



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Trabajo fin de Grado

Grado en Ingeniería Informática-Ingeniería del Software

Realizado por
Alberto Cordón Arévalo
Raúl Navarro Cruz.

Dirigido por:
Jorge Juan Chico

Departamento
Departamento de Tecnología Electrónica

Sevilla, 30 de Junio de 2021 (v.1.0)

Índice general

Resumen	1
I. Introducción	2
I.I Análisis de antecedentes y aportación realizada	2
I.II Comparativa con otras alternativas	3
I.III Definición de objetivos	6
II. Análisis temporal y de costes de desarrollo	7
II.I Planificación	7
II.II Retrasos e incidencias	11
II.III Reuniones	12
II.IV Costes	15
III. Análisis de requisitos, diseño e implementación	18
III.I Requisitos	18
III.II Casos de uso	21
III.III Modelo de datos	50
III.IV Mockups	54
III.V Justificación de las tecnologías y herramientas utilizadas	59
III.VI Gestión de código	64
III.VII Gestión de versiones	66
III.VIII Fuentes e Imágenes	66
III.IX Implementación del cálculo	68
III.X Adaptabilidad del diseño en diferentes resoluciones	70
III.XI Traducción	70
IV. Pruebas	72
IV.I Pruebas de formulario	72
IV.II Pruebas de borrado	73
IV.III Pruebas de listado	73
V. Conclusiones y desarrollos futuros	74
V.I Mejoras	74
Glosario	76
Casos prácticos de competencias	76
Bibliografía	77
Apéndices	78
Manual de usuario	78
Instalación	91

Resumen

El desarrollo de este proyecto surge de la necesidad de una herramienta que lleve a la práctica de forma correcta la evaluación por competencias que está presente en todos los niveles de nuestro sistema educativo.

El objetivo principal del proyecto es el desarrollo de una aplicación web que implemente toda la funcionalidad necesaria para que un profesor pueda gestionar las puntuaciones de sus alumnos dentro del ámbito de la evaluación por competencias, sin restringir la posibilidad del uso de la aplicación para un tipo de evaluación tradicional basada en la ponderación de pruebas.

Esta memoria está compuesta de seis capítulos donde se expondrá el tema a resolver y cómo se ha desarrollado. Además, al final, se encuentran una serie de apartados con el [glosario de términos](#), [la bibliografía](#) y los [apéndices](#). Este último se divide en el manual de usuario, donde se detalla el uso de las principales funcionalidades de la aplicación; y el manual de instalación, donde comentaremos desde dónde y cómo descargar y configurar las herramientas para poner en marcha la aplicación.

En el [capítulo uno](#), introduciremos el proyecto hablando de los objetivos del mismo, analizaremos los antecedentes y compararemos la aplicación con sus alternativas.

En el [capítulo dos](#), trataremos el análisis temporal, que consiste en cómo se ha planificado el desarrollo de la aplicación, así como las desviaciones y las incidencias que han surgido, añadiendo un apartado sobre las reuniones de seguimiento y gestión entre los miembros del grupo o con el tutor. Finalizamos el capítulo con un recuento de los costes, separando los costes previstos de los efectivos.

El [capítulo tres](#) es el más amplio y es donde se encuentra el núcleo del proyecto. Empieza con la especificación de la funcionalidad, pasando de un listado de requisitos al desarrollo de una descripción detallada de las acciones que se realizan en la aplicación para llevar a cabo los procesos derivados de los requisitos. A continuación, veremos una serie de pantallas que describen la interfaz general de la aplicación. Despues justificaremos las herramientas y las tecnologías usadas en el proyecto, así como las reglas y la nomenclatura seguida en el desarrollo. Además, especificaremos los recursos gráficos y tipografías empleadas. Seguiremos con los detalles de la implementación de los cálculos para la obtención de las calificaciones y finalizaremos describiendo cómo se ha implementado la internacionalización con objeto de facilitar la traducción de la aplicación a otros idiomas.

En el [capítulo cuatro](#) se describen las pruebas realizadas para validar la aplicación y sus resultados.

En el [capítulo cinco](#), finalizaremos con las conclusiones derivadas de la realización del proyecto y un resumen de las cosas que hemos aprendido o mejorado durante su realización. Además se listarán una serie de mejoras que creemos que se pueden implementar para mejorar la aplicación y que se proponen como trabajo futuro.

I. Introducción

La **evaluación** de los alumnos, entendida como la valoración del grado de adquisición de conocimientos y competencias por parte de los alumnos, es una tarea esencial del proceso de enseñanza-aprendizaje, y es un problema que ha traído de cabeza a los educadores desde hace siglos.

A menudo, el resultado de la evaluación se representa cuantitativamente mediante una calificación numérica que indica (o debe indicar) el grado de destreza del alumno y si está es suficiente o no para dar la materia por superada.

Si bien el resultado de la evaluación debería proporcionar información sobre qué conocimientos y competencias ha adquirido el estudiante y en qué grado, a menudo los métodos de evaluación que se emplean en la práctica son de tipo sumativo, en los que se obtiene una calificación mediante la suma y promedio de calificaciones de pruebas parciales donde es difícil establecer a posteriori qué competencias concretas no han sido superadas y en qué grado. Este hecho colisiona con la definición en base a competencias que se hace en términos oficiales de los estudios a todos los niveles, desde la educación primaria hasta los grados universitarios, ya que, en la práctica, no es posible o resulta enormemente costoso, evaluar a los alumnos en base a esas mismas competencias que definen los estudios que cursan.

La finalidad de este Proyecto es la creación de una aplicación web (**Verum**) que facilite al educador la evaluación por competencias de los alumnos de la forma más sencilla y eficiente posible, permitiendo la configuración de la misma para su aplicación a todos los niveles educativos que encontramos en nuestro entorno, con énfasis en la educación primaria y secundaria.

Este capítulo se divide en tres apartados principales: la definición de objetivos, el análisis de antecedentes y aportación realizada, y la comparativas con otras alternativas.

I.I Análisis de antecedentes y aportación realizada

Para las enseñanzas medias, el Ministerio de Educación, Cultura y Deporte a través del Boletín Oficial del Estado, enumera siete competencias clave [3], que en el ámbito de este proyecto llamaremos competencias de nivel tres. Además, especifica que las asignaturas constan de bloques de contenidos que, a su vez, derivan en una serie de criterios de evaluación (competencias de nivel dos) y estos están relacionados con estándares de aprendizaje evaluables (competencias de nivel uno) [5]. Es decir cada bloque de contenido está asociado a un grupo de competencias de nivel uno y dos. Por ejemplo, Física y Química de 2º y 3º de la ESO, consta de cinco bloques, el último de ellos, llamado "La energía", tiene seis competencias de nivel dos, y estas, están relacionadas con una a cuatro competencias de nivel uno.

Para las enseñanzas universitarias es distinto. Las competencias generales y específicas (nivel dos) y las competencias básicas (nivel tres) están definidas por las

propias universidades [6]. Además, los criterios del profesor (competencias de nivel uno) son especificadas por los departamentos y profesores de la universidad. Por ejemplo, en nuestro grado, Ingeniería del Software, la Universidad de Sevilla define cinco competencias básicas, catorce generales y treinta y cinco específicas [7].

Aunque las competencias están perfectamente descritas, no se puede saber con claridad si se han superado las competencias ni con qué intensidad, dado que los exámenes están pensados en base al contenido y no a las competencias. La única forma de calcular las calificaciones de las competencias es a través de las notas de los exámenes. Esto es lo contrario a lo que se debería hacer: obtener la calificación total del alumno a través competencias que tiene el alumno sobre la asignatura.

La principal aportación de este trabajo es una herramienta que permite calificar a los alumnos directamente en base a competencias. Hemos permitido que se creen exámenes en torno a las competencias, lo que posibilita una calificación individual y correcta de las mismas. Esto hace que la calificación sobre el conocimiento de un alumno sobre una competencia sea mucho más precisa. Además este sistema no requiere prácticamente el mismo esfuerzo que la evaluación sumativa tradicional.

I.II Comparativa con otras alternativas

El mundo de la evaluación online ha cambiado en este último año. El problema de no poder asistir a las clases ni a los exámenes a causa de la pandemia producida por el COVID-19 [2] ha dejado a muchos educadores sin otra opción que la de enseñar mediante la Web.[4]

Los métodos de evaluación son diversos, cada educador tiene el suyo propio. El uso del papel a la hora de evaluar a un alumno ya es cada vez menos frecuente. Las aplicaciones web realizan y guardan todo ese trabajo, facilitando al profesor un sitio donde poder evaluar de forma segura, fácil y rápida.

Por desgracia, no todas las aplicaciones web se asemejan a los tipos de evaluación de los educadores, ni tan siquiera siguen un patrón homogéneo, e incluso muchas de ellas no tienen en cuenta el sistema de competencias. Esto último puede ser un error muy grave por parte de las aplicaciones, ya que en España las competencias son algo obligatorio y esencial para la evaluación de un alumno. Este es uno de los factores que sí tiene en cuenta Verum.

Para empezar a comparar con nuestra aplicación, hemos realizado una búsqueda de las principales alternativas. Las tres primeras aplicaciones se suelen usar dentro de los siguientes ámbitos educativos: Educación Infantil, Educación Primaria, Educación Secundaria Obligatoria (ESO) y Bachillerato. Estas serían: Google Classroom, Additio y Moodle. La siguiente sería Blackboard, la cual es la plataforma utilizada por la Universidad de Sevilla y que está muy extendida en el ámbito universitario.

De todos estos sistemas, el único que tienen en cuenta de forma específica la evaluación por competencias es Additio.

Un ejemplo del sistema de competencias sería el siguiente. Cada competencia se

clasifica en uno de tres niveles: nivel uno, nivel dos y nivel tres. Cuanto menor sea su nivel, más concreta es. Pondremos como ejemplo la competencia de nivel tres *La actividad científica*, la cual contiene una serie de competencias de nivel dos, un ejemplo de esta sería: *Reconocer e identificar las características del método científico*. CMCT. Esta a su vez contiene una competencia aún más específica de nivel uno, que sería: *Formular hipótesis para explicar fenómenos cotidianos utilizando teorías y modelos científicos*. [3] [5]

Centrándonos en una comparativa entre Additio y Verum, podemos observar que la principal diferencia es como se gestionan las competencias. En primer lugar, al asignar competencias a un ejercicio, Additio no permitirá elegir las competencias de nivel mayor, las de nivel 3, frente a Verum, que permite elegir las de nivel 1, las de menor nivel. Otra diferencia recae en que Additio permite que las competencias de nivel 1 estén relacionadas sin ponderación a las de nivel 3. Sin embargo, Verum permite una relación ponderada de niveles consecutivos, es decir, las de nivel 1 con las de nivel 2, y las de nivel 2 con las de nivel 3. Otra diferencia es el coste de uso, mientras que Verum se va a liberar como un proyecto libre, Additio tiene un coste de aproximadamente 9 euros al año por profesor.

En la siguiente tabla mostraremos de manera visual una comparativa sencilla de nuestra aplicación con las mencionadas anteriormente.

Tabla 1. Comparativas con otras alternativas

	Verum	Additio	Moodle	Google Classroom	Enseñanza Virtual
Se puede usar en todos los niveles educativos.	Sí	Sí	Sí	Sí	No
Permite puntuar competencias dentro de un ejercicio	Sí (Nivel 1)	Sí (Nivel 3)	No	No	No
Permite gestionar evaluaciones	Sí	Sí	No	No	Sí
Permite gestionar todos los niveles de competencias	Sí	No	No	No	No

	Verum	Additio	Moodle	Google Classroom	Enseñanza Virtual
Permite generar informes de las calificaciones de las competencias de un grupo	Sí	Solo las de nivel 3	No	No	No
Permite generar informes de las calificaciones de las evaluaciones de un grupo	Sí	Sí	No	No	Sí
Permite generar informes de las calificaciones de las competencias de un alumno	Sí	Solo las de nivel 3	No	No	No
Permite generar informes de las calificaciones de las evaluaciones de un alumno	Sí	Sí	No	No	No
Disponible licencia gratuita	Sí	No	Sí	Sí	No

En conclusión, todas las aplicaciones tienen sus ventajas y desventajas, algunas más que otras, pero en general Verum es la más completa en lo que se refiere a la capacidad de evaluación por competencias.

I.III Definición de objetivos

Como hemos comentado, la finalidad del proyecto se concreta en una aplicación web que facilite la evaluación por competencias. Los objetivos específicos de la aplicación, que definen, así mismo, el alcance del proyecto, son los siguientes:

- Permitir la evaluación por competencias de forma correcta y sencilla, permitiendo al educador tomar decisiones en función de los resultados obtenidos por los estudiantes como, por ejemplo, programar pruebas que evalúen competencias específicas no superadas por los estudiantes.
- Ser de aplicación a todos los niveles de la enseñanza mediante la configuración de la aplicación por parte de un administrador.
- Permitir y facilitar, así mismo, una calificación tradicional sumativa con objeto de poder adaptar los procedimientos pre-existentes y comparar los resultados con los obtenidos mediante la evaluación por competencias.
- Tener gran flexibilidad a la hora de establecer las ponderaciones que generan las calificaciones, teniendo en cuenta diversos esquemas de evaluación (gestión de recuperaciones, etc.).
- Dotar a la aplicación de una interfaz visual sencilla y adaptable a una diversidad de dispositivos y formatos de pantalla con objeto de facilitar el uso y adopción de la herramienta.
- Tener en cuenta la internacionalización de la aplicación desde el principio para permitir su fácil traducción a otros idiomas y el posterior desarrollo del código en un ambiente internacional.

II. Análisis temporal y de costes de desarrollo

En este capítulo desarrollaremos nuestra gestión del tiempo del proyecto y detallaremos los costes derivados de realizar el Trabajo Fin de Grado. Empezaremos especificando la planificación final de los casos de uso a lo largo de los meses de desarrollo, así como la estrategia que hemos seguido para realizar esta planificación, continuando con las desviaciones del plan y las incidencias que nos han acontecido, terminados por la gestión de costes.

II.I Planificación

La planificación se realizará mediante la metodología *Scrum* [1], constituidos en nuestro caso por cuatro *Sprints*, que no son más que una serie de tiempo prefijado con el fin de crear un incremento utilizable y potencialmente entregable. A lo largo del desarrollo de nuestro proyecto habrá *Sprints* consecutivos con una duración de cuatro semanas, a excepción del último que se extiende hasta las cinco semanas. Todo esto con el fin de empezar el día 1 de Enero y terminar el día 30 de Abril.

Una vez realizado cada *Sprint* se verá reflejado en el proyecto como un incremento en el versionado del mismo, para que así se puedan ver los cambios y avances de este. Para más información del versionado véase [Gestión de versiones](#).

Los *Sprints* son los siguientes:

- Sprint 1: 1 de enero al 28 de enero. Véase [Tabla 1](#).
- Sprint 2: 29 de enero al 25 de febrero. Véase [Tabla 2](#).
- Sprint 3: 26 de febrero al 25 de marzo. Véase [Tabla 3](#).
- Sprint 4: 26 de marzo al 29 de abril. Véase [Tabla 4](#).

Las tablas tienen una cabecera con cada una de las semanas de desarrollo con sus fechas de inicio y de fin. En cada celda se mostrará el nombre de la tarea o el caso de uso y la fecha en la que debe estar finalizada la tarea entre corchetes.

La estrategia respecto a las tareas ha consistido en especificar solo la fecha final, para que no hubiera problema en adelantar tareas. Además, se ha buscado un tiempo para el desarrollo de las tareas amplio. Todo esto nos ha permitido un mayor margen de maniobra ante impedimentos surgidos de otras asignaturas de la carrera. Gracias a esta estrategia el desarrollo ha tenido una presión menor, las desviaciones de la planificación inicial se han mantenido al mínimo y hemos podido resolver las incidencias con un impacto bajo. Véase [Retrasos e incidencias](#).

Tabla 2. Planificación Sprint 1

Semana 1: 1 de Enero al 7 de Enero	Semana 2: 8 de Enero al 14 de Enero	Semana 3: 15 de Enero al 21 de Enero	Semana 4: 22 de Enero al 28 de Enero
Modelo de datos. <i>[3 de Enero]</i>	CU-03 Edición de alumnos. <i>[11 de Enero]</i>	Populate v2. <i>[20 de Enero]</i>	Despliegue de la versión 0.1. <i>[25 de Enero]</i>
Populate v1. <i>[7 de Enero]</i>	CU-04 Eliminación de alumnos. <i>[13 de Enero]</i>	CU-09 Listado de Grupos de clase. <i>[16 de Enero]</i>	Replanificación. <i>[27 de Enero]</i>
CU-01 Listado de alumnos. <i>[4 de Enero]</i>	CU-07 Edición de profesores. <i>[11 de Enero]</i>	CU-10 Creación de Grupos de clase. <i>[18 de Enero]</i>	
CU-02 Creación de alumnos. <i>[5 de Enero]</i>	CU-08 Eliminación de profesores. <i>[13 de Enero]</i>	CU-11 Edición de Grupos de clase. <i>[19 de Enero]</i>	
CU-05 Listado de profesores. <i>[4 de Enero]</i>		CU-12 Eliminación de Grupos de clase. <i>[20 de Enero]</i>	
CU-06 Creación de profesores. <i>[5 de Enero]</i>			

Tabla 3. Planificación Sprint 2

Semana 1: 29 de Enero al 4 de Febrero	Semana 2: 5 de Febrero al 11 de Febrero	Semana 3: 12 de Febrero al 18 de Febrero	Semana 4: 19 de Febrero al 25 de Febrero
CU-13 Asignar alumnos a sus respectivos grupos de clase. <i>[1 de Febrero]</i>	CU-47 Edición de evaluaciones <i>[8 de Febrero]</i>	CU-23 Asignación de competencias. <i>[17 de Febrero]</i>	Despliegue de la versión 0.2. <i>[22 de Febrero]</i>
CU-49 Listado de evaluaciones. <i>[2 de Febrero]</i>	CU-48 Eliminar evaluaciones. <i>[9 de Febrero]</i>	CU-24 Listado de administradores. <i>[13 de Febrero]</i>	Replanificación. <i>[24 de Febrero]</i>

Semana 1: 29 de Enero al 4 de Febrero	Semana 2: 5 de Febrero al 11 de Febrero	Semana 3: 12 de Febrero al 18 de Febrero	Semana 4: 19 de Febrero al 25 de Febrero
CU-46 Creación de evaluaciones. <i>[4 de Febrero]</i>	CU-18 Asignar asignatura a profesores. <i>[11 de Febrero]</i>	CU-25 Creación de administradores. <i>[15 de Febrero]</i>	
CU-14 Listado de asignaturas. <i>[30 de Enero]</i>	CU-19 Listado de competencias. <i>[6 de Febrero]</i>	CU-26 Eliminación de administradores. <i>[16 de Febrero]</i>	
CU-15 Creación de asignaturas. <i>[1 de Febrero]</i>	CU-20 Creación de competencias. <i>[8 de Febrero]</i>	CU-27 Edición de usuarios. <i>[18 de Febrero]</i>	
CU-16 Edición de asignaturas. <i>[3 de Febrero]</i>	CU-21 Edición de competencias. <i>[10 de Febrero]</i>		
CU-17 Eliminación de asignaturas. <i>[4 de Febrero]</i>	CU-22 Eliminación de competencias. <i>[11 de Febrero]</i>		

Tabla 4. Planificación Sprint 3

Semana 1: 26 de Febrero al 4 de Marzo	Semana 2: 5 de Marzo al 11 de Marzo	Semana 3: 12 de Marzo al 18 de Marzo	Semana 4: 19 de Marzo al 25 de Marzo
CU-28 Listar sus grupos de clase. <i>[27 de Febrero]</i>	CU-34 Edición de una actividad. <i>[7 de Marzo]</i>	CU-40 Asignar competencias a un ejercicio con una intensidad. <i>[15 de Marzo]</i>	Despliegue de la versión 0.3. <i>[22 de Marzo]</i>
CU-29 Listar sus asignaturas. <i>[1 de Marzo]</i>	CU-35 Eliminación de una actividad. <i>[8 de Marzo]</i>	CU-36 Ver las calificaciones de un alumno. <i>[16 de Marzo]</i>	Replanificación. <i>[24 de Marzo]</i>

Semana 1: 26 de Febrero al 4 de Marzo	Semana 2: 5 de Marzo al 11 de Marzo	Semana 3: 12 de Marzo al 18 de Marzo	Semana 4: 19 de Marzo al 25 de Marzo
CU-30 Listar las competencias de una asignatura. <i>[3 de Marzo]</i>	CU-37 Creación de ejercicios. <i>[7 de Marzo]</i>	CU-42 Modificar sus datos de usuario. <i>[18 de Marzo]</i>	
CU-31 Listar sus alumnos. <i>[29 de Febrero]</i>	CU-38 Edición de un ejercicio. <i>[9 de Marzo]</i>	Responsive. <i>[18 de Marzo]</i>	
CU-32 Listar actividades. <i>[1 de Marzo]</i>	CU-39 Eliminación de un ejercicio. <i>[10 de Marzo]</i>		
CU-33 Creación de actividades. <i>[3 de Marzo]</i>	Responsive. <i>[11 de Marzo]</i>		

Tabla 5. Planificación Sprint 4

Semana 1: 26 de Marzo al 1 de Abril	Semana 2: 2 de Abril al 8 de Abril	Semana 3: 9 de Abril al 15 de Abril	Semana 4: 16 de Abril al 22 de Abril	Semana 5: 23 de Abril al 29 de Abril
CU-50 Listar bloques (evaluaciones). <i>[29 de Marzo]</i>	CU-41 Puntuar las evaluaciones, las actividades, los ejercicios y las competencias. <i>[7 de Abril]</i>	CU-45 Ver competencias de un alumno. <i>[14 de Abril]</i>	CU-43 Generar informe de un grupo completo. <i>[19 de Abril]</i>	Despliegue de la versión 0.4. <i>[28 de Abril]</i>
CU-51 Crear bloques (evaluaciones). <i>[30 de Marzo]</i>	Responsive. <i>[8 de Abril]</i>	Responsive. <i>[15 de Abril]</i>	CU-44 Generar informe de un alumno. <i>[21 de Abril]</i>	

Semana 1: 26 de Marzo al 1 de Abril	Semana 2: 2 de Abril al 8 de Abril	Semana 3: 9 de Abril al 15 de Abril	Semana 4: 16 de Abril al 22 de Abril	Semana 5: 23 de Abril al 29 de Abril
<p>CU-52 Editar bloques (evaluaciones). [31 de Marzo]</p> <p>CU-53 Eliminar bloques (evaluaciones). [1 de Abril]</p> <p>Responsive. [1 de Abril]</p>			<p>Responsive. [22 de Abril]</p>	

II.II Retrasos e incidencias

Gracias a la estrategia que hemos comentado en el apartado de [Planificación](#) hemos podido mantener los retrasos al mínimo, siendo el desvío grave un total de tres días. También han surgido una serie de incidencias que se han solventado a lo largo del desarrollo. A lo largo de esta sección concretamos cuales son estos retrasos (Véase Tabla 5) e incidencias (Véase Tabla 6).

II.II.I Retrasos

En la **Tabla 5** veremos la lista de los retrasos que ha surgido en el proyecto con sus respectivas fechas de su finalización teórica y la real. Los retrasos más graves a nuestro parecer han sido los acontecidos en los despliegues de las versiones 0.1 y 0.4, que no ha permitido la correspondiente revisión por parte del tutor en la fecha prevista. En resto de retrasos han sido de una gravedad muy baja dado que no ha retrasado otras tareas.

Tabla 6. Tabla de retrasos de desarrollo

Tarea	Fecha prevista	Fecha realizada
CU-02 Creación de alumnos.	5 de Enero	7 de Enero
CU-09 Listado de Grupos de clase.	16 de Enero	17 de Enero
Despliegue de la versión 0.1.	25 de Enero	28 de Enero

Tarea	Fecha prevista	Fecha realizada
Replanificación.	27 de Enero	28 de Enero
Despliegue versión 0.4.	27 de Abril	29 de Abril

II.II.II Incidencias

En la **Tabla 6** observaremos la lista de incidencias que hemos tenido a lo largo del proyecto. Las incidencias más graves han sido la falta del *CRUD* de evaluaciones con anterioridad, lo que nos ha provocado tener que replanificar la dos primera semanas del *Sprint 2*, y la identificación de un par de fallos al asignar y desasignar competencias, que retrasó el despliegue de la versión 0.4.

Tabla 7. Tabla de incidencias

Tarea	Fecha de la incidencia
Añadir CRUD de evaluaciones.	28 de Enero
Reordenación casos de uso.	28 de Enero
Error al mostrar botones de la barra de navegación.	26 de Febrero
Error al copiar los ejercicios.	19 de Marzo
Error al ocultar botones al copiar.	25 de Marzo
Error de selección de nivel para los cálculos de evaluación por competencias.	16 de Abril
Error de selección de nivel para los cálculos de evaluación por competencias.	16 de Abril
Error 404 de fuente.	19 de Abril
Error de actualización de tipo de evaluación.	22 de Abril
Error de cálculo de la nota de un ejercicio.	22 de Abril
Error de intervalo de nota.	22 de Abril
Error al asignar una competencia.	27 de Abril
Error al desasignar una competencia.	27 de Abril

II.III Reuniones

Para mantener a los miembros del grupo informados de la evolución de las tareas de desarrollo y documentación, y recibir la opinión del tutor acerca de estas, las reuniones han jugado un papel muy importante.

Hemos diferenciado varios tipos:

- Una reunión semanal con el tutor, con el fin de recibir *feedback* y ver el avance del proyecto. En las que uno de los compañeros enseñaba el proyecto y el otro cogía apuntes.
- Una reunión fija semanal de dos horas o más entre los miembros del grupo después de la reunión con el tutor. En estas reuniones íbamos asignando y reorganizando las tareas, así como hablando de las incidencias y dudas con respecto a las tareas.
- Alguna reunión esporádica de horas indefinidas.

En la **Tabla 8** están listadas todas las reuniones, ya sean con el profesor o externas, fijas o esporádicas, en la que se resumirá brevemente el contenido de la reunión.

Tabla 8. Tabla de reuniones

Fecha	Tipo de reunión	Resumen
14/9/2020	Con profesor.	Primera reunión del proyecto.
15/9/2020	Interna.	Primera reunión del proyecto.
13/10/2020	Con profesor.	Ideas para especificaciones.
15/10/2020	Interna.	Ideas para especificaciones.
16/10/2020	Interna.	Ideas para especificaciones.
27/10/2020	Con profesor.	Revisión del modelo de datos y de los primeros <i>mockups</i> .
28/10/2020	Interna.	Modificación del modelo de datos.
13/11/2020	Con profesor.	Cambios en el modelo de datos y <i>mockups</i> .
14/11/2020	Interna.	Modificación del modelo de datos.
20/11/2020	Con profesor.	Cambios en las especificaciones, en el modelo de datos y <i>mockups</i> .

Fecha	Tipo de reunión	Resumen
21/11/2020	Interna.	Cambios en las especificaciones, en el modelo de datos y <i>mockups</i> .
04/12/2020	Con profesor.	Revisión de las justificaciones y <i>mockups</i> .
06/12/2020	Interna.	Revisión de las justificaciones y <i>mockups</i> .
11/12/2020	Con profesor.	Revisión de las justificaciones y del modelo de datos.
12/12/2020	Interna.	Cambios de las justificaciones y del modelo de datos.
18/12/2020	Interna.	Cambios de las justificaciones.
08/01/2021	Con profesor.	Revisión del modelo de datos.
11/01/2021	Interna.	Cambios en el modelo de datos.
15/01/2021	Con profesor.	Revisión del desarrollo.
17/01/2021	Interna.	Revisión del desarrollo.
22/01/2021	Con profesor.	Revisión del desarrollo.
23/01/2021	Interna.	Revisión del desarrollo.
05/02/2021	Con profesor.	Revisión del desarrollo.
06/02/2021	Interna.	Revisión del desarrollo.
08/02/2021	Con profesor.	Revisión del desarrollo.
10/02/2021	Interna.	Revisión del desarrollo.
19/02/2021	Con profesor.	Revisión del desarrollo.
20/02/2021	Interna.	Revisión del desarrollo.
26/02/2021	Con profesor.	Revisión del desarrollo.
28/02/2021	Interna.	Revisión del desarrollo.
12/03/2021	Con profesor.	Revisión del desarrollo.
15/03/2021	Interna.	Revisión del desarrollo.

Fecha	Tipo de reunión	Resumen
24/03/2021	Con profesor.	Revisión del desarrollo.
25/03/2021	Interna.	Revisión del desarrollo.
08/04/2021	Con profesor.	Revisión del desarrollo.
09/04/2021	Interna.	Revisión del desarrollo.
15/04/2021	Con profesor.	Revisión del desarrollo.
17/04/2021	Interna.	Revisión del desarrollo.
22/04/2021	Con profesor.	Revisión del desarrollo.
23/04/2021	Interna.	Revisión del desarrollo.
29/04/2021	Con profesor.	Revisión del desarrollo y de la estructura de la memoria.
30/04/2021	Interna.	Revisión del desarrollo y de la estructura de la memoria.
13/05/2021	Con profesor.	Revisión del contenido de la memoria.
15/05/2021	Interna.	Revisión del contenido de la memoria.
10/06/2021	Con profesor.	Revisión del contenido de la memoria.
12/06/2021	Interna.	Revisión del contenido de la memoria.
17/06/2021	Con profesor.	Revisión del contenido de la memoria.
24/06/2021	Con profesor.	Revisión del contenido de la memoria.
28/06/2021	Con profesor.	Revisión del contenido de la memoria.
29/06/2021	Con profesor.	Revisión del contenido de la memoria.

II.IV Costes

En este apartado hablaremos de los gastos de la realización de todo el proyecto, ya sea documentación, reuniones o desarrollo de la aplicación. Empezaremos con la parte más cuantiosa, que son los salarios, después especificaremos apartados menos costosos y terminaremos con los costes totales, que separaremos en teórico y real.

La dedicación al proyecto se ha estimado en 300 horas de trabajo por persona, que coincide con la dedicación prevista para un Trabajo Fin de Grado de 12 créditos ECTS. Por lo tanto, el proyecto supondrá un coste temporal inicial de **600 horas**. Además, los dos miembros del equipo tienen el mismo puesto.

II.IV.I Salarios

Suponemos que cada miembro cobra unos **8,5 euros la hora**, basándonos en salarios de puestos reales. Por lo que 300 horas sumarán un coste de **2550 euros**. También supondremos un **35%** de costes de la Seguridad Social. Estos costes deberían ser calculados mensualmente, así que su coste real, supondría un incremento entre 30 y 35%. Escogemos el más alto. Además, asumiremos que ha habido un sobrecoste por costes por formación, es decir, costes que se han producido al dedicar tiempo en cubrir falta de conocimiento en el desarrollo. Valoramos este sobrecoste del **10%** del **salario total**. Desglosamos este coste en la **Tabla 9**.

Tabla 9. Salarios

Costes por hora (€/hora)	Salario total (€)	Coste de formación (€)	Coste de la Seguridad Social (€)	Coste total por persona (€)	Coste total de la empresa (€)
8,5	2550	510	892,5	3442,5	6885

II.IV.II Amortizaciones

Una amortización es una sucesión de pagos para saldar el coste de bienes físicos para el funcionamiento de la empresa. Lo único que tendríamos que amortizar son los dos ordenadores que suponen un coste total de **2000 euros**. Este tipo de elemento se permite amortizar un **20%** anualmente. Se desglosa en la **Tabla 10**.

Tabla 10. Amortizaciones

Valor total (€)	Coste 2020 (€)	Coste 2021 (€)	Coste total (€)
2000	400	400	800

II.IV.III Licencias

Esta serán las versiones de pago de servicios digitales. La única licencia que hemos utilizado es Photoshop que cuesta **24,19 euros al mes** y que hemos utilizado durante 5 meses. Lo resumiremos en la **Tabla 11**.

Tabla 11. Licencias

Coste (€/mes)	Coste total (€)
24,19	120,95

II.IV.IV Servicios en la nube

Estos son servidores que dan servicio a través de Internet. Para el despliegue se utilizado la versión *Hobby* de Heroku, que cuesta **7 dólares**, que son unos **5,74 euros**. Se ha utilizado durante 5 meses. Lo resumimos en la **Tabla 12**.

Tabla 12. Despliegue

Coste (€/mes)	Coste total (€)
5,74	28,7

II.IV.V Coste teórico total

Reuniendo los datos mencionados en los apartados anteriores podemos resumir los costes en la **Tabla 13**.

Tabla 13. Coste total

Salarios (€)	Amortizacione s (€)	Licencias (€)	Servicios en la nube (€)	Coste total (€)
6885	800	120,95	28,7	7834,65

II.IV.VI Coste real total

Aunque se planificó inicialmente una inversión de 600 horas en el proyecto, se han producido una leve desviación del **4,83%** en la realización del proyecto que aumentará el coste en salarios. Esta desviación la podemos atribuir a los **retrasos e incidencias** que se han producido en el proyecto. Esto supone un aumento de los costes del **4,25%**. Concretamos este sobrecoste en la **Tabla 14**.

Tabla 14. Coste real

Coste en salarios teóricos (€)	Coste en salarios reales (€)	Coste teórico total (€)	Coste real total (€)
6885	7217,78	7834,65	8167,43

III. Análisis de requisitos, diseño e implementación

III.I Requisitos

En la Ingeniería Informática, los requisitos cumplen el papel de entrada en la etapa de diseño. Estos marcan **qué** debe hacer el sistema de una manera real a cómo se implementará en un futuro.

Los tipos de requisitos son variados, y cada uno de estos cumplen una función específica. El conjunto de todos los tipos de requisitos forman una exigencia en forma de documentación sobre el contenido del producto final.

Nosotros hemos dividido los requisitos en tres: requisitos de información, requisitos funcionales y requisitos no funcionales. A continuación, hablaremos de forma más detallada de cada uno de estos.

III.I.I Requisitos de Información:

Este tipo de requisitos describen la información que debe guardar el sistema para poder cumplir sus objetivos. Estos, deben identificar qué es lo relevante para así guardar la información sobre ellos en un futuro.

En nuestro caso hemos realizado nueve requisitos de información, los cuales son los siguientes:

RI-01: Una Competencia debe guardar un código de identificación, un nombre y una descripción y su tipo. Las Competencias se dividen en concretas (nivel 1), específicas (nivel 2) y generales (nivel 3). Además, debe guardar el peso sobre el nivel superior, las puntuaciones de los alumnos y, si es una competencia específica, el peso sobre la asignatura asociada.

RI-02: Una Asignatura debe guardar un nombre, un nivel, el grado, una descripción y las competencias asociadas a ella.

RI-03: Un Profesor debe guardar el año de nacimiento, sus iniciales, un nombre de usuario, una contraseña, un nombre, unos apellidos y un email. Además, debe guardar las asignaturas asociadas.

RI-04: Un Administrador debe guardar el año de nacimiento, sus iniciales, un nombre de usuario, una contraseña, un nombre, unos apellidos y un email.

RI-05: Una Evaluación debe guardar un nombre, una fecha de inicio y de fin, un período, una asignatura, una o más actividades. Las Evaluaciones pueden ser finales o parciales. Las parciales deben guardar el peso sobre la final. Si una evaluación parcial es creada por un profesor se llamará Bloque.

RI-06: Un Grupo debe guardar un nombre, un nivel, un grado, una línea, un profesor,

una asignatura, una evaluación y una lista de alumnos.

RI-07: Una Actividad debe guardar un título, una fecha, un indicador para saber si es una recuperación, una lista de ejercicios, una evaluación asociada y el peso sobre ella, un grupo, y una asignatura.

RI-08: Un Ejercicio debe guardar un peso sobre la actividad, un enunciado, una nota y una lista de competencias. Además, debe guardar el peso de cada competencia sobre el ejercicio y su intensidad.

RI-09: Un Estudiante debe guardar un nombre, unos apellidos, una fecha de nacimiento, sus iniciales y las notas en evaluaciones, actividades, ejercicios y competencias.

III.I.II Requisitos Funcionales:

Este tipo de requisito se basa en la definición de una funcionalidad del sistema o los componentes *software*. Esta, debe tener en cuenta las entradas y las salidas.

Además, este tipo de requisito, de forma general, se define siempre una vez finalizados los casos de uso ya que, en estos, se contemplan lo que debe hacer la aplicación a alto nivel.

De forma resumida, un requisito funcional deberá explicar de forma clara y concisa qué es lo que se **permitirá** hacer en la aplicación.

En nuestro caso, hemos creado veinticuatro de estos, los cuales se han tenido en cuenta los roles:

RF 01: El sistema permitirá a los administradores poder ver el listado de alumnos, así como crear, editar y eliminar dichos alumnos.

RF 02: El sistema permitirá a los administradores poder ver el listado de profesores, así como crear y eliminar dichos profesores.

RF 03: El sistema permitirá a los administradores poder ver el listado de asignaturas, así como crear, editar y eliminar dichas asignaturas.

RF 04: El sistema permitirá a los administradores poder ver el listado de evaluaciones, así como crear, editar y eliminar dichas evaluaciones.

RF 05: El sistema permitirá a los administradores poder ver el listado de grupos de clase, así como crear, editar y eliminar dichos grupos.

RF 06: El sistema permitirá a los administradores poder asignar y desasignar alumnos a un grupo de clase.

RF 06: El sistema permitirá a los administradores poder asignar y desasignar asignaturas a un profesor.

RF 07: El sistema permitirá a los administradores poder ver el listado de competencias,

así como crear, editar y eliminar dichas competencias.

RF 08: El sistema permitirá a los administradores poder asignar y desasignar competencias a una asignatura.

RF 09: El sistema permitirá a los administradores poder ver el listado de administradores, así como crear y eliminar dichos administradores.

RF 10: El sistema permitirá a los administradores poder editar un usuario.

RF 11: El sistema permitirá a los profesores poder listar sus grupos de clase.

RF 12: El sistema permitirá a los profesores poder listar sus asignaturas.

RF 13: El sistema permitirá a los profesores poder listar las competencias de una asignatura.

RF 14: El sistema permitirá a los profesores poder seleccionar el tipo de evaluación para las evaluaciones de un grupo de clase.

RF 15: El sistema permitirá a los profesores poder listar los alumnos de un grupo de clase.

RF 16: El sistema permitirá a los profesores poder ver el listado de actividades, así como crear, editar y eliminar dichas actividades. Además, permitirá copiar actividades que ya han sido creadas.

RF 17: El sistema permitirá a los profesores poder ver las calificaciones de un alumno.

RF 18: El sistema permitirá a los profesores poder crear ejercicios, así como editar y eliminar dichos ejercicios.

RF 19: El sistema permitirá a los profesores poder asignar competencias a un ejercicio con una intensidad. Además, permitirá editar y desasignar las competencias ya asignadas.

RF 20: El sistema permitirá a los profesores poder puntuar evaluaciones, actividades, ejercicios y competencias.

RF 21: El sistema permitirá a los profesores poder generar informes de un grupo completo, así como el de un alumno.

RF 22: El sistema permitirá a los profesores poder ver las competencias de un alumno.

RF 23: El sistema permitirá a los profesores poder ver el listado de bloques, así como crear, editar y eliminar dichos bloques.

RF 24: El sistema permitirá a los usuarios registrados poder modificar sus datos de perfil.

III.I.III Requisitos No Funcionales:

Estos requisitos son normalmente los más escasos, ya que se basan en aspectos de calidad, como la usabilidad, disponibilidad, fiabilidad, rendimiento, etc.

Se podrían definir, como aquellos requisitos que no describen información a guardar. También podría entenderse como que son aquellas restricciones que pone el cliente.

En nuestro caso solo hemos visto oportuno uno de ellos:

RNF 01: El sistema permitirá a los usuarios poder usar la aplicación en español y en inglés.

III.II Casos de uso

En Ingeniería del Software, un caso se basa en la obtención de requisitos fundamentales de un nuevo software o una actualización de software.

En este apartado hemos tenido en cuenta la importancia de los casos de uso en una aplicación web. En nuestro caso en particular hemos querido que fueran lo más detallado posible. Todos estos que veremos a continuación han sido previamente estudiados y hablados con el tutor con el fin de intentar tener todos los posibles de cara al uso de la aplicación por parte de un educador. En ellos podremos ver tanto los casos de uso de un profesor, como de un administrador. Podrán diferenciarse en el subapartado *Descripción* en el cual se nombra el rol al cual va reflejado el caso de uso.

CU-01 Listado de alumnos.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 01.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de alumnos inscrito*.

Secuencia normal:

1. El administrador desea obtener la lista de alumnos inscritos.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de alumnos.
4. El administrador obtiene el listado de alumnos.

CU-02 Creación de alumnos.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 01.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de un alumno*.

Secuencia normal:

1. El administrador desea crear un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de alumnos.
4. El administrador obtiene el listado de alumnos.
5. El administrador solicita crear un alumno.
6. El administrador crea un alumno.

CU-03 Edición de alumnos.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 01.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la edición de un alumno*.

Secuencia normal:

1. El administrador desea editar un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de alumnos.
4. El administrador obtiene el listado de alumnos.
5. El administrador solicita la edición un alumno.
6. El administrador edita un alumno.

CU-04 Eliminación de alumnos.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 01.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de un alumno*.

Secuencia normal:

1. El administrador desea eliminar un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de alumnos.
4. El administrador obtiene el listado de alumnos.
5. El administrador solicita la eliminación un alumno.
6. El administrador elimina un alumno.

CU-05 Listado de profesores.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 02.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de profesores inscrito*.

Secuencia normal:

1. El administrador desea obtener la lista de profesores inscritos.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de profesores.
4. El administrador obtiene el listado de profesores.

CU-06 Creación de profesores.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 02.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de un profesor*.

Secuencia normal:

1. El administrador desea crear un profesor.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de profesores.
4. El administrador obtiene el listado de profesores.
5. El administrador solicita crear un profesor.
6. El administrador crea un profesor.

CU-07 Edición de profesores.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 10.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la edición de un profesor*.

Secuencia normal:

1. El administrador desea editar un profesor.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de profesores.
4. El administrador obtiene el listado de profesores.
5. El administrador solicita la edición un profesor.
6. El administrador edita un profesor.

CU-08 Eliminación de profesores.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 02.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de un profesor*.

Secuencia normal:

1. El administrador desea eliminar un profesor.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de profesores.
4. El administrador obtiene el listado de profesores.
5. El administrador solicita la eliminación un profesor.
6. El administrador elimina un profesor.

CU-09 Listado de Grupos de clase.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 05.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de grupo de clase*.

Secuencia normal:

1. El administrador desea obtener la lista de grupos de clase.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de grupos de clase.
4. El administrador obtiene el listado de grupos de clase.

CU-10 Creación de Grupos de clase.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 05.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de un grupo de clase*.

Secuencia normal:

1. El administrador desea crear un grupo de clase.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de grupos de clase.
4. El administrador obtiene el listado de grupos de clase.
5. El administrador solicita crear un grupo de clase.
6. El administrador crea un grupo de clase.

CU-11 Edición de Grupos de clase.

Versión: 1.1 (17/05/2021)

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la edición de un grupo de clase*.

Dependencias:

- RF 05.

Secuencia normal:

1. El administrador desea editar un profesor.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de grupos de clase.
4. El administrador obtiene el listado de grupos de clase.
5. El administrador solicita la edición un grupo de clase.
6. El administrador edita un grupo de clase.

CU-12 Eliminación de Grupos de clase.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 05.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de un grupo de clase*.

Secuencia normal:

1. El administrador desea eliminar un grupo de clase.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de grupos de clase.
4. El administrador obtiene el listado de grupos de clase.
5. El administrador solicita la eliminación un grupo de clase.
6. El administrador elimina un grupo de clase.

CU-13 Asignar alumnos a sus respectivos grupos de clase.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 06.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador asignar alumnos a sus respectivos grupos de clase*.

Secuencia normal:

1. El administrador desea asignar alumnos a sus respectivos grupos de clase.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de grupos de clase.
4. El administrador obtiene el listado de grupos de clase.
5. El administrador solicita la lista de alumnos no inscritos.
6. El administrador obtiene la lista de alumnos no inscritos.
7. El administrador solicita la asignación de un alumno.

CU-14 Listado de asignaturas.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 03.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de asignaturas*.

Secuencia normal

1. El administrador desea obtener la lista de asignaturas.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de asignaturas.
4. El administrador obtiene el listado de asignaturas.

CU-15 Creación de asignaturas.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 03.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de una asignatura*.

Secuencia normal:

1. El administrador desea crear una asignatura.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de asignaturas.
4. El administrador obtiene el listado de asignaturas.
5. El administrador solicita crear una asignatura.
6. El administrador crea una asignatura.

CU-16 Edición de asignaturas.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 03.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la edición de una asignatura*.

Secuencia normal:

1. El administrador desea editar una asignatura.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de asignaturas.
4. El administrador obtiene el listado de asignaturas.
5. El administrador solicita la edición una asignatura.
6. El administrador edita una asignatura.

CU-17 Eliminación de asignaturas.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 03.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *_el administrador solicite la eliminación de una asignatura*.

Secuencia normal:

1. El administrador desea eliminar un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de asignaturas.
4. El administrador obtiene el listado de asignaturas.
5. El administrador solicita la eliminación una asignatura.
6. El administrador elimina una asignatura.

CU-18 Asignar asignatura a profesores.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 06.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador asignar una asignatura a un profesor*.

Secuencia normal:

1. El administrador desea asignar una asignatura a un profesor.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de profesores.
4. El administrador obtiene el listado de profesores.
5. El administrador solicita la lista de asignaturas no inscritas.
6. El administrador obtiene la lista de asignaturas no inscritas.
7. El administrador solicita la asignación de una asignatura.

CU-19 Listado de competencias.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 07.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de competencias*.

Secuencia normal:

1. El administrador desea obtener la lista de competencias.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de competencias.
4. El administrador obtiene el listado de competencias.

CU-20 Creación de competencias.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 07.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de una competencia*.

Secuencia normal:

1. El administrador desea crear una competencia.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de competencias.
4. El administrador obtiene el listado de competencias.
5. El administrador solicita crear una competencia.
6. El administrador crea una competencia.

CU-21 Edición de competencias.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 07.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la edición de una competencia*.

Secuencia normal:

1. El administrador desea editar una competencia.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de competencias.
4. El administrador obtiene el listado de competencias.
5. El administrador solicita la edición una competencia.
6. El administrador edita una competencia.

CU-22 Eliminación de competencias.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 07.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de una competencia*.

Secuencia normal:

1. El administrador desea eliminar una competencia.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de competencias.
4. El administrador obtiene el listado de competencias.
5. El administrador solicita la eliminación una competencia.
6. El administrador elimina una competencia.

CU-23 Asignación de competencias.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 08.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la asignación de competencias a una asignatura*.

Secuencia normal:

1. El administrador desea asignar una competencia a una asignatura.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de asignaturas.
4. El administrador obtiene el listado de asignaturas.
5. El administrador solicita la lista de competencias.
6. El administrador obtiene el listado de competencias.
7. El administrador asigna la competencia de una asignatura.

CU-24 Listado de administradores.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 09.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de administradores*.

Secuencia normal:

1. El administrador desea obtener la lista de administradores.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de administradores.
4. El administrador obtiene el listado de administradores.

CU-25 Creación de administradores.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 09.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de un nuevo administrador*.

Secuencia normal:

1. El administrador desea crear un administrador.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de administradores.
4. El administrador obtiene el listado de administradores.
5. El administrador solicita crear un administrador.
6. El administrador crea un administrador.

CU-26 Eliminación de administradores.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 09.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de un administrador*.

Secuencia normal:

1. El administrador desea eliminar un administrador.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de administradores.
4. El administrador obtiene el listado de administradores.
5. El administrador solicita la eliminación un administrador.
6. El administrador elimina un administrador.

CU-28 Listar sus grupos de clase.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 11.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite listar sus grupos de clase*.

Secuencia normal:

1. El profesor desea listar sus grupos de clase.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos de clase.
4. El profesor obtiene el listado de sus grupos de clase.

CU-29 Listar sus asignaturas.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 12.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite listar sus asignaturas*.

Secuencia normal:

1. El profesor desea listar sus asignaturas.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus asignaturas.
4. El profesor obtiene el listado de sus asignaturas.

CU-30 Listar las competencias de una asignatura.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 13.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite listar las competencias de una asignatura*.

Secuencia normal:

1. El profesor desea listar las competencias de una asignatura.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de competencias de una asignatura.
4. El profesor obtiene el listado de competencias de una asignatura.

CU-31 Listar sus alumnos.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 15.

Precondición: El profesor ha seleccionado el tipo de evaluación.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite listar sus alumnos*.

Secuencia normal:

1. El profesor desea listar sus alumnos.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de sus alumnos.
6. El profesor obtiene el listado de sus alumnos.

CU-32 Listar actividades.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 16.

Precondición: El profesor ha seleccionado el tipo de evaluación.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite actividades*.

Secuencia normal:

1. El profesor desea listar actividades.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de actividades.
4. El profesor obtiene el listado de actividades.

CU-33 Creación de actividades.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 16.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la creación de una actividad*.

Secuencia normal:

1. El profesor desea crear una actividad.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor crea una actividad.

CU-34 Edición de una actividad.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 16.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la edición de una actividad*.

Secuencia normal:

1. El profesor desea editar una actividad.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor solicita los detalles de la actividad.
8. El profesor obtiene los detalles de la actividad.
9. El profesor edita una actividad.

CU-35 Eliminación de una actividad.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 16.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la eliminación de una actividad*.

Secuencia normal:

1. El profesor desea editar una actividad.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor solicita los detalles de la actividad.
8. El profesor obtiene los detalles de la actividad.
9. El profesor elimina una actividad.

CU-36 Ver las calificaciones de un alumno.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 17.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite ver las calificaciones de un alumno*.

Secuencia normal:

1. El profesor desea ver las calificaciones de un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita la lista de alumnos de ese grupo.
6. El profesor obtiene la lista de alumnos de ese grupo.
7. El profesor ve las calificaciones de un alumno.

CU-37 Listar ejercicios.

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 18.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite el listado de ejercicio*.

Secuencia normal:

1. El profesor desea listar ejercicios.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de ejercicios.
4. El profesor obtiene el listado de ejercicios.

CU-38 Creación de ejercicios.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 18.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la creación de un ejercicio*.

Secuencia normal:

1. El profesor desea crear un ejercicio.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor solicita los detalles de la actividad.
8. El profesor obtiene los detalles de la actividad.
9. El profesor crea un ejercicio.

CU-39 Edición de un ejercicio.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 18.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la edición de un ejercicio*.

Secuencia normal:

1. El profesor desea editar un ejercicio.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor solicita los detalles de la actividad.
8. El profesor obtiene los detalles de la actividad.
9. El profesor edita un ejercicio.

CU-40 Eliminación de un ejercicio.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 18.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la eliminación de un ejercicio*.

Secuencia normal:

1. El profesor desea eliminar un ejercicio.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor solicita los detalles de la actividad.
8. El profesor obtiene los detalles de la actividad.
9. El profesor elimina un ejercicio.

CU-41 Asignar competencias a un ejercicio con una intensidad.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 19.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la asignar competencias a un ejercicio con una intensidad*.

Secuencia normal:

1. El profesor desea asignar competencias a un ejercicio con una intensidad.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita las actividades de un grupo.
6. El profesor obtiene las actividades de un grupo.
7. El profesor solicita los detalles de la actividad.
8. El profesor obtiene los detalles de la actividad.
9. El profesor solicita modificar un ejercicio.
10. El profesor asigna una competencia a un ejercicio con una intensidad.

CU-42 Puntuar las evaluaciones, las actividades, los ejercicios y las competencias.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 20.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite puntuar las evaluaciones, las actividades, los ejercicios y las competencias*.

Secuencia normal:

1. El profesor desea puntuar las evaluaciones, las actividades, los ejercicios y las competencias.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de sus grupos.
4. El profesor obtiene el listado de sus grupos.
5. El profesor solicita la lista de alumnos de ese grupo.
6. El profesor obtiene la lista de alumnos de ese grupo.
7. El profesor solicita ver las calificaciones.
8. El profesor obtiene las calificaciones.
9. El profesor solicita calificar las actividades.
10. El profesor solicita calificar los ejercicios.
11. El profesor solicita calificar las competencias.
12. El profesor puntuá las competencias.
13. El profesor obtiene las calificaciones de las competencias, ejercicios y actividades y evaluación.

CU-43 Modificar sus datos de usuario.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 24.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite modificar sus datos de usuario*.

Secuencia normal:

1. El profesor desea modificar sus datos de usuario.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la modificación sus datos de usuario.
4. El profesor modifica sus datos de usuario.

CU-44 Generar informe de un grupo completo.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 21.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite generar informe de un grupo completo*.

Secuencia normal:

1. El profesor desea generar informe de un grupo completo.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de alumnos.
6. El profesor obtiene el listado de alumnos.
7. El profesor solicita el informe.
8. El profesor exporta el informe.

CU-45 Generar informe de un alumno.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 21.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite generar informe de un alumno*.

Secuencia normal:

1. El profesor desea generar informe de un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de alumnos.
6. El profesor obtiene el listado de alumnos.
7. El profesor ve las calificaciones de un alumno.
8. El profesor exporta el informe.

CU-46 Ver competencias de un alumno.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 22.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite ver las competencias de un alumno*.

Secuencia normal:

1. El profesor desea ver las competencias de un alumno.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de alumnos.
6. El profesor obtiene el listado de alumnos.
7. El profesor solicita las competencias de un alumno.
8. El profesor obtiene las competencias de un alumno.

CU-47 Creación de evaluaciones.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 04.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la creación de una evaluación*.

Secuencia normal:

1. El administrador desea crear una evaluación.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de evaluaciones.
4. El administrador obtiene el listado de evaluaciones.
5. El administrador solicita crear una evaluación.
6. El administrador crea una evaluación.

CU-48 Edición de evaluaciones.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 04.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la edición de una evaluación*.

Secuencia normal:

1. El administrador desea editar una evaluación.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de evaluaciones.
4. El administrador obtiene el listado de evaluaciones.
5. El administrador solicita la edición una evaluación.
6. El administrador edita una evaluación.

CU-49 Eliminación de evaluaciones.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 04.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite la eliminación de una evaluación*.

Secuencia normal:

1. El administrador desea eliminar una evaluación.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de evaluaciones.
4. El administrador obtiene el listado de evaluaciones.
5. El administrador solicita la eliminación una evaluación.
6. El administrador elimina una evaluación.

CU-50 Listado de evaluaciones.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 04.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el administrador solicite el listado de las evaluaciones*.

Secuencia normal:

1. El administrador desea obtener la lista de evaluaciones.
2. El sistema solicita las credenciales para identificarse.
3. El administrador solicita la lista de grupos de clase.
4. El administrador obtiene el listado de evaluaciones.

CU-51 Listado de bloques (evaluaciones).

Versión: 1.1 (17/05/2021)

Dependencias:

- RF 23.

Precondición: El profesor ha seleccionado el tipo de evaluación.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite el listado de los bloques*.

Secuencia normal:

1. El profesor desea obtener la lista de bloques.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de bloques de un grupo de clase.
6. El profesor obtiene el listado de bloques.

CU-52 Creación de bloques (evaluaciones).

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 23.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la creación de un bloque*.

Secuencia normal:

1. El profesor desea crear un bloque.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de bloques de un grupo de clase.
6. El profesor obtiene el listado de bloques.
7. El profesor solicita la creación de un bloque.
8. El profesor crea un bloque.

CU-53 Edición de bloques (evaluaciones).

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 23.

Precondición: El profesor ha creado un bloque.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la edición de un bloque*.

Secuencia normal:

1. El profesor desea editar un bloque.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de bloques de un grupo de clase.
6. El profesor obtiene el listado de bloques.
7. El profesor solicita la edición de un bloque.
8. El profesor edita un bloque.

CU-54 Eliminación de bloques (evaluaciones).

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 23.

Precondición: El profesor ha creado un bloque.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando *el profesor solicite la eliminación de un bloque*.

Secuencia normal:

1. El profesor desea eliminar un bloque.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita la lista de bloques de un grupo de clase.
6. El profesor obtiene el listado de bloques.
7. El profesor solicita la eliminación de un bloque.
8. El profesor elimina un bloque.

CU-55 Seleccionar el tipo de evaluación para las evaluaciones de un grupo de clase.

Versión: 1.0 (17/05/2021)

Dependencias:

- RF 14.

Descripción: El sistema deberá comportarse como se describe en el siguiente caso de uso cuando _el profesor solicite seleccionar el tipo de evaluación para las evaluaciones de un grupo de clase.

Secuencia normal:

1. El profesor desea seleccionar el tipo de evaluación para las evaluaciones de un grupo de clase.
2. El sistema solicita las credenciales para identificarse.
3. El profesor solicita la lista de grupos de clase.
4. El profesor obtiene el listado de grupos de clase.
5. El profesor solicita seleccionar el tipo de evaluación.
6. El profesor selecciona el tipo de evaluación.

Postcondición: Se han habilitado las opciones de listar bloques, actividades y estudiantes.

III.III Modelo de datos

Sabemos que un parte muy importante del desarrollo de una aplicación web es el modelado de los datos. Este no es más que una estructura que nos permite visualizar,

especificar y documentar que datos se van a guardar y cómo. Además, sirve de punto de apoyo para construir la aplicación, algo que es crucial y que se asemeja a lo que serían los cimientos de una casa. Por ello hemos realizado una gran implicación y dedicación al modelo con el fin de que se pudiera construir toda la funcionalidad alrededor de este.

A lo largo del apartado iremos mostrando las partes más relevantes del modelo de datos mediante el uso de diagramas. En primer lugar, veremos la totalidad del modelo en la siguiente imagen.

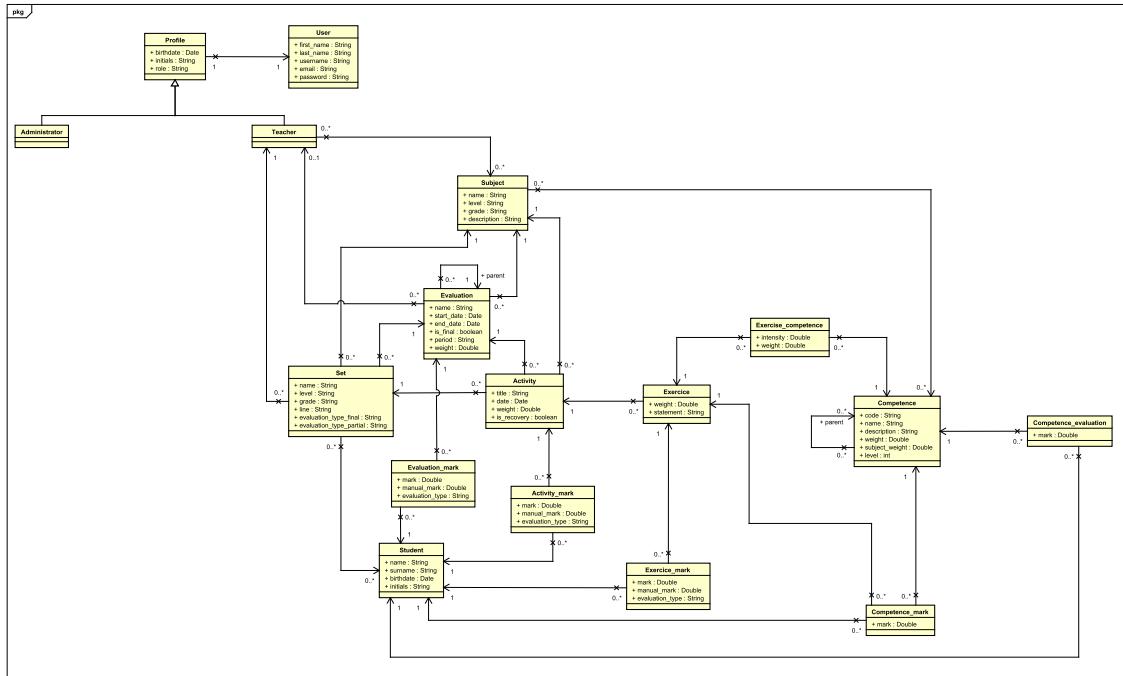


Figura 1. UML

Para gestionar los actores que interactúan con el sistema hemos diseñado una estructura que parte de una clase, es decir, un arquetipo para la creación de datos, llamada *User* (Usuario) proveniente de [Django](#). Este objeto recoge nombre, apellidos y datos de registro como el nombre de usuario, el correo electrónico y la contraseña. Para almacenar más datos hemos creado el objeto *Profile* (Perfil) que servirá como patrón de los objetos que representan los actores del sistema, *Administrator* (Administrador) y *Teacher* (Profesor). Lo veremos gráficamente en la siguiente imagen.

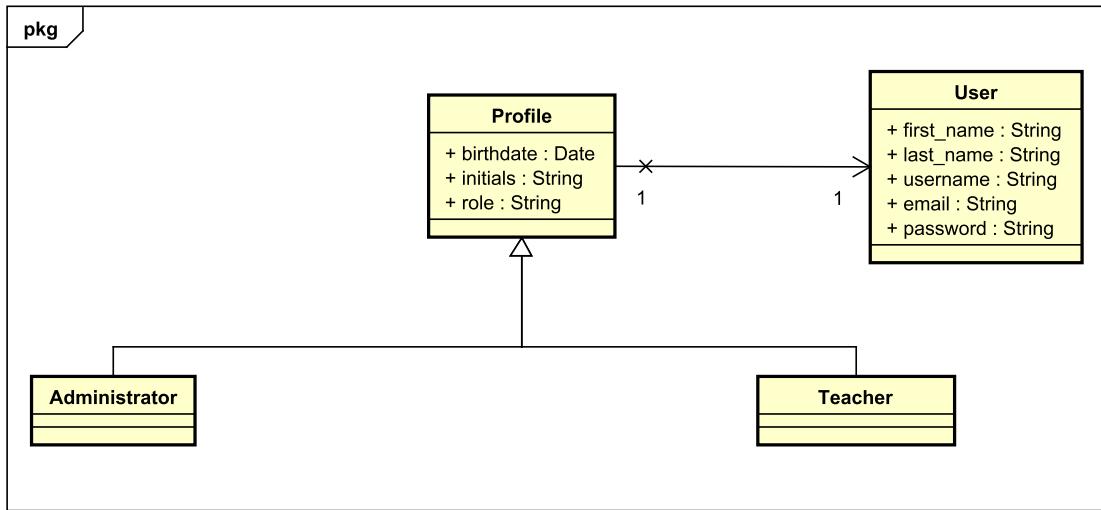


Figura 2. UML de usuarios

Hemos creado una clase para cada una de las partes que se pueden evaluar. Estas son *Evaluation* (Evaluación), *Activity* (Actividad), *Exercise* (Ejercicio) y *Competence* (Competencia). Lo veremos mejor en la siguiente imagen.

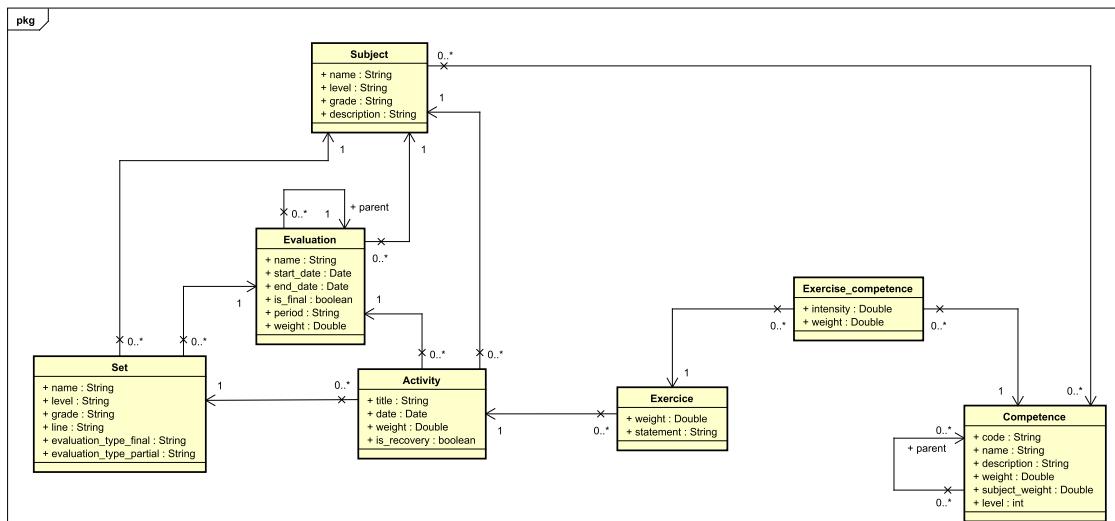


Figura 3. UML de evaluación

Necesitábamos una clase intermedia entre *Exercise* (Ejercicio) y *Competence* (Competencia) que pudiese almacenar una serie de datos que no podrían almacenar en ninguna de las dos, ya que estos no se podrían almacenar en una sola clase. Esta clase es *Exercise_competence* que, como podemos ver en la siguiente imagen, guarda *intensity* (intensidad) y *weight* (peso). La diferencia entre estos conceptos es que la primera marca la "fuerza" con que se debe utilizar la competencia en el ejercicio, y la segunda es la importancia que tiene la competencia en el ejercicio.

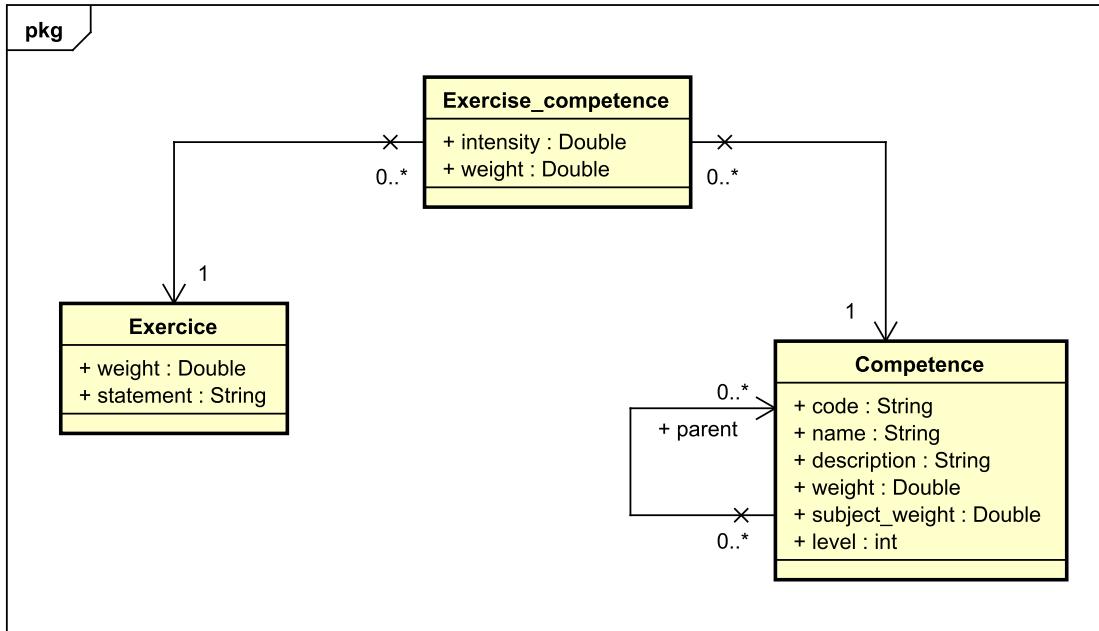


Figura 4. UML de competencias de un ejercicio

Para almacenar los tres niveles de competencias hemos optado por crear una sola clase que se relacione con siyo misma. Lo veremos más en detalle en la siguiente imagen.

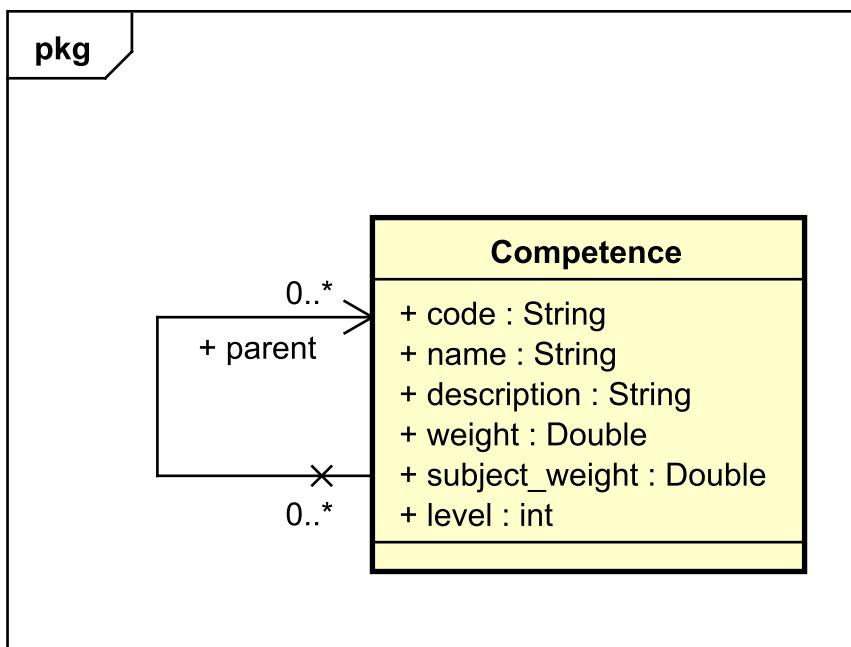


Figura 5. UML de competencia

Para almacenar las notas hemos creado una clase para cada una de las clases que se pueden puntuar, es decir, *Evaluation_mark* para *Evaluation*(Evaluación), *Activity_mark* para *Activity* (Actividad), *Exercise_mark* para *Exercise* (Ejercicio) y, *Competence_mark* y *Competence_evaluation* para *Competence* (Competencia). La diferencia entre *Competence_mark* y *Competence_evaluation* es que la primera clase sirve para

almacenar la nota de una competencia dentro de un ejercicio concreto y, la segunda es para la nota total de la competencia. Lo veremos gráficamente en la siguiente imagen.

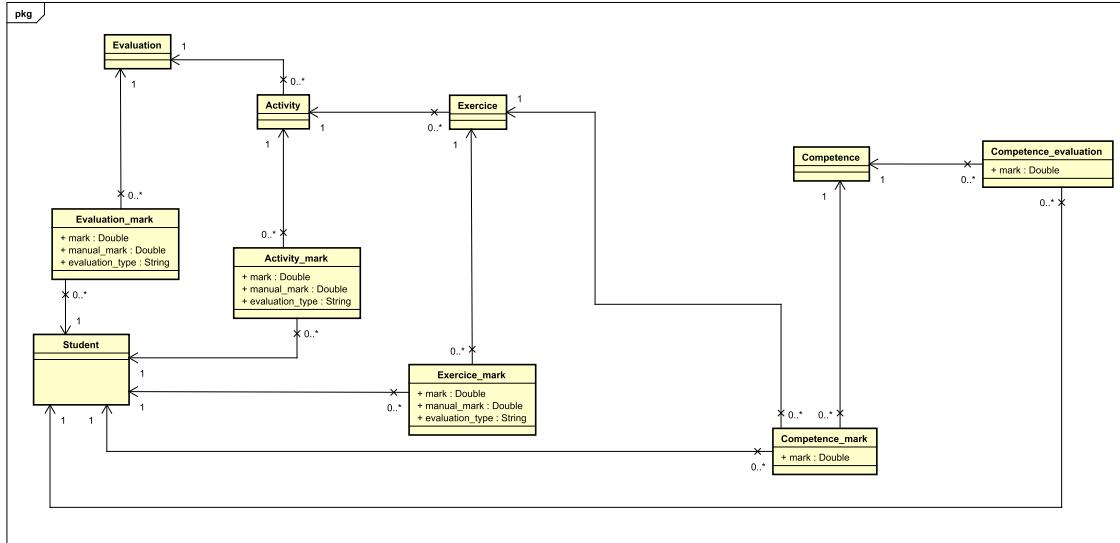


Figura 6. UML de notas

III.IV Mockups

En esta sección mostraremos algunos de los mockups creados previamente al desarrollo de la aplicación. El resto de *mockups* se encuentran disponibles dentro del repositorio de GitHub, en la sección docs/mockups.

También en el directorio de docs podremos encontrar mockups-interactivos, en los que podremos ver de manera interactiva todos los mockups creados.

Dividiremos esta guía en dos partes, administrador y profesor.

III.IV.I Administrador

Aquí podremos ver la creación de un alumno:

LOGO

INICIO ADMINISTRADORES COMPETENCIAS ALUMNOS PROFESORES GRUPOS ASIGNATURAS



Crear alumno

Nombre
Manuel

Apellidos
Copado Machuca

Fecha de nacimiento
29/07/1965

Iniciales
Mancopmac

Email
mancopmac@us.es

Crear

La creación de un grupo:

LOGO

INICIO ADMINISTRADORES COMPETENCIAS ALUMNOS PROFESORES GRUPOS ASIGNATURAS



Crear grupo

Nivel
Select...

Curso
Select...

Línea
Select...

Asignatura

Matemática
Matemáticas 2
Matemáticas Primaria

Tutor
Fernando F

Fernando Fernández González
Fernando Fernández Arévalo

Crear

El listado de asignación de unos alumnos a un grupo:

LOGO

INICIO ADMINISTRADORES COMPETENCIAS ALUMNOS PROFESORES GRUPOS ASIGNATURAS

Asignación de alumnos a grupo

Lina Bo

Buscar

BASIC INFO	COMPANY	PHONE	TAGS	CREATED DATE	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Company Name	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	

La eliminación de estos de un grupo:

LOGO

INICIO ADMINISTRADORES COMPETENCIAS ALUMNOS PROFESORES GRUPOS ASIGNATURAS

Eliminar alumnos de un grupo

Lina Bo

Buscar

BASIC INFO	COMPANY	PHONE	TAGS	CREATED DATE	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Company Name	012 389 438	POPULAR NEW JOB	09 Sep 2017	
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017	

III.IV.II Profesor

El listado de alumnos de un grupo al que pertenece:

LOGO

INICIO MIS ACTIVIDADES MIS GRUPOS MIS ASIGNATURAS

Alumnos Grupo 3º B

Informes

BASIC INFO	COMPANY	PHONE	TAGS
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB
<input type="checkbox"/>  Lina Bo lina@adobe.com	Company Name	012 389 438	POPULAR NEW JOB
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB

La asignación de una competencia a una asignatura:

LOGO

INICIO ADMINISTRADORES COMPETENCIAS ALUMNOS PROFESORES GRUPOS ASIGNATURAS

Asignación de competencia a asignatura

Lina Bo

Buscar

BASIC INFO	COMPANY	PHONE	TAGS	CREATED DATE
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/>  Lina Bo lina@adobe.com	Company Name	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →
<input type="checkbox"/>  Lina Bo lina@adobe.com	Adobe	012 389 438	POPULAR NEW JOB	09 Sep 2017 →

Poder ver las calificaciones de un alumno:

LOGO INICIO MIS ACTIVIDADES MIS GRUPOS MIS ASIGNATURAS

Competencias Generales Alberto Souto Cruz

C-01: Nombre C-01 7.2
Descripción C-01 Competencias Específicas

C-02: Nombre C-02 7.2
Descripción C-02 Competencias Específicas

C-03: Nombre C-03 6.3
Descripción C-03 Competencias Específicas

C-04: Nombre C-04 9.1
Descripción C-04 Competencias Específicas

Poder puntuar las calificaciones mediante competencias:

LOGO INICIO MIS ACTIVIDADES MIS GRUPOS MIS ASIGNATURAS

Calificaciones Alberto Souto Cruz

Competencias Ejercicio 3

Competencia 1: Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.
Nota Guardar

Competencia 2: Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión.
Nota Guardar

Modificar un ejercicio:

The screenshot shows a web interface for modifying exercises. At the top, there are navigation links: INICIO, MIS ACTIVIDADES, MIS GRUPOS, and MIS ASIGNATURAS. A user profile icon is also present.

Modificar ejercicio

Enunciado

Estimación del porcentaje de alumnos que pagan 550 € o más por la matrícula en una población de tamaño infinito.

Peso

Select... Select... Modificar

BASIC INFO	COMPANY	PHONE	Peso	Intensidad
<input type="checkbox"/> Lina Bo lina@adobe.com	Adobe	012 389 438	Select.✓	Select.✓ → ✕
<input type="checkbox"/> Lina Bo lina@adobe.com	Adobe	012 389 438	Select.✓	Select.✓ → ✕

Capacidad o Buscar

BASIC INFO	COMPANY	PHONE	Peso	Intensidad
<input type="checkbox"/> Lina Bo lina@adobe.com	Adobe	012 389 438	Select.✓	Select.✓ →
<input type="checkbox"/> Lina Bo lina@adobe.com	Adobe	012 389 438	Select.✓	Select.✓ →
<input type="checkbox"/> Lina Bo lina@adobe.com	Adobe	012 389 438	Select.✓	Select.✓ →
<input type="checkbox"/> Lina Bo lina@adobe.com	Company Name	012 389 438	Select.✓	Select.✓ →
<input type="checkbox"/> Lina Bo lina@adobe.com	Adobe	012 389 438	Select.✓	Select.✓ →

III.V Justificación de las tecnologías y herramientas utilizadas

A la hora de empezar la búsqueda de las tecnologías tuvimos varios factores a tener en cuenta. Uno de ellos era la profesionalidad y otro era la fiabilidad.

En cuanto a la búsqueda del lenguaje de programación a usar, ambos lo tuvimos claro. Sabíamos del potencial de Python para el uso de aplicaciones web. Fue elegido por su madurez algo que más abajo se justifica en detalle.

El resto de las aplicaciones usadas durante el desarrollo fue elegido por que queríamos usar entornos lo más parecidos a la realidad posibles. A causa de esto elegimos Jira, ya que es una aplicación web para la gestión de trabajo, y está especializada en metodologías ágiles, algo que íbamos a usar para el desarrollo.

Este apartado será desglosado en dos: Tecnologías y Herramientas donde hablaremos de manera más específicas de cada una de ellas, y el por qué la elegimos.

III.V.I Tecnologías

Tabla 15. Django

Django

Este es uno de los mejores entornos de trabajo, o [Framework](#) de Python muy completo y con una buena documentación. Tenemos experiencia en él, ya que lo hemos utilizado en otras asignaturas, y nos gustaría seguir aprendiendo y adquirir experiencia en Django para nuestro futuro profesional.

Alternativas

Existen otros frameworks de Python como:

- Flask.
- Bottle.
- Pyramid.

Bottle y Flask son *microframework* mucho más simples que Django y se utilizan para proyectos más sencillos que el nuestro.

Pyramid es una buena opción, mucho más ligero que Django, esto provoca que sea menos completo. Django no necesita otras librerías para el desarrollo de la aplicación, al contrario que Pyramid, que aparte del enrutamiento y la autenticación, necesita librerías externas.

NodeJs es otra buena alternativa. Es un framework que está creciendo continuamente en popularidad con multitud de librerías que lo nutren. Esto también juega en su contra, ya que NodeJs no dispone de una reputación tan sólida, ni unas librerías externas tan bien documentadas como Django. Tampoco dispone de un sistema sólido de seguridad como el que tiene Django, el cual, es muy fácil de implementar. En otro apartado, Django permite una velocidad de desarrollo muy alta, algo de lo que no dispone NodeJs, por lo que la elección de este último extendería más la etapa de desarrollo, y lo haría menos económico.

Tabla 16. Materialize

Materialize

Materialize es un [Framework](#) dispuesto a facilitar el diseño web en todo tipo de escenarios, escritorio o móvil. Habiendo sido recomendada y después de haber pasado por un proceso de selección, su amplia gama de posibilidades y recursos, junto a su facilidad de uso, fue sin duda el detonante de su elección.

Alternativas

Existen algunas alternativas como:

- Boostrap
- Foundation

A diferencia de las alternativas, Materialize se basa en la normativa Material design. La creación de esta normativa por parte de Google hace que sea algo con lo que los usuarios están más familiarizados a usar. Es de esperar que su soporte.

Materialize no necesita jQuery para su uso, aunque puedes usarlo si así lo deseas.

Otro punto a su favor en comparación con Bootstrap o Foundation es su estética, la cual es muy atractiva, obteniendo un resultado homogéneo.

Uno de los puntos negativos por parte de Bootstrap son los problemas entre versión y versión, ya que cambia la nomenclatura de sus elementos, cosa que con Materialize no ocurre.

A diferencia de sus alternativas, Materialize tiene una gran documentación, la cual, es más fácil de entender para alguien que acaba de empezar, así como una amplia cantidad de ejemplos visuales.

Gráficamente Materialize está muy por encima de Boostrap o Foundation, debido a su variedad de color y a sus componentes adicionales.

Tabla 17. PostgreSQL

PostgreSQL

Es una base de datos SQL gratuita diseñada para ambientes de alto volumen. Tenemos experiencia en esta base de datos por varias asignaturas. Además, dispone de una herramienta gráfica muy útil que facilita mucho su utilización.

Alternativas

Existen otras alternativas como:

- MySQL
- MariaDB

MySQL es la opción más extendida, pero su licencia puede ser un problema, ya que está sujeta a las decisiones de Oracle. La segunda alternativa es MariaDB, un fork de MySQL de código abierto. La gran diferencia de PostgreSQL con esta última, como hemos comentado anteriormente, es que está pensada para alto volumen. Este ha sido el origen de nuestra elección.

Heroku

Es una plataforma en nube gratuita que hemos utilizado anteriormente que está recomendada cuando se usa junto a Django y PostgreSQL.

Alternativas

Existen algunas alternativas como:

- AWS
- Firebase

A diferencia de la versión gratuita de Firebase, la cual, es bastante simple, teniendo limitaciones en el espacio de almacenamiento y el número de usuarios simultáneos. Por tanto, para trabajar con ella de manera profesional, deberás pagar, en cambio, Heroku es gratuito para aplicaciones de poco consumo.

Comparándola con Heroku, AWS no es apto para personas principiantes, ya que la complejidad del mismo es bastante elevada.

En comparación con las alternativas expuestas, Heroku permite desplegar versiones, *rollbacks*, gestionar dependencias, como también desarrollar casi con cualquier lenguaje de programación: Java, Ruby, NodeJS...

Tabla 18. Travis

Travis

Es una herramienta de integración continua que hemos usado en varias asignaturas que nos va a permitir ejecutar nuestros test y comprobar si han surgido fallos al añadir código al repositorio, además vamos a poder conectarlo con SonarCloud.

Alternativa

La segunda opción es Jenkins. La mayor ventaja que podemos destacar de Travis frente Jenkins es no que necesita ningún tipo de instalación, solo se configura un solo archivo y se puede utilizar a la perfección.

Tabla 19. Sonarcloud

Sonarcloud

Nos va a permitir analizar el código de nuestra aplicación de forma automática conectándose a GitHub mediante Travis.

Alternativas

Existen algunas alternativas como:

- Codacy
- SonarQube

A diferencia de Sonarcloud Codacy no tiene soporte para Git privado o Azure DevOps git.

SonarQube tienen un mantenimiento de código más costoso, haciendo que sea necesario invertir más tiempo, así como generar "falsos positivos".

Tabla 20. Asciidoc

Asciidoc

La forma más habitual de redactar sería en formato Word, pero veíamos necesario que toda la documentación estuviese junto al código. Por eso, elegimos Asciidoc, que nos permite redactar nuestros documentos de forma personalizada con un lenguaje sencillo y fácil de aprender.

Alternativa

Como segunda opción tenemos a Markdown, un lenguaje de marcado ligero más extendido que Asciidoc, por lo que hay muchas más herramientas que lo soportan. Al contrario que Asciidoc, carece de un estándar, por lo que al redactar algo con una ligera complejidad, cada herramienta puede interpretarlo de distinta forma.

III.V.II Herramientas

Tabla 21. Adobe XD

Adobe XD

Elegimos Adobe XD por la gran versatilidad en sus apartados para la realización de *mockups* para el proyecto. Dada a su gran facilidad de uso y sus múltiples plantillas y extensiones, su elección fue en nuestra opinión lo más recomendable. También nos ha permitido hacer *mockups* interactivos de forma sencilla.

Tabla 22. Visual Studio Code

Visual Studio Code

Es un editor de código desarrollado por Microsoft y que está siendo muy usado estos últimos años. Nos permite programar en múltiples lenguajes de forma personalizada con las numerosas extensiones que acompañan al editor. Además, implementa una consola de depuración y está vinculado directamente con Git. Lo elegimos, aparte de los puntos anteriores, porque tenemos experiencia en él y es fácil de usar.

Tabla 23. Clockify

Clockify

Nos permite medir nuestro gasto temporal en nuestras tareas de manera sencilla. Existen muchas herramientas de este tipo, pero esta es la que más hemos utilizado, y nos sirve perfectamente para nuestro trabajo.

Tabla 24. Github

Github

Poco que decir en este apartado, GitHub es la plataforma de control de versiones más utilizada. Dado su conocimiento de uso por parte de los miembros del grupo de trabajo, decidimos sin dudarlo su elección.

Tabla 25. Jira

Jira

Jira es una herramienta de gran utilidad para equipos de todo tipo. Siendo también de gran ayuda para gestión de trabajo en todo tipo de casos, y en especial para el desarrollo *software*. Con todo esto y con el conocimiento de que se usa en entornos laborales de grandes empresas como Everis, decidimos utilizarla y aprender cómo funciona.

III.VI Gestión de código

Este apartado es una parte crucial a la hora de trabajar en grupo con un *software*. En nuestro caso hemos usado Git, el cual es una gran herramienta que se basa en la creación de ramas para el trabajo en paralelo.

Algo muy importante a la hora de trabajar con Git es llevar una metodología, nosotros hemos usado la más conocida por parte de ambos: GitFlow. Esto ha dado lugar a una mejora del desarrollo continuo de las prácticas.

Dividiremos este apartado en tres con el fin de desglosar de manera más detallada los mismos: Gestión de ramas, Gestión de commits y Pull requests

III.VI.I Gestión de ramas

Una rama es un espacio de trabajo paralelo que deriva de una versión común. Hemos sido muy estrictos en este apartado, en el cual hemos seguido las siguientes pautas.

Tenemos dos Ramas por defecto:

- master
- dev

En la rama **dev** se incluirán todas las ramas de desarrollo con **pull requests**. La rama **master** solo se podrán añadir desde la rama **dev**, para incluir la release.

El resto de las ramas seguirán el siguiente esquema:

<codigo_rama>/<codigo_sprint>/<nombre_tarea>

Códigos de rama (<codigo_rama>):

- feat/ (nueva feature)
- fix/ (bug fix)
- hotfix/ (bug fixes en producción)
- refactor/ (refactorización de código)
- docs/ (cambios de documentación)
- test/ (añadir o refactorizar tests)
- release/ (nueva release)

Códigos de sprint (<codigo_sprint>):

- s0 (previo al comienzo del desarrollo de código)
- s1/s2/s3/s4 (sprints 1, 2, 3, 4 respectivamente)
- d (tras la finalización del desarrollo, realización de memoria)

Nombre de las tareas (<nombre_tarea>): Nombre especificado en Jira, o abreviación. Por ejemplo, CU-07 Edición de profesores, o simplemente, CU-07. Se pueden añadir varias tareas separadas por &.

III.VI.II Gestión de commits

El comando git commit se basa en guardar todos los cambios realizados en local junto a una breve descripción del usuario con el fin de dar sentido al trabajo realizado.

Los commits deben seguir el siguiente patrón: [**<tipo_commit>**] **<nombre_commit>**. Además, se puede añadir una descripción opcionalmente.

Tipo de commit (<tipo_commit>):

- feat (nueva feature)
- fix (bug fix)
- refactor (refactorización de código)
- doc (cambios de documentación)
- test (añadir o refactorizar tests)
- release (nueva release)

III.VI.III Pull requests

Para añadir funcionalidad a la rama **dev** se abrirá una pull request. Esto se basa en hacer una petición de la tarea realizada, con el fin de añadirla a el proyecto. El que haya desarrollado la funcionalidad se asignará a la pull request y asignará al compañero como revisor. Para que la pull request se como válida debe de pasar el

control y análisis de Travis y Sonarcloud, además del visto bueno del revisor.

Una vez que se haya hecho merge a dev, la rama se eliminará.

III.VII Gestión de versiones

En este apartado detallaremos como hemos nombrado a las versiones relacionadas a cada uno de los incrementos de funcionalidad. La especificación de la versión está marcada por el siguiente patrón: **X.Y.Z**, donde:

- **X**: Versión mayor.
- **Y**: Versión menor.
- **Z**: Versión en el que se solventan *bugs* de la versión anterior.

Las versiones existentes son las siguientes:

- **0.1**: Versión que incorpora todo el sprint 1. Incluye la primera parte de la funcionalidad del administrador, así como el modelo de datos y el primer *populate*.
- **0.2**: Versión que incorpora todo el sprint 2. Incluye más funcionalidad del administrador.
- **0.2.1**: Corrección de algunos errores de la versión **0.2**.
- **0.3**: Versión que incorpora todo el sprint 3. Incluye la mayor parte de la funcionalidad del profesor.
- **0.4**: Versión que incorpora todo el sprint 4. Incluye el resto de funcionalidad.

III.VIII Fuentes e Imágenes

El apartado visual dentro de una aplicación web se le conoce como el *frontend*. Este tiene muchos factores, el color, la forma, las imágenes, etc. En nuestro caso le pusimos mucho hincapié a todo esto, pero especialmente al apartado de las imágenes y la fuente.

Para ambas cosas nos pusimos a estudiar, ya que este apartado tiene temas de derechos, algo que es un poco controvertido. Estuvimos buscando algo que se adaptara lo mejor posible a nuestra web, algo cotidiano y sencillo. Nuestra finalidad era que una persona que no tuviese ningún conocimiento sobre la informática considerase nuestra web acogedora y visualmente atractiva.

A continuación, hablaremos de la fuente escogida y de dónde hemos seleccionado las imágenes mostradas en la aplicación.

III.VIII.I Fuente

La opción a la hora de seleccionar una fuente ha sido elegida con el fin de evitarnos problemas jurídicos, así como poderla usar en aplicaciones de fines lucrativos.

Este fué uno de los factores más importantes de cara a la elección de una fuente. Otro de ellos fué la elección de una fuente que se use habitualmente y que nos fuera de fácil acceso y visualmente atractiva.

III.VIII.I.I Elección

Nos decantamos por la Fuente Ubuntu, la cual obtuvimos del siguiente link:

<https://design.ubuntu.com/font/>

La licencia de esta fuente es *copyleft* permite que las tipografías sean «usadas, estudiadas, modificadas y redistribuidas libremente» cumpliendo los términos estipulados en ella.

La documentación de la licencia se encuentra en la carpeta *fonts*, cuyo nombre es "LICENSE.txt".

III.VIII.II Imágenes

Esta tarea sin duda fue una de las más difíciles de encontrar, puesto que todos los bancos de imágenes tenían derechos de autor y muchas trabas legales para su uso.

La opción a la hora de seleccionar las imágenes para la web fue difícil, ya que buscábamos un banco de imágenes que nos permitiera el uso comercial pensando en un hipotético uso de la misma y sin derechos de autor.

III.VIII.II.I Elección

Nos decantamos por la web <https://unsplash.com/>, la cual tiene un gran número de imágenes muy útiles, y una licencia que nos permitía lo nombrado anteriormente.

Con respecto a la licencia, dice lo siguiente:

Unsplash photos are made to be used freely. Our license reflects that.

- *All photos can be downloaded and used for free.*
- *Commercial and non-commercial purposes.*
- *No permission needed (though attribution is appreciated!).*

What is not permitted ↴

- *Photos cannot be sold without significant modification.*
- *Compiling photos from Unsplash to replicate a similar or competing service.*

Para más información puede consultar dicha licencia en el siguiente enlace:
<https://unsplash.com/license>

III.IX Implementación del cálculo

Existen muchas formas de realizar el cálculo de las notas, y dado que no podemos implementarlas todas, se desarrollará de tal forma que sea posible añadir nueva funcionalidad para el cálculo. Para que el profesor pueda modificar cómo se va a realizar el cálculo, se ha decidido almacenar esa información en el grupo, así todos los alumnos tendrán el mismo método y no supone un coste computacional muy alto. Se ha separado la selección del método del cálculo, pudiéndose elegir un método de cálculo para la evaluación final y otro para las evaluaciones parciales, para que sea más flexible.

Todos los cálculos que se realizan en el proyecto están normalizados, por lo que los pesos son bastantes flexibles y no están limitados de 0 a 10 o de 0 a 1, sino que están limitados de 0 a 999,99. Hemos pensado que el número es lo suficientemente grande para que no sea un impedimento en su uso.

Para mejorar el rendimiento todo lo posible, cuando se puntúa manualmente algún apartado, siempre se calcula "hacia arriba", es decir, que si puntúas una competencia se calculará la nota total de la competencia y la nota del ejercicio, después la nota de la actividad y por último la nota de las evaluaciones, primero las parciales y luego la final. Además, si se puntúa manualmente un ejercicio no se calculará las competencias, pero si la actividad y las evaluaciones.

Los métodos de cálculo implementados son los siguientes:

- Evaluaciones finales:
 - Por competencias: Media ponderada de las competencias de nivel dos.
 - Por evaluaciones (Sin recuperación): Media ponderada de las evaluaciones parciales.
 - Por evaluaciones (Con recuperación): Si hay recuperaciones, la media ponderada de estas, sino media ponderada de las evaluaciones parciales.
- Evaluaciones parciales:
 - Por todas las actividades: Media ponderada de todas las actividades asociadas a la evaluación.
 - Por recuperaciones: Si hay recuperaciones, la media ponderada de estas, sino media ponderada de todas las actividades asociadas a la evaluación.

Al cambiar el método de cálculo de un grupo de clase, se necesita recalcular las evaluaciones de todos los alumnos de esa clase, por lo que necesariamente es el apartado con menor rendimiento.

Para mejorar la experiencia de usuario en la puntuación se ha optado por añadir una funcionalidad que permita puntuar cada una de las partes en la propia tabla. Esto es un progreso respecto a nuestra primera idea. Esta era que al usuario se le presentase una pantalla para que pudiera introducir la nota. Al tener que hacer click en demasiadas ocasiones esta funcionalidad se volvía un poco engorrosa.

Además, se ha añadido otro par de funcionalidades que mejoran la experiencia de usuario. La primera consiste en poder pasar entre la pantalla de un ejercicio concreto de un alumno al siguiente. Esto permite al profesor puntuar un ejercicio en todos los alumnos sin tener que volver a la página donde se lista los alumnos de un grupo de clase. La otra funcionalidad es pasar por ejemplo, de un ejercicio a otro de un alumno o pasar de una actividad a la siguiente.

III.IX.I Añadir nuevos métodos de cálculo

Los pasos son los siguiente:

- En primer lugar, añadir en *forms.py* el valor del método en base de datos y cómo se va a mostrar al usuario. Se deberá modificar *CHOICES_EVALUATION_TYPE_FINAL_EN* y *CHOICES_EVALUATION_TYPE_FINAL* para las evaluaciones finales y/o *CHOICES_EVALUATION_TYPE_PARTIAL_EN* y *CHOICES_EVALUATION_TYPE_PARTIAL* para las parciales. En la siguiente imagen podemos ver la parte del código que se debe modificar.

```
# Choices
CHOICES_YES_NO = ((False, "No"), (True, "Sí"))
CHOICES_YES_NO_EN = ((False, "No"), (True, "Yes"))
CHOICES_LEVEL = ((1º, "1º"), (2º, "2º"), (3º, "3º"), (4º, "4º"), (5º, "5º"), (6º, "6º"))
CHOICES_GRADE = (
    ("PrimarySchool", "Educación Primaria"),
    ("SecondaryEducation", "Educación Secundaria"),
    ("SixthForm", "Bachillerato"),
    ("FurtherEducation", "Grado Medio o Superior"),
    ("University", "Grado Universitario"))
CHOICES_GRADE_EN = (
    ("PrimarySchool", "Primary School"),
    ("SecondaryEducation", "Secondary Education"),
    ("SixthForm", "Sixth Form"),
    ("FurtherEducation", "Further Education"),
    ("University", "University"))
CHOICES_EVALUATION_TYPE_FINAL_EN = [
    ("BY_COMPETENCES", "By Competences"),
    ("BY_EVALUATION_NO_RECOVERY", "By Evaluations (No recovery)"),
    ("BY_EVALUATION_RECOVERY", "By Evaluations (Recovery)")]
CHOICES_EVALUATION_TYPE_FINAL = [
    ("BY_COMPETENCES", "Por Competencias"),
    ("BY_EVALUATION_NO_RECOVERY", "Por evaluaciones (Sin recuperación)"),
    ("BY_EVALUATION_RECOVERY", "Por evaluaciones (Con recuperación)")]
CHOICES_EVALUATION_TYPE_PARTIAL_EN = ("BY_ALL_ACTIVITIES", "By all Activities"), ("BY_RECOVERY_ACTIVITIES", "By Recovery Activities")
CHOICES_EVALUATION_TYPE_PARTIAL = ("BY_ALL_ACTIVITIES", "Por todas las Actividades"), ("BY_RECOVERY_ACTIVITIES", "Por Recuperaciones")
```

- En segundo lugar, implementar el nuevo método de cálculo en *services.py* en la clase *MarkService()*.
- En último lugar, modificar el método *calculate_evaluation_mark* de la clase *MarkService()* dentro de *services.py* para añadir en el condicional el nuevo método. Podemos verlo en la siguiente imagen.

```
def calculate_evaluation_mark(self, evaluation: models.Evaluation, set_object: models.Set, student: models.Student) -> None:
    if evaluation.is_final:
        if set_object.evaluation_type_final == "BY_COMPETENCES":
            self.calculate_evaluation_mark_by_competences(evaluation = evaluation, set_object = set_object, student = student)
        elif set_object.evaluation_type_final == "BY_EVALUATION_NO_RECOVERY":
            self.calculate_evaluation_mark_by_partial_evaluations(evaluation = evaluation, set_object = set_object, student = student)
        elif set_object.evaluation_type_final == "BY_EVALUATION_RECOVERY":
            self.calculate_evaluation_mark_by_recovery_activities(evaluation = evaluation, set_object = set_object, student = student)
    else:
        if set_object.evaluation_type_partial == "BY_ALL_ACTIVITIES":
            self.calculate_evaluation_mark_by_all_activities(evaluation = evaluation, set_object = set_object, student = student)
        elif set_object.evaluation_type_partial == "BY_RECOVERY_ACTIVITIES":
            self.calculate_evaluation_mark_by_recovery_activities(evaluation = evaluation, set_object = set_object, student = student)
```

III.X Adaptabilidad del diseño en diferentes resoluciones

La resolución nativa de la aplicación es *1920x1080*, dado que es la resolución en la que se ha desarrollado la aplicación. La visualización de la aplicación en resoluciones de escritorio son correctas, pero las resoluciones de dispositivos móviles han sido las más complicadas de realizar. El único dispositivo móvil en el que hemos trabajado en su adaptabilidad, es decir en su [diseño responsive](#), es el **iPhone X** porque es de los dispositivos más grandes, por lo tanto, el más fácil de implementar.

El tratamiento de los formularios en pantallas más pequeñas no ha sido complicado dado que [Materialize](#) nos daba las herramientas necesarias para hacerlo. El problema ha recaído en las listas, dado que hay mucha información repartida horizontalmente, por lo que hemos tenido que omitir información. En casi todas las listas se ha agrupado los botones con un cuadro emergente, es decir, un [modal](#), para aprovechar la mayor cantidad de espacio posible. Esto ha permitido que no se pierda funcionalidad.

En el resto de resoluciones móviles tampoco se pierde funcionalidad, pero cuanto menor es la pantalla peor se va a ver.

III.XI Traducción

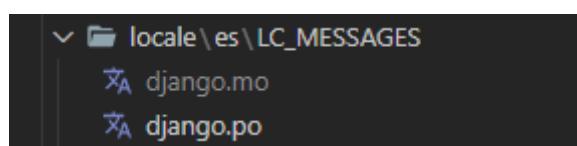
Para la traducción hemos utilizado [gettext](#) y la etiqueta de Django *trans*, los cuales nos permite traducir elementos concretos de las vistas y obtener información del idioma actual de la aplicación.

Para internacionalizar las vistas utilizamos las etiquetas *trans* sobre la palabra o conjunto de palabras que queremos traducir. Podemos ver un ejemplo en el siguiente bloque de código.

Ejemplo de etiqueta de traducción

```
<h6 class="centered">
    {% trans "Set" %}: {{set_object.name}}
</h6>
```

Después debemos tener un archivo de configuración de cada idioma al que queramos traducir. En la siguiente imagen se muestra como se debe ordenar los archivos.



Para crear dicho archivo y compilarlo debemos utilizar los siguientes comandos:

Comando de creación de archivo de configuración

```
django-admin makemessages -l es
```

Comando de compilación

```
py manage.py compilemessages
```

cambiando es (español) por el idioma correspondiente.

Sin embargo, para traducir los errores de formulario no es posible utilizar estas etiquetas, porque hemos implementado un método, que podemos ver en la bloque de código, valiéndose del método de **gettext** llamado *get_language*, que hace saltar los errores de formulario en el idioma correcto.

Método de error de validación

```
def raise_error(self, en_message: str, es_message: str):  
  
    if get_language() == 'en':  
        raise ValidationError(  
            en_message)  
    else:  
        raise ValidationError(  
            es_message)
```

IV. Pruebas

La realización de pruebas es una actividad fundamental en el desarrollo de un proyecto web, ya que permite dar a los usuarios y/o clientes una aplicación que ha demostrado su fiabilidad y calidad en el proceso y la finalización del desarrollo. Como el proyecto no era lo suficientemente grande y complejo, las pruebas manuales cumplían correctamente su función. Por esto no hemos visto necesario la implementación de pruebas automáticas, las cuales requerirían una mayor inversión de tiempo que las pruebas manuales. Así que hemos preferido dedicarle este tiempo al propio desarrollo.

Para las pruebas realizadas que redactamos a continuación, hemos utilizado los navegadores de Google Chrome y Mozilla en las versiones: Versión 91.0.4472.101 (Build oficial) (64 bits) para Chrome y 89.0 (64-bit) para Mozilla.

En el apartado de [adaptabilidad del diseño](#), todo esto ha sido testeado usando Google Chrome, el cual, ofrece una gran ventaja por su herramienta para desarrolladores, que se activa con la tecla F12. En uno de sus múltiples apartados utilizamos la función de cambio de resolución para poder comprobar bien este apartado. Usamos la opción iPhoneX, ya que posee las medidas más parecidas a un teléfono móvil actual.

IV.I Pruebas de formulario

Un tipo de prueba muy común en el mundo del *software* son las pruebas de los formularios. Este es un apartado crucial de las aplicaciones puesto que envían los datos directamente a la base de datos. Este tiene un peligro, y es que si un usuario introduce un dato incorrecto podría llegar a destruir o corromper la base de datos de la aplicación web.

Con estas pruebas nos hemos asegurado de que este tipo de peligros no sucedan en nuestra aplicación. Para ello hemos realizado las siguientes pruebas manuales:

- Comprobamos que la vista esté correctamente traducida.
- Verificamos la obligatoriedad de los campos enviando el formulario completamente vacío y rellenando uno a uno, comprobando si se muestra el mensaje de obligatoriedad.
- Comprobamos las restricciones de modelo, tanto longitud como rango, y vemos si se muestra el error correspondiente. Esto se hace en ambos idiomas.
- Lo enviamos correctamente y verificamos si se ha creado o modificado y que lleva a la vista correspondiente.
- Comprobamos si el botón de *Volver* lleva a la vista correspondiente.
- Verificamos que no se puede acceder al formulario si no tienes permiso (**GET**).
- Comprobamos si cambiando la *URL* del formulario no se puede crear o modificar sin permiso (**POST**).
- Comprobamos la adaptabilidad del diseño.

IV.II Pruebas de borrado

En este apartado nos hemos asegurado que ningún usuario pueda realizar esta acción sin los permisos correspondientes. También hemos comprobado la función de dichos métodos.

Con estas pruebas nos hemos asegurado que lo nombrado anteriormente es correcto. Para ello hemos realizado las siguientes pruebas manuales:

- Comprobamos que la vista esté correctamente traducida.
- Comprobamos si el botón de *Volver* lleva a la vista correspondiente.
- Lo enviamos correctamente y verificamos si se ha borrado y que lleva a la vista correspondiente.
- Verificamos que no se puede acceder a la opción de borrar si no tienes permiso (**GET**).
- Comprobamos si cambiando la url del formulario no se puede borrar sin permiso (**POST**).
- Comprobamos la adaptabilidad del diseño.

IV.III Pruebas de listado

En el apartado de listados nos hemos asegurado de que los usuarios sin permisos para poder ver los mismos no pudieran hacerlo. También hemos comprobado el formato y vista correcta de los listados de la aplicación.

Para ello hemos realizado las siguientes pruebas manuales:

- Comprobamos que la vista esté correctamente traducida.
- Comprobamos que no se pueda acceder a un listado sin permiso mediante url.
- Comprobamos si el botón de *Volver* lleva a la vista correspondiente.
- Comprobamos la adaptabilidad del diseño.

V. Conclusiones y desarrollos futuros

Finalmente, tras más de 600 horas de trabajo y casi nueve meses de dedicación al proyecto, podemos decir con total sinceridad que la aplicación desarrollada cumple los objetivos propuestos.

Unas conclusiones más específicas son las siguientes:

- Django nos ha permitido un desarrollo rápido de los casos de uso.
- Asciidoc nos ha permitido interactuar con la documentación como si fuera cualquier archivo de código.
- El análisis de código mediante SonarCloud ha concluido en un código más limpio.
- Realizar parte de la documentación antes del desarrollo nos ha aclarado el problema a resolver.
- Tener disponible la planificación detallada desde el primer día de desarrollo no ha permitido minimizar las desviaciones del plan.
- Las reuniones han sido clave en el cumplimiento de los plazos.
- Hemos aumentado nuestra experiencia en el despliegue en la nube.
- Hemos aprendido a ser autodidacta en la búsqueda de soluciones a problemas y fallos en el proyecto.
- Hemos aprendido la problemática de la implementación del sistema de competencias.
- Hemos aprendido sobre la educación desde el punto de vista del profesor.
- Hemos aprendido a tener cuidado con los derechos de autor de fuentes e imágenes.
- Hemos aprendido a internacionalizar aplicaciones web en Python.

V.I Mejoras

A la finalización de este Trabajo de Fin de Estudios, la aplicación puede considerarse funcional pero, debido a su naturaleza y a la gran cantidad de escenarios a los que puede aplicarse, cabe pensar en extensiones y mejoras de diversos tipos que podrían implementarse para extender las capacidades de la aplicación o bien para facilitar aún más su uso. A continuación listamos algunas de estas posibles mejoras.

Algunas mejoras relevantes, ya que afectan al modelo de datos, serían las siguientes:

- Particularizar peso de competencias específicas (nivel 2) sobre cada competencia clave a la que afecta (nivel 3).
- Asignar más competencias claves (nivel 3) como padre de una competencia específica (nivel 2).
- Implementar alguna manera de añadir de forma masiva datos en la aplicación.

Otras posibles mejoras menos relevantes son:

- Filtros y/o ordenaciones sobre tablas.
- Añadir buscadores en los campos de los formularios dedicados a relaciones.
- Paginar las tablas cuando hay más de una en una misma pantalla.

Glosario

- **Actividad:** Tarea que crea un profesor para evaluar una o varias competencias de un alumno. Pueden ser pruebas, exámenes, ejercicios de clases, salidas extraescolares, trabajos grupales, etc.
- **Competencia concreta:** Competencia de nivel 1. Competencia muy delimitada, asociada a una asignatura, que se puede puntuar directamente.
- **Competencia específica:** Competencia de nivel 2. Competencia estrechamente relacionada con una asignatura. Se calcula a partir de las competencias concretas.
- **Competencia general:** Competencia de nivel 3. Competencia de alto nivel que engloba varias asignaturas y habilidades.
- **Diseño responsive:** Consiste en configurar los elementos de la vista de la web para adaptarlos para diferentes resoluciones.
- **Ejercicio:** Parte de la que se componen las actividades. Por ejemplo, en un examen de matemáticas, puede ser que un ejercicio sea realizar una ecuación.
- **Grupo:** Conjunto de alumnos asignados a una asignatura de un nivel específico. Por ejemplo, una clase de 3ºB de matemáticas.
- **Modal:** Es un cuadro emergente que se muestra por encima de la pantalla.
- **Framework:** Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática.

Casos prácticos de competencias

En la siguiente tabla detallaremos los nombres que tienen los tres niveles de competencias en los diferentes niveles de enseñanza, ya que la nomenclatura difiere.

Tabla 26. Casos prácticos de competencias

Niveles	Nombre en la aplicación	Enseñanza primaria/ESO/Bachillerato	Títulos universitarios
Nivel 1	Competencias concretas	Estándares de aprendizaje	Criterios del profesor
Nivel 2	Competencias específicas	Criterios de evaluación	Competencias generales y específicas
Nivel 3	Competencias generales	Competencias clave	Competencias básicas

Bibliografía

- [1] Abellán Encarna. Scrum: qué es y cómo funciona esta metodología. Wearemarketing. Recuperada el 28 de junio de 2021, de <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>
- [2] Fernández Carmen. ¿Qué ha aprendido la educación con la Covid-19? La Vanguardia. Recuperada el 28 de junio de 2021 <https://www.lavanguardia.com/vida/formacion/20210121/6185948/que-aprendido-educacion-covid19-brl.html>
- [3] Orden ECD/65/2015, de 21 de enero, por la que se describen las relaciones entre las competencias, los contenidos y los criterios de evaluación de la educación primaria, la educación secundaria obligatoria y el bachillerato. Boletín Oficial del Estado, núm. 25, de 29 de enero de 2015, páginas 6986 a 7003. Recuperada de <https://www.boe.es/eli/es/o/2015/01/21/ecd65>
- [4] Organización Mundial de la Salud. Información básica sobre la COVID-19. Recuperada el 28 de junio de 2021, de <https://www.who.int/es/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19>
- [5] Real Decreto 1105/2014, de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y del Bachillerato. Boletín Oficial del Estado, núm. 3, de 3 de enero de 2015, páginas 169 a 546. Recuperada de <https://www.boe.es/eli/es/rd/2014/12/26/1105>
- [6] Real Decreto 1393/2007, de 29 de octubre, por el que se establece la ordenación de las enseñanzas universitarias oficiales. Boletín Oficial del Estado, núm. 260, de 30/10/2007. Recuperada de <https://www.boe.es/eli/es/rd/2007/10/29/1393/con>
- [7] Universidad de Sevilla. Grado en Ingeniería Informática-Ingeniería del Software. Recuperada el 28 de junio de 2021, de <https://www.us.es/estudiar/que-estudiar/oferta-de-grados/grado-en-ingineria-informatica-ingeneria-del-software>

Apéndices

Manual de usuario

En esta sección será dividida en dos, administrador y profesor. Estas partes serán mostradas de una manera general, con el fin de dar a entender cómo usar la aplicación y cada parte de esta.

Administrador

Empezaremos enseñando la pantalla de inicio, en ella podremos ver un menú superior con las secciones disponibles al ser administrador.



En la sección **Administrators** podremos ver el listado de administradores que contiene el sistema.

A screenshot of the 'List of administrator' page. The top navigation bar is identical to the homepage. The main content shows a table with columns for Surname, Name, Initials, Birthdate, Email, Edit, and Eliminar. One row is visible, showing 'Navarro Cruz', 'Raúl', 'RNC', '29 de Junio de 1998', 'raulnavarrocruz@gmail.com', an edit icon, and a delete icon. At the bottom left is a 'BACK' button, and at the bottom right is the text 'Page 1 of 1.' A green 'CREATE ADMINISTRATOR' button is located at the top right of the table area.

Arriba a la derecha podremos ver el botón para crear un administrador, al pulsarlo nos redirigirá hacia el formulario de creación.

Name: Alberto

Surname: Cordón

Birthdate: 26/04/1996

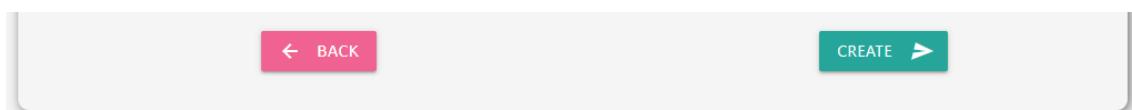
Initials: ACA

Email: alberto@gmail.com

Username: alberto26

Contraseña:|

Una vez rellenado los campos solo debemos darle al botón crear para terminar.



Una vez acabado podremos ver el nuevo administrador creado.

List of administrator

Surname	Name	Initials	Birthdate	Email	Edit	Eliminar
Cordón	Alberto	ACA	26 de Abril de 1996	alberto@gmail.com		
Navarro Cruz	Raúl	RNC	29 de Junio de 1998	raulnavarrocruz@gmail.com		

Page 1 of 1.

BACK

También justo debajo del botón crear usuario podemos ver dos botones más, para editar y eliminar ese administrador.

Todos estos comportamientos (crear, editar y eliminar se repiten la forma de acción en el sistema).

Para estas acciones al igual que la creación, solo hará falta pulsar el botón correspondiente.

Para la acción de editar nos redirigirá al formulario de edición.

VERUM

Inicio Administrators Competences Evaluations Students Teachers Sets Subjects

Edit Administrator

Name	Surname
Raúl	Navarro Cruz
Birthdate	Initials
29/06/1998	RNC
Email	
raulnavarrocruz@gmail.com	

[BACK](#) [EDIT](#) ➤

Y para eliminar tendremos un mensaje informativo sobre el mismo.

VERUM

Inicio Administrators Competences Evaluations Students Teachers Sets Subjects

Delete Administrator

Are you sure about deleting the administrator [Alberto Cordon](#)?

[CANCEL](#) [ELIMINAR](#)

Pasaremos ahora al apartado de **Competences**. Podremos observar que al pulsar en ella aparece un desplegable, para así poder ver las competencias por nivel.

VERUM

Inicio Administrators Competences Evaluations Students Teachers Sets Subjects

List of competences level 3

Code	Name	Description	Level 1	Level 2	Level 3	CREATE COMPETENCE
C69	PRUEBA	PRUEBA				Edit Elimin ➤
CC1	Comunicación lingüística	4º ESO				Edit Elimin ➤
CC1	Comunicación lingüística	3º ESO				Edit Elimin ➤
CC2	Competencia matemática y competencias básicas en ciencia y tecnología	3º ESO				Edit Elimin ➤
CC2	Competencia matemática y competencias básicas en ciencia y tecnología	4º ESO				Edit Elimin ➤

Page 1 of 3, [next](#) [last](#) »

[BACK](#)

Dentro de las competencias de nivel 3 podremos observar un apartado llamado Level 2, el cual nos lleva a las competencias de nivel inferior asociadas a esta.

Code	Name	Description	Edit	Eliminar	Level 1
C69 1.1	PRUEBA 1.1	PRUEBA			

Page 1 of 1.

Una vez en ella también podremos ver la de nivel inferior asociada a esta.

En todas estas secciones podremos crear, editar y eliminar competencias. Al eliminar competencias, si estas tienen otras de nivel inferior asociadas, se eliminarán también.

En la pestaña **Evaluations** podremos ver el listado de las evaluaciones finales, así como editarlas y eliminarlas.

Name	Start date	End date	Subject	Partial evaluations	Edit	Eliminar
Física y Química 3º ESO Final	15 de Septiembre de 2020	19 de Junio de 2021	Física y Química 3º Educación Secundaria			
Física y Química 4º ESO Final	15 de Septiembre de 2020	19 de Junio de 2021	Física y Química 4º Educación Secundaria			
Matemáticas 3º ESO Final	15 de Septiembre de 2020	19 de Junio de 2021	Matemáticas 3º Educación Secundaria			
Matemáticas 4º ESO Final	15 de Septiembre de 2020	19 de Junio de 2021	Matemáticas 4º Educación Secundaria			
Matemáticas Final	2 de Febrero de 2022	2 de Febrero de 2022	Física y Química 3º Educación Secundaria			
Prueba Final	2 de Febrero de 2022	2 de Febrero de 2022	Física y Química 4º Educación Secundaria			
Prueba Final	2 de Febrero de 2022	2 de Febrero de 2022	Matemáticas 3º Educación Secundaria			

Page 1 of 2, [next last »](#)

En esta pestaña aparece un nuevo apartado, Partial evaluations, en el cual podremos ver las evaluaciones parciales que le corresponde.

Pasando a la pestaña **Students** podremos ver el listado de todos los estudiantes, crearlos o eliminarlos.

Pasando a la pestaña **Teachers** podremos ver el listado de todos los estudiantes, así como asignar grupos a un profesor.

VERUM

Inicio Administrators Competences Evaluations Students Teachers Sets Subjects

List of teachers

Surname	Name	Initials	Birthdate	Email	Edit	Eliminar	List of subjects
Campoy	Íñigo	RJX	1 de Enero de 1980	alfaroirene@gmail.com			
Édgar	Jenaro	EAU	1 de Enero de 1980	anastasia23@gmail.com			
Figueroa	Calista	NLJ	1 de Enero de 1980	vilarhernan@hotmail.com			
Naranjo	Ramón	KHN	1 de Enero de 1980	eusebia35@yahoo.com			
Nilda	Jessica	Y2E	1 de Enero de 1980	carmelorebollo@hotmail.com			

Page 1 of 2. [next](#) [last»](#)

[BACK](#)

Podremos asignar o desasignar asignaturas a un profesor, para ello solo debemos pulsar en el botón Assign para asignar o Unassign para desasignar.

VERUM

Inicio Administrators Competences Evaluations Students Teachers Sets Subjects

List of assigned subjects of Íñigo Campoy

Name	Level	Grade	Description	Unassign
Física y Química	4º		Física y Química	

Page 1 of 1.

[BACK](#)

List of not assigned subjects

Name	Level	Grade	Description	Assign
Física y Química	3º		Física y Química	
Matemáticas	3º		Matemáticas	
Matemáticas	4º		Matemáticas	

Pasamos a la sección **Sets** en la que podremos ver el listado de los mismos.

VERUM

Inicio Administrators Competences Evaluations Students Teachers Sets Subjects

List of sets

Name	Level	Grade	Line	Edit	Eliminar	List of students
Física y Química 3ºA ESO	3º	Secondary Education	A			
Física y Química 4ºA ESO	4º	Secondary Education	A			
Matemáticas 3ºA ESO	3º	Secondary Education	A			
Matemáticas 4ºA ESO	4º	Secondary Education	A			

Page 1 of 1.

[BACK](#)

Así como asignar alumnos a un grupo.

The screenshot shows the ERUM software interface. At the top, there is a navigation bar with links: Inicio, Administrators, Competences, Evaluations, Students, Teachers, Sets, Subjects, and a user icon. Below the navigation bar, there are two tables:

List of assigned students

Surname	Name	Birthdate	Unassign
Bolaños	Juanita	1 de Enero de 1980	X
Camilo	Marcos	1 de Enero de 1980	X
Correa	Encarna	1 de Enero de 1980	X
Gallart	Candela	1 de Enero de 1980	X
Iniesta	Lidia	1 de Enero de 1980	X

List of not assigned students

Surname	Name	Birthdate	Assign
Arévalo	Macarena	1 de Enero de 1980	>

At the bottom of the page, there is a footer with the text "Page 1 of 3, next last »".

En el apartado **Subjects** podremos ver el listado de las mismas.

The screenshot shows the ERUM software interface. At the top, there is a navigation bar with links: Inicio, Administrators, Competences, Evaluations, Students, Teachers, Sets, Subjects, and a user icon. Below the navigation bar, there is a table:

List of subjects

Name	Level	Grade	Description	Edit	Eliminar	Competences
Física y Química	4º	Secondary Education	Física y Química			
Física y Química	3º	Secondary Education	Física y Química			
Matemáticas	3º	Secondary Education	Matemáticas			
Matemáticas	4º	Secondary Education	Matemáticas			

At the bottom of the page, there is a footer with the text "Page 1 of 1".

Así como asignar competencias a estas.

VERUM

Inicio Administrators Competences Evaluations Students Teachers Sets Subjects

List of assigned competences

Code	Name	Description	Unassign
CC4	Aprender a aprender	4º ESO	
FQ4 1.1	Reconocer que la investigación en ciencia es una labor colectiva e interdisciplinar en constante evolución e influida por el contexto económico y político.	FQ 4º ESO	
FQ4 1.2	Analizar el proceso que debe seguir una hipótesis desde que se formula hasta que es aprobada por la comunidad científica.	FQ 4º ESO	
FQ4 1.3	Comprobar la necesidad de usar vectores para la definición de determinadas magnitudes.	FQ 4º ESO	
FQ4 1.4	Relacionar las magnitudes fundamentales con las derivadas a través de ecuaciones de magnitudes.	FQ 4º ESO	

Page 1 of 2. [next](#) [last »](#)

[BACK](#)

List of not assigned competences

Code	Name	Description	Assign
------	------	-------------	--------

En el último apartado podremos ver la sección de usuario, algo que se repite tanto en Administrador como Profesor, y en ellas el usuario podrá, ver y editar su datos, cambiar su contraseña o salir de la aplicación.

Welcome administrator

Profesor

La pantalla de inicio es la misma que la de administración, en el menú si podremos ver cambios.

En el apartado **My Sets** en ella podremos ver el listado de grupos que tiene el profesor. En ella podremos ver varias partes

The screenshot shows a table titled 'List of sets' with columns: Name, Level, Grade, Line, Evaluations, Activities, Reports, List of students, and Evaluation type. There are four rows of data:

Name	Level	Grade	Line	Evaluations	Activities	Reports	List of students	Evaluation type
Física y Química 3ºA ESO	3º	Secondary Education	A					
Física y Química 4ºA ESO	4º	Secondary Education	A					
Matemáticas 3ºA ESO	3º	Secondary Education	A					
Matemáticas 4ºA ESO	4º	Secondary Education	A					

Page 1 of 1.

Empezaremos por el apartado **Evaluations**, en esta parte podremos ver las evaluaciones asociadas y la creación de las mismas

The screenshot shows a table titled 'List of evaluations' with columns: Name, Period, Weight, Start date, End date, Edit, and Eliminar. The table has five rows of data:

Name	Period	Weight	Start date	End date	Edit	Eliminar
Física y Química 3º ESO 1er Cuatrimestre	1st	N/A	15 de Septiembre de 2020	22 de Diciembre de 2020	N/A	N/A
Física y Química 3º ESO 2o Cuatrimestre	2nd	N/A	7 de Enero de 2021	26 de Marzo de 2021	N/A	N/A
Física y Química 3º ESO 3er Cuatrimestre	3tr	N/A	5 de Abril de 2021	19 de Junio de 2021	N/A	N/A
Física y Química 3º ESO Final	Final	N/A	15 de Septiembre de 2020	19 de Junio de 2021	N/A	N/A

Page 1 of 1.

En el apartado **Activities** podremos ver la lista de actividades así como ver si es una recuperación, la lista de ejercicios y la opción de crear una actividad, o copiarla.

List of activities

Set: Física y Química 3ºA ESO

Title	Date	Weight	Evaluation	Recovery	List of exercises	Edit	Eliminar
Examen 1	4 de Noviembre de 2020	1,00	Física y Química 3º ESO 1er Cuatrimestre	X			
Copia de Examen 1	4 de Noviembre de 2020	1,00	Física y Química 3º ESO Final	X			
Examen 2	22 de Enero de 2021	1,00	Física y Química 3º ESO 2o Cuatrimestre	X			
Examen 3	26 de Mayo de 2021	1,00	Física y Química 3º ESO 3er Cuatrimestre	X			

Page 1 of 1.

En el apartado **Reports** podremos ver las dos opciones de salida, por competencias y por evaluaciones.

Reports

Name

Name	Type	Value
Física y Química 3ºA ESO	COMPETENCES	
Física y Química 3ºA ESO	EVALUATIONS	
Matemáticas 3ºA ESO	COMPETENCES	
Matemáticas 3ºA ESO	EVALUATIONS	
Matemáticas 4ºA ESO	COMPETENCES	
Matemáticas 4ºA ESO	EVALUATIONS	

Page 1 of 1.

Por competencias:

Report of Competences

Set: Física y Química 3ºA ESO

Student	FQ3 1.1	FQ3 1.1.1	FQ3 1.1.2	FQ3 1.2	FQ3 1.2.1	FQ3 1.3	FQ3 1.3.1	FQ3 1.4	FQ3 1.4.1	FQ3 1.4.2	FQ3 1.5	FQ3 1.5.1	FQ3 1.5.2	FQ3 1.6	FQ3 1.6.1	FQ3 1.6.2	FQ4 1.8	FQ4 1.8.1
Bolíños, Juanita	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Camilo, Marcos	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Correa, Encarna	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Gallart, Candel	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Iniesta, Lidia	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lledó, Araceli	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lumbreras, Toribio	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Por evaluación:

Student	Física y Química 3º ESO 1er Cuatrimestre	Examen 1	Física y Química 3º ESO 2o Cuatrimestre	Examen 2	Física y Química 3º ESO 3er Cuatrimestre	Examen 3	Física y Química 3º ESO Final	Copia de Examen 1	Copia de Examen 1
Bolaños, Juanita	-	-	-	-	-	-	-	-	-
Camilo, Marcos	-	-	-	-	-	-	-	-	-
Correa, Encarna	-	-	-	-	-	-	-	-	-
Gallart, Candela	-	-	-	-	-	-	-	-	-
Iniesta, Lidia	-	-	-	-	-	-	-	-	-
Lledó, Araceli	-	-	-	-	-	-	-	-	-

En el apartado **List of students** podremos ver el listado de estudiantes y la opción de obtener el reporte de las notas de ese alumnos así como ver las competencias y las notas.

Surname	Name	Birthdate	Report	Competences	Marks
Bolaños	Juanita	1 de Enero de 1980			
Camilo	Marcos	1 de Enero de 1980			
Correa	Encarna	1 de Enero de 1980			
Gallart	Candela	1 de Enero de 1980			
Iniesta	Lidia	1 de Enero de 1980			

Page 1 of 3. [next](#) [last](#)

[BACK](#)

Reporte:

 [Inicio](#) [My Sets](#) [My Subjects](#) 

Report of Juanita Bolaños

Set: Física y Química 3ºA ESO

Evaluations	
Name	Mark
Física y Química 3º ESO 1er Cuatrimestre	-
Examen 1	-
Física y Química 3º ESO 2º Cuatrimestre	-
Examen 2	-
Física y Química 3º ESO 3er Cuatrimestre	-
Examen 3	-
Física y Química 3º ESO Final	-
Copia de Copia de Examen 1	-
Copia de Examen 1	-

Competencias:

 [Inicio](#) [My Sets](#) [My Subjects](#) 

List of competences of Bolaños Juanita

Set: Física y Química 3ºA ESO

Code	Level	Mark
FQ4 1.8.1	1	-
FQ4 1.8	2	-
CC1	3	-
CC3	3	-
CC4	3	-

Page 1 of 2. [next](#) [last »](#)



Notas, en esta pantalla podremos modificar el apartado **Manual mark**. Dentro del apartado activities podremos ver el listado de actividades, dentro ejercicios y las competencias asignadas a estos:

En el apartado **Evaluation type** podremos seleccionar el tipo de evaluación.

Y por último el apartado **My Subjects** en el que podremos ver el listado de asignaturas y las competencias asociadas.

Instalación

Prerrequisitos

- Instalación de Python3: <https://www.python.org/downloads/>
- Instalación PostgresSQL: <https://www.postgresql.org/download/>

(Opcional) Entorno virtual

Comando de instalación del entorno virtual

```
pip install virtualenvwrapper-win
```

Comando de creación del entorno virtual

```
mkvirtualenv <NombreEntorno>
```

Comando de uso del entorno virtual

```
workon <NombreEntorno>
```

Configuración

- Creamos la base de datos en PostgresSQL.
- Creamos el archivo local_settings.py, lo rellenamos con la copia del contenido de settings.py y configuramos el apartado **DATABASES** para conectarlo a nuestra base de datos.
- Instalamos los requisitos e migramos el modelo.

Comando de instalación de requitos del proyecto

```
pip install -r requirements.txt
```

Comando de creación de las migraciones

```
py manage.py makemigrations
```

Comando para migrar a la base de datos

```
py manage.py migrate
```

Comandos usuales

Comando para poblar la base de datos

```
py manage.py runscript populate
```

Comando para ejecutar el servidor

```
py manage.py runserver
```