

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 día 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 examen, 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.