



MANUAL TÉCNICO
INGENIERÍA DEL SOFTWARE 2022-2023

RAFAEL GARCÍA PÉREZ
ISAAC MANUEL CEJUDO ALFARO
LAURA HUERTES REYES

Índice de contenidos

Contenido

0.	Introducción al manual técnico.....	3
1.	Especificación de Requisitos (P2)	4
1.1.	Introducción.....	4
1.1.1	Propósito	4
1.1.2	Ámbito del sistema	4
1.1.3.	Definiciones, acrónimos y abreviaturas	5
1.1.4.	Visión general del documento	6
1.2.	Descripción general	6
1.2.1.	Perspectiva del producto	6
1.2.2.	Funciones del producto	7
1.2.3	Características de los usuarios	9
1.2.4.	Restricciones	10
1.2.5.	Suposiciones y dependencias	11
1.2.6.	Requisitos futuros	11
1.3.	Requisitos específicos	11
1.3.1	Interfaces externas	22
1.3.2	Funciones	23
1.3.3	Requisitos de rendimiento.....	24
1.3.4	Restricciones de diseño.....	25
1.3.5	Atributos del sistema.....	25
2.	Historias de usuario (P2).....	27
3.	Casos de uso (P2).....	34
3.1	Diagrama de casos de uso	40
4.	Planificación (P2,P3, y P4)	49
5.	Diseño del sistema (P3)	63
5.1.	Diagrama de clases	63
5.3.	Diagramas de secuencia	75
5.3.1.	El coordinador de cursos (o administrador) crea un curso	75

5.3.2. Asignar recursos.....	76
5.3.3. Un alumno se inscribe en curso.	77
6. Validación del sistema (P4 y P3).....	78
6.1 Matrices de trazabilidad	78
6.2.1 Casos de uso vs requisitos funcionales	78
6.2.2 Casos de uso vs clases.....	79
7. Implementación del sistema (P4).....	80
7.1 IDE usado	89
8. Futuras mejoras (P4).....	90
9. Pruebas del sistema (P4)	91
10. Referencias (P4).....	102

0. Introducción al manual técnico.

Esta sección está orientada de cara a explicar la estructura y organización del documento presente, debido a que la organización de este es un tanto peculiar, al mezclar todas las prácticas hechas durante el curso más lo nuevo implementado para esta última.

Estamos ante el manual técnico de nuestra aplicación UCOCE, realizada por nuestro equipo de trabajo bajo el pseudónimo UCO Apps, compuesto por Rafael García Pérez, Isaac Cejudo Alfaro, y Laura Huertes Reyes, tal y como se especifica en la portada. El objetivo del mismo, no deja de ser explicar el sistema desarrollado, desde un principio en donde recogíamos los requisitos impuestos por el usuario hasta la implementación del sistema desarrollado por el equipo, así como las pruebas del sistema, pasando a su vez por un seguimiento del trabajo llevado a cabo, el cual ha ido progresando de manera incremental mediante los entregables solicitados por el profesor, y acompañado por una planificación de dicho desarrollo para el seguimiento, que queda registrado en la plataforma YouTrack así como el seguimiento del desarrollo del software en sí, mediante la plataforma GitHub. Los enlaces tanto al repositorio, como a la página de YouTrack se encontrarán en su correspondiente sección.

Es hora de hablar del software desarrollado frente a lo que tenemos aquí en el documento presente documentado. Hay que destacar, que como hemos mencionado anteriormente, esto es un trabajo incremental, que comenzó con la extracción de los requisitos del sistema, y que siguió con los diagramas de sistema, de clases, de secuencia, etc. Usando las metodologías explicadas en clase por el profesor. Es por esto por lo que, la primera sección de este documento está centrada a la extracción de requisitos, siguiendo el estándar IEE y hablando también de los casos de uso y de la planificación, esto último, girando en torno a YouTrack. Esta primera parte de la que hablo comprendería desde el punto 1 hasta el punto 4, lo cual corresponde directamente al trabajo realizado y entregado en la práctica 2.

En una segunda sección, que comprendería a los puntos 5 y 6, trataríamos el tema de las matrices de validación, las cuales nos han servido para describir y detallar el diseño del sistema y la validación de este.

Por último, en nuestra tercera sección, que comprenderían desde el punto 7 en adelante, se hace continuamente referencia al sistema y al software desarrollado e implantado. Por temas de tiempo y de no disponer de los conocimientos suficientes, el software implantado no cumple con todas las características previamente detalladas en las dos secciones que antes hemos descrito, ya que esto se hizo de cara a una aplicación cuyo tiempo de desarrollo sería mucho mayor al tiempo del que hemos dispuesto finalmente. Ni que decir tiene que, al principio de esta tercera sección, en el mismo punto 7, se detalla con exactitud qué incluye el software desarrollado y qué no incluye, así como la visión del software y futuras mejoras que se implementarían más adelante.

1. Especificación de Requisitos (P2)

1.1. Introducción

Este documento es una Especificación de Requisitos de Software (ERS) para la aplicación solicitada por el cliente, un sistema de gestión de los cursos de extensión disponibles en la universidad de Córdoba (UCO). Esta ERS se estructurará basándonos en las directrices del estándar IEEE “Práctica recomendada para especificaciones de requisitos software” ANSI/IEEE 830, 1998.

1.1.1 Propósito

El presente documento tiene como propósito concretar y especificar las funcionalidades de nuestro sistema para la gestión de usuarios interesados o inscritos en los cursos de extensión facilitados por la Universidad de Córdoba, con el objetivo de agilizar tareas de tratamiento de datos tanto de los cursos como de los ponentes y coordinadores, así como los alumnos cursantes de dichas actividades. Por tanto, este sistema podrá ser usado tanto por alumnos como por coordinadores y cualquier otra persona que quiera obtener información de los cursos de extensión.

1.1.2 Ámbito del sistema

El motor que impulsa el desarrollo de este sistema es la necesidad de gestionar información en lo referente a los cursos de extensión debido a la ingesta cantidad de alumnos que se registran en estos, de esta manera, se facilitará la gestión de los alumnos y de la información, haciendo así que los usuarios puedan en su aplicación acceder a la información del curso rápidamente y los coordinadores podre asignar recursos de manera rápida eficaz y unívoca.

La situación de partida es que la UCO ya dispone de un sistema de usuarios y de administración de cursos, pero los extrapolaremos a nuestro sistema para dividir el trabajo.

Nuestro sistema recibirá el nombre de “UCOCE”, que es acrónimo de “Universidad de Córdoba Cursos de Extensión”.

Nuestro sistema servirá como hemos objetado anteriormente, para gestionar los cursos de extensión disponibles en la Universidad de Córdoba.

Los límites de nuestro sistema no van más allá de la gestión y administración del curso, no habrá chat con los alumnos, ya que, para ello, ya existe una plataforma de aula virtual, tampoco se gestionarán pagos de los cursos ni nada externo a lo ya nombrado.

1.1.3. Definiciones, acrónimos y abreviaturas

1.1.3.1 Definiciones

Curso de extensión	Cursos que facilita la universidad a los alumnos para ampliar su formación académica
Alumno	Usuario registrado con el fin de participar en algún curso
Visitante	Usuario no registrado, que dispone de la aplicación, pero no de usuario en el sistema.
Coordinador	Usuario del sistema cuya función es organizar el curso y que, en nuestra aplicación, se encargará de asignar los recursos necesarios para el curso.

1.1.3.2 Acrónimos y abreviaturas

ERS	Especificación de Requisitos Software
UCO	Universidad de Córdoba
UCOCE	Universidad de Córdoba Cursos de Extensión
RF	Requisito Funcional
RNF	Requisito No Funcional
RI	Requisito de Información
CU	Caso de Uso
TSU	Técnico Superior Universitario

1.1.3.3 Referencias

- IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE 830, 1998.
- Especificación de Requisitos según el estándar de IEEE 830 830-1998.

1.1.4. Visión general del documento

Este documento consta de tres secciones. En esta primera sección hemos hecho una breve introducción de nuestro proyecto, proporcionando una idea generalizada de lo que vamos a realizar y una visión de nuestro ERS.

En la segunda sección de nuestro ERS se realizará una descripción general del sistema, esto con el fin de describir las funcionalidades principales del mismo, así como datos asociados, usuarios a los que está destinado, restricciones, dependencias etc. Sin entrar en mucho detalle

Por último, en la tercera sección de nuestro ERS, se definirán detalladamente los requisitos del sistema ha de satisfacer.

1.2. Descripción general

En esta sección se presenta una descripción a alto nivel del sistema. Se presentarán principales áreas a las cuales el sistema debe dar soporte, las funciones que el sistema debe realizar, la información utilizada, las restricciones y otros factores que afectan al desarrollo de este.

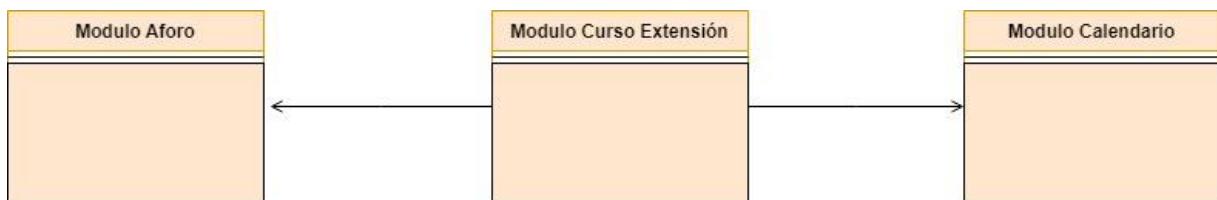
1.2.1. Perspectiva del producto

El sistema UCOCE será un producto diseñado para trabajar en sistemas de aplicación, lo que permitirá una utilización de forma rápida, sencilla, intuitiva y eficaz.

Este sistema no tendrá relación con ningún otro, dentro de sus módulos, se debe de permitir controlar el ingreso de nuevos alumnos al curso específico, el cual está enfocado en este documento, mientras que otros módulos se encargarán de apoyarlo, sin olvidar otros módulos que se encargarán de gestionar otras áreas, como respuestas a solicitudes de sistemas, entre otras funciones.

El módulo Curso de extensión, dependerá del módulo de Aforo para la asignación de una plaza en dicho curso y hará uso del módulo Calendario para la visualización de los cursos disponibles.

1.2.1.1 Diagrama de Componentes



1.2.2. Funciones del producto

En términos generales, nuestro sistema deberá proporcionar soporte para realizar las siguientes tareas:

1. Gestión del administrador
2. Gestión del coordinador del curso
3. Gestión del coordinador de recursos.
4. Gestión de los usuarios y visitantes.
5. Gestión de los cursos.

1.2.2.1 Gestión del administrador

El administrador será quien se va a encargar de gestionarlo todo. Este será el encargado de organizar el estilo de la aplicación, proporcionará también la seguridad necesaria para que la aplicación no sea dañada ni se pierda ningún dato de los usuarios o de los cursos, deberá gestionar también un buen rendimiento y mantenimiento del sistema, pretendiendo que esta tenga los errores mínimos posibles, será el encargado también de las tareas de configuración como, por ejemplo, si un usuario quiere cambiar la contraseña o quiere darse de baja de la aplicación.

1.2.2.2 Gestión de los coordinadores de los cursos

Los coordinadores de los cursos se encargarán de todo lo relacionado con el curso, ya sea imponer la fecha de inicio o fin del curso, establecerán un calendario de este y pondrán un aforo máximo de cada curso correspondiente. También el coordinador de cada curso tendrá acceso a la lista de usuarios inscritos en su curso.

1.2.2.3 Gestión del coordinador de recursos

El coordinador de recursos será el encargado de gestionar todo lo que tenga que ver con el material docente del curso, ya sean archivos, videos o enlaces a los que tendrán acceso los usuarios.

1.2.2.4 Gestión de los cursos

El usuario será toda persona que se registre en la aplicación, tendrá acceso para inscribirse en los cursos y de acceder a los recursos que este les suministra. También podrá contactar con el administrador para darse de baja en la aplicación o por algún error detectado en el sistema.

Y, por otro lado, el visitante es la persona que no se ha registrado, por lo que lo podrá ver los cursos que existen en la aplicación, peor no podrá inscribirse en ellos ni ver el material que proporcionan.

1.2.2.5 Gestión de los usuarios y visitantes

Esto es una visión general de los cursos, donde trabajarán tanto el administrador como los coordinadores, en total un curso deberá proporcionarle al usuario la siguiente información:

- El sistema proporcionará un calendario en el que se mostrarán los períodos de inscripción y de duración de los cursos de extensión
- El sistema generará estadísticas de alcance de los cursos, las cuales podrán ser consultadas por todos los usuarios a excepción de los visitantes.
- Material docente del curso durante toda la duración de este
- Los usuarios no podrán ver las personas que se han inscrito en un curso, aunque este inscrito en él
- El sistema NO es una plataforma educativa, dicho así, no estará hecho con fines para que los alumnos y coordinadores puedan mantenerse en contacto vía chat o similar. Sí se podrá contactar con los alumnos a través del correo que tendrán que facilitar estos al registrarse, por ende, eso es algo de lo que no nos preocuparemos.

1.2.3 Características de los usuarios

Tipo de usuario	Administrador
Formación	TSU en Informática
Actividades	Control y manejo del sistema en general

Tipo de usuario	Coordinador del curso
Formación	Graduado
Actividades	Gestión del curso específico

Tipo de usuario	Coordinador de recursos
Formación	Graduado
Actividades	Gestión de los recursos a utilizar por el curso

Tipo de usuario	Alumno
Formación	Grado Superior o Bachillerato
Actividades	Participación activa en curso

Tipo de usuario	Visitante
Formación	NA
Actividades	Observar los cursos de extensión

1.2.4. Restricciones

- Política de empresa:

Algunas de las políticas que nuestro sistema deberá seguir será facilitar posibles soluciones a las necesidades del sistema, brindar a los usuarios de los cursos que quieran o hacer todo lo posible, proporcionar una organización dentro del sistema, etc..

- Limitaciones del hardware/software:

Esta aplicación exige que el sistema funcione bajo el paradigma usuario/consulta, por lo que el sistema deberá proporcionar las funciones correspondientes para registrar cualquier modificación de los usuarios

- Interfaces con otras aplicaciones:

Para el pago de los cursos, nuestra aplicación no tendrá nada que ver, solo seremos el intermediario entre el usuario y el método de pago escogido.

- Funciones de auditoría y control:

Utilizaremos algunas funciones de mantenimiento y seguridad para comprobar el correcto funcionamiento de nuestro sistema y detectar posibles fallos en él.

- Lenguaje de programación:

Nuestro lenguaje utilizado será JAVA

- Protocolos de comunicación:

Los servidores deben ser capaces de atender consultas concurrentemente.

- Requisitos de habilidad:

El sistema se diseñará según el modelo usuario/curso.

- Criticalidad de la aplicación:

Se proporcionará a los usuarios un formulario de valoración para que los usuarios puedan dejar sus críticas y valoren el desarrollo de la aplicación. Así, poder realizar futuras mejoras en nuestro sistema.

- Consideraciones acerca de la seguridad:

Debemos considerar a este factor como uno de los más importantes de nuestro sistema, ya que es muy importante guardar la privacidad tanto de los usuarios como de los coordinadores y mantener la integridad de todos los datos.

1.2.5. Suposiciones y dependencias

En este punto abordaremos los factores que pudiesen afectar al funcionamiento del sistema, en el caso de que se produjese algún cambio dentro de los requisitos que se hayan obtenido.

Para el funcionamiento del sistema es necesario que el servidor en el cual se está trabajando disponga de una conexión a internet, en caso contrario el sistema no funcionaría.

1.2.6. Requisitos futuros

Como mejora de cara al futuro, no hemos precisado nada con el cliente, puesto que el objetivo es claro y de momento no ha tenido lugar ninguna reunión para tratar dicho tema, por ende, queda cerrada esta sección.

1.3. Requisitos específicos

Identificación del requerimiento	RF01
Nombre del requerimiento	Autentificación de Usuario.
Características	Los usuarios deberán identificarse para acceder a cualquier parte del sistema.
Descripción del requerimiento	El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad suministrando su nombre y contraseña. El sistema ofrecerá la alternativa de entrar sin tener cuenta creada, haciéndolo así bajo el nivel de “visitante”.
Requerimiento NO funcional	<ul style="list-style-type: none">○ RNF01○ RNF02○ RNF05○ RNF08
Prioridad	Alta

Identificación del requerimiento	RF02
Nombre del requerimiento	Registrar Usuarios
Características	Los usuarios deberán registrarse en el sistema para tener acceso a todas las funcionalidades
Descripción del requerimiento	El sistema mediante una pantalla de logueo deberá ofrecer la opción de crear cuenta mediante el suministro de datos tal que nombre de usuario UCO, contraseña y DNI.
Requerimiento NO funcional	<ul style="list-style-type: none"> <input type="radio"/> RNF01 <input type="radio"/> RNF02 <input type="radio"/> RNF05 <input type="radio"/> RNF08
Prioridad	Alta

Identificación del requerimiento	RF03
Nombre del requerimiento	Consultar información del curso de extensión.
Características	Tanto usuarios como visitantes podrán acceder a información de los cursos disponibles
Descripción del requerimiento	El sistema facilitará información general del curso a los usuarios que así lo requieran, mostrando datos como descripción, fechas de inicio y fin, y aforo.
Requerimiento NO funcional	<ul style="list-style-type: none"> <input type="radio"/> RNF01 <input type="radio"/> RNF02
Prioridad	Alta

Identificación del requerimiento	RF04
Nombre del requerimiento	Consultar información del calendario de cursos
Características	Tanto usuarios como visitantes podrán visualizar mediante un calendario las fechas de inicio y fin, la duración y el periodo de inscripción de los cursos existentes.
Descripción del requerimiento	<p>El sistema facilitará información general del curso a los usuarios que así lo requieran, mostrando datos como descripción, fechas de inicio y fin, duración, entre otras.</p> <p>El sistema solo mostrará los cursos disponibles a todos los usuarios.</p>
Requerimiento NO funcional	<ul style="list-style-type: none"> ○ RNF01 ○ RNF02
Prioridad	Alta

Identificación del requerimiento	RF05
Nombre del requerimiento	Estadísticas de alcance
Características	Los usuarios registrados podrán ver estadísticas de alcance de los cursos
Descripción del requerimiento	El sistema generará automáticamente unas estadísticas de alcance para cada curso que podrán ser consultadas por los usuarios. Dichas estadísticas serán generadas en tanto por ciento.
Requerimiento NO funcional	<ul style="list-style-type: none"> ○ RNF01 ○ RNF02
Prioridad	Alta

Identificación del requerimiento	RF06
Nombre del requerimiento	Inscripción al curso
Características	Los usuarios podrán inscribirse al curso deseado.
Descripción del requerimiento	El sistema deberá guardar los datos del usuario para el curso solicitado y actualizar el aforo de dicho curso.
Requerimiento NO funcional	<ul style="list-style-type: none"> <input type="radio"/> RNF01 <input type="radio"/> RNF02 <input type="radio"/> RNF05 <input type="radio"/> RNF06 <input type="radio"/> RNF07 <input type="radio"/> RNF08
Prioridad	Alta

Identificación del requerimiento	RF07
Nombre del requerimiento	Uso de los visitantes
Características	Los usuarios no registrados no podrán más que ver información de los cursos
Descripción del requerimiento	El sistema deberá denegar cualquier tipo de funcionalidad que no sea consultar información de cursos (a excepción de estadísticas de alcance) hasta que no se registren en el sistema
Requerimiento NO funcional	<ul style="list-style-type: none"> <input type="radio"/> RNF01 <input type="radio"/> RNF02
Prioridad	Alta

Identificación del requerimiento	RF08
Nombre del requerimiento	Gestión de cursos
Características	Los coordinadores del curso podrán asignar fechas y activar cursos, así como visualizar usuarios
Descripción del requerimiento	Los usuarios del tipo coordinadores podrán asignar y activar los cursos de extensión que tengan lugar en la UCO, además, también podrá ver los usuarios registrados y consultar información sobre ellos.
Requerimiento NO funcional	<ul style="list-style-type: none"> <input type="radio"/> RNF01 <input type="radio"/> RNF02
Prioridad	Alta

Identificación del requerimiento	RF09
Nombre del requerimiento	Control administrador
Características	El administrador tendrá control total de la aplicación
Descripción del requerimiento	El sistema le proporcionará todos los privilegios al administrador, tales como administrar usuarios, curso, calendario, ver información de los usuarios.
Requerimiento NO funcional	<ul style="list-style-type: none"> <input type="radio"/> RNF02 <input type="radio"/> RNF03
Prioridad	Alta

Identificación del requerimiento		RF10
Nombre del requerimiento		Auditoría del sistema
Características		Garantizar las soluciones de problemas existentes mediante la utilización del sistema.
Descripción del requerimiento		Evaluar y analizar los procesos del sistema, proponiendo solución de problemas existentes dentro del sistema utilizado.
Requerimiento NO funcional		<ul style="list-style-type: none"> <input type="radio"/> RNF03 <input type="radio"/> RNF04 <input type="radio"/> RNF06 <input type="radio"/> RNF07 <input type="radio"/> RNF08
Prioridad		Alta

Identificación del requerimiento		RF11
Nombre del requerimiento		Asignación de los recursos.
Características		Los usuarios del sistema tendrán acceso a diferentes recursos dependiendo del curso al que acceda.
Descripción del requerimiento		El coordinador de recursos será el encargado de asignar los recursos pertinentes al curso, tales como aulas, material escolar, etc.
Requerimiento NO funcional		<ul style="list-style-type: none"> <input type="radio"/> RNF02 <input type="radio"/> RNF03 <input type="radio"/> RNF05 <input type="radio"/> RNF06
Prioridad		Media

Identificación del requerimiento	RF12
Nombre del requerimiento	Contacto con el administrador.
Características	Los usuarios podrán contactar con el administrador en el caso de tener algún problema con respecto al sistema.
Descripción del requerimiento	Se habilitará en nuestra aplicación una sección denominada ‘Contacto’, a través del cual los usuarios podrán contactarse únicamente con el administrador del sistema en caso de tener algún fallo en el sistema o cualquier tipo de duda con respecto a este.
Requerimiento NO funcional	<ul style="list-style-type: none"> <input type="radio"/> RNF01 <input type="radio"/> RNF02 <input type="radio"/> RNF03 <input type="radio"/> RNF05 <input type="radio"/> RNF06 <input type="radio"/> RNF08
Prioridad	Media

Identificación del requerimiento	RF13
Nombre del requerimiento	Valoración de curso
Características	Se facilitará un cuestionario de valoración a los usuarios.
Descripción del requerimiento	De manera opcional, se habilitará en nuestro sistema un cuestionario para que los usuarios dejen su valoración acerca del curso finalizado.
Requerimiento NO funcional	<ul style="list-style-type: none"> <input type="radio"/> RNF01 <input type="radio"/> RNF02 <input type="radio"/> RNF03 <input type="radio"/> RNF05

	<ul style="list-style-type: none"> <input type="radio"/> RNF06 <input type="radio"/> RNF08
Prioridad	Baja

Identificación del requerimiento	RNF01
Nombre del requerimiento	Interfaz del sistema.
Características	El sistema presentará una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios de nuestra aplicación.
Descripción del requerimiento	El sistema debe tener una interfaz de uso intuitiva y sencilla.
Prioridad	Alta.

Identificación del requerimiento	RNF02
Nombre del requerimiento	Ayuda en el uso del sistema.
Características	La interfaz del usuario deberá de presentar un sistema de ayuda
Descripción del requerimiento	La interfaz debe estar complementada con un buen sistema de ayuda para que los mismos usuarios del sistema se les faciliten el trabajo en cuanto al manejo del sistema y el entendimiento de la interfaz.
Prioridad	Alta.

Identificación del requerimiento	RNF03
Nombre del requerimiento	Mantenimiento.
Características	El sistema deberá de tener un manual de instalación y manual de usuario.
Descripción del requerimiento	El sistema debe disponer de una documentación fácilmente actualizable que permita al administrador realizar operaciones de mantenimiento con el menor esfuerzo posible y a los usuarios una pequeña ayuda y guía de uso.
Prioridad	Alta.

Identificación del requerimiento	RNF04
Nombre del requerimiento	Diseño de la interfaz a la característica de la aplicación.
Características	El sistema deberá de tener una interfaz de usuario, teniendo en cuenta las características de nuestra aplicación
Descripción del requerimiento	La interfaz de usuario debe ajustarse a las características de la aplicación, que sea fácil de usar e intuitiva para que los usuarios tengan el mayor nivel de independencia posible.
Prioridad	Media.

Identificación del requerimiento	RNF05
Nombre del requerimiento	Desempeño
Características	El sistema garantizará a los usuarios un gran desempeño en cuanto a los datos almacenados en el sistema ofreciéndole una confiabilidad a esta misma.
Descripción del requerimiento	Garantizar el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta en demasía.
Prioridad	Alta.

Identificación del requerimiento	RNF06
Nombre del requerimiento	Nivel de usuario.
Características	Garantizará al usuario el acceso de información de acuerdo con el nivel que posee.
Descripción del requerimiento	Facilidades y controles para permitir el acceso a la información al personal autorizado a través de internet, con la intención de consultar información pertinente de cada curso de extensión.
Prioridad	Alta.

Identificación del requerimiento		RNF07
Nombre del requerimiento		Confiabilidad continua del sistema.
Características		EL sistema tendrá que estar en funcionamiento las 24 horas los 7 días de la semana.
Descripción del requerimiento		La disponibilidad del sistema debe ser continua con nivel de servicio para los usuarios los 7 días por 24 horas, dotado además de sistemas de avisos y alarmas en caso de fallas y de un plan de contingencias.
Prioridad		Alta.

Identificación del requerimiento		RNF08
Nombre del requerimiento		Seguridad en información.
Características		EL sistema garantizará a los usuarios una seguridad en cuanto a la información que se almacena en el sistema
Descripción del requerimiento		Garantizar la seguridad del sistema con respecto a la información y datos que se manejan ya sean documentos, archivos y contraseñas.
Prioridad		Alta.

Identificación del requerimiento	RI01
Nombre del requerimiento	Datos del curso
Características	El sistema guardará información acerca del curso
Descripción del requerimiento	Nuestro sistema guardará acerca del curso: ID, participantes, ponentes, aforo max, fecha ini_fin, recursos asociados, coordinador.
Prioridad	Alta.

Identificación del requerimiento	RI02
Nombre del requerimiento	Datos de usuarios
Características	El sistema guardará información acerca de los usuarios del sistema
Descripción del requerimiento	Nuestro sistema guardará acerca del usuario la información pedida en el registro: nombre de usuario, correo, y contraseña, lógicamente esta última no será mostrada a quienes consulten información acerca de estos.
Prioridad	Alta.

1.3.1 Interfaces externas

La interfaz con el usuario consistirá en simplemente una aplicación que mostrará varios menús en un desplegable con los que el usuario podrá navegar a través de las distintas funcionalidades que dispone nuestro sistema. Esta interfaz será construida específicamente para la aplicación y será la misma para todos los usuarios.

1.3.2 Funciones

En esta subsección, explicaremos de manera más desarrollada las distintas funcionalidades que tiene nuestro sistema, esta vez, no lo haremos siguiendo una jerarquía funcional, sino que organizaremos esta sección detallando las funcionalidades por tipos de usuario.

1.3.2.1 Visitante

Definiremos visitante como aquel usuario que, sin disponer de una cuenta en el sistema, puede utilizar la aplicación, aunque con funciones limitadas como procederemos a detallar.

El sistema deberá mostrar tanto el calendario de los cursos como los cursos que se encuentran programados.

El sistema deberá mostrar al visitante información relacionada con los cursos de extensión, esta información será una descripción del curso, aforo, y fechas de inicio y fin. En ningún caso el sistema le mostrará el ID del curso y usuarios inscritos (solo visible para administradores).

El sistema no deberá mostrarles a los visitantes las estadísticas de alcance de los cursos, pero sí podrá consultar el calendario facilitado por la aplicación con las fechas de los cursos.

1.3.2.2 Alumno

El alumno será toda aquella persona que se registre en la aplicación web introduciendo algunos de sus datos personales como será un nombre de usuario, DNI, nombre completo, e-mail y número de teléfono entre otros.

Este perfil tendrá acceso a todos los cursos disponibles, pudiéndose inscribir en todos los que desee si y solo si no coinciden las fechas de los cursos en los que se ha inscrito, y si el aforo de ese curso en cuestión no está lleno, aunque de esto se encargará el sistema de indicarlo. Una vez inscrito en un curso, el usuario dispone de la siguiente información: fecha de inicio y fin del curso y material docente correspondiente.

Podrá darse de baja del curso antes de la fecha de inicio o la fecha que indique el coordinador del curso correspondiente o darse de baja contactando con el administrador.

1.3.2.3 Administrador

Se definirá administrador como aquel usuario que dispone del control total de la aplicación para la gestión y manejo de ella, deberá gestionar un buen rendimiento y mantenimiento del sistema, pretendiendo que esta tenga los errores mínimos posibles.

El sistema facilitará información del administrador para contactar con éste en caso de darse algún problema.

El administrador podrá ver todos los cursos que tienen lugar, así como sus usuarios inscritos, recursos asignados, y cualquier información relacionada con el mismo. También deberá de aprobar las solicitudes de los usuarios que soliciten darse de baja de un curso de extensión en el que ya estaban inscritos anteriormente.

1.3.2.4 Coordinadores

Se definirá coordinador como aquel usuario que podrá administrar los cursos que tienen lugar en la Universidad, creándolos y activándolos (coordinador del curso) así como asignándole los distintos recursos que éstos necesiten para ser impartidos (coordinador de recursos).

El sistema deberá de facilitarles a ambos información relativa a los usuarios inscritos en los cursos.

1.3.3 Requisitos de rendimiento

Inicialmente, asignaremos los recursos necesarios para soportar aproximadamente una cantidad cercana a las 1000 consultas de manera simultánea, ya que, de esta forma, nos aseguramos de que, en las épocas de más actividad de cursos y consultas, el sistema no colapse y mantenga un funcionamiento óptimo y constante. En cuanto a transacciones y peticiones se refiere, el servidor estará sobradamente preparado para dichas épocas, puesto que, además, no necesitamos unos recursos muy exigentes dado que las funcionalidades de nuestro sistema son bastante limitadas y no necesita una gran capacidad de cómputo.

Por otro lado, se espera almacenar una cantidad de usuarios próxima a 2/3 de los matriculados en la universidad, proporciones que sobrepasaremos en cierta medida para garantizar la estabilidad del servicio.

1.3.4 Restricciones de diseño.

En esta subsección, no tenemos mucho que detallar, la aplicación a desarrollar no deberá de requerir muchos recursos a nivel de hardware puesto que, en nuestra arquitectura cliente-servidor, es este último el que soportará la carga al recibir las peticiones web enviadas por el cliente, dicho de otra forma, no necesitaremos gran capacidad de cómputo en nuestros clientes, los cuales podrán ser dispositivos móviles con conexión a internet y SO Android 2.2 o superiores y iOS 5.1 o superior. En cuanto a PC se refiere, cualquier SO que sea compatible con un explorador web cualquiera.

1.3.5 Atributos del sistema

En esta sección se detallarán los atributos de calidad de nuestro sistema definidos por seguridad, fiabilidad, disponibilidad, mantenibilidad y portabilidad.

1.3.4.1 Seguridad

El sistema garantizará la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios. En este sentido, la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.

Por otro lado, se garantizará la seguridad del sistema con respecto a la información y datos que se manejan como DNI y contraseñas, entre otras.

1.3.4.2 Fiabilidad

El sistema deberá proporcionar una interfaz de uso intuitiva y trivial. A su vez, la interfaz de usuario deberá ajustarse a las características de la web de la UCO, en la cual estará incorporada el sistema de gestión de cursos.

1.3.4.3 Disponibilidad

La disponibilidad del sistema debe ser continua y máxima, dando servicios a los usuarios de 7 días las 24 horas, garantizando un sistema adecuado que permita contar con un plan de contingencias. Si bien la criticidad del funcionamiento no es máxima, hemos de asegurar en la medida de lo posible el funcionamiento estable de la aplicación como hemos detallado anteriormente.

1.3.4.4 Mantenibilidad

El sistema debe permitir realizar operaciones de mantenimiento con el menor esfuerzo posible a través de habilitar una documentación fácilmente actualizable que lo permita.

La interfaz debe complementarse con un buen sistema de ayuda, tal que el cargo de administración pueda ser realizado por una persona sin experiencia previa de informática.

1.3.4.5 Portabilidad

El sistema será implantado mediante una aplicación web, y aplicación móvil, dicho así, será fácilmente accesible para todo tipo de usuarios como se detalla en el punto [3.4](#) en los requisitos de diseño.

2. Historias de usuario (P2)

Las historias de usuario nos sirven para entender la funcionalidad del sistema desde el punto de vista de los distintos usuarios. Nosotros como equipo, hemos ido creando y determinando las distintas historias de usuario a través de YouTrack por medio de tarjetas. Dichas historias se pueden representar de múltiples formas, pero, en nuestro caso, adjuntaremos capturas de pantalla de la plataforma directamente.

The screenshot shows a YouTrack user story card for 'P208-18'. The card has the following details:

Proyecto	proyecto208
Prioridad	Criticó
Tipo	Historial de usuario
Estado	Hecho
Usuario asignado	Sin asignar
Sprints	Sprint 1
Story points	Sin estimar

The card also includes sections for 'Configuración de actividad' (with options like 'Normal text', 'Visual', 'Markdown', and rich text editor controls) and 'Observadores' (with a count of 1 and a link to 'Dejar de observar').

Ilustración 2

P208-4 Creada por p82garpr Hace 3 semanas
Actualizado por p82ceall ayer

Visible para lectores de incidencias •

★ Registro de usuario

COMO usuario QUIERO registrarme e iniciar sesión PARA mantener la gestión.

El usuario deberá ser capaz de registrarse e iniciar sesión para realizar todas las operaciones que pueda realizar. Como requisitos para su registro, el usuario deberá de determinar un nombre de usuario de la UCO, su DNI y una contraseña, así como nombre y apellidos como descripción general. De dicho nombre de usuario de la UCO se extraerá su correo de la UCO correspondiente.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

P8 p82garpr • comentada 12 Oct 2022 21:29

Si el usuario ya está creado, se notificará, no dejara crear otro usuario, y se ofrecerá un sistema de recuperación de contraseña en caso de olvido

Normal text | B | I | S | A | ... Visual Markdown

Escribir un comentario, @mencionar personas

Añadir comentario Visible para Igual que la incidencia •

Proyecto	proyecto208
Prioridad	Crítico
Tipo	Historial de usuario
Estado	Hecho
Usuario asignado	Sin asignar
Sprints	Sprint 1
Story points	Sin estimar

Observadores 1 > ★ Dejar de observar

Paneles > + Agregar al panel

Ilustración 3

P208-7 Creada por root Hace 3 semanas
Actualizado por p82ceall ayer

Visible para lectores de incidencias •

★ Uso de usuario

COMO usuario QUIERO ver los cursos de extensión disponibles PARA realizar una serie de operaciones como la inscripción a los cursos disponibles.

El usuario deberá ser capaz de ver y navegar en los cursos de extensión disponibles para ver su información, poder la inscripción en ellos, visualizar en qué cursos está inscrito y poder contactar con el administrador cuando surja algún problema sobre la gestión de la aplicación.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

P8 p82garpr • comentada 12 Oct 2022 21:29

Si el usuario ya está creado, se notificará, no dejara crear otro usuario, y se ofrecerá un sistema de recuperación de contraseña en caso de olvido

Normal text | B | I | S | A | ... Visual Markdown

Escribir un comentario, @mencionar personas

Añadir comentario Visible para Igual que la incidencia •

Proyecto	proyecto208
Prioridad	Normal
Tipo	Historial de usuario
Estado	Hecho
Usuario asignado	Sin asignar
Sprints	Sprint 1
Story points	Sin estimar

Observadores 0 > ★ Observar incidencia

Paneles > + Agregar al panel

Ilustración 4

P208-3 Creada por p82garpr Hace 3 semanas
Actualizado por p82ceali ayer

★ Usos de visitante

COMO visitante, QUIERO ver y navegar en los cursos de extensión PARA ver su información.

El visitante deberá ser capaz de ver los cursos de extensión disponibles y navegar en ellos para visualizar su información, pero con la restricción de no poder inscribirse en ellos y así lograr que el sistema solo almacene los datos de todos aquellos que se inscriban para mantener una gestión de los datos de la manera mas optima posible.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad

Normal text | **B** *I* ~~S~~ | A | ... Visual Markdown Aa ?

Escribir un comentario, @mencionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Normal
Tipo	Historial de usuario
Estado	Hecho
Usuario asignado	Sin asignar
Sprints	Sprint 1
Story points	Sin estimar

Observadores 1 > ★ Dejar de observar

Paneles > + Agregar al panel

Ilustración 5

P208-13 Creada por p82garpr Hace 2 días
Actualizado por p82ceali ayer

★ Estadisticas de alcance

COMO usuario QUIERO visualizar las estadísticas de alcance de cada curso de extensión PARA conocer con exactitud el porcentaje de dicho curso

El usuario podrá visualizar las estadísticas de alcance de cada curso de extensión una vez finalizado este.

El sistema deberá de generar unas estadísticas de alcance que serán visualizadas por los usuarios del sistema

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad

Normal text | **B** *I* ~~S~~ | A | ... Visual Markdown Aa ?

Escribir un comentario, @mencionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Normal
Tipo	Tarea
Estado	Hecho
Usuario asignado	Sin asignar
Sprints	Sprint 1
Ideal days	Sin estimar
Original estimation	Sin estimar

Observadores 1 > ★ Dejar de observar

Paneles > + Agregar al panel

Ilustración 6

P208-19 Creada por p82ceali Hace 2 días Visible para lectores de incidencias ▾
Actualizado por p82ceali ayer

★ Auditoria de sistema

COMO Administrador QUIERO evaluar los procesos del sistema PARA proponer soluciones a los problemas que existan.

El administrador evaluara y analizara los procesos del sistema, proponiendo la solución de problemas existentes.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad ▾

Normal text | **B** *I* ~~S~~ | A | ... Visual Markdown Aa ?

Escribir un comentario, @mentionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Normal
Tipo	Historial de usuario
Estado	Para verificar
Usuario asignado	Sin asignar
Sprints	Sprint 2
Story points	Sin estimar

Observadores 1 > ★ Observar incidencia
Paneles > + Agregar al panel

Ilustración 7

P208-17 Creada por p82ceali Hace 2 días Visible para lectores de incidencias ▾
Actualizado por p82ceali ayer

★ Contacto con Administrador

COMO Usuario QUIERO tener contacto con el Administrador PARA resolver cualquier tipo de dudas, problemas...

El usuario contactara con el administrador para la resolución de cualquier tipo de dudas o problemas que le surja a través de un numero de teléfono o correo del mismo.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad ▾

Normal text | **B** *I* ~~S~~ | A | ... Visual Markdown Aa ?

Escribir un comentario, @mentionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Menor
Tipo	Historial de usuario
Estado	Para verificar
Usuario asignado	Sin asignar
Sprints	Sprint 2
Story points	Sin estimar

Observadores 1 > ★ Observar incidencia
Paneles > + Agregar al panel

Ilustración 8

P208-6 Creada por p82garpr Hace 3 semanas Visible para lectores de incidencias ▾ Actualizado por p82ceali ayer

★ Lista matriculados

COMO coordinador del curso, QUIERO ser capaz de ver un listado del curso PARA contactar con los matriculados en caso de necesitarlo.

El coordinador del curso deberá tener acceso a una lista con todos los usuarios inscritos en dicho curso, así como a su información personal para un buen desarrollo del curso.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad ▾

Normal text | **B** *I* ~~S~~ | Aa | ... Visual Markdown

Escribir un comentario, @mentionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Criticó
Tipo	Historial de usuario
Estado	Hecho
Usuario asignado	Sin asignar
Sprints	Sprint 2
Story points	Sin estimar

Observadores 1 > ★ Dejar de observar

Paneles > + Agregar al panel

Ilustración 9

P208-16 Creada por p82garpr Hace 2 días Visible para lectores de incidencias ▾ Actualizado por p82ceali ayer

★ Cuestionario de valoracion

COMO usuario QUIERO que el sistema me ofrezca una encuesta PARA valorar el curso de extensión.

Al finalizar el curso donde el usuario se ha inscrito, el sistema mandará un link a cada usuario del curso para votar en una encuesta y valorar el curso en cuestión de manera opcional. Esta encuesta será informativa para que el coordinador pueda hacer futuras mejoras con respecto al curso.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad ▾

Normal text | **B** *I* ~~S~~ | Aa | ... Visual Markdown

Escribir un comentario, @mentionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Menor
Tipo	Tarea
Estado	Hecho
Usuario asignado	Sin asignar
Sprints	Sprint 2
Ideal days	Sin estimar
Original estimation	Sin estimar

Observadores 1 > ★ Dejar de observar

Paneles > + Agregar al panel

Ilustración 10

P208-12 Creada por p82garpr Hace 2 días Visible para lectores de incidencias ▾ Actualizado por p82ceali ayer

★ Coordinadores del curso

COMO coordinador del curso de extensión, QUIERO realizar modificaciones en los recursos del curso PARA obtener una mejor organización.

El coordinador de recursos será el encargado de asignar los recursos necesarios a los cursos para mantener un inventario de los recursos disponibles, pero el coordinador de cada curso debe poder tanto editar como eliminar, es decir, cambiar archivos de sitio, poner títulos y etc, para poder tener un uso más eficiente de los recursos en cuestión.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

P8 p82garpr • comentada 12 Oct 2022 21:02
– También deberán crear los cursos y definir fechas, se encargará @p82ceali

P8 p82ceali • comentada 12 Oct 2022 21:27
– @p82garpr Para la creación de cada curso se almacenará su ID correspondiente, participantes, ponentes, su aforo máximo, la fecha de inicio y fin, recursos asociados y el coordinador asignado

P8 p82garpr • comentada 12 Oct 2022 21:47
– El coordinador podrá ver un listado con todos los alumnos matriculados

Normal text | B I S | A ... Visual Markdown

Escribir un comentario, @mentionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Normal
Tipo	Historial de usuario
Estado	Hecho
Usuario asignado	Sin asignar
Sprints	Sprint 2
Story points	Sin estimar

Observadores 2 > ★ Dejar de observar

Paneles > + Agregar al panel

Ilustración 11

P208-2 Creada por p82garpr Hace 3 semanas
Actualizado por p82ceali ayer

★ Lista de espera

COMO usuario, QUIERO estar en una lista de espera PARA tener la posibilidad de poder inscribirme en un futuro.

El sistema debe darle al usuario un acceso para que, cuando un curso esté al completo, pueda inscribirse en una lista de espera correspondiente al mismo. Esto servirá para que en caso de que hayan plazas libres, el usuario pueda acceder al curso. Se accederá por ORDEN DE ACCESO a dicha lista de espera, siendo el primero en entrar en el curso , el primero que entró en la lista de espera.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Proyecto	proyecto208
Prioridad	Mayor
Tipo	Historial de usuario
Estado	Hecho
Usuario asignado	Sin asignar
Sprints	Sprint 2
Story points	Sin estimar

Observadores 1 > ★ Dejar de observar
Paneles > + Agregar al panel

Normal text | **B** *I* ~~S~~ | A | ... Visual Markdown

Escribir un comentario, @mentionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Ilustración 12

P208-14 Creada por p82garpr Hace 2 días
Actualizado por p82ceali ayer

★ Calendario con el periodo de inscripción

COMO usuario QUIERO tener acceso al calendario del curso PARA obtener información sobre las fechas de este.

El sistema debe dejar al usuario poder ver en la interfaz un calendario con fecha del comienzo de curso, duración y fecha del final del curso, que será proporcionado por el coordinador del curso. Antes del comienzo del curso, solo se muestra la fecha de inicio y el periodo de inscripción, y una vez el curso empezado, se deben ir actualizando las fechas que correspondan.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Proyecto	proyecto208
Prioridad	Normal
Tipo	Historial de usuario
Estado	Hecho
Usuario asignado	Sin asignar
Sprints	Sprint 2
Story points	Sin estimar

Observadores 1 > ★ Dejar de observar
Paneles > + Agregar al panel

Normal text | **B** *I* ~~S~~ | A | ... Visual Markdown

Escribir un comentario, @mentionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Ilustración 13

3. Casos de uso (P2)

Como hemos visto en clase, los casos de uso no son más que medios que usamos para capturar los requisitos del sistema, pero especificando el comportamiento de este, así como detalles varios que nos ayudarán a entender el contexto y los objetivos, implicando una interacción entre el usuario y el sistema. A continuación, expondremos mediante tablas los distintos casos de uso que hemos extraído para esta práctica, seguido de un diagrama de los casos de uso hecho con UML.

Identificador	CU1
Objetivo	Un nuevo usuario quiere registrarse en la aplicación
Contexto	El usuario debe acceder al sistema sin cuenta previa
Actor principal	Alumno
Escenario principal	<ol style="list-style-type: none">1. El usuario abre la aplicación y se encuentra con la pantalla de bienvenida2. El sistema muestra inicialmente una pantalla de logueo, con una opción de nuevo registro, que deberá elegir el usuario para registrarse.3. El sistema pedirá al nuevo usuario, nombre y apellidos, usuario de la UCO, DNI, y contraseña y lo guardará en su base de datos.4. El usuario ya está registrado y se redireccionará a la pantalla de inicio de sesión para acceder con su cuenta.
Extensiones	<ol style="list-style-type: none">3.1. Si el usuario que se está intentando registrar ya existe, el sistema lo notificará y le redirigirá a la pantalla de inicio de sesión.

Identificador	CU2
Objetivo	Un usuario ya existente quiere iniciar sesión
Contexto	El usuario debe acceder al sistema con cuenta previamente creada
Actor principal	Alumno, Coordinador de recursos, Coordinador de cursos, Administrador
Escenario principal	<ol style="list-style-type: none"> 1. El usuario abre la aplicación y se encuentra con la pantalla de bienvenida 2. El sistema muestra inicialmente una pantalla de inicio de sesión, con una opción de nuevo registro, que esta vez no usaremos 3. El alumno deberá introducir usuario y contraseña.
Extensiones	<ol style="list-style-type: none"> 3.1. Si el usuario introduce la contraseña correctamente, se le dará acceso al sistema. 3.2. Si el usuario introduce la contraseña erróneamente, se le denegará el acceso al sistema y se le ofrecerá la opción de recuperar contraseña 3.3. Si el sistema detecta que el usuario introducido no existe, se le ofrecerá crear una cuenta nueva con dicho usuario.

Identificador	CU3
Objetivo	Un usuario alumno quiere inscribirse en un curso
Contexto	El usuario debe acceder al sistema con cuenta previamente creada
Actor principal	Alumno
Escenario principal	<p>1. El usuario accede al sistema</p> <p>2. El alumno accederá a la pantalla de cursos y elegirá el curso en el que se quiere inscribir</p>
Extensiones	<p>2.1. Si el curso no está completo, se inscribirá y se le notificará de su inscripción.</p> <p>2.2. Si el curso está al completo, se le notificará al usuario y se le preguntará si quiere acceder a la cola de espera, si pulsa sí, pues accede a dicha cola, la cual está ordenada preferentemente por orden de acceso.</p>

Identificador	CU4
Objetivo	Un usuario quiere darse de baja
Contexto	El usuario quiere darse de baja en la aplicación y que su información sea eliminada completamente de ella
Actor principal	Alumno, administrador
Escenario principal	<p>1. El usuario abre la aplicación y se dirigirá a la sección de contacto.</p> <p>2. En dicha sección, podrá contactar con el administrador, proporcionándole sus razones para darse de baja.</p> <p>3. Una vez aceptada la solicitud, el administrador eliminará al usuario y toda su información de la aplicación.</p>
Extensiones	<p>3.1. Si el usuario quiere volver a utilizar la aplicación, deberá volver a registrarse e introducir sus datos personales.</p>

Identificador	CU5
Objetivo	Un usuario tiene alguna duda acerca del sistema.
Contexto	El usuario quiere que se le resuelva alguna duda o cualquier especie de problema acerca del sistema.
Actor principal	Alumno, coordinador de recursos, coordinador de cursos administrador
Escenario principal	<ol style="list-style-type: none"> 1. El usuario se dirige a la sección Contacto. 2. Una vez dentro, se dirigirá al apartado Contacto con el Administrador y podrá visualizar el correo del administrador para la resolución de dudas o problemas. 3. El usuario se dirigirá a su correo personal y le detallará su problema al administrador mediante el correo proporcionado. 4. El administrador le resolverá el problema.
Extensiones	<i>Sin extensiones</i>

Identificador	CU6
Objetivo	Un visitante quiere acceder al sistema sin cuenta
Contexto	El usuario quiere probar la aplicación y ver los cursos de extensión sin tener cuenta.
Actor principal	Visitante
Escenario principal	<ol style="list-style-type: none"> 1. El usuario abre la aplicación y se encuentra con la pantalla de bienvenida 2. El sistema muestra inicialmente una pantalla de inicio de sesión, con una opción de nuevo registro, y una última de acceso como visitante, opción que deberá elegir nuestro usuario. 3. El sistema le permitirá acceso con privilegios de visitante
Extensiones	<i>Sin extensiones</i>

Identificador	CU7
Objetivo	Un coordinador de cursos quiere crear un curso de extensión
Contexto	El usuario debe ser coordinador para tener los privilegios de creación de cursos de extensión, además del administrador
Actor principal	Coordinador de cursos
Escenario principal	<ol style="list-style-type: none"> 1. El usuario abre la aplicación y se encuentra con la pantalla de bienvenida, iniciaría sesión con su usuario coordinador de cursos 2. En cursos de extensión, presionara el botón para crear dicho curso 3. El usuario deberá administrar los datos requeridos por el curso, a excepción del ID, que lo generará automáticamente el sistema. 4. El coordinador podrá ver los participantes y toda la información que tiene el curso.
Extensiones	3.1. Si el curso ya existe, el sistema no dejará crearlo.

Identificador	CU8
Objetivo	El coordinador de recursos asigna recursos a un curso
Contexto	El coordinador tiene que estar designado en un curso para estar permitido para agregar dichos recursos.
Actor principal	Coordinador de recursos
Escenario principal	<ol style="list-style-type: none"> 1. El usuario accede al sistema con sus credenciales y se dirige al curso al que quiere asignarle los recursos. 2. Dispondrá de un botón para ello, el cual pulsará y mediante un menú seleccionará lo que quiere asignar. 3. Guardará los cambios y el sistema los almacenará
Extensiones	2.1. Si intenta asignar recursos no disponibles, el sistema se lo denegará y notificará

Identificador	CU9
Objetivo	Visualizar estadísticas de alcance de los cursos
Contexto	Los usuarios (a excepción del visitante) podrán ver estadísticas de alcance que el sistema generará previamente sobre un curso
Actor principal	Alumno, administrador, coordinador de cursos, coordinador de recursos.
Escenario principal	<ol style="list-style-type: none"> 1. El sistema generará estadísticas de alcance de un curso. 2. El usuario accederá al curso. 3. Al acceder, podrá ver dichas estadísticas
Extensiones	<i>Sin extensiones</i>

Identificador	CU10
Objetivo	Resolver problemas acerca del sistema
Contexto	El administrador se encargará de resolver cualquier tipo de problema del sistema
Actor principal	Administrador
Escenario principal	<ol style="list-style-type: none"> 1. El sistema generara un informe de error 2. El administrador lo visualizara y acto seguido lo intentara resolver
Extensiones	<i>Sin extensiones</i>

3.1 Diagrama de casos de uso

Con este diagrama podremos visualizar todo lo anteriormente expuesto usando lenguaje UML, para ello haremos uso de Visual Paradigm, una herramienta que nos ayudará a representar nuestros casos de uso como veremos a continuación.

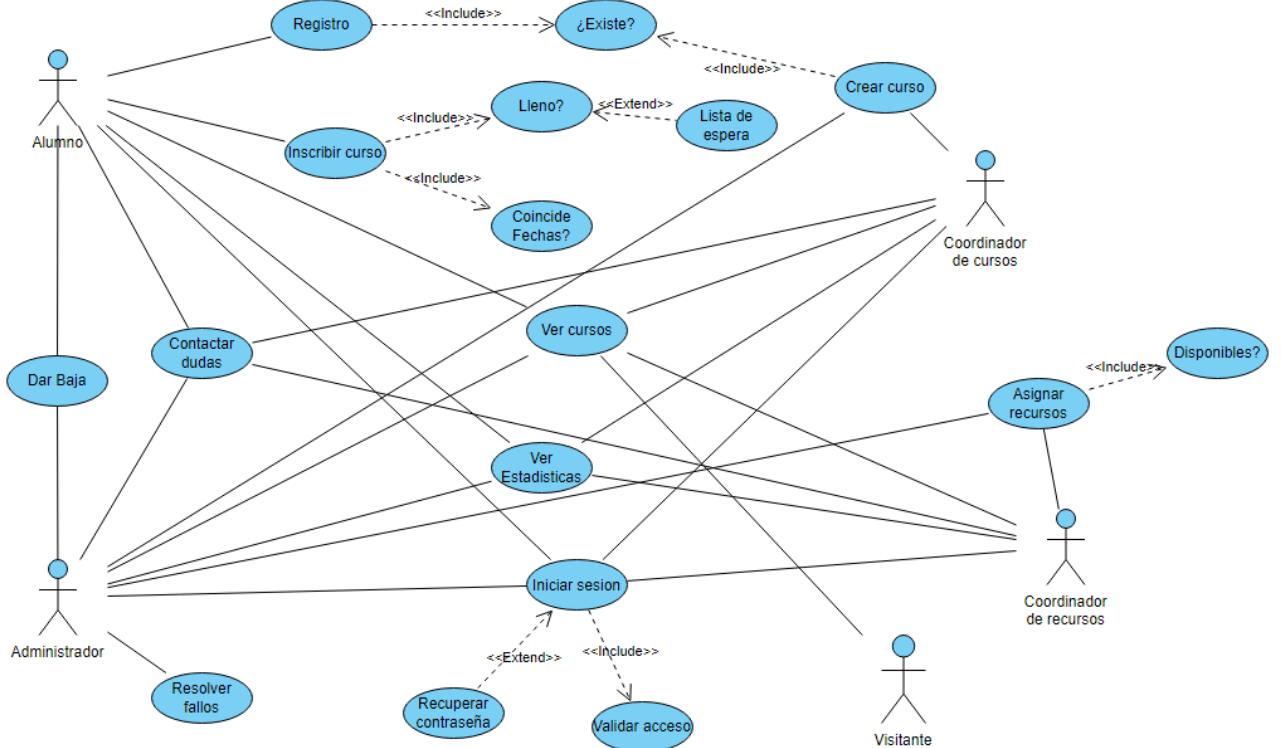


Ilustración 14

Clase	Usuario			
Descripción	Esta clase representa a cualquier usuario del sistema menos al visitante			
Atributos				
	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	nombre_completo	String	Nombre completo de dicho usuario
	-	fecha_nacimiento	Date	Fecha de nacimiento del usuario
	-	DNI	String	DNI del usuario
	-	contraseña	String	Contraseña del usuario
Operaciones				
	+ iniciar_sesion()		Bool	Operación para realizar el acceso al sistema
Parametros	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	contraseña	String	Contraseña del usuario
	+ validar_acceso()		Bool	Operación para la validación de un usuario en el sistema
	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	contraseña	String	Contraseña del usuario
	+ ver_curso()		Void	Operación para visualizar el curso solicitado
Parametros	-	id_curso	Int	Id del curso especificado
	+ buscar_curso()		Bool	Operación para a búsqueda del curso solicitado
	-	id_curso	Int	Id del curso especificado
	+ devolver_curso()		Int	Operación que devuelve el curso solicitado
	N/A	N/A	N/A	N/A
	+ ver_estadisticas()		Void	Operación para visualizar las estadísticas del curso especificado
Parametros	-	id curso	Int	Id del curso especificado
	+ consultar_calendario()		Void	Operación para visualizar el calendario
Parametros	N/A	N/A	N/A	N/A

Clase	Administrador			
Descripción	Clase heredada de la clase usuario cuya funcionalidad es el control total del sistema			
Atributos				
Atributos	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	nombre_completo	String	Nombre completo de dicho usuario
	-	fecha_nacimiento	Date	Fecha de nacimiento del usuario
	-	DNI	String	DNI del usuario
	-	contraseña	String	Contraseña del usuario
Operaciones				
	+ dar_baja()		Bool	Operación para dar de baja a un alumno
Parametros	-	usuario_uc0	String	Nombre del usuario de la UCO del alumno que quiere darse de baja
	+ crear_curso()		Bool	Operación para crear un curso
Parametros	-	nombre_curso	String	Nombre del curso especificado
	+ asignar_recursos()		Bool	Operación para asignar los recursos al curso especificado
Parametros	-	id_curso	Int	Id del curso especificado
	-	id_recurso	Int	Id del recurso a asignar
	+ editar_calendario()		Void	Operación para gestionar el calendario
Parametros	N/A	N/A	N/A	N/A

Clase	Alumno			
Descripción	Clase heredada de la clase usuario con funcionalidad específica			
Atributos				
	-	usuario_uco	String	Nombre del usuario de la UCO
	-	nombre_completo	String	Nombre completo de dicho usuario
	-	fecha_nacimiento	Date	Fecha de nacimiento del usuario
	-	DNI	String	DNI del usuario
	-	Contraseña	String	Contraseña del usuario
Operaciones				
	+ inscribir_curso()		Bool	Operación para poder inscribirse en un curso
Parametros	-	usuario_uco	String	Nombre del usuario de la UCO
	-	contraseña	String	Contraseña del usuario
	+ registro()		Bool	Operación para registrarse en el sistema
Parametros	-	usuario_uco	String	Nombre del usuario de la UCO
	-	nombre_completo	String	Nombre completo de dicho usuario
	-	fecha_nacimiento	Date	Fecha de nacimiento del usuario
	-	DNI	String	DNI del usuario
	-	contraseña	String	Contraseña del usuario
	+ fecha_disponible()		bool	Operación para comprobar si está disponible la fecha solicitada
Parametros	-	usuario_uco	String	Nombre del usuario de la UCO
	-	fecha_ini	Date	Fecha inicio curso
	-	fecha_fin	Date	Fecha fin curso

Clase	Coordinador de recursos			
Descripción	Clase heredada de la clase usuario con funcionalidad específica			
Atributos				
	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	nombre_completo	String	Nombre completo de dicho usuario
	-	fecha_nacimiento	Date	Fecha de nacimiento del usuario
	-	DNI	String	DNI del usuario
	-	contraseña	String	Contraseña del usuario
Operaciones				
	+ asignar_recurso()		Bool	Operación para poder inscribirse en un curso
Parametros	-	id_curso	Int	Id del curso especificado
	-	id_recurso	Int	Id del recurso a asignar
	-	nombre_recurso	String	Nombre del recurso a asignar
	+buscar_recurso()		Bool	Operación para buscar un recurso en el sistema
Parametros	-	id_recurso	Int	Id del recurso a buscar
	+comprobar_stock()		Bool	Operación para comprobar si existe stock del recurso buscado
Parametros	N/A	N/A	N/A	N/A

Clase	Coordinador del curso			
Descripción	Clase heredada de la clase usuario con funcionalidad específica			
Atributos				
	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	nombre_completo	String	Nombre completo de dicho usuario
	-	fecha_nacimiento	Date	Fecha de nacimiento del usuario
	-	DNI	String	DNI del usuario
	-	contraseña	String	Contraseña del usuario
Operaciones				
	+ crear_curso()		Bool	Operación para poder inscribirse en un curso

Parametros	-	id_curso	Int	Id del curso especificado
	-	nombre_curso	String	Nombre del curso
	-	fecha_ini	Date	Fecha de inicio del curso
	-	fecha_fin	Date	Fecha de finalización del curso
	-	descripcion	String	Descripción general del curso
		+ editar_calendario()	Void	Operación para editar el calendario e insertar fechas
Parametros	-	fechas_curso	Date	Vector con fecha_ini y fecha_fin a insertar en el calendario o eliminar.
		+ ver_matriculados()	Void	Operación que le sirve al coordinador para ver los alumnos matriculados en el curso
Parametros	-	id_curso	Int	ID del curso a consultar

Clase	Curso			
Descripción	Esta clase representa un curso del sistema			
Atributos				
	-	ID	Int	Identificador único de cada curso
	-	nombre	String	Nombre del curso
	-	fecha_ini	Date	Fecha de inicio del curso
	-	fecha_fin	Date	Fecha de finalización del curso
	-	descripcion	String	Descripción general del curso
	-	estadistica_alcance	float	Estadísticas del curso anterior
	-	alumnos_pasados	Int	Alumnos inscritos
	-	Aforo_disponible	Int	Aforo disponible en el curso especificado
Operaciones				
		+mostrar_informacion()	Void	Operación para mostrar toda la información acerca del curso elegido
Parametros	-	Id_curso	Int	ID del curso a consultar
		+generar_estadisticas()	Void	Operación para visualizar las estadísticas de los cursos

Parametros	- -	estadistica_alcance alumnos_pasados	Float Int	Estadísticas del curso Alumnos inscritos
		+mostrar_fecha_ini/fin()	Date	Operación para mostrar fechas de inicio y de fin de un curso
Parametros	-	id_curso	Int	ID del curso a consultar
		+ aforo_disponible()	Int	Función que nos devuelve la cantidad de plazas disponibles en el curso
Parámetros	-	id_curso	Int	ID del curso a consultar
		+ restar_cantidad()	Bool	Operación para restar la cantidad deseada del recurso
Parametros	- -	id_curso cantidad	Int Int	Identificador del recurso Cantidad para restar

Clase	Visitante			
Descripción	Usuario no registrado en el sistema			
Atributos				
	N/A	N/A	N/A	N/A
Operaciones				
	+ ver_curso()		Void	Operación para visualizar el curso solicitado
Parametros	-	id_curso	Int	Id del curso especificado
	+ consultar_calendario()		Void	Operación para visualizar el calendario
Parametros	N/A	N/A	N/A	N/A

Clase	Lista Espera			
Descripción	Lista de espera perteneciente a un curso			
Atributos				
	-	id_lista_espera	Int	Identificador de la lista
	-	posicion	Int	Posición en la que se encuentra el usuario
	-	usuario_uco	String	Nombre de usuario de la UCO
Operaciones				
		+ insertar_alumno()	Bool	Inserta un alumno en la cola y devuelve si lo ha encontrado e insertado
Parámetros	-	usuario_uco	String	Identificador único del alumno
		+ eliminar_alumno()	Bool	Elimina un alumno en la cola y devuelve si lo ha encontrado y eliminado
Parámetros	-	usuario_uco	String	Identificador único del alumno

Clase	Calendario			
Descripción	Calendario con todos los cursos activos en el sistema			
Atributos				
	-	fecha_ini_curso	Date	Fecha en la que empieza un curso.
	-	fecha fin_curso	Date	Fecha en la que termina un curso.
Operaciones				
	+insertar_curso()		Bool	Operación para insertar el curso en el calendario
Parametros	-	id_curos	Int	Id del curso especificado a insertar
	+eliminar_curso()		Bool	Operación para visualizar el calendario
Parametros	-	id_curso	Int	Id del curso especificado a insertar

Clase	Recursos			
Descripción	Representan los recursos para su posterior asignación			
Atributos				
	-	id_recurso	Int	Identificador único de cada curso
	-	nombre	String	Nombre del recurso
	-	cantidad	Int	Cantidad disponible del recurso
	-	stock	Bool	Indica si la cantidad es >0
Operaciones				
		+ consultar_stock()	Bool	Operación para consultar si hay stock disponible del recurso
Parametros	-	id_recurso	Int	Identificador del recurso
		+ restar_cantidad()	Bool	Operación para restar la cantidad deseada del recurso
Parametros	-	id_recurso	Int	Identificador del recurso
	-	cantidad	Int	Cantidad para restar
		+ reponer_cantidad()	Bool	Operación para añadir una cantidad al stock
Parámetros	-	id_recurso	Int	Identificador del recurso
	-	cantidad	Int	Cantidad para sumar
		+consultar_cantidad()	Int	Operación para consultar la cantidad disponible del recurso
Parámetros	-	id_recurso	Int	Identificador del recurso
		+insertar_recurso()	Bool	Operación para insertar un recurso en el curso especificado
	-	id_curso	Int	Id del curso para asignar
	-	id_recurso	Int	Id del recurso a asignar
	-	nombre_recurso	String	Nombre del recurso a asignar

4. Planificación (P2,P3, y P4)

Nuestra metodología aplicada para la planificación, como hemos estudiado en clase, es la SCRUM, la cual como hemos visto, es un marco que permite el trabajo colaborativo en equipos. Si bien es cierto, que, para ver todo su potencial, sería conveniente llevarla a cabo en grandes proyectos, donde los distintos roles de usuarios marcarán lo que tienen o no que hacer, así como la división del trabajo que ha de tener lugar. En nuestro caso, al ser solo tres integrantes y trabajar por igual, en cuanto a roles y división del trabajo no hay mucho que destacar, puesto que trabajamos por igual a excepción del dueño del producto, que será rotativo semanalmente dentro el equipo de desarrollo. Por lo que en esta sección adaptaremos la propia metodología a las dimensiones de nuestro proyecto, haciendo hincapié en los aspectos más básicos como los “sprints” o las “reuniones diarias” para definir las tareas de los usuarios desarrolladores del proyecto.

El sprint es el núcleo central de la metodología SCRUM, hemos entendido que se tratan de ‘mini proyectos’ con ciclos de ejecución muy cortos (en nuestro caso de aproximadamente una semana) cuyo objetivo es avanzar con el proyecto que estamos realizando, marcando unas metas y asignando dentro de los mismos unas tareas.

En nuestro caso, tal y como adjuntaremos con capturas de pantalla, hemos realizado 11 sprints a lo largo del vigente proyecto, todos ellos con algunas tareas clasificadas como abiertas, en curso, para verificar, o hechas, todo ello en función de cómo las gestionemos y el estado en el que se encuentren de desarrollo.

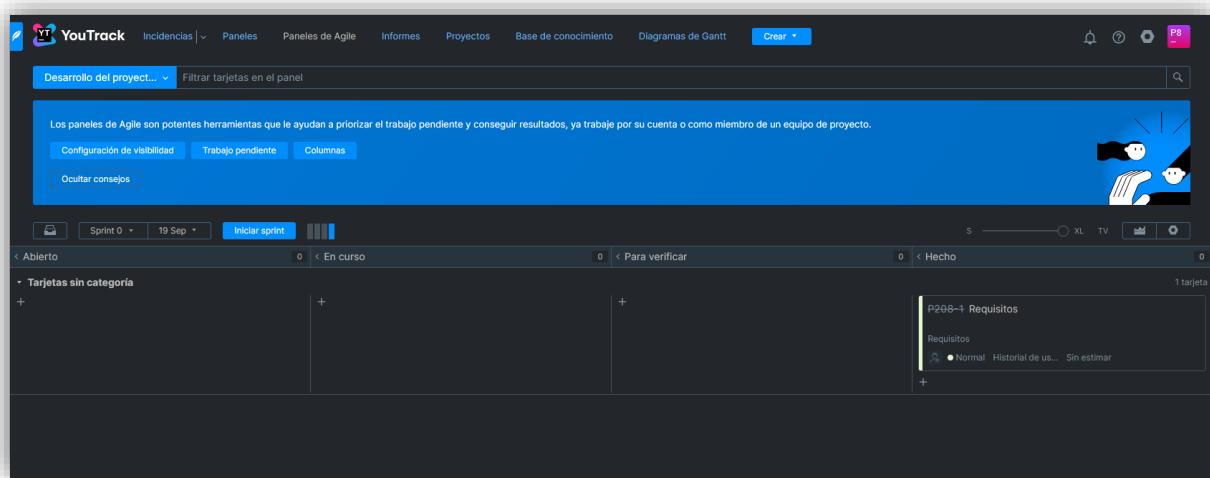


Ilustración 15

En la siguiente imagen, se detalla la primera reunión que tuvimos con el cliente.

The screenshot shows a Jira ticket for issue P208. The ticket details are as follows:

- Proyecto:** proyecto208
- Prioridad:** Normal
- Tipo:** Historial de usuario
- Estado:** Hecho
- Usuario asignado:** Sin asignar
- Sprints:** Sprint 0
- Story points:** Sin estimar

The ticket was created by p82garpr 3 weeks ago and last updated 42 minutes ago. It is visible to readers of incidents. A comment from user P8 is present:

P8 p82garpr • comentada 19 Sep 2022 21:39

En nuestra primera reunión hemos realizado una entrevista en grupo al cliente para realizar un análisis de requisitos de las funcionalidades que el cliente quiere para la aplicación solicitada, recopilando información tal que propósito general de la aplicación, así como tipos de usuarios que manejarán o administrarán la aplicación, información que la misma aplicación almacenará, y más información que hemos recogido para hacernos una idea de a lo que nos enfrentamos.

The ticket also includes a rich text editor interface and a footer with "Añadir comentario" and "Visible para Igual que la incidencia".

Ilustración 16

The screenshot shows a Jira board with the following columns and items:

- Aabierto:** P208-15 PAGOS (Menor Error)
- En curso:** P208-20 Primera reunión
- Para verificar:** P208-18 Inicio de sesión (Criticó)
- Hecho:** P208-11 Inscripción cursos (Criticó), P208-4 Registro de usuario (Criticó), P208-5 Baja Usuario (Criticó)

The board indicates 9 tarjetas (cards) in total. The footer shows "Propietario del panel: root".

Ilustración 17

En esta imagen vemos las historias de usuario y las tarjetas correspondientes al primer sprint, así como la duración en las fechas correspondientes.

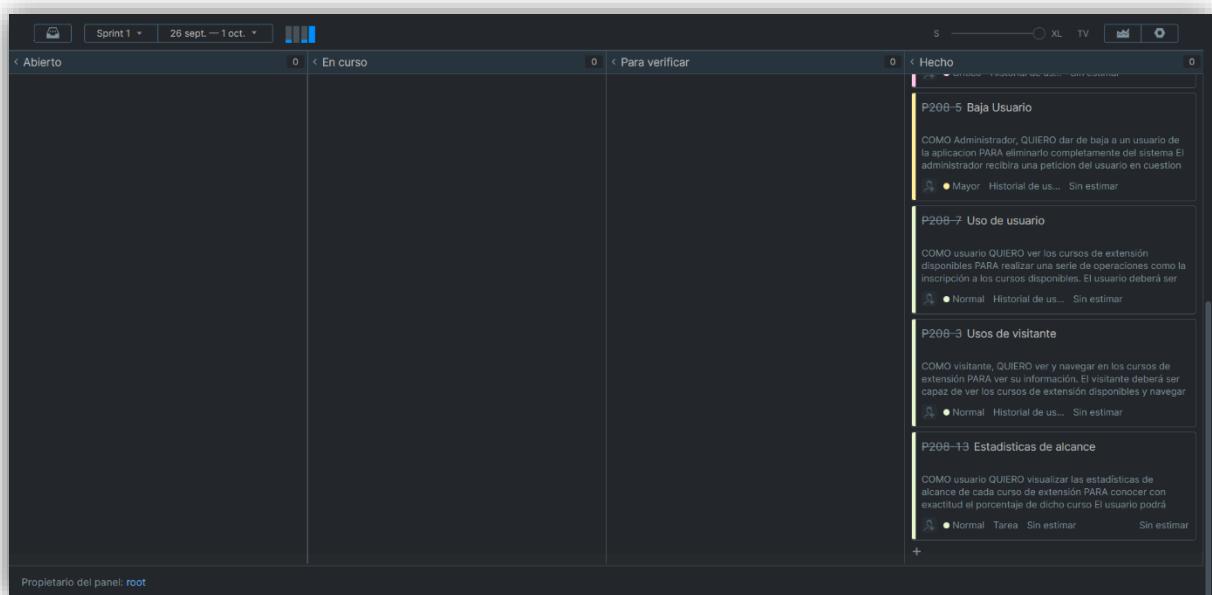


Ilustración 18

The screenshot shows the YouTrack application interface. At the top, there is a navigation bar with links: Incidencias, Paneles, Paneles de Agile, Informes, Proyectos, Base de conocimiento, Partes de horas, Más, and Incidencia nueva. Below the navigation bar, there is a search bar with the placeholder "Introducir búsqueda" and a dropdown menu set to "Todo".

The main area displays an incident detail view for "P208-20". The incident was created by "root" 2 days ago and last updated by "root" on October 15, 2022, at 20:13. The title of the incident is "★ Segunda reunión". A note in the incident states: "En nuestra primera reunión con el cliente, hemos ido recogiendo la información principal sobre las necesidades principales para poder llevar a cabo nuestro sistema e ir organizando una base esencial de él." Below the note is a text editor with a rich text toolbar and a text input field that says "Escribir un comentario, @mencionar personas".

To the right of the incident details, there is a sidebar with various settings and information:

- Visible para lectores de incidencias:** Includes fields for Projecto (proyecto208), Prioridad (Normal), Tipo (Tarea), Estado (Para verificar), Usuario asignado (Sin asignar), Sprints (Sprint 1), and Observadores (0).
- Configuración de actividad:** Includes buttons for Texto normal, Bold, Italic, etc., and tabs for Visual and Markdown.
- Añadir comentario:** A button to add a comment.
- Visible para igual que la incidencia:** A checkbox.
- Panel de información:** A blue box containing tips: "Esta vista le muestra todo lo que necesita saber para supervisar, actualizar y resolver una incidencia.", "Estas son algunas funcionalidades que le podrían resultar útiles:", and a list of buttons: "Restricciones de visibilidad", "Controles para copiar", "Comandos", "Tipos de actividad", and "Votos".

Ilustración 19

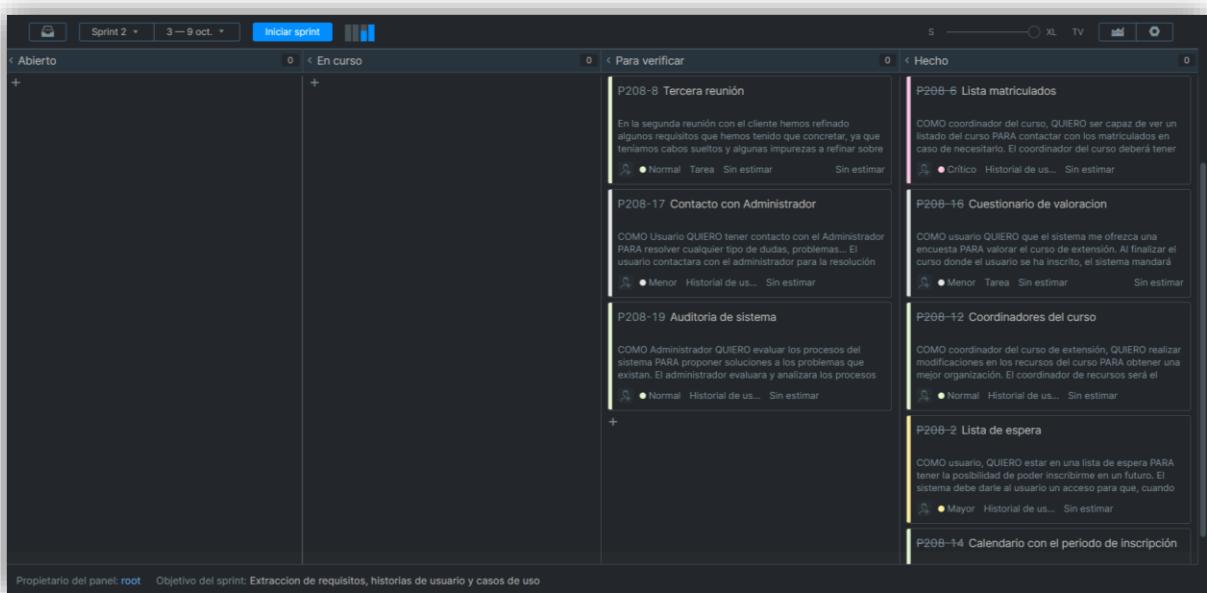


Ilustración 20

The screenshot shows the YouTrack application interface with the following details:

- Header:** YouTrack, Incidencias, Panel, Panel de Agile, Informes, Proyectos, Base de conocimiento, Partes de horas, Más, Incidencia nueva.
- Search Bar:** Todo, Introducir búsqueda.
- Incident Details:**
 - Title:** P208-8 Tercera reunión
 - Description:** En la segunda reunión con el cliente hemos refinado algunos requisitos que hemos tenido que concretar, ya que teníamos cabos sueltos y algunas impurezas a refinar sobre el sistema sobre todo en lo referido a qué puede hacer cada usuario.
 - Attachments:** Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen.
 - Activity:** Configuración de actividad.
 - Comments:** p82garpr comentada 3 oct. 2022 18:02 (@p82ceali)
 - Text Editor:** Texto normal, Visual, Markdown.
 - Buttons:** Añadir comentario, Visible para igual que la incidencia.
- Right Sidebar:**
 - Proyecto: proyecto208
 - Prioridad: Normal
 - Tipo: Tarea
 - Estado: Para verificar
 - Usuario asignado: Sin asignar
 - Sprints: Sprint 2
 - Observadores: 1 (p82ceali)
 - Paneles: Agregar al panel
 - Información: Esta vista le muestra todo lo que necesita saber para supervisar, actualizar y resolver una incidencia. Estas son algunas funcionalidades que le podrían resultar útiles: Restricciones de visibilidad, Controles para copiar, Comandos, Votos, Tipos de actividad.

Ilustración 21

YouTrack Agile Board Screenshot:

- Panel Title:** Desarrollo del proyecto...
- Header Buttons:** Incidencias, Paneles, Paneles de Agile, Informes, Proyectos, Base de conocimiento, Partes de horas, Diagramas de Gantt, Crear, Notificaciones, Help, RO.
- Search Bar:** Filtrar tarjetas en el panel.
- Panel Description:** Los paneles de Agile son potentes herramientas que te ayudan a priorizar el trabajo pendiente y conseguir resultados, ya trabajando por tu cuenta o como miembro de un equipo de proyecto.
- Card 1: P208-10 Cuarta reunion**
 - Content: En la reunión prevista para hoy nos hemos organizado en equipo el resto del trabajo en vistas a una última reunión para ultimar detalles y entregar.
 - Status: In progress
 - Labels: Normal, Tarea, Sin estimar, Sin estimar
- Card 2: P208-9 Documento ERS**
 - Content: Tenemos como equipo que confeccionar un documento especificando los requisitos que hemos extraído siguiendo el estándar IEEE, además de unos diagramas de casos de uso.
 - Status: To verify
 - Labels: Crítico, Tarea, Sin estimar, Sin estimar
- Board Columns:** Abierto, En curso, Para verificar, Hecho.
- Bottom Footer:** Propietario del panel: root | Objetivo del sprint: Elaboración documento ERS, casos de uso, e historias de usuario

Ilustración 22

YouTrack Incident View Screenshot (P208-10 Cuarta reunion):

- Incident Details:**
 - Creada por p82gerpr Hace 3 días | Actualizado por root ahora mismo
 - Visible para lectores de incidencias
- Incident Content:**
 - ★ Cuarta reunion
 - En la reunión prevista para hoy nos hemos organizado en equipo el resto del trabajo en vistas a una última reunión para ultimar detalles y entregar.
 - Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen
- Activity Configuration:** Configuración de actividad +
- Text Editor:** Texto normal, Bold, Italic, Strike, Underline, Subscript, Superscript, Paragraph, List, Table, Image, Visual, Markdown.
- Comment Area:** Escribir un comentario, @mencionar personas.
- Buttons:** Añadir comentario, Visible para igual que la incidencia.
- Right Sidebar:**
 - Proyecto: proyecto208
 - Prioridad: Normal
 - Tipo: Tarea
 - Estado: Para verificar
 - Usuario asignado: Sin asignar
 - Sprints: Sprint 3
 - Observadores: 1 > Observar incidencia
 - Paneles: > Agregar al panel
 - Información: Esta vista le muestra todo lo que necesita saber para supervisar, actualizar y resolver una incidencia. Estas son algunas funcionalidades que le podrían resultar útiles:
 - Botones: Restricciones de visibilidad, Controles para copiar, Comandos, Votos, Tipos de actividad.

Ilustración 23

YouTrack Agile board interface:

- Sprint 4** (24 — 30 oct.)
- Iniciar sprint**
- Tarjetas sin categoría**
- P208-22 Quinta reunión**: En nuestra cuarta reunión, pasamos a a introducir conceptos nuevos en nuestro proyecto, tales como un diagrama de clases junto a la matriz de trazabilidad. En esta reunión nos enfocaremos más en obtener las distintas clases en vistas a completar el diagrama de clases, además de completar también las matrices de trazabilidad.
- P208-25 Creación de las clases**: Después de analizarlo todo, hemos sacado en conclusión que nuestro sistema tendrá 6 clases principales que serán: Usuario (que es heredada por la clase Administrador, ...)
- P208-26 Matrices de trazabilidad**: Nuestra primera matriz Ise ha realizado antes de empezar a hacer el diagrama de clases, para tener los casos de uso y los requisitos funcionales bien afianzados. Una vez acabado

Ilustración 24

YouTrack incident creation interface:

Proyecto	proyecto208
Prioridad	Normal
Tipo	Tarea
Estado	Para verificar
Usuario asignado	Sin asignar
Sprints	Sprint 4
Ideal days	Sin estimar
Original estimation	Sin estimar

Quinta reunión

En nuestra cuarta reunión, pasamos a a introducir conceptos nuevos en nuestro proyecto, tales como un diagrama de clases junto a la matriz de trazabilidad. En esta reunión nos enfocaremos más en obtener las distintas clases en vistas a completar el diagrama de clases, además de completar también las matrices de trazabilidad

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Escribir un comentario, @mencionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Ilustración 25

Ilustración 26

Ilustración 27

The screenshot shows the YouTrack Agile board interface. At the top, there's a navigation bar with links like 'Incidencias', 'Panels', 'Panels de Agile', 'Informes', 'Proyectos', 'Base de conocimiento', 'Partes de horas', 'Diagramas de Gantt', 'Crear', and 'RO'. Below the navigation is a search bar with the placeholder 'Filtrar tarjetas en el panel'.

The main area displays an Agile board with four columns: 'Abierto', 'En curso', 'Para verificar', and 'Hecho'. There are two cards visible:

- P208-23 Sexta reunión**: Created by root one week ago. Description: 'En esta reunión, nos hemos enfocado en como desarrollar la matriz de trazabilidad, una vez hecho, nos centraremos en los diagramas de secuencia, los cuales hemos visto en clase e implantaremos en función de lo que obtenemos en nuestra aplicación'. Status: Para verificar.
- P208-27 Diagramas de secuencia**: Created by root one week ago. Description: 'Cada integrante del grupo hemos realizado un diagrama de secuencia. Para ello, hemos elegido 3 casos de uso, que en nuestro caso serían: CU3, un alumno quiere inscribirse en un'. Status: Sin estimar.

At the bottom left, it says '1 tarjeta seleccionada' and at the bottom right, there's a button 'Mover al siguiente sprint'.

Ilustración 28

This screenshot shows the YouTrack incident creation form for an incident titled 'Sexta reunión'.

Top navigation bar: 'P208-23 Creada por root hace una semana' and 'Visible para lectores de incidencias'.

Main content area:

- Sexta reunión**
- Description: 'En esta reunión, nos hemos enfocado en como desarrollar la matriz de trazabilidad, una vez hecho, nos centraremos en los diagramas de secuencia, los cuales hemos visto en clase e implantaremos en función de lo que obtenemos en nuestra aplicación'.
- Text input field: 'Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen'.
- Activity configuration: 'Configuración de actividad' with options for Texto normal, Bold, Italic, Strike, etc., and a switch between Visual and Markdown.
- Comment input field: 'Escribir un comentario, @mentionar personas'.
- Buttons: 'Añadir comentario' and 'Visible para Igual que la incidencia'.

Right sidebar (Project Details):

Proyecto	proyecto208
Prioridad	Normal
Tipo	Tarea
Estado	Para verificar
Usuario asignado	Sin asignar
Sprints	Sprint 5
Ideal days	Sin estimar
Original estimation	Sin estimar

Bottom sidebar (Observers and Panels):

- Observadores: 1 person (root) with an option to 'Dejar de observar'.
- Panel: 'Panels' with an option to 'Agregar al panel'.

Ilustración 29

P208-27 Creada por root Hace 4 horas
Actualizado por root ahora mismo

Visible para lectores de incidencias ▾

★ Diagramas de secuencia

Cada integrante del grupo hemos realizado un diagrama de secuencia. Para ello, hemos elegido 3 casos de uso, que en nuestro caso serían: CU3, un alumno quiere inscribirse en un curso, CU7, un coordinador de cursos quiere realizar un curso de extensión, y el caso CU8, el coordinador de recursos quiere asignar recursos a un curso.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad ▾

Texto normal | B I S | A | ... Visual Markdown

Escribir un comentario, @mencionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Normal
Tipo	Tarea
Estado	Hecho
Usuario asignado	Sin asignar
Sprints	Sprint 5
Ideal days	Sin estimar
Original estimation	Sin estimar

Observadores 1 > ★ Dejar de observar

Paneles > + Agregar al panel

Ilustración 30

P208-28 Creada por root Hace 4 horas
Actualizado por root Hace 4 horas

Visible para lectores de incidencias ▾

★ Séptima reunión

En nuestra ultima reunión con respecto a la práctica 2, hemos terminado de poner los ultimos detalles tanto en los diagramas como en nuestro proyecto del trabajo. Una vez terminado, hemos empezado con nuestra próxima práctica.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad ▾

Texto normal | B I S | A | ... Visual Markdown

Escribir un comentario, @mencionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Normal
Tipo	Tarea
Estado	En curso
Usuario asignado	Sin asignar
Sprints	Sprint 6
Ideal days	Sin estimar
Original estimation	Sin estimar

Observadores 1 > ★ Dejar de observar

Paneles > + Agregar al panel

Ilustración 31

P208-29 Creada por root Hace 3 semanas Actualizado por root hace una semana

Visible para lectores de incidencias

Octava reunión

En nuestra primera reunión de esta última entrega, empezamos a organizar cómo vamos a realizar las partes que nos quedan del proyecto y cómo vamos a gestionar la aplicación pedida.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad

Texto normal | B I S | A ... Visual Markdown

Escribir un comentario, @mencionar personas

Añadir comentario Visible para Igual que la incidencia

Proyecto	proyecto208
Prioridad	Normal
Tipo	Tarea
Estado	Hecho
Usuario asignado	Sin asignar
Sprints	Sprint 7
Ideal days	Sin estimar
Original estimation	Sin estimar

Observadores 1 > Observar incidencia

Paneles > Agregar al panel

Ilustración 32

P208-30 Creada por root Hace 3 semanas Actualizado por root Hace 3 semanas

Visible para lectores de incidencias

Novena reunión

En esta semana, nos vamos a centrar en empezar a programar nuestra base de datos en sql, y al final de esta semana intentaremos empezar el código en eclipse.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad

Texto normal | B I S | A ... Visual Markdown

Escribir un comentario, @mencionar personas

Añadir comentario Visible para Igual que la incidencia

Proyecto	proyecto208
Prioridad	Normal
Tipo	Tarea
Estado	En curso
Usuario asignado	Sin asignar
Sprints	Sprint 8
Ideal days	Sin estimar
Original estimation	Sin estimar

Observadores 1 > Observar incidencia

Paneles > Agregar al panel

Ilustración 33

P208-36 Creada por root Hace 3 semanas
Actualizado por root Hace 5 días

Visible para lectores de incidencias ▾

★ Décima reunión (Reunión 30/11/22)

El Product Owner contacta con los miembros del trabajo de forma conjunta para conocer el estado de las tareas. En esta reunión hemos terminado de entender los conceptos para linkear la bbdd y a lo largo de la semana vamos a intentar terminar la historia de usuario 1, que será crear un curso de extensión, más adelante nos repartiremos el trabajo de esta historia en concreto.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad ▾

Texto normal | B I S | A | ... Visual Markdown

Escribir un comentario, @mencionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Normal
Tipo	Tarea
Estado	En curso
Usuario asignado	Sin asignar
Sprints	Sprint 9
Ideal days	Sin estimar
Original estimation	Sin estimar

Observadores 1 > ★ Observar incidencia

Paneles > + Agregar al panel

Ilustración 34

P208-37 Creada por p82garpr Hace 2 semanas
Actualizado por p82garpr Hace 5 días

Visible para lectores de incidencias ▾

★ Undécima reunión (Reunión 2/12/22)

El Product Owner contacta con los miembros del trabajo de forma conjunta para conocer el estado de las tareas. Por causa de un imprevisto con los conectores en c++, nos vemos obligados a cambiar a un plan B, organizar nuestra base de datos con ficheros de texto plano. Por ello, comprobamos que los miembros van un poco atrasados con las tareas.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad ▾

Texto normal | B I S | A | ... Visual Markdown

Escribir un comentario, @mencionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Crítico
Tipo	Tarea
Estado	En curso
Usuario asignado	Sin asignar
Sprints	Sprint 9
Ideal days	Sin estimar
Original estimation	Sin estimar

Observadores 1 > ★ Dejar de observar

Paneles > + Agregar al panel

Ilustración 35

P208-50 Creada por root Hace 2 semanas
Actualizado por p82garpr Hace 5 días

Visible para lectores de incidencias ▾

☆ Duodécima reunión (Reunión 6/12/22) i

El Product Owner contacta con los miembros del trabajo de forma conjunta para conocer el estado de las tareas. En esta reunión, por falta de tiempo y de recursos, hemos decidido finalmente no usar una interfaz en nuestra aplicación, por lo que esta semana nos dedicaremos a añadir algunas historias de usuario más y volver a repasar todo el código para que no de errores.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad ▾

Texto normal | **B** *I* ~~S~~ | A | ... Visual Markdown Aa ?

Escribir un comentario, @mentionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Normal N
Tipo	Tarea
Estado	En curso
Usuario asignado	Sin asignar
Sprints	Sprint 10
Ideal days	Sin estimar
Original estimation	Sin estimar

Observadores 1 > ☆ Observar incidencia

Paneles > + Agregar al panel

Ilustración 36

P208-61 Creada por root Hace 5 días
Actualizado por p82garpr Hace 5 días

Visible para lectores de incidencias ▾

☆ Decimotercera reunión (Reunión 9/12/22) i

El Product Owner contacta con los miembros del trabajo de forma conjunta para conocer el estado de las tareas. Los miembros del equipo van a buen ritmo, por lo que creo que podrían ir un poco más avanzados, pero es preferible que el código quede lo mejor posible antes del control de errores.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad ▾

Texto normal | **B** *I* ~~S~~ | A | ... Visual Markdown Aa ?

Escribir un comentario, @mentionar personas

Añadir comentario Visible para Igual que la incidencia ▾

Proyecto	proyecto208
Prioridad	Normal N
Tipo	Tarea
Estado	En curso
Usuario asignado	Sin asignar
Sprints	Sprint 10
Ideal days	Sin estimar
Original estimation	Sin estimar

Observadores 1 > ☆ Observar incidencia

Paneles > + Agregar al panel

Ilustración 37

P208-62 Creada por root Hace 5 días Actualizado por root ayer Visible para lectores de incidencias

★ Decimocuarta reunión (Reunión 13/12/22)

El Product Owner contacta con los miembros del trabajo de forma conjunta para conocer el estado de las tareas. En esta reunión, Laura ha acabado los controles de errores correctamente y hemos empezado con las pruebas unitarias y la documentación.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad

Texto normal | B I S | A ... Visual Markdown

Escribir un comentario, @ Tachado (Ctrl+Shift+S)

Añadir comentario Visible para Igual que la incidencia

Proyecto	proyecto208
Prioridad	Normal
Tipo	Tarea
Estado	En curso
Usuario asignado	Sin asignar
Sprints	Sprint 11
Ideal days	Sin estimar
Original estimation	Sin estimar

Observadores 1 > ★ Observar incidencia

Paneles > + Agregar al panel

Ilustración 38

P208-64 Creada por root ayer Actualizado por root ayer Visible para lectores de incidencias

★ Decimoquinta reunión (Reunión 16/12/22)

El Product Owner contacta con los miembros del trabajo de forma conjunta para conocer el estado de las tareas. En esta reunión, los miembros del equipo hemos empezado y lo llevamos bastante avanzado trabajando mucho. La H6 se acabó en un día por su simplicidad.

Haga clic para adjuntar, arrastre archivos o pulse Ctrl+V para pegar una imagen

Configuración de actividad

Texto normal | B I S | A ... Visual Markdown

Escribir un comentario, @mentionar personas

Añadir comentario Visible para Igual que la incidencia

Proyecto	proyecto208
Prioridad	Normal
Tipo	Tarea
Estado	En curso
Usuario asignado	Sin asignar
Sprints	Sprint 11
Ideal days	Sin estimar
Original estimation	Sin estimar

Observadores 1 > ★ Observar incidencia

Paneles > + Agregar al panel

Ilustración 39

Hasta aquí se encontraría las capturas del seguimiento del trabajo mediante la plataforma YouTrack en lo referente a la extracción de los requisitos y a la realización de los diagramas de secuencia, los diagramas de clases, y las matrices de trazabilidad especificadas para la segunda entrega. Más adelante, en la implantación de este, se adjuntarán capturas para el seguimiento de dicha fase del proyecto.

Pincha [aquí](#) para acceder al enlace de YouTrack.

5. Diseño del sistema (P3)

En este apartado, tal y como especificamos por entonces en la segunda entrega, se mostrarán diagramas de clases y diagramas de secuencia, así como matrices de trazabilidad, están elaboradas conforme a los requisitos extraídos del cliente, lo cual no quiere decir, que para el programa final (el cual es una demo sin incluir todas las funcionalidades) se tengan dichas funcionalidades implementadas.

5.1. Diagrama de clases

A continuación, insertaremos el diagrama de clases del nuestro sistema con sus clases correspondientes.

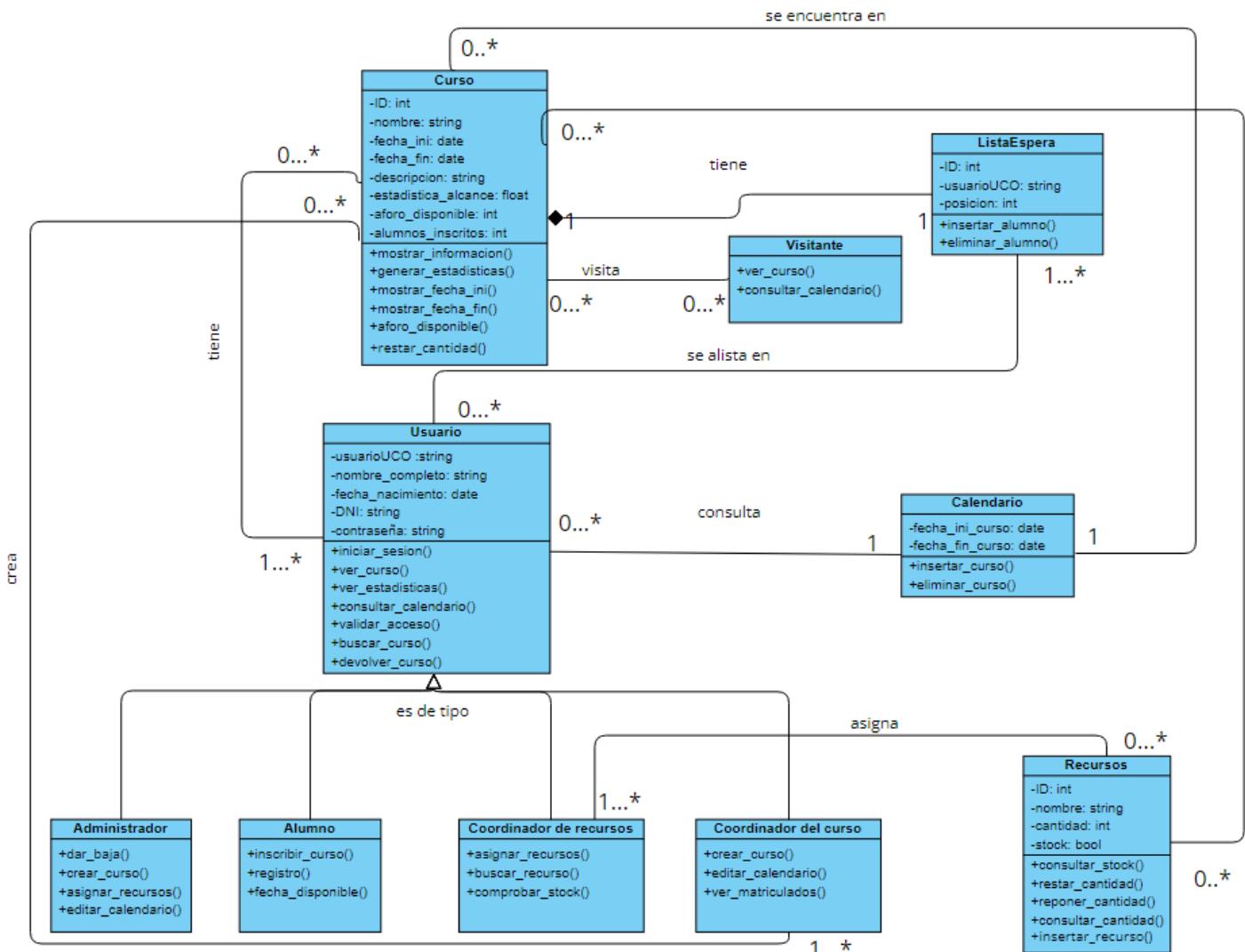


Ilustración 40

Y a continuación, las respectivas tablas con todas las clases, sus atributos y sus operaciones.

Clase	Usuario			
Descripción	Esta clase representa a cualquier usuario del sistema menos al visitante			
Atributos				
	-	usuario_UCO	String	Nombre del usuario de la UCO
	-	nombre_completo	String	Nombre completo de dicho usuario
	-	fecha_nacimiento	Date	Fecha de nacimiento del usuario
	-	DNI	String	DNI del usuario
	-	contraseña	String	Contraseña del usuario
Operaciones				
	+ iniciar_sesion()		Bool	Operación para realizar el acceso al sistema
Parámetros	-	usuario_UCO	String	Nombre del usuario de la UCO
	-	contraseña	String	Contraseña del usuario
	+ validar_acceso()		Bool	Operación para la validación de un usuario en el sistema
Parámetros	-	usuario_UCO	String	Nombre del usuario de la UCO
	-	contraseña	String	Contraseña del usuario
	+ ver_curso()		Void	Operación para visualizar el curso solicitado
Parámetros	-	id_curso	Int	Id del curso especificado
	+ buscar_curso()		Bool	Operación para a búsqueda del curso solicitado
Parámetros	-	id_curso	Int	Id del curso especificado
	+ devolver_curso()		Int	Operación que devuelve el curso solicitado

Parámetros	<i>Sin Parámetros</i>			
	+ ver_estadisticas()		Void	Operación para visualizar las estadísticas del curso especificado
Parámetros	-	id curso	Int	Id del curso especificado
	+ consultar_calendario()		Void	Operación para visualizar el calendario
Parámetros	<i>Sin Parámetros</i>			

Clase	Administrador			
Descripción	Clase heredada de la clase usuario cuya funcionalidad es el control total del sistema			
Atributos				
Atributos	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	nombre_completo	String	Nombre completo de dicho usuario
	-	fecha_nacimiento	Date	Fecha de nacimiento del usuario
	-	DNI	String	DNI del usuario
	-	contraseña	String	Contraseña del usuario
Operaciones				
	+ iniciar_sesion()		Bool	Operación para realizar el acceso al sistema
Parámetros	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	contraseña	String	Contraseña del usuario
	+ validar_acceso()		Bool	Operación para la validación de un usuario en el sistema
Parámetros	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	contraseña	String	Contraseña del usuario
	+ ver_curso()		Void	Operación para visualizar el curso solicitado
Parámetros	-	id_curso	Int	Id del curso especificado
	+ buscar_curso()		Bool	Operación para a búsqueda del curso solicitado

Parámetros	-	id_curso	Int	Id del curso especificado
	+ devolver_curso()		Int	Operación que devuelve el curso solicitado
Parámetros	<i>Sin Parámetros</i>			
	+ ver_estadisticas()		Void	Operación para visualizar las estadísticas del curso especificado
Parámetros	-	id_curso	Int	Id del curso especificado
	+ consultar_calendario()		Void	Operación para visualizar el calendario
Parámetros	<i>Sin Parámetros</i>			
	+ dar_baja()		Bool	Operación para dar de baja a un alumno
Parámetros	-	usuario_uco	String	Nombre del usuario de la UCO del alumno que quiere darse de baja
	+ crear_curso()		Bool	Operación para crear un curso
Parámetros	-	nombre_curso	String	Nombre del curso especificado
	+ asignar_recursos()		Bool	Operación para asignar los recursos al curso especificado
Parámetros	-	id_curso id_recurso	Int Int	Id del curso especificado Id del recurso a asignar
	+ editar_calendario()		Void	Operación para gestionar el calendario
Parámetros	<i>Sin Parámetros</i>			

Clase	Alumno			
Descripción	Clase heredada de la clase usuario con funcionalidad específica			
Atributos				
	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	nombre_completo	String	Nombre completo de dicho usuario
	-	fecha_nacimiento	Date	Fecha de nacimiento del usuario
	-	DNI	String	DNI del usuario
	-	Contraseña	String	Contraseña del usuario
Operaciones				
	+ iniciar_sesion()		Bool	Operación para realizar el acceso al sistema
Parámetros	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	contraseña	String	Contraseña del usuario
	+ validar_acceso()		Bool	Operación para la validación de un usuario en el sistema
Parámetros	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	contraseña	String	Contraseña del usuario
	+ ver_curso()		Void	Operación para visualizar el curso solicitado
Parámetros	-	id_curso	Int	Id del curso especificado
	+ buscar_curso()		Bool	Operación para a búsqueda del curso solicitado
Parámetros	-	id_curso	Int	Id del curso especificado
	+ devolver_curso()		Void	Operación que devuelve el curso solicitado
Parámetros	<i>Sin parámetros</i>			
	+ ver_estadisticas()		Void	Operación para visualizar las estadísticas del curso especificado
Parámetros	-	id curso	Int	Id del curso especificado

	+ consultar_calendario()		Void	Operación para visualizar el calendario
Parámetros	<i>Sin parámetros</i>			
	+ inscribir_curso()		Bool	Operación para poder inscribirse en un curso
Parámetros	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	contraseña	String	Contraseña del usuario
	+ registro()		Bool	Operación para registrarse en el sistema
Parámetros	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	nombre_completo	String	Nombre completo de dicho usuario
	-	fecha_nacimiento	Date	Fecha de nacimiento del usuario
	-	DNI	String	DNI del usuario
	-	contraseña	String	Contraseña del usuario
	+fecha_disponible()		bool	Operación para comprobar si está disponible la fecha solicitada
Parámetros	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	fecha_ini	Date	Fecha inicio curso
	-	fecha_fin	Date	Fecha fin curso

Clase	Coordinador de recursos			
Descripción	Clase heredada de la clase usuario con funcionalidad específica			
Atributos				
	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	nombre_completo	String	Nombre completo de dicho usuario
	-	fecha_nacimiento	Date	Fecha de nacimiento del usuario
	-	DNI	String	DNI del usuario
	-	contraseña	String	Contraseña del usuario
Operaciones				
	+ iniciar_sesion()		Bool	Operación para realizar el acceso al sistema

Parámetros	-	usuario_uco contraseña	String String	Nombre del usuario de la UCO Contraseña del usuario
		+ validar_acceso()	Bool	Operación para la validación de un usuario en el sistema
Parámetros	-	usuario_uco contraseña	String String	Nombre del usuario de la UCO Contraseña del usuario
		+ ver_curso()	Void	Operación para visualizar el curso solicitado
Parámetros	-	id_curso	Int	Id del curso especificado
		+ buscar_curso()	Bool	Operación para a búsqueda del curso solicitado
Parámetros	-	id_curso	Int	Id del curso especificado
		+ devolver_curso()	Int	Operación que devuelve el curso solicitado
Parámetros				<i>Sin Parámetros</i>
		+ ver_estadisticas()	Void	Operación para visualizar las estadísticas del curso especificado
Parámetros	-	id curso	Int	Id del curso especificado
		+ consultar_calendario()	Void	Operación para visualizar el calendario
Parámetros				<i>Sin Parámetros</i>
		+ asignar_recurso()	Bool	Operación para poder inscribirse en un curso
Parámetros	-	id_curso id_recurso nombre_recurso	Int Int String	Id del curso especificado Id del recurso a asignar Nombre del recurso a asignar
		+buscar_recurso()	Bool	Operación para buscar un recurso en el sistema
Parámetros	-	id_recurso	Int	Id del recurso a buscar
		+comprobar_stock()	Bool	Operación para comprobar si existe stock del recurso buscado
Parámetros				<i>Sin Parámetros</i>

Clase	Coordinador del curso			
Descripción	Clase heredada de la clase usuario con funcionalidad específica			
Atributos				
	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	nombre_completo	String	Nombre completo de dicho usuario
	-	fecha_nacimiento	Date	Fecha de nacimiento del usuario
	-	DNI	String	DNI del usuario
	-	contraseña	String	Contraseña del usuario
Operaciones				
	+ iniciar_sesion()		Bool	Operación para realizar el acceso al sistema
Parámetros	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	contraseña	String	Contraseña del usuario
	+ validar_acceso()		Bool	Operación para la validación de un usuario en el sistema
Parámetros	-	usuario_uc0	String	Nombre del usuario de la UCO
	-	contraseña	String	Contraseña del usuario
	+ ver_curso()		Void	Operación para visualizar el curso solicitado
Parámetros	-	id_curso	Int	Id del curso especificado
	+ buscar_curso()		Bool	Operación para a búsqueda del curso solicitado
Parámetros	-	id_curso	Int	Id del curso especificado
	+ devolver_curso()		Int	Operación que devuelve el curso solicitado
Parámetros	<i>Sin Parámetros</i>			
	+ ver_estadisticas()		Void	Operación para visualizar las estadísticas del curso especificado
Parámetros	-	id curso	Int	Id del curso especificado
	+ consultar_calendario()		Void	Operación para visualizar el calendario
Parámetros	<i>Sin Parámetros</i>			
	+ crear_curso()		Bool	Operación para poder inscribirse en un curso

Parámetros	-	id_curso nombre_curso fecha_ini fecha_fin descripcion	Int String Date Date String	Id del curso especificado Nombre del curso Fecha de inicio del curso Fecha de finalización del curso Descripción general del curso
		+ editar_calendario()	Void	Operación para editar el calendario e insertar fechas
Parámetros	-	fechas_curso	Date	Vector con fecha_ini y fecha_fin a insertar en el calendario o eliminar.
		+ ver_matriculados()	Void	Operación que le sirve al coordinador para ver los alumnos matriculados en el curso
Parámetros	-	id_curso	Int	ID del curso a consultar

Clase	Curso			
Descripción	Esta clase representa un curso del sistema			
Atributos				
	-	ID	Int	Identificador único de cada curso
	-	nombre	String	Nombre del curso
	-	fecha_ini	Date	Fecha de inicio del curso
	-	fecha_fin	Date	Fecha de finalización del curso
	-	descripcion	String	Descripción general del curso
	-	estadistica_alcance	float	Estadísticas del curso anterior
	-	alumnos_pasados	Int	Alumnos inscritos
	-	Aforo_disponible	Int	Aforo disponible en el curso
Operaciones				
		+mostrar_informacion()	Void	Operación para mostrar toda la información acerca del curso elegido
Parámetros	-	id_curso	Int	ID del curso a consultar
		+generar_estadisticas()	Void	Operación para visualizar las estadísticas de los cursos
Parámetros	-	estadistica_alcance	Float	Estadísticas del curso
	-	alumnos_pasados	Int	Alumnos inscritos
		+mostrar_fecha_ini/fin()	Date	Operación para mostrar fechas de inicio y de fin de un curso
Parámetros	-	id_curso	Int	ID del curso a consultar
		+ aforo_disponible()	Int	Función que nos devuelve la cantidad de plazas disponibles en el curso
Parámetros	-	id_curso	Int	ID del curso a consultar
		+ restar_cantidad()	Bool	Operación para restar la cantidad deseada del recurso
Parámetros	-	id_curso	Int	Identificador del recurso
	-	cantidad	Int	Cantidad para restar

Clase	Visitante							
Descripción	Usuario no registrado en el sistema							
Atributos								
<i>Sin Atributos</i>								
Operaciones								
	+ ver_curso()		Void	Operación para visualizar el curso solicitado				
Parámetros	-	id_curso	Int	Id del curso especificado				
	+ consultar_calendario()		Void	Operación para visualizar el calendario				
Parámetros	<i>Sin Parámetros</i>							

Clase	Lista Espera							
Descripción	Lista de espera perteneciente a un curso							
Atributos								
	-	ID	Int	Identificador de la lista				
	-	posicion	Int	Posición de usuario				
	-	usuario_uco	String	Nombre de usuario de la UCO				
Operaciones								
	+ insertar_alumno()		Bool	Inserta un alumno en la cola y devuelve si lo ha encontrado e insertado				
Parámetros	-	usuario_uco	String	Identificador único del alumno				
	+ eliminar_alumno()		Bool	Elimina un alumno en la cola y devuelve si lo ha encontrado y eliminado				
Parámetros	-	usuario_uco	String	Identificador único del alumno				

Clase	Calendario			
Descripción	Calendario con todos los cursos activos en el sistema			
Atributos				
	-	fecha_ini_curso	Date	Fecha en la que empieza un curso.
	-	fecha_fin_curso	Date	Fecha en la que termina un curso.
Operaciones				
	+insertar_curso()		Bool	Operación para insertar el curso en el calendario
Parámetros	-	id_cursos	Int	Id del curso especificado a insertar
	+eliminar_curso()		Bool	Operación para visualizar el calendario
Parámetros	-	id_curso	Int	Id del curso especificado a insertar

Clase	Recursos			
Descripción	Representan los recursos para su posterior asignación			
Atributos				
	-	ID	Int	Identificador único de cada curso
	-	nombre	String	Nombre del recurso
	-	cantidad	Int	Cantidad disponible del recurso
	-	stock	Bool	Indica si la cantidad es >0
Operaciones				
	+ consultar_stock()		Bool	Operación para consultar si hay stock disponible del recurso
Parámetros	-	id_recurso	Int	Identificador del recurso
	+ restar_cantidad()		Bool	Operación para restar la cantidad deseada del recurso
Parámetros	-	id_recurso	Int	Identificador del recurso
	-	cantidad	Int	Cantidad para restar
	+ reponer_cantidad()		Bool	Operación para añadir una cantidad al stock
Parámetros	-	id_recurso	Int	Identificador del recurso
	-	cantidad	Int	Cantidad para sumar

		+consultar_cantidad()	Int	Operación para consultar la cantidad disponible del recurso
Parámetros	-	id_recurso	Int	Identificador del recurso
		+insertar_recurso()	Bool	Operación para insertar un recurso en el curso especificado
Parámetros	-	id_curso	Int	Id del curso para asignar
	-	id_recurso	Int	Id del recurso a asignar
	-	nombre_recurso	String	Nombre del recurso a asignar

5.3. Diagramas de secuencia

En esta sección insertaremos los diagramas de secuencia que nos detallan el comportamiento del sistema de manera gráfica.

5.3.1. El coordinador de cursos (o administrador) crea un curso

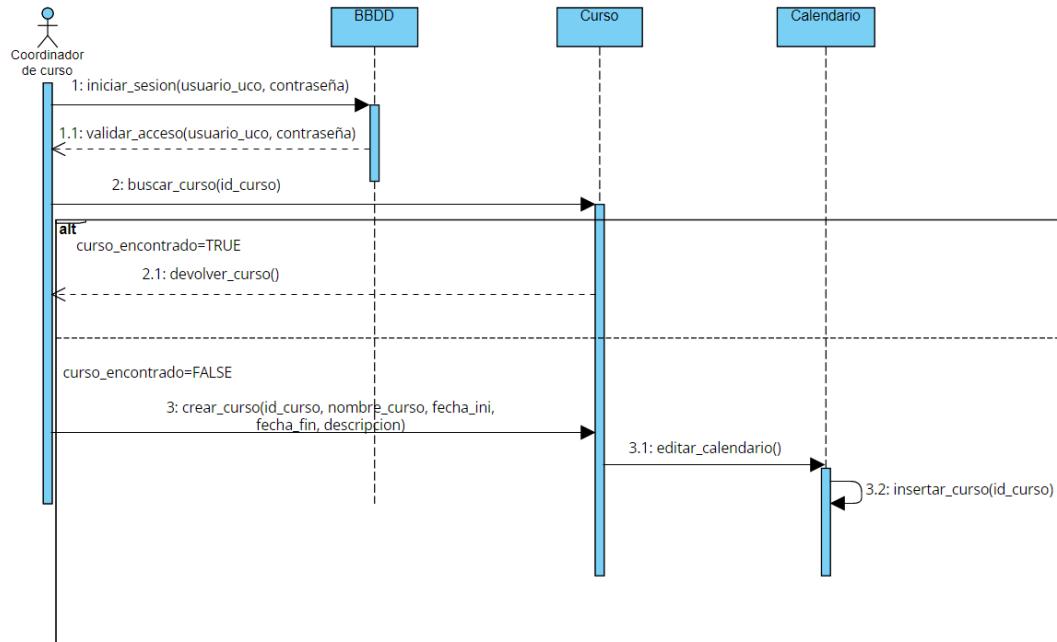


Ilustración 41

5.3.2. Asignar recursos

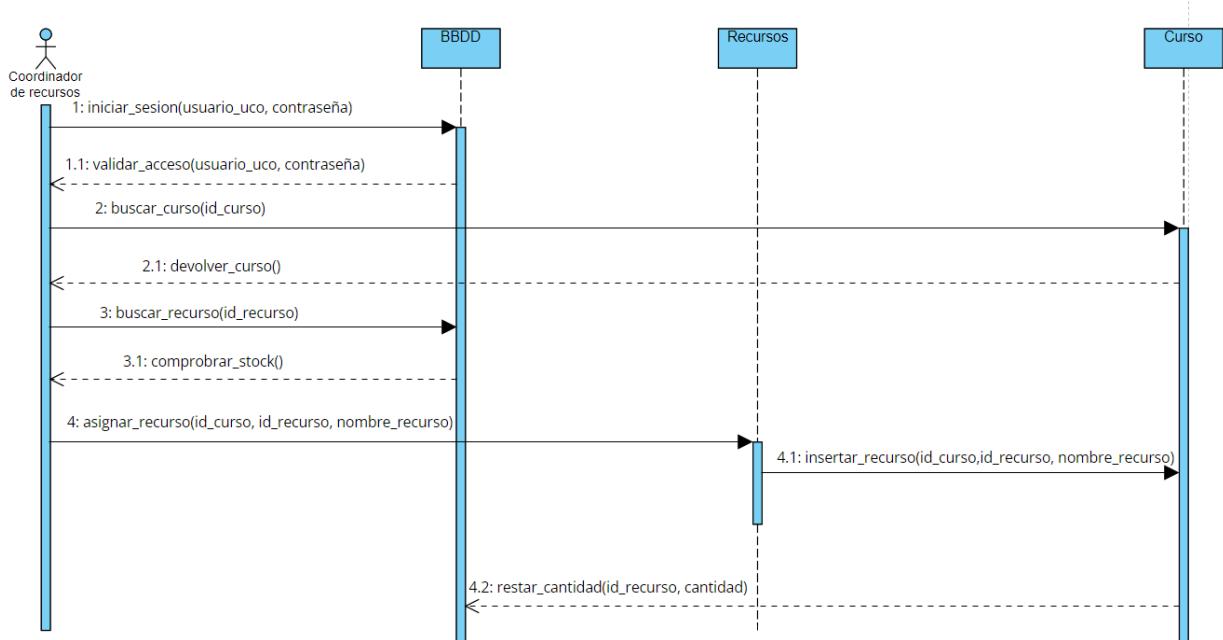


Ilustración 42

5.3.3. Un alumno se inscribe en curso.

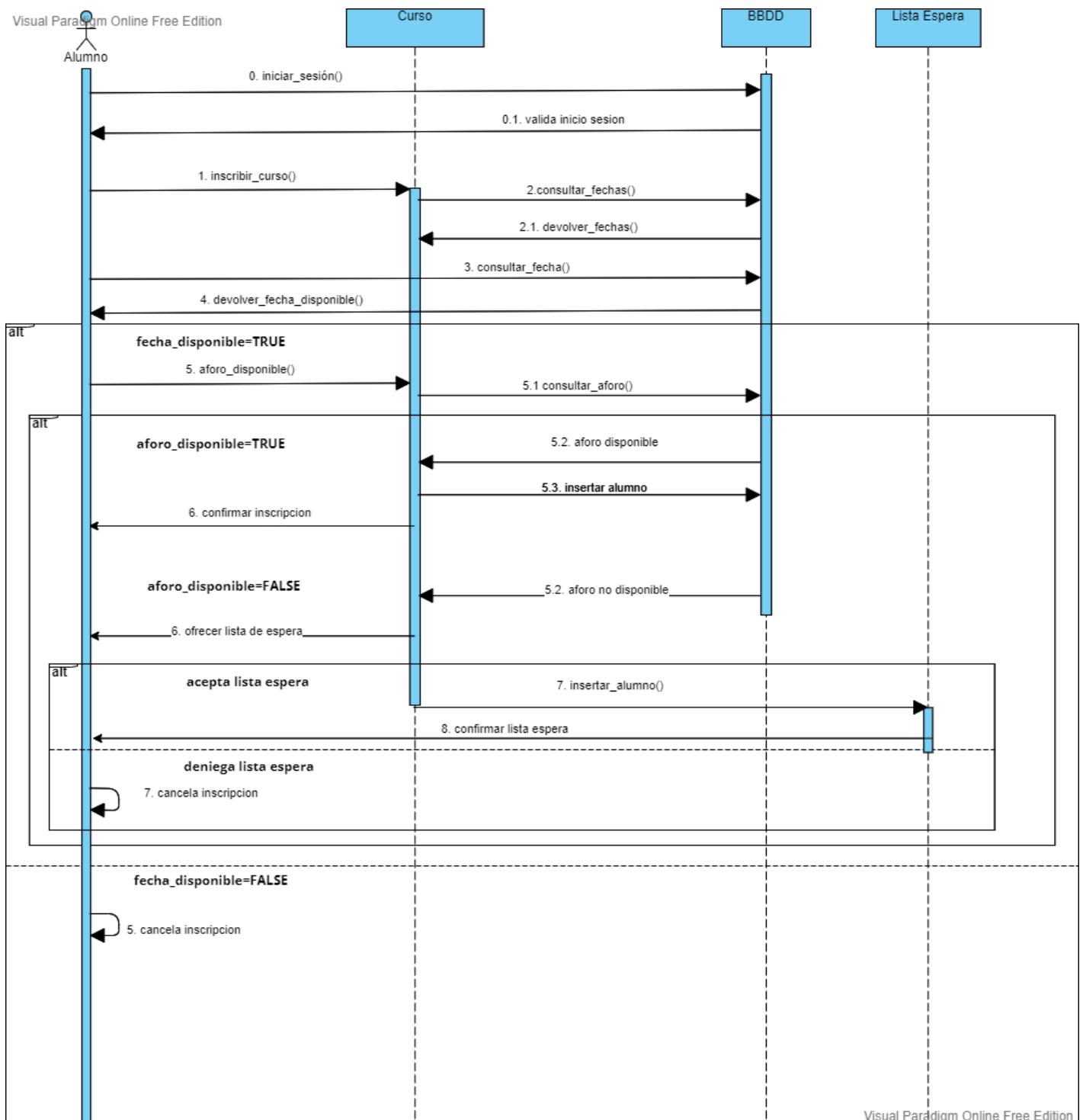


Ilustración 43

6. Validación del sistema (P4 y P3)

En este apartado detallaremos el diseño estructural que tendrá nuestro sistema, enfrentando de la siguiente manera casos de uso, requisitos funcionales, y las clases que hemos diseñado para nuestro sistema.

6.1 Matrices de trazabilidad

Las matrices de casos de uso vs requisitos funcionales, así como las de clases vs requisitos funcionales, son una manera de validar el trabajo que llevamos hecho comprobando que los elementos diseñados dan respuesta a

6.2.1 Casos de uso vs requisitos funcionales

	CU1	CU2	CU3	CU4	CU5	CU6	CU7	CU8	CU9	CU10
RF 1		X				X				
RF 2	X									
RF 3			X			X	X	X	X	
RF 4			X			X	X	X		
RF 5			X		X		X	X	X	
RF 6			X		X				X	
RF 7						X				
RF 8							X	X		
RF 9		X		X	X		X	X	X	
RF 10				X	X					X
RF 11									X	
RF 12				X	X					
RF 13			X							

6.2.2 Casos de uso vs clases

	CU1	CU2	CU3	CU4	CU5	CU6	CU7	CU8	CU9	CU10
Usuario		X			X				X	
Visitante						X				
Admin		X		X	X		X	X	X	X
Alumno	X	X	X	X	X				X	
Coord. recursos		X			X			X	X	
Coord. cursos		X			X		X		X	
Calendario			X			X	X			
Curso			X			X	X	X	X	
Lista Espera			X							
Recursos								X		

7. Implementación del sistema (P4)

En este apartado, más orientado al sistema desarrollado y entregado en la entrega final, hablaremos del mismo conforme a lo que hemos desarrollado. Pues bien, por falta de tiempo, en realidad, los requisitos no están todos cubiertos de la manera especificada anteriormente. No obstante, eso no quiere decir que lo anteriormente expuesto no tenga validez, pues está haciendo referencia a un producto final que no está terminado en su completitud, haciendo referencia, por ejemplo, al diagrama de secuencia de la inscripción en un curso, como ya detallaremos más adelante, no se tiene en cuenta la fecha disponible, así como la lista de espera, funcionalidad que quedaría pendiente para una futura entrega.

Asimismo, nuestro sistema se centra en cubrir tres funcionalidades esenciales, las cuales serían:

- 1) Gestión de usuarios, con esto nos referimos al registro e inicio de sesión con los mismos para navegar a través del sistema, implementando y cubriendo así los siguientes requisitos funcionales: RF01, RF02, RF07 (visitantes), RF09 (administrador) y todos los RNF asociados a dichos requisitos funcionales, detallados anteriormente en el documento presente, concretamente en la sección de requisitos específicos.

- 2) Gestión de cursos de extensión. Con esto cubrimos los requisitos funcionales: RF03, RF05^{*1}, RF08^{*2}. Con esto, seremos capaces, en nuestro sistema desarrollado, tanto de crear los cursos con sus datos correspondientes, como de visualizarlos en un listado de estos.

***1:** Las estadísticas de alcance, las maneja el sistema en su estructura de datos, pero no hacemos aún nada con ellas, puesto que no disponemos aun de un histórico de alumnos registrados, y habría también que desarrollar funciones que nos ayuden a mostrarlas, por lo que, por ahora, cada curso, por defecto, tiene un 0% de estadísticas de alcance.

***2:** En cuanto a este requisito, hay que destacar que no queda totalmente cubierto, puesto que, no tenemos el usuario coordinador implementado totalmente y es el administrador el que realizará esta labor.

- 3) Inscripción a los cursos, con esto hacemos que los usuarios registrados en el sistema sean capaces de inscribirse en un curso disponible, dejando su DNI registrado en dicho curso y decrementando en 1 el número de plazas disponibles para el curso o, en su defecto, aumentando en 1 el numero de inscritos, haciendo así que no deje inscribirnos en el curso cuando se llegue al aforo máximo. Los requisitos funcionales cubiertos son: RF03 y RF06, así como todos los RNF asociados a los mismos, especificados anteriormente en la sección 1.3.

Cabe destacar por ello, que, en cuanto a los distintos tipos de usuarios, no hemos creado el coordinador, puesto que lo que le diferencia con el administrador es la posibilidad de asignar recursos a los distintos cursos que tienen lugar. Como hemos especificado, en cuanto a los cursos solo se implementa lo referente a la creación y visualización de estos. En realidad, en

un futuro, se implementaría todo el tema de los recursos especificado anteriormente, junto con el usuario “Coordinador”, que sea capaz de asignar dichos recursos además de crear los cursos de extensión, esto último, por ahora, realizado por el administrador. Como hemos podido denotar, tampoco hemos hecho mención alguna al calendario, del cual sí hemos hablado anteriormente y, que como podemos entender, tampoco esta implementado y se dejará como tarea futura.

Ahora pasaremos a hablar de la estructura que tendrá nuestro programa; hemos considerado que, dados los objetivos del sistema y la concurrencia que pueda tener, la manera más lógica de desarrollarlo es mediante una arquitectura cliente-servidor, que consta, como su nombre indica, de un cliente que dispone de un ejecutable, que establece una conexión con el servidor central, que, digámoslo así, almacena la base de datos y el programa principal en memoria, y que mantiene la integridad de los datos realizando copias de seguridad, manteniendo software y hardware redundante, evitando así pérdidas de información y garantizando la integridad de los datos. Nosotros como desarrolladores y no expertos en sistemas, hemos desarrollado una aplicación que funciona de manera local, aunque luego más adelante, en futuras mejoras, hablaremos sobre la arquitectura y la estructura de los datos entre otras cosas.

En cuanto a la estructura de los datos, hay que destacar el uso de la POO, para mantener una “plantilla” de los datos a almacenar y trabajar de una manera mucho más cercana al mundo real, facilitando así la programación y garantizando en consecuencia la integridad de los datos mediante el uso de instancias de objetos. Dichos datos, se encuentran almacenados en una estructura de ficheros, en los cuales se irá volcando o actualizando la información conforme vayamos realizando operaciones con los datos de nuestro sistema. Hablando más específicamente, nos encontraremos con que la información de los cursos existentes en nuestro sistema se encontrará en un fichero llamado *cursos.txt*, adjuntamos captura de la estructura de fichero que sigue.

```
class curso {
private:
    int id_curso_;
    string nombre_;
    string fecha_ini_;
    string fecha_fin_;
    string descripcion_;
    float estadistica_alcance_;
    int aforo_;
    int alumnos_inscritos_;
```

Ilustración 44

Dicha captura es la estructura que sigue nuestra clase “curso”, que coincide con la estructura que almacena el fichero de texto línea por línea. Cada línea que almacena el fichero será un tipo de dato de un curso, y un registro ocupará 8 líneas obligatoriamente. Esto será controlado a posteriori con la función que nos facilita crear un curso y que realiza la inserción de este en la base de datos, y, por tanto, en el fichero de texto.

En cuanto a los usuarios registrados en la aplicación, en nuestro caso, enteramente alumnos, se almacenarán en un fichero llamado *AlumnosRegistrados.txt*. El usuario mantendrá la siguiente estructura:

```
class usuario {  
private:  
    string usuario_ucos;  
    string nombre_completo_;  
    string fecha_nacimiento_;  
    string dni_;  
    string contrasena_;
```

Ilustración 45

Pero por, comodidad, la estructura que almacenamos en el fichero, se hace siguiendo otro orden, ya que es preferente que el primer campo sea el identificativo, en nuestro caso, el DNI.

```
/*  
 * LA FORMA DE GUARDAR LOS REGISTROS ES:  
 *      DNI  
 *      NOMBRE COMPLETO  
 *      FECHA NAC  
 *      USUARIO UCO  
 *      CONTRASEÑA  
 *      ¿CURSOS REGISTRADOS?  
 */
```

Ilustración 46

Cursos registrados se encuentra entre interrogaciones puesto que era una línea reservada de cara a futuras mejoras, para que dentro del registro de cada alumno se almacenen los cursos en los que se encuentra el mismo registrado. Por esto mismo, cada registro ocupa un total de 6 líneas en nuestro fichero de texto. Finalmente, dicha mejora fue implantada y en dicha línea se encontrarán los cursos en los que está registrado el usuario.

Por último, para almacenar información de los alumnos que se encuentran registrados en un curso, el sistema generará automáticamente por curso un fichero llamado *IDCurso + "Registrados.txt"*, por tanto, si el curso con ID 1 tiene dos alumnos registrados en el con DNI XXXXX e YYYYYY, el fichero 1Registrados.txt tendrá dos líneas con XXXXX e YYYYYY respectivamente. Esto nos ayudará a que un alumno no pueda registrarse dos veces en un mismo curso.

Conforme a los tipos de usuario se refiere, solo habrá tres tipos de inicios de sesión en nuestra aplicación, se hará en calidad de alumno, iniciando sesión en nuestra base de datos,

comprobando la existencia del registro en nuestro fichero de alumnos registrados, en calidad de visitante sin necesidad de iniciar sesión, o en calidad de administrador. Este último tendrá como usuario y contraseña por defecto “root” y “root”, y vendrá intrínseco en el sistema, sin encontrarse almacenado en ningún fichero. Para cambiar dichos parámetros habrá que comunicárselo a uno de los desarrolladores de la aplicación.

Estas decisiones de diseño anteriormente comentadas han sido tomadas ante el hecho de la necesidad de tener un entregable en una fecha determinada, por lo que todo lo que no esté implementado en este entregable tiene como principal razón la falta de tiempo y la necesidad de tener una “demo” para la fecha indicada, ya que, incluso en lo referente a los datos, lo más óptimo sería hacerlo con una base de datos, decisión que incluso durante el desarrollo se llegó a tomar, pero que dada la complejidad que esto conllevaba, tuvimos que revertir y buscar alternativas, dando lugar así a la estructura actualmente vigente mediante ficheros.

A continuación, se verá detalladamente en las capturas el trabajo realizado a lo largo de los sprints semanales para el desarrollo de la aplicación.

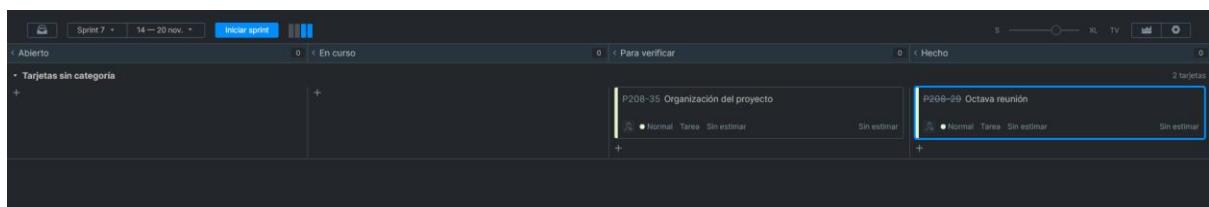


Ilustración 47



Ilustración 48

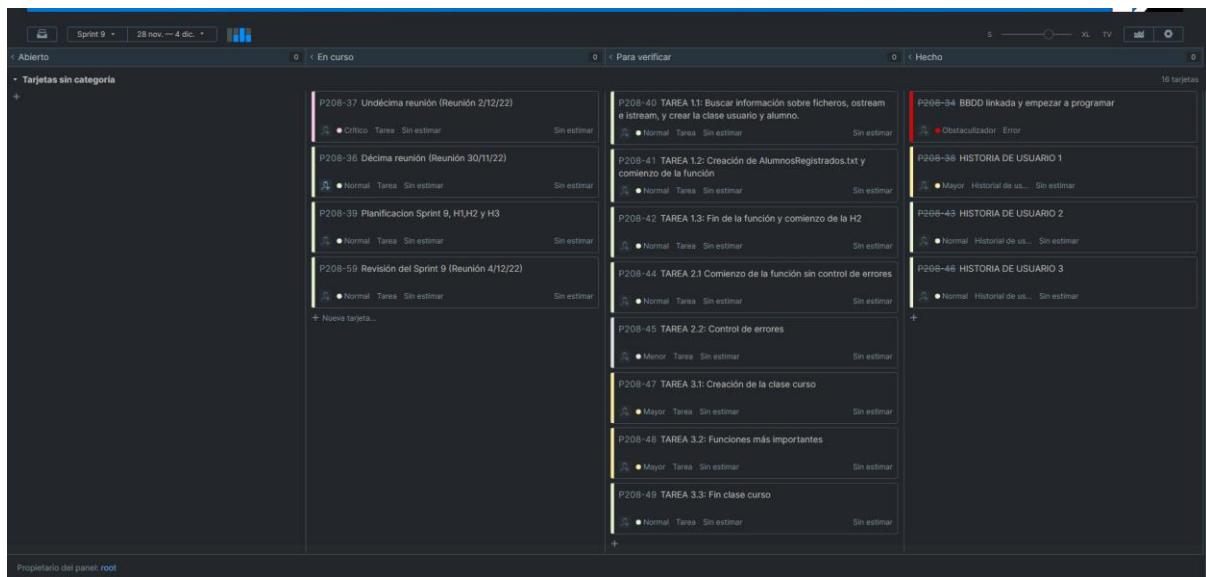


Ilustración 49

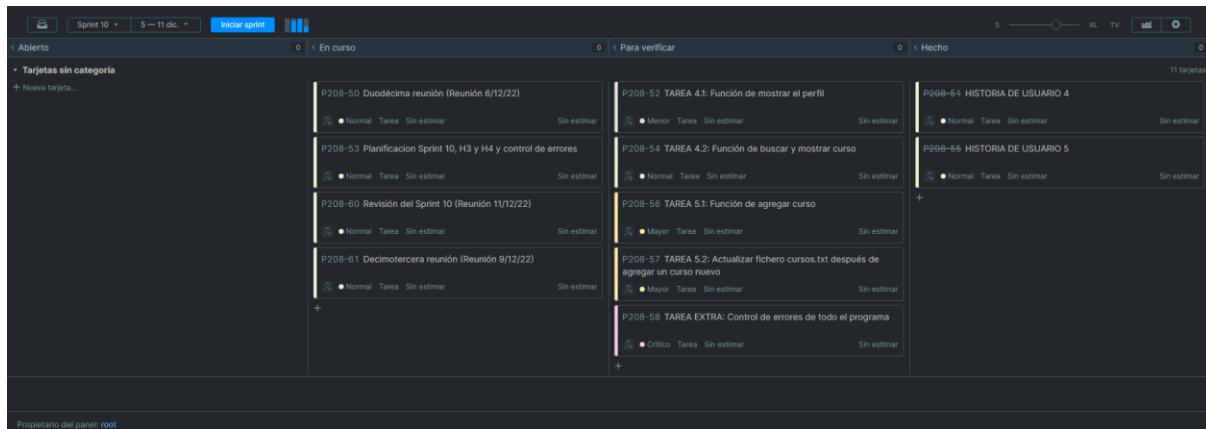


Ilustración 50

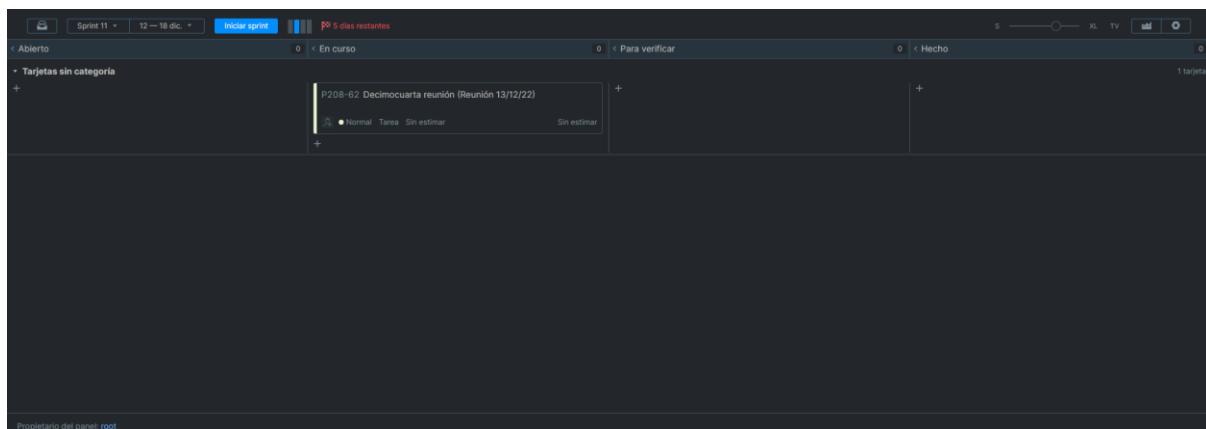


Ilustración 51

Aquí vemos como se encuentran distribuidas las tareas de los nuevos sprints creados para el desarrollo de la aplicación a lo largo de los paneles de YouTrack, a continuación, adjuntamos capturas de la base de conocimiento, donde planificamos y revisamos los sprints, que, en nuestro caso, como especificamos en su sección, duran una semana.

Hay que destacar que la revisión se hace en caso de tener tareas en el sprint, por lo que, en aquellos que no tengan tareas asignadas, no se realiza revisión y por tanto no hay captura de pantalla al respecto.

The screenshot shows a YouTrack issue titled "★ (14-11-22) PLANIFICACIÓN SPRINT 7". It includes a "Product Owner" field set to "Laura Huertes", a note about the planning session on November 14 at 19:00, and a message about organizing the remaining project parts. A comment input field is visible at the bottom.

Ilustración 52

The screenshot shows a YouTrack issue titled "★ (21-11-22) PLANIFICACIÓN SPRINT 8". It includes a "Product Owner" field set to "Isaac Cejudo", a note about the planning session on November 21 at 19:00, and a message about centering on learning SQL and Eclipse. A comment input field is visible at the bottom.

Ilustración 53

The screenshot shows a YouTrack issue titled "★ (28-11-22) PLANIFICACIÓN SPRINT 9". It includes a "Product Owner" field set to "Rafa García", a note about the planning session on November 28 at 19:00, and a message about choosing user history H1 for implementation. It also mentions dividing histories H1, H2, and H3 into X tasks, and roles for Rafa, Isaac, and Laura. A "Subartículos" section is present, along with a "REVISIÓN SPRINT 9" link and a comment input field.

Ilustración 54

★ (05-12-22) PLANIFICACIÓN SPRINT 10 ⚡ ...

Product Owner: Isaac Cejudo.

La planificación del sprint 10 se realiza el día 05/12/21 a las 19:00.

El product owner ha escogido las historia de usuario H3 y H4 para ser implementada a lo largo de este sprint.

El product owner ha decidido dividir estas H en X tareas:

Rafa se encargará de terminar las historias de usuario restantes junto a Isaac, se encargarán de la H4 y de la H3 respectivamente. Laura repasará todo el código corrigiendo errores posibles y ayudando a sus compañeros en lo que necesiten.

Se acuerda que el PO contactará con los miembros del equipo cada tres días para saber cómo va cada miembro del equipo y si necesita cualquier tipo de ayuda.

Subartículos¹

REVISIÓN SPRINT 10

Configuración de actividad ▾

P8 Escibir un comentario, @mencionar personas

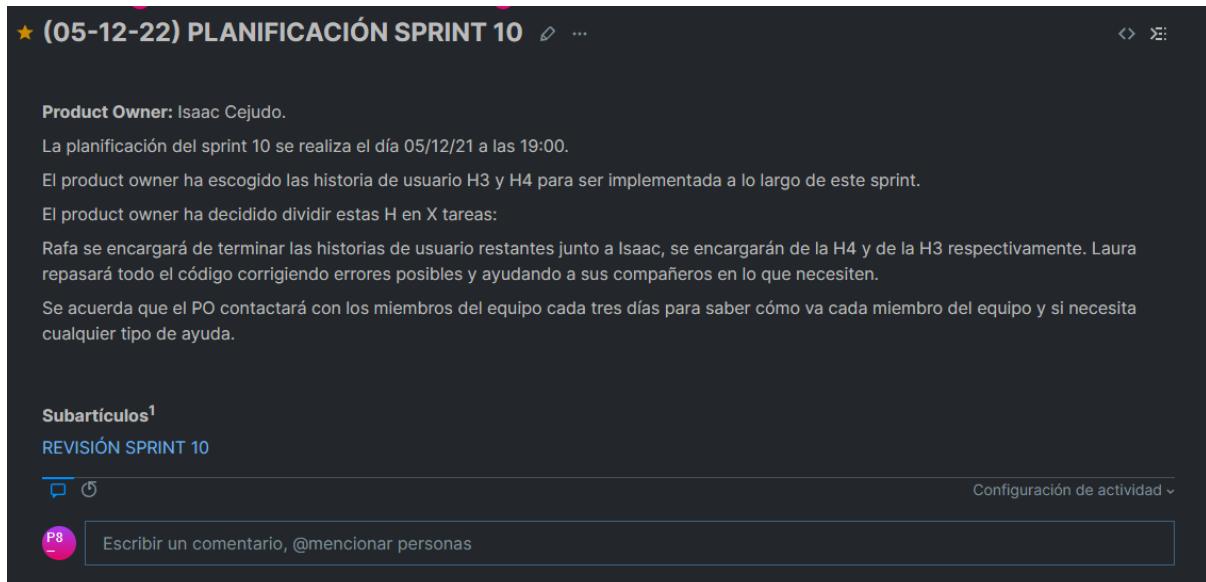


Ilustración 55

★ REVISIÓN SPRINT 10 ⚡ ...

Se han completado con éxito todas las tareas asignadas para este sprint, excepto la ultima tarea. En la ultima tarea (tarea extra) nos falta por terminar algunos controles de errores de las clases.

De momento, el funcionamiento del sistema es el siguiente:

Podemos registrarnos en el sistema, iniciar sesión y el sistema nos validará el acceso o nos lo denegará, y podemos buscar un curso de extensión en el sistema.

Configuración de actividad ▾

P8 Escibir un comentario, @mencionar personas

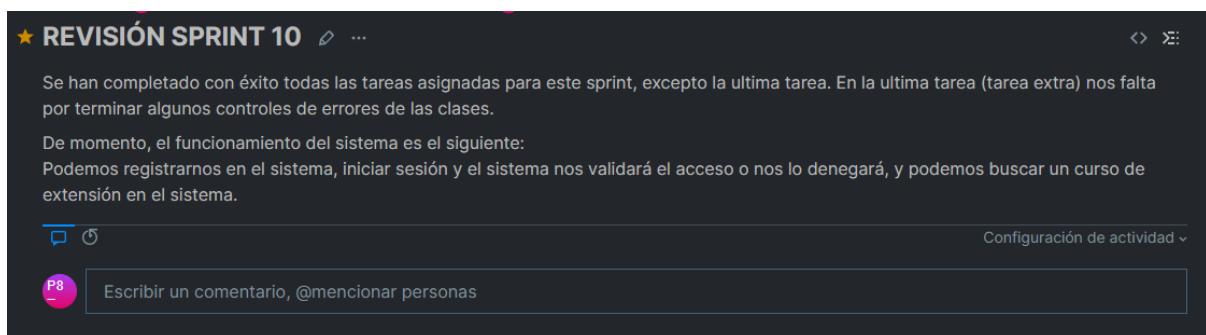


Ilustración 56

★ (12-12-22) PLANIFICACIÓN SPRINT 11 ⚡ ...

Product Owner :Laura Huertas.

El product owner ha escogido la historia de usuario H6 y lo que queda de documentación para ser implementada a lo largo de este sprint.

El product owner ha decidido dividir estas tareas en:

Laura volverá a repasar el código, y tras esto, cada miembro del equipo se encargará de hacer dos pruebas unitarias. Una vez terminadas, Rafa se encargará de terminar el manual técnico y del material audiovisual. Mientras, Laura e Isaac se encargarán de hacer el manual de usuario. Todos ellos pedirán ayuda si la necesitan.

Se acuerda que el PO contactará con los miembros del equipo cada tres días para saber cómo va cada miembro del equipo y si necesita cualquier tipo de ayuda.

Subartículos¹

REVISIÓN SPRINT 11

Configuración de actividad ▾

P8 Escibir un comentario, @mencionar personas

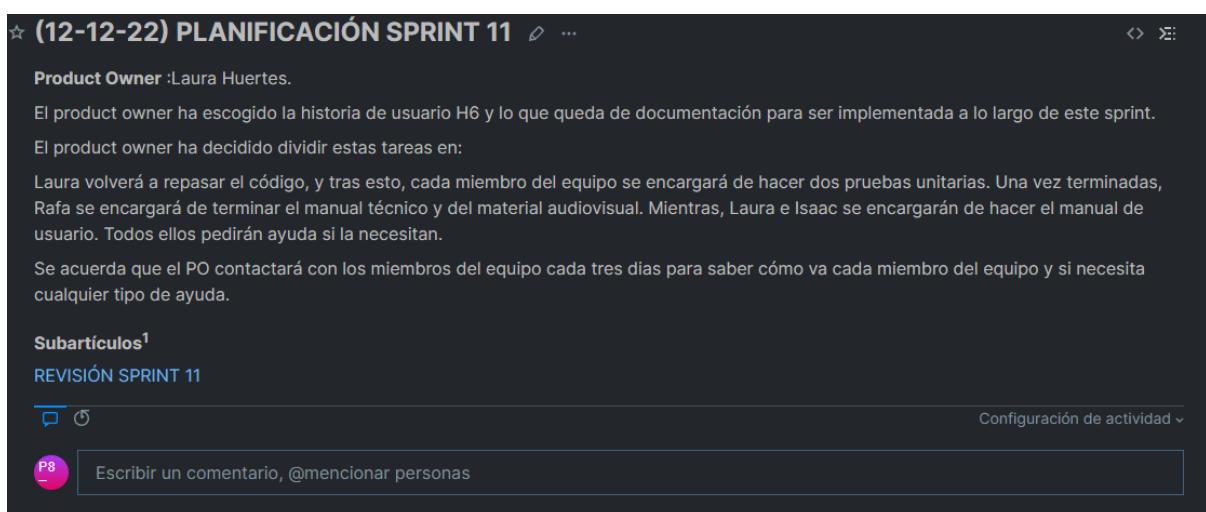


Ilustración 57

Todo el trabajo aquí descrito, así como el cumplimiento de los requisitos funcionales, o la estructura de los ficheros y de los datos almacenados en los mismos, podrán verse más a fondo en el vídeo explicativo de la aplicación. No obstante, se adjuntará aquí abajo una captura de los ficheros mencionados anteriormente, para obtener una ligera idea de cómo se estructuran en base a lo explicado anteriormente.

Como vemos en la figura de la derecha, tenemos los ficheros de los alumnos registrados en los cursos, así como el fichero que hace de base de datos de alumnos de la aplicación, y, por último, un fichero cursos que hace de base de datos de los cursos de extensión que ofrece nuestro sistema al alumnado.

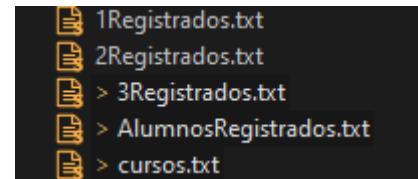


Ilustración 58

Echémosses un vistazo a los ficheros nombrados por dentro, denotaremos la estructura descrita anteriormente.

```
1 1
2 is
3 08/04/2022
4 08/05/2022
5 descripcion
6 0
7 20
8 3
9 2
10 bbdd
11 18/04/2020
12 19/04/2020
13 desc
14 0
15 20
16 3
17 3
18 POO
19 23/12/2022
20 31/12/2022
21 Aprende la POO
22 0
23 35
24 1
```

Ilustración 59: cursos.txt

```
1 31030157Y
2 Isaac Cejudo
3 19/09/2000
4 p82ceali
5 1234
6 1,2,
7 54589209C
8 Laura Huertes
9 01/12/2003
10 i12hurel
11 lago
12 2,1,3,
13 31030156M
14 Rafa Garcia
15 29/07/2000
16 p82garpr
17 1234
18 2,1,3,
```

Ilustración 60: AlumnosRegistrados.txt

```
1 31030157Y
2 31030156M
3 54589209C
4
```

Ilustración 61: 1Registrados.txt

Volviendo a hablar de la estructura de ficheros, hay que hablar de la toma de decisión para hacerlo con ficheros de texto, pues, en un principio, se iba a hacer con ficheros binarios, pero dada la complejidad que conlleva serializar los datos al trabajar con cadenas de texto, lo dejamos para futuras mejoras como alternativa a la base de datos, ya que, serializar es un trabajo muy costoso, el cual merece la pena para ganar en rapidez de procesamiento y privacidad, pero debido a tener que entregar un entregable, decidimos hacerlo con ficheros de texto para perder el menor tiempo posible y centrarnos en el resto del código de la aplicación. Como decimos, de todo esto se hablará más adelante en el siguiente apartado de futuras mejoras.

El código de la aplicación se encuentra disponible en nuestro repositorio de GitHub, al cual se puede acceder mediante el siguiente enlace pinchando [aquí](#).

7.1 IDE usado

Para concluir con esta sección, en esta subsección simplementearemos mención de Eclipse, el cual es el IDE que hemos usado a lo largo de todo el desarrollo del software, el cual fue introducido en clase y nos ha facilitado bastante el trabajo, si bien es verdad que el proceso de aprendizaje fue algo tedioso dada la complejidad del IDE respecto a nuestra experiencia, al final, a la larga, nos ha sido muy útil, sobre todo de cara al trabajo colaborativo y a las pruebas unitarias, usando el protocolo Git y sus plugin correspondientes así como CUTE Test para realizar las pruebas unitarias, también, con su correspondiente plugin facilitado por Eclipse en su Marketplace. Ni que decir tiene, que el lenguaje usado es C++ y todo esta desarrollado en C++.



Ilustración 62: Logotipo Eclipse



Ilustración 63: Logotipo CUTE

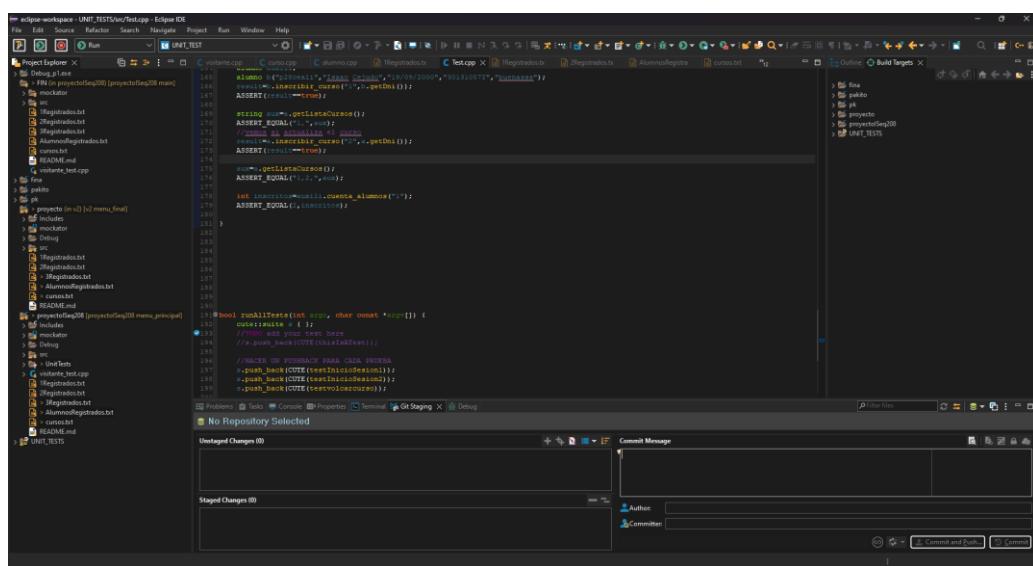


Ilustración 64: Imagen demostración del IDE usado

8. Futuras mejoras (P4)

En esta sección del documento, iremos puliendo detalles que no hemos desarrollado en el código pero que tenemos en cuenta de cara a una implementación posterior de una versión más estable y cercana a los requisitos que se nos han impuesto por parte del cliente.

Lo primero a tratar, yendo a lo más general, es hablar de nuestra arquitectura cliente-servidor ideada pero no implantada, dicho lo cual, nos permitirá en un futuro tener centralizados los datos y que no sean accesibles por los clientes, solo lo serían por parte de personal autorizado o el administrador de turno, esta medida no es una que dependa completamente de nosotros, pues nuestro trabajo se centra más en la aplicación en sí, pero, en caso de implementarlo, conllevaría unos cambios en el código, sobre todo en lo referente a las rutas de los ficheros donde se almacenan, vuelcan, y actualizan los datos de la aplicación.

Ahora pasando a hablar, sobre los datos, como hemos hablado anteriormente, los mismos se encuentran guardados en una estructura fija en ficheros de texto, lo cual lo hace muy poco flexible, porque un cambio en la estructura conllevaría un cambio en todas las funciones que gestionen nuestros ficheros. Esto, se paliaría usando bases de datos, facilitándole al programa una base de datos implantada y mantenida por un técnico superior en sistemas, los cuales nos daría muchísima más flexibilidad a la hora de tratar los dichos datos, de establecer restricciones, y de garantizar la integridad de estos, asimismo, también ganaríamos en seguridad, pues también es mucho más fácil mantener copias de seguridad o bases de datos en espejo, y en mantenimiento, el cual es mucho más fácil e intuitivo, siempre y cuando sea llevado a cabo por un experto en el ámbito.

En cuanto a interfaz se refiere, ya será más detallado en el manual de usuario, usamos una interfaz simple basada en la consola de Windows, en la cual elegiremos lo que queremos realizar en el sistema introduciendo números que referencian a dichas opciones. Esto, en un futuro, se verá sustituido por un sistema de ventanas donde no dependamos tanto del teclado, si no del ratón y podamos clicar en las funciones que deseemos, esto lo hará mucho más amigable al usuario final y nos dará más seguridad, puesto que no es lo mismo pinchar en una opción que dar lugar a la entrada de datos al sistema mediante el uso de teclado.

Otras mejoras, muchas más cercanas a su implementación, serían el borrado de cursos o de alumnos de la base de datos. Realmente, no son tareas muy costosas en relación con lo ya hecho, pero no lo hemos considerado prioritario en nuestros sprints realizados, por lo que dichas funcionalidades, por ahora, no se encuentran implantadas.

9. Pruebas del sistema (P4)

Siguiendo el guion de lo aprendido en clase, seguir unas buenas prácticas de programación durante el desarrollo del software es tan importante como probarlo, realizar un proceso de prueba de ensayo prueba y error, poniendo a prueba nuestro sistema para comprobar la existencia de errores. Resulta utópico realizar un programa con 0 fallos, pero realizando pruebas unitarias, si que podremos minimizar el número de errores y aumentar la calidad de nuestro programa. Para ello, hemos seguido el siguiente diseño tabular, en el cual se especifican las pruebas documentándolas con ID, historia de usuario, responsable de la prueba, descripción, requisitos... etc. Después de ello, adjuntaremos capturas del código tal y como lo hemos implementado, pues hay que hablar de CUTE, framework de unit-testing usado a través de Eclipse mediante plugins.

ID	01	Fecha	13/12/2022
Historia de usuario	H2		
Título de la prueba	testInicioSesion1		
Responsable de la prueba	Rafael García Pérez		
Descripción			
En esta prueba unitaria pondremos a prueba uno de los dos modos para iniciar sesión, en este caso, usando variables alojadas en el objeto creado con las credenciales que el usuario introduce.			
Prerrequisitos			
Tener un objeto creado para asignarle los valores, tener “setters” para asignarle los mismos, tener “getters” para obtenerlos y dar lugar a la comprobación de los datos.			
Entradas			
String DNI y String Password.			
Acciones a realizar			
Comprobar que la función devuelve TRUE en caso de introducir las credenciales correctas y FALSE en caso de no hacerlo.			
Salida esperada			
TRUE si los datos son validados, FALSE si la contraseña es incorrecta.			
Salida obtenida			
La salida obtenida es la esperada.			
Observaciones			
Ninguna al respecto, es un inicio de sesión sencillo pero importante, por lo que debe funcionar a la perfección, ya que es fundamental para el funcionamiento del sistema en caso de distribuir los datos en estructuras de objetos.			

ID	02	Fecha	13/12/2022
Historia de usuario	H2		
Título de la prueba	testInicioSesion2		
Responsable de la prueba	Rafael García Pérez		
Descripción			
En esta prueba unitaria pondremos el segundo de los dos modos para iniciar sesión, en este caso, usando un fichero de texto donde se encuentran los usuarios registrados en nuestro sistema.			
Prerrequisitos			
Tener un fichero llamado “AlumnosRegistrados.txt” en el directorio donde se ejecuta el programa, y donde se encuentran unos usuarios ya creados previamente, con sus respectivos usuarios y contraseñas, dichos datos se usarán para comparar si los datos introducidos coinciden.			
Entradas			
String DNI y String Password.			
Acciones a realizar			
Comprobar que la función devuelve TRUE en caso de introducir las credenciales correctas y FALSE en caso de no hacerlo.			
Salida esperada			
TRUE si los datos son validados, FALSE si la contraseña es incorrecta.			
Salida obtenida			
La salida obtenida es la esperada.			
Observaciones			
En este caso, es muy importante que se encuentre alojado el fichero en el directorio tal y como se especifica en los prerrequisitos, esto, en un futuro, con la arquitectura cliente-servidor pretendida, dicho fichero solo ha de encontrarse en el servidor que aloje todo el sistema. PD: El inicio de sesión como administrador se lleva a cabo en el MAIN del programa, ya que tendremos un usuario y contraseña específico para el inicio de sesión.			

ID	03	Fecha	13/12/2022
Historia de usuario	H5		
Título de la prueba	testvolcarcurso		
Responsable de la prueba	Isaac Manuel Cejudo Alfaro		
Descripción			
En esta prueba unitaria se pondrá a prueba la funcionalidad crear curso del administrador, en la cual, se proporcionará los datos de dicho curso y se comprobará que se inserta correctamente en el fichero “cursos.txt”			
Prerrequisitos			
El fichero “cursos.txt” deberá estar creado y vacío.			
Entradas			
Instancia de la clase Curso			
Acciones a realizar			
Comprobar que la función devuelve TRUE en caso de insertarse correctamente y FALSE en caso de no hacerlo. Además, se comprobará que las fechas proporcionadas en los atributos “fecha_fin_” y “fecha_ini_” tienen su formato correcto, en caso contrario, no se podrá insertar el curso.			
Salida esperada			
TRUE si se ha insertado correctamente y no hay redundancia en la BBDD, FALSE si las fechas de inicio y fin no tienen el formato correcto			
Salida obtenida			
La salida obtenida es la esperada.			
Observaciones			
Una vez ejecutada la prueba, hay que vaciar el fichero cursos.txt para que la próxima ejecución salga correcta.			

ID	04	Fecha	13/12/2022
Historia de usuario	H4		
Título de la prueba	testbuscarcurso		
Responsable de la prueba	Isaac Manuel Cejudo Alfaro		
Descripción			
En esta prueba unitaria se pondrá a prueba la funcionalidad buscar curso, en la cual, se proporcionará el id del curso a buscar y se comprobará que se devuelve correctamente el número de registro donde se encuentra dicho curso			
Prerrequisitos			
El fichero “cursos.txt” deberá estar creado y el curso a buscar deberá estar insertado.			
Entradas			
String id_curso			
Acciones a realizar			
Comprobar que la función devuelve el valor esperado (número de registro) en caso de encontrarse en el fichero y 0 en caso de no encontrarse.			
Salida esperada			
Número de registro donde se encuentra dicho curso, 0 si no existe ningún curso con el id buscado.			
Salida obtenida			
La salida obtenida es la esperada			
Observaciones			
Esta función ha sido creada principalmente para actualizar los inscritos de dicho curso, aunque se puede utilizar para saber si existe un curso con ese id en nuestra BBDD.			

ID	05	Fecha	13/12/2022
Historia de usuario	H1		
Título de la prueba	TestRegistro		
Responsable de la prueba	Laura Huertes Reyes		
Descripción			
En esta prueba unitaria pretendemos controlar el registro de un usuario en nuestro sistema y a su vez, que quede registrado en nuestra base de datos, en este caso en su fichero correspondiente que sería “AlumnosRegistrados.txt”.			
Prerrequisitos			
Tener creados “setters” para asignarle los mismos, tener “getters” para obtenerlos y dar lugar a la comprobación de los datos de la clase alumno.			
Entradas			
Un objeto de la clase alumno (alumno a).			
Acciones a realizar			
Comprobar que la función devuelve TRUE en caso de introducir las credenciales correctas y FALSE en caso de no hacerlo.			
Salida esperada			
TRUE si los datos son válidos y no se repiten en la base de datos, FALSE si el DNI no consta de 9 caracteres y la fecha de nacimiento no consta de 10 caracteres.			
Salida obtenida			
La salida obtenida es la esperada.			
Observaciones			
Lo único que debe tener en cuenta el usuario es en introducir un dni válido y una fecha de nacimiento con el formato impuesto en el sistema. Y por supuesto, no introducir un dni ya introducido anteriormente. Es una función sencilla y muy importante para llevar a cabo un seguimiento de nuestro sistema.			

ID	06	Fecha	13/12/2022
Historia de usuario	H3		
Título de la prueba	TestInscribirCurso		
Responsable de la prueba	Laura Huertes Reyes		
Descripción			
En esta prueba unitaria, comprobaremos si un alumno puede inscribirse en un curso correctamente, por lo que si se ha inscrito, deberían actualizarse el número de inscritos del curso respectivo en el fichero "cursos.txt" (se sumaría 1 al número anterior), se debería actualizar la lista de los cursos de ese alumno que se ha registrado en el fichero "AlumnosRegistrados.txt" (añadir el curso al que se acaba de registrar junto a los cursos que ya estaba registrado anteriormente) y por último, añadir el DNI del alumno al respectivo txt, que se han llamado "IDCurso+""Registrados.txt" " por lo tanto, si un alumno se registra en el curso 1, su DNI deberá aparecer en "1Registrados.txt" junto a los demás DNIs registrados en ese curso, si los hubiese.			
Prerrequisitos			
Necesitamos que existan en el fichero "cursos.txt" los cursos con el ID 1 y el ID 2, ya que son los cursos donde vamos a inscribir a los alumnos de la prueba unitaria.			
Entradas			
string IDCurso, string DNI			
Acciones a realizar			
Comprobar que la función devuelve TRUE en caso de haber inscrito al alumno correctamente y haberlo añadido en sus respectivos ficheros y FALSE en caso de no hacerlo o ya estar registrado en ese curso.			
Salida esperada			
TRUE si se ha inscrito correctamente, FALSE sino se ha inscrito.			
Salida obtenida			
La salida obtenida es la esperada.			
Observaciones			
Esta función es la más importante de nuestro sistema, ya que toda la aplicación va en función de los cursos y los alumnos que se registran en él. Una aplicación como la nuestra necesita obligatoriamente esta función.			

ID Prueba	Acción	Error	Solución
01	Escribir credenciales para el inicio de sesión.	No podemos automatizar el proceso porque nos piden los datos dentro de la propia función.	La solución pasa por pedir los parámetros fuera de la misma función, haciendo así que llamemos a la función por parámetros sin pedirlos dentro de la misma. De esta manera queda todo automatizado y no se usan tantas variables dentro de la función.
03	Añadir la fecha de inicio	Podemos añadir cualquier número en la variable de la fecha de inicio	Añadir en el programa un control de errores para que solo te permita introducir una fecha de inicio con 10 caracteres (formato DD/MM/YYYY), y si no se realiza correctamente que vuelva a pedirlo y así sucesivamente.
03	Añadir la fecha de fin	Podemos añadir cualquier número en la variable de la fecha de fin	Añadir en el programa un control de errores para que solo te permita introducir una fecha de fin con 10 caracteres (formato DD/MM/YYYY), y si no se realiza correctamente que vuelva a pedirlo y así sucesivamente.
05	Añadir un dni	Podemos añadir cualquier número en la variable del dni	Añadir en el programa un control de errores para que solo te permita introducir un dni con 9 caracteres, y si no se realiza correctamente que vuelva a pedirlo y así sucesivamente.
05	Añadir la fecha de nacimiento	Podemos añadir cualquier número en la variable de la fecha de nacimiento	Añadir en el programa un control de errores para que solo te permita introducir una fecha de nacimiento con 10 caracteres (formato DD/MM/YYYY), y si no se realiza correctamente que vuelva a pedirlo y así sucesivamente.
06	Obtener el getListaCursos	No se actualizaba	Implementar la función creando otro fichero auxiliar,

		bien el fichero “Alumnos registrados.txt”, por lo que la variable se llenaba con errores	donde se modifique correctamente la variable y asigne los cursos en los que está registrado cada alumno con su alumno correspondiente
06	Obtener el número de alumnos inscritos en un curso	Devolvía un número más de los alumnos que había registrados actualmente (Ej: debería devolver 2 y devuelve 3)	En la función de cuenta_alumnos, hemos eliminado un incremento de la variable a retornar, y ya nos devuelve el número correctamente

A continuación, veremos unas capturas del código, así como de la salida generada.

```
10
11•void testInicioSesion1(){  
12    usuario user1("a","b","c","d","e");  
13    user1.setDni("31885866Z");  
14    user1.setContrasena("123?321_");  
15    bool iniciarSesion= user1.iniciar_sesion(user1.getDni(), user1.getContrasena());  
16    ASSERT(iniciarSesion == true);  
17    iniciarSesion= user1.iniciar_sesion(user1.getDni(),"RandomPassword");  
18    ASSERT(iniciarSesion == false);  
19    user1.setContrasena("123");  
20    ASSERT(iniciarSesion == false);  
21    iniciarSesion= user1.iniciar_sesion(user1.getDni(),"123");  
22    ASSERT(iniciarSesion == true);  
23    iniciarSesion= user1.iniciar_sesion(user1.getDni(), user1.getContrasena());  
24    ASSERT(iniciarSesion == true);  
25    usuario user2("a","b","c","d","e");  
26    user2.setDni("31885867Y");  
27    user2.setContrasena("1234");  
28    iniciarSesion= user2.iniciar_sesion(user2.getDni(), user1.getContrasena());  
29    ASSERT(iniciarSesion == false);  
30    iniciarSesion= user2.iniciar_sesion(user2.getDni(), user2.getContrasena());  
31    ASSERT(iniciarSesion == true);  
32}
```

Ilustración 65: testInicioSesion1

```

34 void testInicioSesion2(){
35     /*PRE-CONDITION: FILE NAMED AlumnosRegistrados.txt CREATED ON $PROJ_DIR WITH
36     THE USER 31885866Z AND PASS 1223?321_ SETTED.*/
37     //THIS TEST IS DONE TO CHECK IF FILES BBDD SYSTEM WORKS AS EXPECTED.
38     alumno a;
39     a.setDni("31885866Z");
40     a.setContrasena("123?321_");
41     bool fileSignIn= a.inicio_sesion_bbdd(a.getDni(),a.getContrasena());
42     ASSERT(fileSignIn == true);
43     fileSignIn= a.inicio_sesion_bbdd(a.getDni(),"RandomPassword");
44     ASSERT(fileSignIn == false);
45     alumno b;
46     b.setDni("31885867S");
47     b.setContrasena("123123");
48     fileSignIn= b.inicio_sesion_bbdd(b.getDni(),b.getContrasena());
49     ASSERT(fileSignIn == true);
50     fileSignIn= b.inicio_sesion_bbdd("RandomDNI",b.getContrasena());
51     ASSERT(fileSignIn == false);
52 }

```

Ilustración 66: testInicioSesion2

```

54 void testvolcarcurso(){
55     /*PRE-CONDITIONS: FILE NAMED cursos.txt CREATED ON $PROJ_DIR AND THE COURSE MUSTN'T BE ON IT.*/
56     //THIS TEST IS DONE TO CHECK IF THE COURSE IS WRITTEN IN THE FILE.
57     curso c;
58     c.setId(1);
59     c.setAlumnosInscritos(2);
60     c.setFechaIni("02-08-2022");
61     c.setFechaFin("09-08-2022");
62     c.setDescripcion("Prueba de curso");
63     c.setEstadisticaAlcance(2.5);
64     c.setAforo(20);
65     bool fileCourseWritten;
66     fileCourseWritten=c.volcar_curso(c);
67     ASSERT(fileCourseWritten == true);
68
69     curso c2;
70     c2.setId(1);
71     fileCourseWritten=c2.volcar_curso(c2);
72     ASSERT(fileCourseWritten == false);
73
74     curso c3;
75     c3.setId(2);
76     fileCourseWritten=c3.volcar_curso(c3);
77     ASSERT(fileCourseWritten == true);
78
79     curso aux;
80     aux.setId(3);
81     aux.setFechaIni("12/08");
82     fileCourseWritten=aux.volcar_curso(aux);
83     ASSERT(fileCourseWritten == false);
84
85     curso aux2;
86     aux2.setId(4);
87     aux2.setFechaFin("15/09/09");
88     fileCourseWritten=aux2.volcar_curso(aux2);
89     ASSERT(fileCourseWritten == false);
90
91
92 }

```

Ilustración 67: testvolcarcurso

```

95 void testbuscarcurso(){
96     /*PRE-CONDITIONS: FILE NAMED cursos.txt CREATED ON $PROJ_DIR AND THE COURSE MUST BE ON IT.*/
97     //THIS TEST IS DONE TO CHECK IF THE COURSE IS IN THE BBDD AND RETURNS THE NUMBER OF REGISTER.
98     alumno a;
99     a.setDni("31030157Y");
100    curso c;
101    c.setId(5);
102    bool fileCourseWritten=c.volcar_curso(c);
103    ASSERT(fileCourseWritten == true);
104    ASSERT_EQUAL(3,a.buscar_curso(to_string(c.getIdCurso())));
105    curso c2;
106    c2.setId(6);
107    fileCourseWritten=c2.volcar_curso(c2);
108    ASSERT(fileCourseWritten == true);
109    ASSERT_EQUAL(4,a.buscar_curso(to_string(c2.getIdCurso())));
110    curso c3;
111    c3.setId(7);
112    ASSERT_EQUAL(0,a.buscar_curso(to_string(c3.getIdCurso())));
113
114 }

```

Ilustración 68: testbuscarcurso

```

116 void testRegistro(){
117     //THIS TEST IS USED TO VERIFY IF A USER IS ABLE TO REGISTER IN THE SYSTEM CORRECTLY, IT HAS NOT PRECONDITIONS.
118     bool result=false;
119     alumno a("il2hurel","Laura Huertes","01/12/2003","54589209C","holis");
120     result=a.registro(a);
121     ASSERT(result==true);
122
123     alumno b;
124     b.setDni("54589209C");
125     result=b.registro(b);
126     ASSERT(result==false);
127     //error ya esta registrado
128
129     alumno c;
130     c.setDni("33");
131     result=c.registro(c);
132     ASSERT(result==false);
133     //error dni
134
135     alumno x("p28ceali","Isaac Cejudo","19/09/2000","30131057Y","buenasss");
136     result=x.registro(x);
137     ASSERT(result==true);
138
139
140     alumno d("i34lolai","Iola","19/9","20245894Y","buenasss");
141     result=d.registro(d);
142     ASSERT(result==false);
143     //error fecha
144
145 }

```

Ilustración 69: testRegistro

```

147 void testInscribirCurso(){
148     /*PRE-CONDITION: FILE NAMED cursos.txt CREATED ON $PROJ_DIR WITH
149      THE COURSES WITH ID 1 AND 2 SETTED.*/
150      //THIS TEST IS TO CHECK IF A STUDENT HAVE BEEN REGISTERED IN A COURSE CORRECTLY
151
152     bool result=false;
153     alumno a("il2hurel","Laura Huertes","01/12/2003","54589209C","holis");
154     result=a.inscribir_curso("IDrandom","DNIrandom");
155     ASSERT(result==false);
156     //Inscribimos a un alumno en un curso que no existe
157
158     result=a.inscribir_curso("1",a.getDni());
159     ASSERT(result==true);
160
161     result=a.inscribir_curso("1",a.getDni());
162     ASSERT(result==false);
163     //Insertamos un alumno ya insertado
164     alumno auxili;
165     alumno b("p28ceali","Isaac Cejudo","19/09/2000","30131057Y","buenasss");
166     result=b.inscribir_curso("1",b.getDni());
167     ASSERT(result==true);
168
169     string aux=a.getListaCursos();
170     ASSERT_EQUAL("1,",aux);
171     //vemos si actualiza el curso
172     result=a.inscribir_curso("2",a.getDni());
173     ASSERT(result==true);
174
175     aux=a.getListaCursos();
176     ASSERT_EQUAL("1,2,",aux);
177
178     int inscritos=auxili.cuenta_alumnos("1");
179     ASSERT_EQUAL(2,inscritos);
180 }

```

Ilustración 70: testInscribirCurso

Mostrado esto, en la siguiente ilustración veremos la salida de los tests validada.

```

130 alumno a;
131 a.setDni("54589209C");
132 result=a.registro();
133 ASSERT(result==false);
134 //error dni
135 alumno b("p28ceali","Isaac Cejudo","19/09/2000","30131057Y","buenasss");
136 result=b.registro();
137 ASSERT(result==true);
138
139 alumno c("il2hurel","Laura Huertes","01/12/2003","54589209C","holis");
140 result=c.registro();
141 ASSERT(result==false);
142 //error fecha
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181

```

Ilustración 71: Resultado tests

Como vemos en las capturas, todas estas funciones necesitan de precondiciones que están anteriormente detalladas en el formato tabulado.

10. Referencias (P4)

En esta sección adjuntaremos enlaces con referencias que hemos consultado y nos ha servido de ayuda para realizar todas las prácticas llevadas a cabo.

- https://wiki.eclipse.org/Main_Page
- <https://dev.mysql.com/doc/connector-cpp/1.1/en/connector-cpp-examples-complete-example-1.html>
- <https://docs.phpmyadmin.net/es/latest/>
- <https://stackoverflow.com/>
- <https://docs.github.com/es>
- <https://github.com/>
- <https://cute-test.com/guides/>
- <https://cplusplus.com/doc/tutorial/functions/>
- https://www2.eii.uva.es/fund_inf/cpp/temas/10_ficheros/ficheros_cpp.html
- http://jbgarcia.webs.uvigo.es/asignaturas/TO/cursilloCpp/14_archivos.html
- <https://devdocs.io/cpp/>
- <https://www.youtube.com/watch?v=GaqqQL3wnQ>
- <https://www.youtube.com/watch?v=gPcYDjWKutE&t=881s>
- <https://www.youtube.com/watch?v=mGDs-xC-L7E>
- <https://www.youtube.com/watch?v=LSJsOZ9onSw>
- <https://www.jetbrains.com/es-es/youtrack/documentation/>
- <https://git-scm.com/docs/gittutorial>
- <https://www.visual-paradigm.com/>
- <https://www.lucidchart.com/pages/uml-sequence-diagram>
- <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>

A todo esto, habría que añadir todos los recursos facilitados por el profesor de la asignatura mediante la plataforma educativa de Moodle, con los cuales hemos obtenido distintas plantillas tanto para la confección de este documento como los demás, diversos tutoriales para CUTE, Eclipse, Git, MinGW etc.