



EduAgend@

**Nombre Estudiante: Ángela Bañuls Serrano**

Máster Universitario de Desarrollo de Aplicaciones para Dispositivos Móviles

**Nombre Consultor: Francesc D'Assís Giralt Queralt**

**Profesor responsable de la asignatura: Carles Garrigues Olivella**

Fecha de entrega; 07/06/2017



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

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	EduAgend@
<b>Nombre del autor:</b>	Ángela Bañuls Serrano
<b>Nombre del consultor:</b>	Francesc D'Assís Giralt Queralt
<b>Fecha de entrega (mm/aaaa):</b>	06/2017
<b>Titulación:</b>	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>
<b>Resumen del Trabajo (máximo 250 palabras):</b>	
<p>Este documento describe el proceso de desarrollo de una aplicación híbrida, utilizando el framework Ionic, que permite a sus usuarios (alumnos/docentes) gestionar una agenda virtual en el sector educativo.</p> <p>Esta aplicación, EduAgend@, pretende dar solución en el sector educativo no solo a la planificación de las tareas, sino ofrecer una herramienta de comunicación entre el docente y sus alumnos.</p>	
<b>Abstract (in English, 250 words or less):</b>	
<p>This document describes the process of developing a hybrid application, using the Ionic framework, which allows its users (students / teachers) to manage a virtual agenda in the education sector.</p> <p>This application, EduAgend @, aims to provide a solution in the education sector not only to the planning of the tasks, but also to offer a communication tool between the teacher and his students.</p>	
<b>Palabras clave (entre 4 y 8):</b>	
agenda escolar digital alumno docente	

# Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	2
1.3 Enfoque y método seguido.....	3
1.4 Planificación del Trabajo.....	3
1.5 Breve resumen de productos obtenidos.....	8
1.6 Breve descripción de los otros capítulos de la memoria.....	8
2. Diseño Centrado en el Usuario.....	9
2.1 Análisis.....	9
2.2 Diseño.....	10
2.3 Diseño del prototipo.....	13
2.4 Evaluación del prototipo.....	18
3. Diseño técnico.....	20
3.1 Casos de uso.....	20
.....	20
3.2 Diseño de la Arquitectura.....	28
3.3 Modelo de datos.....	30
4. Implementación.....	32
4.1 Hardware utilizado.....	32
4.2 Software empleado.....	32
4.3 Pruebas.....	33
5. Conclusiones.....	34
6. Glosario.....	35
7. Bibliografía y referencias.....	36
8. Anexos.....	37
Anexo 1 - Manual de Instalación.....	37
Anexo 2 - Manual de Usuario.....	40

## Lista de figuras

Ilustración 1: Diagrama de Gantt: Proyecto completo (Tabla).....	5
Ilustración 2: Diagrama de Gantt: Proyecto completo (Gráfico).....	5
Ilustración 3: Diagrama de Gantt: Plan de Trabajo (Tabla).....	5
Ilustración 4: Diagrama de Gantt: Plan de Trabajo (Gráfico).....	5
Ilustración 5: Diagrama de Gantt: Diseño (Tabla).....	6
Ilustración 6: Diagrama de Gantt: Diseño (Gráfico).....	6
Ilustración 7: Diagrama de Gantt: Implementación (Tabla).....	6
Ilustración 8: Diagrama de Gantt: Implementación (Gráfico).....	7
Ilustración 9: Diagrama de Gantt: Entrega final (Tabla).....	7
Ilustración 10: Diagrama de Gantt: Entrega final (Gráfico).....	7
Ilustración 11: Árbol de Navegación del usuario alumno.....	13
Ilustración 12: Árbol de Navegación del usuario docente.....	14
Ilustración 13: Diagrama UML Usuario no Identificado.....	20
Ilustración 14: Diagrama UML usuario alumno.....	20
Ilustración 15: Diagrama UML Usuario Docente.....	21
Ilustración 16: Esquema de IONIC.....	28
Ilustración 17: Arquitectura de una aplicación desarrollada con Angular 2.....	29
Ilustración 18: Logotipo de Firebase.....	29
Ilustración 19: Firebase - Sincronización de un dispositivo con el resto.....	30
Ilustración 20: Esquema relacional de los datos.....	30
Ilustración 21: Aplicación EduAgenda - Testeo en el navegador.....	33
Ilustración 22: Creación de un proyecto en Firebase.....	37
Ilustración 23: Configuración de un proyecto en Firebase.....	38
Ilustración 24: EduAgenda - Pantalla de Login.....	40
Ilustración 25: EduAgenda - Menu alumno.....	40
Ilustración 26: EduAgenda - Pantalla de Registro.....	41
Ilustración 27: Eduagenda. Pantalla del calendario en vista mensual.....	42
Ilustración 28: Eduagenda. Pantalla del calendario en vista semanal.....	42
Ilustración 29: EduAgenda - Nuevo evento en modo alumno.....	43
Ilustración 30: EduAgenda - Vista del calendario.....	44
Ilustración 31: EduAgenda - Nuevo Evento en modo docente.....	46

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

Las agendas son unas herramientas muy útiles para planificar de forma óptima la secuencia de tareas, actividades, etc que se deben realizar en un período determinado. De esta forma, nos permiten una correcta gestión del tiempo disponible y no olvidar ningún evento importante.

En la actualidad, podemos encontrar bastantes aplicaciones de agendas digitales para dar solución a la planificación de tareas en todos los sectores. La aplicación que se va a desarrollar, EduAgend@, pretende dar solución en el sector educativo no solo a la planificación de las tareas, sino ofrecer una herramienta de comunicación entre el docente y sus alumnos.

Se pretende que sea una herramienta de apoyo no solo para el alumnado, sino también para el profesor en cursos de formación para adultos.

Tras realizar un análisis de las aplicaciones disponibles en el mercado, se observa que existen bastante aplicaciones enfocadas hacia este sector, pero siempre están orientadas hacia el estudiante o profesor de forma separada y no se han tratado de forma conjunta.

EduAgend@, pretende ser una agenda para el alumno y profesor, que permita su uso en de forma conjunta: profesor/alumnos. En este modo de trabajo el alumno obtendría las siguientes ventajas:

- Gestión de calificaciones sin necesidad de introducirlos de forma manual. El profesor cuando introduzca las notas en su agenda se sincronizarán con la de los alumnos.
- Gestión de fechas importantes: exámenes, entrega de tareas de forma automática, una vez que el profesor realice la anotación.

Por otra parte, el profesor también obtendrá una serie de beneficios derivados de esta forma de trabajo:

- Comunicación directa con sus alumnos: En el caso de tener que avisar de algún hecho importante podrá enviar mensajes en tiempo real.

## **1.2 Objetivos del Trabajo**

El objetivo principal de este Trabajo Final de Máster (TFM) es desarrollar una aplicación para móvil que realice las funciones de una agenda escolar digital y permita una comunicación entre el docente y sus alumnos.

### **1.2.1 Objetivos generales**

El Trabajo Final de Máster (TFM) contribuye a alcanzar los siguientes objetivos generales:

- Poner en práctica los conocimientos adquiridos a lo largo de todo el máster.
- Saber definir una planificación realista para un proyecto, teniendo en cuenta su alcance y los recursos disponibles.
- Analizar el mercado en búsqueda de opciones que cubran esta idea o similar, recogiendo que ofrecen y detectando donde fallan o que perfil de usuario no cubren
- Ser capaz de documentar y justificar las decisiones tomadas en el desarrollo y los resultados conseguidos.

### **1.2.2 Objetivos específicos**

La aplicación estará dirigida a dos tipos de usuarios: "alumno" y "profesor".

Ambos usuarios podrán realizar las siguientes acciones:

- Introducir las asignaturas de las que está matriculado o imparte docencia.
- Introducir el horario lectivo de cada una de estas asignaturas.
- Incluir anotaciones en la agenda.
- Mostrar tareas pendientes y avisos de eventos.
- Enviar un mensaje privado.
- Sincronizar los datos con la plataforma Moodle

Los usuarios de tipo "profesor" dispondrán de las siguientes opciones:

- Incluir fechas de exámenes para cada una de las asignaturas en las que imparte docencia.
- Incluir tareas a realizar por el alumno definiendo una fecha de inicio/fin.
- Calificar tareas.
- Planificar las tareas pendientes marcadas por un orden de preferencia.

Los usuarios de tipo "alumno" podrán realizar las siguientes acciones:

- Planificar las tareas pendientes marcadas por un orden de prioridad.
- Marcar las tareas como entregadas/finalizadas. Se gamificará todo el proceso de realización de tareas.

### 1.3 Enfoque y método seguido

Para llevar a cabo el trabajo se creará un producto nuevo, añadiendo todas las funcionalidades indicadas en el apartado anterior. Se ha elegido esta opción, ya que no existe en la actualidad un producto que cumpla con los objetivos marcados.

Para el desarrollo de la aplicación se ha elegido una metodología en cascada, ya que cada fase del desarrollo de la aplicación viene fijada por la fecha de entrega de cada Prueba de Evaluación Continua (PEC).

### 1.4 Planificación del Trabajo

La planificación del trabajo se ha realizado de acuerdo a los hitos establecidos en cada una de las Pruebas de Evaluación Continua.

El proyecto se ha dividido en cinco tareas principales:

- **Plan de Trabajo:** Se definen los objetivos del trabajo final y el plan de trabajo a seguir para lograrlos



- **Formación:** Se han destinado 37 horas para la formación. Se pretende ampliar los conocimientos de los frameworks Angular 2 e IONIC. Esta tarea se realizará de forma paralela a la tarea de Diseño, y se realizarán los siguientes cursos online de la plataforma Udemy:
  - **Angular 4 (formerly Angular 2) - The Complete Guide (22,5 h) :** <https://www.udemy.com/the-complete-guide-to-angular-2/>
  - **Ionic 2 - The Practical Guide to Building iOS & Android Apps . (14,5h) :** <https://www.udemy.com/ionic-2-the-practical-guide-to-building-ios-android-apps/>
  
- **Diseño:** Se realizará el diseño centrado en el usuario, así como el diseño técnico de la aplicación.
  
- **Implementación:** Se centrará en el desarrollo de la aplicación móvil, así como el conjunto de pruebas para definir el correcto funcionamiento de la aplicación
  
- **Entrega Final:** Se finalizará toda la documentación del proyecto, es decir, la memoria del proyecto y la memoria del usuario. Además, se realizará un vídeo de presentación del proyecto.

Para las tres tareas iniciales se ha establecido una dedicación diaria de lunes a viernes de 2 horas y una dedicación de 10 horas los fines de semana. La última tarea, entrega final, se ha planificado asignando una dedicación diaria de 1 hora de lunes a viernes y 5 horas los fines de semana.

GANTT project						
Nombre	Fecha de inicio	Fecha de fin	N Horas	M Horas	Total Horas	
* Plan de Trabajo	4/03/17	15/03/17	7	8	15	
o Formacion	16/03/17	5/04/17	19	18	37	
o Diseño	16/03/17	5/04/17	30	30	60	
o Implementación	6/04/17	17/05/17	60	60	120	
o Entrega Final	18/05/17	7/06/17	0	0	50	

Ilustración 1: Diagrama de Gantt: Proyecto completo (Tabla)

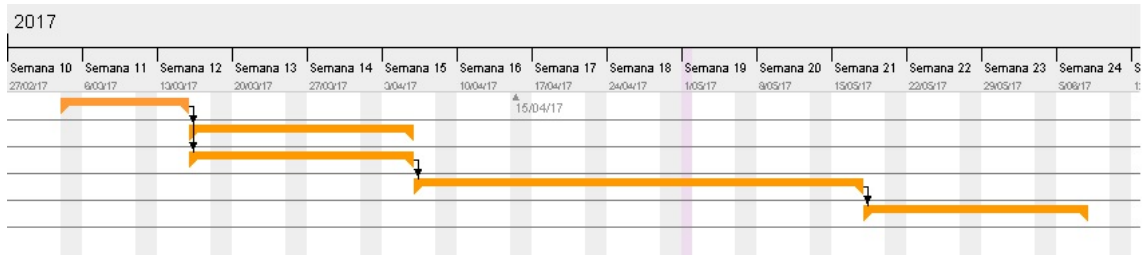


Ilustración 2: Diagrama de Gantt: Proyecto completo (Gráfico)

GANTT project						
Nombre	Fecha de inicio	Fecha de fin	N Horas	M Horas	Total Horas	
* Plan de Trabajo	4/03/17	15/03/17	7	8	15	
o Contexto y justificación del trabajo	4/03/17	5/03/17	0	3	3	
o Objetivos del trabajo	6/03/17	9/03/17	4	0	4	
o Enfoque y método elegido	10/03/17	10/03/17	1	0	1	
o Planificación del trabajo	11/03/17	13/03/17	1	5	6	
o Breve resumen de productos obtenidos	14/03/17	15/03/17	1	0	1	
o Formacion	16/03/17	5/04/17	19	18	37	
o Diseño	16/03/17	5/04/17	30	30	60	
o Implementación	6/04/17	17/05/17	60	60	120	
o Entrega Final	18/05/17	7/06/17	0	0	50	

Ilustración 3: Diagrama de Gantt: Plan de Trabajo (Tabla)

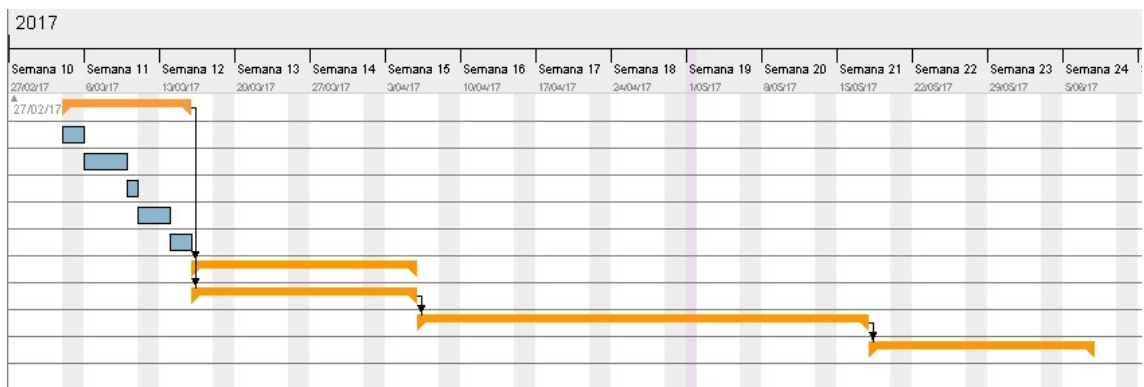


Ilustración 4: Diagrama de Gantt: Plan de Trabajo (Gráfico)

GANTT project						
Nombre	Fecha de inicio	Fecha de fin	N Horas	M Horas	Total Horas	
✚ * Plan de Trabajo	4/03/17	15/03/17	7	8	15	
⊕ • Formación	16/03/17	5/04/17	19	18	37	
⊖ • Diseño	16/03/17	5/04/17	30	30	60	
⊖ • Diseño centrado en el usuario	16/03/17	22/03/17	10	10	20	
• Usuarios y contexto de uso	16/03/17	17/03/17	3	0	3	
• Diseño conceptual	17/03/17	18/03/17	1	5	6	
• Prototipado	19/03/17	21/03/17	4	5	9	
• Evaluación	22/03/17	22/03/17	2	0	2	
⊖ • Diseño técnico	23/03/17	4/04/17	0	0	38	
• Casos de uso	23/03/17	26/03/17	4	10	14	
• Diseño de la arquitectura	27/03/17	4/04/17	14	10	24	
• Documentación	5/04/17	5/04/17	2	0	2	
⊕ • Implementación	6/04/17	17/05/17	60	60	120	
⊕ • Entrega Final	18/05/17	7/06/17	0	0	50	

Ilustración 5: Diagrama de Gantt: Diseño (Tabla)

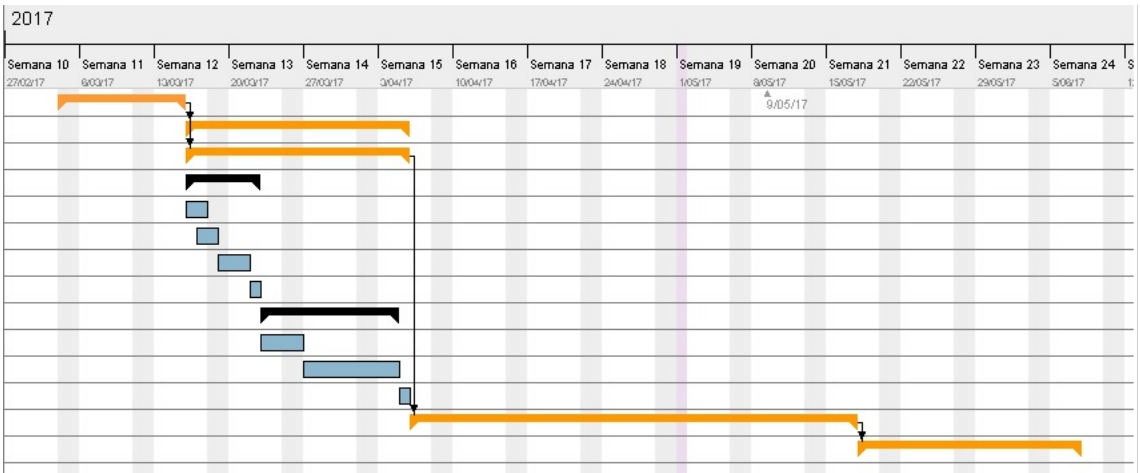


Ilustración 6: Diagrama de Gantt: Diseño (Gráfico)

GANTT project						
Nombre	Fecha de inicio	Fecha de fin	N Horas	M Horas	Total Horas	
✚ * Plan de Trabajo	4/03/17	15/03/17	7	8	15	
⊕ • Formación	16/03/17	5/04/17	19	18	37	
⊕ • Diseño	16/03/17	5/04/17	30	30	60	
⊖ • Implementación	6/04/17	17/05/17	60	60	120	
• Desarrollo	6/04/17	8/05/17	45	50	95	
• Pruebas unitarias	8/05/17	14/05/17	9	6	15	
• Documentación	14/05/17	17/05/17	6	4	10	
⊕ • Entrega Final	18/05/17	7/06/17	0	0	50	

Ilustración 7: Diagrama de Gantt: Implementación (Tabla)

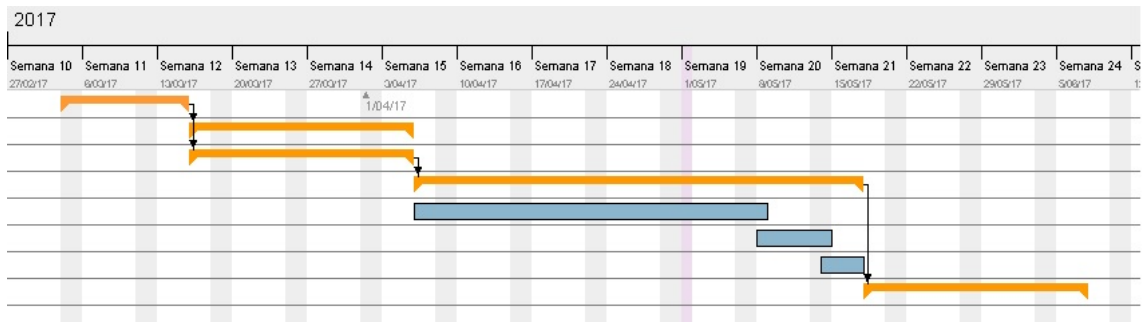


Ilustración 8: Diagrama de Gantt: Implementación (Gráfico)

Nombre	Fecha de inicio	Fecha de fin	N Horas	M Horas	Total Horas
✚ * Plan de Trabajo	4/03/17	15/03/17	7	8	15
⊕ ◦ Formación	16/03/17	5/04/17	19	18	37
⊕ ◦ Diseño	16/03/17	5/04/17	30	30	60
⊕ ◦ Implementación	6/04/17	17/05/17	60	60	120
⊖ ◦ Entrega Final	18/05/17	7/06/17	0	0	50
◦ Documentación	18/05/17	28/05/17	14	20	34
◦ Vídeo y presentación	29/05/17	7/06/17	8	8	16

Ilustración 9: Diagrama de Gantt: Entrega final (Tabla)

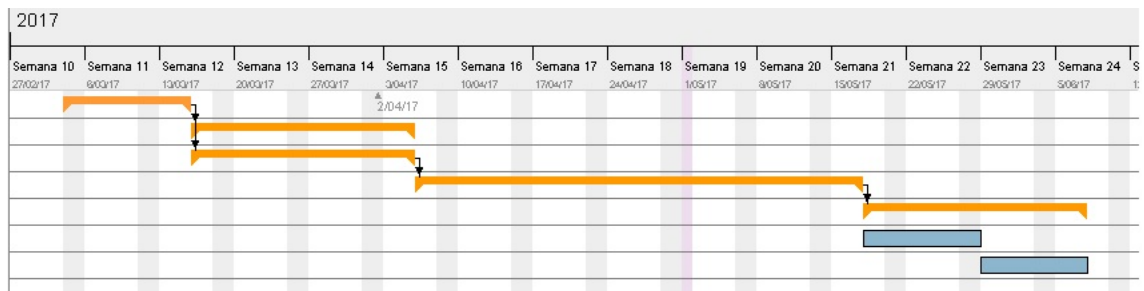


Ilustración 10: Diagrama de Gantt: Entrega final (Gráfico)

## 1.5 Breve resumen de productos obtenidos

Al finalizar este proyecto se obtendrán los siguientes productos:

- Aplicación móvil que permitirá gestionar la agenda escolar de un alumno/docente. Esta aplicación estará disponible para Android e iOS.
- Memoria del proyecto.
- Manual de usuario.
- Vídeo de presentación del proyecto.

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

La memoria del proyecto incluye los siguientes capítulos:

- **Análisis:** Se detallan las características de los usuarios, sus necesidades, objetivos y el contexto de uso.
- **Diseño conceptual:** A partir del análisis realizado se generan diferentes escenarios de uso.
- **Diseño del prototipo:** Incluye el árbol de navegación de la aplicación, y los prototipos de bajo y alto nivel.
- **Diseño técnico de la aplicación:** Se definen los casos de uso y la arquitectura de la aplicación.
- **Desarrollo:** Recoge todos los problemas encontrados y soluciones aportadas durante la implementación de la aplicación, así como el conjunto de pruebas diseñado para probar su correcto funcionamiento.
- **Conclusiones:** Incluye una reflexión acerca de los objetivos logrados, así como un análisis del seguimiento de la planificación y metodología aplicados.

## 2. Diseño Centrado en el Usuario

El Diseño Centrado en el Usuario (DCU) es un enfoque de diseño cuyo proceso está dirigido por información sobre las personas que van a hacer uso del producto. El principal objetivo es crear productos que resuelvan sus necesidades, consiguiendo la mejor experiencia de uso con el mínimo esfuerzo por parte de los usuarios finales.

### 2.1 Análisis

EduAgend@ es una aplicación móvil desarrollada para el sector educativo que será utilizada por dos tipos de usuario: docentes y alumnos.

Los docentes utilizarán la aplicación para realizar las siguientes tareas:

- Registrar las asignaturas de las que imparte módulo.
- Incluir fechas de exámenes para su asignatura.
- Añadir tareas a realizar por el alumno.
- Calificar las tareas entregadas por sus alumnos.

Los alumnos dispondrán de las siguientes funciones:

- Inscribirse a una asignatura.
- Visualizar tareas de entrega pendiente.
- Marcar tareas como finalizadas.
- Mostrar calificaciones

Ambos usuarios tendrán la posibilidad de :

- Incluir anotaciones en la agenda
- Registrar el horario semanal de las asignaturas de las que imparte o recibe clases.
- Añadir tareas.
- Planificar la ejecución de tareas.

## 2.2 Diseño

### 2.2.1 Modelado del usuario

Para la definición de los usuarios potenciales de nuestra aplicación, necesidades y contextos de uso se han desarrollado unas fichas de usuario y escenario de uso que se adjuntan a continuación.

Ficha 1	
<b>Nombre:</b>	Ángela Bañuls
<b>Sexo:</b>	Mujer
<b>Edad:</b>	38 años
<b>Estado social:</b>	Casada
<b>Estado laboral:</b>	Activo
<b>Estudios académicos:</b>	Ingeniería Técnica en Informática de Gestión
<b>Idiomas que domina:</b>	Castellano, Inglés.
<b>Experiencia laboral:</b>	6 años de analista/programadora 10 años de profesora en ciclos formativos de Informática.
<b>Habilidad en tecnología:</b>	Muy buena
<b>Descripción del usuario:</b>	
Ángela es una mujer que trabaja en un centro como profesora de ciclos formativos de informática. En el centro dispone de acceso a Internet, a través de WiFi, en todas las aulas e instalaciones. Sus clases son de carácter eminentemente práctico, y para cada asignatura asigna varias tareas de carácter práctica, y les asigna una fecha máxima de entrega.	
<b>Descripción del escenario:</b>	
Ángela se encuentra en la sala de profesores preparando sus clases. Accede a EduAgend@ y consulta las actividades que tiene pendiente de corrección. A medida que corrige cada actividad incluye la nota de cada alumno y automáticamente la aplicación sincroniza el dato con la agenda del alumno que recibirá una notificación indicando la nota obtenida.	

Ficha 2	
<b>Nombre:</b>	David
<b>Sexo:</b>	Hombre
<b>Edad:</b>	17 años
<b>Estado social:</b>	Soltero
<b>Estado laboral:</b>	Paro
<b>Estudios académicos:</b>	Graduado en ESO
<b>Idiomas que domina:</b>	Castellano.
<b>Experiencia laboral:</b>	Sin experiencia
<b>Habilidad en tecnología:</b>	Buena
<b>Descripción del usuario:</b>	
David es un chico que se encuentra cursando el ciclo de informática de grado medio "Sistemas Microinformáticos y Redes". Actualmente cursa primero.	
<b>Descripción del escenario:</b>	
David se encuentra en casa planificando sus tareas y accede a la agenda para visualizar las tareas que todavía tiene pendientes. En base a la fecha de entrega y su prioridad, la agenda le ayuda a realizar una óptima planificación.	

Podemos distinguir dos usuarios potenciales de la aplicación: alumnos y docentes.

#### **Perfil de Usuario: Alumno**

Normalmente su edad estará comprendida entre 15-25 años, ya que esta aplicación está pensada principalmente para estudiantes de ciclos formativos. Este perfil de usuario no tendrán ningún problema con el manejo de la aplicación, ya que están muy acostumbrados al uso de aplicaciones móviles y nuevas tecnologías.

#### **Perfil de Usuario: Docentes**

Podemos encontrar distintas franjas de edad:

- De 25 a 50 años: Están bastante acostumbrados al uso de aplicaciones móviles y se adaptarán a la aplicación con un breve período de aprendizaje.
- Mayores de 50 años: No están muy acostumbrados al uso de smartphones y la aplicación debe ser lo más sencilla e intuitiva posible para facilitar su uso.



La aplicación se utilizará en los siguientes contextos :

- Los alumnos usarán la aplicación para consultar el horario, realizar anotaciones y verificar tareas durante las clases. También harán uso de ella desde casa o la biblioteca para planificar la realización de sus tareas.
- Los docentes, al igual que los alumnos, harán uso de la aplicación durante las clases y fuera de ellas cuando estén preparando sus clases, bien para consultar las tareas pendientes de corrección, agregar notas, etc.

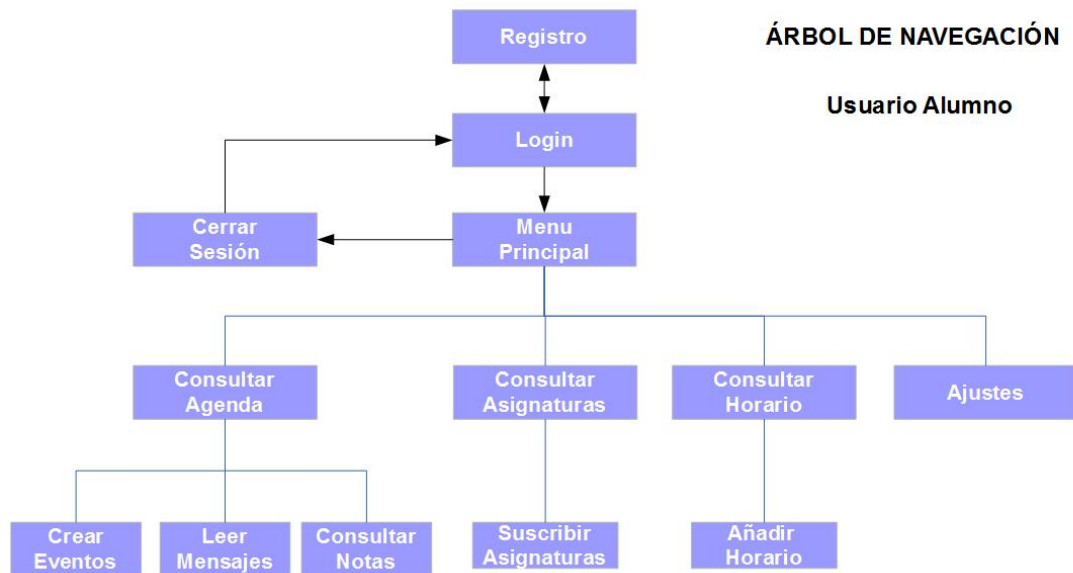
## 2.3 Diseño del prototipo

A partir de las necesidades detectadas durante el proceso de diseño conceptual se procede a efectuar el diseño del prototipo. Para la realización de este proceso debe quedar previamente establecido el árbol de navegación de la aplicación.

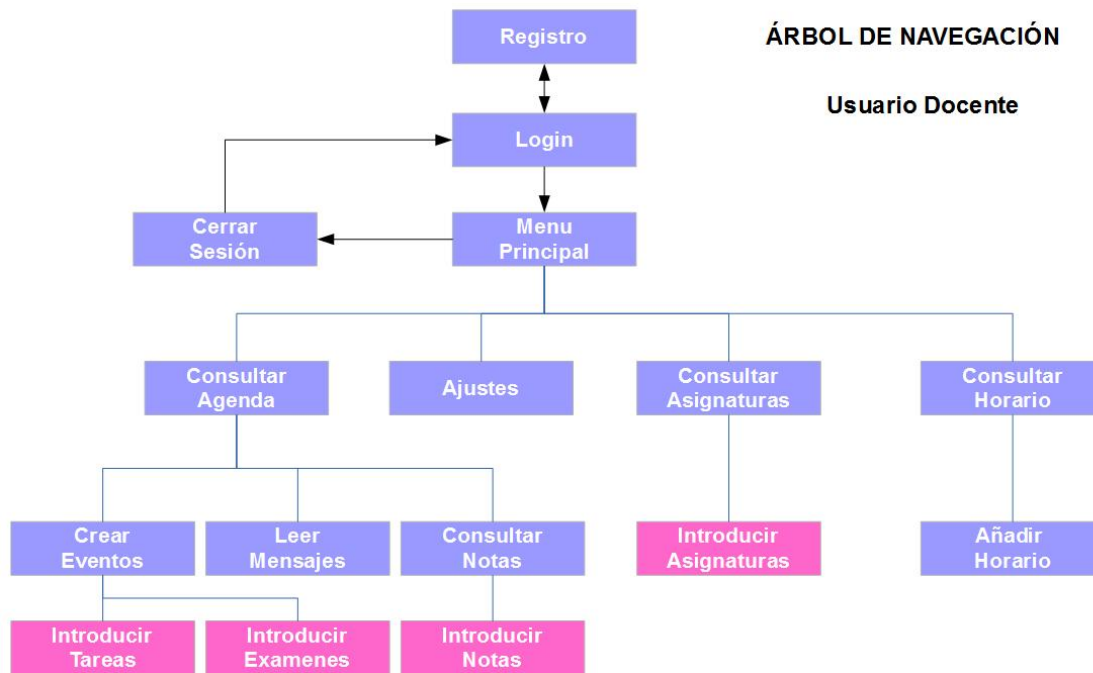
### 2.3.1 El árbol de navegación

El árbol de navegación es un esquema que representa de forma muy sencilla la navegación en la aplicación.

Tal y como se ha indicado en la fase de análisis, existen dos tipos de usuario: alumno y docente. Cada uno de ellos dispone de acceso a unas determinadas funciones, reflejadas en los siguientes árboles de navegación.



*Ilustración 11: Árbol de Navegación del usuario alumno*



*Ilustración 12: Árbol de Navegación del usuario docente*

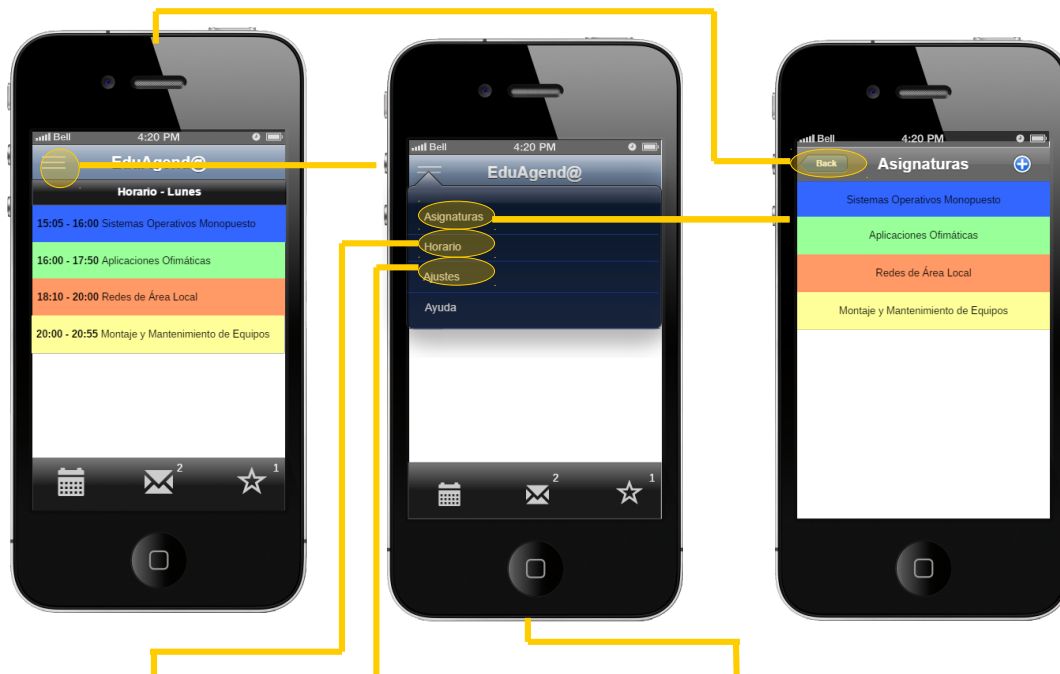
Ambos usuarios tienen acceso a las siguientes funciones:

- **Consultar Agenda:** En esta pantalla se visualiza la agenda del usuario. El usuario dispone de la función de crear un nuevo evento de tipo personal, así como de leer los mensajes recibidos de otros alumnos/docentes y consultar las notas de las tareas y exámenes recibidos. El usuario de tipo docente dispone además de las opciones de introducir tareas y exámenes para las asignaturas en las que imparte docencia.
- **Ajustes:** En esta pantalla el usuario puede configurar varios ajustes de la aplicación, tales como la notificación de mensajes.
- **Consultar Asignaturas:** El usuario puede consultar las asignaturas a las que se encuentra suscrito, en el caso del alumno, y las asignaturas que imparte docencia, en el caso del docente.
- **Introducir Asignaturas:** El usuario docente puede agregar asignaturas.
- **Suscribirse a una asignatura:** El usuario alumno se puede suscribir a las asignaturas impartidas por un profesor.
- **Consultar Horario:** Ambos usuarios pueden consultar el horario definido.
- **Añadir Horario:** Ambos usuarios pueden crear un horario semanal.

### 2.3.2 Prototipo de la aplicación

En este apartado se mostrará un prototipo de la aplicación, distinguiendo cada una de las pantallas más importantes de la aplicación para los dos perfiles definidos anteriormente: docente y alumno.

#### Modo Alumno/Docente



## Modo Alumno



## Modo Docente



## 2.4 Evaluación del prototipo

La evaluación es un proceso fundamental en el diseño de un sistema centrado en el usuario. Su objetivo principal es lograr que el usuario pueda interactuar con la aplicación sin ayuda externa y logrando una experiencia satisfactoria.

Para el proceso de evaluación se ha tenido en cuenta la opinión de alumnos y docentes acerca de la interfaz y la usabilidad de la aplicación. Se les ha proporcionado el prototipo de la aplicación y se les ha pedido a los alumnos que intentasen realizar las siguientes acciones:

- Suscribirse a una asignatura.
- Crear una tarea nueva en la agenda.
- Visualizar los eventos de la agenda.
- Consultar las notas obtenidas en las tareas.

Por otro lado, se le ha solicitado a los profesores que intentasen realizar las siguientes acciones:

- Añadir una asignatura.
- Crear un evento de tipo examen en la agenda.
- Crear un evento de tipo tarea.
- Asignar notas a una tarea determinada.

Una vez concluido el proceso se les ha indicado que respondan a las siguientes preguntas::

- ¿Ha podido realizar todas las tareas sin ayuda?
- ¿Los iconos utilizados durante la aplicación son concretos y familiares para el usuario?
- ¿La navegación entre los diferentes elementos de la aplicación resulta satisfactoria? ¿Se ha sentido desorientado en alguna pantalla?
- ¿Le ha gustado el diseño o lo ha encontrado excesivamente recargado?
- ¿Ha podido acceder a las opciones de la aplicación sin problemas? ¿O por el contrario ha encontrado algún elemento demasiado pequeño para sus dedos?
- ¿Ha encontrado alguna de las acciones realizadas demasiado lenta o costa en el proceso?
- ¿Cómo se podría mejorar la herramienta?



# 3 Diseño técnico

## 3.1 Casos de uso

A continuación se definen los casos de uso que permitirán identificar el comportamiento de la aplicación.

En los diagramas se pueden observar tres actores: un usuario no identificado , el usuario alumno y el usuario docente.

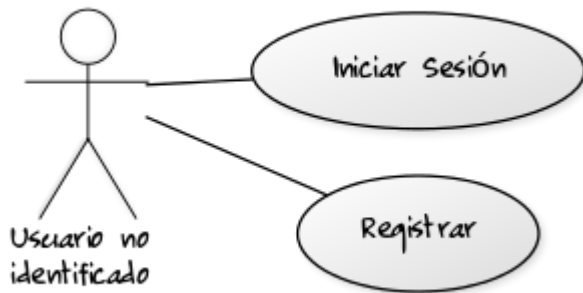


Ilustración 13: Diagrama UML Usuario no Identificado

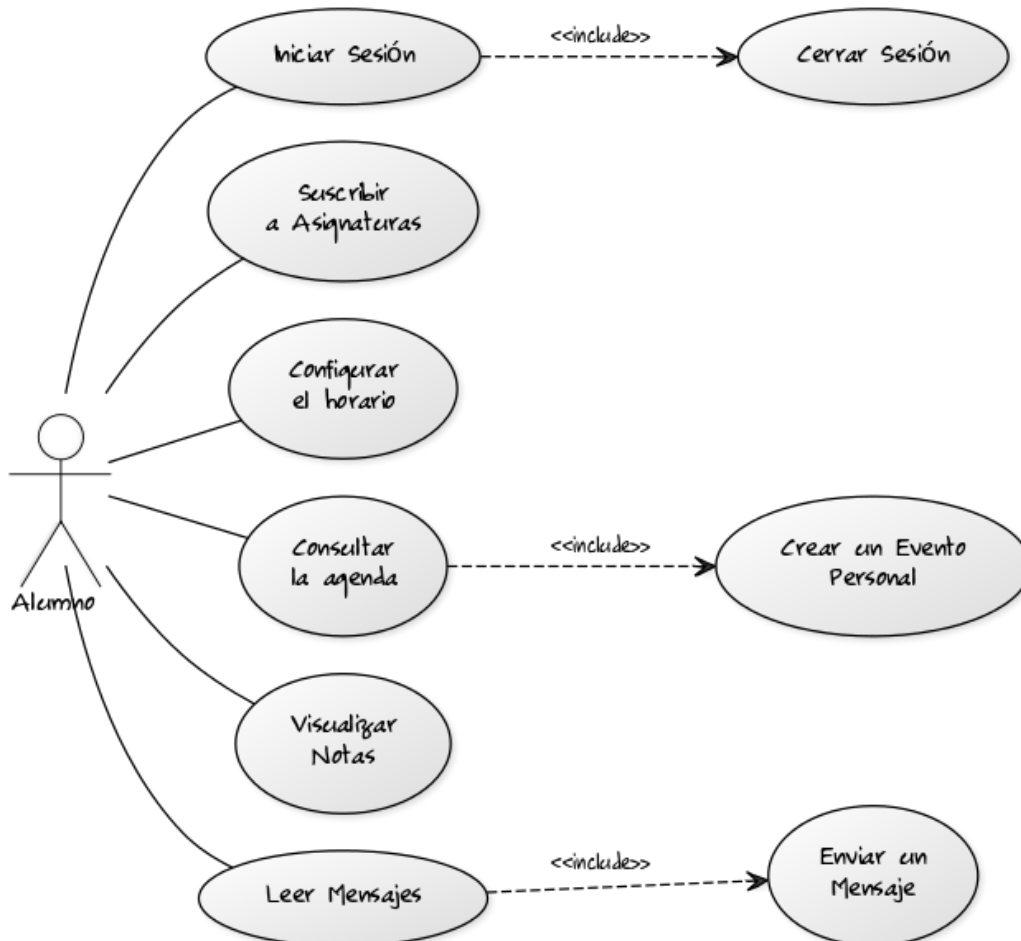


Ilustración 14: Diagrama UML usuario alumno

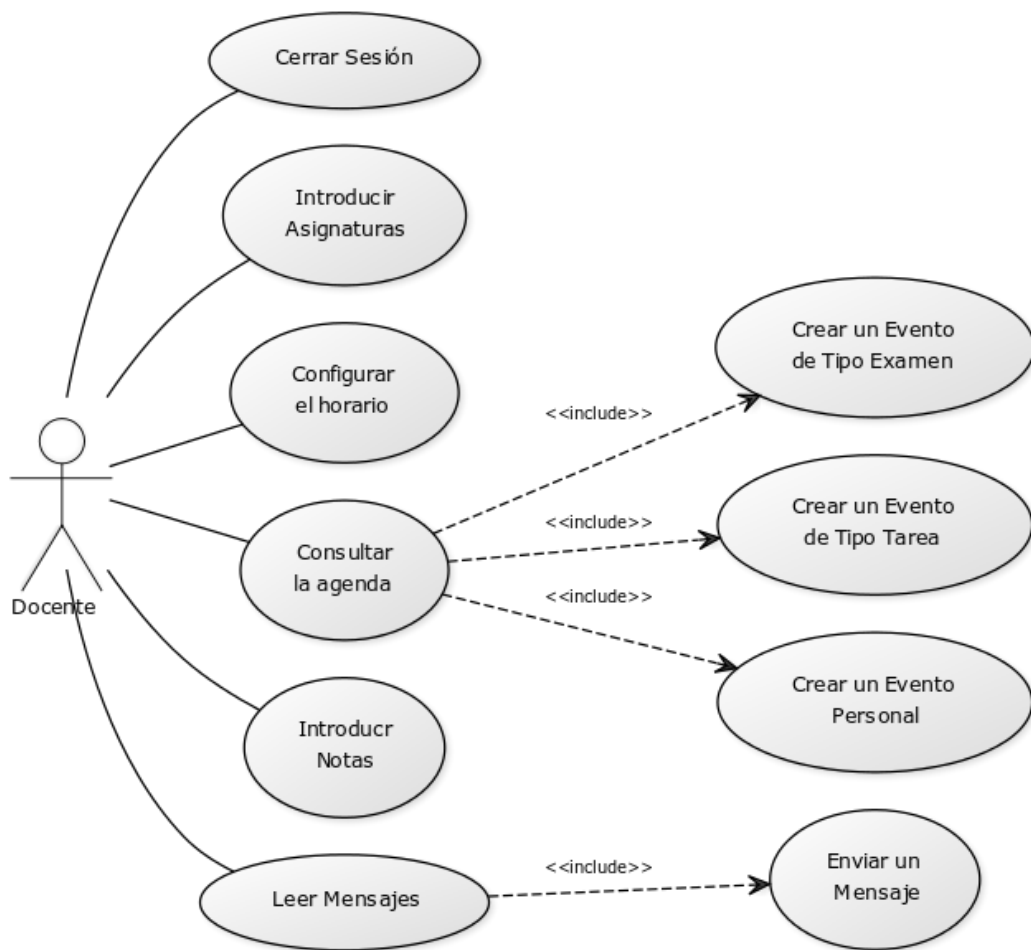


Ilustración 15: Diagrama UML Usuario Docente

Caso de uso 1	Iniciar Sesión
<b>Descripción</b>	Acción que permite al usuario iniciar sesión en el sistema
<b>Actor</b>	Usuario no identificado
<b>Precondiciones</b>	No estar autenticado en el sistema a través de la aplicación.
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario introduce un usuario y una contraseña.</li> <li>2. El sistema comprueba la validez del usuario y la contraseña.</li> <li>3. El usuario accede a la pantalla inicial de la agenda.</li> </ol>
<b>Postcondiciones</b>	El usuario ha iniciado sesión correctamente.

<b>Caso de uso 2</b>	<b>Registrar un usuario</b>
<b>Descripción</b>	Acción que permite registrar un usuario.
<b>Actor</b>	Usuario no identificado.
<b>Precondiciones</b>	No estar autenticado en el sistema a través de la aplicación.
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario introduce su nombre, cuenta de e-mail y una contraseña.</li> <li>2. El sistema comprueba que no exista en el sistema una cuenta con el mismo e-mail introducido.</li> <li>3. El usuario accede a la pantalla de login.</li> </ol>
<b>Postcondiciones</b>	Se ha registrado una cuenta de usuario.

<b>Caso de uso 3</b>	<b>Cerrar Sesión</b>
<b>Descripción</b>	Acción que permite cerrar una sesión establecida
<b>Actor</b>	Alumno/Docente
<b>Precondiciones</b>	Estar autenticado en el sistema a través de la aplicación.
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario seleccionará la opción del menú principal "Cerrar Sesión".</li> <li>2. La aplicación mostrará la pantalla de inicio de sesión.</li> </ol>
<b>Postcondiciones</b>	No hay ninguna sesión iniciada.

<b>Caso de uso 4</b>	<b>Introducir asignaturas</b>
<b>Descripción</b>	Acción que permite al docente añadir una asignatura.
<b>Actor</b>	Docente
<b>Precondiciones</b>	Estar autenticado en el sistema a través de la aplicación con un usuario de tipo docente.
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario seleccionará la opción del menú principal "Asignaturas".</li> <li>2. Visualizará la pantalla de asignaturas a las que imparte docencia.</li> <li>3. Seleccionará la opción de introducir una nueva asignatura.</li> <li>4. Introducirá una descripción para la asignatura y un color asociado.</li> </ol>
<b>Postcondiciones</b>	La asignatura se ha introducido en el sistema.

<b>Caso de uso 5</b>	<b>Suscribir a asignaturas</b>
<b>Descripción</b>	Acción que permite al alumno suscribirse/matricularse a las asignaturas.
<b>Actor</b>	Alumno
<b>Precondiciones</b>	Estar autenticado en el sistema a través de la aplicación. Las asignaturas deben estar añadidas en el sistema.
<b>Flujo</b>	<ol style="list-style-type: none"> <li>5. El usuario seleccionará la opción del menú principal "Asignaturas".</li> <li>6. Selecciona la asignatura a la que desea suscribirse.</li> <li>7. Muestra un listado con todas las asignaturas a las que se encuentra suscrito.</li> </ol>
<b>Postcondiciones</b>	El usuario se encuentra suscrito a una asignatura.

<b>Caso de uso 6</b>	<b>Añadir horario semanal</b>
<b>Descripción</b>	Acción que permite al usuario añadir su horario semanal.
<b>Actor</b>	Alumno/Docente
<b>Precondiciones</b>	Estar autenticado en el sistema a través de la aplicación. Las asignaturas deben estar añadidas
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario seleccionará la opción del menú principal "Horario".</li> <li>2. Para cada día seleccionará la asignatura y detallará su horario.</li> </ol>
<b>Postcondiciones</b>	El usuario visualiza el horario semanal.

<b>Caso de uso 7</b>	<b>Consultar la agenda</b>
<b>Descripción</b>	Acción que permite al usuario consultar su agenda
<b>Actor</b>	Alumno/Docente
<b>Precondiciones</b>	Estar autenticado en el sistema a través de la aplicación.
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario seleccionará la opción del menú principal "EduAgend@".</li> <li>2. Se mostrará la agenda con vista mensual.</li> </ol>
<b>Postcondiciones</b>	El usuario visualiza la agenda

<b>Caso de uso 8</b>	<b>Crear un Evento de Tipo Personal</b>
<b>Descripción</b>	Acción que permite al usuario añadir un evento personal en su agenda.
<b>Actor</b>	Alumno/Docente
<b>Precondiciones</b>	Estar autenticado en el sistema a través de la aplicación.
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la pantalla de consultar agenda.</li> <li>2. El usuario seleccionará la opción crear evento.</li> <li>3. Introducirá un título, una fecha de inicio y una fecha de fin .</li> <li>4. Marcará el evento de tipo personal.</li> </ol>
<b>Postcondiciones</b>	El evento se ha añadido a la agenda.

<b>Caso de uso 9</b>	<b>Crear un evento de Tipo Tarea</b>
<b>Descripción</b>	Acción que permite al docente agregar una tarea para una asignatura.
<b>Actor</b>	Docente
<b>Precondiciones</b>	Estar autenticado en el sistema a través de la aplicación. Las asignaturas deben estar añadidas
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la pantalla de consultar agenda.</li> <li>2. El usuario seleccionará la opción crear evento.</li> <li>3. Introducirá un título, una fecha de inicio y una fecha de fin .</li> <li>4. El docente seleccionará la asignatura para la que desea crear la tarea.</li> <li>5. Marcará el tipo de evento como Tarea.</li> <li>6. Los datos se almacenan en el servidor Firebase.</li> <li>7. El servidor envía la tarea a todas las agendas de los alumnos suscritos a la asignatura.</li> </ol>
<b>Postcondiciones</b>	El usuario docente visualiza el evento de tipo tarea agregado. El usuario alumno recibe el evento de tipo tarea en su agenda.

<b>Caso de uso 10</b>	<b>Crear un evento de Tipo Examen</b>
<b>Descripción</b>	Acción que permite al docente agregar un examen para una asignatura.
<b>Actor</b>	Docente
<b>Precondiciones</b>	Estar autenticado en el sistema a través de la aplicación. Las asignaturas deben estar añadidas
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la pantalla de consultar agenda.</li> <li>2. El usuario seleccionará la opción crear evento.</li> <li>3. Introducirá un título, una fecha de inicio y una fecha de fin .</li> <li>4. El docente seleccionará la asignatura para la que desea agregar el examen.</li> <li>5. Marcará el tipo de evento como Examen.</li> <li>6. Los datos se almacenan en el servidor Firebase.</li> <li>7. El servidor envía el evento de tipo examen a todas las agendas de los alumnos suscritos a la asignatura.</li> </ol>
<b>Postcondiciones</b>	El usuario docente visualiza el evento de tipo examen agregado. El usuario alumno recibe el evento de tipo examen en su agenda.

<b>Caso de uso 11</b>	<b>Introducir Notas</b>
<b>Descripción</b>	Acción que permite al usuario docente añadir notas de tareas o exámenes de sus asignaturas.
<b>Actor</b>	Docente
<b>Precondiciones</b>	Estar autenticado en el sistema a través de la aplicación con un usuario de tipo Docente. Las asignaturas que imparte docencia deben estar añadidas.
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la pantalla de introducir notas</li> <li>2. El usuario seleccionará la asignatura a la que desea introducir una nota.</li> <li>3. Se indicará la tarea o examen que desea evaluar o consultar.</li> <li>4. Se mostrarán las calificaciones de esa tarea para todos los alumnos suscritos a la asignatura.</li> <li>5. El docente introducirá las notas correspondientes.</li> </ol>
<b>Postcondiciones</b>	Las notas para una tarea o examen de cierta asignatura han sido añadidas.

<b>Caso de uso 12</b>	<b>Visualizar Notas</b>
<b>Descripción</b>	Acción que permite al usuario alumno consultar las notas obtenidas a las tareas o exámenes correspondientes a una asignatura.
<b>Actor</b>	Alumno
<b>Precondiciones</b>	Estar autenticado en el sistema a través de la aplicación con un usuario de tipo Alumno. El alumno debe estar matriculado como mínimo en una asignatura.
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la pantalla de consultar notas</li> <li>2. El usuario seleccionará la asignatura para la que desea consultar las notas.</li> <li>3. Se mostrarán las calificaciones de esa tarea para todos los alumnos suscritos a la asignatura.</li> </ol>
<b>Postcondiciones</b>	El usuario visualiza las notas obtenidas en una asignatura.

<b>Caso de uso 13</b>	<b>Enviar un Mensaje</b>
<b>Descripción</b>	Acción que permite al usuario enviar un mensaje
<b>Actor</b>	Alumno/Docente
<b>Precondiciones</b>	Estar autenticado en el sistema a través de la aplicación.
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la pantalla de enviar mensaje</li> <li>2. El usuario seleccionará la asignatura en la que se encuentra el usuario al que desea enviar un mensaje.</li> <li>3. Se mostrarán todos los usuarios suscritos a esa asignatura, así como el docente responsable.</li> <li>4. Se seleccionará un usuario y se introducirá un mensaje.</li> </ol>
<b>Postcondiciones</b>	El mensaje se ha enviado al usuario seleccionado

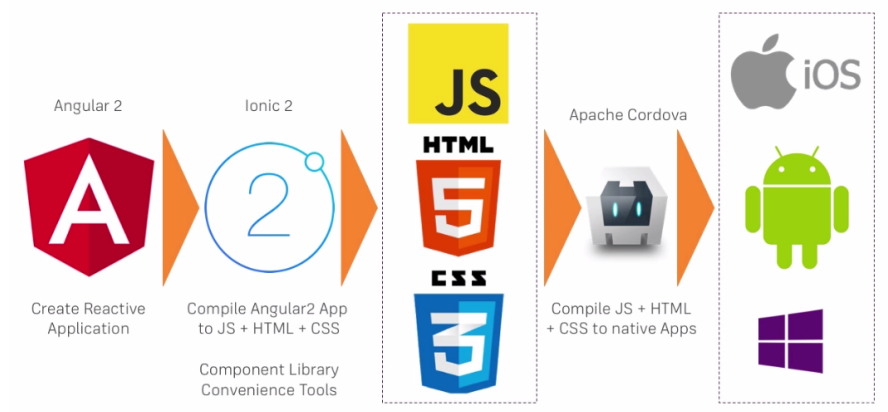
Caso de uso 14	Leer Mensajes
<b>Descripción</b>	Acción que permite al usuario consultar los mensajes recibidos
<b>Actor</b>	Alumno/Docente
<b>Precondiciones</b>	Estar autenticado en el sistema a través de la aplicación.
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la pantalla de leer mensajes.</li> <li>2. Selecciona el mensaje que desea leer.</li> </ol>
<b>Postcondiciones</b>	El mensaje se marca como leído.



## 3.2 Diseño de la Arquitectura

Para el desarrollo de la aplicación se va a utilizar el framework de desarrollo para aplicaciones móviles híbridas **IONIC 2**. La mayor ventaja de este framework es que permite construir aplicaciones tanto para IOS, Android o Windows Phone con el mismo código de desarrollo.

Ionic 2 está basado en el framework **Angular 2** y **Apache Cordova**. A continuación se muestra la siguiente figura que muestra el funcionamiento de IONIC2.



*Ilustración 16: Esquema de IONIC*

Nuestra aplicación, al estar basado en Angular 2, utiliza el patrón de diseño conocido como **Modelo Vista Controlador** (MVC), que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

### 3.2.1 Vista

Muestra la información a través de una interfaz de usuario. En el caso de Angular 2 equivaldría a los **templates**, es decir, los archivos generados con extensión .html.

### 3.2.2. Controlador

Actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno. En Angular 2 son los **componentes** los que realizan la función de controlador.

### 3.2.3 Modelo

Contiene una representación de los datos que maneja el sistema, su lógica de negocio y sus mecanismos de persistencia. El modelo es representado por los **servicios y factorías**.

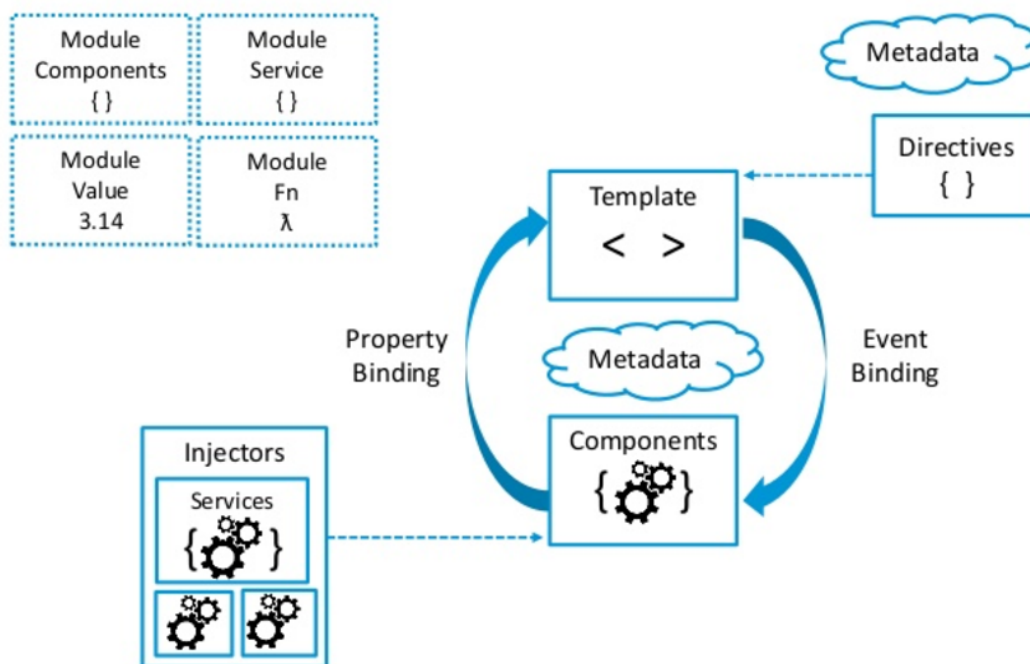


Ilustración 17: Arquitectura de una aplicación desarrollada con Angular 2

### 3.2.4 Persistencia y sincronización de los datos

Para almacenar y sincronizar datos en la nube en tiempo real se ha utilizado una base de datos NoSQL **Firestore Realtime Database**. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. De tal forma que, cada cliente comparte una instancia de Realtime Database y reciben actualizaciones de forma automática con los datos más nuevos.

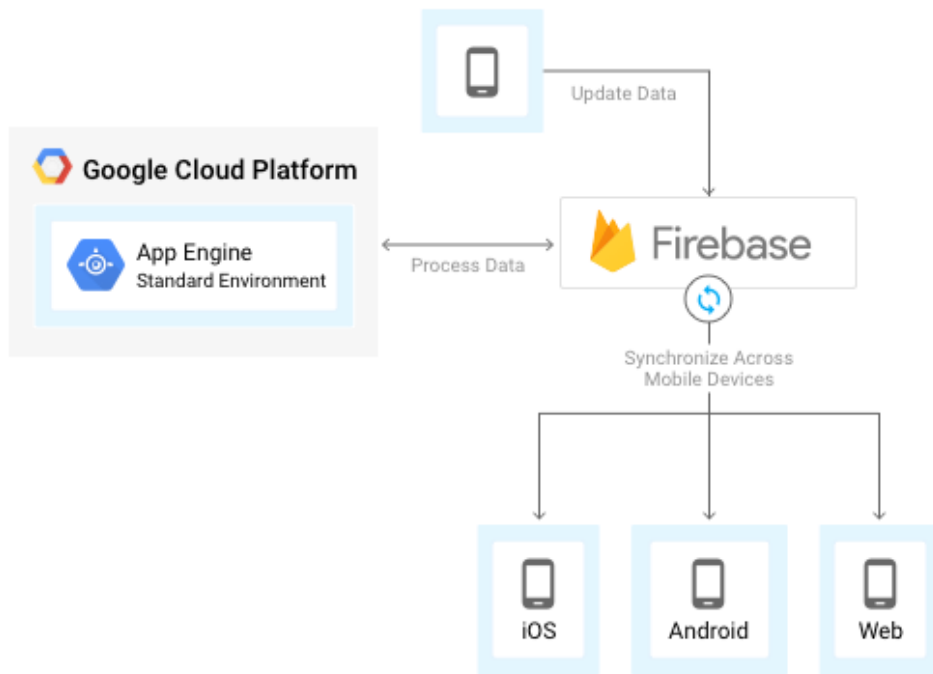


Ilustración 18: Logotipo de Firebase

Se ha escogido esta opción, debido a que Firebase resuelve de forma transparente la **sincronización de datos** con todos los dispositivos conectados, requerimiento indispensable en nuestra aplicación.

Además, aunque el dispositivo no tenga conexión el SDK Firebase Realtime Database hace que tus datos persistan en el disco. Una vez que se recupera la

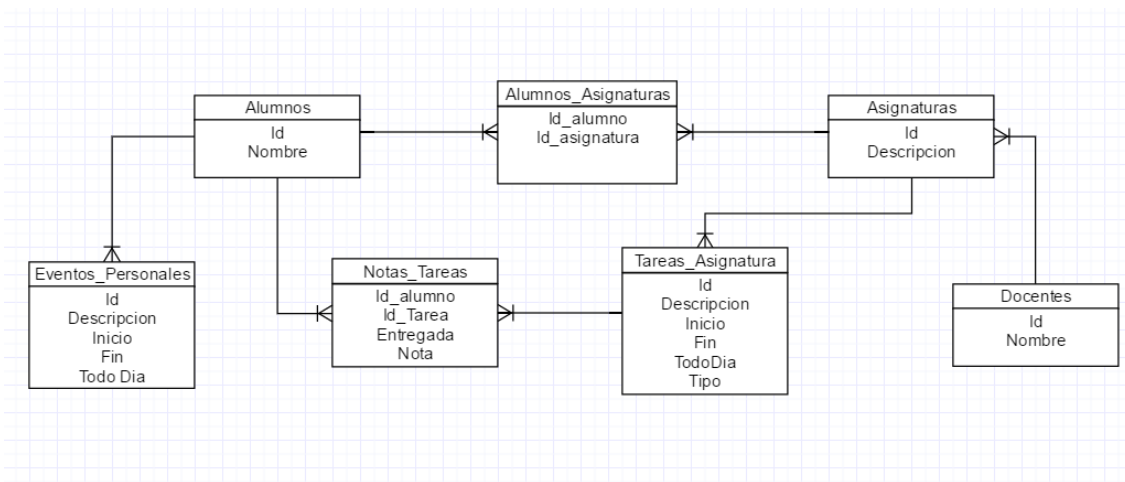
conectividad, el dispositivo del cliente recibe cualquier modificación que se haya perdido , sincronizándola con el estado actual del servidor.



*Ilustración 19: Firebase - Sincronización de un dispositivo con el resto*

### 3.3 Modelo de datos

A continuación se muestra un esquema relacional de la base de datos de nuestra aplicación.



*Ilustración 20: Esquema relacional de los datos*

Como ya se ha comentado, se utilizará una base de datos NoSQL, por ello transformaremos el esquema anterior en un modelo de datos orientado a documentos de tipo JSON.

En este tipo de base de datos existen varios patrones de modelado básico que nos ayudarán a establecer la estructura que tendrán los documentos para lograr la relación de datos reflejada en el esquema relacional anterior:

- **Modelo de datos embebidos:** Se almacenan en un mismo documentos dos colecciones de datos. Con el sistema de datos embebidos la aplicación puede obtener toda la información en una única consulta.
- **Modelo de datos con referencias.** Este patrón busca imitar el comportamiento de las claves ajenas . Insertaremos en la colección de datos una referencia a otra colección .

A partir de los patrones de modelado comentados anteriormente, nuestro esquema de la base de datos NoSQL quedaría de la siguiente forma:

```
Alumno = {  
  Id: string;  
  nombre: string;  
  suscripciones: [ id_asignatura:string ]  
  tareas: [ id_tarea:string ]  
  eventos : [{  
    id: string;  
    descripcion: string;  
    inicio: string;  
    fin: string;  
    todoDia: boolean;  
  }]  
}]
```

```
Asignatura= {  
  Id: string;  
  descripcion: string;  
  id_docente:string;  
  matriculados: [ id_alumno:string ]  
  
  tareas: [{  
    id:string;  
    descripcion: string;  
    inicio: string;  
    fin: string;  
    todoDia: boolean;  
    tipo: string;  
    notas: [ {  
      id_alumno:string;  
      nota:float;  
    }  
  ]  
}  
}]
```

```
Docente = {  
  Id:string;  
  nombre:string  
}
```

## 4 Implementación

### 4.1 Hardware utilizado

Para el desarrollo de la aplicación se ha utilizado un equipo con las siguientes características técnicas:

- Portátil Ultrabook Toshiba
- Procesador : Core i5
- Memoria RAM: 8GB.

Durante la fase de testeo de la aplicación se ha utilizado un iPhone 6 Plus.

### 4.2 Software empleado

Se ha requerido la instalación de la versión más reciente de NodeJS, Ionic CLI y Cordova

Además se ha instalado el módulo ionic-calendar para poder utilizarlo en nuestra aplicación.

También se ha instalado el módulo moments para el correcto tratamiento de fechas.

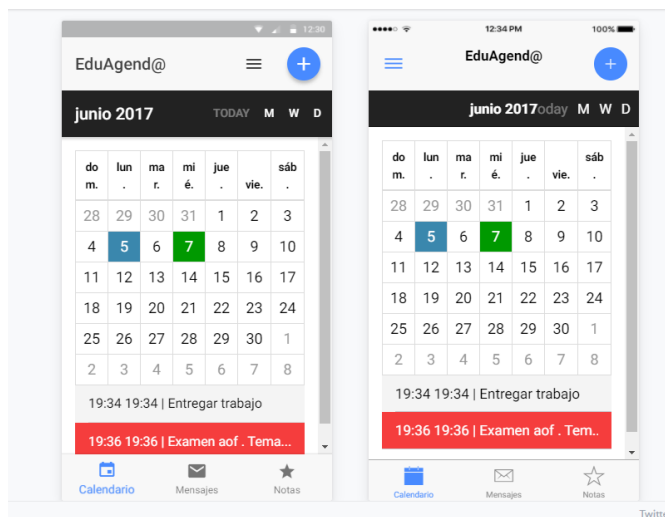
Todos los iconos que aparecen en la aplicación se han extraído del paquete de IONICONS que proporciona IONIC.

### 4.3 Pruebas

Para la comprobación del correcto funcionamiento de la aplicación, se han utilizado las herramientas proporcionadas por **Ionic**.

Por un lado, nos permite testear desde el navegador el correcto funcionamiento de a aplicación. Para ello sólo debemos ejecutar el comando **ionic serve**. Si ejecutamos el comando seguido de la opción **l ionic serve -l** nos permite testearlo para diferentes plataformas.

A continuación, se muestra una imagen de la aplicación en el navegador mostrando para la plataforma Android e IOS.



*Ilustración 21: Aplicación EduAgenda - Testeo en el navegador*

También, se ha procedido a verificar la aplicación en el dispositivo utilizando la herramienta **Ionic View**.

En primer lugar, se ha subido la aplicación a nuestra cuenta en IONIC : **ionic upload**.

Se ha procedido a la instalación de la aplicación Ionic View en el smartphone y tras logear en la cuenta IONIC hemos ejecutado la aplicación.

La aplicación ha sido testeada en un Iphone 6 plus.

## 5. Conclusiones

Al término de este Trabajo Fin de Master, puedo concluir que el resultado ha sido satisfactorio a nivel personal porque me ha aportado muchos conocimientos en el área de desarrollo de aplicaciones híbridas para dispositivos móviles.

La planificación no se ha ajustado a la realidad en todas las fases, y la fase de implementación ha sufrido un importante retraso debido a la falta de experiencia en la tecnología usada .

Los requisitos iniciales básicos no se han cumplido en su totalidad, pero sí han quedado completamente desarrollados los requisitos prioritarios para el correcto funcionamiento de la aplicación

Como posibles ampliaciones a este trabajo se proponen las siguientes mejoras :

- Desarrollar el sistema de mensajes entre los usuarios matriculados en una asignatura.
- Gamificar todo el proceso de finalización de tareas.
- Sincronizar los datos con la plataforma Moodle.

## 6. Glosario

- **Android:** Sistema operativo basado en el núcleo Linux.
- **Angular:** Framework para aplicaciones web de código abierto y mantenido por Google.
- **Ionic:** Herramientas de software destinadas a la creación de aplicaciones híbridas para móviles.
- **Firebase:** Plataforma para desarrollo de aplicaciones web y móviles.
- **Framework:** Término inglés utilizado para referirse a un conjunto de herramientas de software, destinadas a un uso genérico, empaquetadas y distribuidas para poder ser adaptadas en un desarrollo.
- **NodeJS:** Entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor.



## 7. Bibliografía y referencias

- **Pencil Project.** Herramienta open-source para la realización del prototipado.  
<http://pencil.evolus.vn/>
- **yUML.** Herramienta online para la creación de diagramas UML.  
<https://yuml.me/>
- **Libreoffice Writer.** Software para la creación de documentación.  
<https://es.libreoffice.org/>
- **Gliffy.** Herramienta on-line para la creación de diagramas UML.  
<https://www.gliffy.com/>
- **Libreoffice Draw.** Software para la creación de diagramas.  
<https://es.libreoffice.org/>
- **Ionic Documentation.** <https://ionicframework.com/docs/>
- **Angular Documentation.** <https://angular.io/docs/ts/latest/>
- **Firestore Guides.** Estructura tu base de datos  
<https://firebase.google.com/docs/database/web/structure-data>
- **Firestore Blog.** Common SQL Queries Converted for Firestore.  
<https://firebase.googleblog.com/2013/10/queries-part-1-common-sql-queries.html>
- **Ionic2 - Calendar.**  
<https://github.com/twinssbc/Ionic2-Calendar/blob/master/README.md>
- **Moment.** Parse, validate, manipulate and display dates and times in JavaScript.  
<http://momentjs.com/>
- **Stackoverflow.** Resolución de errores.  
<https://stackoverflow.com/>

## 8. Anexos

### Anexo 1 - Manual de Instalación

#### 1. Instalación de IONIC.

En primer lugar instalaremos la versión más reciente de Nodejs  
[Download the installer](#)

A continuación instalaremos Ionic CLI y Cordova ejecutando los siguientes comandos:

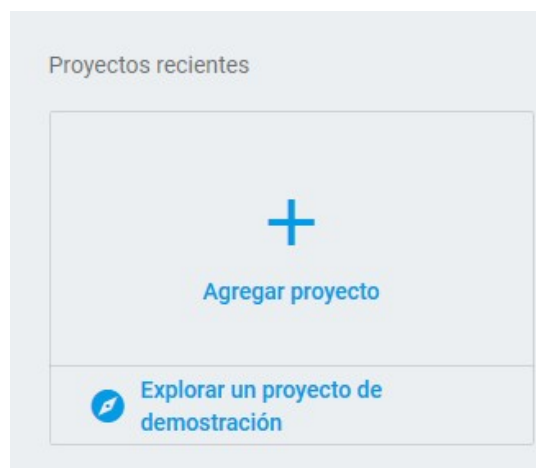
```
npm install -g ionic cordova
```

Copiaremos en una carpeta todo el código fuente suministrado

#### 2. Configuración de Firebase en nuestra aplicación.

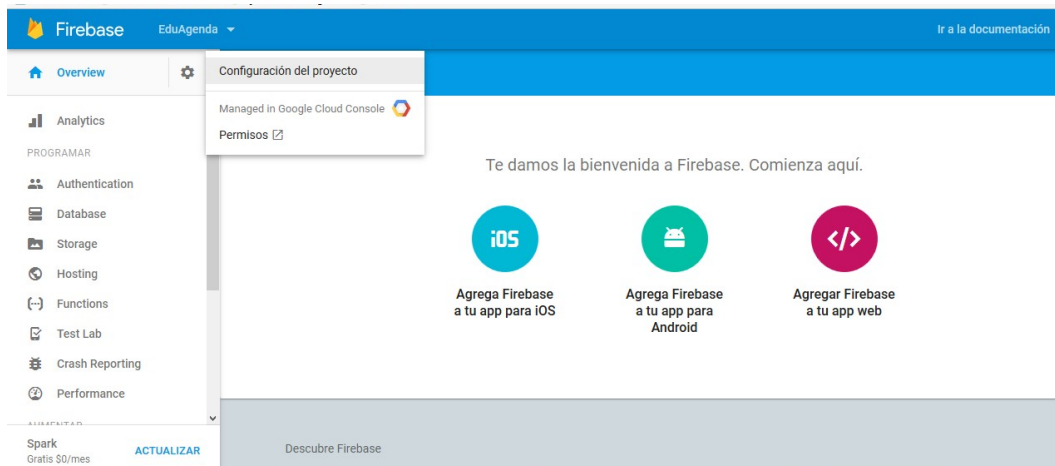
En primer lugar accedemos a la consola de Firebase a través de la siguiente URL: <https://console.firebase.google.com/>

A continuación creamos un proyecto y establecemos un nombre y el país.



*Ilustración 22: Creación de un proyecto en Firebase*

Accedemos al proyecto y hacemos clic en Configuración del proyecto.



*Ilustración 23: Configuración de un proyecto en Firebase*

En la siguiente pantalla accemos clic en Agregar Firebase a tu web y copiamos las siguientes variables:

```
// Initialize Firebase
var config = {
  apiKey: "AIzaSyCl1hIlzYVT9TzHr4csCw28XmGpzGUp27U",
  authDomain: "eduagenda-b60d7.firebaseio.com",
  databaseURL: "https://eduagenda-b60d7.firebaseio.com",
  projectId: "eduagenda-b60d7",
  storageBucket: "eduagenda-b60d7.appspot.com",
  messagingSenderId: "1020840969752"
};
```

A continuación introduciremos estos valores en la aplicación:

- Accede a la carpeta src/app/

En el fichero app.component.ts busca firebase.initializeApp e introduce las variables copiadas anteriormente:

```
firebase.initializeApp({
  apiKey: "AIzaSyCl1hIlzYVT9TzHr4csCw28XmGpzGUp27U",
  authDomain: "eduagenda-b60d7.firebaseio.com",
  databaseURL: "https://eduagenda-b60d7.firebaseio.com"
});
```

### 3. Probar la aplicación

Para probar la aplicación podemos hacerlo en el **navegador** o en un smartphone a través de la aplicación **Ionic View**.

#### 3.1 Probar la aplicación a través del navegador.

Si decidimos probarlo en el navegador debemos ejecutar el comando:  
**ionic serve.**

Una vez cargado nos abrirá una ventana en el navegador **http://localhost:8100** y podremos testear la aplicación.

#### 3.1 Probar la aplicación a través de Ionic View.

Si decidimos utilizar Ionic View debemos crear una cuenta en Ionic:  
**http://ionicframework.com/**

A continuación nos situamos en la carpeta del proyecto y ejecutamos **ionic upload.**

Nos solicitará la cuenta de usuario y password.

Una vez terminado de subir la aplicación, instalaremos la app **Ionic View** en el smartphone.

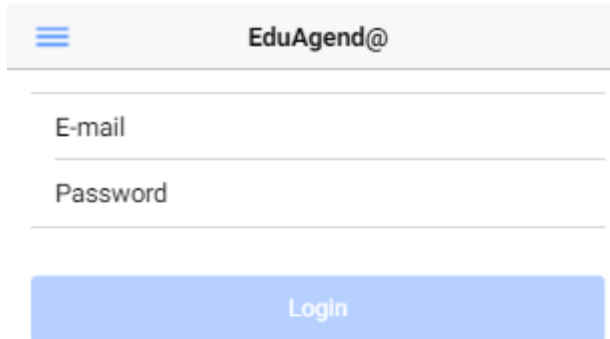
Realizaremos el proceso de Login

Finalmente podremos probar la aplicación.

## Anexo 2 - Manual de Usuario

### Anexo 2. 1. Sección de Usuario no identificado.

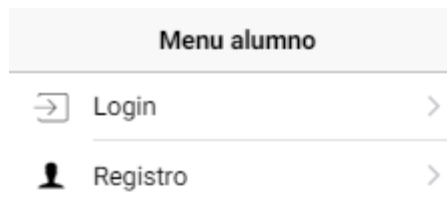
La aplicación comienza mostrando la siguiente pantalla:



The screenshot shows the login interface of the EduAgenda application. At the top, there is a header bar with a hamburger menu icon on the left and the text 'EduAgend@' on the right. Below the header, there are two input fields: the first is labeled 'E-mail' and the second is labeled 'Password'. At the bottom of the form is a blue button with the text 'Login'.

*Ilustración 24: EduAgenda - Pantalla de Login*

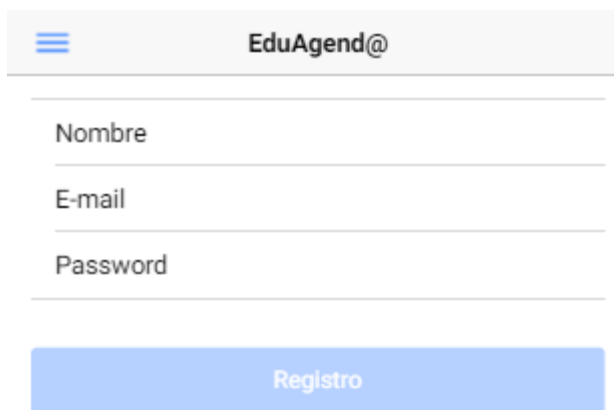
En el caso de que el usuario no esté registrado en la aplicación, puede registrarlo accediendo al **menú** y pulsando sobre **Registro**.



The screenshot shows the 'Menu alumno' screen. At the top, there is a header bar with the text 'Menu alumno'. Below the header, there are two menu items: 'Login' with a right-pointing arrow icon and 'Registro' with a person icon and a right-pointing arrow icon.

*Ilustración 25: EduAgenda - Menu alumno*

En la pantalla de Registro podrá registrarse en la aplicación proporcionando un nombre, email y password.



The image shows a mobile application registration screen. At the top, there is a header bar with a hamburger menu icon on the left and the text 'EduAgend@' on the right. Below the header, there are three input fields stacked vertically, labeled 'Nombre', 'E-mail', and 'Password'. At the bottom of the screen, there is a prominent blue button with the text 'Registro' centered on it.

*Ilustración 26: EduAgenda - Pantalla de Registro*

**NOTA.-** El registro automático le crea una cuenta en Modo Alumno. Si desea operar en Modo Docente, debe enviar un email al Administrador.

## Anexo 2 - Manual de Usuario

### Anexo 2. 1. Sección de Usuario Modo Alumno.

Tras autenticarse en la aplicación se muestra la pantalla principal de la aplicación.

En esta pantalla puede observarse la agenda en vista mensual con todos los eventos marcados en la agenda. Podemos cambiar la vista a semanal o diaria pulsando en las letras W (semanal) y D(diaria).

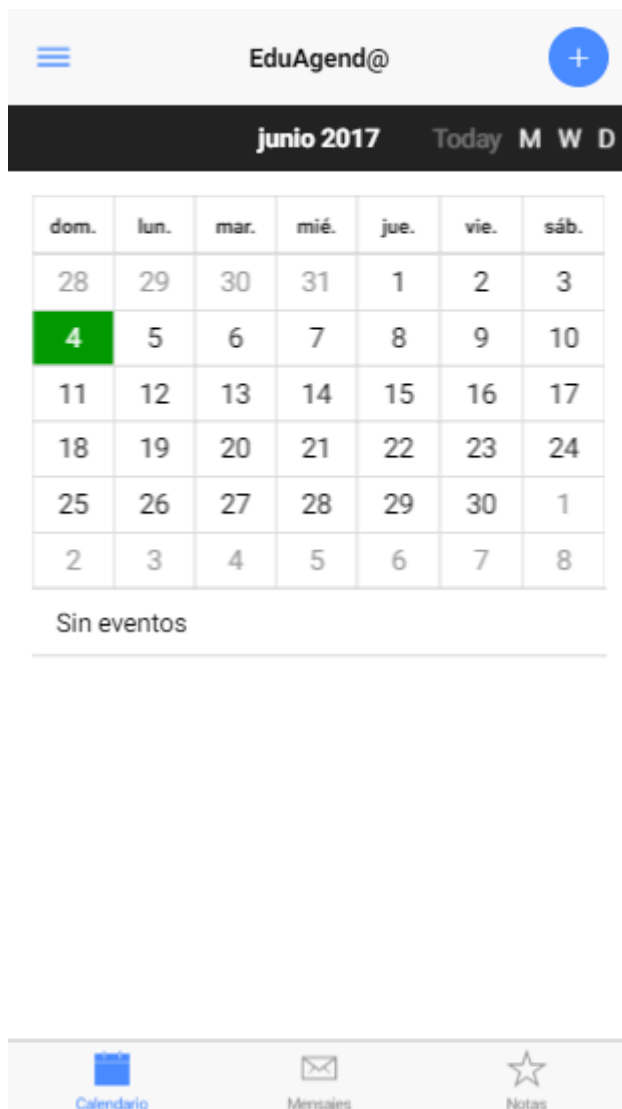


Ilustración 27: Eduagenda. Pantalla del calendario en vista mensual.

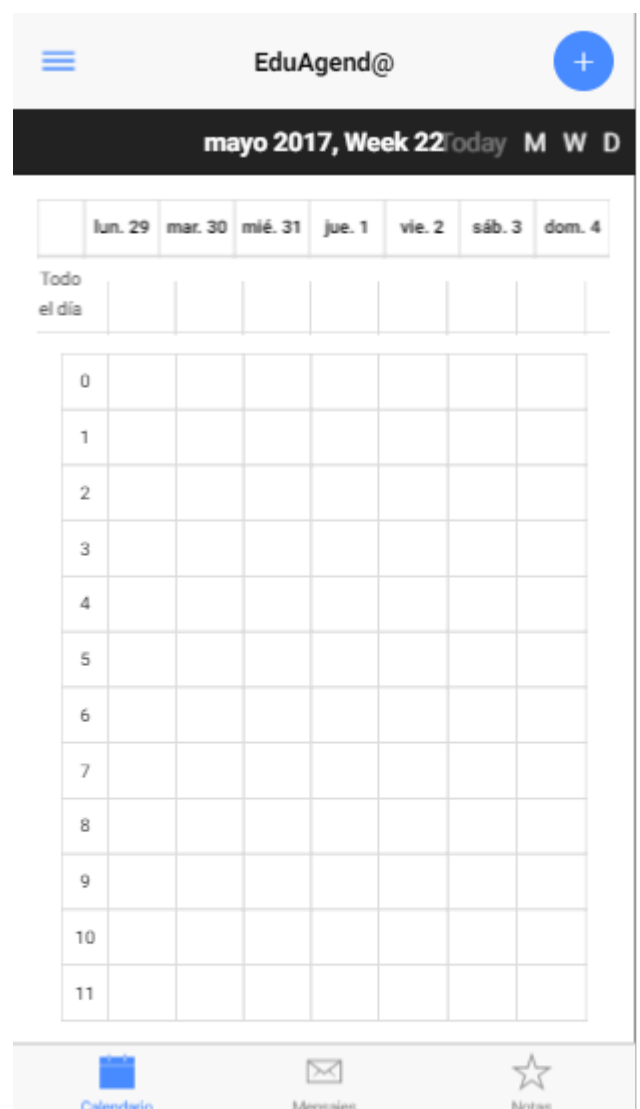
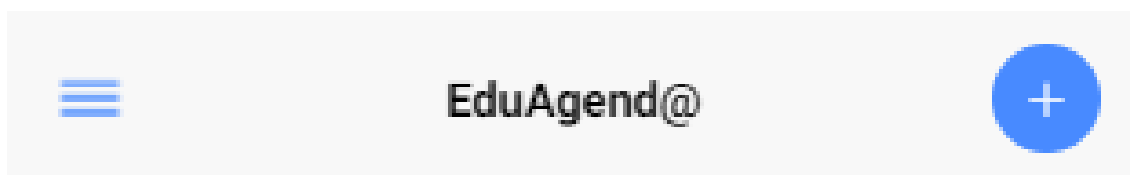
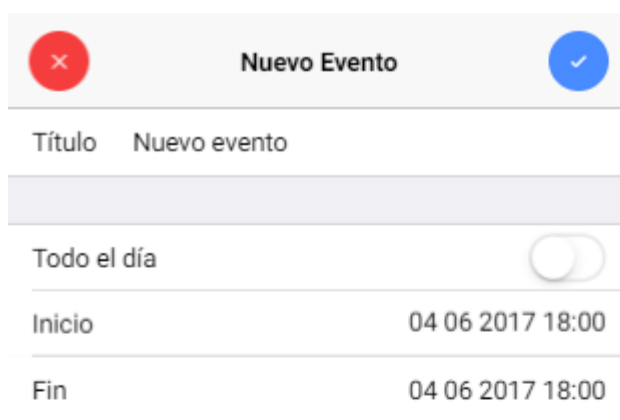


Ilustración 28: Eduagenda. Pantalla del calendario en vista semanal.

A través de esta pantalla podemos crear una nueva tarea pulsando en el botón que aparece en la barra superior con el símbolo +



Al pulsar en el botón accede a la siguiente pantalla que le permite agregar una nueva tarea.

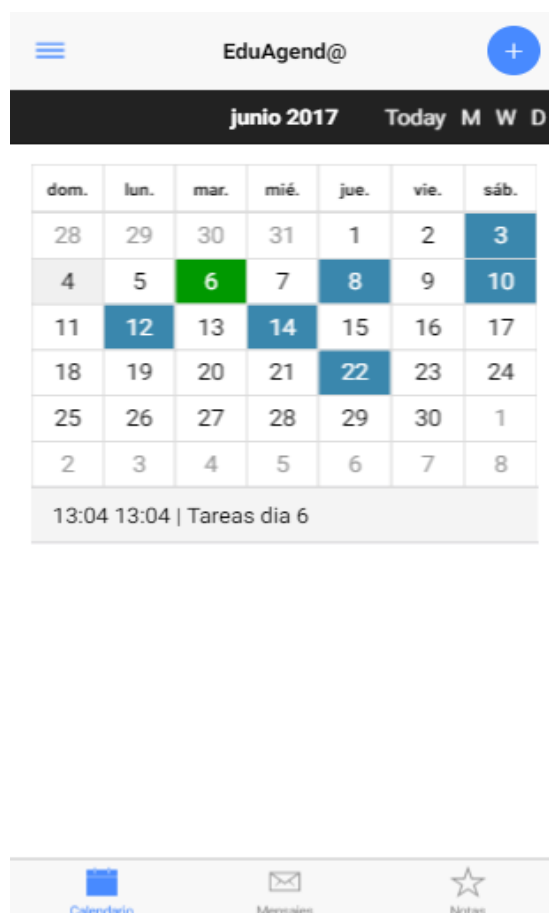


Nuevo Evento	
Título	Nuevo evento
Todo el día <input type="checkbox"/>	
Inicio	04 06 2017 18:00
Fin	04 06 2017 18:00

*Ilustración 29: EduAgenda - Nuevo evento en modo alumno*

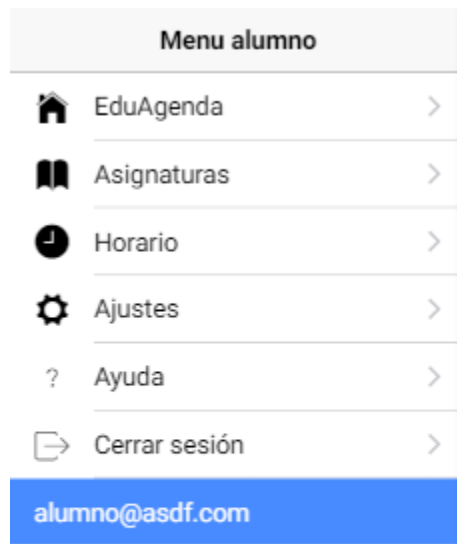


Una vez confirmada la creación del evento, el programa marca en su agenda el evento creado.

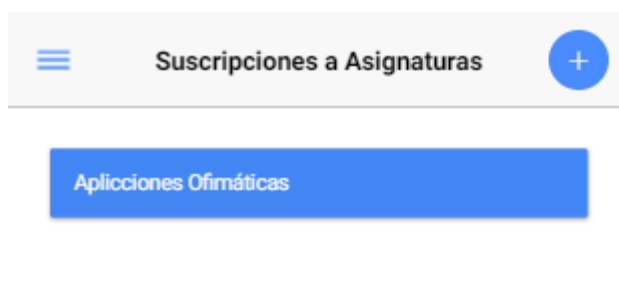


*Ilustración 30: EduAgenda - Vista del calendario*

Si accede al menú podrá verificar las credenciales proporcionadas en la pantalla de login, así como el Modo en el que se encuentra. En este caso puede observarse que el usuario se encuentra en **Modo Alumno**



Para poder recibir todas las fechas de entrega de tareas y exámenes previsto debemos suscribirnos a las asignaturas. Para ello debe acceder a la opción del menú Asignaturas.



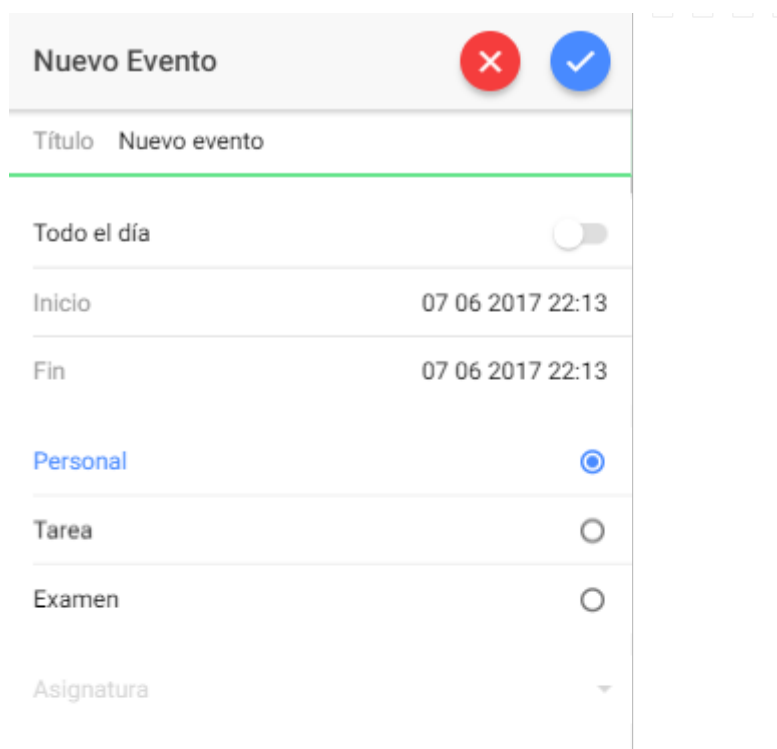
En la imagen anterior puede observarse que el alumno se encuentra matriculado en la asignatura Aplicaciones Ofimáticas. En el caso de querer suscribirse a otra asignatura debe hacer clic en el botón superior con el símbolo de +.

Una vez terminado el proceso de suscripción recibirá en su agenda todas las tareas programadas en la asignatura. Cada vez que el docente añade una tarea, el alumno recibirá de forma automática la tarea y aparecerá reflejada en su calendario.

## Anexo 2. 2. Sección de Usuario Modo Docente.

Básicamente el funcionamiento de la aplicación en **Modo Docente** es muy similar al Modo Alumno exceptuando las siguientes pantallas.

### Creación de Eventos



Nuevo Evento	
Título	Nuevo evento
-----	
Todo el día	<input type="checkbox"/>
Inicio	07 06 2017 22:13
Fin	07 06 2017 22:13
Personal	<input checked="" type="radio"/>
Tarea	<input type="radio"/>
Examen	<input type="radio"/>
Asignatura	▼

*Ilustración 31: EduAgenda - Nuevo Evento en modo docente*

Como podemos observar , la aplicación en Modo Docente ofrece la posibilidad de crear un evento de tipo Personal, Tarea o Examen.

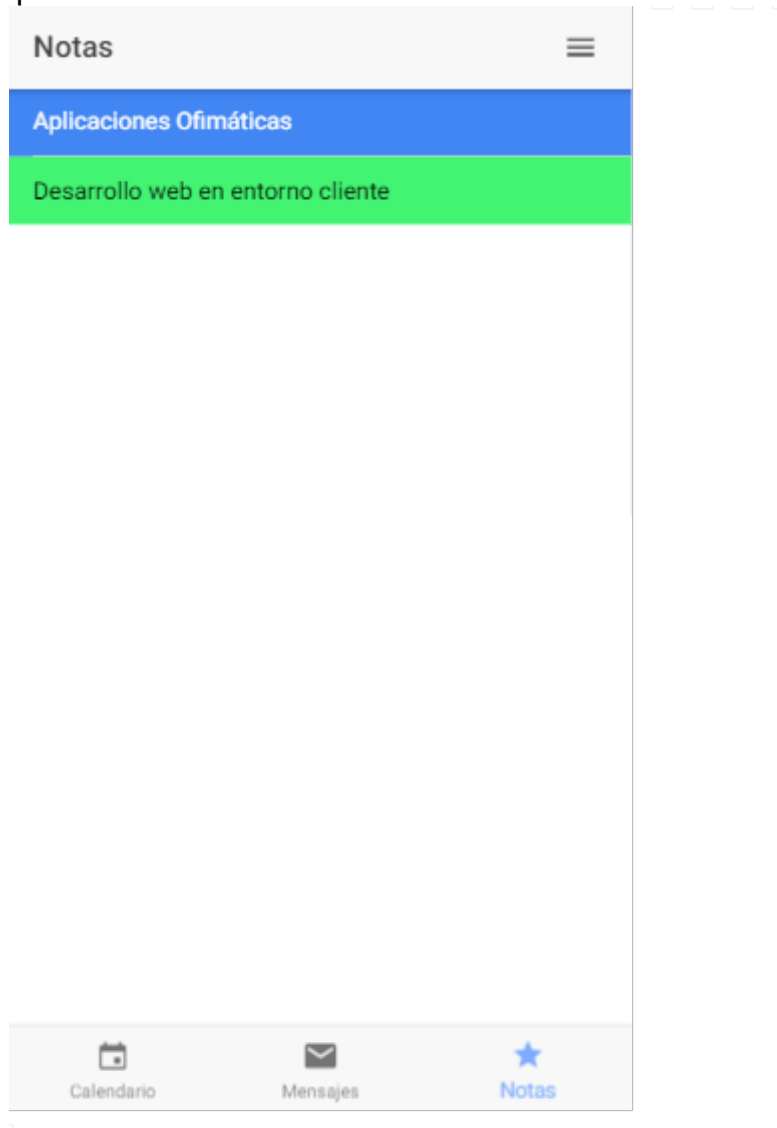
Si el usuario selecciona la opción Tarea o Examen se habilitará el control Asignatura y deberá completarse.

Una vez validada la creación del evento, la agenda de los alumnos matriculados en esa asignatura se actualizará de forma automática.

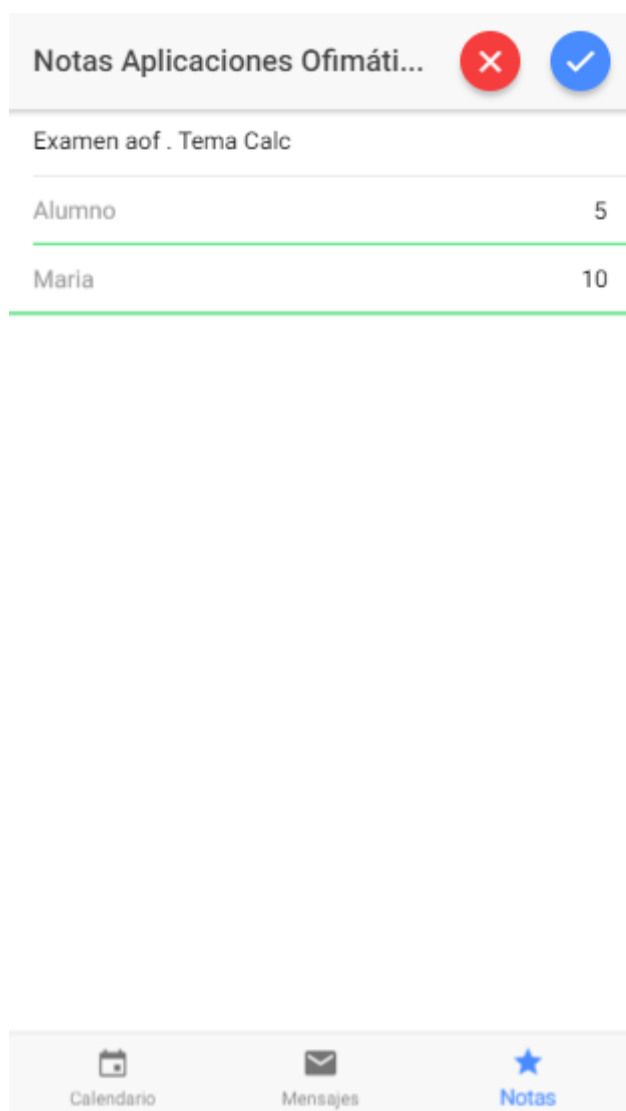
## Introducción de Notas

Otra de las opciones disponibles en **Modo Docente** es la introducción de Notas.

Para ello simplemente realizaremos clic en Notas



A continuación seleccionamos la asignatura, la tarea a corregir y podremos asignar el resultado obtenido a cada uno de nuestros alumnos



The screenshot shows a mobile application interface for managing grades. At the top, there is a header bar with the text "Notas Aplicaciones Ofimáti..." and two circular icons: a red one with a white 'X' and a blue one with a white checkmark. Below the header, the text "Examen aof . Tema Calc" is displayed. A table follows, with a light gray background and a thin green border. The table has two columns: "Alumno" and a numerical value. The first row shows "Alumno" and "5". The second row shows "Maria" and "10". At the bottom of the screen, there is a navigation bar with three icons: a calendar icon labeled "Calendario", an envelope icon labeled "Mensajes", and a blue star icon labeled "Notas".

Alumno	
Alumno	5
Maria	10