computing ha sido Google (figura 2.8). Para Google, que ha sido la empresa que más ha crecido en la última década, permitir a sus usuarios acceder a cualquier búsqueda que deseen desde cualquier parte del mundo, de una manera rápida, era una gran prioridad para mantener su liderato con respecto a otros competidores. Para ello, no podían tener los datos en servidores aislados, sino disponer de grandes bases de datos virtualizadas, y a las que se pudiera acceder en cualquier momento. Como solución a este problema, Google ha construido "fábricas de información" mediante granjas de servidores en distintas partes del mundo, que permiten el acceso a la información desde cualquier lugar. Hoy en día, cualquier empresa puede alquilar servicios de Cloud Computing y utilizarlos, pagando por tiempo y servicio usado, y sin tener que adquirir infraestructuras que después no vuelvan a ser utilizadas [16].

Figura 2.8: Interfaz de Google.

Son numerosas las ventajas que ofrece el Cloud Computing, y es por eso que la oferta de servicios en este ámbito es cada vez mayor, y también se consume más. Sin lugar a dudas, en los próximos años se moverán millones de dólares a través de este servicio, que hay quien lo compara con el descubrimiento de la energía eléctrica, recogiendo nuevamente la idea de John McCarthy de la importancia que podría tener la computación en la nube.

2.4. Características del Cloud Computing

En este epígrafe hablaremos de las características más importantes del Cloud Computing. Para NIST (National Institute of Standards and Technology), los cinco aspectos más importantes del Cloud Computing son los siguientes [17]:

1. On-demand self-service (Autoservicio bajo demanda): Este término se utiliza para expresar que los proveedores de cloud computing permiten que los usuarios tengan acceso a los recursos de la nube bajo demanda cuando éstos lo requieran. Es una característica principal de la mayoría

de los servicios en la nube donde el usuario puede ampliar las capacidades de la infraestructura que ha demandado hasta un nivel sustancial sin interrumpir las operaciones que ya estuvieran en marcha.

2. Rapid Elasticity (Elasticidad rápida): se define como la capacidad de escalar los recursos tanto hacia arriba como hacia abajo según sea necesario. La nube aparece como un recurso ilimitado para el consumidor, y puede comprar tanto o tan poco poder de procesamiento como necesite en el momento que lo requiera; y además, modificar este en cualquier momento.

3. Broad network access (Acceso amplio de red): Este aspecto se refiere a los recursos alojados en una nube privada (operado dentro del firewall de la empresa) disponibles para el acceso desde una amplia gama de dispositivos que tengan acceso a la red, y permitiendo la interconexión entre las personas que tienen acceso a la nube. Por otro lado, las empresas que tienen acceso amplio a la red dentro de la nube necesitan hacer frente a ciertos problemas de seguridad. A menudo, las empresas optan por un servicio de nube privada ya que les preocupa la posibilidad de fugas de información a través de los agujeros de seguridad que dejan abiertos a redes externas si esta fuese una nube pública. Este aspecto será nuevamente nombrado en el epígrafe referido a las limitaciones del Cloud Computing.

4. Resource pooling (Puesta en común de recursos): Los recursos informáticos que posee el proveedor se utilizan para servir a múltiples consumidores mediante un modelo "multi-tenant" tal y como se ha explicado en la sección 2.2.3., con diferentes recursos físicos y virtuales dinámicamente asignados y reasignados, de acuerdo a la demanda de los consumidores. El Cloud Computing, a pesar de esto, abstrae la ubicación al consumidor ya que éste no tiene control o conocimiento sobre la localización exacta de los recursos proporcionados. Ejemplos de recursos podrían ser el almacenamiento, el procesamiento, la memoria, ancho de banda y/o máquinas virtuales.

5. Measured Service (Servicio medido): El uso de un sistema en la nube se puede controlar automáticamente y así medir el uso que ha hecho el usuario del servicio con la idea de que este pague exclusivamente por esto. El uso de recursos puede ser monitorizado y controlado, proporcionando transparencia tanto para el proveedor como para el consumidor del servicio utilizado.

2.5. Clasificación de modelos de Cloud Computing

En esta sección, se explicarán las diferentes clasificaciones de modelos de Cloud Computing establecidas por el NIST (National Institute of Standards and Technology). Para empezar debemos dividir las clasificaciones de modelos de la nube en modelos de servicios y modelos de implementación [17].

2.5.1. Modelos de servicio

Un modelo de servicios describe los diferentes tipos de nube a nivel de responsabilidades de cada recurso y la gestión y utilización de las redes. También delimita el tipo de servicio que el usuario recibe, y qué tipo de recursos se ofrecen. A continuación, se explican los tres modelos de servicio principales, tal y como se muestran en la figura 2.9, en los que se puede clasificar la nube [17]:

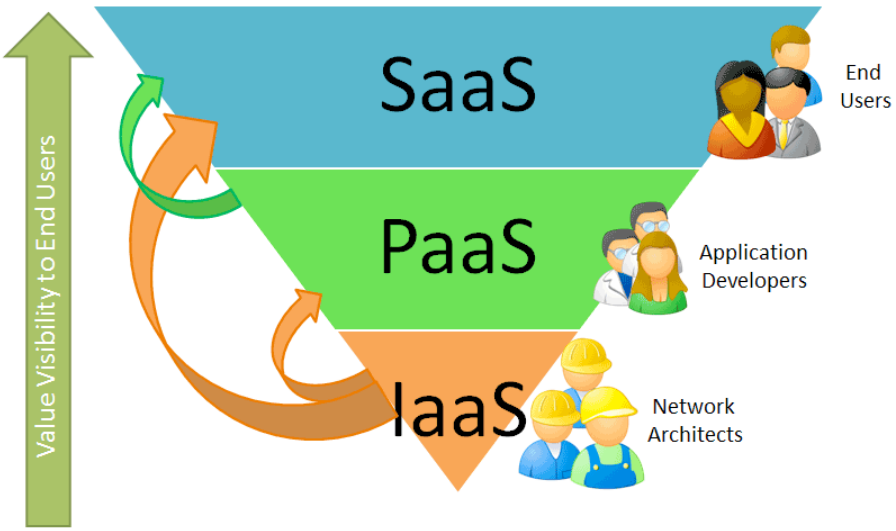

Figura 2.9: Modelos de servicio de la nube.

Infraestructura como servicio (IaaS): Se ofrece al consumidor la capacidad de procesamiento, almacenamiento, redes y otros recursos fundamentales de computación, los cuales el

consumidor puede usar para desplegar y ejecutar el software que desee, que puede incluir sistemas operativos y aplicaciones. El consumidor no tiene control sobre la infraestructura que se le ofrece, pero sí sobre los sistemas operativos, almacenamiento, aplicaciones desplegadas, y el control posiblemente limitado de componentes de red seleccionados (por ejemplo, servidores de seguridad host) [18]. Algunos ejemplos de infraestructuras en la nube son: Amazon Elastic Compute Cloud (EC2) [15], Gogrid [19] y Rackspace Cloud [20].

Platform como servicio (PaaS): La capacidad ofrecida al consumidor es el despliegue de la infraestructura de la nube con aplicaciones ya creadas o adquiridas utilizando lenguajes de programación y herramientas de apoyo por parte del proveedor. El consumidor no administra ni controla la infraestructura de la nube subyacente, es decir, que no tiene control sobre la red, servidores, sistemas operativos, o sobre el almacenamiento. Sin embargo, si el servicio así lo especifica, sí que tiene control sobre las configuraciones del entorno de alojamiento de aplicaciones [21]. Windows Azure [22], Force.com [23], y Google App Engine [6] son algunos ejemplos de plataformas en la nube.

Software como Servicio (SaaS): Al consumidor se le ofrece el uso de las aplicaciones del proveedor que se ejecutan en una infraestructura en la nube. Las aplicaciones son accesibles desde diferentes dispositivos cliente a través de una interfaz de cliente. El usuario no tiene control sobre la plataforma (PaaS) ni la infraestructura (IaaS) en las que el software se ejecuta [24]. Algunos ejemplos de Software como servicio son Google Apps [5], Dropbox [3], y SalesForce.com [23]. Además, debemos destacar que este será el modelo de la aplicación en la que se va a basar este proyecto.

Para terminar, cabe destacar la aparición de nuevos tipos de modelos que ya no se incluyen en estos tres principales, pero que en un estudio del Cloud Computing tambien deberíamos conocer. Algunos de los nombres que podemos mencionar son:

- StaaS: Storage as a Service.
- IdaaS: Identity as a Service.
- CmaaS: Compliance as a Service.

2.5.2. Modelos de implementación

Un modelo de implementación define el propósito de la nube y el modo en que ésta se localiza. La nube se puede clasificar según su modelo de implementación (figura 2.10) en cuatro tipos diferentes, que son los siguientes [1][17]:

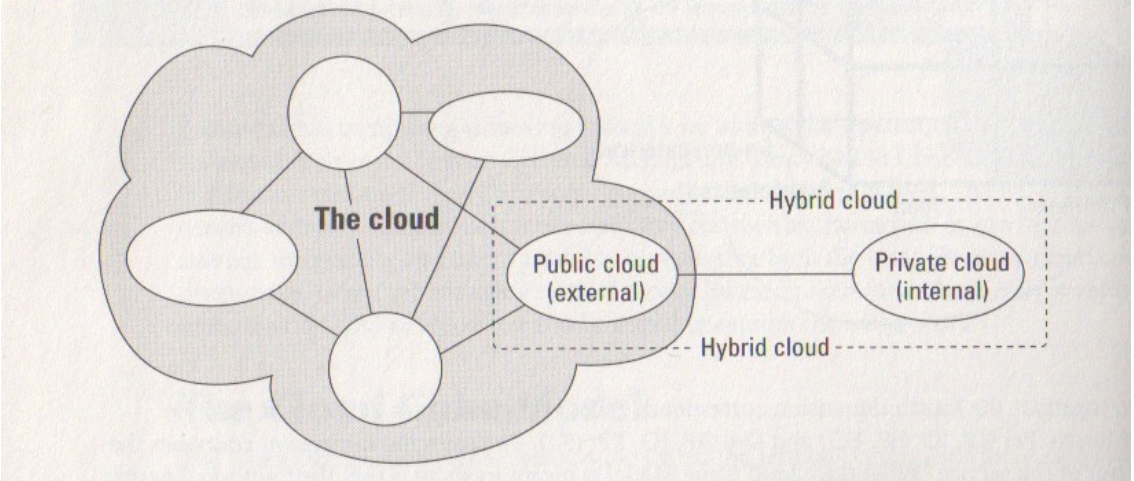

Figura 2.10: Modelos de implementación de la nube [1].

Cloud privada: Este tipo de nube es explotada solamente por una organización, de tal manera que el acceso a la información solo puede llevarse a cabo desde ésta. Puede ser administrada por la organización o por un tercero.

Cloud comunitaria: La nube es compartida por varias organizaciones y apoya a una comunidad específica que comparte información (por ejemplo, la misión, los requisitos de seguridad, la política, y las consideraciones de cumplimiento). Puede ser gestionada por las propias organizaciones que la utilizan o por un tercero. Este tipo de nube también puede considerarse un modelo de cloud privada para uso compartido a nivel de comunidad.

Cloud pública: Es el modelo más extendido, y en él la infraestructura de la nube se pone a disposición del público en general o de algún grupo importante de la industria. En este caso, la nube es propiedad de la organización que ofrece los servicios a los clientes.

Cloud híbrida: La infraestructura en la nube es una composición de dos o más tipos de nube (ya sea privada, comunitaria o pública) que permanecen como entidades únicas, pero están unidas por alguna tecnología y que permite el paso de datos y la portabilidad entre aplicaciones.

2.6. Arquitecturas, Plataformas y Servicios

Aquí profundizaremos sobre los modelos de servicio que se comentaron en el apartado 2.5.1. Destacamos este apartado sobre los demás, por el simple hecho de que esta clasificación nos define la estructura y arquitectura básica del Cloud Computing, definiendo con ello también los diferentes modelos de servicio de Cloud que existen, y a quién van dirigidos. La sección se divide en tres apartados, uno por cada uno de los principales modelos de servicio que ya se han nombrado, que se explican a continuación.

2.6.1. Infrastructure as a Service (IaaS)

La infraestructura como servicio (Iaas) no es conceptualmente nueva. Siempre se ha intentando centralizar la información desde que ésta ha existido. Lo que es diferente con IaaS son las herramientas que hay detrás y que nos permiten hacer realidad las bases teóricas que planteaba John McCarthy.

La infraestructura como servicio proporciona un mecanismo para que las personas sustituyan la totalidad de sus necesidades de hardware por hardware en un centro de datos, de manera que, nosotros desde un terminal podamos acceder virtualmente al hardware que el proveedor nos proporciona, sin disponer físicamente de éste [25]. Algunas de las principales características de la infraestructura como servicio son:

- Posibilidad de almacenamiento de información.
- Servicios añadidos a la computación de utilidad y modelos de facturación online.
- Posible automatización de tareas administrativas.
- Escalabilidad dinámica.
- Virtualización de escritorios.
- Servicios basados en políticas determinadas.

- Conectividad a Internet.
- Aprovisionamiento de hosting.
- Conectividad a redes públicas y/o privadas ya configuradas.
- Cortafuegos administrados con el sistema.

Todos los servicios que subyacen de éstos también se proporcionan, es decir, que incluye la supervisión, energía, refrigeración, reparación, seguridad, seguimiento de inventarios, y quizás lo más importante, los trabajadores que se encargan de ello. Por ejemplo, si en casa un usuario instala un servidor, su mantenimiento y reparación corren de su cuenta; sin embargo, esto no es así en la nube, ya que estas labores serían llevadas a cabo por los trabajadores profesionales que trabajen para la empresa proveedora. Esto ofrece una nueva ventaja, que es la profesionalización de los recursos de los que disponemos [26].

La mayoría de los proveedores de IaaS tienen soluciones convenientes para diversificar geográficamente los recursos informáticos. Esto les permite beneficiarse de ventajas competitivas trabajando en diferentes lugares, donde les es más rentable económicamente, pero también otro tipo de ventajas, como pueden ser de ahorro energético. Por ejemplo, Google dispone de centros de datos enormes en países en el norte de Europa, debido a que las temperaturas bajas evitan el sobrecalentamiento de los sistemas y los correspondientes gastos en refrigeración [27].

Todo esto lleva a permitir servicios, que de ninguna manera serían posibles con la informática tradicional. Los costes de estos servicios, se pagarían por tiempo y servicio; por tanto, no gastaríamos ni un céntimo más de lo necesario en labores, que de cualquier otro modo, nos podrían resultar muy costosas e incluso imposibles de costear. Además, al ser servicios a través de la red, es importante destacar que pueden llevarse a cabo veinticuatro horas al día los trescientos sesenta y cinco días del año, independientemente de domingos, días festivos u horas de descanso, proporcionando eficiencia en todos los sentidos a los consumidores [28].

A continuación, se van a nombrar algunos ejemplos de Infraestructuras como servicios en la nube:

OpenNebula: OpenNebula.org es un proyecto de código abierto que propone una solución standard a la industria para la creación y gestión de centros de datos virtualizados empresariales y nubes privadas empresariales.

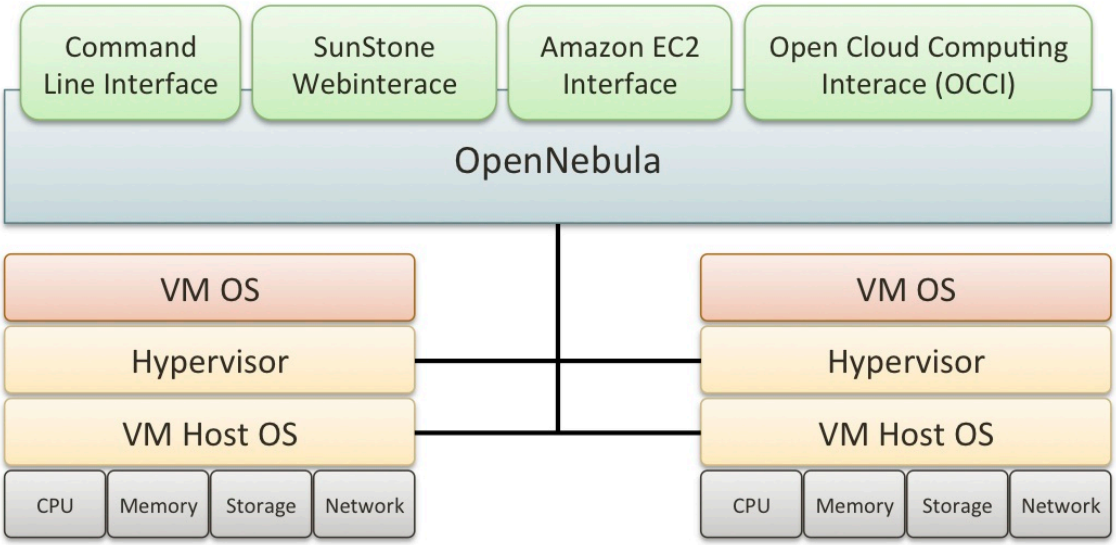

Figura 2.11: Estructura de un sistema implementado con OpenNebula [29].

La interoperabilidad en OpenNebula permite que el usuario disponga de la infraestructura infórmática que tenga ya existente y pueda acceder siempre a ésta, protegiendo así sus inversiones y evitando la dependencia de un proveedor. OpenNebula tiene como objetivo proporcionar una capa de gestión abierta, flexible, extensible y completa para automatizar y gestionar el funcionamiento de centros de datos virtualizados mediante el aprovechamiento de soluciones de marketing desplegadas y de integración existentes para la creación de redes, almacenamiento, virtualización, monitorización y gestión de usuarios [29].

La estructura de Openebula puede observarse en la figura 2.11. En el nivel más bajo se encuentran los elementos físicos como la CPU, la memoria, la unidad de almacenamiento y la red. La siguiente capa es la virtualización del dispositivo físico (VM Host OS), que mediante un hipervisor consigue un nivel de virtualización (VM OS) a nivel del sistema operativo. Es sobre esta capa donde actúa Opennebula.

Amazon Elastic Compute Cloud (Amazon EC2): es un servicio web que permite a sus usuarios alquilar ordenadores virtuales para ejecutar sus propias aplicaciones informáticas Con ello,

proporciona capacidad informática en la nube con tamaño modificable, es decir, a elección del usuario. Está diseñado para facilitar a los desarrolladores recursos informáticos escalables basados en la Web. Encontramos un ejemplo de aplicación software implementada sobre Amazon EC2 en la figura 2.12.

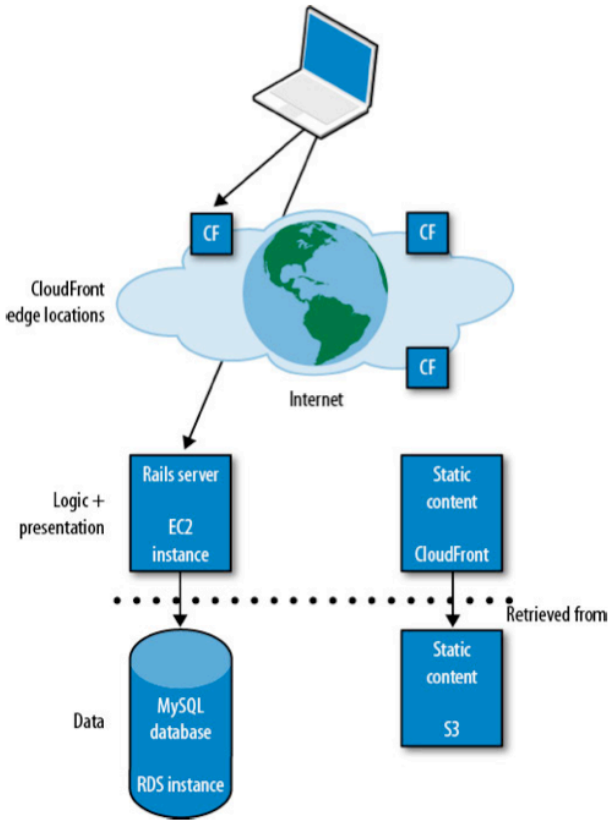

Figura 2.12: Arquitectura de Kulitzer, un programa creado sobre Amazon EC2.

La interfaz de servicios web de Amazon EC2 permite obtener y configurar su capacidad de una manera sencilla, además de proporcionar un control completo sobre sus recursos informáticos. Un usuario puede crear, iniciar y finalizar las instancias de servidor, según le sea necesario. Esto permite escalar rápidamente la capacidad, ya sea aumentándola o reduciéndola, en función de las necesidades del consumidor. Amazon EC2 permite pagar al usuario sólo por la capacidad que utiliza realmente. En noviembre de 2010, Amazon migró su propio sitio Web a AWS y EC2 [15].

2.6.2. Platform as a Service (PaaS)

A diferencia de las infraestructuras y el software como servicio, las plataformas como servicio (PaaS) son un concepto mucho más abstracto. En una estructura piramidal, la plataforma ocuparía la zona central de la pirámide invertida, pues no sería ni solamente el hardware, ni solamente el software, tal y como se mostró anteriormente en la figura 2.9.

Los proveedores de PaaS ofrecen al público una plataforma, es decir, una infraestructura con algunos servicios y tecnologías añadidos que lo complementen. Por ejemplo, un proveedor de PaaS se encarga de proporcionar al consumidor todo lo necesario para desarrollar software en un lenguaje específico o sobre una tecnología en concreto [30]. Muchos proveedores de PaaS proporcionan a sus clientes soluciones muti-tenant, característica que ya se ha comentado en este mismo capítulo [31], ofreciendo así a cada usuario una instancia virtualizada de la plataforma.

El objetivo de PaaS es proporcionar al consumidor los medios, herramientas, bibliotecas y tecnologías necesarias para desarrollar el software que quiere ofrecer al usuario final. PaaS ofrece esto evitando al usuario pagar los costes de la infraestructura y la complejidad que llevarían su instalación y mantenimiento. La plataforma como servicio puede incluir también instalaciones para el diseño y desarrollo de aplicaciones, así como servicios tales como la colaboración en equipo, integración de servicios web y la integración de bases de datos entre otros [32].

A continuación, y tal como se mostró en el apartado de IaaS, se muestran algunos ejemplos de plataformas como servicios en la nube. Además, al final de la sección se dedicará un apartado completo a Google App Engine, por ser la plataforma sobre la que se realiza este proyecto.

Force.com: Force.com es la progresión natural de Salesforce.com, un sistema CRM (Customer relationship management) para empresas ofrecido como un servicio (SaaS). La necesidad de ir más allá de la aplicación SaaS ofreciendo no solo el software en la nube, sino también una plataforma para la creación de software por parte de terceros ha dado lugar a un cambio radical del modelo de prestación de Salesforce.com de SaaS a PaaS, cuyo resultado es Force.com [23]. Se muestra su interfaz en la figura 2.13.

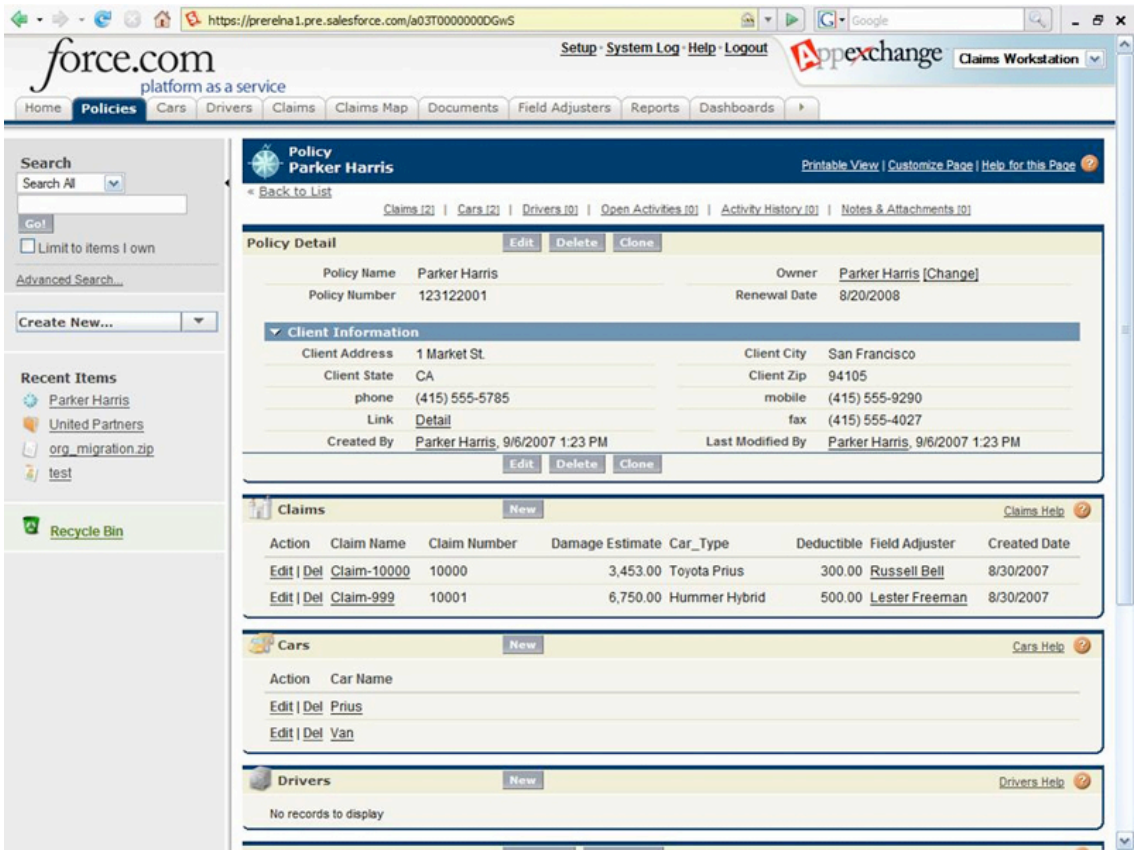

Figura 2.13: Interfaz de Force.com.

OpenShift: OpenShift es una plataforma libre como servicio en la nube de la empresa Red Hat. Esta plataforma se puede utilizar de dos maneras: O bien se utiliza la infraestructura pública que Red Hat ofrece, o bien se crea un propio IaaS sobre el que utilizar OpenShift como PaaS, como aparece en la figura 2.14. Los desarrolladores pueden usar Git[1] para cargar sus aplicaciones Web desarrolladas en los diferentes lenguajes que la plataforma ofrece, disponiendo así de la infraestructura y servicios necesarios para desarrollar sus aplicaciones [33].

[1] Sitio Web oficial de Git: http://git-scm.com/

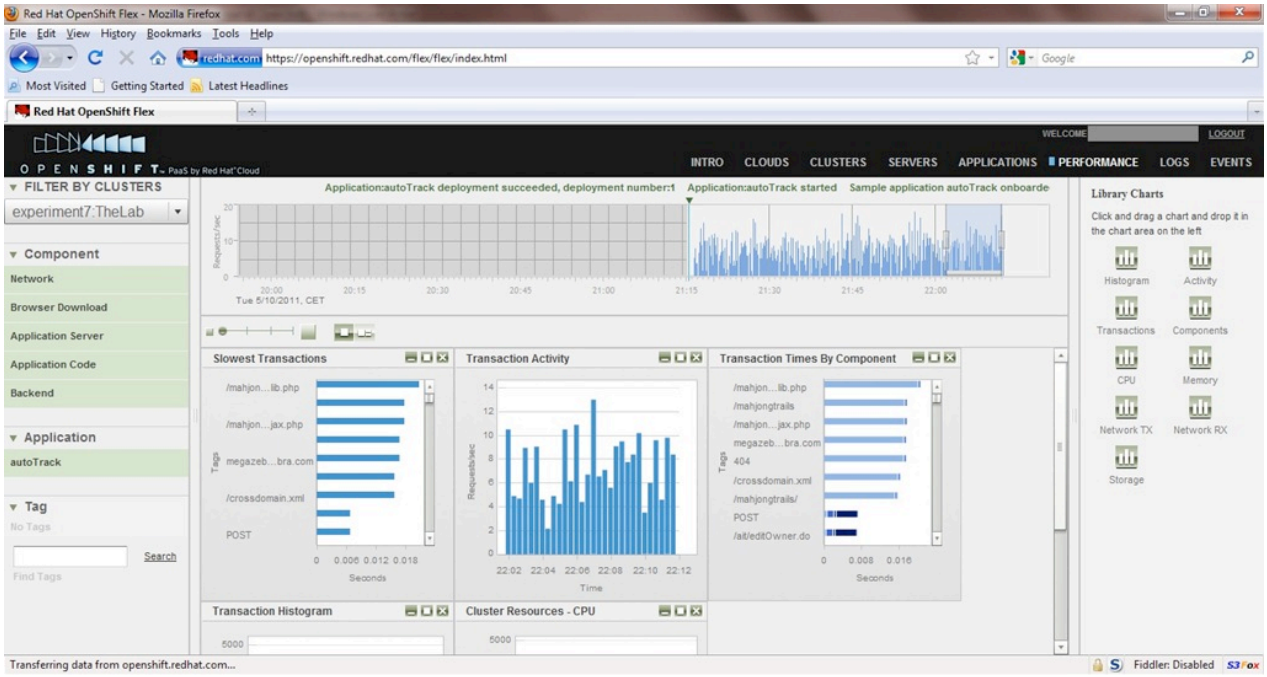

Figura 2.14: Interfaz de OpenShift.

2.6.2.1.Google App Engine

Google App Engine fue creada en abril de 2008 con la idea de que la enorme infraestructura de Google fuera puesta a disposición del público en general simplemente disponiendo de una cuenta de correo gmail. Esto daría la posibilidad a muchas personas ajenas a la empresa de desplegar sus aplicaciones en la Web a través de los servicios y la experiencia que ofrece el gigante de Internet. La plataforma de desarrollo de Google permite a los usuarios crear y alojar aplicaciones web escalables en la nube y sin necesidad de disponer de un hardware subyacente complejo y costoso, abstrayendo también su localización y mantenimiento al cliente, concentrándose éste simplemente en su propia aplicación. Google se encarga de todas éstas tareas [6].

La plataforma comenzó desde su origen como una plataforma en la nube. Esto se diferencia de otros servicios en la nube como Amazon Web Services o Azure de Microsoft, pues el servicio ofrecido por Google es desde el primer momento un servicio PaaS y no IaaS a diferencia de lo ocurrido en otras empresas. Esto significa que Google ha apostado desde el primer momento por ofrecer una plataforma orientada a la creación de software por parte de terceros.

Google App Engine o también más conocido comúnmente como GAE, como nos vamos a referir a ella a partir de ahora, pone a nuestra disposición la infraestructura informática de Google como plataforma de desarrollo y hospedaje de aplicaciones web de manera gratuita, hasta 500 MB de almacenamiento y suficiente CPU y ancho de banda como para permitir un servicio eficaz de alrededor de cinco millones de visitas a la página al mes . Si pasamos estos límites , que se renuevan diariamente, tendremos que pagar a Google, si bien es verdad que superar este límite supondría unos costes bastante asequibles al proveedor de la aplicación en relación al número de usuarios que utilizan la aplicación [34]. Además Google nos proporciona un dominio propio para la URL de nuestra aplicación, y toda la información sobre el uso que se ha hecho de nuestra aplicación (tal y como se aprecia en la figura 2.15).

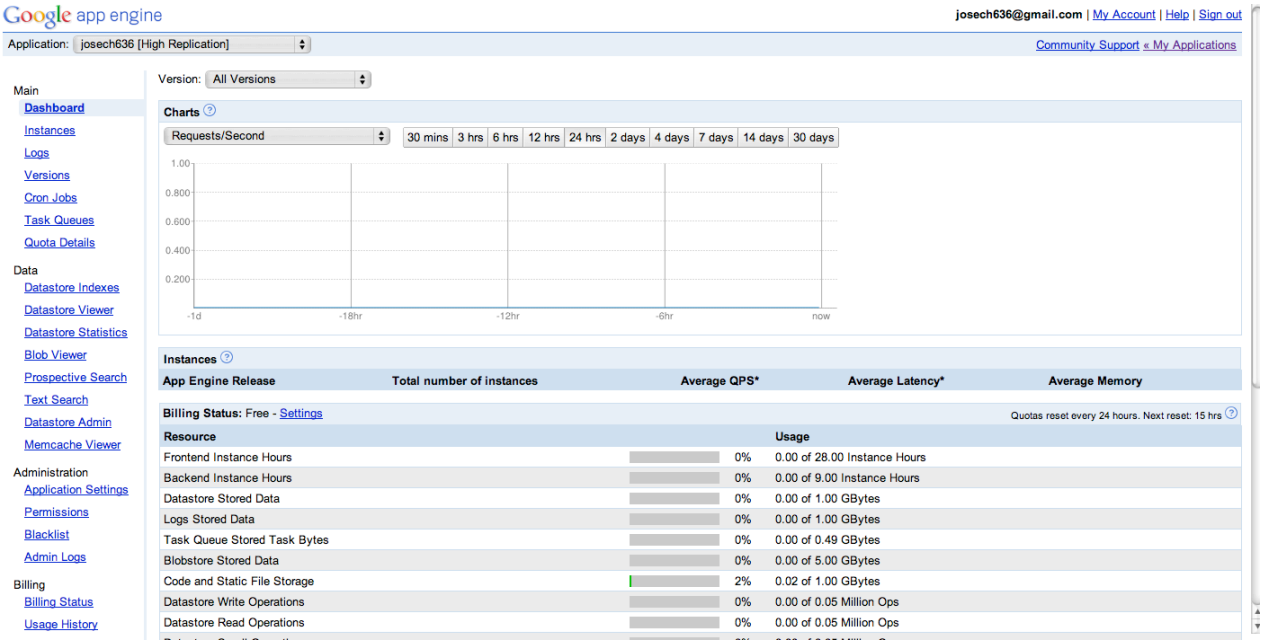

Figura 2.15. Interfaz de Google App Engine.

GAE comenzó ofreciendo Python únicamente como lenguaje de programación para la plataforma. Sin embargo, poco después se añadió Java como "segunda lengua oficial" y, debido a su potencial y número de programadores que utilizan esta tecnología, permitió el crecimiento y expansión de GAE. Al soportar Java, esto incluye también los lenguajes que se pueden ejecutar

sobre su máquina virtual (PHP, JRuby y Perl son algunos ejemplos) y algunos de sus frameworks más utilizados [34].

Google App Engine tiene una estructura diferente a la aplicación web cliente - servidor clásica. En su esencia, GAE restringe cualquier acceso a la infraestructura física por parte de las aplicaciones que se desplieguen en ella. De esta forma la infraestructura queda abstraída de cualquier proceso gestionado por el usuario de la plataforma. Esto también ofrece grandes ventajas a su seguridad, pues impide abrir sockets, ejecutar procesos en segundo plano (si bien se puede realizar si el cliente lo desea), entre otros [35]. GAE está diseñado también para hacer frente a las preocupaciones del usuario acerca de la escalabilidad y fiabilidad. Se basa en el concepto de escalamiento horizontal, que, en esencia, significa que en lugar de ejecutar la aplicación en un hardware más potente, se ejecuta sobre diversas instancias con un hardware menos potente [35].

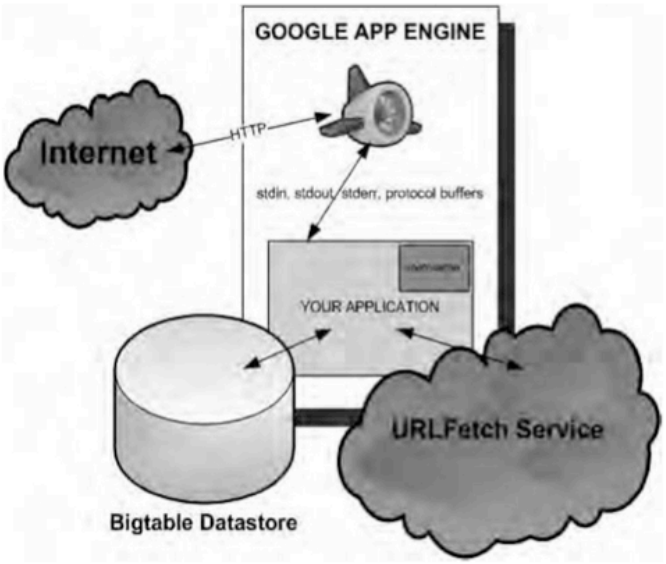

Figura 2.16. Estructura de Google App Engine.

Se ofrece una imagen de la arquitectura utilizada en Google App Engine en la figura 2.16. GAE comparte recursos entre múltiples aplicaciones, pero aísla los datos y la seguridad a través de diferentes inquilinos también (multi-tenant). Hay que tener en cuenta que con GAE sólo se tiene acceso a la capa de aplicación. Hay algunos proyectos de código abierto, por ejemplo, el sistema

de archivos virtual Google, que permite el acceso a una unidad virtual emulada con GAE, pero de momento no se ofrece dentro de los servicios ofrecidos con la plataforma [35].

El SDK (Kit de Desarrollo de Software) de Google App Engine puede descargarse como un entorno de desarrollo que puede ser ejecutado de forma local en nuestra máquina antes de subir los cambios a la nube y que simula completamente el entorno del App Engine. GAE también permite instalar un plugin para el entorno de desarrollo Eclipse, integrando el SDK en su entorno y facilitando al usuario el trabajo desde Eclipse para la implementación de código. Por lo cuál, desde este entorno el usuario puede fácilmente programar la aplicación y desplegar tantas versiones como desee [34] [35].

2.6.3. Software as a Service (SaaS)

El Software como Servicio (SaaS) se sitúa en el nivel más alto de los modelos de servicios en la nube, tal y como se observó en las figuras 2.1 y 2.9. Las aplicaciones se ofrecen como software alojado en un host remoto del que el usuario no tiene conocimiento y al cual accede a través de Internet. Por lo tanto, el usuario no invierte por adelantado en una infraestructura informática ni tiene que mantener su software con actualizaciones regulares. Todo esto corre a cargo de la empresa proveedora del SaaS en cuestión. Estas aplicaciones de negocio normalmente proporcionan un nivel de configuración que permite personalizar la aplicación para los usos y recursos que el consumidor o usuario necesite mediante la propiedad “multi-tenant” que anteriormente se explicó en este mismo capítulo [24] [36].

La idea de "software como servicio" no es nueva, pero sí que lo es el término SaaS. SaaS se refiere simplemente a software que se proporciona por su demanda de uso. Tradicionalmente, cuando alguien quería usar software debía ir a la tienda, recoger el CD, y posteriormente, instalarlo en su propio ordenador. Con SaaS, sólo tiene que utilizar el software alojado en la nube y del que no tiene ni la más mínima idea de donde se sitúa físicamente, pues este proceso es abstracto al usuario. No hay instalación, no hay actualizaciones, y por tanto, no hay complejidad. Otra enorme ventaja del software en la nube es la consistencia de los datos de una empresa, ya que el software es el mismo en todas las sedes y departamentos de ésta. La reducción de los gastos en sistemas informáticos gracias al software como servicio también es una ventaja a tener cuenta [36] [37].

Los proveedores de SaaS se encargan de todo el trabajo de mantenimiento y supervisión que hay detrás del funcionamiento de este tipo de aplicaciones, es decir, que nuevamente éstas son abstractas al usuario.

Algunos ejemplos dentro de esta categoría son:

Dropbox: Dropbox es un servicio gratuito que permite almacenar nuestra información en la nube. También dispone de aplicación de escritorio, que sincroniza nuestros documentos de la propia carpeta del sistema con la nube, por lo que no hay que preocuparse por su sincronización ya que ésta se realiza de manera automática. Para terminar, también podemos compartir documentos con otros usuarios mediante la opción de compartir carpetas con alguien. A Dropbox, cuya interfaz se presenta en la figura 2.17, se le han presentado algunas de las empresas informáticas más importante como competidores en su segmento, así que otras opciones a Dropbox serían Google Drive de Google, ICloud de Apple y Skydrive de Microsoft [3].

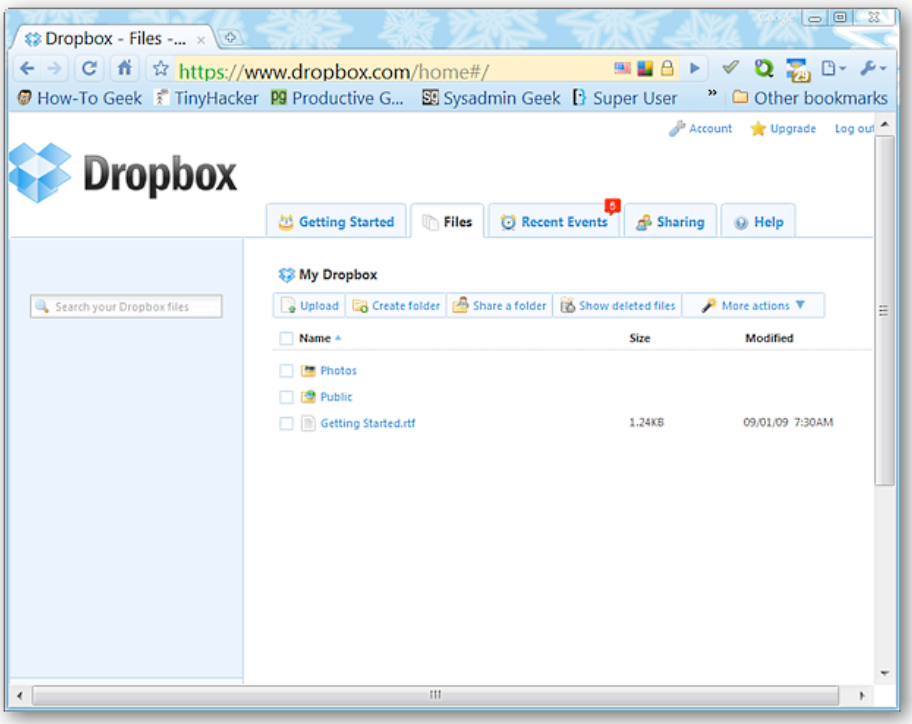

Figura 2.17: Interfaz de Dropbox

Google Apps: El sistema de documentos de Google es en sí una aplicación en la nube, pues desde cualquier lugar podemos acceder a nuestros documentos, crear nuevos, modificarlos y eliminarlos.

Google nos proporciona el software necesario sin necesidades de instalación ni actualización, con lo cual ya no tenemos necesidad de adquirir la última versión de herramientos ofimáticas, algo muy común hace menos de una década, teniendo nuestros documentos siempre disponibles [5]. Se muestra su interfaz en la figura 2.18.

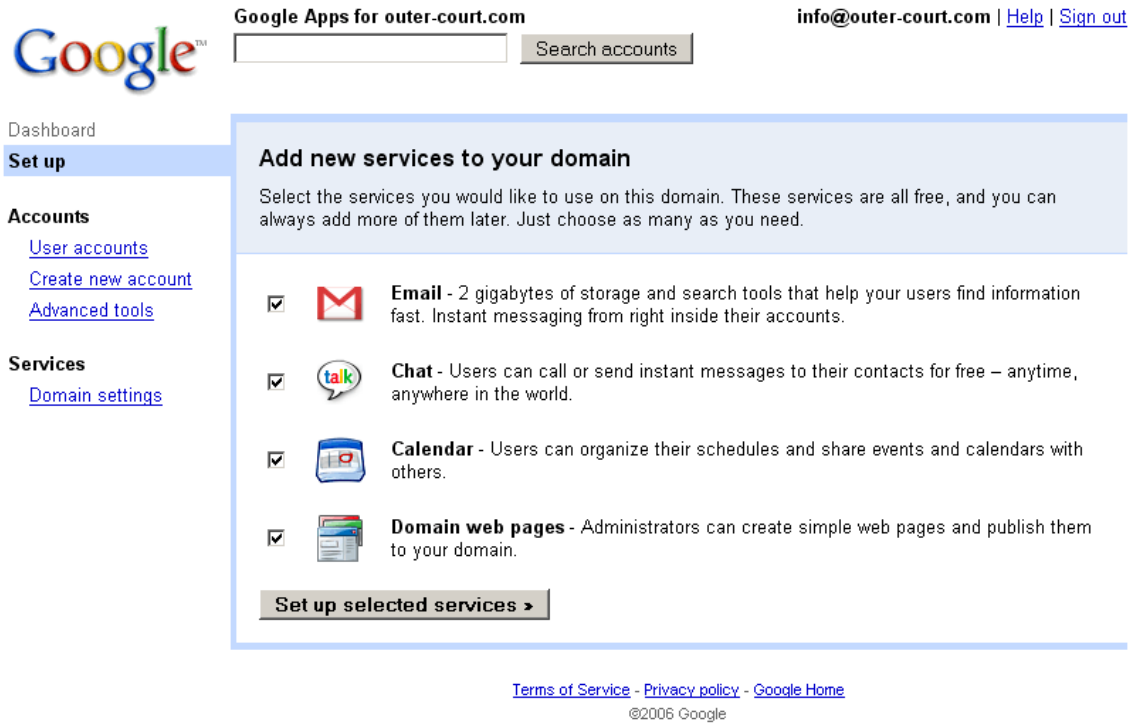

Figura 2.18: Interfaz de Google Apps

2.7. Limitaciones del Cloud Computing y retos para el futuro

La limitación principal del Cloud Computing, es que todo lo que podemos hablar de bueno sobre él, puede esfumarse completamente debido solamente a no disponer de una conexión a Internet. Hoy en día, Internet sigue en auge, cada día disponemos de más capacidad, velocidad y de más dispositivos para conectarnos. Además, las grandes empresas han apostado por proporcionar los medios e infraestructuras necesarias para que cada vez más gente pueda

disfrutar de una conexión a la red. Es el caso de Google, que en fase experimental ha lanzado enormes globos de helio en Nueva Zelanda que posibilitan la conexión a Internet desde áreas de difícil acceso con la intención de que sean muchas más personas las puedan conectarse a la red (Proyecto Loon - Figura 2.19) [38].

Figura 2.19: Propósito general del proyecto "Loon" de Google [38].

Este problema no se ciñe solo a si hay o no Internet, sino que además, la conexión debe ser de banda ancha. De nada vale disponer de conexión a Internet si este está limitado a 1 Gb al mes y después va a funcionar a menos velocidad o con menos prestaciones, pues el Cloud Computing requiere bastante información, y con ello, de una gran capacidad para descargar y subir contenidos a la nube.

Además, otro punto a mejorar, es la seguridad. Bien es cierto que a día de hoy las empresas disponen de suficientes medios para proteger los datos en la red. Pero no olvidemos que la historia muestra que entre hackers y empresas hay una guerra continua que dependiendo del momento y los avances van ganando unos u otros [39]. Por lo pronto, los documentos que compartamos en la nube deberían no ser comprometidos, o si bien no nos queda más remedio, subirlos de manera cifrada manteniendo en todo momento las claves fuera de la nube.

A veces, que un servicio se haga lo suficientemente conocido como para que muchos más usuarios de lo habitual se registren y lo utilicen en poco tiempo, tiene como consecuencia que la infraestructura no esté preparada para esa cantidad de usuarios y se produzcan por tanto caídas que se traducen en la imposibilidad de usar el servicio por parte de los consumidores. Normalmente estos períodos de tiempo suelen ser bastante pequeños, pero conviene tenerlos en cuenta si necesitamos acceder a algún servicio de manera urgente.

En último lugar, debemos destacar que la distribución de las aplicaciones y el almacenamiento de los datos origina una interdependencia de los proveedores en todo momento.

Como conclusión, decir que las ventajas que el Cloud Computing ofrece son enormes, pero las limitaciones aún siguen siendo bastante importantes. Obviamente con el tiempo se irán resolviendo estas cuestiones e irán surgiendo nuevos problemas, ya que el Cloud Computing se encuentra en su fase de crecimiento y aún le queda mucho por madurar, pero no podemos olvidar estas limitaciones y usar la nube de manera apropiada y sin poner en riesgo nuestra información.

CAPÍTULO 3:

Desarrollo del proyecto

3.1 Descripción del Proyecto

En este capítulo se asentarán y explicarán los procesos de Ingeniería del Software llevados a cabo para alcanzar el objetivo del proyecto. Como establecimos en el capítulo de introducción, el propósito de este proyecto es el diseño y desarrollo de una aplicación de escritorio para la gestión de tareas académicas basada en el paradigma Cloud Computing. Tras haber explicado en el capítulo anterior en qué consiste el Cloud Computing, procedemos a mostrar cada una de las fases de la Ingeniería de Software que se han tenido en cuenta para la realización del proyecto en cuestión.

A continuación, vamos a definir la Ingeniería del Software para comprender mejor los epígrafes que se integran en este capítulo y el objetivo de cada uno de ellos.

No existe una definición única y estandarizada para la Ingeniería del Software. A pesar de esto, las dos definiciones que a continuación vamos a conocer resultarán perfectamente válidas para cumplir nuestros objetivos y entender mejor el propósito de este capítulo.

La Ingeniería del Software consiste en la construcción de software de calidad con un presupuesto limitado y un plazo de entrega en contextos de cambio continuo [40].

La Ingeniería del Software es el establecimiento y uso de principios y métodos firmes de ingeniería, para obtener software económico que sea fiable y funcione de manera eficiente en máquinas reales [41].

La Ingeniería del Software se compone de las siguientes actividades [42]:

Especificación de requerimientos: Se obtiene el propósito del sistema a desarrollar, así como sus propiedades y restricciones que se han impuesto.

Análisis del sistema: Se obtiene un modelo del sistema de manera completa, correcta, consistente, clara y verificable.

Diseño del sistema: Se establecen los objetivos del proyecto y las estrategias a seguir para conseguirlos.

Implementación: Consiste en la transformación del modelo conceptual del sistema al código fuente en sí.

Pruebas: Verifican y validan el funcionamiento del sistema.

Sin más demora, comenzamos a describir cada una de las fases anteriores en el desarrollo de nuestro proyecto.

3.2 Especificación de Requerimientos

La Especificación de Requerimientos es la descripción completa del comportamiento del sistema que se va a desarrollar, mediante la descripción de cada una de las interacciones que tendrá el usuario con el sistema. Los requerimientos, a su vez, pueden dividirse en requerimientos funcionales y requerimientos no funcionales. Ambos vamos a tratarlos en los siguientes subepígrafes [43].

3.2.1 Especificación de Requerimientos funcionales

Los requisitos funcionales de un sistema software son aquellos que establecen el comportamiento de dicho sistema y definen la funcionalidad que éste debe proporcionar a los usuarios [42].

Las funcionalidades que el usuario potencial, esto es, el estudiante, podría esperar de nuestro sistema son las siguientes:

- RF-01: Registro en el sistema.
- RF-02: Validación de usuario.
- RF-03: Gestión de asignaturas.
- RF-04: Gestión de tareas.
- RF-05: Gestión de prácticas.
- RF-06: Gestión de exámenes.
- RF-07: Gestión del perfil del usuario.

• RF-08: Cerrar sesión.

A continuación se explica cada una de estas funcionalidades de manera específica y detallada.

1. RF-01: Registro en el sistema.

El usuario puede registrarse en el sistema mediante la creación de una cuenta de usuario en el mismo, aportando sus datos, y a partir de ese momento será un usuario registrado en el sistema.

2. RF-02: Validación de usuario.

El usuario puede acceder al sistema con su cuenta de usuario una vez registrado.

3. RF-03: Gestión de asignaturas.

El usuario crea en el sistema las asignaturas que estudia, y podrá modificarlas o eliminarlas en todo momento. Además, se ofrece la posibilidad de adjuntar tareas, prácticas y exámenes a una asignatura.

4. RF-04: Gestión de tareas.

En cualquier momento, el usuario puede añadir tareas con una fecha de realización determinada (por ejemplo, buscar una información que el profesor ha mandado en clase, realizar una actividad, o consultar referencias bibliográficas para adelantar el temario). Además existe la posibilidad de vincular la tarea a una asignatura. El usuario puede modificar las tareas o eliminarlas siempre que lo desee.

5. RF-05: Gestión de prácticas.

El sistema debe ofrecer al usuario la posibilidad de añadir, editar y eliminar las prácticas. Existe la posibilidad de vincular la práctica a una asignatura. Además, a estas se les podrá añadir la fecha de finalización, de tal manera que el usuario sepa en cada momento cuanto falta para la próxima entrega.

6. RF-06: Gestión de exámenes.

El sistema ha de permitir al usuario añadir los exámenes que el usuario vaya a realizar y la fecha de cada uno de éstos; de tal manera que el alumno disponga en la aplicación, en todo momento,

de las fechas de sus exámenes y tenga éstos siempre en mente. Cada examen puede vincularse a una de las asignaturas del usuario.

7. RF-07: Gestión del perfil de usuario.

El usuario debe tener la posibilidad de modificar su nombre de usuario e información académica relativa a sus estudios así como eliminar su cuenta de usuario, borrando así la información relativa a su cuenta en el sistema.

8. RF-08: Cerrar sesión.

El sistema proporciona al usuario esta opción para salir de la aplicación.

3.2.2 Especificación de Requerimientos no funcionales

Los requerimientos no funcionales son aquellas características que posee el sistema en su totalidad, sin referirse a comportamientos o funcionalidades, e incluyen aquellas propiedades deseables que debe presentar el sistema o producto en sí (velocidad, rendimiento, etc.), y las características propias que poseerá la interfaz [42], las cuales se describen en mayor detalle a continuación.

Requerimientos de la Interfaz

Consideramos que el diseño de la interfaz de usuario es una parte fundamental en la creación de nuestro proyecto, ya que será la parte del sistema con la que interactuará el usuario. Es también la parte visual de la aplicación, que será determinante para el cliente en su decisión por utilizar la aplicación. A continuación damos a conocer las características que se van a tener en cuenta a la hora de crear la interfaz [44].

- **Usabilidad aplicada a la interfaz:** Que la aplicación sea intuitiva y fácil de visualizar en cada uno de los servicios que ofrece al cliente.
- **Facilidad de uso:** Es fundamental junto con la usabilidad, puesto que ambas facilitan al usuario el uso de la aplicación, y su comprensión sobre ésta.

- **Visualmente atractiva:** Al igual que recordamos a las personas por su aspecto físico, el aspecto visual de nuestra aplicación será fundamental en la relación que se establezca entre el usuario y ésta. Intentaremos crear una interfaz agradable para el usuario, sin que resulte tan llamativa como para ser molesta. Muchas veces la simplicidad y la facilidad de uso, utilizándose en diversos aspectos de la vida cotidiana, puede facilitar bastante el entendimiento de la aplicación al usuario.
- **Fiabilidad:** Otra característica fundamental de una aplicación, es que sea capaz de responder correctamente en todo momento a las acciones que realiza el usuario. Para ello, realizaremos una serie de pruebas (incluidas al final de este capítulo), que llevaremos a cabo con el objetivo de verificar la fiabilidad de nuestra aplicación en todas las situaciones posibles.
- **Tiempo de respuesta:** Es el tiempo necesario para que el sistema pueda mostrar los cambios realizados por el usuario. Se minimizará lo máximo posible, indicando al usuario que debe esperar para realizar alguna acción si así fuera necesario.

3.2.3. Requerimientos del Equipo Informático

Dado que nuestro proyecto consiste en una aplicación informática de escritorio, se describe a continuación el equipo informático con el que se ha realizado todo el proyecto. Este se divide a su vez en requerimientos software y hardware.

Hardware del Ordenador Personal

- **Velocidad:** el equipo debe ser lo suficientemente rápido como para ejecutar la aplicación en el menor tiempo posible y con la mayor fiabilidad. Cualquier microprocesador actual es capaz de cumplir con esta labor.
- **Memoria:** el equipo ha de disponer de la suficiente memoria RAM libre para realizar las operaciones que se soliciten entre la aplicación y la base de datos.
- **Almacenamiento:** el equipo debe tener una capacidad de almacenamiento suficiente para almacenar la base de datos con la que trabaja la aplicación.
- **Tarjeta gráfica:** las tarjetas gráficas de las que disponen los equipos informáticos actuales son de gran potencia por lo que, no es necesario establecer ningún requerimiento en este aspecto.
- **Monitor:** el monitor debe soportar una resolución de 1024x768 o superior.

- **Conexión a Internet:** la aplicación necesitará conexión a Internet para sincronizar los datos con la nube, así como para realizar tareas como el registro del usuario o la eliminación de su cuenta. Por tanto será necesaria una conexión a Internet lo suficientemente ancha para el envío y recepción de estos datos.

Software del Ordenador Personal

• **Sistema Operativo:** la aplicación se ejecuta sobre la máquina virtual de Java. Por tanto puede ejecutarse sobre cualquier Sistema Operativo que sea compatible con esta plataforma.
• **Máquina virtual de Java:** Será necesario para ejecutar la aplicación descargándose desde su página Web o mediante el CD que se adjunta a esta memoria. Se encontrará más información en el Apéndice A.

3.3 Análisis del Sistema

Tras conocer el propósito del proyecto software, sus propiedades y las restricciones a las que debe someterse, es el momento de analizar el sistema y crear un modelo del mismo que sea correcto, completo, consistente, claro y verificable. Con este fin, se definirán los casos de uso según los requerimientos previamente obtenidos y, acto seguido, se describirán los principales escenarios y flujos de eventos de dichos casos de uso [42].

3.3.1 Modelo de Casos de Uso

El primer paso para realizar el modelo de casos de uso de nuestro sistema es la obtención de los diversos diagramas de casos de uso del mismo [45]. El primero de ellos es un diagrama frontera, es decir, un diagrama que describe completamente la funcionalidad de un sistema (figura 3.1).

Los casos de uso reflejados en un diagrama frontera pueden ser lo suficientemente precisos o, por el contrario, necesitar ser explicados en mayor detalle. A la hora de detallar un caso de uso se pueden emplear dos tipos de relaciones.

<<extend>>: Se trata de una relación cuyo sentido es hacia el caso de uso a detallar, que representa comportamientos opcionales o excepcionales del caso de uso.

<<include>>: Es una relación cuyo sentido es contrario al de la relación <<extend>>, que representa un comportamiento común del caso de uso, que siempre debe llevarse a cabo.
En el caso de nuestro sistema nos encontramos ante varios casos de uso que deben ser descritos con mayor profundidad.

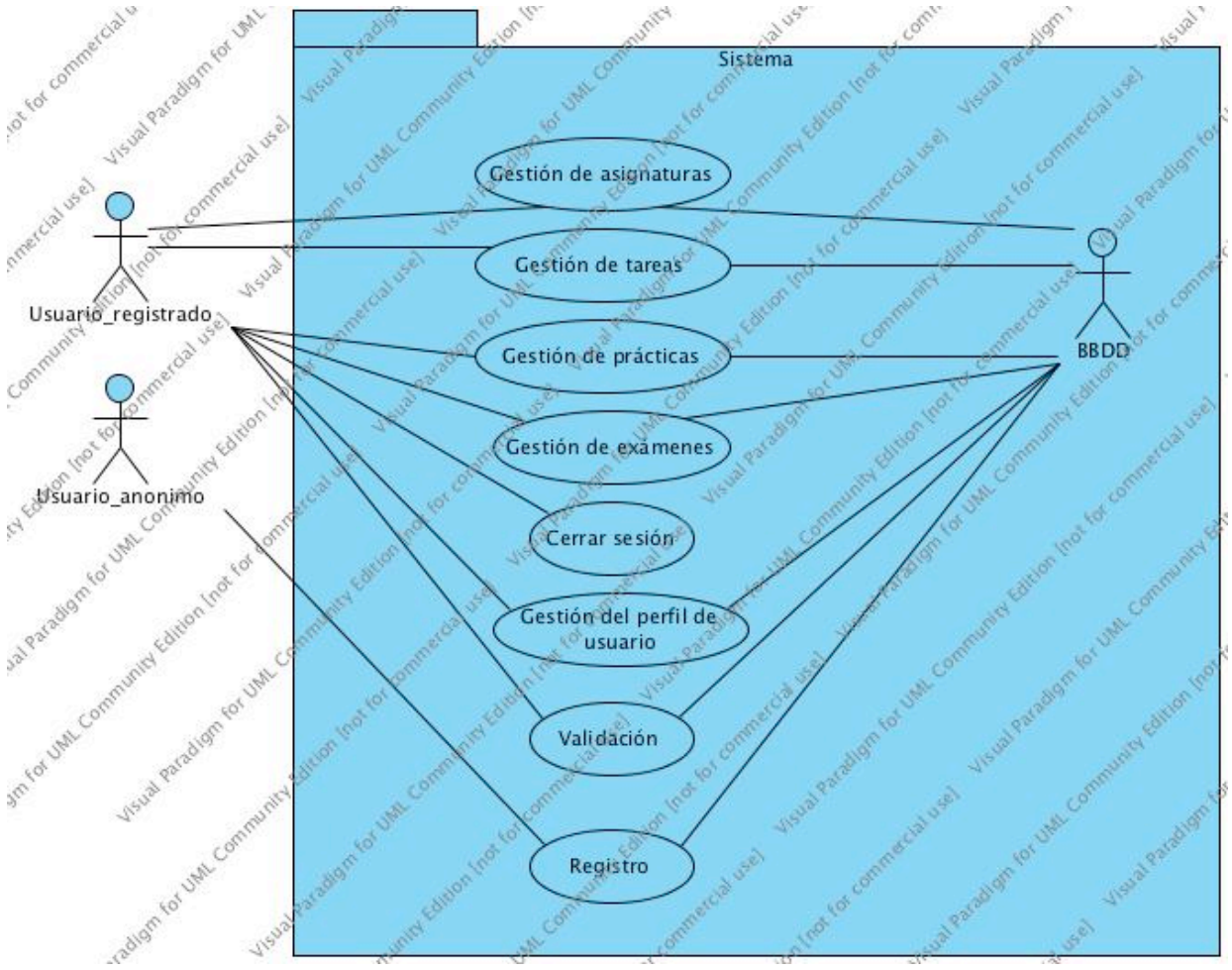

Figura 3.1: Diagrama Frontera de la aplicación.

A continuación, describimos detalladamente cada uno de los casos de uso mostrados en las figuras anteriores, y exponemos los diagramas que se identifican con aquellos casos de uso que necesitan un mayor nivel de detalle en su descripción.

Caso de Uso 1: Registro

Actores participantes: Usuario_anónimo, BBDD.

Condiciones de entrada: Que un usuario desee darse de alta en el sistema, no esté ya dado de alta en el sistema y exista una conexión a Internet.

Flujo de eventos:

1. El usuario inicia la aplicación.
2. El sistema muestra una pestaña para el registro de usuarios.
3. El usuario accede a la página de registro.
4. El sistema muestra un formulario de entrada.
5. El usuario introduce una dirección de correo electrónico, un nombre de usuario y una contraseña.
6. El sistema comprueba que los datos introducidos son válidos y los envía a la nube para comprobar su validez (E-1)(E-2).
7. Los datos del nuevo usuario quedan almacenados en la nube y en la BBDD.
8. Se muestra la pantalla principal de la aplicación para el nuevo usuario registrado.

Condiciones de salida: El nuevo usuario ha sido registrado en la BBDD del sistema y accede a la aplicación.

Excepciones

E-1: La dirección de correo electrónica ya está asignada a otra cuenta. El sistema informa al usuario de dicha situación. El usuario puede introducir otro identificador de usuario diferente o salir del caso de uso.

E-2: La dirección de correo electrónico no es válida (no dispone de los símbolos necesarios para autentificar un correo: "@" y "."). El sistema informa al usuario de dicha situación. El usuario puede introducir otra dirección de correo-e diferente o salir del caso de uso.

Caso de Uso 2: Validación de usuario

Actores participantes: Usuario_registrado, BBDD.

Condiciones de entrada: Que un usuario registrado pretenda acceder a la aplicación.

Flujo de eventos:

1. El usuario inicia la aplicación.
2. El sistema muestra una página con un formulario de inicio de sesión.
3. El usuario introduce su identificador y su contraseña.

4.a. Si hay conexión a Internet, el sistema comprueba que el identificador y la contraseña son válidos en la nube (E-1) y realiza una sincronización de los datos entre la nube y la BBDD.

4.b. Si no hay conexión a Internet, el sistema comprueba que el identificador y la contraseña son válidos en la BBDD (E-1).

5. El usuario entra al sistema con los privilegios de usuario registrado.

Condiciones de salida: El usuario accede al sistema con privilegios de usuario registrado tras validar su nombre de usuario y su contraseña.

Excepciones

E-1: El nombre de usuario no se conoce o la contraseña no es la indicada. El sistema muestra un aviso para informar al usuario de esta situación y puede introducir los datos de nuevo si lo desea o salir del caso de uso. También podrá volver al caso de uso "Registro" para adquirir otra cuenta de usuario.

Caso de Uso 3: Gestión de asignaturas.

Actores participantes: Usuario_registrado, BBDD.

Condiciones de entrada: El usuario ha entrado en el sistema y se ha validado como usuario registrado.

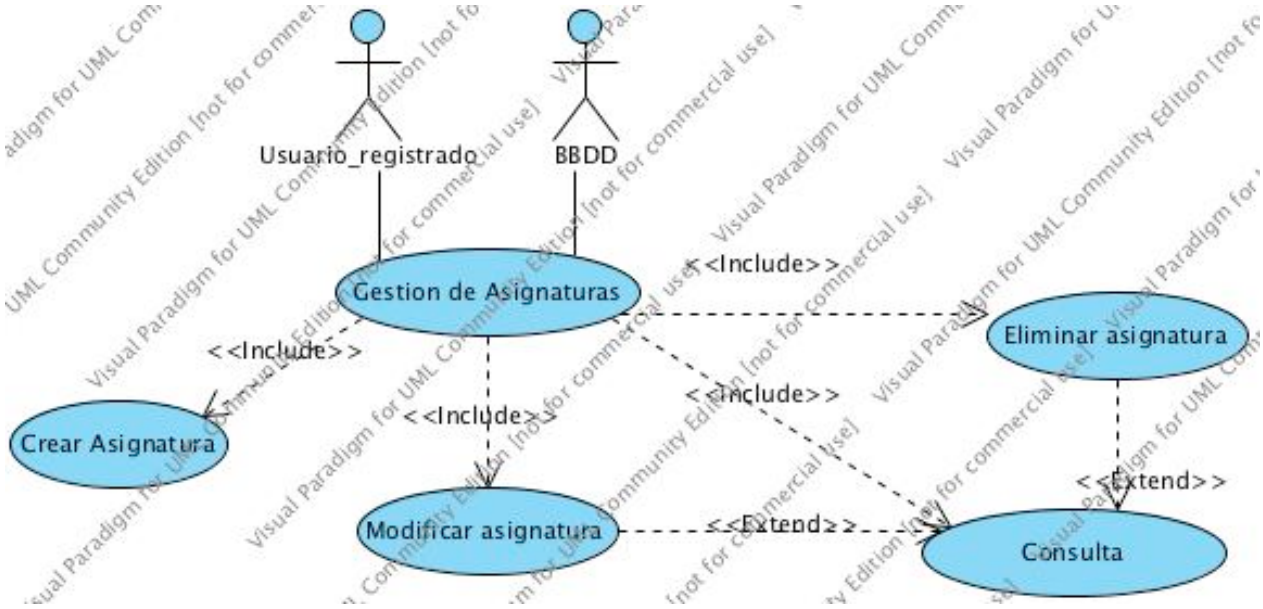

Figura 3.2: Caso de Uso "Gestión de asignaturas".

Flujo de eventos:

1. El usuario accede a la pestaña "Asignaturas".
2. El usuario desea crear una asignatura, se realiza S-1.
3. El usuario quiere modificar una asignatura ya existente, se realiza S-2.
4. El usuario desea eliminar una asignatura, se realiza S-3.
5. El usuario quiere consultar una de las asignaturas existentes, se realiza S-4.

Subflujo de eventos:

-S1: Crear asignatura

Flujo de eventos:

1.1. El usuario selecciona la opción "Crear Asignatura".
1.2. El sistema muestra una pantalla para rellenar los datos de la asignatura: nombre, profesor y aula.
1.3. El sistema registra esta información en la base de datos y a continuación se muestra la pantalla principal con la asignatura ya creada.

-S2: Modificar asignatura

2.1. El usuario hace click sobre el botón "Editar" en la asignatura que desee modificar.
2.2. El sistema muestra ahora el panel de la asignatura seleccionada como modificable, de tal manera que pueda modificar todos los campos que sean necesarios.
2.3. Una vez realizados los cambios deseados, el usuario debe pulsar el botón "Ok".
2.4. El sistema se comunica con la base de datos y añade la información de la asignatura a la cuenta del usuario (E-1).

-S3: Eliminar asignatura

3.1. El usuario hace click en el botón "Eliminar" de la asignatura que desee eliminar.
3.2. El sistema pregunta al usuario si está seguro que desea llevar a cabo esta acción.
3.3.a. Si el usuario pulsa que sí, la asignatura se elimina de la base de datos.
3.3.b. Si el usuario pulsa que no, se cancela la opción y no se modifica ningún dato (E-1).

-S4: Consulta

4.1. El usuario hace click en el cuadro de búsqueda desde la pestaña de "Asignaturas" e introduce el texto del nombre de una asignatura que desee buscar.

4.2. Una vez introducida la cadena de texto a buscar, pulsa el botón "Buscar" que se encuentra a la derecha del cuadro de texto.

4.3. El sistema mostrará las asignaturas que cumplan las condiciones de la búsqueda en caso de encontrarse, y nada si ninguna asignatura cumple estas condiciones.

Condiciones de salida: El nuevo usuario ha gestionado sus asignaturas mediante la creación, modificación y eliminación de sus asignaturas.

Excepciones

E-1: No se puede acceder a la BBDD, con lo que se muestra al usuario un mensaje de que no se han podido añadir, modificar o eliminar los datos de su cuenta.

Caso de Uso 4: Gestión de tareas.

Actores participantes: Usuario_registrado, BBDD.

Condiciones de entrada: El usuario ha entrado en el sistema y se ha validado como usuario registrado.

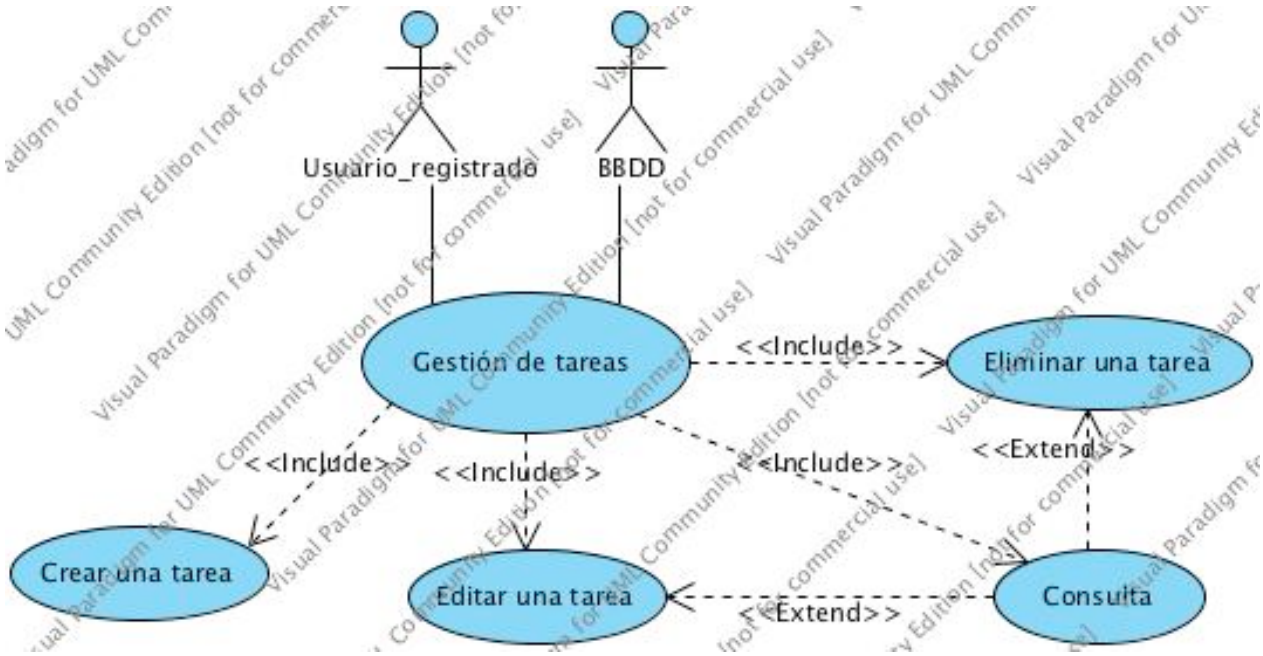

Figura 3.3: Caso de Uso "Gestión de tareas".

Flujo de eventos:

1. El usuario hace click sobre la pestaña "Tareas".
2. El usuario desea crear una tarea, se realiza S-1.
3. El usuario quiere modificar una tarea ya existente, se realiza S-2.
4. El usuario desea eliminar una tarea, se realiza S-3.
5. El usuario quiere consultar una de las tarea existentes, se realiza S-4.

Subflujo de eventos:

-S1: Crear una tarea

1.1. El sistema muestra la lista de tareas creadas hasta el momento y las opciones disponibles de éstas.
1.2. El usuario pulsa sobre "Añadir tarea".
1.3. El sistema muestra un panel de nueva tarea, donde el usuario podrá rellenar los datos de ésta (Nombre, Fecha de finalización y Descripción).
1.4. El usuario pulsa sobre "Aceptar".
1.5. El sistema se comunica con la base de datos y añade la información de la tarea a la cuenta del usuario (E-1).

-S.2: Editar una tarea

1.1. El usuario hace click sobre el botón "Editar" en la tarea que desee modificar.
1.2. El sistema muestra ahora el panel de la tarea seleccionada como modificable, de tal manera que pueda modificar todos los campos que sean necesarios.
1.3. Una vez realizados los cambios deseados, el usuario debe pulsar el botón "Ok".
1.4. El sistema se comunica con la base de datos y añade la información de la tarea a la cuenta del usuario (E-1).

-S.3: Eliminar una tarea

1.1. El usuario clickea el botón "Eliminar" de la tarea que desee eliminar.
1.2. El sistema pregunta al usuario si está seguro que desea llevar a cabo esta acción.

1.3.a. Si el usuario pulsa que sí, la tarea se elimina de la base de datos.

1.3.b. Si el usuario pulsa que no, se cancela la opción y no se modifica ningún dato (E-1).

-S4: Consulta

4.1. El usuario hace click en el cuadro de búsqueda desde la pestaña de "Tareas" e introduce el texto del nombre de una tarea que desee buscar.
4.2. Una vez introducida la cadena de texto a buscar, pulsa el botón "Buscar" que se encuentra a la derecha del cuadro de texto.
4.3. El sistema mostrará las tareas que cumplan las condiciones de la búsqueda en caso de encontrarse, y nada si ninguna tarea cumple estas condiciones.

Condiciones de salida: El nuevo usuario ha gestionado sus tareas mediante su creación, modificación y eliminación.

Excepciones

E-1: No se puede acceder a la BBDD, con lo que se muestra al usuario un mensaje de que no se han podido añadir, modificar o eliminar los datos de su cuenta.

Caso de Uso 5: Gestión de prácticas.

Actores participantes: Usuario_registrado, BBDD.

Condiciones de entrada: El usuario ha entrado en el sistema y se ha validado como usuario registrado.

Flujo de eventos:

1. El usuario hace click sobre la pestaña "Prácticas".
2. El usuario desea crear una práctica, se realiza S-1.
3. El usuario quiere modificar una práctica ya existente, se realiza S-2.
4. El usuario desea eliminar una práctica, se realiza S-3.

5. El usuario quiere consultar una de las prácticas existentes, se realiza S-4.

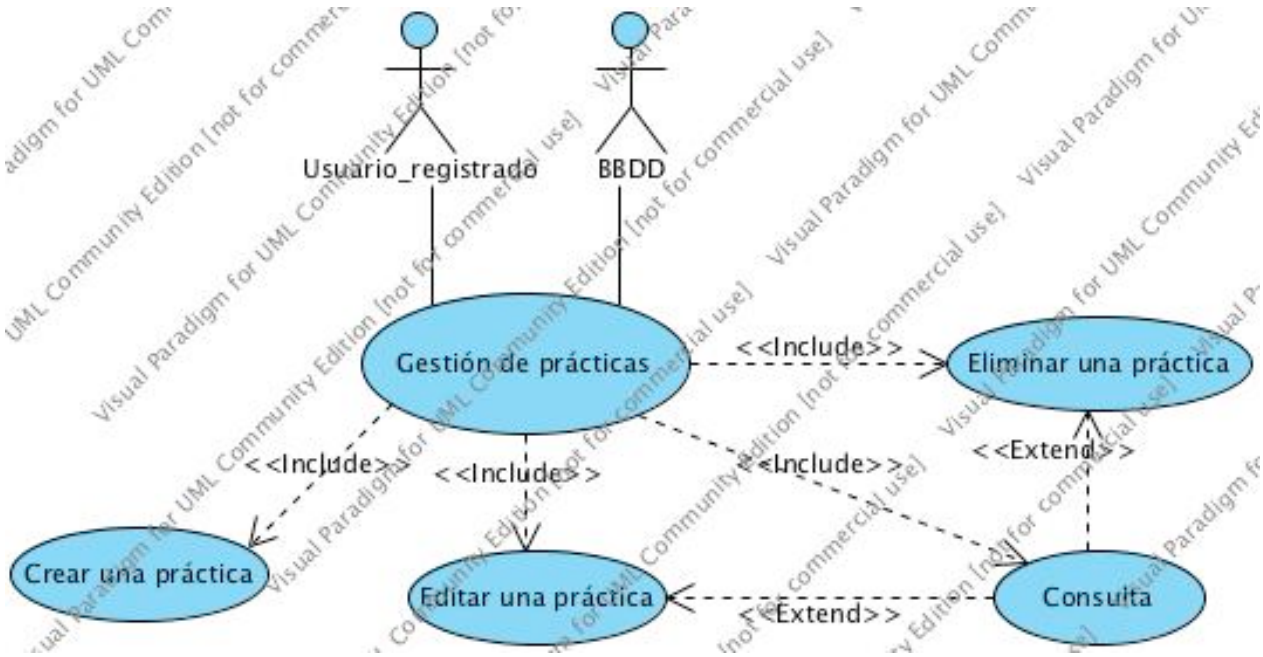

Figura 3.4: Caso de Uso "Gestión de prácticas".

Subflujo de eventos:

-S1: Crear una práctica

1.1. El sistema muestra la lista de prácticas creadas hasta el momento y las opciones disponibles de éstas.
1.2. El usuario pulsa sobre "Añadir práctica".
1.3. El sistema muestra un panel de nueva práctica, donde el usuario podrá rellenar los datos de ésta (Nombre y Descripción).
1.4. El usuario pulsa sobre "Aceptar".
1.5. El sistema se comunica con la base de datos y añade la información de la práctica a la cuenta del usuario (E-1).

-S2: Editar una práctica

2.1. El usuario hace click sobre el botón "Editar" en la práctica que desee modificar.
2.2. El sistema muestra ahora el panel de la práctica seleccionada como modificable, de tal manera que pueda modificar todos los campos que sean necesarios.

2.3. Una vez realizados los cambios deseados, el usuario debe pulsar el botón "Ok".

2.4. El sistema se comunica con la base de datos y añade la información de la práctica a la cuenta del usuario (E-1).

-S3: Eliminar una práctica

3.1. El usuario clickea el botón "Eliminar" de la práctica que desee eliminar.

3.2. El sistema pregunta al usuario si está seguro que desea llevar a cabo esta acción.

3.3.a. Si el usuario pulsa que sí, la práctica se elimina de la base de datos.

3.3.b. Si el usuario pulsa que no, se cancela la opción y no se modifica ningún dato (E-1).

-S4: Consulta

4.1. El usuario hace click en el cuadro de búsqueda desde la pestaña de "Prácticas" e introduce el texto del nombre de una tarea que desee buscar.

4.2. Una vez introducida la cadena de texto a buscar, pulsa el botón "Buscar" que se encuentra a la derecha del cuadro de texto.

4.3. El sistema mostrará las prácticas que cumplan las condiciones de la búsqueda en caso de encontrarse, y nada si ninguna práctica cumple estas condiciones.

Condiciones de salida: El nuevo usuario ha gestionado sus prácticas mediante su creación, modificación y eliminación.

Excepciones

E-1: No se puede acceder a la BBDD, con lo que se muestra al usuario un mensaje de que no se han podido añadir, modificar o eliminar los datos de su cuenta.

Caso de Uso 6: Gestión de exámenes.

Actores participantes: Usuario_registrado, BBDD.

Condiciones de entrada: El usuario ha entrado en el sistema y se ha validado como usuario

registrado.

Flujo de eventos:

1. El usuario hace click sobre la pestaña "Examen".
2. El usuario desea crear un examen, se realiza S-1.
3. El usuario quiere modificar un examen ya existente, se realiza S-2.
4. El usuario desea eliminar un examen, se realiza S-3.
5. El usuario quiere consultar uno de los exámenes existentes, se realiza S-4.

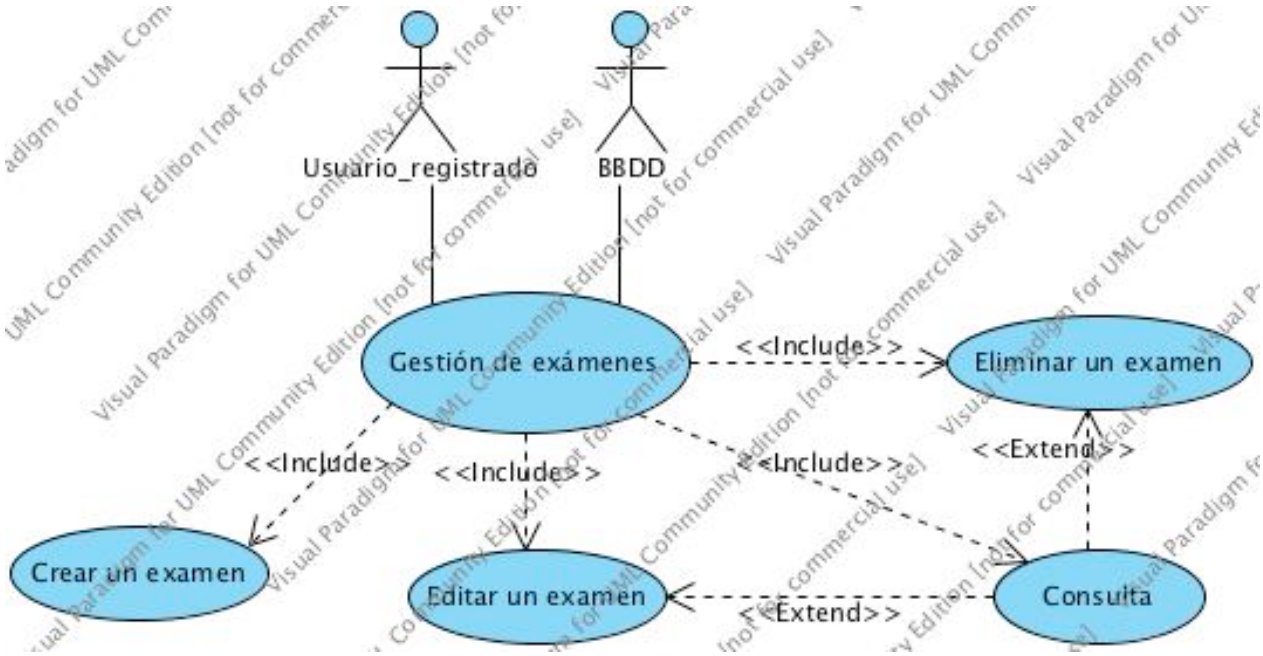

Figura 3.5: Caso de Uso "Gestión de exámenes".

Subflujo de eventos:

-S1: Crear un examen

1.1. El sistema muestra la lista de exámenes creadas hasta el momento y las opciones disponibles de éstas.
1.2. El usuario pulsa sobre "Añadir examen".
1.3. El sistema muestra un panel de nuevo examen, donde el usuario podrá rellenar los datos de éste (Nombre y fecha).
1.4. El usuario pulsa sobre "Aceptar".

1.5. El sistema se comunica con la base de datos y añade la información de la práctica a la cuenta del usuario (E-1).

-S2: Editar un examen

2.1. El usuario hace click sobre el botón "Editar" en el examen que desee modificar.
2.2. El sistema muestra ahora el panel del examen seleccionado como modificable, de tal manera que pueda modificar todos los campos que sean necesarios.
2.3. Una vez realizados los cambios deseados, el usuario debe pulsar el botón "Ok".
2.4. El sistema se comunica con la base de datos y añade la información del examen a la cuenta del usuario (E-1).

-S3: Eliminar un examen

3.1. El usuario clickea el botón "Eliminar" del examen que desee eliminar.
3.2. El sistema pregunta al usuario si está seguro que desea llevar a cabo esta acción.
3.3.a. Si el usuario pulsa que sí, el examen se elimina de la base de datos.
3.3.b. Si el usuario pulsa que no, se cancela la opción y no se modifica ningún dato (E-1).

-S4: Consulta

4.1. El usuario hace click en el cuadro de búsqueda desde la pestaña de "Exámenes" e introduce el texto del nombre de una tarea que desee buscar.
4.2. Una vez introducida la cadena de texto a buscar, pulsa el botón "Buscar" que se encuentra a la derecha del cuadro de texto.
4.3. El sistema mostrará los exámenes que cumplan las condiciones de la búsqueda en caso de encontrarse, y nada si ningún examen cumple estas condiciones.

Condiciones de salida: El nuevo usuario ha gestionado sus exámenes mediante su creación, modificación y eliminación.

Excepciones

E-1: No se puede acceder a la BBDD, con lo que se muestra al usuario un mensaje de que no se han podido añadir, modificar o eliminar los datos de su cuenta.

Caso de Uso 7: Gestión del perfil de usuario.

Actores participantes: Usuario_registrado, BBDD.

Condiciones de entrada: El usuario ha entrado en el sistema y se ha validado como usuario registrado.

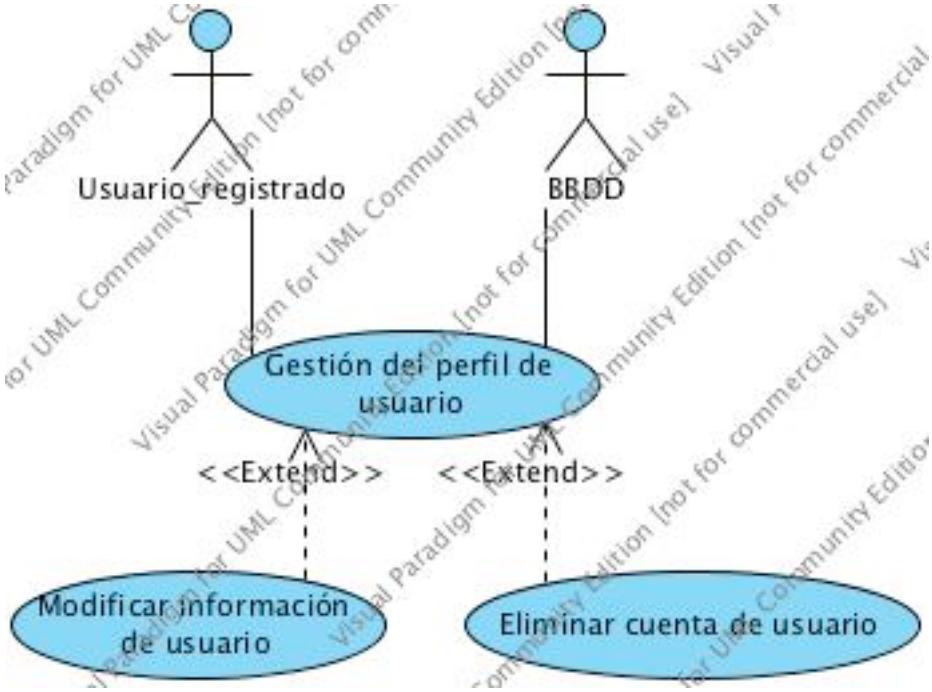

Figura 3.6: Caso de uso "Gestión del perfil de usuario".

Flujo de eventos:

1. El usuario selecciona la pestaña "Configuración del menú principal".
2. El usuario desea editar su información de usuario, se realiza S-1.
3. El usuario quiere eliminar su cuenta, se realiza S-2.

Subflujo de eventos:

-S1: Editar información de usuario

1. El sistema muestra los datos personales del usuario.
2. El usuario, si lo desea, modifica su nick o el nombre de la carrera que estudia.
3 El sistema guarda los cambios efectuados (E-1).

-S2: Eliminar cuenta de usuario

1. El usuario pulsa la opción "Eliminar cuenta".
2. El sistema pregunta al usuario nuevamente si realmente desea borrar todos sus datos, ya que serán borrados para siempre.

3.a. Si el usuario responde que sí, el sistema borra de la BBDD los datos del usuario (E-1).

3.b. Si el usuario responde que no, se sale del caso de uso sin ninguna modificación.

Excepciones

E-1: Ha habido un error en la comunicación entre el sistema y la base de datos. El sistema informa al usuario de dicha situación. El caso de uso se reanuda desde el punto en que se quedó.

Condiciones de salida: El usuario edita la información de su cuenta, o bien elimina ésta y sus datos de la aplicación.

Caso de Uso 8: Cerrar sesión.

Actores participantes: Usuario_registrado.

Condiciones de entrada: Que el usuario esté registrado y validado en el sistema.

Flujo de eventos:

1. El usuario selecciona la opción "Cerrar sesión" para cerrar su cuenta.

2.a Si existe conexión a Internet, el sistema sincroniza los datos del equipo local con los de la nube.

3. El sistema se cierra.

Condiciones de salida: El usuario cierra su sesión.

3.3.2 Escenarios

Un caso de uso es una representación abstracta de una funcionalidad del sistema a realizar. La representación concreta de un caso de uso se realiza mediante la creación de uno o más escenarios que muestren todas las posibles interacciones entre el sistema y los usuarios.

Los escenarios son historias imaginarias que nos sirven de ejemplo para describir posibles interacciones de un caso de uso. Son de gran utilidad porque permiten a los diseñadores anticiparse a los problemas. A pesar de ser historias ficticias, deben desarrollarse con el mayor nivel de detalle posible. Con ello, es mucho más sencillo para los diseñadores discutir sobre la interfaz ya que a las personas les cuesta más trabajo discutir las posteriores decisiones de diseño sobre una situación abstracta.

Esta forma de proceder, además, fuerza a los diseñadores a tener en cuenta el rango de usuarios que va a usar el sistema y el rango de actividades para las que lo van a usar. Los escenarios permiten hacer diferentes combinaciones de usuarios y actividades de forma que se tengan en cuenta todas las posibilidades que pueden darse en la aplicación.

Un escenario está formado por los siguientes elementos:

- Un nombre único y unívoco
- Una descripción
- Los actores participantes.
- El flujo de eventos.

Los escenarios que vamos a mostrar de nuestro proyecto serán los necesarios para representar los casos de uso más importantes: registro, validación y gestión de asignaturas, tareas, prácticas y exámenes.

Escenario 1: Registro.

Nombre: Registrar usuario.

Descripción: El usuario Juan entra en el sistema y decide registrarse para organizar sus estudios en la universidad.

Actores participantes: Usuario anónimo Juan y Base de Datos.

Flujo de eventos:

1. Juan entra como invitado en el sistema y pulsa sobre la pestaña "Registrar".

2. El sistema muestra un formulario con todos los datos requeridos para el registro.

3. El usuario rellena los siguientes campos:
   a) Usuario: Juan.
   b) E-mail: juan@hotmail.com .
   c) Contraseña: Juan introduce su contraseña.

4. Una vez rellenado el formulario, Juan pulsa sobre el botón 'Registrar' en la parte inferior de la ventana.

5. El sistema almacena los datos de Juan en la BBDD, muestra un mensaje informando de que el registro ha sido llevado a cabo con éxito, y accede a la aplicación donde Juan accede ya como usuario registrado.

Escenario 2: Validación.

Nombre: Validar un usuario.

Descripción: El usuario Juan ya está registrado en el sistema y entra a su cuenta para acceder a su información.

Actores participantes: Usuario registrado Juan y Base de Datos.

Flujo de eventos:

1. Juan entra en el sistema.

2. El sistema muestra el formulario para iniciar sesión como usuario registrado.

3. El usuario introduce sus datos para acceder a su cuenta:
   a) Usuario: juan@hotmail.com .
   b) Contraseña: La contraseña que anteriormente Juan registró.

4. Juan pulsa el botón 'Acceder'.

5. El sistema comprueba los datos introducidos por Juan en la BBDD, valida correctamente al usuario, sincroniza sus datos con los de la nube y entra en la aplicación como usuario registrado.

6. El sistema muestra la pantalla principal de la aplicación, ofreciendo a Juan la posibilidad de gestionar sus asignaturas, tareas, prácticas y exámenes, así como de acceder a su configuración.

Escenario 3: Gestión de asignaturas.

Nombre: Gestión de asignaturas.

Descripción: El usuario Juan crea una asignatura y añade una práctica vinculada a ella.

Actores participantes: Usuario registrado Juan y Base de Datos.

Flujo de eventos:

1. Juan entra en el sistema y se valida de manera correcta.

2. El sistema muestra la pantalla principal.

3. Juan pulsa el botón "Añadir Asignatura", y rellena los datos correspondientes:
   a) Nombre: Sistemas informáticos.
   b) Profesor: Luis Martínez.
   c) Aula: A-3, Lab 110.

4. Juan pulsa el botón 'Ok'.

5. El sistema añade la asignatura que Juan acaba de crear a la base de datos.

6. A continuación, Juan pulsa sobre el botón "Prácticas" dentro del cuadro correspondiente a la asignatura "Sistemas Informáticos".

7. Juan pulsa el botón "Añadir Práctica", y rellena los datos correspondientes:
   a) Nombre: Consulta Web del Conocimiento.
   b) Fecha: 12/09/2013.
   c) Descripción: Asistir y realizar las actividades de clase.

8. Juan pulsa el botón 'Ok'.

9. El sistema añade la práctica que Juan acaba de crear a la base de datos y la muestra por pantalla.

Escenario 4: Gestión de tareas.

Nombre: Gestión de tareas.

Descripción: El usuario Juan crea una tarea, la modifica y posteriormente la elimina.

Actores participantes: Usuario registrado Juan y Base de Datos.

Flujo de eventos:

1. Juan entra en el sistema y se valida de manera correcta.

2. El sistema muestra la pantalla principal.

3. Juan pulsa sobre el botón "Tareas".

4. El sistema muestra la pantalla de Tareas.

5. Juan pulsa el botón "Añadir Tarea", y rellena los datos correspondientes:
   a) Nombre: Revisar bibliografía.
   b) Descripción: Revisar la bibliografía para la memoria.
   c) Fecha: 5/10/2013.

6. Juan pulsa el botón 'Ok'.

7. El sistema añade la tarea que Juan acaba de crear a la base de datos y la muestra entre las tareas que Juan tiene por hacer.

8. A continuación, Juan decide editar la fecha y pulsa sobre el icono del lápiz (botón edición) de la tarea correspondiente.

9. El sistema vuelve a mostrar la tarea modificable para que Juan pueda cambiar los siguientes campos:
   a) Fecha: 6/10/2013.

10. Juan pulsa el botón 'Ok'.

11. El sistema muestra las tareas con las modificaciones que Juan acaba de realizar.

12. Por último, Juan decide no realizar su tarea y eliminarla de la aplicación. Para ello pulsa sobre el icono de la cruz (botón de borrado) sobre la tarea a eliminar.

13. El sistema pregunta a Juan si desea realmente eliminar esta tarea.

14. Juan está seguro y pulsa que sí.

15. El sistema elimina la tarea que Juan ha decidido eliminar y muestra las tareas restantes por pantalla.

Escenario 5: Gestión de prácticas.

Nombre: Gestión de prácticas.

Descripción: El usuario Juan crea una práctica, la modifica y posteriormente la elimina.

Actores participantes: Usuario registrado Juan y Base de Datos.

Flujo de eventos:

1. Juan entra en el sistema y se valida de manera correcta.

2. El sistema muestra la pantalla principal.

3. Juan pulsa sobre el botón "Prácticas".

4. El sistema muestra la pantalla de Prácticas.

5. Juan pulsa el botón "Añadir Prácticas", y rellena los datos correspondientes:
   a) Nombre: Prácticas de Coche.
   b) Descripción: Prácticas de coche sobre ciudad.
   c) Fecha: 7/10/2013.

6. Juan pulsa el botón 'Ok'.

7. El sistema añade la práctica que Juan acaba de crear a la base de datos y la muestra entre las prácticas que Juan tiene por hacer.

8. Posteriormente, Juan se da cuenta de que la fecha de su práctica se ha cambiado y será finalmente el 8 de octubre. Para ello, pulsa sobre el icono del lápiz (botón de edición) para modificar su práctica.

9. El sistema vuelve a mostrar la práctica modificable para que Juan pueda cambiar los siguientes campos:
   a) Fecha: 8/10/2013.

10. Juan pulsa el botón 'Ok'.

11. El sistema muestra las prácticas con las modificaciones que Juan acaba de realizar.

12. Finalmente, Juan decide no asistir a la práctica y borrarla de la aplicación. Para ello pulsa sobre el icono de la cruz (botón de borrado) sobre la práctica a eliminar.

13. El sistema pregunta a Juan si desea realmente eliminar esta práctica.

14. Juan está seguro y pulsa que sí.

15. El sistema elimina la práctica que Juan ha decidido eliminar y muestra las prácticas restantes por pantalla.

Escenario 6: Gestión de exámenes.

Nombre: Gestión de exámenes.

Descripción: El usuario Juan crea un examen en la aplicación.

Actores participantes: Usuario registrado Juan y Base de Datos.

Flujo de eventos:

1. Juan entra en el sistema y se valida de manera correcta.
2. El sistema muestra la pantalla principal.
3. Juan pulsa sobre el botón "Exámenes".
4. El sistema muestra la pantalla de Exámenes.
5. Juan pulsa el botón "Añadir Examen", y rellena los datos correspondientes:
   a) Nombre: B2 - Inglés.
   b) Fecha: 10/10/2013.
   c) Hora: 9:00.
6. Juan pulsa el botón 'Ok'.
7. El sistema añade el examen que Juan acaba de crear a la base de datos y lo muestra por pantalla.

Escenario 7: Gestión del perfil de usuario 1.

Nombre: Edición del perfil de usuario.

Descripción: El usuario Juan decide cambiar el nombre de sus estudios.

Actores participantes: Usuario registrado Juan y Base de Datos.

Flujo de eventos:

1. Juan entra en el sistema y se valida de manera correcta.

2. El sistema muestra la pantalla principal.

3. Juan pulsa el botón "Configuración".

5. El sistema muestra la pantalla "Configuración".

6. A continuación, Juan escribe el nuevo nombre de sus estudios:
    a) Estudios en curso: "Grado en Informática".

7. Juan pulsa el botón "Cambiar estudios", y responde afirmativamente a la pregunta del sistema de si desea realizar esta acción.

8. El sistema añade el nuevo nombre a sus estudios y lo muestra en el panel superior de la aplicación.

Escenario 8: Gestión del perfil de usuario 2.

Nombre: Eliminar cuenta.

Descripción: El usuario Juan eliminar su cuenta y borra con ello sus datos de la aplicación.

Actores participantes: Usuario registrado Juan y Base de Datos.

Flujo de eventos:

1. Juan entra en el sistema y se valida de manera correcta.
2. El sistema muestra la pantalla principal.
3. Juan pulsa el botón “Configuración”.
4. El sistema muestra el panel Configuración con sus diferentes opciones disponibles.
5. Juan pulsa el botón ‘Eliminar cuenta’.
6. El sistema pregunta a Juan si está seguro que desea eliminar su cuenta.
7. Juan pulsa sobre el botón “Sí”.
8. El sistema borra al usuario Juan de la base de datos y elimina todas las asignaturas, prácticas, tareas y exámenes vinculados a su cuenta

Escenario 9: Cerrar sesión.

Nombre: Cerrar sesión.

Descripción: El usuario Juan desea cerrar su sesión.

Actores participantes: Usuario registrado Juan y Base de Datos.

Flujo de eventos:

1. Juan se encuentra dentro del sistema validado de manera correcta.

2. Juan pulsa el botón de "Cerrar sesión", que siempre se encuentra presente en la parte superior derecha de la aplicación.

3. El sistema sincroniza los datos de la aplicación local con los de la nube, cierra la sesión de Juan y muestra la pantalla de acceso a la aplicación.

3.4 Diseño del Sistema

El diseño del sistema es la actividad de la Ingeniería del Software en la que se identifican los objetivos finales del sistema y se plantean las diversas estrategias para alcanzarlos en la actividad de implementación [45].

A su vez, el diseño del sistema puede dividirse en dos partes bien diferenciadas, que son el diseño de la estructura que mantendrán los datos de nuestra aplicación y el diseño que se va a llevar a cabo en la interfaz.

3.4.1 Diseño de clases

Los diagramas de clases forman la estructura principal del sistema y se utilizan durante el análisis de requerimientos para modelar los conceptos del dominio del problema a resolver, durante el diseño de datos para modelar los subsistemas e interfaces y durante la implementación con el fin de modelar las clases. En el caso de este proyecto, se ha optado por realizar dos diagramas de clases para dividir así la parte de la aplicación de escritorio que se ubica en el equipo local del usuario (figura 3.7) con la parte de la aplicación que permanecería en la nube (figura 3.8).

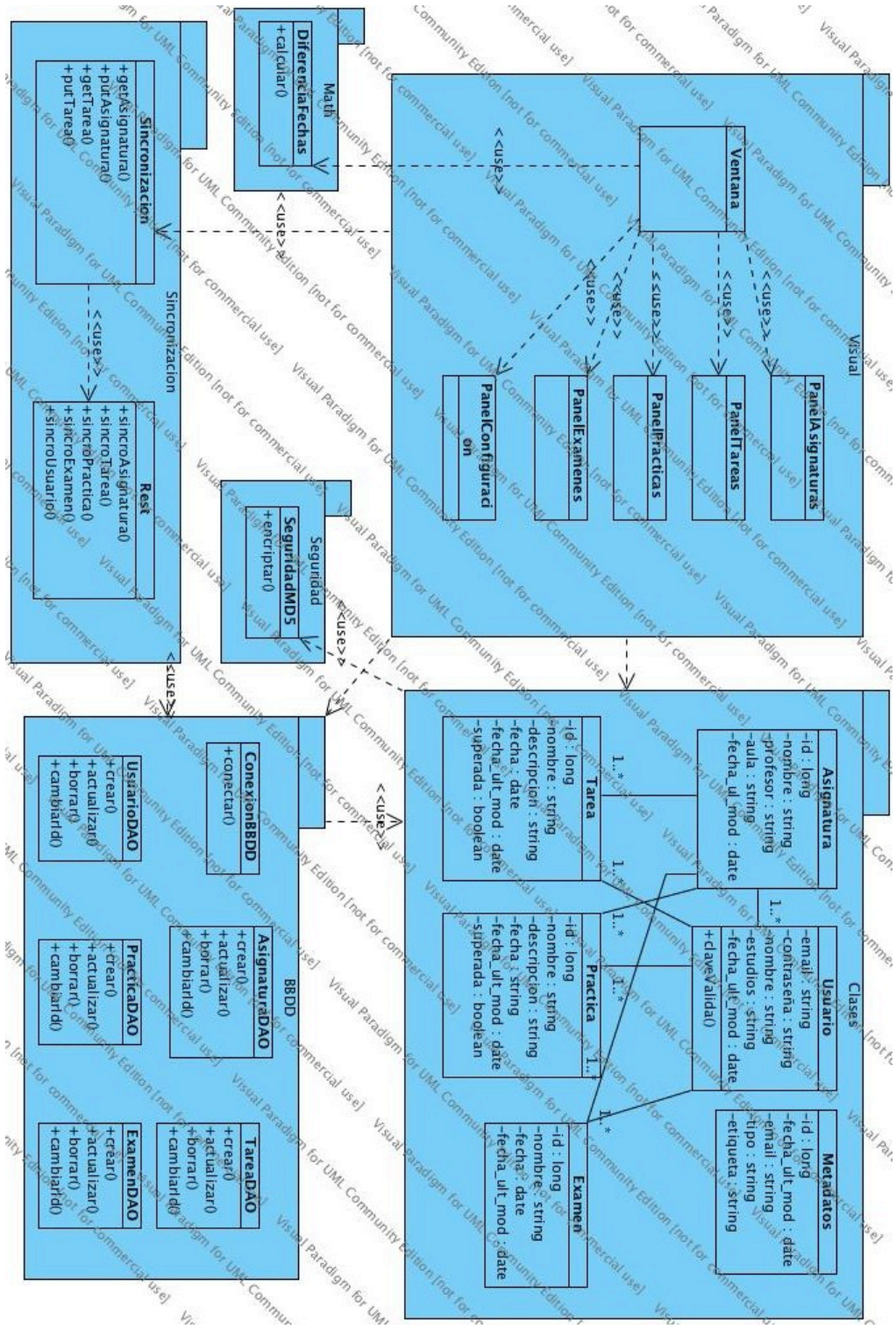

Figura 3.7: Diagrama de Clases de la aplicación de escritorio.

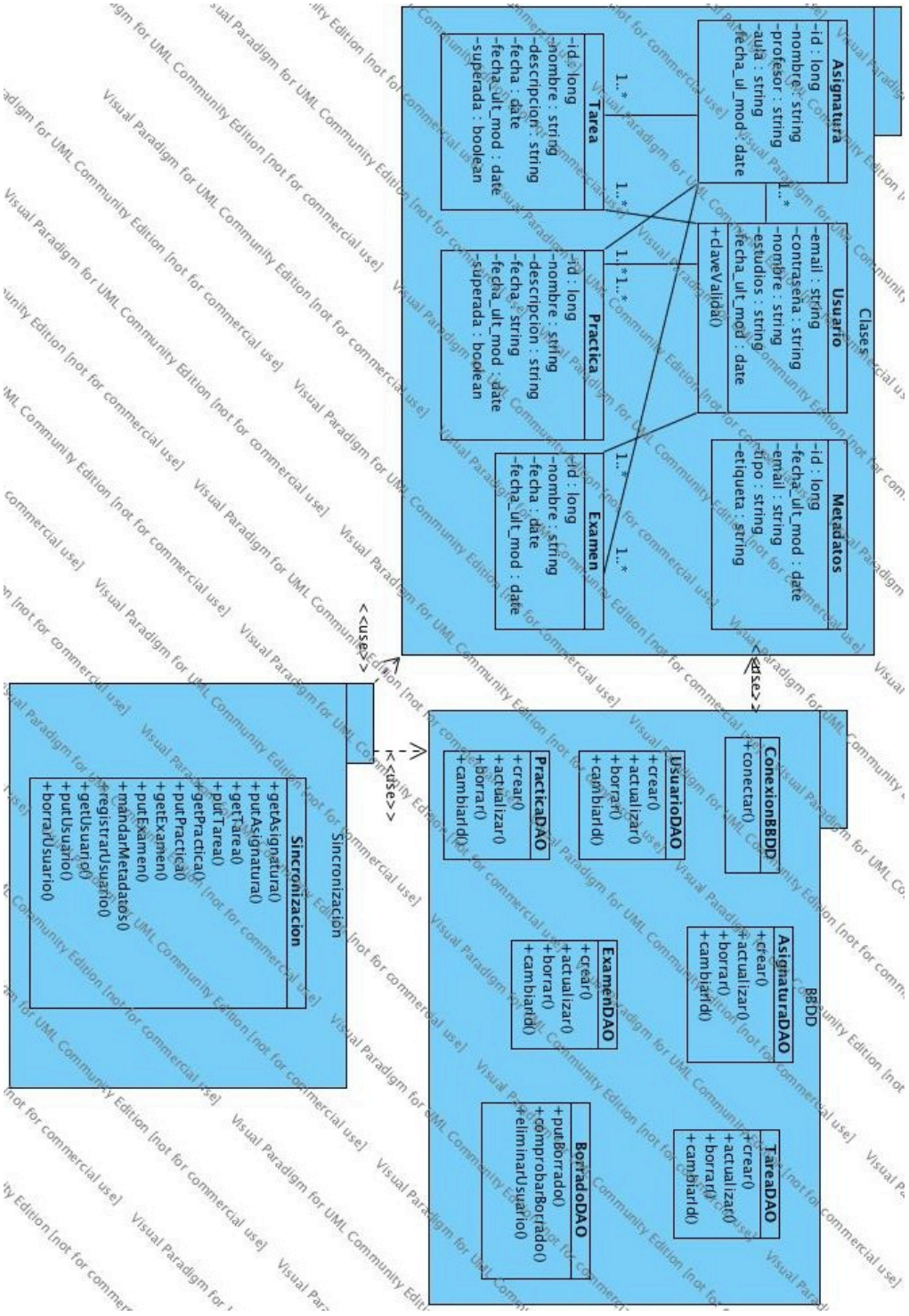

Figura 3.8: Diagrama de Clases de la aplicación en la nube.

3.4.2 Diseño de los Datos

El objetivo en este epígrafe es diseñar la estructura que poseen cada uno de los elementos de información del sistema software que se está desarrollando, es decir, la estructura de los datos sobre los que a continuación vamos a trabajar. Estos elementos son:

- Los usuarios, de los que conocemos su identificador, dirección de e-mail, nombre de usuario, contraseña y fecha de última modificación.
- Las asignaturas de cada usuario, de las que sabemos su identificador, su nombre, el profesor que las imparte, el aula donde tienen lugar y la fecha de última modificación.
- Las tareas de cada usuario, de las que conocemos su identificador, su nombre, una descripción de ellas, la fecha de finalización, la fecha de última modificación y si ya han sido superadas o no.
- Las prácticas de cada usuario, de las que sabemos su identificador, su nombre, una descripción de ellas, la fecha de finalización, la fecha de última modificación y si ya han sido superadas o no.
- Los exámenes que tiene cada usuario, con su identificador, su nombre, la fecha donde tendrá lugar y la fecha de última modificación.

Una vez determinados cuales son los elementos de información del sistema, se debe definir su representación en forma de tablas de una base de datos [46]. Para ello, se realiza en primer lugar un diseño conceptual de la base de datos y, posteriormente, se obtienen las tablas requeridas. Para realizar este diseño conceptual se utilizará el modelo Entidad-Relación.

Modelo Entidad-Relación

El modelo Entidad-Relación (también conocido por sus iniciales: E-R) es una técnica de modelado de datos que utiliza diagramas entidad-relación. No es la única técnica de modelado, pero sí que es con bastante diferencia, la más extendida y la que más se emplea [46].

Un diagrama entidad-relación está compuesto por tres tipos de elementos principales:

- **Entidades:** Son los objetos (cosas, conceptos o personas) sobre los que se tiene información. Se representan mediante rectángulos etiquetados con un nombre en su interior. Una instancia es cualquier ejemplar concreto de una entidad, tal y como se muestra en la figura 3.9.

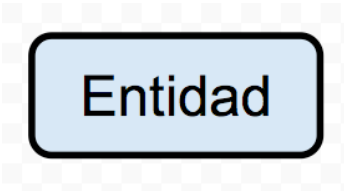

Figura 3.9: Entidad en el modelo E-R.

- **Relaciones:** Son las interdependencias entre una o más entidades. Se representan mediante rombos etiquetados en su interior con un verbo. Si la relación es entre una entidad consigo misma se denomina reflexiva; si es entre dos entidades se denomina binaria, si es entre tres se denominaría ternaria y múltiple si es entre cuatro o más entidades, si bien este último caso ya es menos común. Se muestra un ejemplo en la figura 3.10.

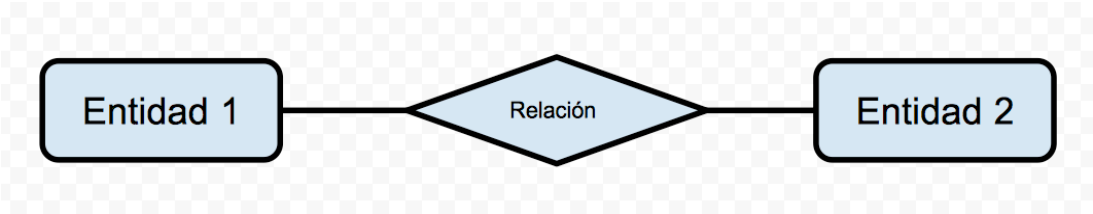

Figura 3.10: Relación entre dos entidades en el modelo E-R.

- **Atributos:** características propias de una entidad o relación. Se representan mediante elipses con un nombre en su interior, como se puede apreciar en la figura 3.11.

Figura 3.11: Atributo de una entidad en el modelo E-R.

Para crear diagramas E-R debemos tener en cuenta otros aspectos, además de los ya considerados, que se exponen brevemente a continuación:

- **Entidades débiles:** son aquellas entidades que no se pueden identificar unívocamente sólo con sus atributos, sino que, además, necesitan estar relacionadas con otra entidad fuerte (explicada anteriormente) para existir. Se representan con dos rectángulos concéntricos de distinto tamaño y con un nombre en el interior del rectángulo más pequeño.

- **Cardinalidad de las relaciones:** Existen tres tipos de cardinalidades de una relación según el número de instancias de cada entidad que involucren.

    - ✓ Uno a uno: Una instancia de la entidad A se relaciona solamente con una instancia de la entidad B. Se representan como 1:1.

    - ✓ Uno a muchos: Cada instancia de la entidad A se relaciona con varias instancias de la entidad B. Se representan como 1:*.

    - ✓ Muchos a muchos: Cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B. Se representan como *:*.

• **Claves:** Cada entidad de un diagrama entidad-relación debe tener obligatoriamente una clave, que puede estar formada por uno o más de sus atributos. Dicha clave es la que distingue inequívocamente una instancia de la citada entidad del resto.

• **Claves foráneas:** Es aquella combinación de atributos de una entidad que es clave en otra entidad distinta de aquella en la que se encuentra.

Una vez conocidos los elementos que forman parte de un diagrama entidad-relación podemos empezar a desarrollar el modelo entidad-relación. Los pasos a seguir son los siguientes:

1. Convertir el enunciado del problema (o, como es nuestro caso, los requerimientos del sistema software) en un Esquema Conceptual del mismo.

2. Convertir este Esquema Conceptual (o EC) en uno más perfeccionado y refinado, conocido como Esquema Conceptual Modificado (ECM).

3. Obtener las tablas de la base de datos a partir del Esquema Conceptual Modificado. Estas serán las mismas tablas que utilizaremos en el sistema.

Normalización en el modelo Entidad-Relación

La normalización en el modelo E-R consiste en imponer a las tablas ciertas restricciones mediante una serie de transformaciones consecutivas. Con este proceso, nos aseguramos de que las tablas contengan los atributos necesarios y suficientes para describir la realidad de la entidad que representan, separando de una manera clara aquellos atributos que puedan contener información cuya relevancia permite la creación de otra nueva tabla, y asegurando así la integridad de los datos.

Para asegurar la normalización, Codd estableció tres formas normales, las cuales hacen que todas las base de datos que las cumplan se consideren normalizadas [46]. Estas formas normales se exponen a continuación:

- **Primera forma Normal (FN1)**: Una tabla está en FN1 si todos los atributos no clave dependen funcionalmente de la clave, o lo que es lo mismo, no existen grupos repetidos para un valor de clave.

- **Segunda forma Normal (FN2):** Una tabla está en FN2 si está en FN1 y además todos los atributos que no pertenecen a la clave dependen funcionalmente de ella de forma completa. De esta definición podemos concluir que una tabla en FN1 y cuya clave está compuesta por un único atributo ya está en FN2.

- **Tercera forma Normal (FN3):** Una tabla está en FN3 si está en FN2 y además no existen atributos no clave que dependan transitivamente de la clave.

Esquema conceptual

Necesitamos convertir nuestros elementos de información en entidades o relaciones. En nuestro caso, Usuarios, Asignaturas, Tareas, Prácticas y Exámenes pasarán a convertirse en entidades de nuestro esquema conceptual. De esta manera, nuestro esquema conceptual quedaría como podemos ver en la Figura 3.12.

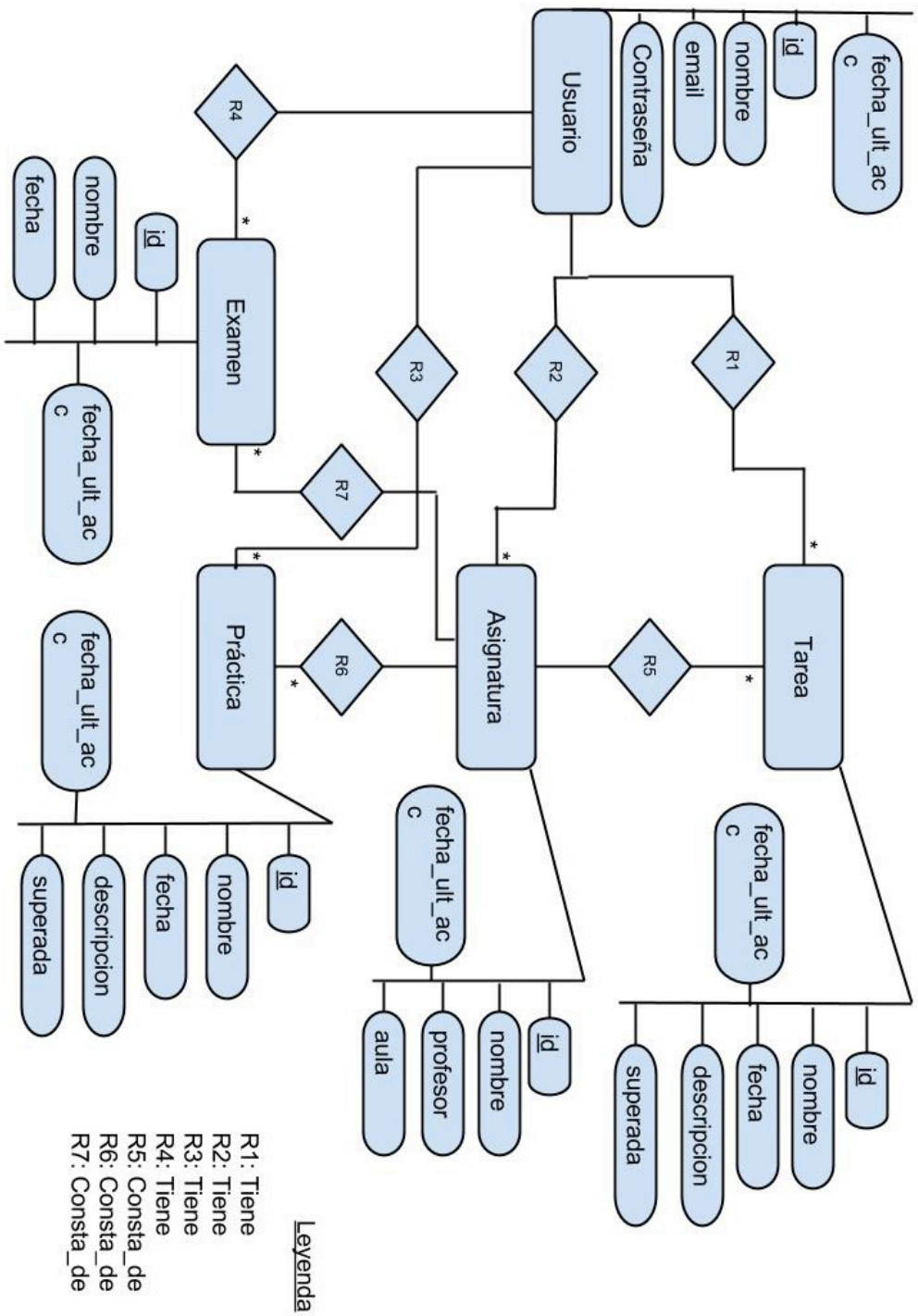

Figura 3.12: Esquema Conceptual E-R.

Esquema Conceptual Modificado

Para la obtención del Esquema Conceptual Modificado a partir del Esquema Conceptual se deben hacer los cambios que enunciamos a continuación:

- Eliminar todas las entidades débiles.
- Eliminar las relaciones de muchos a muchos.
- Eliminar las relaciones con atributos existentes en el Esquema Conceptual.

En el caso de este proyecto, no es necesario eliminar ninguna entidad débil puesto que no las hay. Además, nos encontramos con que ni tenemos relaciones de muchos a muchos, ni las relaciones poseen atributos como tal, siempre provienen de su entidad, por lo que nuestro Esquema Conceptual Modificado (ECM) quedará exactamente como el anterior Esquema Conceptual mostrado en la figura 3.12.

Tablas de la aplicación

A partir del ECM que acabamos de representar, podemos obtener un total de 5 tablas en la base de datos, teniendo en cuenta que:
-Cada entidad del ECM se transforma en una tabla.
-Los atributos de una entidad se convierten en los campos de las tablas respectivas. Por tanto, según el ECM, obtendríamos las siguientes tablas:

De esta forma, nos quedarían las siguientes tablas:
USUARIO
TAREA
ASIGNATURA
PRACTICA
EXAMEN

A continuación se explican los atributos de cada una de las tablas:

TABLA USUARIO

Se trata de una tabla que contiene tantas filas como usuarios hay registrados en nuestra aplicación.

Los atributos que esta tabla contiene son los siguientes:

- MAIL: Cadena de 64 caracteres. Llave primaria. Identificador numérico único para el usuario. Dirección de correo-e del usuario.
- NOMBRE: Cadena de 25 caracteres. Nombre del usuario.
- CONTRASEÑA: Cadena de 32 caracteres. Contraseña del usuario codificada mediante un algoritmo de cifrado.
- FECHA_ULT_ACC: Cadena de caracteres con formato de fecha para guardar la fecha de última modificación de los datos del usuario.

TABLA TAREA

Se trata de una tabla que contiene las tareas almacenadas en el sistema.

Los atributos que esta tabla contiene son los siguientes:

- ID: Entero. Llave primaria. Identificador numérico único para la tarea.
- NOMBRE: Cadena de 25 caracteres. Nombre de la tarea.
- DESCRIPCIÓN: Cadena de 60 caracteres. Describe el contenido de la tarea.
- FECHA: Cadena de caracteres con formato determinado “dd/mm/aaaa”. Indica la fecha de finalización.
- SUPERADA: Valor booleano. Muestra si la tarea ya ha sido completada o no.
- FECHA_ULT_ACC: Cadena de caracteres con formato de fecha para guardar la fecha de última modificación de los datos del usuario.

TABLA ASIGNATURA

Se trata de una tabla que contiene las asignaturas almacenadas en el sistema.

Los atributos que esta tabla contiene son los siguientes:

- ID: Entero. Llave primaria. Identificador numérico único para la asignatura.
- NOMBRE: Cadena de 25 caracteres. Nombre de la asignatura.
- PROFESOR: Cadena de 25 caracteres. Nombre del profesor de la asignatura.
- AULA: Cadena de 25 caracteres. Indica el aula en el que se imparte la asignatura.
- FECHA_ULT_ACC: Cadena de caracteres con formato de fecha para guardar la fecha de última modificación de los datos del usuario.

TABLA PRACTICA

Se trata de una tabla que contiene las prácticas almacenadas en el sistema.

Los atributos que esta tabla contiene son los siguientes:

- ID: Entero. Llave primaria. Identificador numérico único para la práctica.
- NOMBRE: Cadena de 25 caracteres. Nombre de la práctica.
- DESCRIPCIÓN: Cadena de 60 caracteres. Describe el contenido de la práctica.
- FECHA: Cadena de caracteres con formato determinado “dd/mm/aaaa”. Indica la fecha de finalización.
- SUPERADA: Valor booleano. Muestra si la práctica ya ha sido completada o no.
- FECHA_ULT_ACC: Cadena de caracteres con formato de fecha para guardar la fecha de última modificación de los datos del usuario.

TABLA EXAMEN

Se trata de una tabla que contiene los exámenes almacenados en el sistema.

Los atributos que esta tabla contiene son los siguientes:

- ID: Entero. Llave primaria. Identificador numérico único para el examen.
- NOMBRE: Cadena de 25 caracteres. Nombre del examen.
- FECHA: Cadena de caracteres con formato determinado "dd/mm/aaaa". Indica la fecha de finalización.
- SUPERADA: Valor booleano. Muestra si la práctica ya ha sido completada o no.
- FECHA_ULT_ACC: Cadena de caracteres con formato de fecha para guardar la fecha de última modificación de los datos del usuario.

3.4.2 Diseño de la Interfaz

En esta etapa del diseño del Sistema Software definimos la apariencia visual de nuestra aplicación, esto es, la interfaz visual que el sistema mostrará al usuario y a través del cual se relacionarán sistema y usuario. Sin lugar a duda, llevar a cabo un buen diseño de la interfaz resultará determinante para nuestra aplicación, como hemos dicho en el epígrafe referido a los requerimientos (Ver sección 3.2.2.), debido a que esta debe ser atractiva para el usuario de la aplicación pero al mismo tiempo debe resultar fácil de entender y de trabajar sobre ella [47].

Para llevar a cabo el diseño de la interfaz, puesto que el lenguaje de programación utilizado es Java, contamos con Swing, que es la principal herramienta GUI (del inglés Graphical User Interface) de Java. Forma parte de las clases Java fundamentales implementadas por Oracle, y dada su integración en Java se ha considerado que sería la mejor opción para nuestro proyecto.

La ingeniería del software define en esta fase los siguientes aspectos:
- Guía de estilo.
- Metáforas.
- Pantallas.
- Caminos de navegación (Storyboards).

3.4.2.1 Guía de estilo

Antes de diseñar una interfaz de usuario, es conveniente definir el estilo de la misma. Esto es de vital importancia cuando el diseño va a ser compartido entre varios diseñadores, ya que evita el caos y la desorganización entre éstos y mantiene la coherencia interna de la interfaz.

Sin embargo, a pesar de lo que pueda parecer en un principio, también es de gran utilidad definir una guía de estilo cuando sólo hay un diseñador encargado de la interfaz. Esto es debido a varias razones:

- A veces mantener la coherencia y consistencia de una interfaz, si ésta es muy grande o muy ambiciosa, es complicado incluso si sólo hay un diseñador.

- El diseñador puede, por las más diversas razones, abandonar el diseño y es útil para sus sustitutos contar con una guía de estilo predefinida para no tener que empezar de cero o perderse en los diseños ya realizados. Lo mismo puede aplicarse si no es el diseñador original el que se encarga del mantenimiento o la actualización de la interfaz.

- A lo largo del tiempo el mismo diseñador no tiene por qué tener contacto con la aplicación, y para solucionar algún problema en un momento dado, esta guía podría servirle de apoyo y recordatorio.

Una vez demostrada la utilidad del uso de la guía de estilo, pasamos a definir las reglas y normas que contendrá la guía de estilo de nuestra interfaz:

Fuentes: para los formularios de la aplicación, así como para los datos que se muestran de cada usuario:

- Tipo de letra: "Lucida Grande"
- Tamaño: 13pt;
- Color de la letra: # 000000;

En algunos casos, el tamaño de la letra es mayor para destacar el contenido (Es el caso del nombre de usuario, de la carrera y el nombre de cada actividad del usuario, que se mostrará a 13pt.). En otros, el tamaño de la letra es menor, puesto que los campos no son relevantes (se mostrará la letra a 11pt.).

Imagen de fondo: En todo momento, la aplicación mostrará la imagen de fondo que se muestra en la figura 3.13.

Colores: El color viene dado por la imagen de fondo. Aunque en los paneles que muestren datos de información, para separar las asignaturas, tareas, prácticas y exámenes, se mostrarán en un fondo gris plata.

Figura 3.13: Imagen de fondo de la aplicación.

3.4.2.2 Metáforas

Entendemos por metáfora el empleo de la imagen de un objeto o proceso familiar para el usuario, con el fin de representar alguna tarea abstracta realizada por ordenador. Algunas se vuelven completamente estándares para el usuario, como puede ser la metáfora de utilizar una papelera para eliminar los documentos. Al diseñar una interfaz gráfica, la utilización de metáforas resulta muy útil ya que permiten al usuario, por comparación con otro objeto o concepto, comprender de una manera más intuitiva las diversas tareas que se van a desarrollar mediante la interfaz.

Para que una metáfora cumpla bien su función, el desarrollador de la aplicación y el usuario final de ésta deben tener una base cultural similar. De lo contrario, no se entenderá el mensaje, e

incluso puede provocar un rechazo si éste se considera ofensivo en la cultura de destino. Por ejemplo, es muy posible que el uso de un icono de manera metafórica sea entendido de una manera por el usuario occidental y de otra, bien distinta, por un usuario oriental. En definitiva, hay que intentar que las metáforas que usemos sean lo más universales posibles, con la intención de que sean comprendidas a la perfección por la mayor parte del público potencial.

Las metáforas, dentro del proceso de diseño de una interfaz de usuario, pueden considerarse de dos tipos: de proceso o visuales. Metáforas de proceso son aquellas en las que la metáfora se realiza con un proceso o acción a realizar. Un ejemplo de ello sería la acción `copiar y pegar'. Las metáforas visuales se realizan simplemente mediante la utilización de iconos con los que establezcamos una relación. Como ejemplos de metáforas visuales tenemos los iconos y botones de pulsación.

Las aplicaciones de escritorio de Java Swing se adaptan al sistema operativo, siguiendo en cada uno su guía de estilo propio y facilitando la familiarización del usuario con el entorno. En nuestro caso, al utilizar Mac, Java tomaría la Guía de Estilo Mac Os X y utilizaría una serie de metáforas con las que el usuario Mac ya está plenamente familiarizado.

No conviene tampoco abusar de este tipo de metáforas, pues si bien hasta un punto facilitan la interacción usuario - sistema, en sobreabundancia podría resultar incómodas e ineficaces para el usuario. El objetivo a la hora de utilizarlas siempre debe ser hacer más sencilla e intuitiva la aplicación, nunca lo contrario.

En nuestra aplicación hemos utilizado las metáforas que se muestran en la siguiente tabla (figura 3.14):

Icono	Función
	Permite la edición de la asignatura, tarea, práctica o examen al que acompaña.
	Confirma la edición realizada sobre la asignatura, tarea, práctica o examen al que corresponde.

Icono	Función
	Elimina la asignatura, tarea, práctica o examen al que acompaña.
	Accede a las tareas de la asignatura a la que corresponde.
	Accede a las prácticas de la asignatura a la que corresponde.
	Accede a los examenes de la asignatura a la que corresponde.
	Permite cerrar la sesión de usuario.

Figura 3.14: Iconos de la Interfaz.

3.4.2.3 Prototipos de la Aplicación

En este apartado, vamos a definir la estructura de nuestra interfaz con el usuario, es decir, la interfaz de la aplicación de escritorio, puesto que la parte del servidor no tendrá interfaz ninguna al ejecutarse automáticamente sin necesidad de intervenir con el usuario.

Mediante el diseño de prototipos pretendemos alcanzar un esbozo de lo que será la interfaz final de usuario de la aplicación. Dichos prototipos no muestran el diseño final en su conjunto, pues lo único que muestran es una idea o boceto de cómo sería la interfaz final; con lo cual estos prototipos que se van a mostrar a continuación están sujetos a posibles cambios.

He optado por hacer los diseños con la aplicación Mockups de Balsamiq[2], por su simplicidad y las numerosas opciones que tiene de añadir componentes de aplicaciones reales, siendo un fiel reflejo de una aplicación real que nos muestra como queda nuestra aplicación. Las pantallas se muestran entre las figuras 3.15 y 3.21.

[2] Pagina web de la aplicación Mockups: http://balsamiq.com/products/mockups/

Pantalla de Acceso

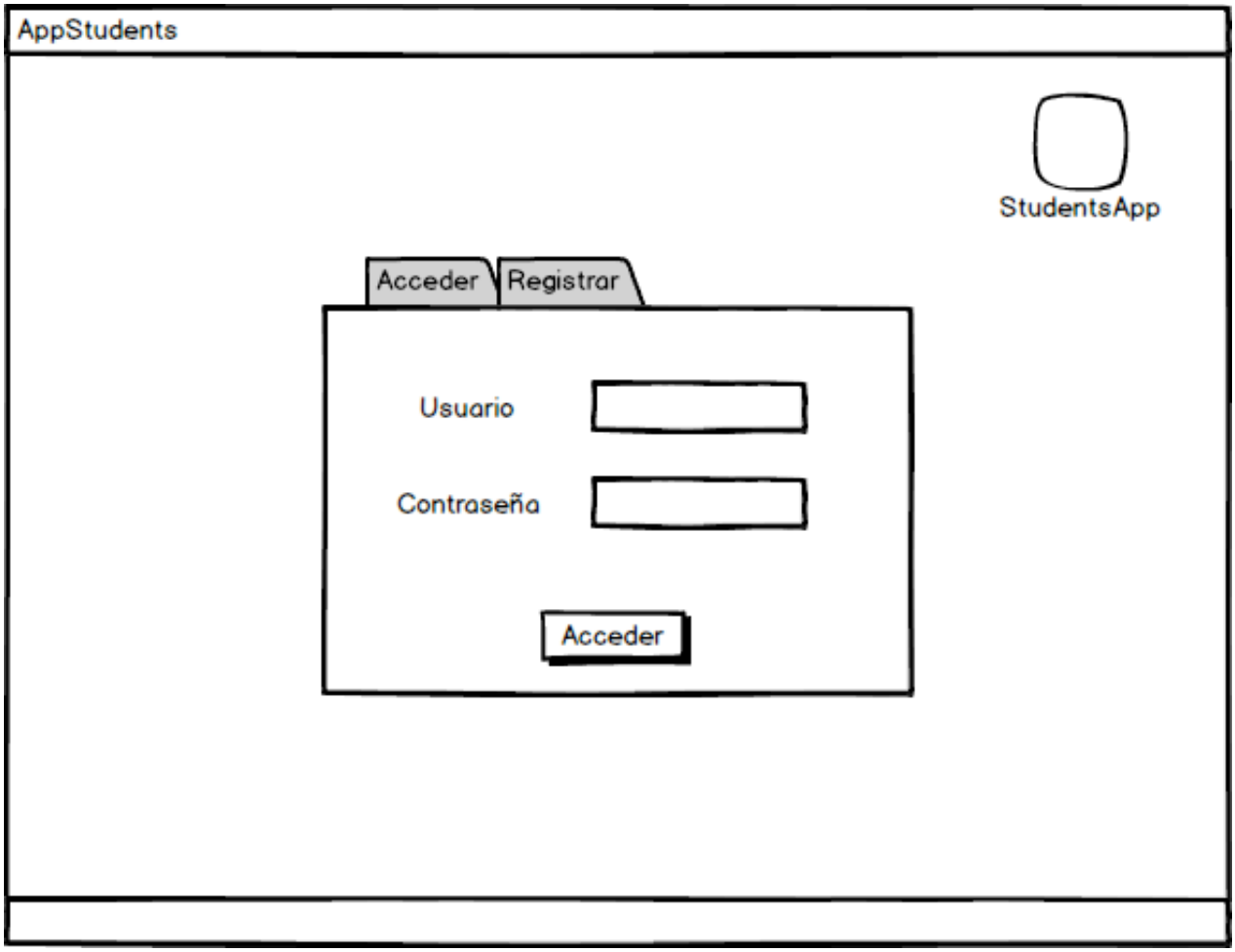

Figura 3.15: Pantalla de acceso.

Pantalla de Registro

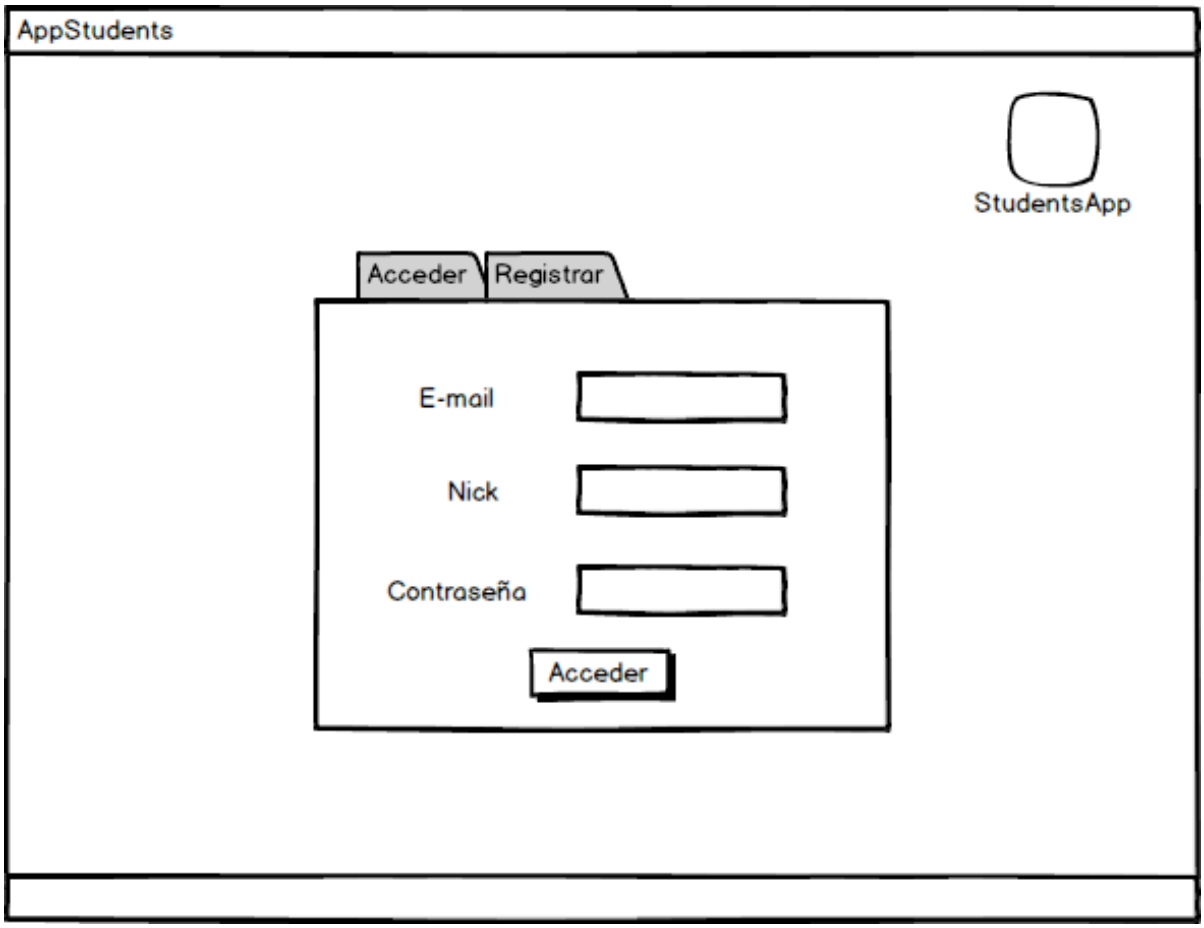

Figura 3.16: Pantalla de registro.

Pantalla principal: Asignaturas

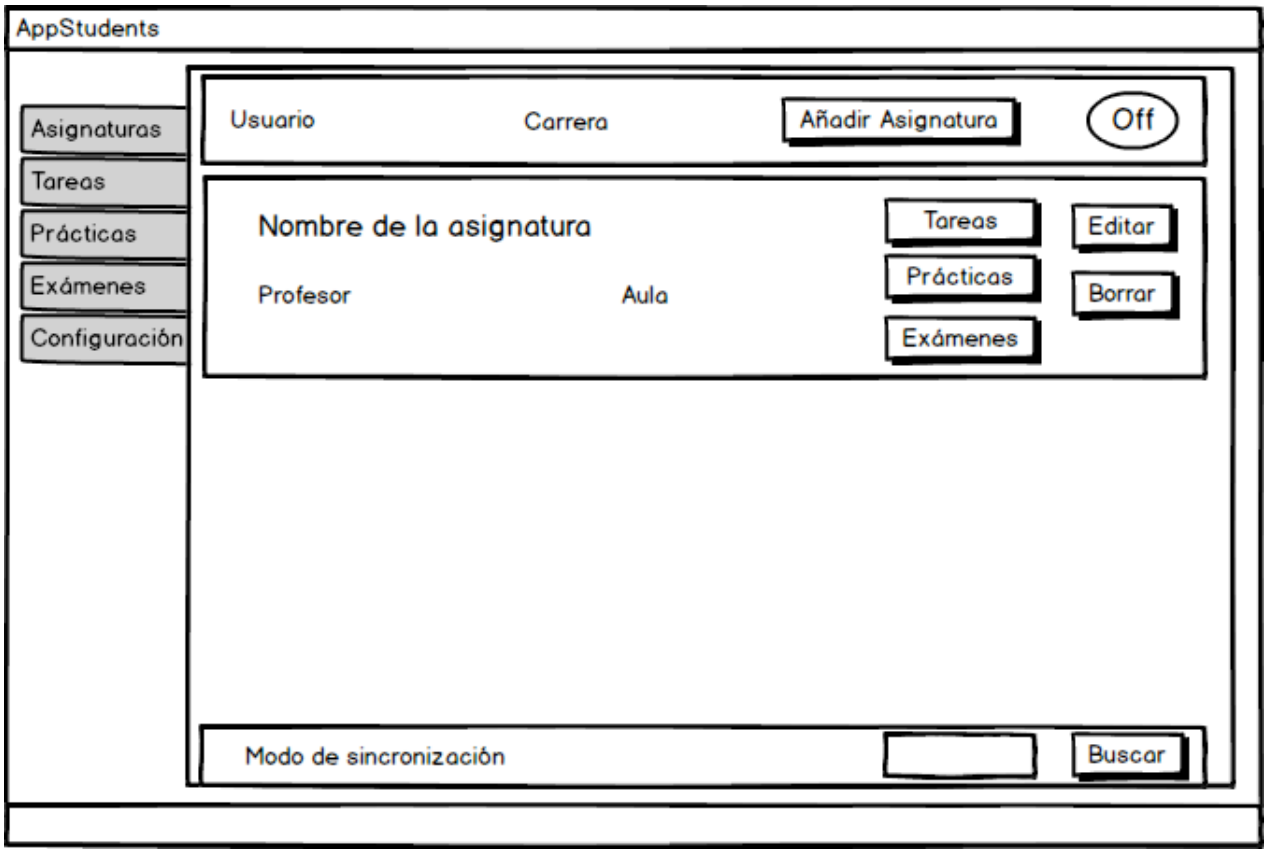

Figura 3.17: Pantalla principal: Asignaturas.

Pantalla de Tareas

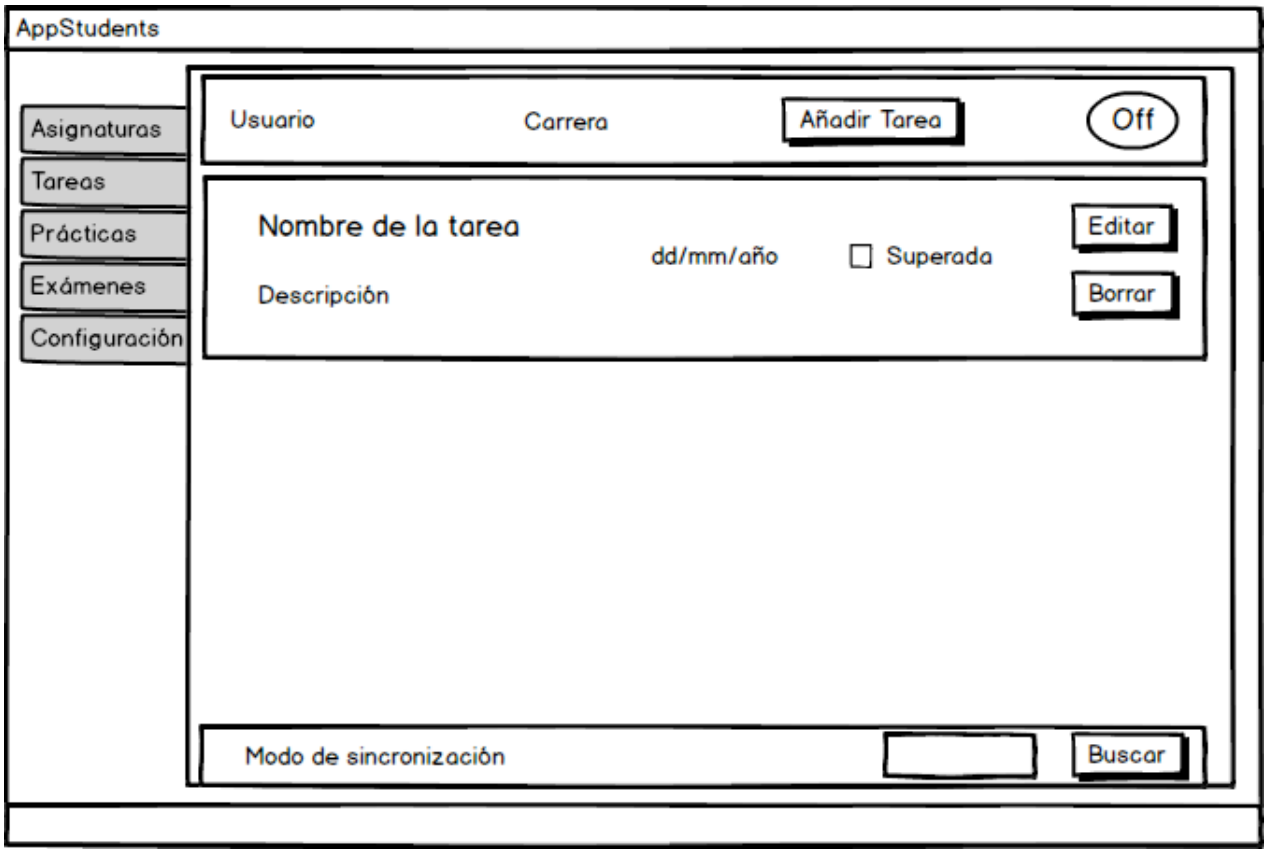

Figura 3.18: Pantalla de Tareas.

Pantalla de Prácticas

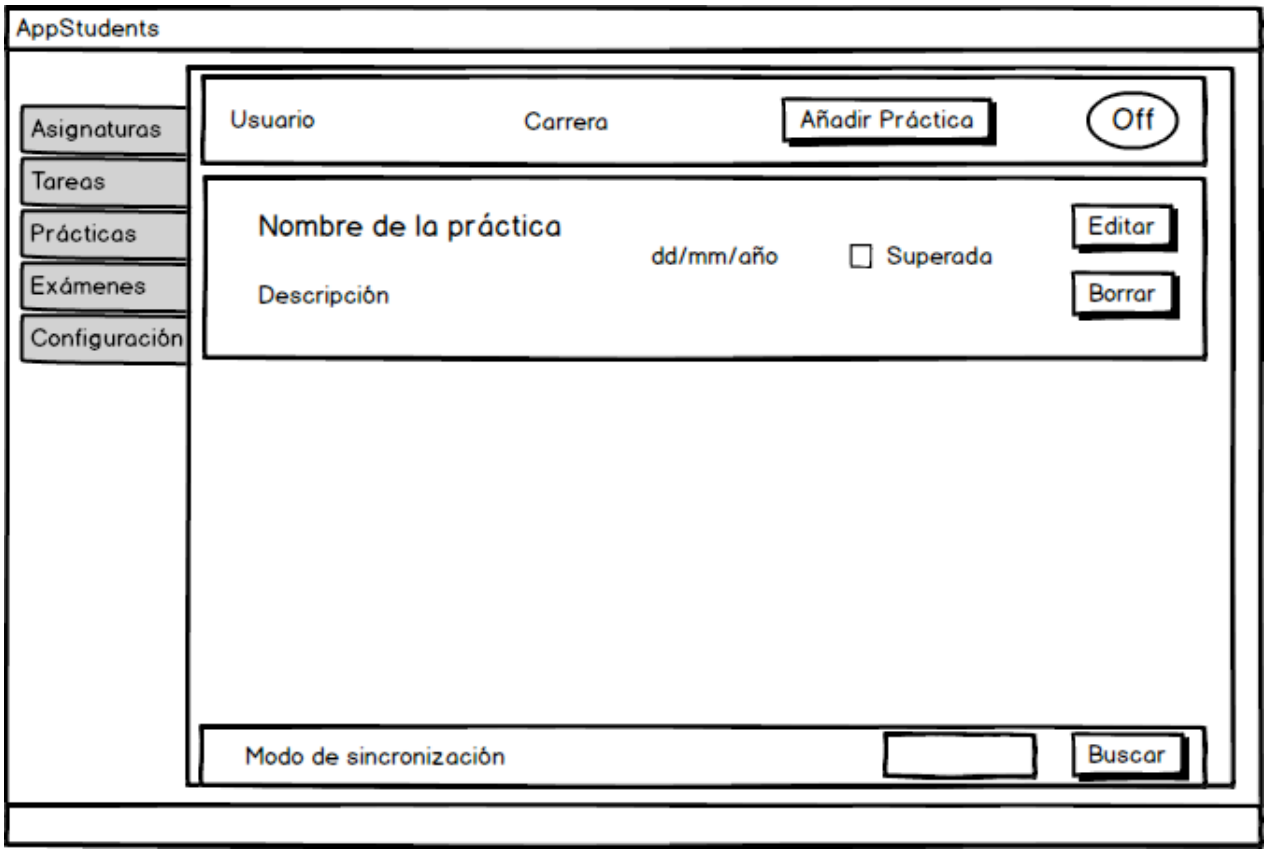

Figura 3.19: Pantalla de Prácticas.

Pantalla de Exámenes

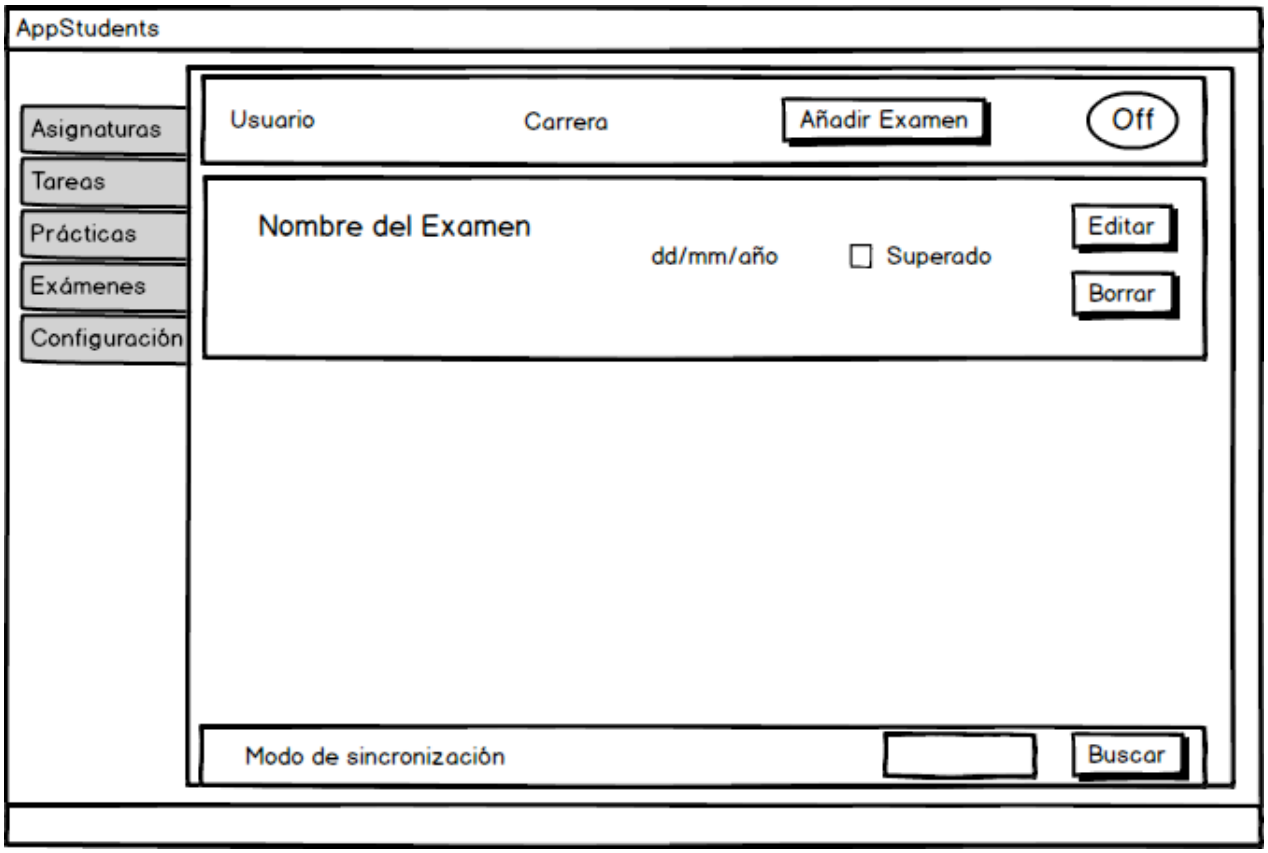

Figura 3.20: Pantalla de Exámenes.

Pantalla de Perfil

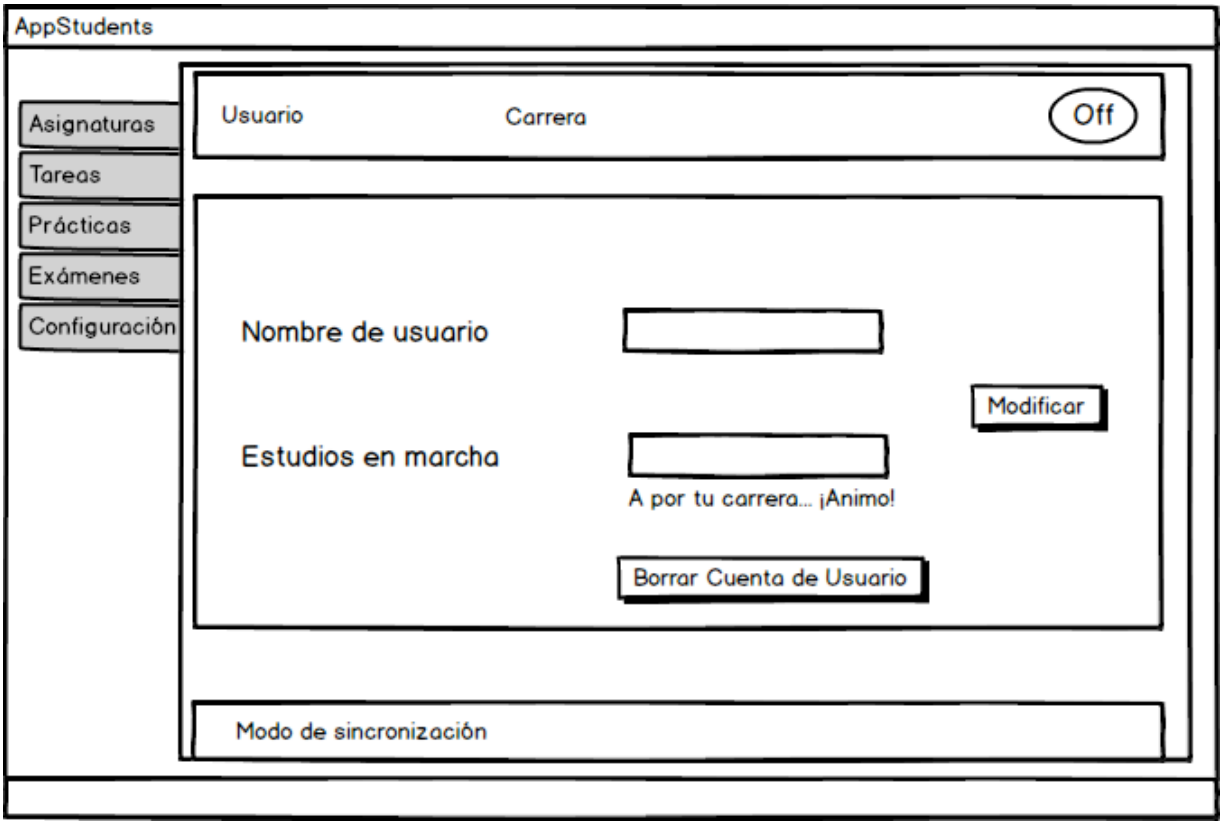

Figura 3.21: Pantalla de Perfil.

Storyboards

Hasta ahora el diseño que se ha mostrado se ha hecho de manera estática, esto es, las pantallas separadas entre ellas sin saber qué interacción realizan realmente entre ellas. Para mostrar la interacción que se produciría al utilizar el usuario la aplicación, vamos a diseñar los "Storyboards", que no son más que las herramientas que permiten mostrar, a modo de secuencia, las diferentes pantallas por las que el usuario va pasando al realizar cada una de las posibles acciones que la aplicación nos permite llevar a cabo.

El procedimiento sería el siguiente: se disponen las capturas de pantalla de la interfaz unidas en una misma imagen y relacionadas mediante flechas, de tal manera que se indique así el camino que seguiría la interacción.

La posición de origen de cada flecha indica cuál es el elemento que desencadena la acción en la pantalla de origen, y el destino de la flecha, cuál sería el resultado de esa acción, es decir, qué pantalla se mostraría a continuación. No obstante, en cada caso se incorpora un pequeño comentario explicando el significado de la acción. Además, en cada Storyboard se ha creído conveniente incorporar números que indiquen el orden de las acciones que se van realizando, para facilitar así la comprensión del lector.

El storyboard serviría de prototipo para ser evaluado por usuarios de prueba y poder modificar así las correcciones en fases tempranas; pues cuanto más tiempo se tarde en validar una interfaz, más coste de tiempo y trabajo (y también económico, si se tratara de un proyecto comercial) tendrá.

Los storyboards están bastante relacionados con los casos de uso que presentamos en la sección 3.3.1; se han intentado agrupar de una manera comprensible y que incluya cada una de las acciones que puede llevar a cabo nuestra aplicación:
Storyboard 1: "Acceso, registro y cierre de sesión", mostrado en la figura 3.22.
Storyboard 2: "Gestión de asignaturas y tareas", mostrado en la figura 3.23.
Storyboard 3: "Gestión de prácticas", mostrado en la figura 3.24.
Storyboard 4: "Gestión de exámenes", mostrado en la figura 3.25.
Storyboard 5: "Consulta y gestión del perfil de usuario", mostrado en la figura 3.26.

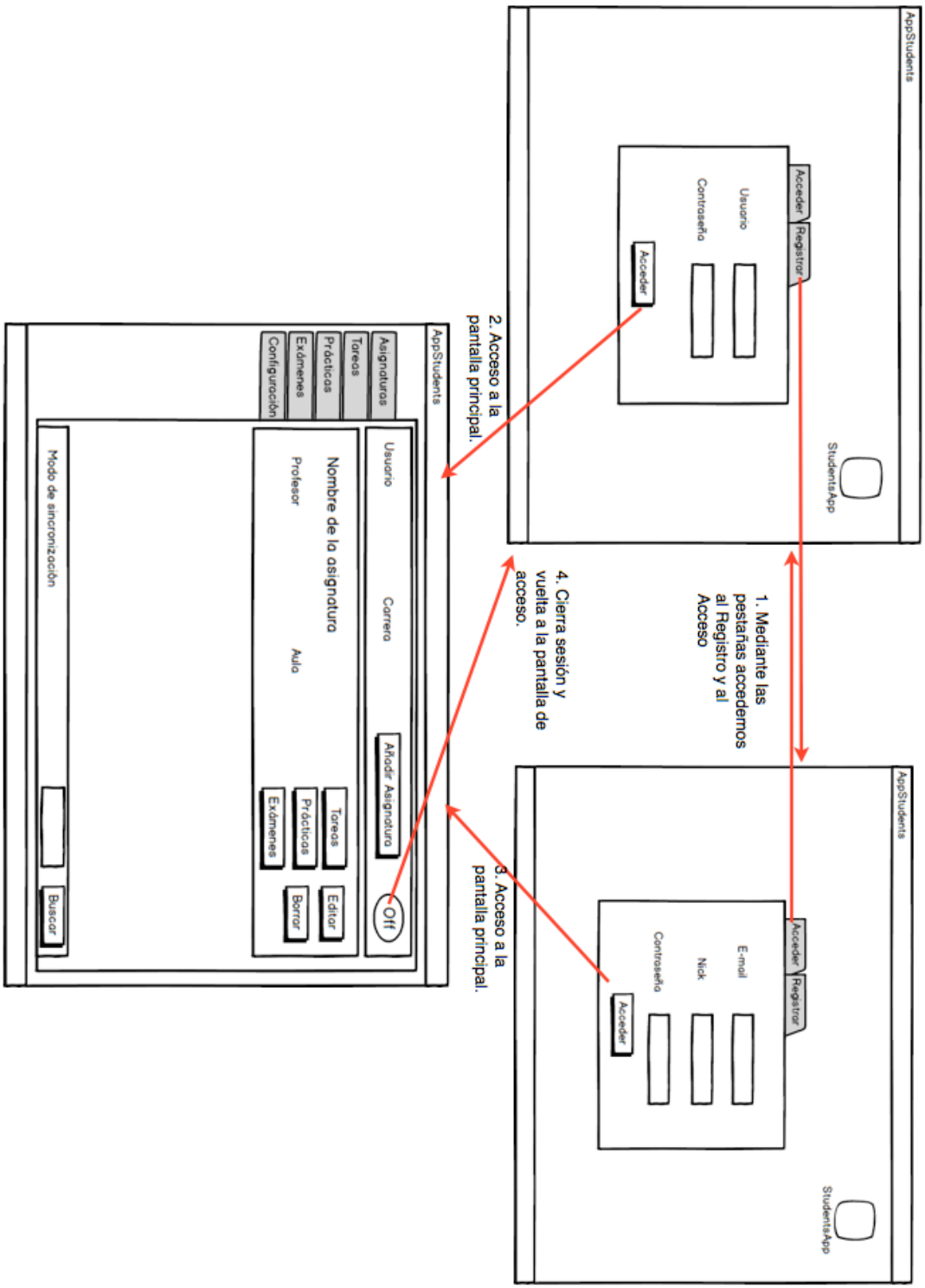

Figura 3.22: Storyboard 1: Acceso, registro y cierre de sesión.

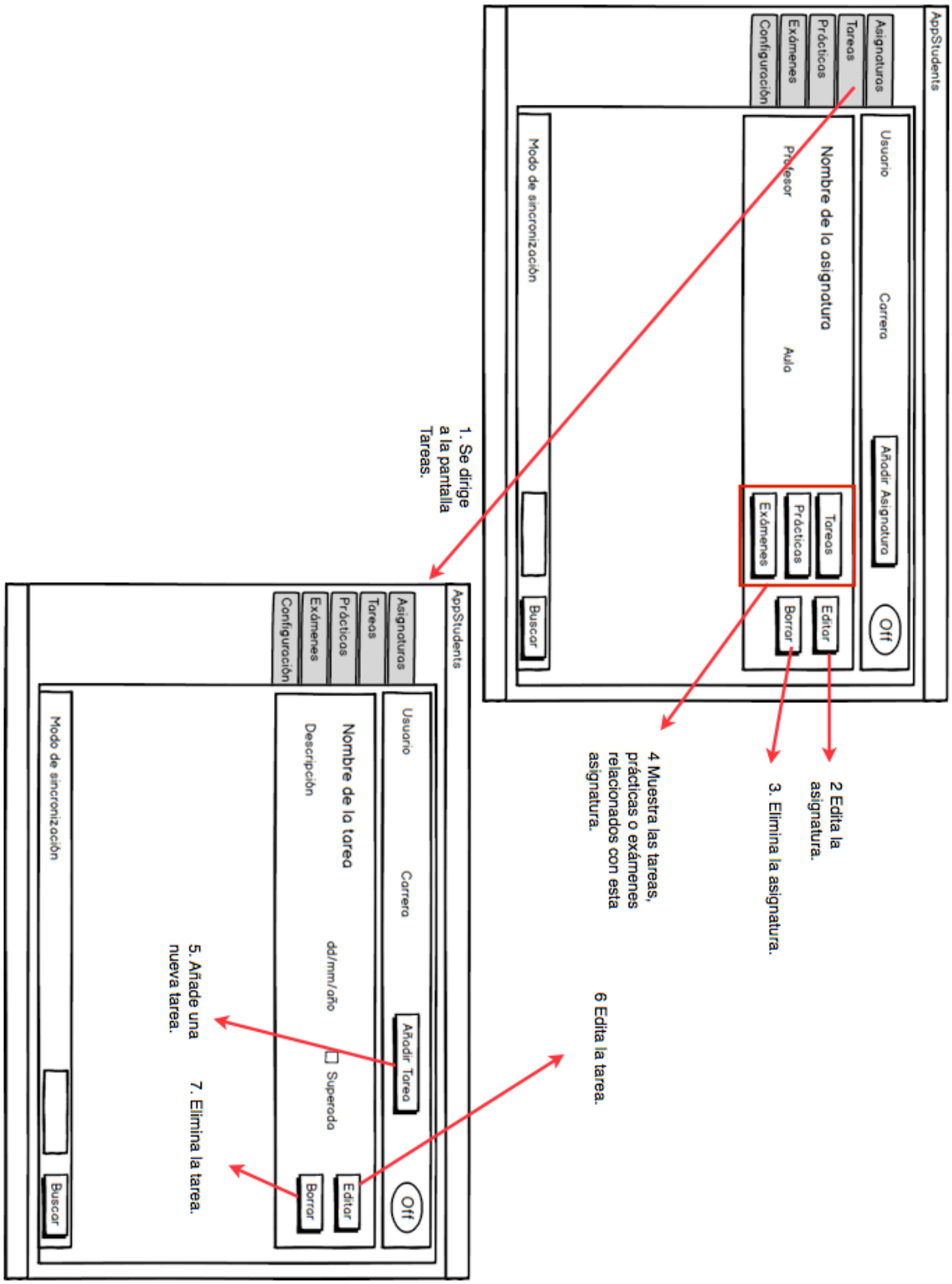

Figura 3.23: Storyboard 2: Gestión de asignaturas y tareas.

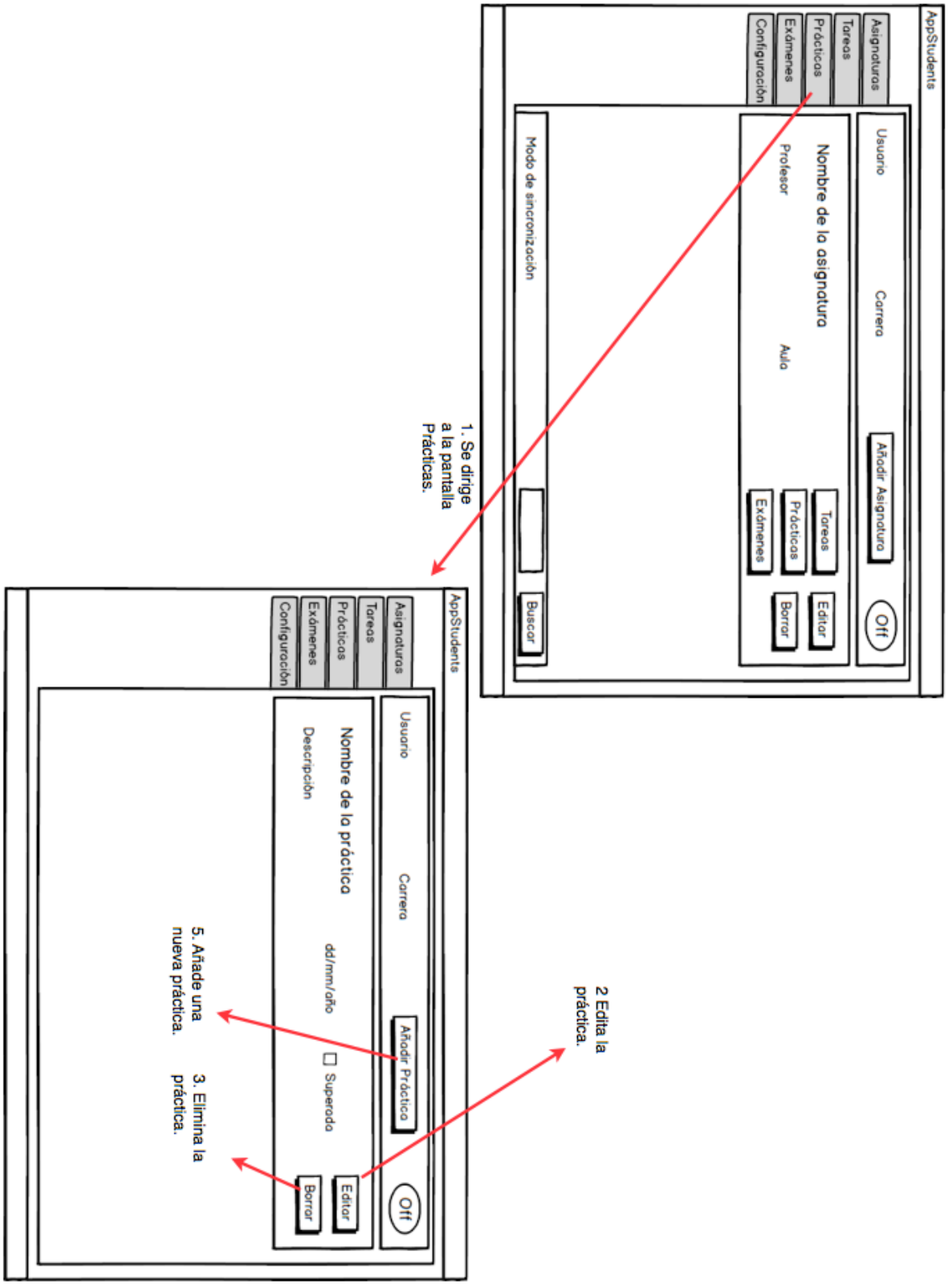

Figura 3.24: Storyboard 3: Gestión de prácticas.

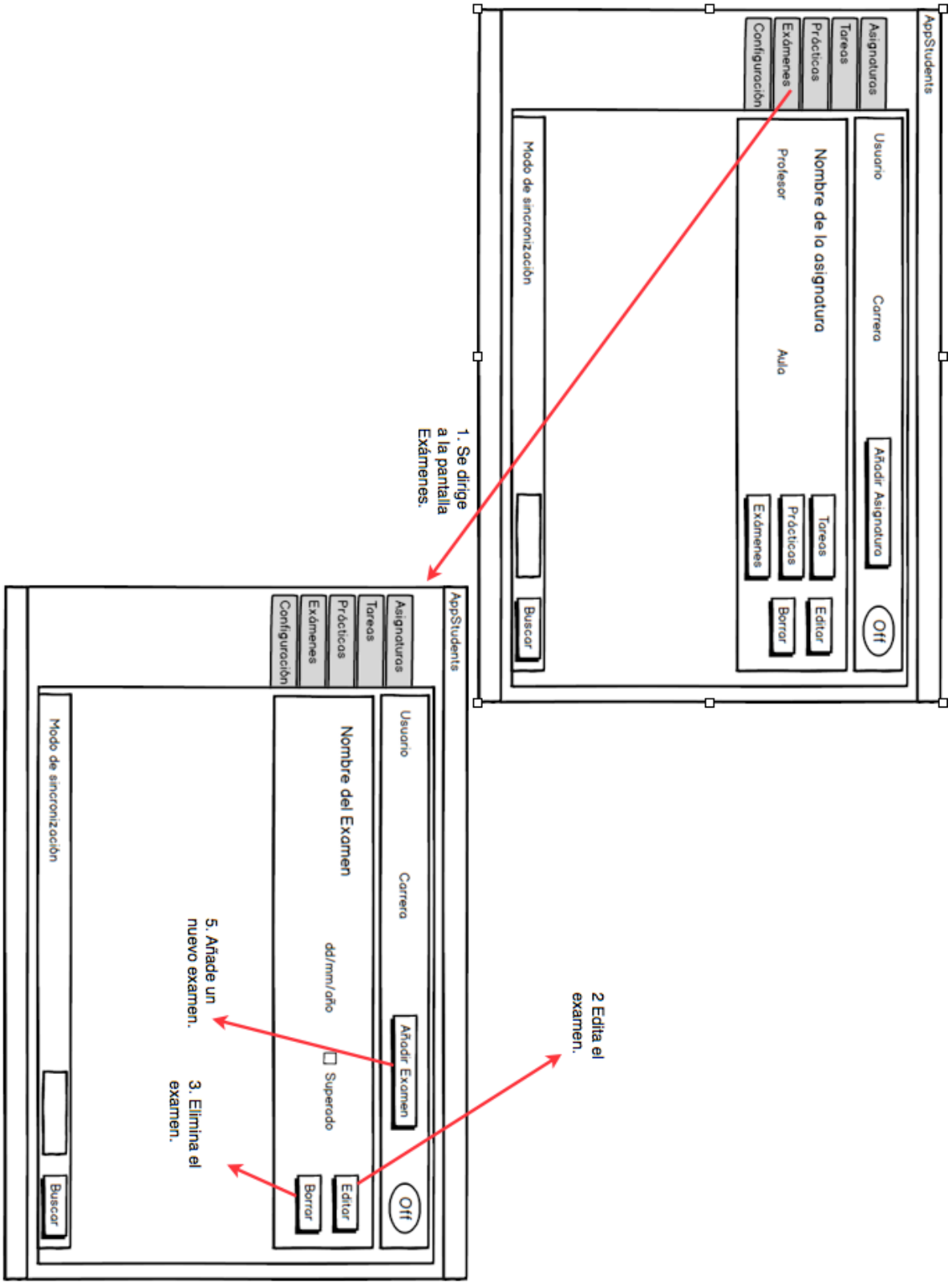

Figura 3.25: Storyboard 4: Gestión de exámenes.

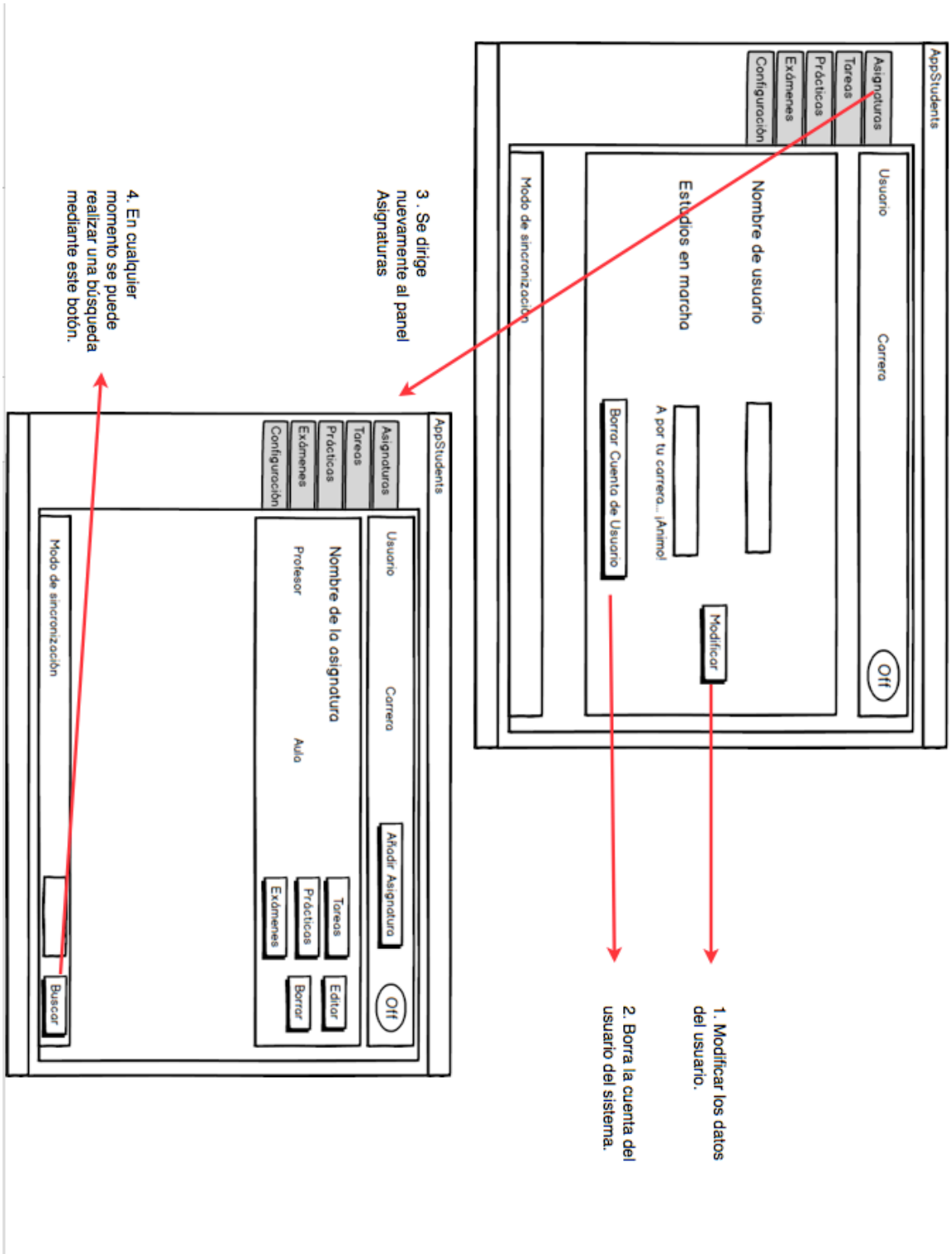

Figura 3.26: Storyboard 5: Consulta y gestión del perfil de usuario.

3.5 Implementación

La implementación es la fase de la Ingeniería del Software en la que el modelo obtenido en las actividades anteriores se debe traducir a código fuente. Durante esta fase se eligen las herramientas y tecnologías que van a utilizarse para la codificación del proyecto.

En nuestro caso, la elección del lenguaje de programación así como la herramienta utilizada para su desarrollo viene dada desde la definición del proyecto. Aún así, en este apartado se hablará mas detalladamente sobre la aplicación, las herramientas utilizadas, la plataforma que utiliza la aplicación (Google App Engine), la tecnología que conecta la aplicación de escritorio con la plataforma (servicios REST), y el algoritmo utilizado para la sincronización de los datos.

3.5.1 Tipo de arquitectura de la aplicación

Nuestro proyecto se divide en dos partes bien diferenciadas:

- La parte de la aplicación de escritorio, que será implementada en el lenguaje de programación Java, llevará a cabo la comunicación con el usuario a través de la interfaz definida en la fase de diseño, y dispondrá de una base de datos local donde se almacenarán los datos temporales del usuario.

- La parte del servidor web en la plataforma de Google para la nube, Google App Engine, que almacenará los datos del usuario en la nube y los cargará nuevamente cuando el usuario los necesite.

Las conexiones entre cada una de las partes de la aplicación se realizarán mediante las peticiones que ofrece la tecnología REST (Representational State Transfer), que posibilita el paso de cada dato del usuario entre la aplicación de escritorio y la nube sin problemas.

A continuación, en la figura 3.27 se muestra un esquema de la arquitectura general que se utiliza en el proyecto.

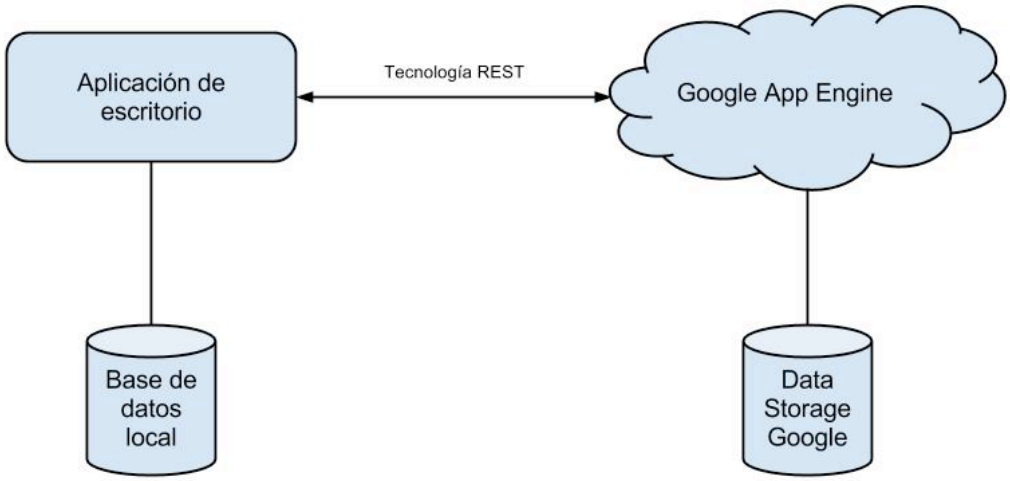

Figura 3.27. Arquitectura del proyecto.

3.5.2 Herramienta de desarrollo.

La herramienta escogida para desarrollar todo el código ha sido Eclipse Juno (versión 4.2), al cual se le ha añadido el plugin "appengine-java-sdk-1.8.1.1" distribuido por Google para permitir el funcionamiento de la plataforma cloud de Google dentro de nuestro entorno de desarrollo Eclipse. Podemos ver la interfaz del entorno en la figura 3.28.

También debemos destacar que se ha utilizado SQLite para gestionar la base de datos local, donde se almacenarán los datos del usuario en su equipo. La elección por ésta ha sido por su simplicidad, y porque los datos de la aplicación tampoco presentan una elevada complejidad.

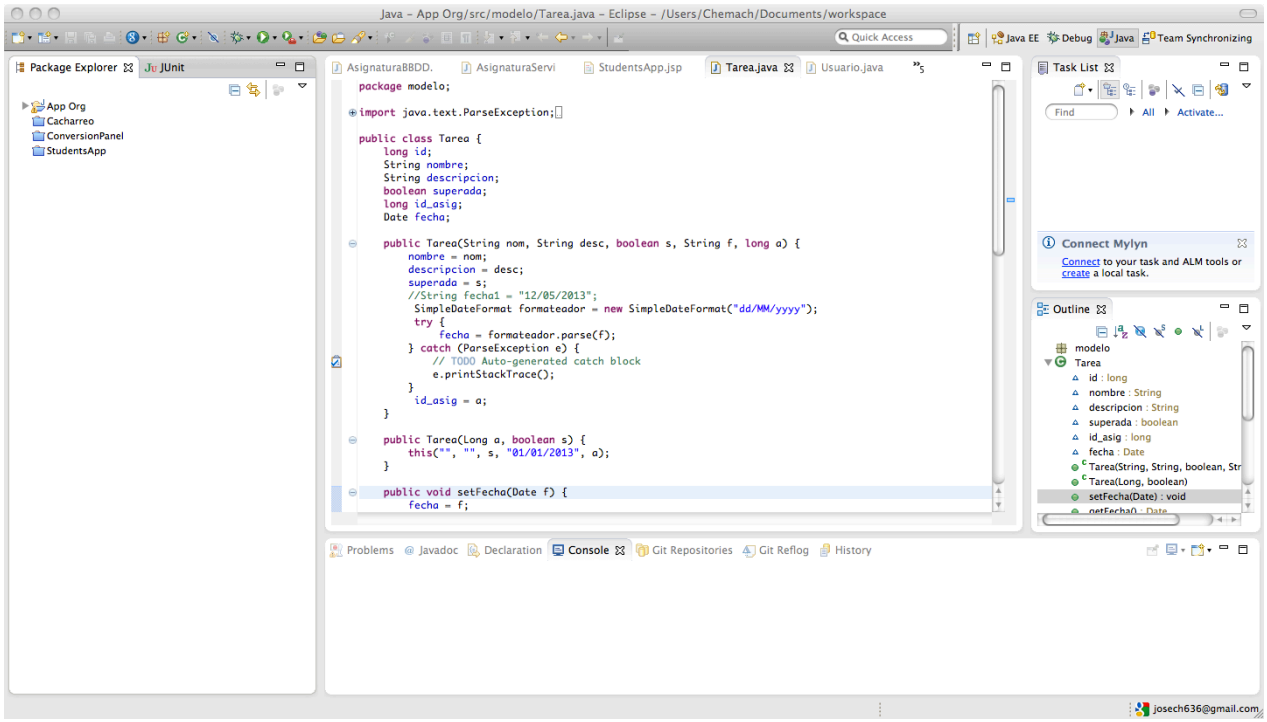

Figura 3.28. Vista de Eclipse Juno con el plugin de Google App Engine.

3.5.3 Almacenamiento de datos en Google App Engine

Para la gestión de la base de datos, que es lo que nos concierne aquí (pues recordemos que vamos a utilizar GAE en nuestro proyecto para cargar y descargar los datos de los usuarios desde la nube) cabe destacar que GAE dispone de su propio almacén de datos, ahorrándonos el trabajo de configurar y gestionar dicho almacén con MySQL, SQLite u otros. Nuestros datos quedan almacenados en Google Datastore[3], cuyo enfoque jerárquico está orientado a objetos. Además Google facilita la implementación al usuario, permitiendo en GAE el uso de JPA (Java Persistence API) y JDO (Java Data Objects), aunque también dispone de funciones propias para que el usuario utilice el almacén de datos a más bajo nivel [34]. Esta última opción es por la que se ha optado en este proyecto, pues los datos que se manejan en la aplicación no son lo suficientemente complejos, y además la utilización de estas funciones en Google es bastante simple (consiste en un conjunto de entidades con atributos que el mismo Google se encarga de traducir a valores más simples).

[3] Web de Google App Engine: Datastore. https://developers.google.com/appengine/docs/java/datastore/

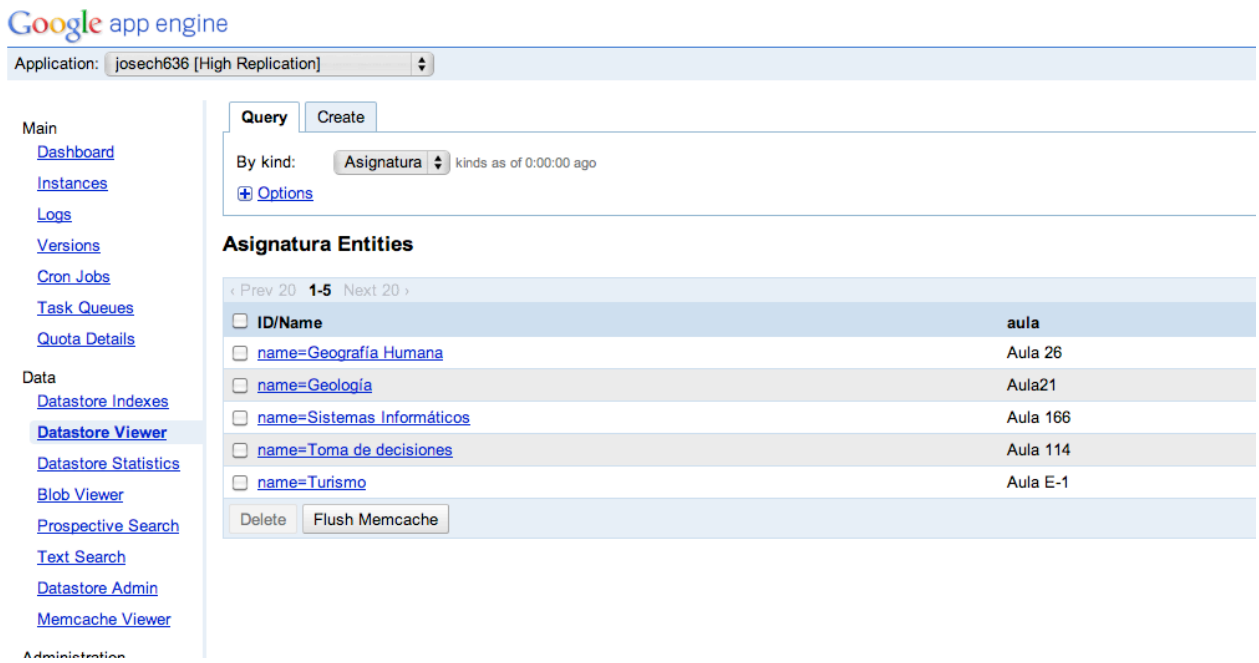

Figura 3.29. Google App Engine: Datastore.

Una vez realizado nuestro proyecto, no tenemos más que cargarlo en GAE y permitir que los datos se sincronicen en la nube. Se muestra el visor de Google Datastore en la figura 3.29.

3.5.4 Tecnología REST (Representational State Transfer)

Como parte de un trabajo de doctorado, Roy Fielding generalizó los principios de la arquitectura de la Web y los presentó como un framework. A través de este marco, Fielding describió cómo se construyen y operan los sistemas de información distribuidos en la Web. Él describió la interacción entre los recursos y el papel de los identificadores únicos en el sistema de búsqueda. Además, habló sobre el uso de un conjunto limitado de operaciones con una semántica uniforme para construir una infraestructura ubicua que pudiera soportar cualquier tipo de aplicación. A esto, Fielding lo denominó REST. REST (Representational State Transfer) describe la Web como una enorme aplicación distribuida hipermedia cuyos recursos vinculados se comunican mediante el intercambio de representaciones de los estados de los recursos [11].

Las arquitecturas REST constan usualmente de clientes y servidores. Los clientes inician solicitudes a los servidores, y los servidores se encargan de tramitar estas solicitudes y devolver

las respuestas apropiadas. Las solicitudes y las respuestas se construyen alrededor de la transferencia de las representaciones de recursos, como se muestra en la figura 3.30.

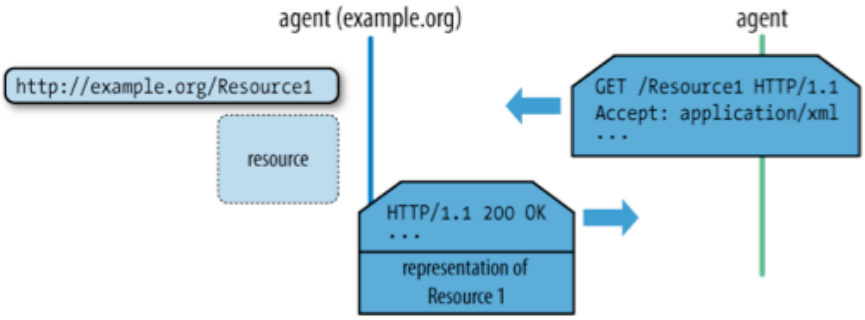

Figura 3.30. Usando HTTP para obtener un recurso mediante servicios REST [11].

Un recurso puede ser cualquier concepto, dato o información que queramos pasar del cliente al servidor o viceversa.

Con relación a nuestro proyecto, la labor de REST será transportar los recursos que el cliente aloje en su base de datos en el ordenador SQLite a la nube, cuando éstos se modifiquen, y en caso contrario, si los datos de la nube son posteriores a aquellos alojados en la base de datos local. Hablaremos de los criterios de orden y prioridad en la sincronización en el siguiente epígrafe.

3.5.5 Algoritmo de Sincronización

El algoritmo de sincronización posibilita la sincronización de los datos de cada usuario a su última actualización sin importar desde qué terminal se conecte el usuario. Para lograr esto, se revisa cada entidad perteneciente al usuario que existe en la nube, esto es, sus asignaturas, tareas, prácticas y exámenes y se obtienen metadatos de cada uno de ellos. Los metadatos que se obtienen son: el id de la entidad, el tipo de entidad, el email del usuario, y la fecha de última modificación. A continuación se envía una lista de los metadatos desde la nube, que comprobando los metadatos de la nube con los propios de la aplicación etiquetará éstos indicando qué operación ha de realizarse con cada metadato en concreto [48]. Las posibles operaciones son:

- Nuevo: si la entidad se ha creado por primera vez y se ha comprobado en Google Datastore que no pertenecía a la lista de ids borradas.

- Borrar: si la entidad existe pero se ha comprobado en Google Datastore que sí pertenecía a la lista de ids borradas. Esto puede ocurrir cuando un usuario borra una entidad desde un ordenador, pero posteriormente entra a otro ordenador que ya tenía un estado anterior de sus datos y existen esas entidades que en realidad ya han sido borradas.

- Modificación a la nube: Si la fecha de última modificación en la nube es anterior a la de la aplicación. En este caso, posteriormente, se actualizarán los datos a la nube.

- Modificación a la aplicación local: Si la fecha de última modificación en la nube es posterior a la de la aplicación. A continuación, se descargará la entidad de la nube y se actualizará en la base de datos local.

Mediante las etiquetas de estos metadatos, se procederá a la descarga, subida, creación o borrado de las entidades según sea necesario para sincronizar la aplicación. Además, cabe destacar que la aplicación se sincronizará si existe posibilidad de conexión a Internet, teniendo a cada momento información sobre la última fecha de sincronización si ésta existe. Si no es posible esta sincronización, se indicará que se está trabajando en modo local, esto es, utilizando los últimos datos del usuario alojados en la base de datos local de la aplicación. El código implementado del algoritmo de sincronización se encuentra en el Apéndice C que se incluye al final de esta misma memoria.

3.6 Pruebas

El objetivo de esta fase es realizar un conjunto de pruebas sobre el sistema que comprueben el correcto funcionamiento de éste. Con esto pretendemos garantizar que el sistema no incluye errores, consiguiendo, de esta manera, una alta calidad en el software ofrecido.

3.6.1 Casos de Test

Los test diseñados son los siguientes:

Test 1: Registro de usuario

Requisitos testeados	RF-01
Acción	Un usuario introduce su nombre de usuario, dirección de correo electrónico y contraseña y pulsa el botón “Registrar”.
Checkpoint 1	El sistema debe mostrar un mensaje de usuario registrado y mostrar el menú principal de dicho usuario.

Test 2: Registro de usuario incorrecto

Requisitos testeados	RF-01
Acción	Un usuario introduce su nombre de usuario, correo electrónico y contraseña y pulsa el botón “Registrar”.
Checkpoint 1	El sistema debe mostrar un mensaje de error informando de que el nombre de usuario o la contraseña o ambos son incorrectos.

Test 3: Autenticación de usuario

Requisitos testeados	RF-02
Acción	Un usuario introduce su nombre de usuario y contraseña y pulsa el botón “Acceder”.
Checkpoint 1	El sistema debe mostrar el menú principal de dicho usuario.

Test 4: Autenticación de usuario incorrecta

Requisitos testeados	RF-02
Acción	Un usuario introduce su nombre de usuario y contraseña y pulsa el botón "Acceder".
Checkpoint 1	El sistema debe mostrar un mensaje de error informando de que el nombre de usuario o la contraseña o ambos son incorrectos.

Test 5: Crear una asignatura

Requisitos testeados	RF-03
Precondiciones	Un usuario registrado se encuentra en el menú principal de la aplicación.
Acción	Accede a Asignaturas.
Checkpoint 1	El sistema debe mostrar la pantalla de Asignaturas.
Acción	El usuario pulsa sobre el botón "Añadir Asignatura".
Checkpoint 2	A las asignaturas ya añadidas, si las hubiera, se les añade un formulario de asignatura nueva.
Acción	El usuario rellena los datos de la nueva asignatura y pulsa sobre "Ok".
Checkpoint 3	El sistema guarda la nueva información, que ya no se muestra editable en pantalla.

Test 6: Modificar una asignatura

Requisitos testeados	RF-03
Precondiciones	Un usuario registrado se encuentra en la pantalla "Asignaturas" de la aplicación.
Acción	El usuario pulsa sobre el botón "Editar Asignatura" de la asignatura a modificar.
Checkpoint 1	Se muestra el formulario con los datos de la asignatura.

Acción	El usuario rellena nuevamente los datos que desee modificar y pulsa "Ok".
Checkpoint 2	El sistema guarda la nueva información, que ya no se muestra editable en pantalla.

Test 7: Eliminar una asignatura

Requisitos testeados	RF-03
Precondiciones	Un usuario registrado se encuentra en la pantalla "Asignaturas" de la aplicación.
Acción	El usuario pulsa sobre el botón "Borrar Asignatura" de la asignatura a eliminar.
Checkpoint 1	Se pregunta al usuario si realmente desea realizar esta acción.
Acción	El usuario pulsa "sí" para borrar la información.
Checkpoint 2	La asignatura borrada desaparece de la pantalla "Asignaturas".

Test 8: Crear una tarea

Requisitos testeados	RF-04
Precondiciones	Un usuario registrado se encuentra en el menú principal de la aplicación.
Acción	Accede a Tareas.
Checkpoint 1	El sistema debe mostrar la pantalla de Tareas.
Acción	El usuario pulsa sobre el botón "Añadir Tarea".
Checkpoint 2	A las tareas ya añadidas, si las hubiera, se les añade un formulario de tarea nueva.
Acción	El usuario rellena los datos de la nueva tarea y pulsa sobre "Ok".
Checkpoint 3	El sistema guarda la nueva información, que ya no se muestra editable en pantalla.

Test 9: Modificar una tarea

Requisitos testeados	RF-04
Precondiciones	Un usuario registrado se encuentra en la pantalla "Tareas" de la aplicación.
Acción	El usuario pulsa sobre el botón "Editar Tarea" de la tarea a modificar.
Checkpoint 1	Se muestra el formulario con los datos de la tarea.
Acción	El usuario rellena nuevamente los datos que desee modificar y pulsa "Ok".
Checkpoint 2	El sistema guarda la nueva información, que ya no se muestra editable en pantalla.

Test 10: Eliminar una tarea

Requisitos testeados	RF-04
Precondiciones	Un usuario registrado se encuentra en la pantalla "Tareas" de la aplicación.
Acción	El usuario pulsa sobre el botón "Borrar Tarea" de la tarea a eliminar.
Checkpoint 1	Se pregunta al usuario si realmente desea realizar esta acción.
Acción	El usuario pulsa "sí" para borrar la información.
Checkpoint 2	La tarea borrada desaparece de la pantalla "Tareas".

Test 11: Crear una práctica

Requisitos testeados	RF-05
Precondiciones	Un usuario registrado se encuentra en el menú principal de la aplicación.
Acción	Accede a Prácticas.
Checkpoint 1	El sistema debe mostrar la pantalla de Prácticas.

Acción	El usuario pulsa sobre el botón “Añadir Práctica”.
Checkpoint 2	A las práctica ya añadidas, si las hubiera, se les añade un formulario de práctica nueva.
Acción	El usuario rellena los datos de la nueva práctica y pulsa sobre “Ok”.
Checkpoint 3	El sistema guarda la nueva información, que ya no se muestra editable en pantalla.

Test 12: Modificar una práctica

Requisitos testeados	RF-05
Precondiciones	Un usuario registrado se encuentra en la pantalla “Prácticas” de la aplicación.
Acción	El usuario pulsa sobre el botón “Editar Práctica” de la práctica a modificar.
Checkpoint 1	Se muestra el formulario con los datos de la práctica.
Acción	El usuario rellena nuevamente los datos que desee modificar y pulsa “Ok”.
Checkpoint 2	El sistema guarda la nueva información, que ya no se muestra editable en pantalla.

Test 13: Eliminar una práctica

Requisitos testeados	RF-05
Precondiciones	Un usuario registrado se encuentra en la pantalla “Prácticas” de la aplicación.
Acción	El usuario pulsa sobre el botón “Borrar Práctica” de la práctica a eliminar.
Checkpoint 1	Se pregunta al usuario si realmente desea realizar esta acción.
Acción	El usuario pulsa “sí” para borrar la información.
Checkpoint 2	La práctica borrada desaparece de la pantalla “Prácticas”.

Test 14: Crear un examen

Requisitos testeados	RF-06
Precondiciones	Un usuario registrado se encuentra en el menú principal de la aplicación.
Acción	Accede a Exámenes.
Checkpoint 1	El sistema debe mostrar la pantalla de Exámenes.
Acción	El usuario pulsa sobre el botón “Añadir Examen”.
Checkpoint 2	A las exámenes ya añadidos, si los hubiera, se les añade un formulario de examen nuevo.
Acción	El usuario rellena los datos del nuevo examen y pulsa sobre “Ok”.
Checkpoint 3	El sistema guarda la nueva información, que ya no se muestra editable en pantalla.

Test 15: Modificar un examen

Requisitos testeados	RF-06
Precondiciones	Un usuario registrado se encuentra en la pantalla “Exámenes” de la aplicación.
Acción	El usuario pulsa sobre el botón “Editar Examen” del examen a modificar.
Checkpoint 1	Se muestra el formulario con los datos del examen a modificar.
Acción	El usuario rellena nuevamente los datos que desee modificar y pulsa “Ok”.
Checkpoint 2	El sistema guarda la nueva información, que ya no se muestra editable en pantalla.

Test 16: Eliminar un examen

Requisitos testeados	RF-06
Precondiciones	Un usuario registrado se encuentra en la pantalla "Exámenes" de la aplicación.
Acción	El usuario pulsa sobre el botón "Borrar Examen" del examen a eliminar.
Checkpoint 1	Se pregunta al usuario si realmente desea realizar esta acción.
Acción	El usuario pulsa "sí" para borrar la información.
Checkpoint 2	El examen borrado desaparece de la pantalla "Exámenes".

Test 17: Modificar el perfil de usuario

Requisitos testeados	RF-07
Precondiciones	Un usuario registrado se encuentra en la pantalla "Configuración" de la aplicación.
Acción	El usuario añade un nuevo nombre de usuario en blanco y pulsa "Cambiar nombre de usuario".
Checkpoint 1	El sistema muestra el nuevo nombre de usuario en el panel superior de la aplicación
Acción	El usuario añade un nuevo nombre a sus estudios y pulsa "Cambiar estudios".
Checkpoint 2	El sistema muestra el nuevo nombre de los estudios del usuario en el panel superior de la aplicación

Test 18: Eliminar la cuenta de usuario

Requisitos testeados	RF-07
Precondiciones	Un usuario registrado se encuentra en la pantalla "Configuración" de la aplicación.
Acción	El usuario pulsa sobre el botón "Eliminar cuenta de usuario".
Checkpoint 1	Se pregunta al usuario si realmente desea realizar esta acción.

Acción	El usuario pulsa "sí" para borrar su información.
Checkpoint 2	La aplicación vuelve a la pantalla de acceso.

Test 19: Cerrar sesión

Requisitos testeados	RF-08
Precondiciones	Un usuario registrado se encuentra logueado en la aplicación
Acción	El usuario pulsa sobre el botón "Log Off" del panel superior.
Checkpoint 1	La aplicación vuelve a la pantalla de acceso.

3.6.2 Resultados obtenidos

A continuación mostramos una tabla con los resultados obtenidos, después de realizar los test diseñados sobre nuestra aplicación.

TEST	RESULTADO
Test 1	
Checkpoint 1	OK
Test 2	
Checkpoint 1	OK
Test 3	
Checkpoint 1	OK
Test 4	
Checkpoint 1	OK
Test 5	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK

Test 6	
Checkpoint 1	OK
Checkpoint 2	OK
Test 7	
Checkpoint 1	OK
Checkpoint 2	OK
Test 8	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Test 9	
Checkpoint 1	OK
Checkpoint 2	OK
Test 10	
Checkpoint 1	OK
Checkpoint 2	OK
Test 11	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Test 12	
Checkpoint 1	OK
Checkpoint 2	OK
Test 13	
Checkpoint 1	OK
Checkpoint 2	OK
Test 14	

Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Test 15	
Checkpoint 1	OK
Checkpoint 2	OK
Test 16	
Checkpoint 1	OK
Checkpoint 2	OK
Test 17	
Checkpoint 1	OK
Checkpoint 2	OK
Test 18	
Checkpoint 1	OK
Checkpoint 2	OK
Test 19	
Checkpoint 1	OK

CAPÍTULO 4:

Conclusiones

4.1. Conclusiones finales

En este proyecto, se ha realizado una aplicación de escritorio que permita a los estudiantes organizar sus asignaturas, tareas, prácticas y exámenes, permitiendo mediante Cloud Computing acceder a sus datos desde cualquier terminal y permitiendo la sincronización entre éstos. El proyecto se ha realizado de manera satisfactoria, completando los requerimientos funcionales, no funcionales, y cada una de las fases de la Ingenieria del Software que han sido especificadas en el capítulo 3.

A nivel del proyecto, implantar una aplicación de esta magnitud puede facilitar la vida a muchos estudiantes en sus tareas de organización, mejorando su eficacia. Además, con algunas modificaciones y funcionalidades añadidas podría presentarse una aplicación comercial que se pudiera descargar desde la Web.

A nivel de Cloud Computing, quiero subrayar las enormes ventajas que a lo largo de este documento hemos ido conociendo. Esta tecnología puede proporcionarle a la humanidad en las próximas décadas un nivel de vida, de conocimientos y de infraestructuras que ninguna generación humana ha conocido. Por otra parte, todavía le queda mucho camino por recorrer y no recomiendo a ninguna empresa o particular que confíe al 100% en sus servicios, pues siempre es necesario un mínimo de sensatez que nos permita beneficiarnos de esta tecnología teniendo claro hasta donde llega hoy en día.

A nivel personal, y desde el punto de vista que puede tener un estudiante de Ingeniería Informática, tenía ganas de trabajar en un proyecto sobre Cloud Computing por su actualidad, por ampliar mis conocimientos sobre el tema, y sobretodo porque suponía para mí un reto como estudiante realizar un proyecto de esta envergadura. Además, cabe destacar la gran ventaja que me ofrece este proyecto en cuanto a los conocimientos adquiridos si en mi cada vez más cercano futuro laboral me vuelvo a encontrar con esta tecnología. No ha sido un proyecto fácil, pero supone para mí una gran satisfacción personal su realización, así como lo ha sido trabajar con el equipo de profesionales que han supervisado mi proyecto.

4.2. Trabajos futuros

En el futuro, se podría abordar diferentes mejoras sobre el proyecto presentado, tales como:

- Ampliar el acceso a la aplicación mediante programación para teléfonos inteligentes, ofreciendo la posibilidad al usuario de conectarse a la aplicación y acceder a sus asignaturas, tareas, prácticas y exámenes desde su terminal móvil Android o Iphone.

- Proponer soluciones a las desventajas que tiene hoy en día el Cloud Computing (ver apartado 2.7.) y aplicarlas al proyecto. Como ejemplo, se podría llevar a cabo una mejora de la seguridad de la aplicación mediante protocolos, cifrados y algoritmos criptográficos que se ajusten a las particularidades de la nube.

APÉNDICE A:

Manual de Instalación

Este manual indica como llevar a cabo la instalación para poder utilizar la aplicación. Al tratarse de una aplicación de escritorio el proceso es bastante simple. Para poder utilizar la aplicación el usuario solo tendrá que tener instalado en su equipo la Máquina Virtual de Java para permitir el uso de Java en su ordenador. Con una conexión a Internet y la Máquina Virtual de Java el usuario estará preparado para usar la aplicación. Todo el material necesario para instalar y dejar operativa la aplicación se encuentra disponible en el CD que acompaña a esta memoria. El contenido del CD es el siguiente:

- La máquina virtual de Java necesaria para ejecutar la aplicación.
- El archivo .jar propio de la aplicación.
- La memoria del proyecto en formato .pdf.

A.1 Instalación de la Máquina Virtual de Java

La instalación de la Máquina Virtual de Java (JVM) permite ejecutar las aplicaciones realizadas en esta tecnología en cualquier ordenador. Para instalar JMV, el usuario debe dirigirse a la dirección Web: http://www.java.com/es/download/.

Tras descargar el instalador, se procede a su instalación ejecutando el archivo descargado.

Después de algunos minutos, la instalación habrá terminado y el usuario podrá ejecutar la aplicación de este proyecto.

Figura A.1. Descarga de Java Virtual Machine.

A.2 Instalación de la Aplicación

Para el uso de la aplicación simplemente se necesitará insertar en el ordenador el cd que se adjunta a esta memoria, copiar la aplicación, esto es, el archivo ".jar" en alguna carpeta del equipo, y pulsar sobre él para ejecutarlo (figura A.2).

El archivo creará una subcarpeta denominada "Datos", dentro del directorio en el que ejecutemos la aplicación, donde añadirá la base de datos local sobre la que almacenará y gestionará los datos. Una vez hagamos click en el archivo de la aplicación, se nos mostrará la pantalla inicial tal y como vemos en la figura A.3.

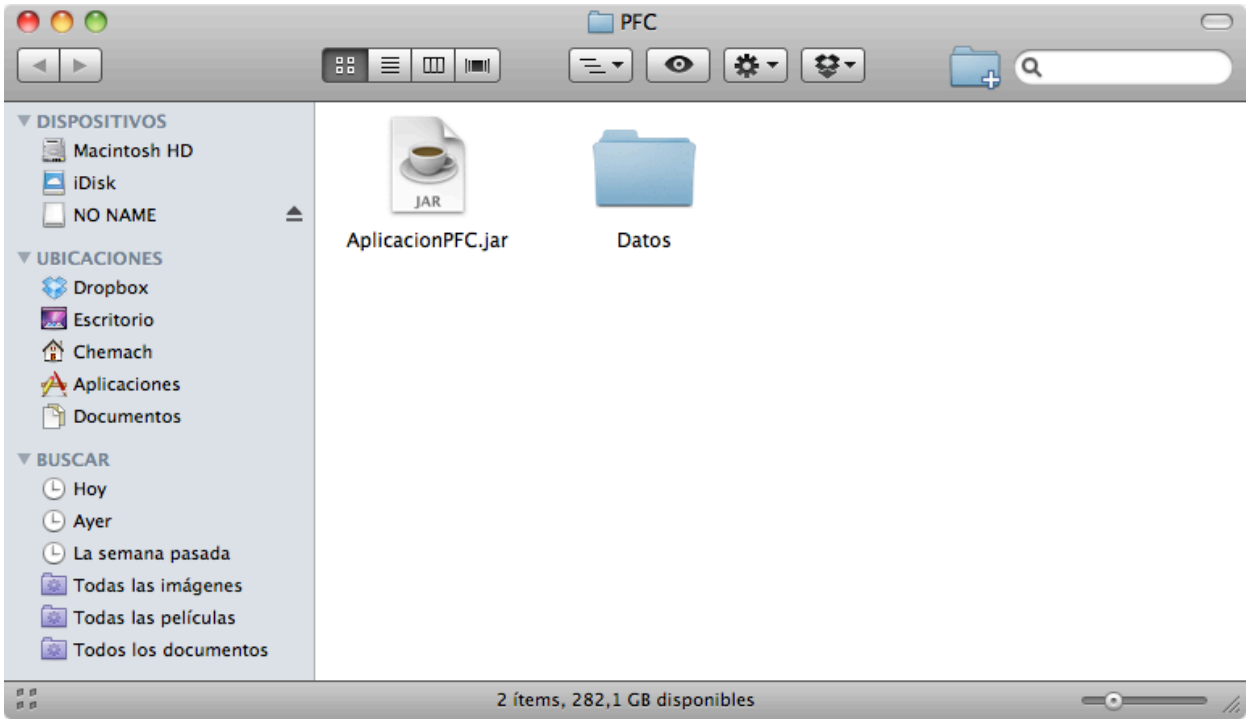

Figura A.2. Copia de la aplicación del cd en el sistema local.

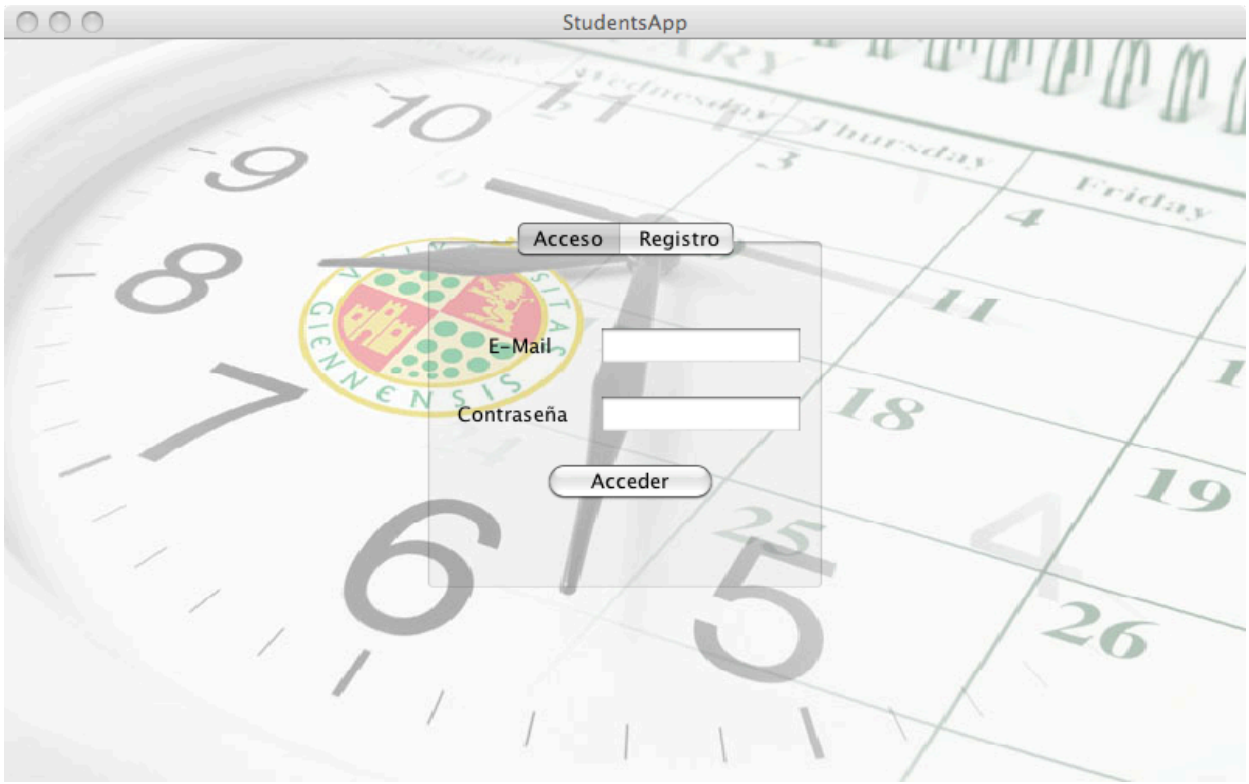

Figura A.3. Pantalla inicial del sistema.

APÉNDICE B:

Manual de Usuario

El manual de usuario explica cómo utilizar la interfaz de usuario y así garantizar su correcto uso y el funcionamiento del sistema. Se describe a continuación cada una de las funciones que el sistema permite, y el manejo del sistema a través de su interfaz.

B.1 Registro en el sistema

Al pulsar sobre el botón para iniciar la aplicación, la primera ventana a la que tendremos acceso será el llamado panel inicial o panel de acceso, tal y como se aprecia en la figura B.1. Sin embargo, al iniciar la aplicación por primera vez, el paso inicial a realizar para poder usar el sistema será registrarse en él con el fin de crear una cuenta de usuario. Para ello, el usuario debe pulsar sobre la pestaña "Registro" (figura B.2), e introducir aquí su email, un nombre de usuario (que después siempre podrá modificar) y su contraseña, y posteriormente pulsar el botón "Registro". En caso de no completar alguno de estos datos, el sistema no le permitirá el registro y notificará al usuario qué datos son necesarios para que el registro se lleve a cabo correctamente.

Una vez realizado el registro se nos muestra un mensaje de que el usuario ha sido registrado, y a continuación accedemos a la pantalla principal de la aplicación (figura B.3).

Es importante destacar que para registrarse en el sistema es necesario disponer de conexión a Internet, pues no pueden crearse usuarios de manera local.

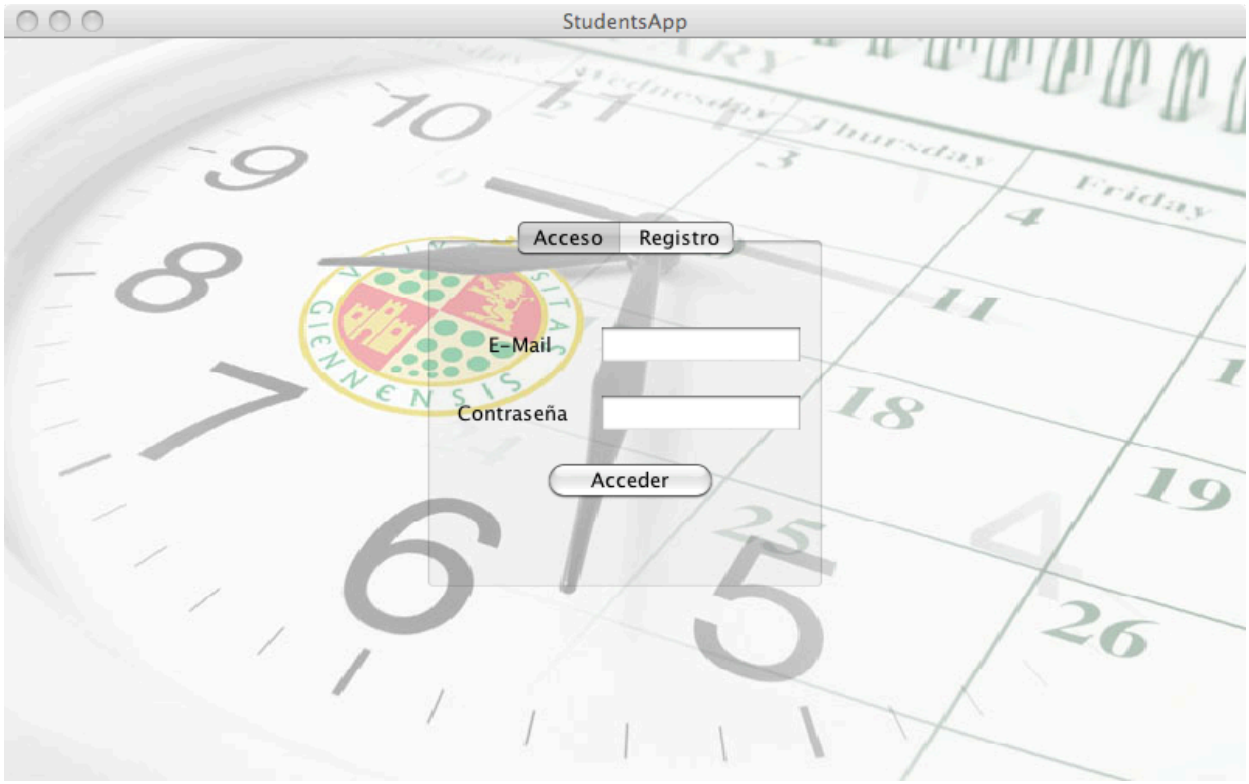

Figura B.1. Pantalla inicial del sistema.

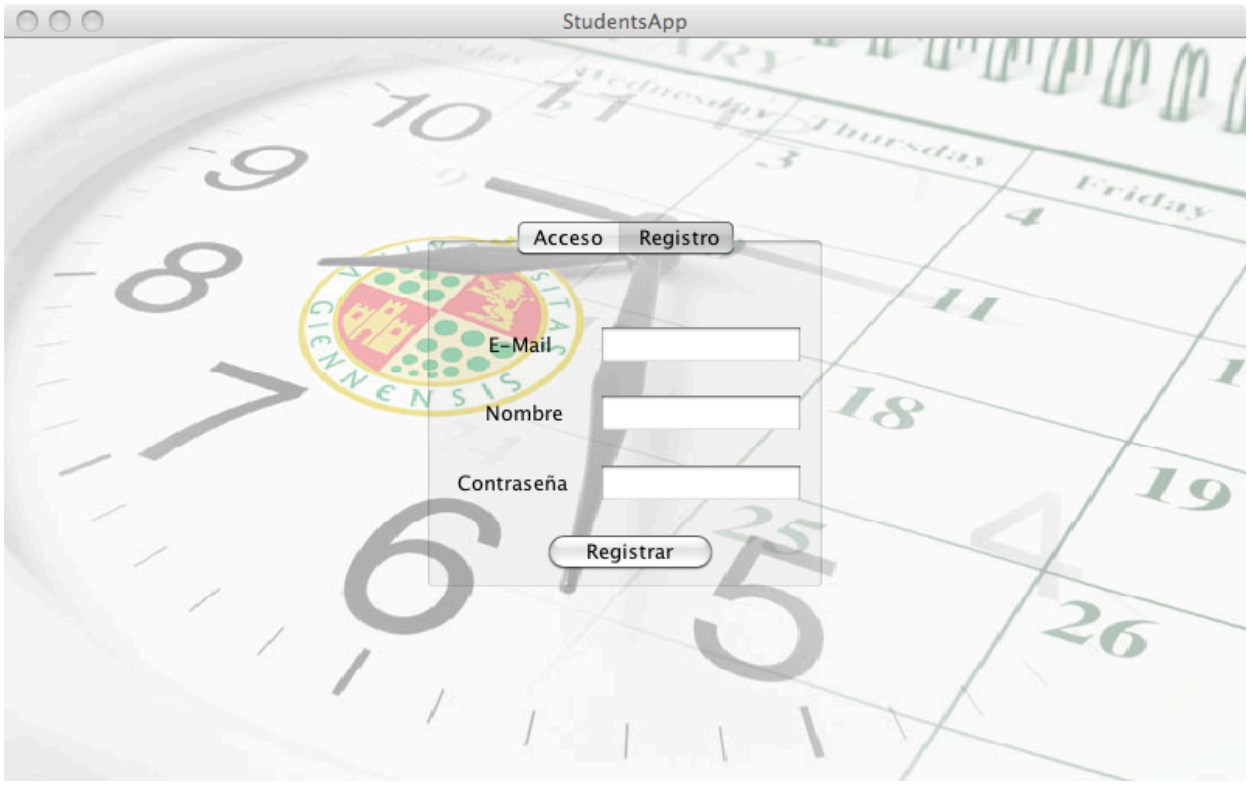

Figura B.2. Registro del sistema.

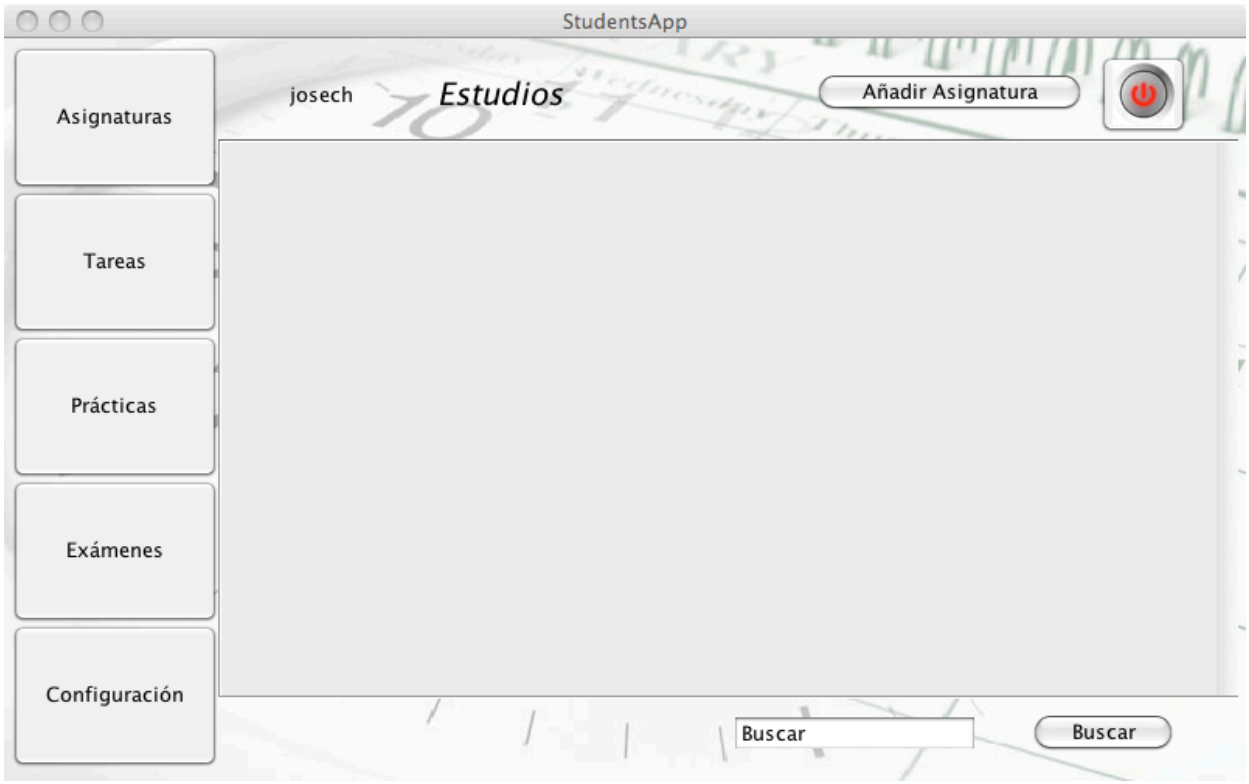

Figura B.3. Pantalla principal del sistema.

B.2 Validación del usuario

Una vez el usuario esté registrado, cada vez que quiera acceder a la aplicación accederá a través del panel de acceso, como se mostró en la figura B.1. Para ello debe escribir su email y contraseña y pulsar sobre el botón de acceso. Existen dos posibilidades a la hora de acceder a la aplicación: de manera local, usando la base de datos local existente, si no se dispone de Internet, y de manera online si se dispone de Internet, esto es, sincronizando los datos con la nube y mostrando los datos actualizados en la pantalla principal (figura B.4). Es importante tener en cuenta que solo se tienen los datos del usuario de manera local si éste ya ha accedido antes a la aplicación desde este terminal. En caso contrario, se mostrará un mensaje indicando que se necesita conexión a Internet para acceder a más información sobre ese usuario.

Figura B.4. Pantalla principal del sistema.

B.3 Gestión de asignaturas, tareas, prácticas y exámenes

La gestión de asignaturas, tareas, prácticas y exámenes es el principal fin de la aplicación y, por consiguiente, la actividad más importante. Para acceder a éstos, se dispone del panel izquierdo, mediante el cuál podemos acceder por separado a Asignaturas, Tareas, Prácticas, Exámenes y Configuración. A continuación, se divide el apartado en: creación, edición y eliminación de asignaturas. Sin embargo, las funciones se realizan de la misma manera tanto para asignaturas como para tareas, prácticas y exámenes.

B.3.1 Creación de asignaturas

Para la creación de asignaturas dentro de la aplicación, se utiliza el botón "Añadir Asignatura" situado en el panel superior a la derecha desde la pantalla "Asignaturas" (ver figura B.5). Una vez pulsado el botón, se muestra en pantalla un nuevo formulario para añadir la información

referente a la nueva asignatura. Cuando se haya añadido dicha información, se deberá pulsar el botón de verificación (icono de verificación), para confirmar la creación de la asignatura.

Figura B.5. Añadir una entidad.

B.3.2 Edición de asignaturas

Para editar alguna de las asignaturas existentes, pulse sobre el botón "Editar" (el icono del lápiz) que existe dentro de la asignatura que desee modificar (ver figura B.6). Así se mostrarán los datos de la asignatura seleccionada en forma de cuadros de texto modificables para editar toda la información referente a ella. Una vez realizados los cambios oportunos, pulse sobre el botón de verificación para confirmar los cambios.

Figura B.6. Editar una entidad.

B.3.3 Eliminación de asignaturas

Para eliminar alguna de las asignaturas creadas, pulse sobre el botón "Eliminar" (el icono X rojo) que existe dentro de la asignatura a eliminar. El sistema le informará que esta asignatura va a ser borrada de la base de datos y le preguntará si desea realmente realizar esta acción, tal y como se muestra en la figura B.7. Si pulsa que no, no se producirá ninguna acción. Pero en caso afirmativo, se eliminará la asignatura del sistema.

B.3.4 Gestión de tareas, prácticas y exámenes

Tal y como se ha indicado anteriormente con las asignaturas, se realiza de igual manera la creación, edición y eliminación de tareas, prácticas y exámenes, con la única diferencia de que se realizará desde sus respectivas pantallas.

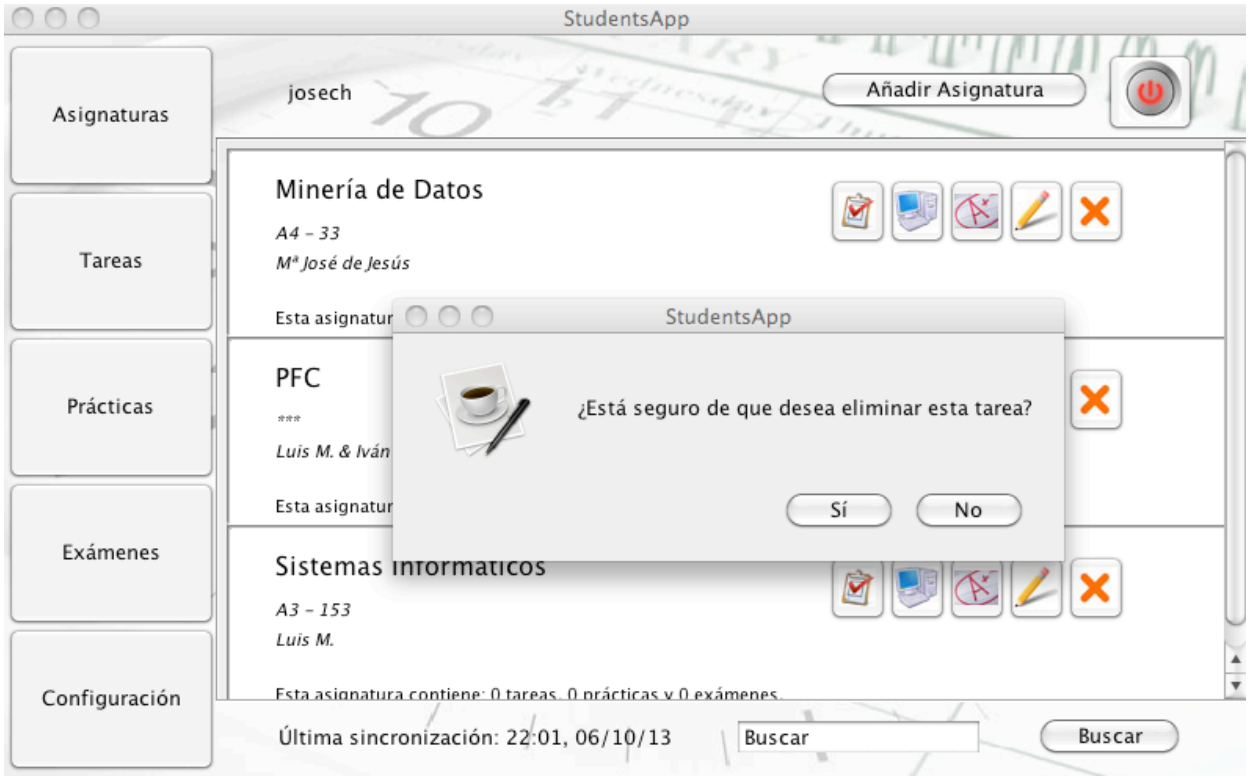

Figura B.7. Eliminar una entidad.

B.3.5 Tareas, prácticas y exámenes asociados a asignaturas

Existe la posibilidad de asociar tareas, prácticas y exámenes a las asignaturas existentes. Para ello, cuando desee crear una entidad (tarea, práctica o examen) asociada a una asignatura, pulse sobre el botón "Asignaturas", y en la asignatura a asociar, haga click sobre el botón "Tareas", "Prácticas" o "Exámenes", según la entidad que desee asociar. Se mostrarán en pantalla todas las entidades de esa categoría asociadas a la asignatura seleccionada (figura B.8). Para crear una nueva entidad, simplemente pulse el botón "Añadir" desde este panel. Su modificación y eliminación puede producirse accediendo de la misma manera, o bien desde los botones "Tareas", "Prácticas" o "Exámenes" que se sitúan en el panel izquierdo. Haciéndolo mediante esta última opción, accedemos a todas las entidades de esa categoría.

Figura B.8. Tarea asociada a una asignatura.

B.4 Modificación del perfil de usuario

Para modificar el perfil de usuario se dispone del panel "Configuración", al que se puede acceder desde el botón "Configuración" situado en el panel izquierdo. Desde este panel, tal y como se muestra en la figura B.9, podrá modificar su nombre de usuario y el nombre de sus estudios. El sistema le pedirá confirmar si desea realizar los cambios en cada modificación. Una vez realizados los cambios, podrá comprobar estos en el panel superior, donde se muestra su nombre de usuario y el nombre de sus estudios.

Figura B.9. Pantalla Configuración del sistema.

B.5 Cerrar sesión

Podrá cerrar sesión siempre que lo desee desde cualquier pantalla de la aplicación mediante el botón superior derecho "Log Off" (icono del botón de apagado) y se volverá a la pantalla inicial de la aplicación (figura B.10). Cada vez que cierre sesión, si existe conexión a Internet, sus datos serán sincronizados a la nube para guardar los últimos cambios.

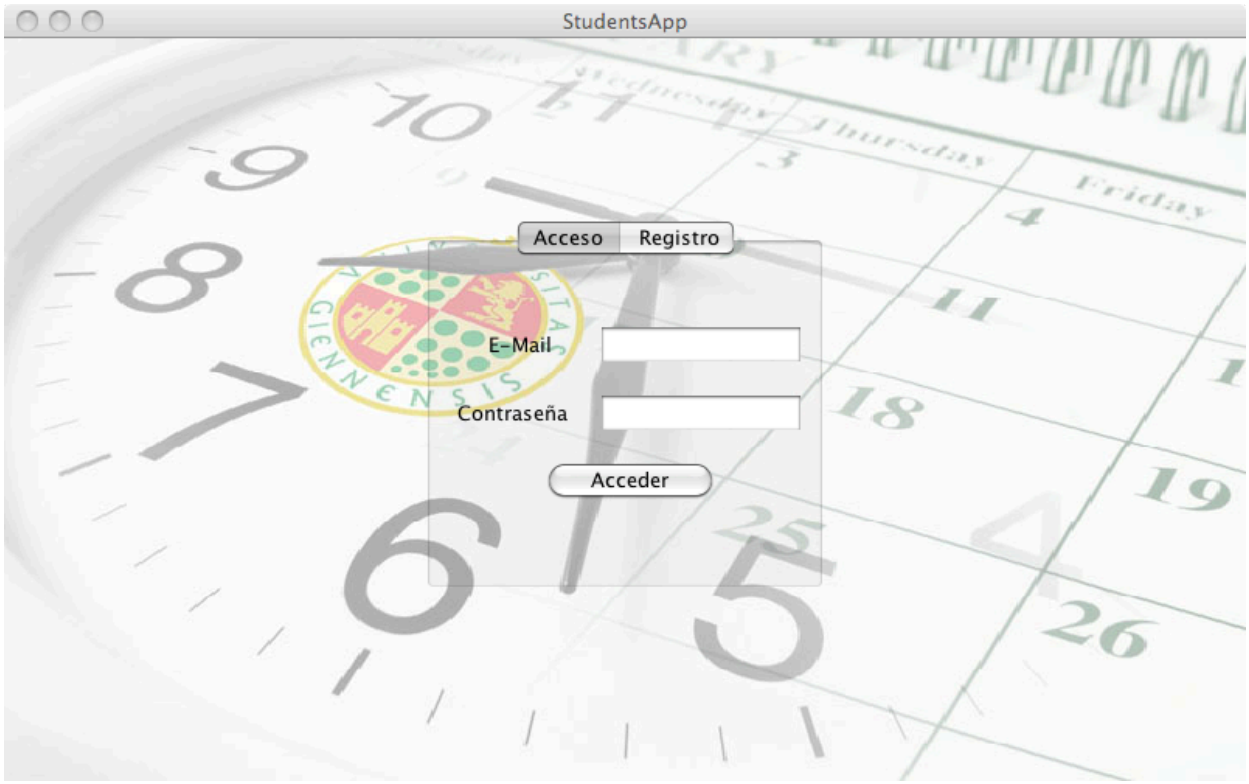

Figura B.10. Pantalla inicial del sistema.

B.6 Eliminar cuenta de usuario

El usuario puede darse de baja en el sistema, accediendo al panel "Configuración" del panel izquierdo y pulsando sobre el botón "Eliminar cuenta de usuario". El sistema le pedirá confirmación para realizar esta acción (ver figura B.11), pues una vez confirmado se procederá al borrado de todos los datos del usuario tanto de la base de datos local como de la nube. Para realizar esta acción, es necesaria la conexión a Internet; de lo contrario el sistema mostrara un mensaje indicando que no se puede llevar a cabo la acción sin una conexión posible.

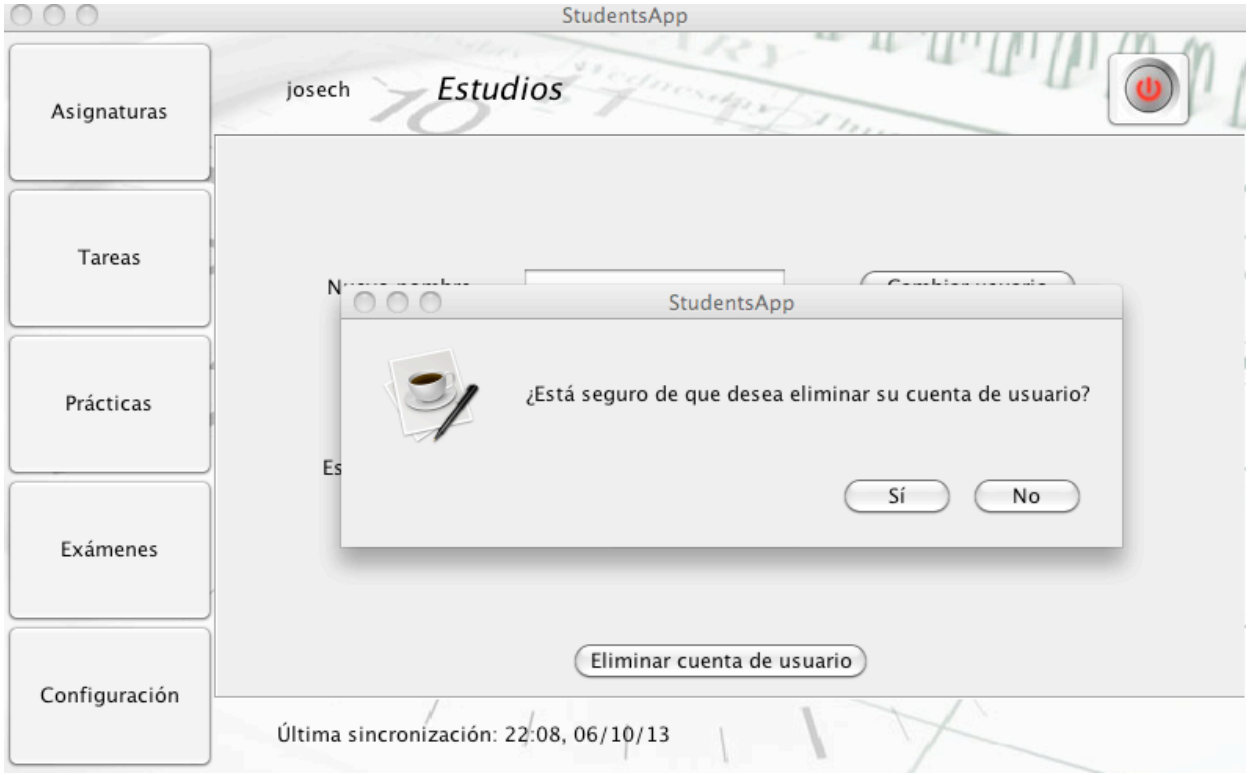

Figura B.11. Pantalla principal del sistema.

APÉNDICE C:

Código fuente del algoritmo

En este apéndice se incluye el código fuente del algoritmo de sincronización utilizado, el cual se expone a continuación:

```
//Se obtienen uno a uno la lista de metadatos correspondiente a los datos del equipo local.
for(Metadatos mLocal : listaMetadatosLocal) {
    //Variable existe a falso para indicar que no se ha encontrado en los metadatos de la nube.
    Boolean existe = false;

    //Se obtienen uno a uno la lista de metadatos correspondiente a los datos de la nube.
    for(Metadatos mGoogle : listaMetadatosGoogle) {
        //Se comprueba que se trata del mismo metadato para ambos.
        if(mLocal.getId() == mGoogle.getId()) {
            //Variable existe a verdadero para indicar que se ha encontrado en la nube.
            existe = true;
            //Se evalúa por la fecha de su última modificación.
            if(mLocal.getFechaUM().after(mGoogle.getFechaUM())) {
                //Última modificación en la aplicación.
                mLocal.setEtiqueta("modToNube");
            } else if(mLocal.getFechaUM().before(mGoogle.getFechaUM())) {
                //Última modificación en la nube.
                mLocal.setEtiqueta("modToApp");
            } else {
                mLocal.setEtiqueta("igual");
            }
        }
    }

    //Si el metadato no existe
    if(!existe) {
        //Se envían los metadatos para comprobar si han sido borrados de la nube o son
        //nuevos.
        String cad = mLocal.getEmailUsuario() + "&" + String.valueOf(mLocal.getId());
```

```
            String o = r.path("rest").path("hola/borrado").queryParam("cadena",cad)
                    .accept(MediaType.TEXT_XML).get(String.class);

            if(o != null) {
                Boolean borrado = new Boolean(o);
                if(borrado) {
                    //El metadato se encuentra borrado de la nube.
                    mLocal.setEtiqueta("borrado");
                } else {
                    //El metadato no existe entre los borrados. Por tanto es nuevo.
                    mLocal.setEtiqueta("nuevo");
                }
            }
        }
}
```

Bibliografía

[1] Sosinsky. Barrie (2011), "Cloud Computing". Ed. Wyley.

[2] Rittinghouse J, Ransome J. "Cloud Computing: Implementation, Management and Security" Ed. CRC Press, 2009.

[3] "Dropbox y su competencia: nuevos servicios en la nube", http://bits.blogs.nytimes.com/2012/07/31/box-and-dropbox-coming-of-age-in-cloud-computing/ (Última fecha de acceso: 10/09/2013)

[4] "Salesforce, Software en la nube", http://www.salesforce.com/es/ (Última fecha de acceso: 15/09/2013)

[5] Jared J. "Going Google: Powerful Tools for 21st Century Learning". Corwin. 2012.

[6] Severance C. "Using Google App Engine". Ed. O'Reilly. 2009.

[7] Declaraciones de Steve Mill. Joyanes L. "La Computación en Nube (Cloud Computing): El nuevo paradigma tecnológico para empresas y organizaciones en la Sociedad del Conocimiento". 2008.

[8] Portnoy M. "Virtualization: Essentials". Ed. Wiley. 2012.

[9] Tidwell D, Snell J, Kulchenko P. "Programming Web Service with SOAP". Ed. O'Reilly. 2001.

[10] Potts S, Kopack M. "Sams teach yourself Web Services". Ed. Sams. 2003.

[11] Webber J, Parastatidis S, Robinson I. "REST in Practice". Ed. O'Reilly. 2010.

[12] Tang B, Sandhu R, Li Q. "Multi-tenancy authorization models for collaborative cloud services". Proceedings of the 2013 International Conference on Collaboration Technologies and Systems, CTS 2013 , art. no. 6567218 , pp. 132-138. 2013.

[13] Voras I, Mihaljevic B, Orlic M y otros. "Evaluating open-source cloud computing solutions". MIPRO 2011 - 34th International Convention on Information and Communication Technology, Electronics and Microelectronics - Proceedings , art. no. 5967051 , pp. 209-214. 2011.

[14] Alvarado M, Agrawal R, Baker Y. "Security mechanisms utilized in a secured cloud infrastructure". Conference Proceedings - IEEE SOUTHEASTCON , art. no. 6567430. 2013.

[15] Bermudez I, Traverso S, Mellia M, Munafo M. "Exploring the cloud from passive measurements: The Amazon AWS case". Proceedings - IEEE INFOCOM , art. no. 6566769 , pp. 230-234. 2013.

[16] Pjanic E, Hasanovic A, Mujcic A, Samuelsen D y otros. "Using cloud infrastructure to support higher education: A case study of managing a course web page with the google sites". International Journal of Emerging Technologies in Learning 8 (1) , pp. 33-37. 2013.

[17] Mell P, Grance T. "The NIST Definition of Cloud Computing". National Institute of Standards and Technology. 2011.

[18] Reese G. "Cloud Application Architectures: Building Applications and Infrastructure in the Cloud". Ed. O'Reilly. 2009.

[19] Vazquez C, Huedo E, Montero, R. "On the use of clouds for grid resource provisioning". Future Generacion Computer Systems - The International Journal of Grid Computing and Science. Volume 27, Issue 5. pp. 600-605. 2011.

[20] Munz F. "Middleware and Cloud Computing: Oracle on Amazon Web Services (AWS), Rackspace Cloud and RightScale". Munz & More Publishing. 2012.

[21] McGrath M. "Unterstanding PaaS". Ed. O'Reilly. 2012.

[22] Krishnan S. "Programming Windows Azure: Programming the Microsoft Cloud". O'Reilly Media. 2010.

[23] Arora A. "Force.com: Tips and Tricks". Packt Publishing. 2013.

[24] Cusumano, M. "Cloud Computing and SaaS as New Computing Platforms". Communications of the ACM 53, vol. 4. pp. 27-29. 2010.

[25] Vaquero L, Rodero-Merino L, Moran D. "Locking the sky: a survey on IaaS cloud security". Computing Vol. 91. Issue 1, pp. 93 - 118. Enero 2011.

[26] Nathani A, Chaudhary S, Somani G. "Policy based resource allocation in IaaS cloud". Future Generation Computer Systems - The International Journal of Grid Computing and Science. Vol. 28, Issue 1, pp. 94 - 103. Enero 2012.

[27] Dans E, "Todo va a cambiar". Editorial Deusto. 2010.

[28] Yang C, Chen B, Chen W. "On Implementation of a KVM IaaS with Monitoring System on Cloud Environments". Communication and Networking, PT I. Vol. 265, pp 300 - 309. 2011.

[29] Opennebula: Infrastructure as a Service: www.opennebula.org (Última fecha de acceso: 21/09/2013)

[30] Beimborn Daniel, Miletzki T, Wenzel S. "Platform as a Service (PaaS)". Business & Information Systems Engineering, Vol. 3. Issue 6, pp 381-384. Diciembre 2011.

[31] Chang W, Hosame A, Sandford J. "Transforming Enterprice Cloud Services". Ed. Springer. 2010.

[32] Rodero-Merino L, Vaquero Luis, Caron E. "Building safe PaaS clouds: A survey on security in multitenant software platforms". Computers & Security. Vol. 31, Issue 1, pp. 96 - 108. Febrero 2012.

[33] Baset S. "Open Source Cloud Technologies". Proceedings of the 3rd ACM Symposium on Cloud Computing, SoCC 2012.

[34] Sanderson D. "Programming Google App Engine". Ed. O'Reilly. 2012.

[35] Roche K, Douglas J. "Beginning Java Google App Engine". Ed. Apress. 2009.

[36] Benlian A, Hess T, Buxmann P. Drivers of SaaS-Adoption - An Empirical Study of Different Application Types. Engineering. Vol. 1, Issue 5, pp. 357-+. Octubre 2009.

[37] Erl T, Puttini R, Mahmood Z. "Cloud Computing: Concepts, Technology & Architecture". Ed. Prentice Hall. 2013.

[38] Proyecto Loon de Google: www.google.com/loon (Última fecha de acceso: 25/09/2013)

[39] Lucena M. "Criptografía y Seguridad en Computadores". Versión 0.8.1. 2010.

[40] Martínez L. "Introducción a SSII". Apuntes de Sistemas Informáticos. Universidad de Jaén.

[41] Pressman, R. "Ingeniería del software: Un enfoque práctico". Ed. McGraw-Hill. 2005.

[42] Ludewig J, Lichter H. "Software Engineering: Grundlagen, Menschen, Prozesse und Techniken". Dpunkt Verlag. 2013.

[43] Sommerville I. "Ingeniería del Software". Ed. Pearson. 2012.

[44] Anderson, S. "Diseño que seduce: Cómo crear webs y aplicaciones atractivas al usuario". Ed. Anaya. 2011.

[45] Larman C. "UML y Patrones". Ed. Prentice Hall. 2010.

[46] Molina A. "Apuntes de Bases de Datos". Universidad de Jaén.

[47] Freeman, E., Sierra, K., Bert Bates, "Head First Design Patterns". Ed. O'Reilly. 2005.

[48] Uppoor S, Flouris M, Bilas A. "Cloud-based Synchronization of Distributed File System Hierarchies". Foundation for Research and Technology - Hellas (Forth). Institute of Computer Science (ICS).