



**UNIVERSIDAD DE CASTILLA-LA MANCHA
FACULTAD DE CIENCIAS SOCIALES DE TALAVERA**

**FORSUIN:
GESTIÓN DE TÍTULOS PROPIOS
1º Parte**

Miriam Fernández Osuna (Miriam.Fdez@alu.uclm.es)

Sergio García Muñoz (sergio.garcia75@alu.uclm.es)

Rubén Gómez Villegas (Ruben.Gomez10@alu.uclm.es)

Carlos Rincón González(carlos.rincon1@alu.uclm.es)



Asignatura: Ingeniería del Software II

Titulación: Grado en Ingeniería Informática

Fecha: 03-11-2022

Ficha del trabajo:

Empresa: forSuin		
Apellidos, Nombre (Rol)	Firma	Sueldo (€)
Fernández Osuna, Miriam (CEO y desarrolladora)		
García Muñoz, Sergio (Desarrollador)		
Gómez Villegas, Rubén (Desarrollador)		
Rincón González, Carlos (Desarrollador)		
Remuneración total de la empresa (€):		1.400.000

CEO de la empresa: Miriam Fernández Osuna

Índice

1. Introducción:	4
2. Organización del proyecto y sprints:	4
2.1 Sprint 01:	5
2.2 Sprint 02:	6
2.3 Sprint 03:	8
3. Diagrama UML:	11
4. GitHub y ramificación:	12
5. Interfaz:	12
6. Base de datos:	15
6.1 Modelo Relacional:	15

1. Introducción:

ForSuin es una empresa de desarrollo digital para clientes externos.

En este caso se nos ha pedido que desarrollemos una plataforma para la gestión de cursos de enseñanzas propias. Estos cursos tienen como objetivo extender los conocimientos de los universitarios más allá de las materias impartidas en su grado correspondiente.

Se han establecido una serie de requisitos, los cuales nos proporcionan un camino que tomar, con el objetivo final de una aplicación con toda la funcionalidad requerida por nuestro cliente.

Se requiere que la aplicación diferencie entre cursos que requieren de titulación universitaria previa y los que no lo hacen, cada uno de ellos con unos objetivos y unos cursos distintos de los otros. Todo ello implementado a través de interfaces gráficas, que faciliten el uso de la plataforma a todo tipo de usuarios, además todos los datos serán guardados en una base de datos, para facilitar el acceso a la información, la cual tiene que estar almacenada.

2. Organización del proyecto y sprints:

A la hora de trabajar en el proyecto hemos empleado una metodología ágil y para ello nos hemos servido de la herramienta GitHub Projects para manejar las tareas y entregables del proyecto. Además hemos usado Maven para el control de dependencias y para hacer el desarrollo más eficiente.

Al inicio del proyecto se han expuesto los requisitos a ser entregados en la versión final del proyecto, estos han sido expuestos en “Product Backlog” de forma que haya una visión clara del trabajo que queda por hacer. En las reuniones de los sprints, estos requisitos han sido divididos en tareas menores, las cuales son más sencillas de resolver, facilitando el progreso de la entrega haciendo que el proyecto fluya dando productos periódicamente.

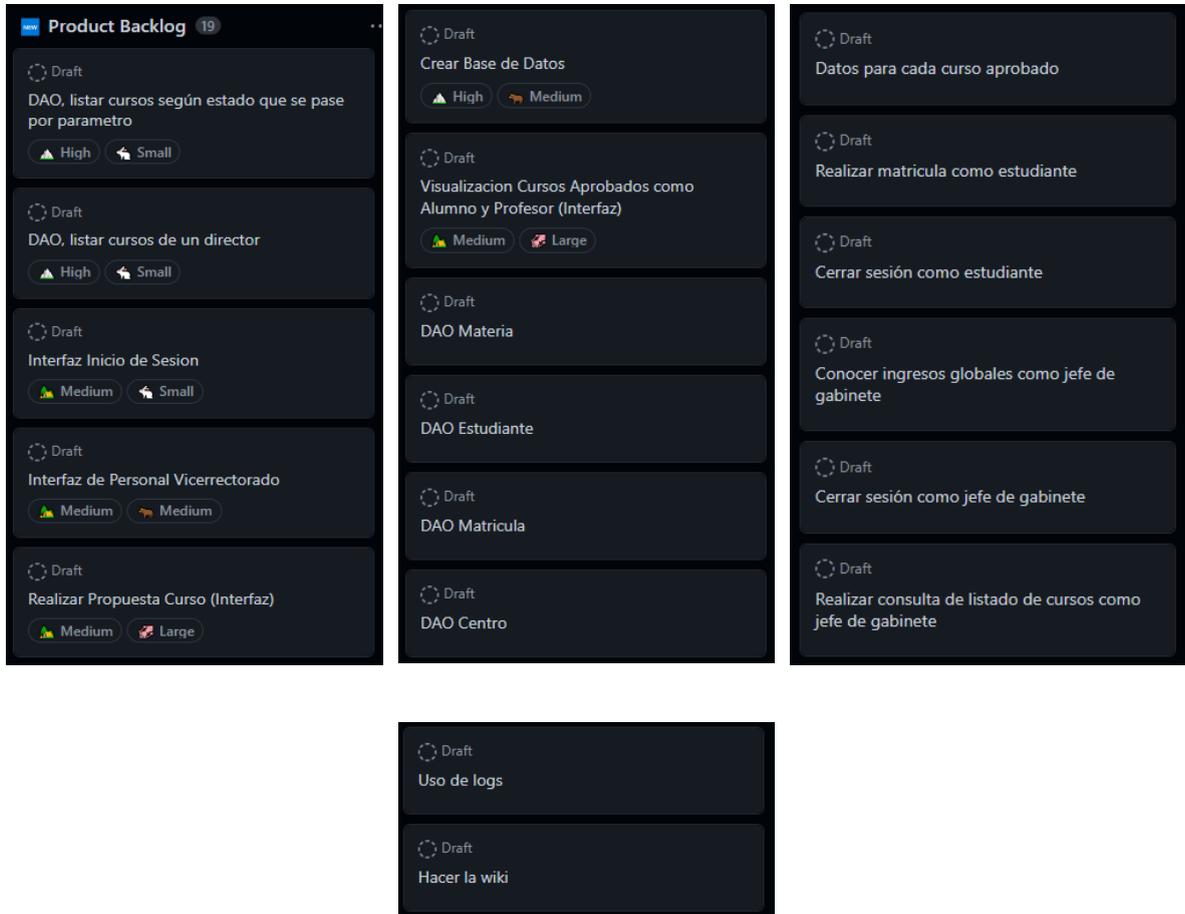
Estas divisiones han sido colocadas en “Sprint Backlog”, de este sitio los trabajadores han ido tomando sus tareas, de forma que nadie se ve forzado a hacer un trabajo del que no se siente calificado, pero todo el trabajo es realizado al final del sprint.

Los desarrolladores han capturado las tareas asignándoles a sí mismos (o a varias personas, en el caso de hacerlo cooperativamente) y llevándolos a la pila de “In progress” en el momento que empiezan a trabajar en ellos. Cuando una tarea es terminada pasa a la pila de “In review” a la espera de que el resto del grupo lo revise, con el objetivo de que el proyecto funcione correctamente como un conjunto y siga unas líneas comunes. Finalmente si pasa este último paso se coloca en la pila de trabajo “Done”, la cual va creciendo desde el principio del proyecto hasta su finalización, manteniendo un historial de los procesos que se han ido siguiendo y asegurando que se ha quedado ninguna funcionalidad por implementar.

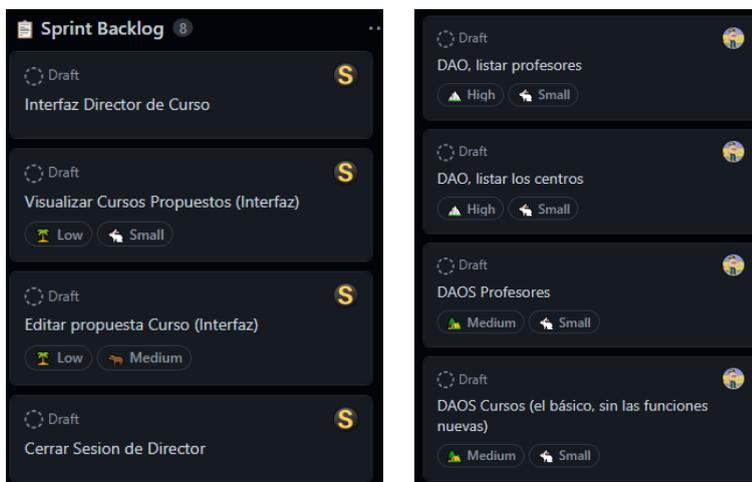
2.1 Sprint 01:

Del 09 al 16 de Octubre:

- Product Backlog:



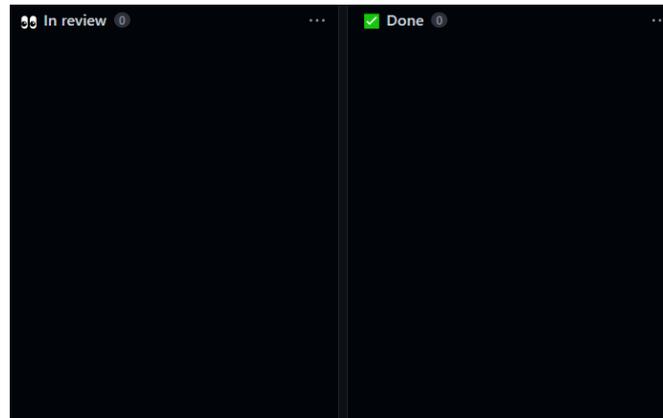
- Sprint Backlog:



- In progress:



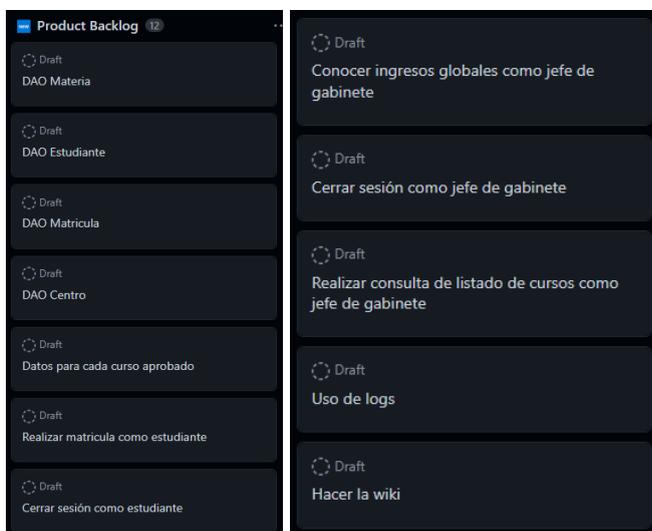
- Review + done:



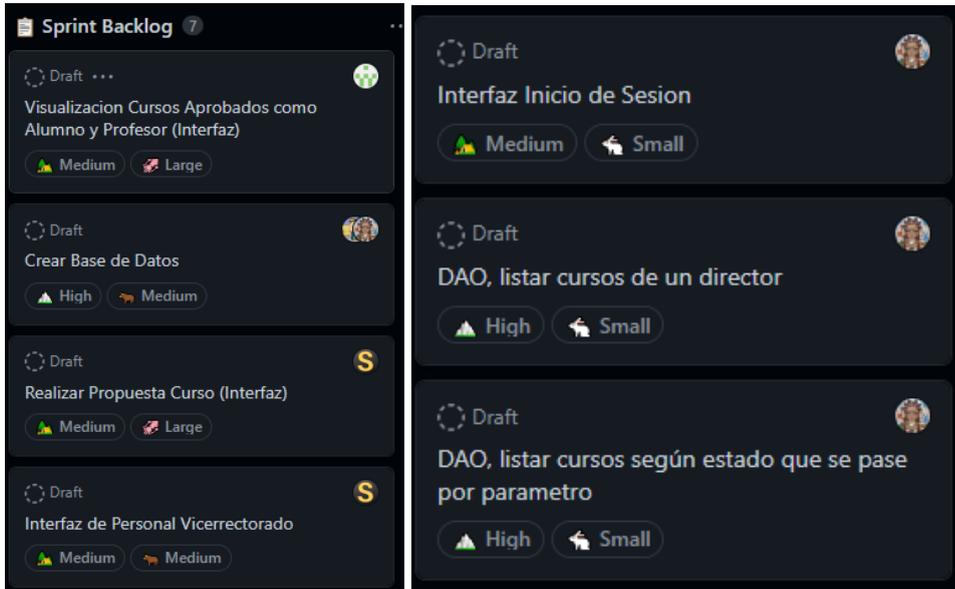
2.2 Sprint 02:

Del 16 al 26 de Octubre:

- Product Backlog:



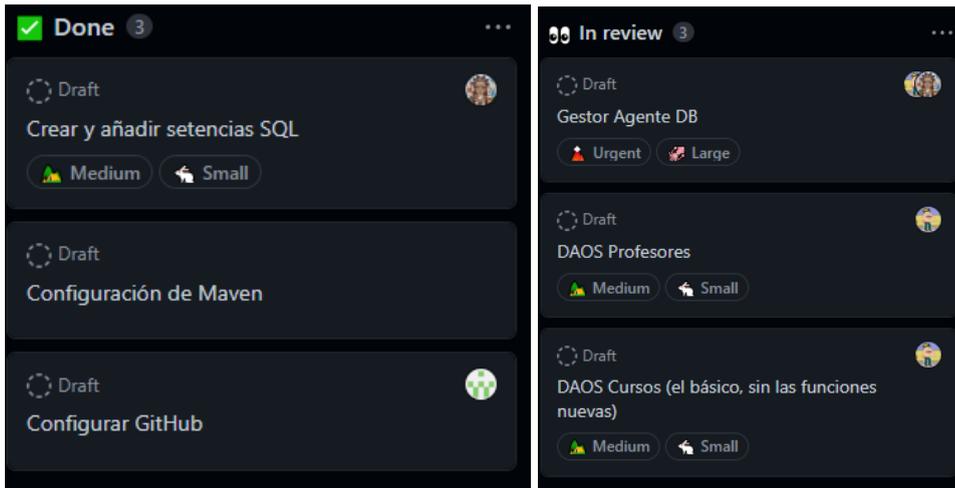
- Sprint Backlog:



- In progress:



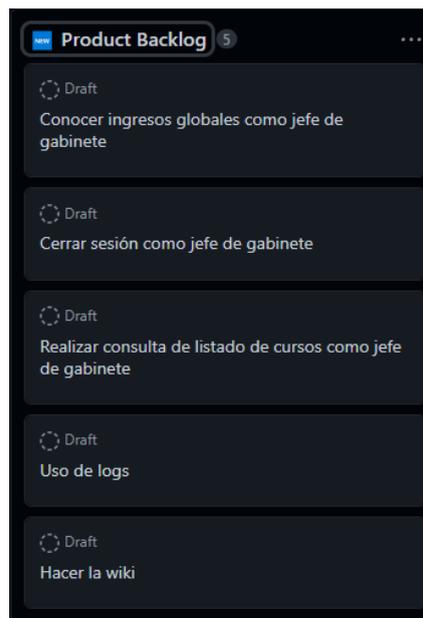
- Review + done:



2.3 Sprint 03:

Del 26 de octubre al 02 de Noviembre:

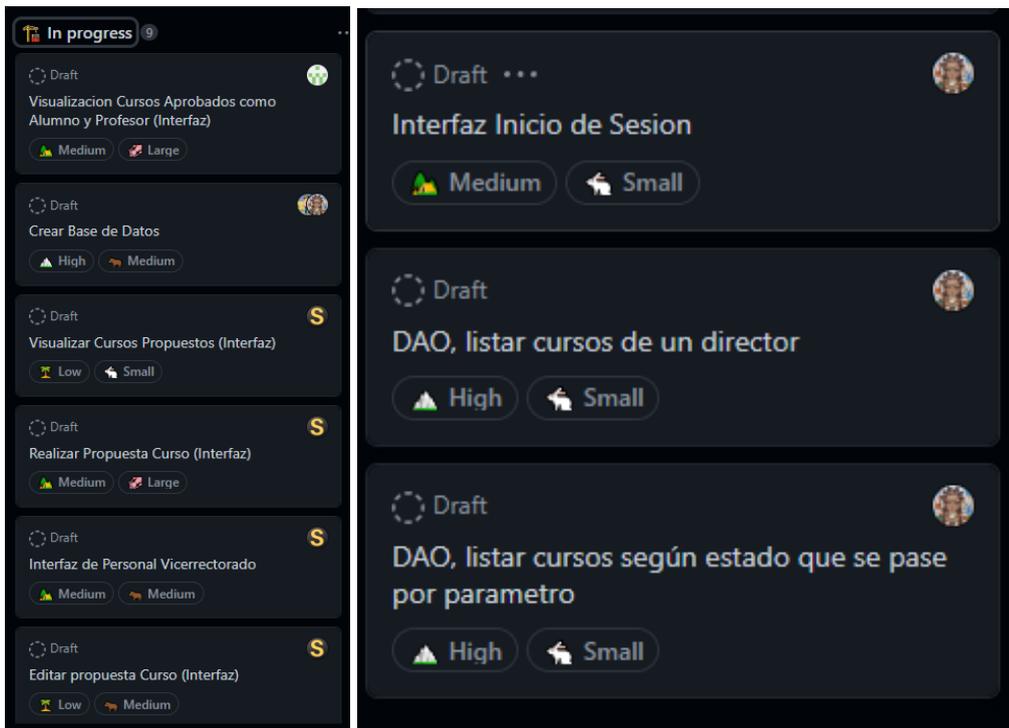
- Product Backlog:



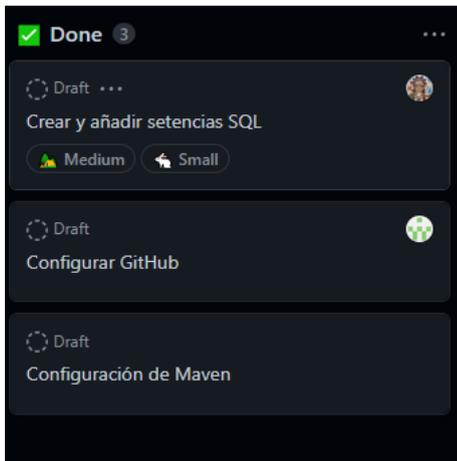
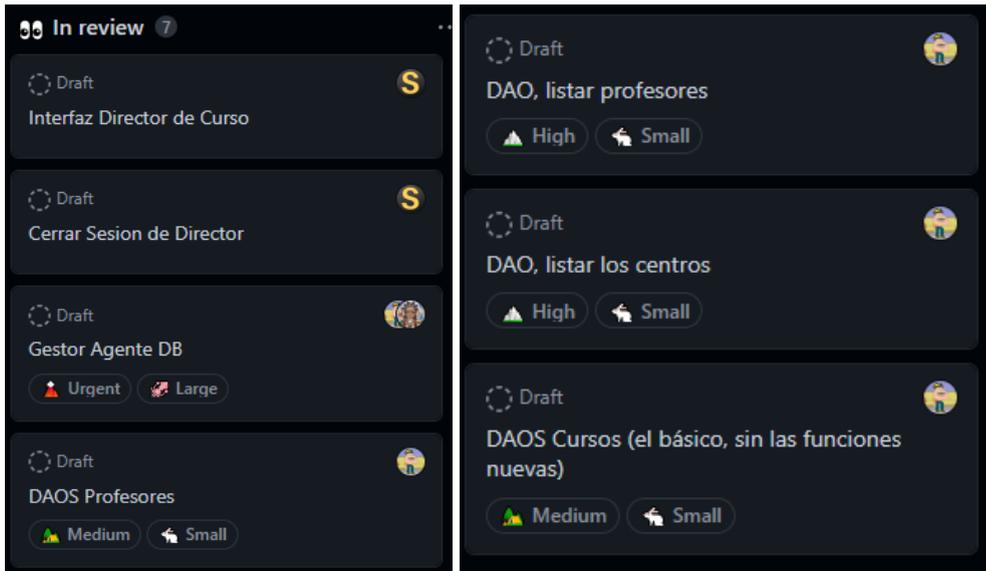
- Sprint Backlog:



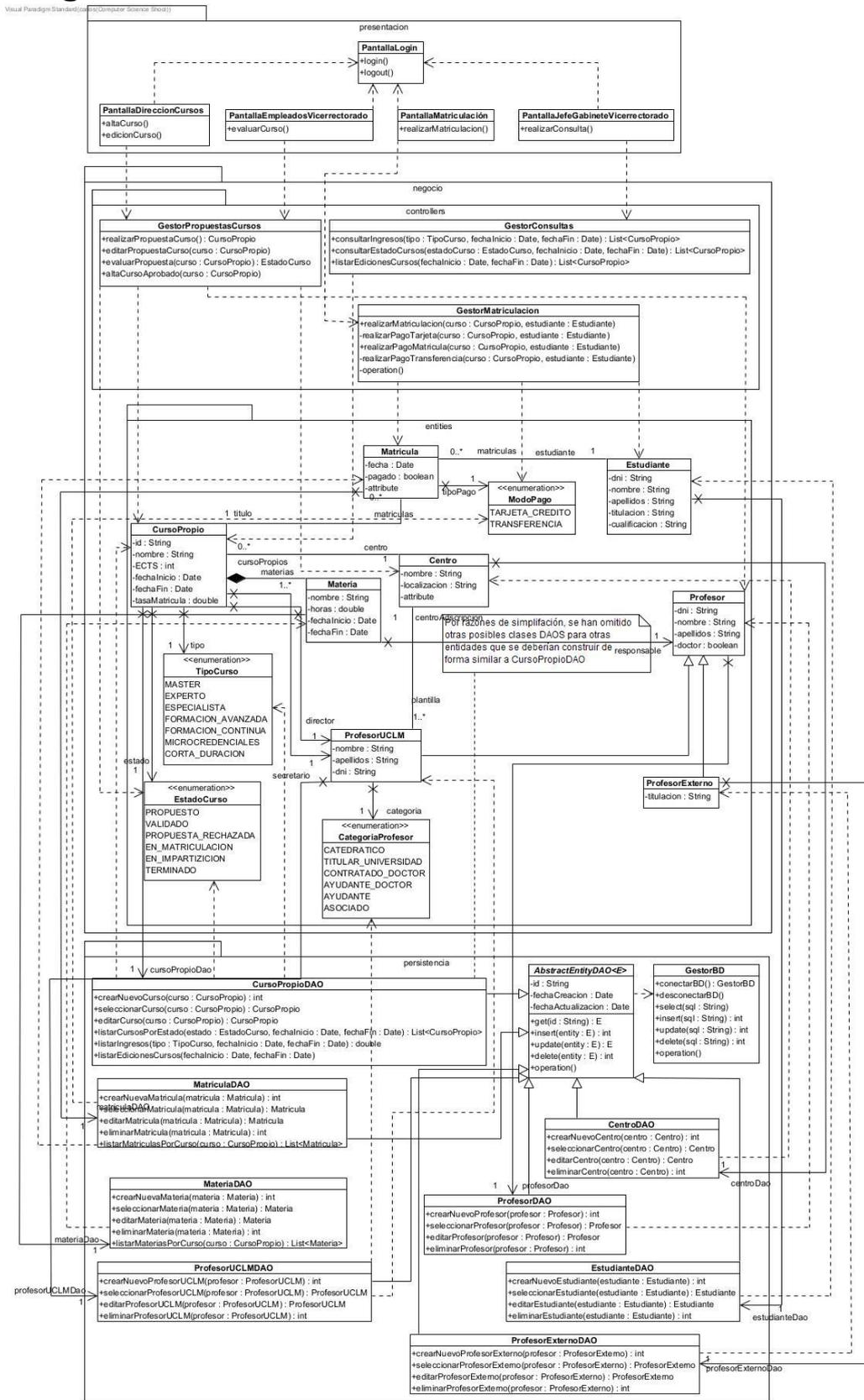
- In progress:



- Review + done:



3. Diagrama UML:



4. GitHub y ramificación:

Como herramienta de control de versiones y de flujo del desarrollo para este proyecto se ha utilizado GitHub. Ha sido adoptado desde el inicio del proyecto, estableciendo unas ramas claves para el progreso exitoso del trabajo.

Se ha establecido una rama principal, en la cual se lanzan grandes funcionalidades cuando están listas, la rama “Master”

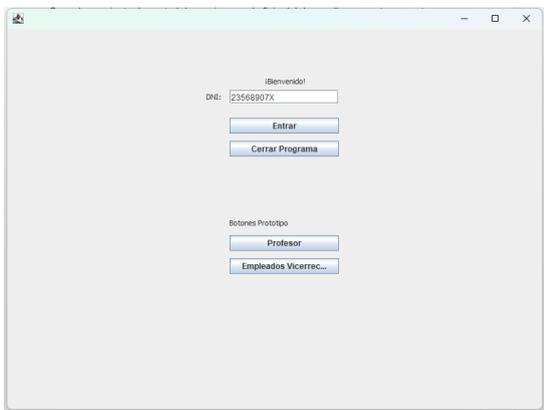
De la rama Master se ha generado una rama, en la cual se inicia el desarrollo de las nuevas funcionalidades, llamada rama de “Desarrollo”, de esta rama se generan las ramas de trabajo para realizar las nuevas funcionalidades sin interrumpirse entre ellas.

Estas ramas secundarias que luego hacen merge en Desarrollo son, en nuestro caso, las ramas “Bases de Datos” e “Interfaces”, ambas ramas son auto explicativas con nombres claros que representan claramente el objetivo que quieren obtener, funcionalidades claras que posteriormente se fusionan en la rama de desarrollo para comprobar su funcionamiento conjunto.

Por último, en nuestro proyecto hay una última rama. Esta rama no es una rama de trabajo persé, sino una rama en la cual se encuentra el estado inicial del proyecto. Esta rama sirve como copia de seguridad para el fatídico caso en el que si se produjese alguna pérdida irreparable en nuestros datos, para tener algo en lo que volcarnos.

5. Interfaz:

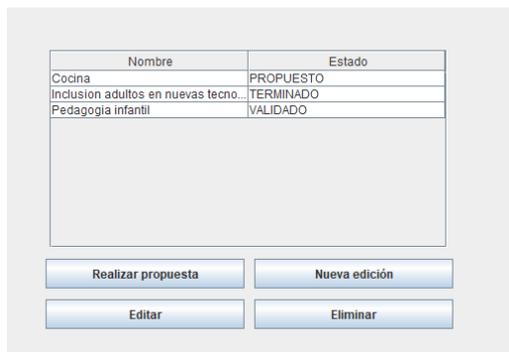
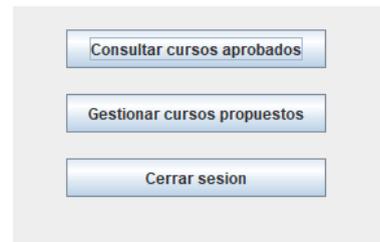
Para nuestro proyecto hemos utilizado la biblioteca de JFrame para ayudarnos a la realización de unas interfaces gráficas más simples, pero a la vez con toda la información y toda la funcionalidad necesaria. Las interfaces creadas para esta plataforma no tienen el objetivo de ser visualmente atractivas sino funcionales, el apartado artístico y visual es algo en lo que se puede trabajar posteriormente a través de otras herramientas, pero por el momento, con el objetivo de generar una funcionalidad y dar un servicio, se ha optado por esto.



La primera interfaz con la que interactuamos es la de “login” ya que es importante hacer una diferenciación entre los usuarios que acceden para registrarse en cursos de los profesores, que lo hacen para proponer nuevos cursos y los empleados del vicerrectorado que tienen el objetivo de aprobar o denegar los cursos propuestos .

(Es importante tener en cuenta que esta ventana en su estado actual no está completa y se encuentra en desarrollo activo.)

Al acceder como profesor al sistema se nos plantean tres opciones, la de consultar los cursos aprobados, con el objetivo de continuar con el proceso de impartirlos, gestionar los cursos propuestos, lo cual incluye hacer la solicitud de un nuevo curso y por último, por supuesto, la opción de salir del sistema.



Si pulsamos sobre la opción de gestión de los cursos propuestos se nos muestra un listado de los cursos con el estado en el que se encuentran actualmente. Pulsando en los cursos podemos editar la solicitud, generar una nueva edición del mismo curso e incluso eliminar la solicitud.

Desde esta pestaña también procedemos al proceso de la nueva solicitud de un curso.

Tanto si editamos un curso ya existente como si intentamos crear uno nuevo, se nos presentará una pestaña con toda la información que es necesaria para la realización de la solicitud. Desde cosas tan básicas como el nombre y la fecha en la que se impartirá el curso, hasta cosas tan específicas como puede ser el tipo de curso a impartir, con los correspondientes requisitos formales que esto implica en créditos ECTS. Además se pueden asignar asignaturas y profesores que las imparten en el curso.

Título de curso

Edición de curso: 1

Profesor secretario

Nombre	Categoría	Doctor
Jose	TITULAR_UNIVERSID...	true
Antonio	ASOCIADO	false
Marcos	AYUDANTE_DOCTOR	false
Miguel	CONTRATADO_DOC...	true
Laura	CATEDRATICO	true

Tasa Matricula

Fecha inicio (DD/MM/AAAA) Fecha final (DD/MM/AAAA)

1 1 2022 1 1 2022

Centro en el que se imparte

- UCLM CUE
- UCLM TAL
- UCLM ALM
- UCLM CIU
- UCLM ALB
- UCLM GUA
- UCLM TOL

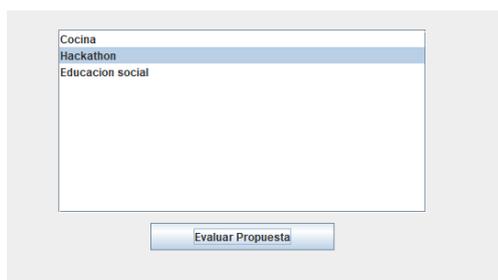
Categoría

- Másteres de Formación Permanente
- Especialistas
- Expertos
- Cursos Universitarios de Formación Avanzada
- Cursos de Formación Continua
- Microcredenciales
- Actividades formativas de corta duración
- Cursos de Verano y Extensión Universitaria
- Formación de Mayores

Requisito

ECTS

Si en la pantalla de login nos hemos registrado como personal del vicerrectorado, nos encontraremos con un listado de los cursos que se encuentran actualmente a la espera de ser revisados.



La idea es que al pulsar sobre el botón de evaluar la propuesta, se nos muestre toda la información que hemos visto que se tiene que incluir en la solicitud.

Por desgracia esta funcionalidad aún no ha sido implementada y no se puede mostrar.

6. Base de datos:

6.1 Modelo Relacional:

Puesto que el objetivo de éste proyecto no es realizar una base de datos al detalle, a través del modelo relacional:

CursoPropio(*id*, nombre, ECTS, fechaInicio, fechaFin, tasaMatricula, edicion, estadoCurso, tipoCurso, **centro_FK**, **secretario_Prof_UCLM_FK**, **director_Prof_UCLM_FK**)

Matricula(fecha, pagado, atributo, modoPago, **curso_PropioFK**, **estudiante FK**)

Estudiante(*dni*, nombre, apellidos, titulacion, cualificacion)

Materia(*nombre*, horas, fechaInicio, fechaFin, **curso_PropioFK**, **responsable_Prof_FK**)

Centro(*nombre*, localizacion, atributo)

Profesor(*dni*, nombre, apellidos, doctor)

ProfesorUCLM(**profesor_dni_FK**, categoriaProfesor, **centro_FK**)

ProfesorExterno(**profesor_dni_FK** , titulacion)