1. Definición del proyecto

# 1.1 Ámbito.

Para el desarrollo de las clases de Ciclos Formativos de Grado Superior como el de Desarrollo de Aplicaciones Multiplataforma es necesario prescindir de plataformas como Edmodo, que permiten interacción entre los profesores y los alumnos al poder poner mensajes, asignaciones y disponer de una manera sencilla de llevar las clases día a día. Sin embargo, esta plataforma es de uso multidisciplinar, por lo que no está adaptada a un Ciclo Formativo como el nuestro.

Por ello, gracias a la App desarrollada, podremos disponer de una plataforma especializada que nos permita la interacción entre profesores y alumnos de una manera más específica, modelada en torno a los requisitos de profesores y alumnos del ciclo. Sin nuestra App no solo dependemos de una herramienta externa que puede no estar adaptada o no funcionar en momentos necesarios para el ritmo del ciclo, si no que además perderíamos ciertas funciones adicionales que la diferencian de otras plataformas actuales, como son el vincular inicios de temario con los exámenes correspondientes a estos, requerir la subida de proyectos solo en un determinado lenguaje de programación con requisitos específicos como la necesidad de añadir documentación específica del proyecto o test de unidad de JUnit, por ejemplo.

Nuestra App también dispondrá de un autoevaluador, con el que una asignación podrá ser evaluada por el propio alumno en función de unos criterios determinados por el profesor, permitiendo almacenar información adicional sobre cada práctica entregada por el alumno de una manera sencilla.

Además, la App dispondrá de un evaluador de código, gracias al cual la corrección de las prácticas será mucho más ligero para el profesor, y para el alumno supondrá una ayuda a la hora de evitar entregar prácticas con fallos básicos.

Finalmente, se incorporará un apartado de Serious Game, con el objetivo de ayudar a los alumnos de cada asignatura con el aprendizaje del temario que se vaya dando en clase con juegos didácticos sobre los apartados explicados en cada tema de cada asignatura.

Es por esto que nuestra App no sólo reúne las funcionalidades básicas de una plataforma de comunicación entre profesorado y alumnado, si no que está adaptada a las necesidades específicas de nuestro Ciclo Formativo, incorporando funciones específicas como las restricciones de entrega, autoevaluación, corrector de código y serious game, convirtiendo en nuestra App en una herramienta complementaria a toda la etapa educativa a lo largo del curso, y ayudando en su labor tanto a profesores como alumnos.

# 1.2 Alcance del proyecto.

El proyecto se dividirá en 3 partes:

1. Organización temporal de la asignaturas

- La parte principal de la aplicación permitirá a un usuario iniciar sesión en la aplicación tanto a alumnos, profesores, como a un administrador.

- Alumnos:

Podrán visualizar en un calendario el mes completo con todas las anotaciones, asignaciones y exámenes de las asignaturas en las que estén matriculados. Además, podrá visualizar un dia en concreto, entregar las asignaciones de ese día, proceder a la autoevaluación, explicaciones del temario o acceder a los Serious Games en caso de implementarse.

- Profesor:

Dispondrá de un calendario similar al alumno, pero al acceder a un día podrá crear un evento (Iniciar temario, crear asignación, crear exámen, etc) de una asignatura que imparta. Además, podrá añadir alumnos de los distintos cursos en los que imparta clases a sus asignaturas, con el fin de no depender de un administrador para dar de alta o baja alumnos que dejen o se incorporen al módulo.

- Administrador:

Se le permitirá dar de alta, modificar y eliminar la gran mayoría de entidades de la base de datos, como los profesores, alumnos, asignaturas, etc.

2. Autoevaluación y corrector de código

- La siguiente parte del programa será implementar una opción de autoevaluación, de manera de que en las asignaciones que así lo requiera el profesor, el alumno disponga de un panel para autoevaluarse en cada criterio de corrección haya sido establecido previamente.

Así mismo, se implementará un corrector de código, que permita al alumno ver fallos en los proyectos subidos a la aplicación. Esta opción estará inicialmente disponible para el lenguaje Java a un nivel básico, pero se modelará la aplicación para que sea ampliable en concepto de lenguajes diferentes y profundidad en la detección de errores y corrección de código.

3. Serious Game

- Finalmente se incluirán una serie de serious games con el objetivo de ayudar en el aprendizaje del alumno a lo largo de todos los temarios. Serán accesibles a través del inicio de los nuevos temas que se den en el curso.

2. Análisis y requisitos del sistema

2.1 Modelado de datos.

Para comprobar que el diagrama Entidad Relación está correcto, necesitamos determinar si las tablas de éste están en primera, segunda y tercera forma normal.

1FN:

-Se dice que una relación está en 1FN, si y sólo si, satisface la restricción de que todos los atributos toman un único valor cada vez, es decir, no se deben admitir grupos repetitivos. En el caso de que una relación no lo cumpla, para pasarla a 1FN habrá que repetir el resto de atributos de la tupla para cada uno de los valores del grupo repetitivo.

-Todas las tablas lo cumplen.

2FN:

-Se dice que una relación está en 2FN si: - Está en 1FN. - Cada atributo que no forma parte de la clave tiene dependencia funcional completa respecto a la clave, es decir, los atributos suministran información acerca de la clave completa y no sólo de alguna de sus partes.

-Las que tienen solo una clave (usuario, profesor, alumno, asignatura, tema, asignacion) están en 2ªFN.

-Las que no tienen atributos (profesor\_asignatura, tema\_asignacion, alumno\_asignatura) están en 2ªFN.

criterio\_evaluacion(**cod\_asignacion, cod\_criterio,** nombre, porcentaje)

FK cod\_asignacion / asignacion

cod\_asignacion, cod\_criterio => nombre, porque

cod\_asignacion -/-> nombre

cod\_criterio -/-> nombre

cod\_asignacion, cod\_criterio => porcentaje, porque

cod\_asignacion -/-> porcentaje

cod\_criterio -/-> porcentaje

Está en 2FN.

alumno\_criterio\_evaluacion(**cod\_alumno, cod\_asignacion, cod\_criterio,** nota\_autoevaluacion, nota\_evaluacion)

cod\_alumno, cod\_asignacion, cod\_criterio => nota\_autoevaluacion, porque

cod\_alumno -/-> nota\_autoevaluacion

cod\_asignacion -/-> nota\_autoevaluacion

cod\_criterio -/-> nota\_autoevaluacion

cod\_alumno, cod\_asignacion, cod\_criterio => nota\_evaluacion, porque

cod\_alumno -/-> nota\_evaluacion

cod\_asignacion -/-> nota\_valuacion

cod\_criterio -/-> nota\_evaluacion

Está en 2FN.

alumno\_asignacion(**cod\_alumno, cod\_asignacion,** ruta\_archivo, comentario,nota\_autoevaluacion, nota\_evaluacion)

cod\_alumno, cod\_asignacion => ruta\_archivo, porque

cod\_alumno -/-> ruta\_archivo

cod\_asignacion -/-> ruta\_archivo

cod\_alumno, cod\_asignacion => comentario, porque

cod\_alumno -/-> comentario

cod\_asignacion -/-> comentario

cod\_alumno, cod\_asignacion => nota\_autoevaluacion, porque

cod\_alumno -/-> nota\_autoevaluacion

cod\_asignacion -/-> nota\_autoevaluacion

cod\_alumno, cod\_asignacion => nota\_evaluacion, porque

cod\_alumno -/-> nota\_evaluacion

cod\_asignacion -/-> nota\_evaluacion

Está en 2FN.

3FN:

-Se dice que una relación está en 3FN si: - Está en 2FN. - Ningún atributo que no forma parte de la clave depende transitivamente de la clave de la relación, es decir, los atributos facilitan información acerca de la clave y no acerca de otros atributos. Por lo que toda relación que no tenga atributos que no sean clave o que solamente tengan uno, están en 3FN.

-Las que tienen un atributo o ninguno están en 3ªFN (alumno, profesor, asignatura, profesor\_asignatura, tema\_asignacion).

usuario(**nombre\_login**, clave, nombre, apellidos)

nombre\_login -> clave -/-> nombre

nombre\_login -> clave -/-> apellidos

nombre\_login -> nombre -/-> clave

nombre\_login -> nombre -/-> apellidos  
 nombre\_login -> apellidos -/-> nombre

nombre\_login -> apellidos -/-> clave

Está en 3FN.

tema (**cod\_tema**, tema\_padre, nombre, descripcion, fecha\_comienzo, fecha\_fin)

FK tema\_padre / tema

cod\_tema -> tema\_padre -/-> nombre

cod\_tema -> tema\_padre -/-> descripcion

cod\_tema -> tema\_padre -/-> fecha\_comienzo  
 cod\_tema -> tema\_padre -/-> fecha\_fin

cod\_tema -> nombre -> tema\_padre



cod\_tema -> nombre -> descripcion



Como nombre -> cod\_tema, no hace falta comprobar los demás atributos con nombre.

cod\_tema -> descripcion -> tema\_padre



Como descripción -> cod\_tema, no hace falta comprobar los demás atributos con nombre.

cod\_tema -> fecha\_comienzo -/-> tema\_padre

cod\_tema -> fecha\_comienzo -/-> nombre

cod\_tema -> fecha\_comienzo -/-> descripcion

cod\_tema -> fecha\_comienzo -/-> fecha\_fin

cod\_tema -> fecha\_fin -/-> tema\_padre

cod\_tema -> fecha\_fin -/-> nombre

cod\_tema -> fecha\_fin -/-> descripcion

cod\_tema -> fecha\_fin -/-> fecha\_comienzo

Está en 3FN.

criterio\_evaluacion(**cod\_asignacion, cod\_criterio**, nombre, porcentaje)

cod\_asignacion, cod\_criterio -> nombre -/-> porcentaje

cod\_asignacion, cod\_criterio -> porcentaje -/-> nombre

Está en 3FN.

alumno\_criterio\_evaluacion(**cod\_alumno, cod\_asignacion, cod\_criterio**, nota\_autoevaluacion, nota\_evaluacion)

cod\_alumno, cod\_asignacion, cod\_criterio -> nota\_autoevaluacion -/-> nota\_evaluacion

cod\_alumno, cod\_asignacion, cod\_criterio -> nota\_evaluacion -/-> nota\_autoevaluacion

Está en 3FN.

alumno\_asignacion(**cod\_alumno, cod\_asignacion**, nota\_autoevaluacion, nota\_evaluacion)

cod\_alumno, cod\_asignacion -> nota\_autoevaluacion -/-> nota\_evaluacion

cod\_alumno, cod\_asignacion -> nota\_evaluacion -/-> nota\_autoevaluacion

Está en 3FN.