



UNIVERSITAT DE
BARCELONA

Trabajo Final de Grado
GRADO DE INGENIERÍA INFORMÁTICA

Facultad de Matemáticas e Informática
Universidad de Barcelona

**EadaMyCampus 2: Nueva versión de la
aplicación de EADA**

Marlon Orlando Manjarrez Aguirre

Director: Josep Vañó Chic
Realizado en: Departamento de
Matemática e Informática. UB
Barcelona, 19 de enero de 2019

Índice

Índice de ilustraciones	5
Resumen	6
Resum	7
Abstract.....	8
1. Introducción	9
1.1 Contexto	9
1.2 Motivaciones.....	10
1.3 Metodología del proyecto.....	11
1.4 Descripción del proyecto.....	11
1.5 Objetivos.....	12
1.5.1 Objetivos generales	12
1.5.2 Objetivos específicos	12
2. Planificación del TFG.....	13
2.1 Oferta o idea inicial	13
2.2 Primera propuesta	14
2.3 Propuesta final.....	16
2.4 Comparativa entre planificaciones.....	19
3. Análisis de requerimientos.....	20
3.1 Primer análisis	20
3.2 Requisitos funcionales.....	20
3.2.1 Diagramas de casos de uso.....	21
3.2.2 Descripción de los casos de uso textuales.....	25
3.3 Requisitos no funcionales.....	34
4. Diseño	36
4.1 Arquitectura	36
4.2 Templates	36

5.	Implementación	39
5.1	Estructuración del proyecto	39
5.2	Tecnología del proyecto	41
5.3	Autenticación	43
5.4	Diseño adaptativo según rol	45
5.5	Chat privado de EADA (eChat).....	47
5.6	Llamadas asíncronas AJAX.....	48
5.7	Protocolo de comunicación del web Service.....	49
6.	Posibles ampliaciones	52
9.1	Versión de la aplicación para profesorado.....	52
9.2	Versión de la aplicación para gestores de programa.....	53
9.3	Versión de la aplicación para exalumnos	53
7.	Conclusiones	55
8.	Bibliografía.....	56

Índice de ilustraciones

Ilustración 1 Desarrollo en cascada	11
Ilustración 2 Diagrama de Gantt de la planificación inicial	15
Ilustración 3 Etapas definidas en la primera propuesta de planificación	15
Ilustración 4 Diagrama de Gantt de la planificación final	17
Ilustración 5 Etapas definidas en la propuesta final de la planificación	18
Ilustración 6 Casos de uso de alumno	21
Ilustración 7 Casos de uso de profesor.....	22
Ilustración 8 Casos de uso de gestor de programa	23
Ilustración 9 Casos de uso de exalumno	24
Ilustración 10 Estructura Apache Cordova ^[2]	39
Ilustración 11 Estructura Carpeta www.....	40
Ilustración 12 Logos de algunas de las tecnologías utilizadas.....	42
Ilustración 13 Vista de la herramienta GitKraken	42
Ilustración 14 Gráfico de inicio de sesión.....	43
Ilustración 15 Página de Login.....	44
Ilustración 16 A la izquierda home de rol alumno y a la derecha home de otro rol	45
Ilustración 17 A la izquierda menú de rol alumno y a la derecha menú de otro rol	46
Ilustración 18 A la izquierda conversaciones activas del chat y a la derecha una conversación.....	47
Ilustración 19 Spinner de carga	48
Ilustración 20 Protocolo REST	49
Ilustración 21 Métodos de la clase EadaAppCampusService.java	51

Resumen

Este proyecto se encarga del desarrollo e implementación de la segunda versión de la aplicación EadaMyCampus cuya propietaria es la empresa EADA Business School Barcelona^[1] y que ha aceptado colaborar en este trabajo final de grado.

EadaMyCampus es una aplicación para Android e IOS que permitirá el acceso a los participantes, assistants y directores de programa al Campus Virtual de EADA, facilitando la comunicación entre los actores.

Su principal objetivo es proveer un servicio que permita la comunicación ágil entre los participantes y EADA (assistants y directores de programa), mejorando y complementando la experiencia de cursar un programa. También se requiere de una aplicación rápida y funcional que sea una alternativa simplificada a la versión Web del Campus Virtual.

El entorno de desarrollo de la aplicación es Apache Cordova^[2], este entorno nos permite construir aplicaciones para dispositivos móviles utilizando Css3, Html5 y Javascript sustituyendo API's específicas de cada plataforma (Android, IOS o Windows phone). La aplicación se nutre de un Webservice (REST-Full)^[3] implementado en JAVA cuya comunicación será mediante peticiones con AJAX. También se utilizará JQuery y Bootstrap. En el back office tenemos un EJB Container (Jboss^[7]) y un Servlet Engine (Tomcat^[8]). Por último, un MySql para la gestión de base de datos.

Resum

Aquest projecte s'encarrega del desenvolupament i implementació de la segona versió de l'aplicació EadaMyCampus la propietària de la qual és l'empresa EADA Business School Barcelona^[1] i que ha acceptat col·laborar en aquest treball final de grau.

EadaMyCampus és una aplicació per a Android i IOS que permetrà l'accés als participants, assistents i directors de programa al Campus Virtual de EADA, facilitant la comunicació entre els actors.

El seu principal objectiu és proveir un servei que permeti la comunicació àgil entre els participants i EADA (assistents i directors de programa), millorant i complementant l'experiència de cursar un programa. També es requereix d'una aplicació ràpida i funcional que sigui una alternativa simplificada a la versió Web del Campus Virtual.

L'entorn de desenvolupament de l'aplicació és Apache Cordova^[2], aquest entorn ens permet construir aplicacions per a dispositius mòbils utilitzant Css3, Html5 i Javascript substituint API's específiques de cada plataforma (Android, IOS o Windows Phone). L'aplicació es nodreix d'un Webservice (Rest-Full) ^[3] implementat a JAVA la comunicació del qual serà mitjançant peticions amb AJAX. També s'utilitzarà JQuery i Bootstrap. En el back office tenim un EJB Container (Jboss^[7]) i un Servlet Engine (Tomcat^[8]). Finalment, un MySql per a la gestió de base de dades.

Abstract

This project is responsible for the development and implementation of the second version of the EadaMyCampus application whose owner is the company EADA Business School Barcelona^[1] and has agreed to collaborate in this final degree project.

EadaMyCampus is an application for Android and IOS that will allow access to participants, assistants and program directors to the EADA Virtual Campus, easing communication between them.

The main objective for the application is to provide a service that allows agile communication between participants and EADA (assistants and program directors), improving and complementing the experience of studying a program. Additionally, a fast and functional application that is a simplified alternative to the Web version of the Virtual Campus is also required.

The application development environment is Apache Cordova^[2], this environment allows us to build applications for mobile devices using Css3, Html5 and Javascript replacing specific APIs of each platform (Android, IOS or Windows phone) with a single one. The application draws on a Webservice (REST-Full)^[3] implemented in JAVA whose communication will be through requests with AJAX. JQuery and Bootstrap are also part of the technology stack. In the back office there is an EJB Container (Jboss^[7]) and a Servlet Engine (Tomcat^[8]). Finally, a MySql for database management.

1. Introducción

1.1 Contexto

EADA es una institución independiente y sin ánimo de lucro, fundada en 1957 y constituida como una fundación privada universitaria en 1984, dedicada a la formación, investigación aplicada y transferencia de conocimiento al mundo empresarial y a la sociedad en un contexto internacional.

Según la lista de la publicación británica Financial Times de diciembre de 2018, EADA se encuentra entre las 30 mejores escuelas de negocios de Europa. Por sus aulas han pasado más de 120.000 alumnos de los cuales el 90% provienen del extranjero, en su mayoría de Europa y Latinoamérica, mientras que poco más de un 22% proceden de América del Norte y Asia-Oceanía, los dos escenarios en los que quiere captar talento en los próximos años.

Con la intención de seguir creciendo, EADA decidió invertir una cuantiosa cantidad de dinero en obras de renovación de sus instalaciones en Barcelona, que comenzaron en 2016 y terminaron en el verano de 2019. Jordi Pursals, presidente de la Fundación EADA, afirmó que la renovación del edificio supuso una fuerte inversión, pero necesaria para que la escuela sea reconocida como lo que académicamente es. En vista de esta inversión y del cambio de imagen, se buscó dar un paso más y actualizar tanto su campus virtual como las herramientas internas que son utilizadas a diario por el personal de EADA.

Con los objetivos marcados, se comenzó con el desarrollo de herramientas de las que se nutre EADA:

- **Atlas:** Es una intranet colaborativa de EADA que es utilizada, únicamente, por el personal de EADA. En esta herramienta es donde se crean los cursos, se matriculan usuarios, se gestiona la facturación, etc... Esta herramienta es el corazón de EADA y es de donde se obtiene la información que nutre el Campus virtual.
- **Irene:** Es una intranet colaborativa de EADA, utilizada fundamentalmente en la comunicación interna. Es el eje del trabajo transversal en EADA.

- **Campus Virtual:** Aquí es donde conviven alumnos, profesores y gestores de programa. En este espacio es donde se desarrolla la actividad académica de EADA. En esta plataforma los alumnos pueden acceder a los contenidos que disponen los cursos en los que se han matriculado.
- **EadaMyCampus V2:** Segunda versión de la aplicación de EADA.

Después de un largo tiempo, este periodo de actualización comienza a ver su fin. Tanto Atlas como Irene ya están plenamente integradas en el trabajo diario de EADA y en breves, Campus Virtual de la mano de su aplicación, serán presentados por todo lo alto.

1.2 Motivaciones

Para comenzar, he de decir que la idea para realizar este Trabajo final de Grado surgió casi por casualidad. Llevaba dos años trabajando como desarrollador de software en una empresa que usaba Moodle para crear cursos de formación para empresas. Me encargaba de desarrollar nuevos módulos para Moodle tanto a nivel interno como para comercializarlos y esto hizo que ganara mucha experiencia en el funcionamiento de dicho software.

Gracias a mi experiencia con dicha plataforma, EADA se puso en contacto conmigo ofreciéndome un puesto de trabajo ya que precisaban de una persona especializada en Moodle para completar la actualización de su Campus Virtual. Una vez en EADA y después de haber transcurrido un par de meses, el director del departamento sabía que tenía que realizar mi TFG y me propuso la idea de actualizar la aplicación.

Era una idea en la que las dos partes salíamos beneficiadas. La empresa cubriría una necesidad y yo enriquecería mis conocimientos aprendiendo a utilizar nuevas herramientas. Los motivos por los que decidí realizar este proyecto son los siguientes:

- En primer lugar, teniendo en cuenta como está el mercado de trabajo y que cada vez hay más empresas que se dedican al desarrollo de aplicaciones para móviles, el hecho de ganar experiencia en este tema es un punto muy a tener en cuenta ya que me enriquece a nivel profesional.

- En segundo lugar, formar parte de un proyecto tan ambicioso que no busca una solución para el presente, sino que, se plantea escenarios en el futuro para seguir evolucionando y ofrecer un producto muy potente.
- En tercer lugar, aumentar mis capacidades para poder trabajar en equipo.
- Por último, conocer la herramienta de desarrollo de aplicaciones Apache Cordova^[2] para desarrollar una aplicación multiplataforma utilizando HTML5, CSS3, JavaScript y JQuery.

1.3 Metodología del proyecto

El desarrollo de este proyecto se ha llevado a cabo de forma secuencial, es decir, se ha utilizado el desarrollo en cascada^[11]. Este enfoque metodológico ordena rigurosamente las etapas del proceso de desarrollo de software, de tal forma que el inicio de la siguiente etapa no puede comenzar sin haber finalizado la etapa anterior.

Al finalizar cada etapa, se realiza la revisión final para valorar si el proyecto puede avanzar o de lo contrario, si se ha identificado cualquier error de diseño, proceder al rediseño y nueva programación del código afectado.

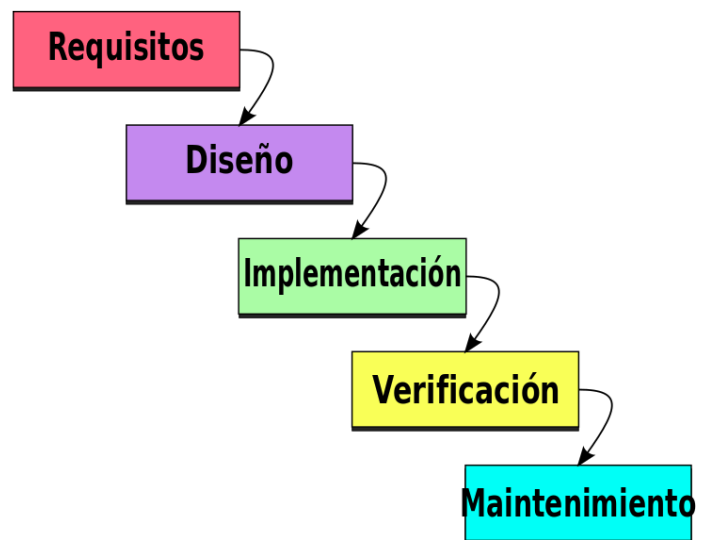


Ilustración 1 Desarrollo en cascada

1.4 Descripción del proyecto

Este proyecto consiste en el desarrollo de una aplicación móvil para la empresa EADA Business School Barcelona^[1] que mejore la antigua versión de la aplicación de la que disponían.

Esta aplicación, busca ofrecer una herramienta para que los alumnos puedan acceder desde sus dispositivos móviles a los contenidos de los cursos que están impartiendo en EADA

de forma rápida y sencilla. Además, se busca implementar un chat privado para suplantar la manera en la que actualmente se comunican los alumnos con el personal de EADA y viceversa.

Dicha aplicación será desarrollada utilizando el entorno de desarrollo Apache Cordova^[2] y utilizará un RESTful Web Service para obtener la información necesaria. Gracias al entorno mencionado anteriormente, podremos crear una aplicación que sea compatible tanto para dispositivos Android como IOS sin necesidad de crear diferentes versiones de código.

1.5 Objetivos

1.5.1 Objetivos generales

El objetivo de este proyecto es el desarrollo de la segunda versión de la aplicación de EADA para dispositivos Android e IOS. También se fijó como objetivo crear una aplicación propia, ya que la versión anterior había sido desarrollada por una empresa externa. Esta aplicación, respetará las funcionalidades que ofrecía la versión anterior y estará preparada para futuras implementaciones.

1.5.2 Objetivos específicos

Además de los objetivos generales, los objetivos específicos de este proyecto pretenden profundizar en el aprendizaje en el desarrollo de aplicaciones para dispositivos Android e IOS y en la utilización de nuevas tecnologías. Algunas de las tecnologías que se tratarán son las siguientes:

- RESTful Web Service
- Apache Cordova^[2]: Entorno de desarrollo para aplicaciones móvil
- AJAX (Asynchronous Javascript And Xml)
- JBOSS^[7]: Servidor de aplicaciones Java EE
- TOMCAT^[8]: Contenedor de servlets^[9]
- Docker: Contenedor portable para aplicaciones software
- JQuery: Biblioteca multiplataforma de JavaScript
- Lenguajes de programación JAVA, CSS3, HTML5, JavaScript

2. Planificación del TFG

Para la realización de este Trabajo final de Grado, han sido necesarias varias reuniones periódicas con Marco Peña, director del Departamento de Bits y Miguel Galindo, responsable del proyecto y desarrollador de software. Además, se han requerido diversas reuniones con todos los departamentos de la universidad de EADA para obtener un feedback y continuar con el desarrollo del proyecto.

2.1 Oferta o idea inicial

La propuesta inicial de este Trabajo final de Grado está basada en el desarrollo de una nueva versión de la aplicación que dispone EADA. El desarrollo de esta nueva versión vino marcada por la actualización a una versión más moderna de su campus virtual que había quedado anticuado.

La idea es mantener todas las funcionalidades que se ofrecían en la primera versión de la APP y, además, proporcionar nuevas mejoras con la intención de que esta nueva versión sea más utilizada. Algunas de las funcionalidades que se mantendrán son:

- **Consultar próximas sesiones.**
- **Consultar últimas notas.**
- **Consultar asignaturas en curso.**
- **Consultar Notificaciones.**
- **Acceder al campus.**
- **Reportar incidencias.**

A pesar de que la aplicación disponía de mucho potencial, no se pudo sacar el máximo rendimiento en la primera versión ya que fue desarrollada por una empresa externa que se limitó a cumplir con lo acordado, provocando un resultado muy pobre y poco atractivo. Para poder recuperar dicho atractivo, las mejoras que incluyen esta nueva versión son:

- **Cambio total del diseño**
- **Chat privado**
- **Navegación por las aulas Moodle**
- **Modificación de los datos del perfil**

Cabe destacar que esta versión quedará preparada para futuras mejoras en cuanto a los diferentes perfiles de usuario que existen en EADA ya que la primera versión estaba enfocada, únicamente, a los alumnos.

2.2 Primera propuesta

Después de analizar la oferta o idea inicial, se presentó una posible estructura de planificación en la que se dio prioridad a tener una estructura en la que las funcionalidades de la primera versión de la aplicación funcionen correctamente. Después, se incluiría el chat a la aplicación y se crearía una versión de prueba.

De esta forma, la primera propuesta de planificación que se decidió en una reunión del departamento fue la siguiente:

- Preparación del entorno de desarrollo de Apache Cordova^[2] en un Docker. Decidimos utilizar dicha tecnología para facilitar la instalación de los paquetes necesarios sin afectar a la máquina local.
- Preparación de una maqueta de la aplicación, por parte del diseñador gráfico, utilizando el tema elegido por EADA.
- Preparar el entorno de trabajo duplicando el RESTful Web Service en la máquina local para poder realizar pruebas en la propia máquina de desarrollo.
- Utilizando la maqueta, comenzar a implementar las diferentes funcionalidades de la aplicación con el fin de crear un prototipo para su evaluación final.

Con esta propuesta, nos aseguramos realizar las diferentes tareas en un orden adecuado para lograr un producto estable y que nos de pie a reaccionar con los posibles inconvenientes que surjan durante el desarrollo. Además, la mayoría de las tareas podían ser realizadas de forma simultánea y esto permitía un desarrollo más ágil.

Con esta primera propuesta, se establecieron unos tiempos aproximados en su realización y se creó el Diagrama de Grantt con la planificación.

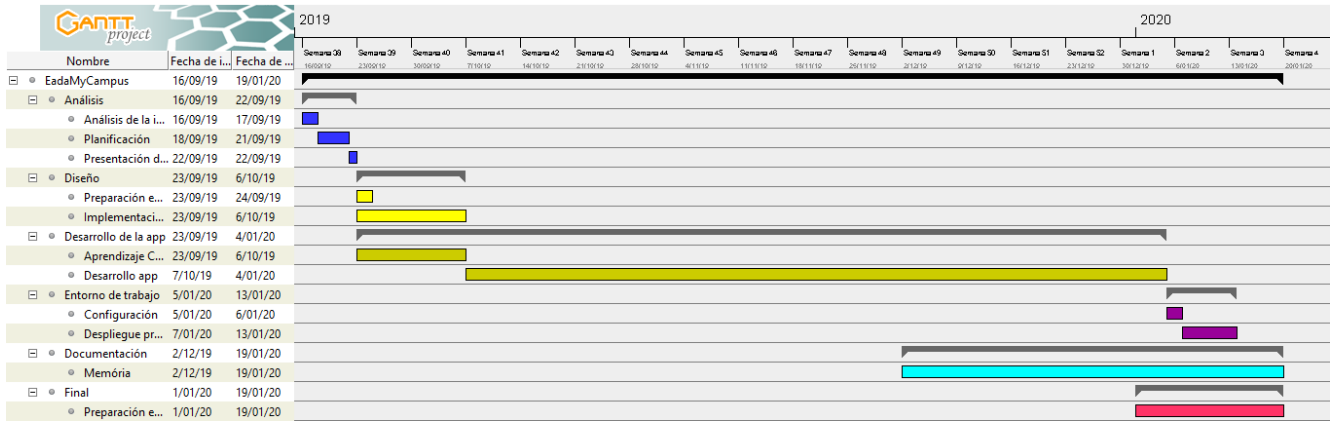


Ilustración 2 Diagrama de Gantt de la planificación inicial

La ilustración 2 muestra el diagrama de Gantt con la planificación ideal del proyecto. La fecha establecida como inicio del proyecto es el día en que se comenzó la colaboración con la empresa y la fecha final es la fecha límite para la entrega de la memoria.

Nombre	Fecha de inicio	Fecha de...
EadaMyCampus	16/09/19	19/01/20
Análisis	16/09/19	22/09/19
Análisis de la idea	16/09/19	17/09/19
Planificación	18/09/19	21/09/19
Presentación de la propuesta	22/09/19	22/09/19
Diseño	23/09/19	6/10/19
Preparación entorno de trabajo	23/09/19	24/09/19
Implementación de la maqueta	23/09/19	6/10/19
Desarrollo de la app	23/09/19	4/01/20
Aprendizaje Cordova	23/09/19	6/10/19
Desarrollo app	7/10/19	4/01/20
Entorno de trabajo	5/01/20	13/01/20
Configuración	5/01/20	6/01/20
Despliegue proyecto	7/01/20	13/01/20
Documentación	2/12/19	19/01/20
Memoria	2/12/19	19/01/20
Final	1/01/20	19/01/20
Preparación entrega	1/01/20	19/01/20

Ilustración 3 Etapas definidas en la primera propuesta de planificación

La ilustración 3 muestra las etapas definidas en la primera propuesta de planificación de este proyecto.

2.3 Propuesta final

Después de realizar un análisis exhaustivo a la primera propuesta, se llegó a la conclusión de que podríamos incluir más mejoras para una primera entrega de la aplicación. Así pues, se decidió incluir una funcionalidad en la aplicación que permitiría realizar el check-in a las aulas.

Una de las muchas obligaciones de los alumnos que están matriculados en cursos de EADA es la obligación de asistir a clase. Una falta de asistencia penaliza la nota final de la asignatura y para poder llevar un control, cada alumno dispone de una tarjeta proporcionada por el departamento de BITS que sirve para entrar a las aulas que le corresponden. El alumno acerca la tarjeta a un lector que hay al lado de la puerta del aula y mediante un web service se registra su asistencia a clase. Dicha tecnología está proporcionada por SONY y en un futuro se espera poder sustituirla por una propia.

Esta tecnología lleva mucho tiempo en uso, aunque tiene un gran problema. Los alumnos acostumbran a olvidar su tarjeta con la consecuencia de que no pueden acceder al aula. Para solucionar el problema, se ponen en contacto con los gestores de programa y estos, después de confirmar que el alumno haya asistido, introducen el chek-in de forma manual.

Una vez identificado este continuo problema, en el departamento, llegamos a la conclusión que la aplicación nos brindaba una oportunidad para poder paliar dicha incidencia. A la espera de poder implementar nuestro propio sistema para acceder a las aulas, se propuso una idea pasajera que consiste en colocar códigos QR fijos en la entrada de cada aula para que, utilizando el lector de QR de la aplicación, el alumno pueda acceder al aula sin problema. Así, si el alumno ha olvidado su tarjeta, es menos probable que haya olvidado su móvil y por ende podrá acceder.

También se decidió comprar un tema para la aplicación debido a las facilidades y el ahorro de trabajo que esto nos representaba.

Así pues, después de confirmar los cambios a realizar, la planificación final quedaría de la siguiente forma:

- Preparación del entorno de desarrollo de Apache Cordova^[2] en un Docker. Decidimos utilizar dicha tecnología para facilitar la instalación de los paquetes necesarios sin afectar a la máquina local.
- Preparación de una maqueta de la aplicación, por parte del diseñador gráfico, utilizando el tema comprado por EADA.
- Preparar el entorno de trabajo duplicando el RESTful Web Service en la máquina local para poder realizar pruebas en la propia máquina de desarrollo.
- Utilizando la maqueta, comenzar a implementar las diferentes funcionalidades de la aplicación, añadiendo la nueva funcionalidad, con el fin de crear un prototipo para su evaluación final.

Teniendo en cuenta que el desarrollo de esta aplicación es más que una simple aplicación que muestre contenidos, nos pareció buena idea incluir las mejoras y cambiar un poco la planificación inicial. De este modo, el diagrama de Gantt queda de la siguiente forma:

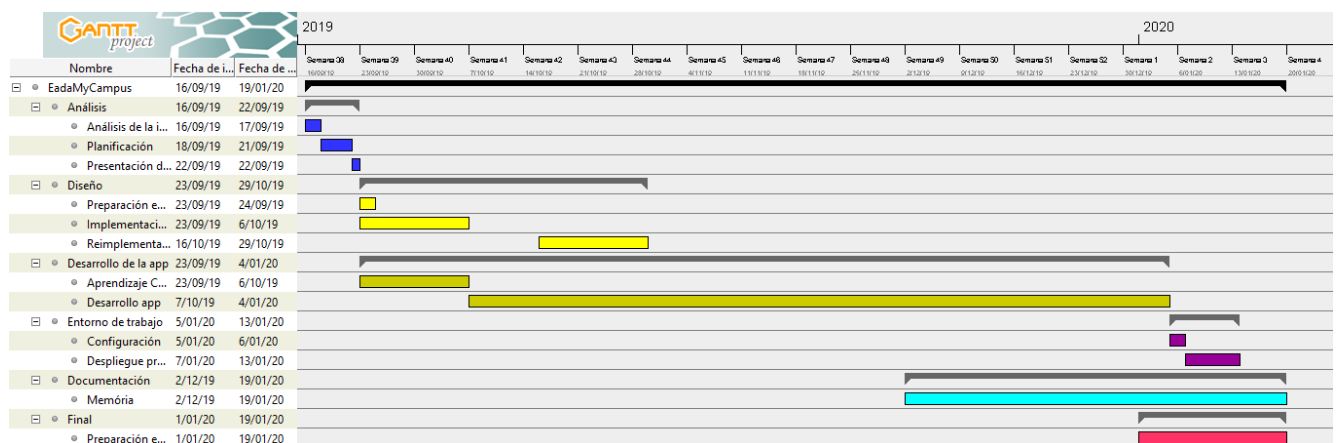
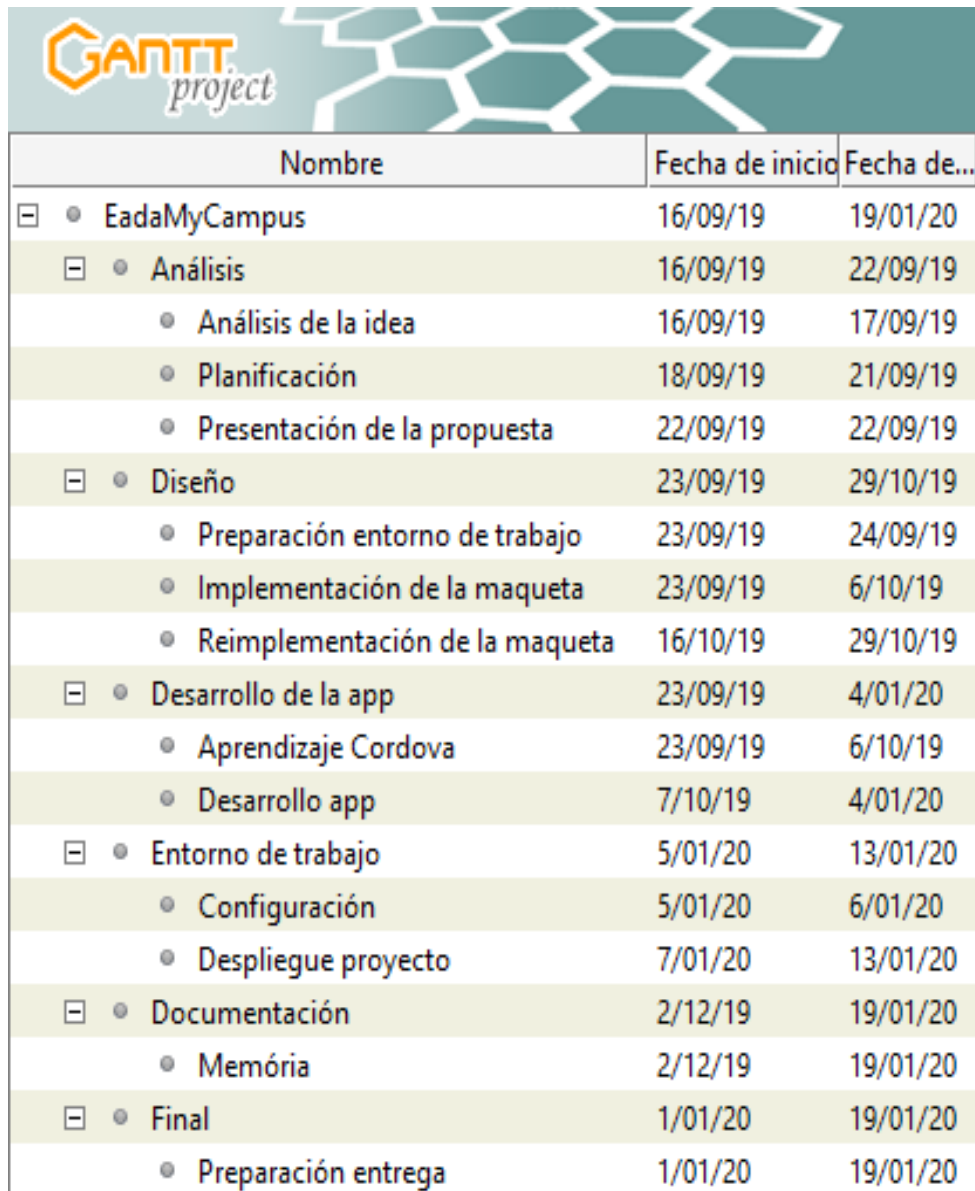


Ilustración 4 Diagrama de Gantt de la planificación final

La ilustración 4 muestra el diagrama de Gantt con la planificación ideal definitiva del proyecto. La fecha establecida como inicio del proyecto es el día en que se comenzó la colaboración con la empresa y la fecha final es la fecha límite para la entrega de la memoria. También se incluyeron los cambios acordados en las reuniones a posterior de la primera propuesta.



Nombre		Fecha de inicio	Fecha de...
☐	• EadaMyCampus	16/09/19	19/01/20
☐	• Análisis	16/09/19	22/09/19
	• Análisis de la idea	16/09/19	17/09/19
	• Planificación	18/09/19	21/09/19
	• Presentación de la propuesta	22/09/19	22/09/19
☐	• Diseño	23/09/19	29/10/19
	• Preparación entorno de trabajo	23/09/19	24/09/19
	• Implementación de la maqueta	23/09/19	6/10/19
	• Reimplementación de la maqueta	16/10/19	29/10/19
☐	• Desarrollo de la app	23/09/19	4/01/20
	• Aprendizaje Cordova	23/09/19	6/10/19
	• Desarrollo app	7/10/19	4/01/20
☐	• Entorno de trabajo	5/01/20	13/01/20
	• Configuración	5/01/20	6/01/20
	• Despliegue proyecto	7/01/20	13/01/20
☐	• Documentación	2/12/19	19/01/20
	• Memoria	2/12/19	19/01/20
☐	• Final	1/01/20	19/01/20
	• Preparación entrega	1/01/20	19/01/20

Ilustración 5 Etapas definidas en la propuesta final de la planificación

La ilustración 5 muestra las etapas definidas en la propuesta final de planificación de este proyecto.

2.4 Comparativa entre planificaciones

Como podemos observar, no hay mucha diferencia entre la planificación inicial y la planificación final. El único punto a destacar fue la nueva funcionalidad que se añadió para cubrir un leve, pero continuo, problema en el día a día de EADA.

Gracias a que la nueva petición se identificó a tiempo, se pudo incluir en la planificación sin afectar a los tiempos de entrega y seguir con el rumbo marcado. El único inconveniente fue el cambio de diseño de la aplicación ya que el diseñador tenía la primera maqueta lista y presentada. Por parte de los desarrolladores, no hubo mayor inconveniente, es cierto que ya se comenzó a implantar el diseño en el desarrollo, pero el cambio no representaba mayor esfuerzo.

Valoramos, también, la buena planificación inicial del proyecto ya que no nos vimos saturados por los inconvenientes y pudimos continuar trabajando a un ritmo aceptable haciéndonos cargo de este y de otros proyectos que están activos en la empresa. He de añadir que por motivos ajenos al proyecto nos encontramos en momentos en los que pensábamos que el trabajo nos desbordaría, sin embargo, conseguimos sacar adelante cada uno de los frentes.

3. Análisis de requerimientos

3.1 Primer análisis

Desde un principio, la idea principal era crear una aplicación que mejore en prestaciones a la antigua versión de la aplicación manteniendo las funcionalidades que se ofrecían y añadiendo nuevas.

Debido al tiempo del que se dispone, se decidió crear una primera versión enfocada, únicamente, a las funcionalidades que un alumno puede realizar dejando para un futuro los demás perfiles de usuarios que existen en EADA. Por este motivo, aunque cualquier persona que disponga de un perfil de EADA pueda acceder a la aplicación, solo los alumnos dispondrán de más funcionalidades, así como más contenido. Al resto de perfiles solo se les mostrará noticias relevantes y notificaciones. Sin embargo, todos los perfiles tendrán acceso al chat privado de EADA.

Gracias a este planteamiento, dejamos preparado el entorno para el desarrollo de las funcionalidades para los distintos perfiles de usuarios. Por el momento, en los siguientes apartados, se procede a explicar los requisitos de esta versión de la aplicación.

3.2 Requisitos funcionales

Como hemos visto anteriormente, en EADA existen varios perfiles de usuarios y aunque la primera versión de la aplicación, en primera instancia esté enfocada en los alumnos, los otros perfiles también disponen de algunas funcionalidades. Dicho esto, los tipos de usuarios son los siguientes:

- **Alumnos**
- **Exalumnos**
- **Profesores**
- **Gestores de programa**

A continuación, se describen los casos de uso de cada uno de los actores:

3.2.1 Diagramas de casos de uso

Alumnos

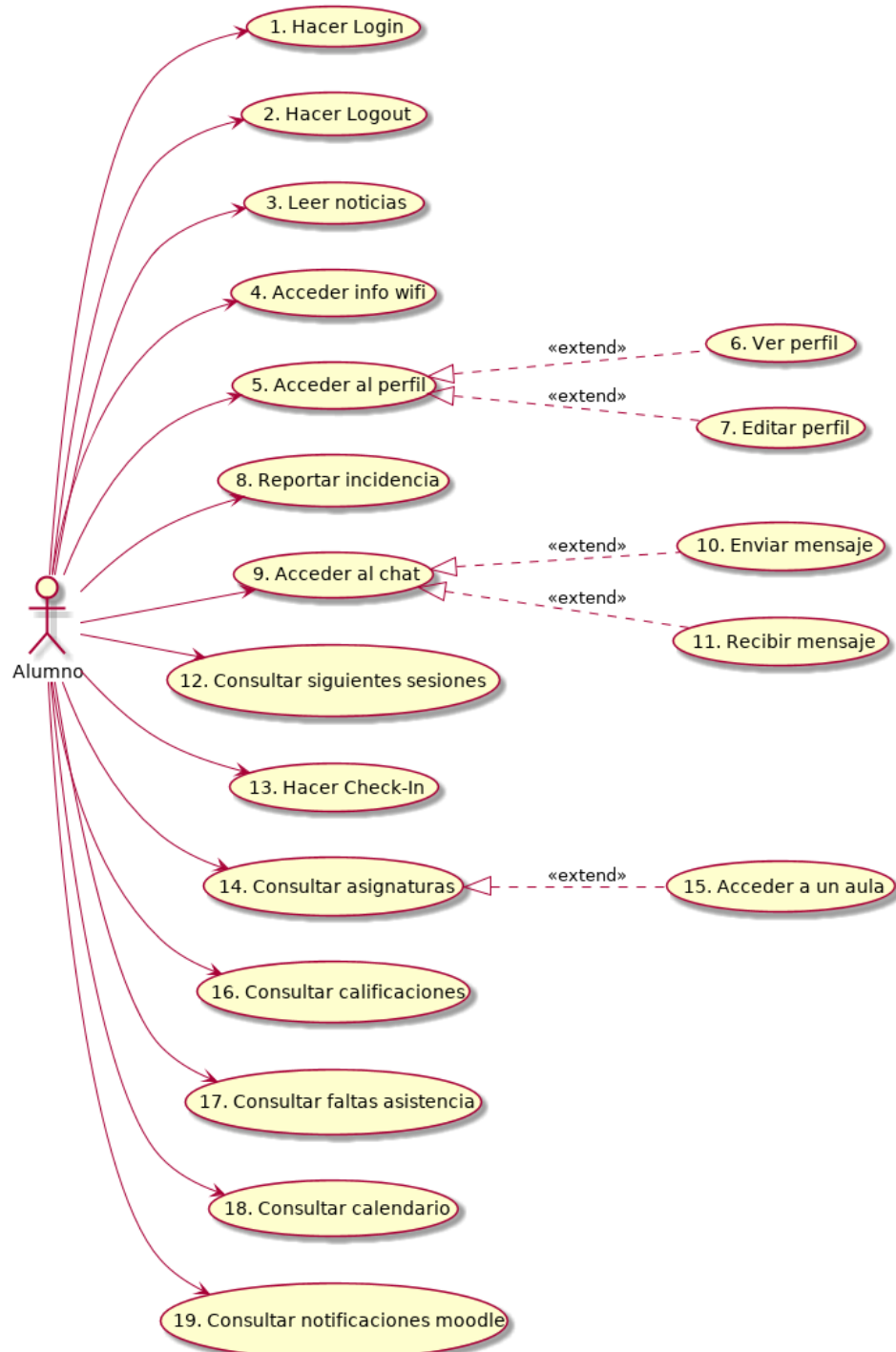
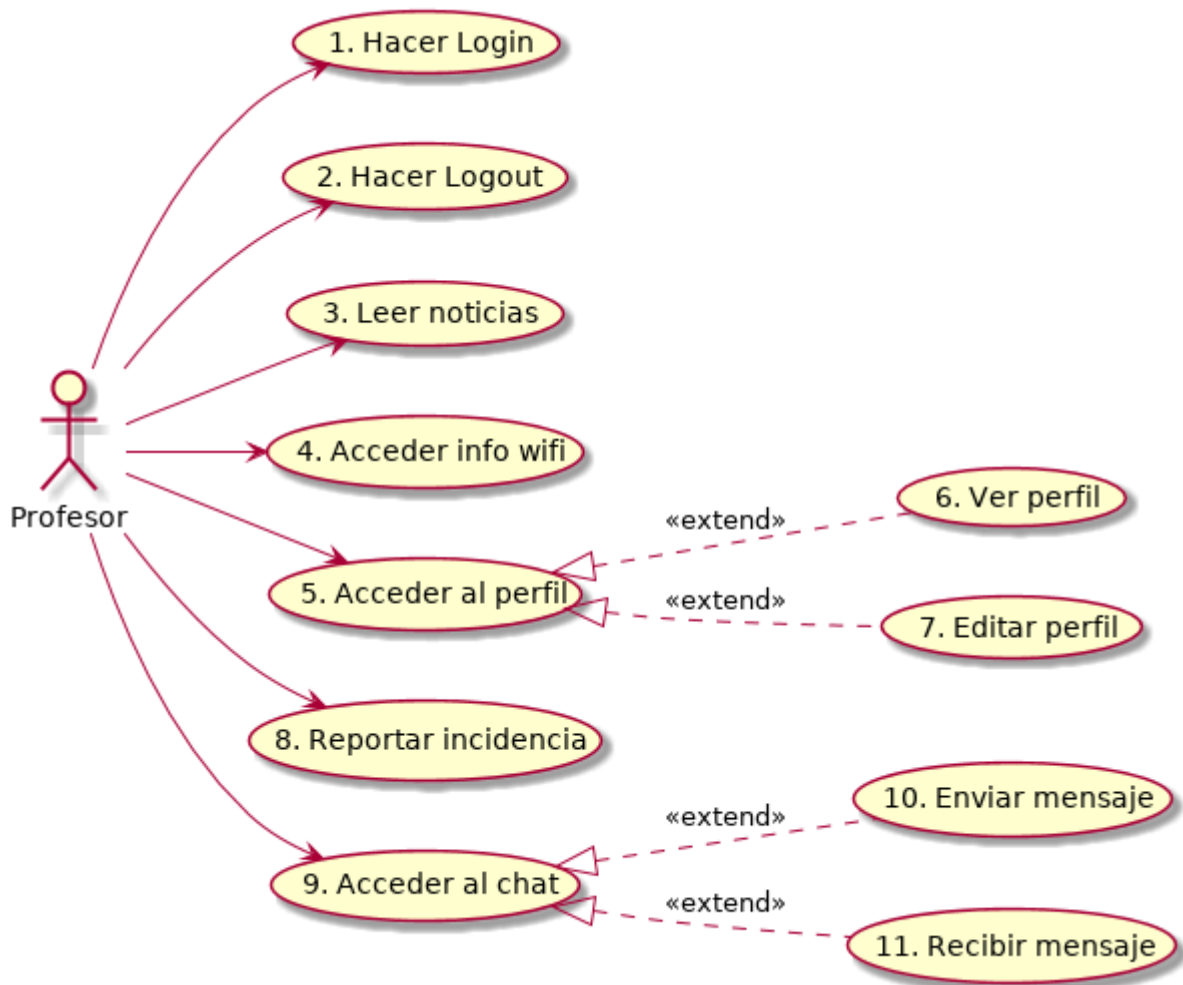
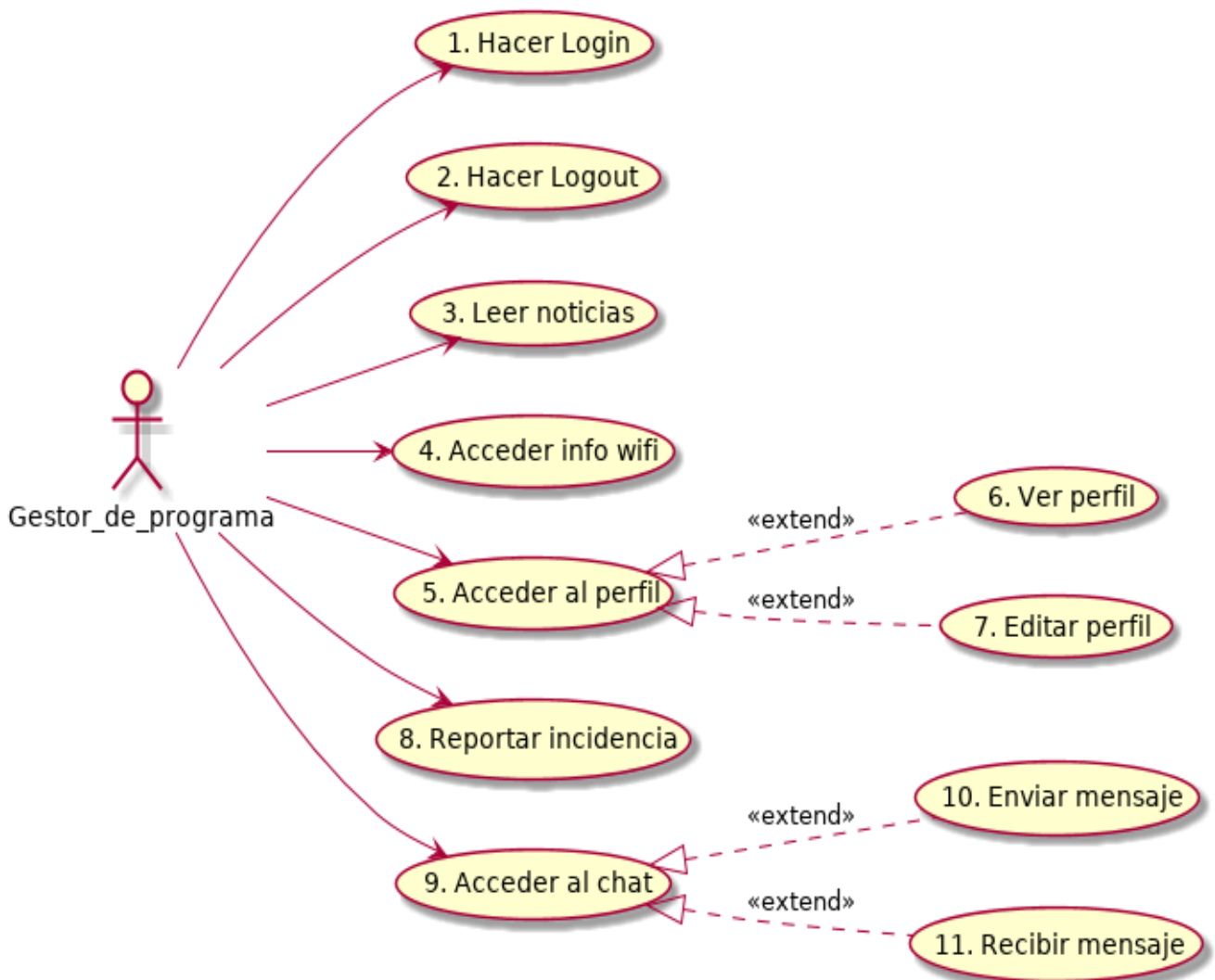


Ilustración 6 Casos de uso de alumno

En la ilustración 6 podemos observar las acciones que puede realizar un usuario con el perfil de un alumno. Este usuario, al ser en el que se ha enfocado la primera versión de la aplicación, es el que más acciones puede hacer con respecto a los otros perfiles.

Profesores*Ilustración 7 Casos de uso de profesor*

Como podemos observar en la ilustración 7, las acciones que un profesor puede realizar en la aplicación son significativamente menores a las de un alumno. Esto es debido a que las funcionalidades que son exclusivas del perfil de un profesor no han sido incluidas en esta versión de la aplicación. Por lo tanto, las acciones que puede hacer el profesor son las básicas que se permiten a cualquier usuario que pueda acceder a la aplicación.

Gestor de programa*Ilustración 8 Casos de uso de gestor de programa*

Como podemos observar en la ilustración 8, las acciones que un gestor de programa puede realizar son las mismas que un profesor, es decir, las acciones básicas de cualquier usuario con acceso a la aplicación de EADA. En un futuro se podrán añadir las funcionalidades específicas para este tipo de usuario.

Exalumnos

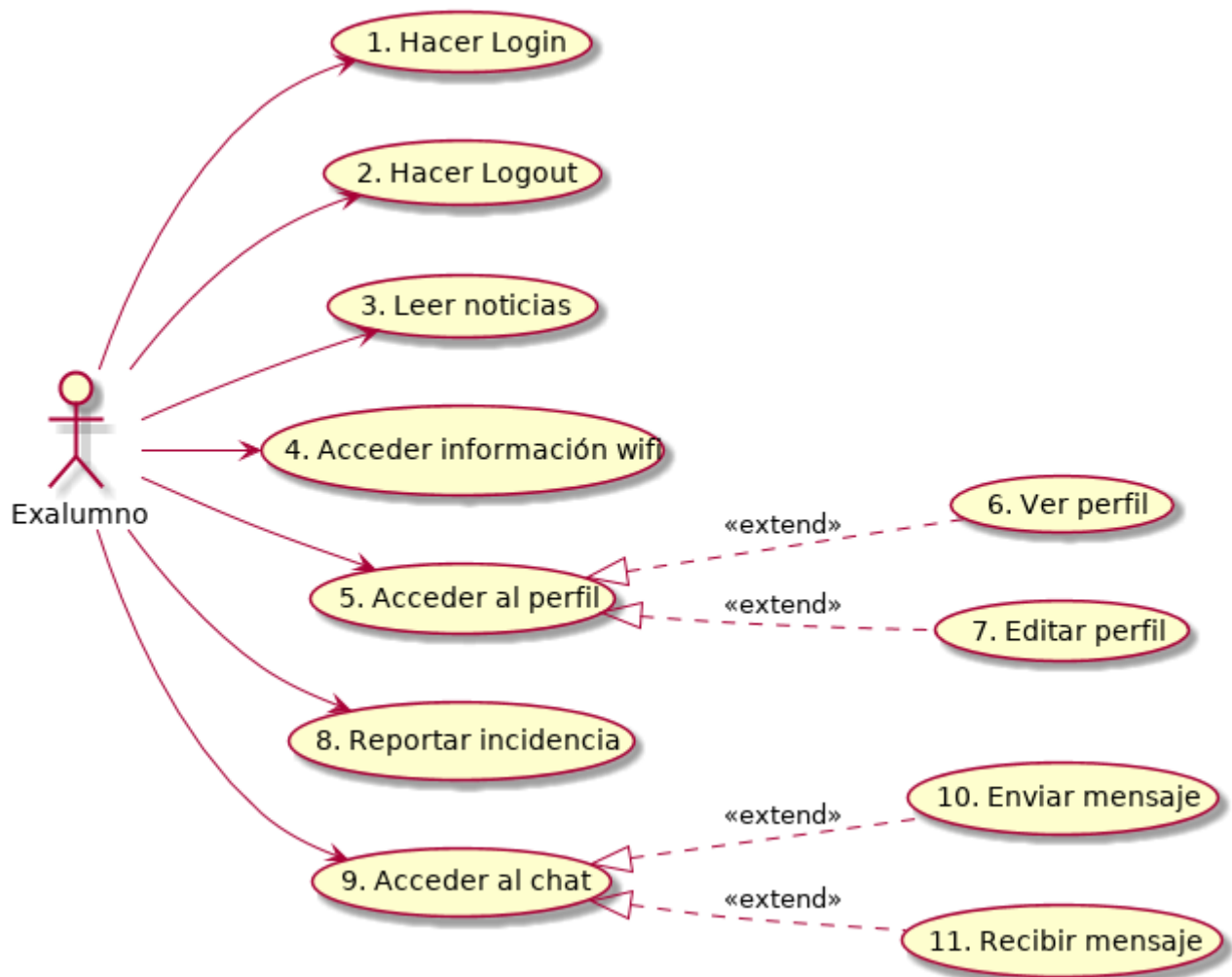


Ilustración 9 Casos de uso de exalumno

En la ilustración 9 podemos observar las acciones que puede realizar un exalumno. Al igual que los profesores y gestores de programa, estas acciones son las básicas para cualquier tipo de usuario que acceda a la aplicación de EADA.

3.2.2 Descripción de los casos de uso textuales

A continuación, se muestran los casos de uso textuales identificados anteriormente:

CU 1	Iniciar sesión
Objetivo	Iniciar sesión en la aplicación
Actores	Alumno, Profesor, Gestor de programa, Exalumno
Precondiciones	El usuario no está logeado en el sistema
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Entra en la aplicación 2. Sistema: Presenta página de Login 3. U: Introduce usuario y contraseña 4. S: Valida datos del usuario 5. S: Permite acceso y muestra la Home
Extensiones	4.1 Si los datos introducidos son incorrectos muestra mensaje de error.

CU 2	Cerrar sesión
Objetivo	Cerrar sesión en la aplicación
Actores	Alumno, Profesor, Gestor de programa, Exalumno
Precondiciones	El usuario está logeado en el sistema
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Hace clic en el botón de menú 2. Sistema: Muestra el menú 3. U: Hace clic en Logout 4. S: Cierra sesión en la aplicación 5. S: Muestra la página de Login

Extensiones	-
-------------	---

CU 3 Leer noticias	
Objetivo	Leer las noticias de EADA
Actores	Alumno, Profesor, Gestor de programa, Exalumno
Precondiciones	El usuario está logeado en el sistema y tiene la aplicación cerrada
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Inicia la aplicación 2. Sistema: Valida usuario y muestra home 3. U: Baja hasta la sección de noticias
Extensiones	-

CU 4 Acceder información wifi	
Objetivo	Obtener la información para acceder a la configuración del wifi de EADA
Actores	Alumno, Profesor, Gestor de programa, Exalumno
Precondiciones	El usuario está logeado en el sistema
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Hace clic en el botón de menú 2. Sistema: Muestra el menú 3. U: Hace clic en Eduroam 4. S: Muestra la página de Eduroam 5. U: Navega por la página
Extensiones	-

CU 5	Acceder al perfil
Objetivo	Acceder al perfil de usuario
Actores	Alumno, Profesor, Gestor de programa, Exalumno
Precondiciones	El usuario está logeado en el sistema
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Hace clic en el botón de My perfil 2. Sistema: Muestra la página con los datos del perfil
Extensiones	-

CU 7	Editar datos de perfil
Objetivo	Editar los datos del perfil
Actores	Alumno, Profesor, Gestor de programa, Exalumno
Precondiciones	El usuario está en la página del perfil
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Cambia los datos que desea 2. U: Pulsa el botón de guardar 3. Sistema: Procesa los datos 4. S: muestra mensaje de datos guardados
Extensiones	3.1 Si los datos son incorrectos muestra mensaje de error en el dato incorrecto

CU 8	Reportar una incidencia
Objetivo	Reportar una incidencia
Actores	Alumno, Profesor, Gestor de programa, Exalumno
Precondiciones	El usuario está logeado
Procedimiento	<ol style="list-style-type: none"> 1. Hace clic en el botón de menú 2. Sistema: Muestra el menú 3. U: Hace clic en Reportar incidencia 4. S: Muestra la página de Reportar incidencia 5. U: Selecciona el tipo de incidencia 6. S: Muestra formulario para rellenar los datos 7. U: Rellena los datos 8. U: Pulsa el botón enviar 9. S: Procesa la petición 10.S: Muestra mensaje conforme se ha enviado la incidencia
Extensiones	9.1 Si el envío no se puede realizar, se muestra un mensaje de error

CU 9	Acceder al chat
Objetivo	Acceder al chat privado de EADA
Actores	Alumno, Profesor, Gestor de programa, Exalumno
Precondiciones	El usuario está logeado en el sistema
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Hace clic en el botón de Chat 2. Sistema: Muestra la página del chat

Extensiones	-
-------------	---

CU 10		Enviar mensaje por el chat
Objetivo	Enviar un mensaje por el chat	
Actores	Alumno, Profesor, Gestor de programa, Exalumno	
Precondiciones	El usuario está en la página del chat y tiene una conversación activa	
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Escribe el texto o selecciona un documento/imagen para enviar 2. U: Pulsa el botón de enviar 3. Sistema: Envía el mensaje utilizando el Web Service 4. S: Muestra el mensaje enviado en la página del usuario 	
Extensiones	4.1 Si el envío no se puede realizar, se muestra un mensaje de error en el mensaje enviado.	

CU 12		Consultar siguientes sesiones
Objetivo	Consultar las siguientes sesiones disponibles	
Actores	Alumno	
Precondiciones	El usuario está logeado en el sistema y tiene la aplicación cerrada	
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Inicia la aplicación 2. Sistema: Valida usuario y muestra home 3. U: Visualiza las siguientes sesiones en la home 	

Extensiones	-
-------------	---

CU 13 Realizar check-in en un aula	
Objetivo	Acceder a un aula utilizando el código QR de un aula
Actores	Alumno
Precondiciones	El usuario está logeado en el sistema y tiene la aplicación cerrada
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Inicia la aplicación 2. Sistema: Valida usuario y muestra home 3. U: Visualiza las siguientes sesiones en la home 4. U: Hace clic en el botón de chek-in de la sesión correspondiente 5. S: Abre plugin de la cámara para leer el código QR 6. U: Enfoca la cámara al código QR 7. S: Comprueba que el usuario tiene acceso 8. S: vuelve a la home y muestra mensaje para poder pasar
Extensiones	<p>7.1 Si no tiene acceso, muestra mensaje conforme no puede pasar</p> <p>7.2 Si ha llegado tarde, muestra mensaje conforme ha llegado tarde</p> <p>7.3 Si ha llegado pronto, muestra mensaje conforme aún no es la hora de pasar</p>

CU 14 Consultar asignaturas	
Objetivo	Consultar las asignaturas del curso
Actores	Alumno
Precondiciones	El usuario está logeado en el sistema y tiene la aplicación cerrada

Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Inicia la aplicación 2. Sistema: Valida usuario y muestra home 3. U: Hace clic en el botón de subjects 4. S: Carga el contenido de las asignaturas 5. S: Muestra la página 6. U: Visualiza asignaturas
Extensiones	4.1 Si la carga falla, se comunica el error

CU 15 Acceder a un aula virtual	
Objetivo	Acceder a un aula virtual desde la aplicación
Actores	Alumno
Precondiciones	El usuario está logeado en el sistema y estar en la página de subjects
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Hace clic en una asignatura 2. Sistema: Abre el navegador incorporado 3. S: Muestra el aula virtual 4. U: Visualiza el aula virtual
Extensiones	-

CU 16 Consultar calificaciones	
Objetivo	Consultar las notas disponibles
Actores	Alumno
Precondiciones	El usuario está logeado en el sistema y tiene la aplicación cerrada

Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Inicia la aplicación 2. Sistema: Valida usuario y muestra home 3. U: Hace clic en el botón de Grades 4. S: Carga el contenido de las calificaciones 5. S: Muestra la página 6. U: Visualiza las notas
Extensiones	4.1 Si la carga falla, se comunica el error

CU 17 Consultar faltas de asistencia	
Objetivo	Consultar las faltas de asistencia
Actores	Alumno
Precondiciones	El usuario está logeado en el sistema y tiene la aplicación cerrada
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Inicia la aplicación 2. Sistema: Valida usuario y muestra home 3. U: Hace clic en el botón de Absences 4. S: Carga el contenido de las faltas de asistencia 5. S: Muestra la página 6. U: Visualiza las faltas de asistencia
Extensiones	4.1 Si la carga falla, se comunica el error

CU 18 Consultar calendario	
Objetivo	Consultar calendario

Actores	Alumno
Precondiciones	El usuario está logeado en el sistema y tiene la aplicación cerrada
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Inicia la aplicación 2. Sistema: Valida usuario y muestra home 3. U: Hace clic en el botón de Calendar 4. S: Carga el contenido del calendario 5. S: Muestra la página 6. U: Visualiza el calendario
Extensiones	4.1 Si la carga falla, se comunica el error

CU 19 Consultar notificaciones moodle	
Objetivo	Consultar las notas disponibles
Actores	Alumno
Precondiciones	El usuario está logeado en el sistema y tiene la aplicación cerrada
Procedimiento	<ol style="list-style-type: none"> 1. Usuario: Inicia la aplicación 2. Sistema: Valida usuario y muestra home 3. U: Hace clic en notificaciones 4. S: Carga las notificaciones 5. S: Muestra la página 6. U: Visualiza las notificaciones
Extensiones	4.1 Si la carga falla, se comunica el error

3.3 Requisitos no funcionales

Después de haber visto los requisitos funcionales de este proyecto, pasamos a los requisitos no funcionales^[4]. Estos requisitos son las condiciones o restricciones que son impuestos por parte del cliente y que pueden usarse para juzgar la operación de un sistema en lugar de su comportamiento específico. Los requisitos no funcionales más importantes de este proyecto son los siguientes:

- **Interfaz responsive:** Debido a la amplia diversidad de dispositivos que se utilizan en EADA, es muy importante crear una interfaz que sea capaz de adaptarse a los diferentes tipos de pantalla que tienen los dispositivos. Por este motivo, era primordial conseguir un diseño estable que se adapte a cualquier tipo de pantalla para mostrar todos los contenidos en distintos dispositivos correctamente.

Además, con este requisito, evitamos invertir tiempos en crear versiones para diferentes terminales y así ahorrar tiempo en el desarrollo y a posteriori en el mantenimiento.

- **Escalabilidad:** Debido al plazo de entrega del que se disponía, no se pudieron añadir ciertas funcionalidades que otorgarían más valor a la aplicación. Sin embargo, estas funcionalidades no quedaron descartadas y se planteó su desarrollo en futuras versiones de la aplicación. Por lo tanto, se exigió este requisito para que dichas mejoras puedan ser incluidas de forma sencilla en futuras implementaciones.
- La interfaz ha de ser **sencilla** e **intuitiva** para los usuarios que la utilicen.
- La aplicación ha de ser **compatible** para dispositivos Android y IOS.
- Para el desarrollo de la aplicación, se utilizará el **entorno de desarrollo Apache Cordova**^[2] que permite construir aplicaciones para dispositivos móviles utilizando CSS3, HTML5 y JavaScript en lugar de utilizar APIs específicas de cada plataforma.
- Se hará uso del **RESTful Web Service que dispone EADA** para obtener la información necesaria para ser mostrada en la aplicación.

- Se procurará que los **tiempos de respuesta** sean **inferiores a cinco segundos**. Si no es posible, se mostrará una **animación de carga** para mostrar al usuario de que se está trabajando en la solicitud.
- Se han de **notificar**, en un mensaje popup, los **errores** que puedan suceder en la comunicación con el Web Service para informar al usuario que la aplicación funciona correctamente.
- El sistema de **Login** se hará mediante el **Web Service** que realizará la petición al servidor CAS que dispone EADA.
- La aplicación solo podrá ser utilizada por **usuarios** que estén **registrados en EADA** como alumnos, profesores, personal, etc... Si una persona ajena a EADA se descarga la aplicación no podrá ver más que la pantalla de login sin posibilidad de registrarse.
- En todas las peticiones realizadas por el usuario, **se comprobará que la sesión sea correcta**. Es decir, el sistema comprobará el token facilitado por el servidor CAS que será guardado en la memoria interna del dispositivo, en cada petición que se haga al Web Service.
- **La fecha de lanzamiento** de la aplicación será la misma que la fecha de lanzamiento del nuevo Campus Virtual, que es a finales de enero.
- El lanzamiento de la aplicación ha de venir acompañada con un **manual de ayuda**, dirigido a los alumnos, que serán publicados en la página de inicio del Campus Virtual.
- La aplicación ha de tener disponible un **prototipo funcional**, antes de la fecha de lanzamiento, para poder ser validada en la Apple Store como en la Play Store.

4. Diseño

4.1 Arquitectura

Para la implementación de este proyecto se ha utilizado el tipo de arquitectura **Modelo-Vista-Controlador**^[5] (**MVC**). Este patrón de arquitectura de software separa los datos y la lógica de negocio de la representación de la aplicación y del módulo encargado de gestionar los eventos que se realicen con la información antes de ser visualizada.

En este proyecto, la vista viene a ser todas las páginas implementadas que se muestran en la aplicación. El contenido que se visualiza en la aplicación se obtiene mediante un Web Service implementado con JavaScript y que utiliza AJAX para realizar las peticiones al servidor. La parte del Web Service que se aloja en el servidor está implementada en Java. El paquete del controlador incluye la parte del Web Service que está alojada en el cliente y la parte alojada en el servidor.

Para finalizar, en el paquete del modelo, se encuentran todas las clases implementadas en Java de la cual se obtienen los datos de la forma correspondiente para ser servidos a la aplicación.

4.2 Templates

A continuación, se definirán los templates que permitirán cargar las diferentes pantallas que dispone la aplicación. Los nombres de estos templates han de ser intuitivos para el usuario, es decir, han de tener un nombre que sea fácil de interpretar por el usuario y que defina con exactitud su contenido.

Además, los nombres de los diferentes templates han de ser cortos y vendrán acompañados por iconos que representen la página que cargan ya que, serán utilizados dentro de la aplicación en la descripción de los botones. También, en algunas ocasiones, estos botones dispondrán de colores para diferenciar cada página. Los templates son los siguientes:

Nombre del template	Descripción de la página que carga	Perfiles
Login	Página que solicitará el usuario y contraseña para que el usuario pueda iniciar sesión en la aplicación.	Alumno, Profesor, Gestor de programa, Exalumno
Home	Página principal donde se mostrará, de forma general, el contenido que dispondrá cada usuario dependiendo de su rol. Para los alumnos, se mostrarán accesos directos a asignaturas, notas, check-in, siguientes sesiones, próximos eventos, noticias y faltas de asistencia. Para el resto de roles, se mostrarán, únicamente, noticias.	Alumno, Profesor, Gestor de programa, Exalumno
Asignaturas	En esta página, el alumno podrá visualizar las asignaturas del curso.	Alumno
Calificaciones	En esta página, el alumno podrá visualizar las notas de las asignaturas del curso.	Alumno
Faltas de asistencia	En esta página, el alumno podrá visualizar las faltas de asistencia que ha realizado durante el curso.	Alumno
Calendario	En esta página, el alumno podrá visualizar los próximos eventos de las asignaturas del curso.	Alumno
Check-in	En esta página, el alumno podrá visualizar un histórico con los accesos a aulas durante el transcurso del curso.	Alumno
eChat	En esta página, podrán acceder al chat privado de EADA para comunicarse con el resto de participantes.	Alumno, Profesor, Gestor de programa, Exalumno
Mi perfil	En esta página, se podrá consultar los datos del perfil de usuario, así como modificar dichos datos.	Alumno, Profesor, Gestor de programa, Exalumno
Notificaciones	En esta página, los usuarios podrán consultar las notificaciones que hayan recibido del campus virtual.	Alumno, Profesor, Gestor de programa, Exalumno

Eduroam	En esta página, los usuarios podrán acceder a la información para configurar la red wifi privada de EADA en sus dispositivos.	Alumno, Profesor, Gestor de programa, Exalumno
Reportar incidencia	En esta página, los usuarios podrán reportar cualquier tipo de incidencia rellenando un formulario.	Alumno, Profesor, Gestor de programa, Exalumno
Confirmar salida	En esta página, se solicitará la confirmación por parte del usuario para salir de la aplicación.	Alumno, Profesor, Gestor de programa, Exalumno

5. Implementación

5.1 Estructuración del proyecto

La estructura que presenta este proyecto es más complicada de lo que a primera vista parece. El desarrollo de esta aplicación no es más que otro complemento añadido a todo el sistema que se ha creado especialmente para EADA y que ha ido evolucionando con el paso de los años.

La base de datos y toda la lógica de negocios de la que se nutre esta aplicación es la misma que se utiliza en otros muchos desarrollos de los que dispone EADA y que se han explicado en el contexto de este proyecto. Por este motivo, tan solo explicaré de manera breve la estructura que tiene el proyecto creado en Apache Cordova^[2] para el desarrollo de esta aplicación.

En la ilustración 10 se muestra la estructura que presenta un proyecto en Apache Cordova^[2]. Las carpetas que nos interesan son las siguientes:

- **Platforms:** En esta carpeta se guardará la configuración necesaria para que la aplicación pueda ejecutarse en la plataforma correspondiente.
- **Plugins:** En esta carpeta se almacenarán todos los ficheros que requieren los plugins que se añadan a la aplicación.
- **www:** En esta carpeta es donde hemos de incluir todo el código de nuestra aplicación. Es decir, este es el directorio que contendrá todo el código fuente basado en web.

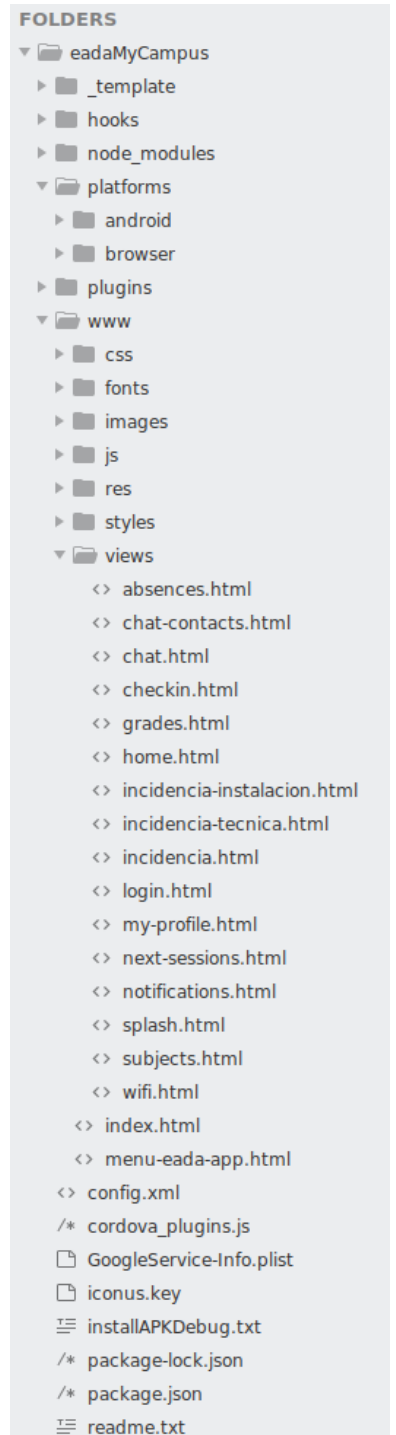


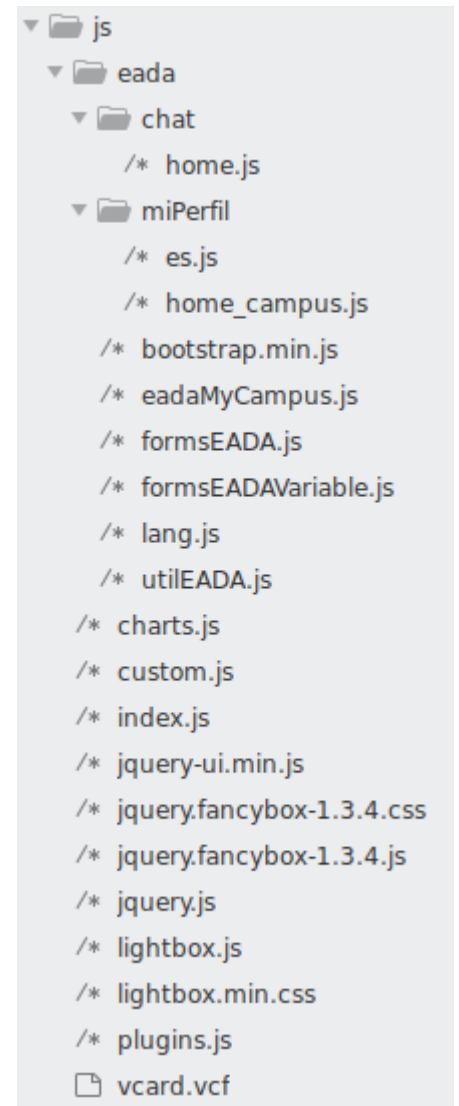
Ilustración 10 Estructura Apache Cordova^[2]

Ahora, nos centraremos en la estructura que tiene la carpeta **www** puesto que es donde está el código de esta aplicación.

Como en cualquier web, la estructura se divide en diferentes carpetas para mantener un orden. Las carpetas más importantes son:

- **styles:** Contiene el código css para el diseño de esta aplicación.
- **Js:** En esta carpeta se almacenan todos los ficheros JavaScript del proyecto y es donde está toda la lógica de esta aplicación. En la ilustración 11 podemos observar su estructura, donde el fichero principal es eadaMyCampus.js.
- **Views:** Contiene las vistas implementadas en código HTML5 que, mediante funciones JavaScript, se cargan en el index.html para poder ser visualizadas en las páginas correspondientes de la aplicación.

He de mencionar que, Apache Cordova^[2] dispone de ciertos comandos predefinidos que sirven para instalar plugins, generar la apk para la plataforma que se desee, entre otras funciones. Es una herramienta muy útil y muy sencilla de aprender ya que dispone de una guía de uso muy bien estructurada y explicada.



*Ilustración 11 Estructura
Carpeta www*

5.2 Tecnología del proyecto

Para la realización de este proyecto se ha precisado de diferentes tecnologías además de tres entornos de trabajo. El primer entorno será en la máquina local, el segundo es el entorno de test y el último es el entorno final, es decir, el entorno de producción. Los dos primeros entornos mencionados anteriormente están preparados para simular al de producción y así poder realizar las pertinentes pruebas antes de lanzar el producto. A continuación, se detallarán algunas de las tecnologías utilizadas.

Para poder ofrecer los contenidos en la web se ha utilizado **Apache**^[4], que “es un servidor web gratuito y de código abierto para plataformas Unix. Es uno de los servidores web más antiguos y confiables, con la primera versión lanzada hace más de 20 años, en 1995”. Otra de las tecnologías utilizadas es **Jboss**^[7] y **Apache Tomcat**^[8], el primero es un servidor de aplicaciones Java EE de código abierto implementado en Java y puede ser utilizado en cualquier sistema operativo en la que esté disponible la máquina virtual de Java. En cambio, Apache Tomcat^[8] funciona como un contenedor de servlets^[9] que son una clase en el lenguaje de programación Java utilizadas para ampliar las capacidades de un servidor.

Como se ha comentado en varios puntos de este proyecto, se ha utilizado el entorno de desarrollo de aplicaciones **Apache Cordova**^[2] ya que nos facilita crear una aplicación multiplataforma utilizando un único código y lenguajes de programación propios de una web como lo son **HTML5**, **CSS3** y **JavaScript**. Para manipular el código de la aplicación, se ha utilizado **Sublime Text** que es un editor de texto y de código fuente. Para la parte del Web Service, se ha utilizado el IDE **Eclipse**.

Para poder llevar a cabo un control de las versiones y poder trabajar en equipo sin llegar a molestar a los otros desarrolladores, se ha utilizado **Git** en conjunto con una herramienta cliente cuyo nombre es **GitKraken**. Esta herramienta es muy sencilla de utilizar y gracias a su interfaz permite visualizar el estado del proyecto de forma muy cómoda.

Para finalizar, se ha utilizado **Jersey** conjuntamente con **Jackson** para reunir un objeto Java en JSON y viceversa. Estas tecnologías son servicios RESTful que producen y consumen secuencias JSON.



Ilustración 12 Logos de algunas de las tecnologías utilizadas

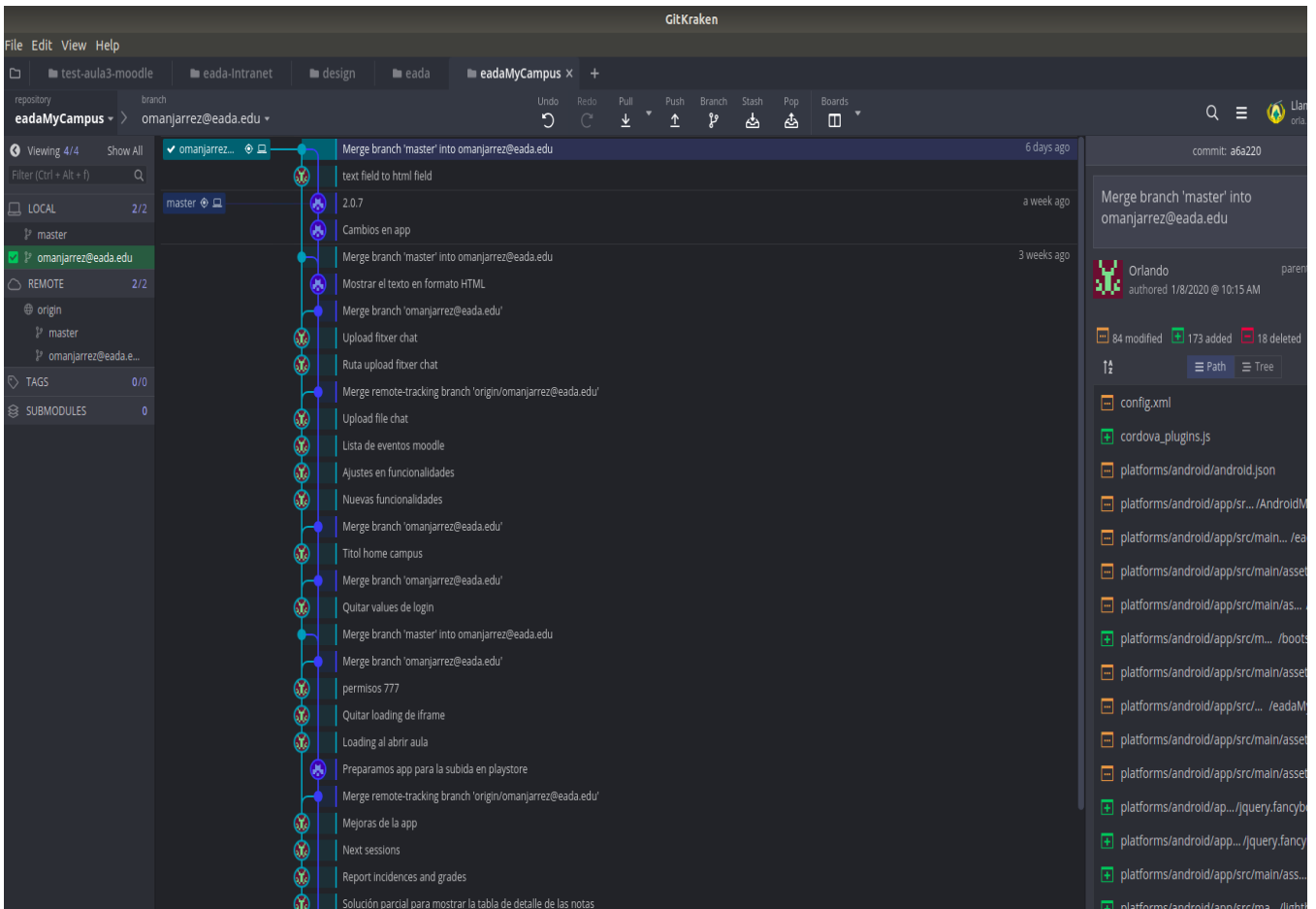


Ilustración 13 Vista de la herramienta GitKraken

5.3 Autenticación

El sistema que se utiliza para iniciar sesión en esta aplicación es una variante del que dispone EADA para autenticarse en todos sus servicios. Dispone de un servidor CAS que se encarga de comprobar si el usuario que intenta acceder al servicio es un usuario válido.

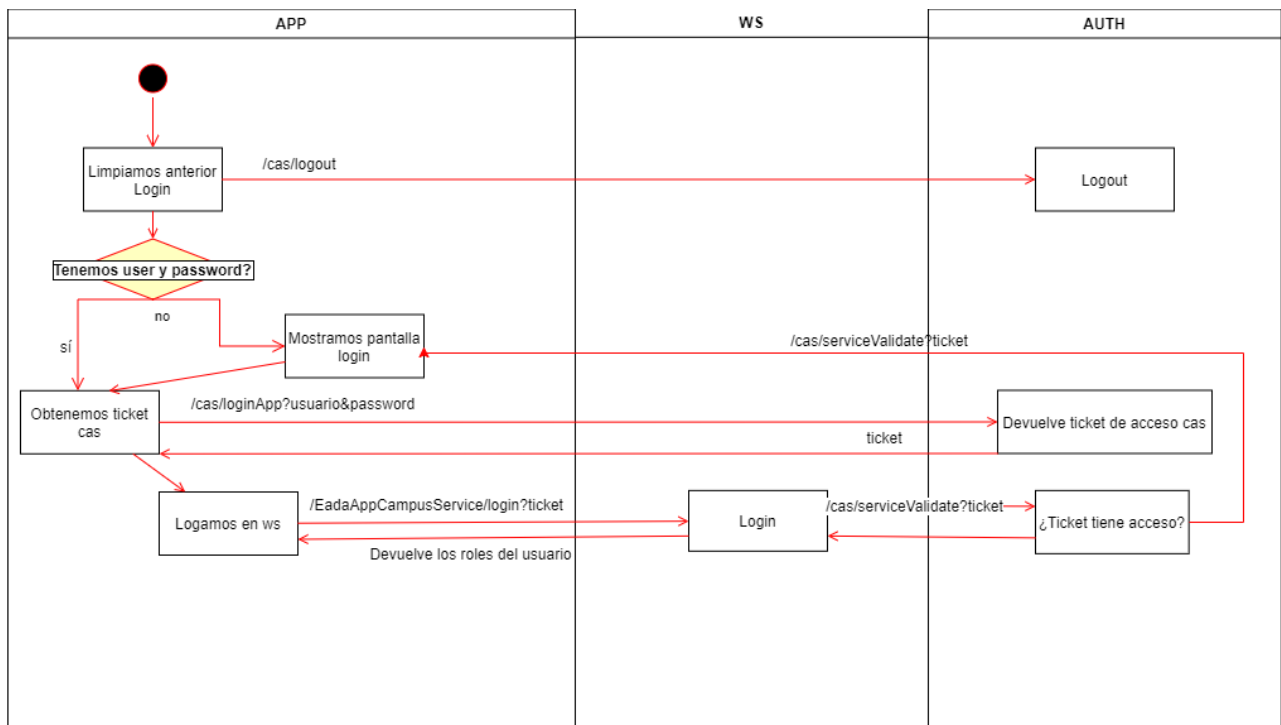


Ilustración 14 Gráfico de inicio de sesión

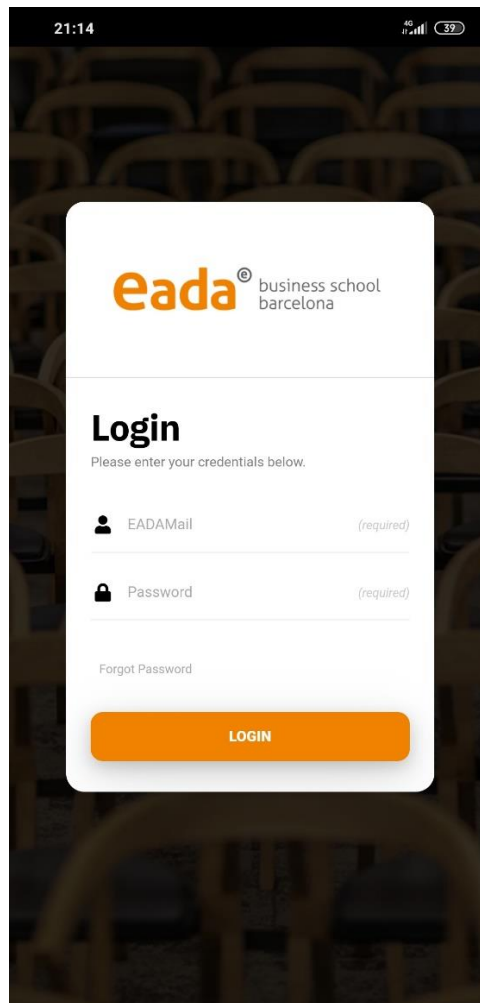


Ilustración 15 Página de Login

En la Ilustración 14 se muestra el ciclo que realiza la aplicación para iniciar sesión en la aplicación. Destacan tres elementos, la aplicación, el Web Service y el servidor donde está alojado el CAS. En la Ilustración 15 se puede observar el diseño de la página de Login.

Para garantizar que la sesión es correcta, se eliminan los datos almacenados mediante una petición Logout al servidor CAS. Después, se comprueba que no existan guardados los datos de inicio de sesión en el dispositivo. Si existen, se realiza la autenticación al servidor CAS, de lo contrario, se muestra la página de Login.

Fuera de la aplicación, este sistema permite mantener una sesión abierta en todos los servicios de EADA gracias al sistema de SSO (Single Sign-On) que incorpora CAS.

5.4 Diseño adaptativo según rol

Gracias al sistema de autenticación, en el momento que un usuario inicia sesión, la aplicación es capaz de identificar los roles del usuario. Debido a esto, los contenidos que podemos mostrar cambian en función del rol que el usuario tenga asignado.

A pesar de que la aplicación se enfoque en el rol de los alumnos, no deja de lado los otros roles y ofrece un contenido básico además de permitirles usar el chat privado de EADA. Con el paso del tiempo, se irán añadiendo las distintas funcionalidades propias de cada rol y que puedan funcionar correctamente en la aplicación.

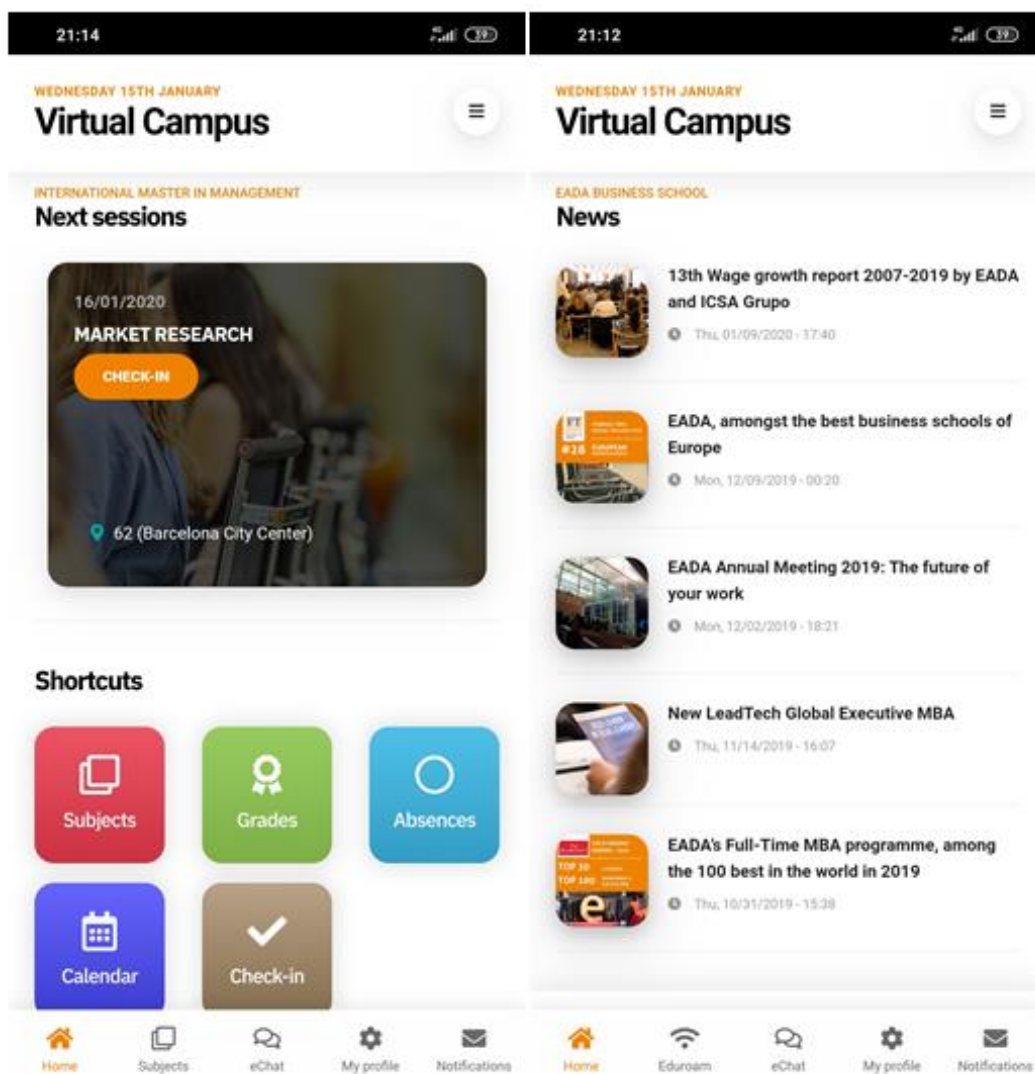


Ilustración 16 A la izquierda home de rol alumno y a la derecha home de otro rol

Como se puede observar en la ilustración 16, el contenido que la aplicación muestra en la home a un usuario con el rol de estudiante es más completa que la de un usuario con un rol

distinto. Mientras que el usuario de la izquierda podrá desplazar la pantalla hacia abajo y ver las noticias, al usuario de la derecha solo dispone del apartado de noticias además de que, el menú inferior del alumno dispone de un acceso directo a las asignaturas mientras que los otros usuarios tienen un acceso directo a la configuración del wifi.

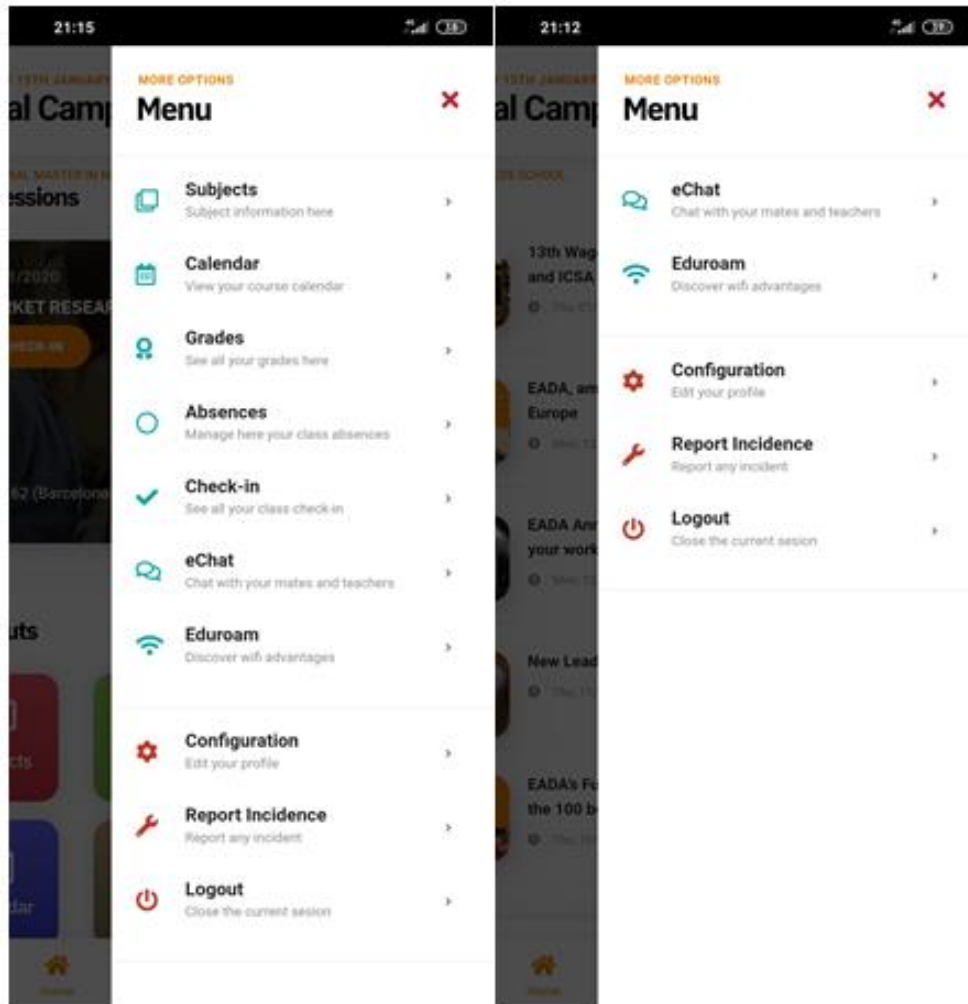


Ilustración 17 A la izquierda menú de rol alumno y a la derecha menú de otro rol

Tanto en la ilustración 16 como en la ilustración 17, podemos observar que el contenido de un alumno es más extenso que los otros roles. Sin embargo, la aplicación no deja de ser atractiva para el resto de usuarios ya que, la herramienta de chat otorga un plus al uso de esta aplicación. Con las futuras implementaciones, se espera poder añadir más funcionalidades para cada rol. Estas funcionalidades se explicarán en el apartado de posibles ampliaciones.

5.5 Chat privado de EADA (eChat)

Esta herramienta es uno de los puntos fuertes de esta aplicación. No solo está incluida en la app si no que, esta herramienta está disponible en todas las plataformas que dispone EADA. Es decir, podremos usar esta herramienta desde el Campus Virtual, desde las Aulas Virtuales y desde la plataforma de gestión interna, así como desde la aplicación.

Para poder hacer esto posible, esta funcionalidad utiliza un protocolo libre de mensajería instantánea basado en XMPP llamado Jabber^[10]. Aplicando pequeños cambios al añadir la herramienta a la aplicación, los mensajes recibidos por el chat se mostrarán como notificaciones push además de las notificaciones propias del campus.

Con esta herramienta, es posible crear conversaciones privadas entre usuarios, comunidades donde los integrantes pueden participar en la conversación o comunidades unilaterales, es decir, comunidades que solo sirvan como un tablón de anuncios.

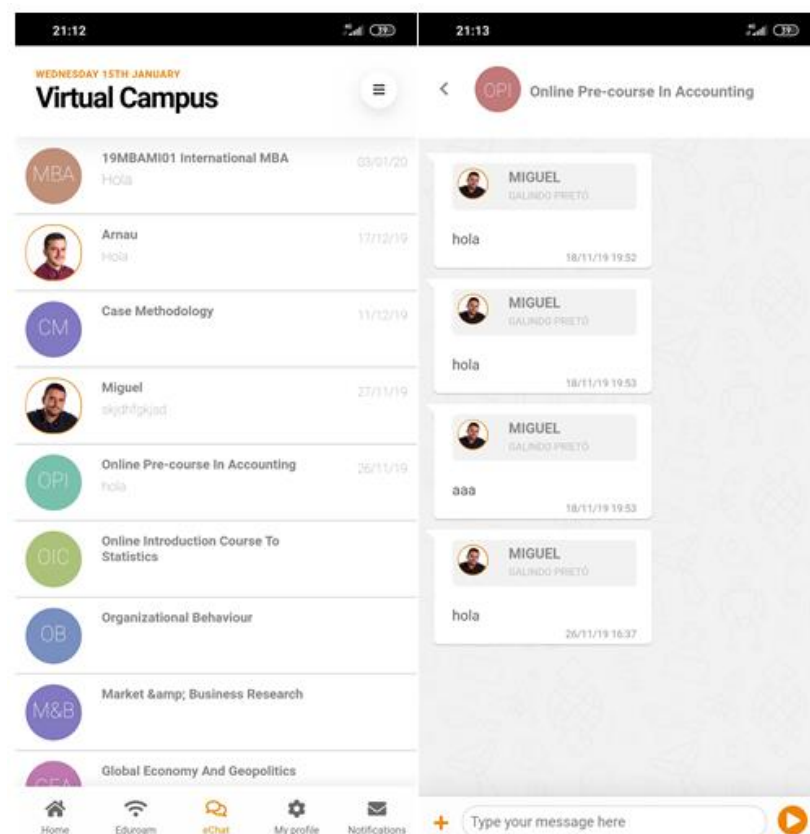


Ilustración 18 A la izquierda conversaciones activas del chat y a la derecha una conversación

5.6 Llamadas asíncronas AJAX

Debido a que toda la información disponible en la aplicación proviene de un Web Service, la utilización de la Tecnología AJAX es muy importante para el correcto funcionamiento de la aplicación.

Al iniciar la aplicación, parece que todo el contenido esté incluido estáticamente pero no es así. Cada vez que el usuario cambia de página, se ejecuta una petición AJAX que devuelve el contenido según la página que se desee visualizar. Gracias al entorno de desarrollo Apache Cordova^[2], podemos utilizar esta tecnología sin ningún problema ya que, al usar CSS3, HTML5 y JavaScript funciona como una web normal. En el momento de compilado, Apache Cordova^[2] usa sus librerías para crear una APK compatible para las plataformas que se hayan especificado en la configuración del proyecto.

Todas las peticiones serán procesadas mediante POST y si requiere de datos, estos son añadidos en un JSON. Si no ocurre ningún contratiempo, el Web Service devuelve una cadena JSON que será procesada mediante JavaScript y JQuery. En general, todas las peticiones tardan menos de cinco segundos, sin embargo, cuando una petición requiere de más tiempo, se muestra una pantalla de carga que se inicia cada vez que se hace una petición al Web Service y se oculta cuando este devuelve una respuesta, ya sea un error o el resultado esperado.



eada[®] Business school
Barcelona



This will only take a second. It's totally worth it!

Ilustración 19 Spinner de carga

5.7 Protocolo de comunicación del web Service

La estructura que se utiliza en este proyecto está desarrollada hace mucho tiempo y, con el transcurso del tiempo, ha ido evolucionando de tal forma que existen demasiadas clases como para poder hacer un diagrama exacto. En lugar de dicho diagrama, se incluye este apartado con la explicación del protocolo de comunicación de la aplicación con el servidor.

El protocolo de intercambio implementado en este desarrollo utiliza la tecnología REST. Esta tecnología se comunica mediante peticiones HTTP con el servidor y este responde con una cadena JSON. Para transformar estas cadenas JSON en clases de JAVA y viceversa, se utiliza Jersey en conjunto con Jackson^[12].

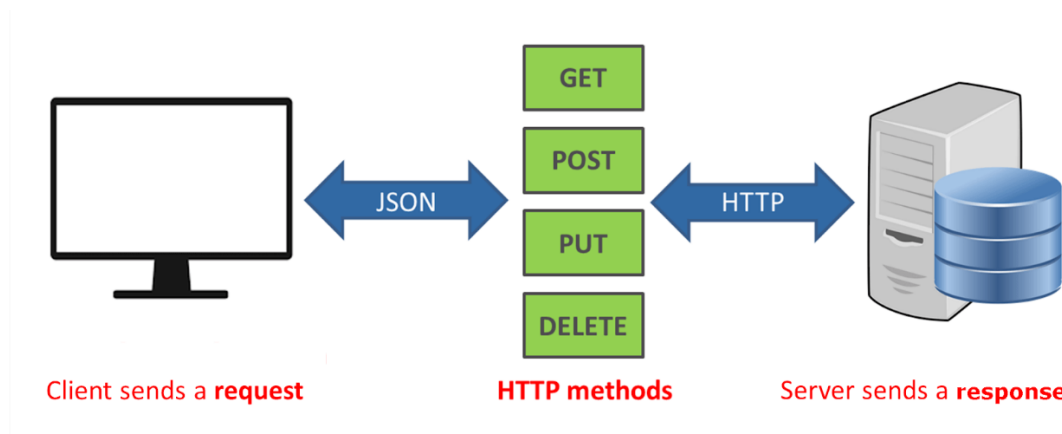


Ilustración 20 Protocolo REST

La ilustración 20 muestra el diagrama del protocolo Rest. Gracias a esta tecnología, se han implementado los métodos para poder obtener la información necesaria para llenar de contenidos a la aplicación. La comunicación se realiza mediante una ruta del estilo siguiente:

“url del servidor”/rs/externalServices/EadaAppCampusService/”método”/”id usuario”/”user name”/”sesión id”

La cadena Json que devuelve el servidor, tiene una apariencia muy similar a la que se presenta a continuación:

```
{“id”: 9, “error”: false, “values”: [1,2,3,4], “titol”: “Aula Virtual 55”}
```

Cuando el servidor recibe la solicitud, se encarga de realizar las operaciones pertinentes y responde al cliente mediante la cadena en JSON. Los métodos utilizados en el servicio EadaAppCampusService.java son los siguientes:

- **Login:** Se encarga de verificar si el usuario tiene permiso para iniciar sesión.
- **Home:** Es el método más grande implementado ya que, hace un resumen de todo el curso para poder ser mostrado en la página home.
- **getCurso:** Obtiene el curso activo del usuario.
- **seleccioCurs:** Si el usuario dispone de más de un curso, este método cambia el curso activo para ser visualizado en la aplicación.
- **proximasSesiones:** Devuelve las futuras sesiones, también están incluidas en la home.
- **getAsigAceptadasEnCurso:** Obtiene las aulas virtuales que están activas en el curso seleccionado en la aplicación para ser visualizadas en la sección subjects.
- **getUserMoodleEvents:** Obtiene las próximas tareas disponibles en Moodle para ser mostradas en la home.
- **listUltimasNotesAlumne:** Obtiene el listado de calificaciones para ser visualizadas en la página de calificaciones.
- **getNotificaciones:** Obtiene todas las notificaciones del usuario para mostrarlas en la página de Notificaciones.
- **getNotificacionesPendants:** Obtiene las últimas notificaciones para mostrarlas como notificación push.
- **personaCheckInFindView:** Obtiene la lista del registro de accesos a clases del alumno para visualizarlo en la página de check-In.
- **alumneFaltaAsistenciaFindView:** Obtiene las faltas de asistencia del alumno para mostrarlas en la página de Abscences.
- **getMiPerfil:** Obtiene los datos del perfil para mostrarlos en la página de perfil.
- **updateMiPerfil:** Este método actualiza los datos del perfil del usuario
- **checkIn:** Registra la entrada a clase de un alumno mediante la lectura del código QR.
- **getSesionesCalendar:** Obtiene los eventos que se mostrarán en el calendario.

Además de estos métodos, en esta clase se incluyen métodos propios de la API de Jabber^[10] para utilizar el chat de EADA.

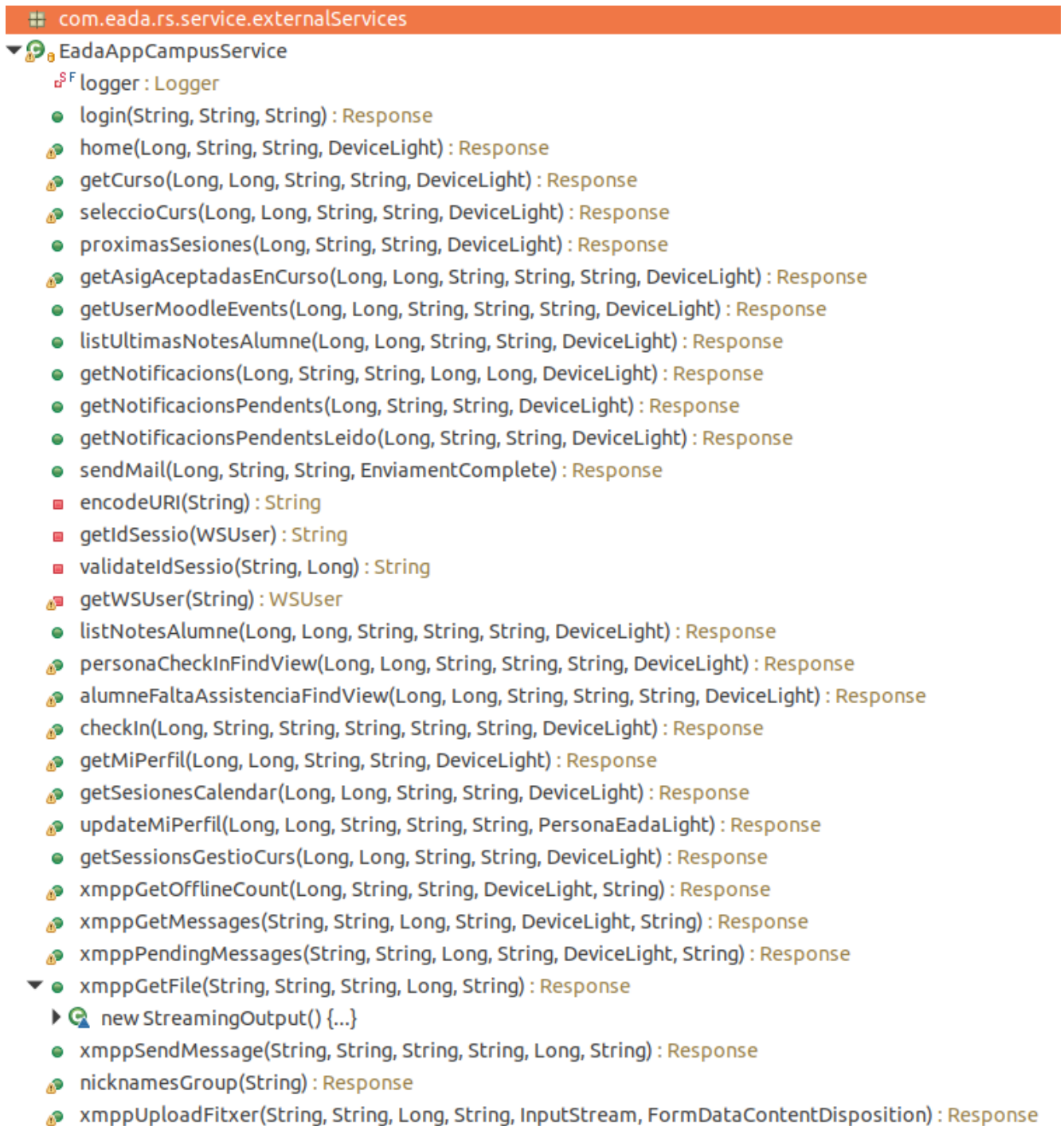


Ilustración 21 Métodos de la clase EadaAppCampusService.java

6. Posibles ampliaciones

Desde el principio, se planteó crear un producto escalable, es decir, un producto que sea capaz de evolucionar para incluir mejoras que sean capaces de cubrir las necesidades de los usuarios de EADA. Gracias a la manera que está programada la aplicación, podemos saber el tipo de rol que tiene el usuario que acaba de acceder y con esto seleccionar las herramientas que este puede utilizar.

Debido a este planteamiento, disponemos de un amplio abanico de mejoras que se irán implantando con el transcurso del tiempo. A continuación, se detallan algunas de ellas:

9.1 Versión de la aplicación para profesorado

Con esta mejora, se buscará crear una nueva versión donde el profesor pueda realizar la mayoría de acciones que hace en el campus virtual como:

- **Poder ver las asignaturas que imparte en EADA:** Disponer de un apartado donde pueda acceder de forma sencilla al contenido del aula virtual.
- **Poder calificar las tareas a los alumnos:** Disponer de una sección donde poder acceder a las tareas de los alumnos y calificarlas sin necesidad de acceder al campus virtual.
- **Facturar las clases para poder cobrarlas a posteriori:** Disponer de un apartado donde, de manera sencilla, el profesor pueda marcar las horas que ha dedicado a dar clases para poder cobrarlas.
- **Revisar el contrato con EADA:** Disponer de un apartado donde pueda ver las condiciones del contrato con EADA.
- **Revisar su agenda:** Disponer de un apartado con un calendario para poder visualizar su agenda.
- **Tener acceso a los recursos proporcionados por EADA:** Poder visualizar las guías de ayuda que EADA facilita a los profesores para poder realizar de forma correcta las diferentes tareas que realizan en EADA.
- **Crear guías metodológicas:** Disponer de un apartado donde pueda crear las guías metodológicas de cada asignatura.

- **Comprobar la evaluación obtenida:** Disponer de un apartado donde se visualice la evaluación que ha obtenido por los alumnos después de finalizar la asignatura.
- **Controlar la asistencia a clase:** El profesor podrá ver las fotos de los alumnos matriculados en su asignatura cuando hagan Check-In y así identificar las posibles trampas que estos realizar para marcar su asistencia en clase.

9.2 Versión de la aplicación para gestores de programa

Con esta mejora, se buscará crear una nueva versión donde los gestores de programa puedan realizar la mayoría de acciones que hacen en el campus virtual como:

- **Poder ver las asignaturas de las que es encargado:** Disponer de un apartado para visualizar las asignaturas de las que es encargado y poder gestionarlas.
- **Matricular participantes:** Disponer de un apartado para matricular participantes dentro de un curso o una asignatura.
- **Validar notas:** Disponer de un apartado para poder validar las notas que el profesor ha puesto en las tareas y así notificar a los alumnos.
- **Crear asignaturas:** Disponer de un apartado donde, usando la guía metodológica proporcionada por el profesor, crear las asignaturas con los materiales correspondientes.
- **Crear aulas virtuales:** Disponer de un apartado donde, después de crear una asignatura en el campus, poder gestionar las aulas virtuales.

9.3 Versión de la aplicación para exalumnos

Con esta mejora, se buscará crear una nueva versión donde los exalumnos puedan acceder a los contenidos que dispongan. Realmente, el perfil de un exalumno es el mismo que el de un alumno con la diferencia de que, un alumno dispone de un curso activo y el exalumno no. Sabiendo esto, las posibles mejoras son:

- **Selección de cursos:** Permitir al usuario poder seleccionar entre los cursos que ha realizado en EADA y así acceder al contenido.
- **Sección de ofertas:** Disponer de un apartado donde ofrecer ofertas de diferentes cursos que puedan interesar al usuario.

Como podemos ver, las mejoras que se han explicado están separadas por el tipo de rol de un usuario. Esto nos permite trabajar de forma más ordenada dando prioridad a un perfil u otro. Debido a la importancia que tienen los profesores, será este perfil de usuario en el que nos centraremos en una futura versión de la aplicación.

7. Conclusiones

Personalmente, la realización de este Trabajo final de Grado representó un desafío, ya que me permitió poner a prueba todos los conocimientos que he ido adquiriendo durante la realización del Grado de Informática sumados a los años de experiencia laboral.

A pesar de que ya había desarrollado aplicaciones para dispositivos Android, el reto de utilizar una plataforma como Apache Cordova^[2], la cual me permitía ampliar conocimientos en lenguajes de programación como CSS3, HTML5 y JavaScript, me resultó motivador, además de ser un valor añadido para mi currículum.

En cuanto a los objetivos planteados al principio, se han cumplido con creces. La aplicación logró pasar todos los filtros impuestos por la Play Store y Apple Store, dejando vía libre para poder publicar la nueva versión en el momento del lanzamiento del Campus Virtual. Los tiempos de desarrollo también se han cumplido, a pesar de los contratiempos, hemos sabido trabajar en equipo y solucionar todos los inconvenientes que han ido surgiendo durante el desarrollo. He de recalcar que, gracias a la buena planificación y a la comunicación del equipo, estos inconvenientes fueron mínimos.

También he de mencionar que, gracias a este trabajo, he podido realizar un proyecto en el que estaban involucrados más desarrolladores, haciendo que la comunicación fuese fundamental en el correcto desarrollo de la aplicación. Anteriormente, me había hecho cargo de proyectos en los que era el único responsable en desarrollar la aplicación. La oportunidad de colaborar en la realización de este proyecto ha sido gratificante ya que me he encontrado muy cómodo trabajando en equipo.

Para finalizar, me muestro muy satisfecho del resultado final de la aplicación. Es un producto con mucho potencial, atractivo e intuitivo en cuanto a diseño y está preparado para las futuras mejoras que el departamento tiene pensado aplicar. También he de recalcar que, las opiniones del resto de departamentos sobre esta aplicación han sido más que satisfactorias y todos esperan con ansia su despliegue.

8. Bibliografía

[1] “EADA Business School Barcelona” [Online] Disponible: <https://www.eada.edu/> [Acceso: 2/12/2019]

[2] “Apache Cordova” [Online] Disponible: https://es.wikipedia.org/wiki/Apache_Cordova [Acceso: 2/12/2019]

[3] “Web Service (REST-full)” [Online] Disponible: <https://www.codigonaranja.com/2018/restful-web-service> [Acceso: 2/12/2019]

[4] “Requisito no funcional” [Online] Disponible: https://es.wikipedia.org/wiki/Requisito_no_funcional [Acceso: 15/12/2019]

[5] “Modelo-Vista-Controlador” [Online] Disponible: <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador> [Acceso: 20/12/2019]

[6] “Apache” [Online] Disponible: <https://www.hostinger.com.ar/tutoriales/que-es-apache> [Acceso: 4/1/2020]

[7] “Jboss” [Online] Disponible: <https://es.wikipedia.org/wiki/WildFly> [Acceso: 9/1/2020]

[8] “Tomcat” [Online] Disponible: <https://es.wikipedia.org/wiki/Tomcat> [Acceso: 9/1/2020]

[9] “Servlets” [Online] Disponible: https://es.wikipedia.org/wiki/Java_Servlet [Acceso: 10/1/2020]

[10] “Jabber” [Online] Disponible: <https://www.genbeta.com/a-fondo/que-es-jabber-y-por-que-usarlo> [Acceso: 15/1/2020]

[11] “desarrollo en cascada” [Online] Disponible: https://es.wikipedia.org/wiki/Desarrollo_en_cascada [Acceso: 16/1/2020]

[12] “**Jersey** en conjunto con **Jackson**” [Online] Disponible:

<https://examples.javacodegeeks.com/enterprise-java/rest/jersey/json-example-with-jersey-jackson/> [Acceso: 16/1/2020]