



UNIVERSIDAD DE CÓRDOBA
ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA

INGENIERÍA INFORMÁTICA
SEGUNDO CURSO. PRIMER CUATRIMESTRE
INGENIERÍA DEL SOFTWARE

PRÁCTICA 4. MANUAL TÉCNICO

Ana Cruz Durán i82crdua@uco.es

Oscar J. Godoy Calderón i72gocao@uco.es



ED-UCO

Curso académico 2022-2023
Córdoba, 19 de diciembre de 2022

Índice

Índice de figuras	II
1. Introducción	4
2. Planificación	4
2.1. Reuniones realizadas	6
2.2. Asignación de roles	7
2.3. Historias de usuario	8
3. Especificación de Requisitos	10
3.1. Requisitos funcionales	10
3.2. Requisitos de Información	11
3.3. Requisitos no funcionales	11
4. Análisis de Requisitos	11
4.1. Historias de usuario	11
4.2. Casos de uso	14
5. Diseño del sistema	18
5.1. Diseño estructural – diagrama de clases	18
5.2. Diseño del comportamiento – diagrama de secuencia	19
6. Validación del sistema	20
6.1. Matriz de trazabilidad RF/CU	21
6.2. Matriz de trazabilidad CU/Diagrama de clases	21
7. Implementación del sistema	22
8. Pruebas del sistema	23
9. Referencias	24

Índice de figuras

1. Figura 1: Diagrama 4L.	6
2. Figura 2: Reuniones de Trello	6
3. Figura 3: Asignación de Roles	7
4. Figura 4: HUs y PB	8
5. Figura 5: Estado del tablero durante el sprint 2	9
6. Figura 6: Estado del tablero durante el sprint 3	9
7. Figura 7: Diagrama de clases	18
8. Figura 8: Diagrama de secuencia. Solicitud y registro de usuarios.	19
9. Figura 9: Diagrama de secuencia. Información de recursos para un curso.	20
10. Figura 10: Tests	23

1. Tabla 1: CU01	14
2. Tabla 2: CU02	15
3. Tabla 3: CU03	15
4. Tabla 4: CU04	15
5. Tabla 5: CU05	16
6. Tabla 6: CU06	16
7. Tabla 7: CU07	17
8. Tabla 8: CU08	17
9. Tabla 9: CU09	17
10. Tabla 10: CU10	18

1.-Introducción

El objetivo que se pretende llevar a cabo durante la realización de esta práctica es la realización de un manual técnico completo acerca de la página web desarrollada, basada en los requisitos establecidos en prácticas anteriores, siguiendo las indicaciones proporcionadas por el cliente que, en nuestro caso, ha sido el profesor de prácticas de la asignatura de Ingeniería del Software.

La estructura presentada en el documento consta de la sección 2 “Planificación”, donde veremos la evolución que ha tenido el trabajo del proyecto, apoyándonos en la herramienta “*Trello*”. La sección 3 “Especificación de requisitos” indica cuáles han sido los requisitos que se han tenido en cuenta, como son los requisitos funcionales, requisitos de información y requisitos no funcionales. La sección 4 “Análisis de requisitos” hace un repaso de las historias de usuario que han sido creadas, que incluye capturas de pantalla del tablero de trabajo de *Trello* y los casos de uso de cada historia de usuario. La sección 5 “Diseño del sistema” enseña el diseño estructural, mediante el empleo de diagramas de clases y la especificación de las clases. La sección 6 “Validación del sistema” incluye varias matrices de validación, como es la matriz de trazabilidad de los requisitos funcionales frente a los casos de uso y la matriz de trazabilidad de los casos de uso frente al diagrama de clase. La sección 7 “Implementación del sistema” explica cómo se ha implementado el sistema siguiendo la metodología Scrum y priorizando las historias de usuario por valor. La sección 8 “Pruebas del sistema” informa de cómo se han llevado a cabo las pruebas realizadas durante la implementación, detectando errores y buscando su posible solución. En la sección 9 “Referencias” encontraremos las referencias utilizadas durante el desarrollo del trabajo.

2.-Planificación

Para la realización de cada una de las prácticas, hemos seguido la metodología Scrum, por lo que nuestro equipo de trabajo estaba formado por un Product Owner (PO) y un Scrum Master (SM), y actuando como Equipo Técnico en todo momento ambos integrantes del equipo.

Hemos decidido utilizar la herramienta *Trello* para la gestión visual del proyecto debido a la facilidad de uso que aporta la herramienta. Hemos creado columnas de trabajo según hemos ido necesitando.

La asignación de roles ha sido reflejada en *Trello*, pero primeramente veremos cuáles son las funciones y responsabilidades que tienen que llevar a cabo cada rol.

Funciones del PO:

- Definir el objetivo de cada sprint con claridad.
- Gestionar el producto backlog para cumplir el objetivo.
- Negociar con el equipo de especialistas el alcance de cada sprint.

Responsabilidades del PO:

- Ordenar y priorizar los ítems del product backlog.
- Asegurarse de que todo el mundo entiende los ítems del product backlog.
- Estar disponible y accesible para el equipo.

Funciones del SM:

- Enseñar el modelo Scrum.
- Facilitar los eventos de Scrum según se requiera.
- Ayudar a eliminar impedimentos.
- Motivar cambios que incrementen la productividad del Equipo Scrum.

Responsabilidades del SM:

- Asegurarse de que la filosofía Scrum ha sido entendida y aplicada.
- Procurar que el equipo técnico sea auto-organizado y multifuncional.
- Garantizar el modelo Scrum.

Funciones del ET:

- Explicar cómo se va a conseguir el objetivo comprometido y cómo se va a generar el incremento esperado.
- Proporcionar las estimaciones (relativas) de cada elemento comprometido.
- Realizar un seguimiento diario del avance.
- Mantener el progreso actualizado.
- Garantizar la calidad del incremento.

Responsabilidades del ET:

- Garantizar las entregas.
- Ser auto-organizados y multifuncionales.
- Proporcionar calidad en cada entrega.
- Gestionar el Sprint Backlog.

Siguiendo el marco de trabajo de Scrum, hemos hecho diferentes reuniones, como por ejemplo daily meetings. Las daily meetings han sido reuniones de muy corta duración, unos 5 minutos, en las que el equipo veía qué trabajo había que desarrollar y si existían impedimentos para ello. Podemos ver en el tablero *Trello*, que estas reuniones han sido documentadas.

Otra de las reuniones que hemos llevado a cabo ha sido la Sprint Review. En esta reunión están presentes el PO y el ET, y consiste en presentar a los stakeholders el incremento del producto.

Otra de las reuniones ha sido la Sprint Retrospective, que ha tenido lugar después de la sprint review. La sprint retrospective consiste en hacer una reflexión sobre el sprint e identificar posibles mejoras de cara al siguiente. Para ello, el método que hemos empleado ha sido realizar una encuesta, a los miembros del equipo de trabajo con la herramienta Google Forms. La técnica de retrospectiva aplicada ha sido “Las 4L”. Ésta consiste en 4 apartado:

- Liked: cosas que nos han gustado.

- Learned: lo que hemos aprendido.
- Lacked: lo que hemos hecho, pero consideramos que se puede mejorar.
- Longed for: lo que hemos echado en falta, o que nos gustaría incluir.

Nos ha quedado el siguiente diagrama:

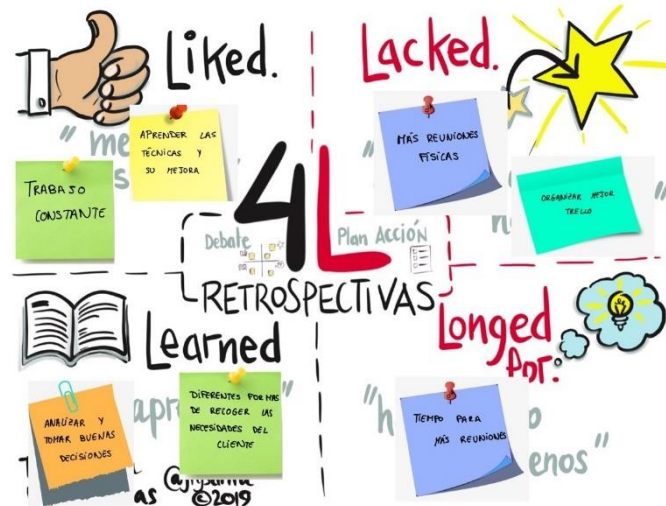


Figura 1: Diagrama 4L.

Veamos como ha ido evolucionando el tablero de *Trello* durante el desarrollo de las prácticas.

2.1. Reuniones realizadas

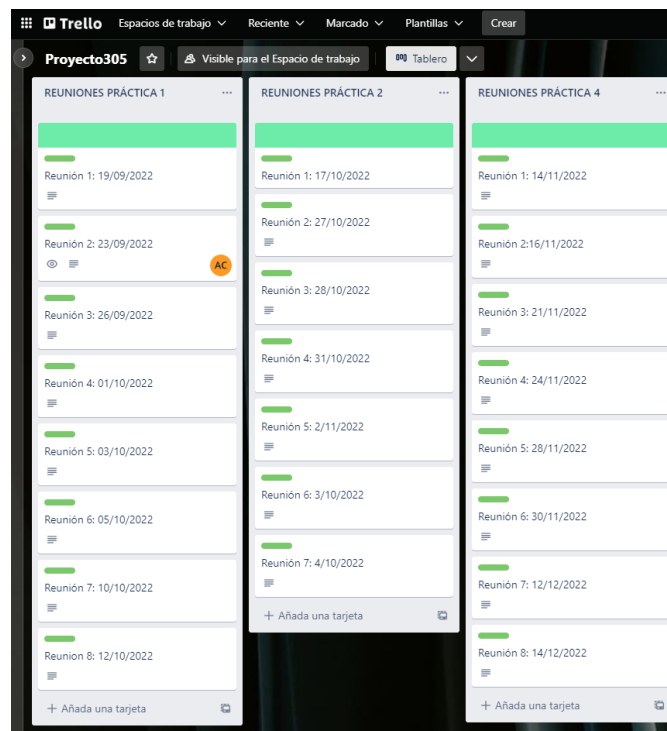


Figura 2: Reuniones Trello.

En la figura 2, podemos observar todas las reuniones que se realizaron durante el desarrollo de la práctica. Podemos identificar 3 columnas, cada una correspondiente a la práctica 1, práctica 2 y práctica 4, respectivamente. Podemos considerarlas como si se tratasen de los daily meetings.

Durante estas reuniones se abordaron todos los temas que han ido surgiendo, en la práctica 1 estuvieron centradas en el análisis y especificación de requisitos. En la práctica 2 trataron temas como el diseño y la validación del sistema y en la práctica 4 se abordó la implementación del sistema. En todas las prácticas tuvieron lugar reuniones para identificar posibles problemas y cómo solucionarlos.

2.2.-Asignación de roles

Para la realización de la práctica 4, aplicamos el marco de trabajo de Scrum, y dividimos la práctica en 3 sprints, y el equipo ha tomado todos los roles según sucedían los sprints.

Todo ha quedado reflejado en *Trello*. Tenemos plasmada esta información en una historia de usuario que se llama “Asignación de Roles”. Si entramos en la historia podemos ver lo siguiente:

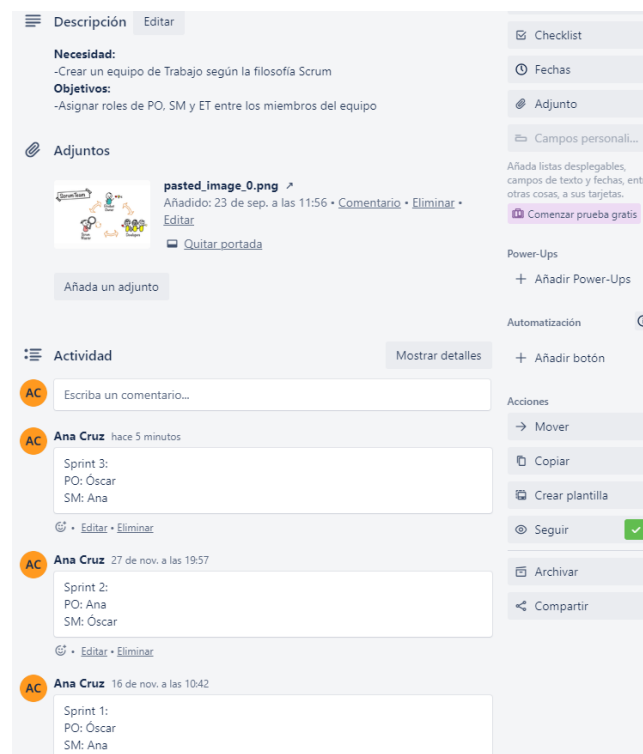


Figura 3: Asignación de Roles.

Se han detallado qué rol ha tomado cada integrante del equipo durante los sprints, conociendo previamente las funciones y responsabilidades.

2.3. Historias de usuario

A medida que han sucedido las prácticas, hemos creado las Historias de Usuario (HUs), que para la realización de la práctica 4, han sido descompuestas en tareas y éstas han sido priorizadas por valor, es decir, nuestro cliente nos pidió varias funciones para sacarle rentabilidad a la página web.

Las HUs en *Trello* han quedado de la siguiente manera:



Figura 4: HUs y PB.

Como se puede observar en la figura 4, la columna de Historias de Usuario fue creada para la práctica 1, mientras que la columna del Product Backlog han sido creadas para el desarrollo de los sprints de la práctica 4.

Durante el sprint 2, nos encontrábamos con el estado del tablero indicado en la figura 5.

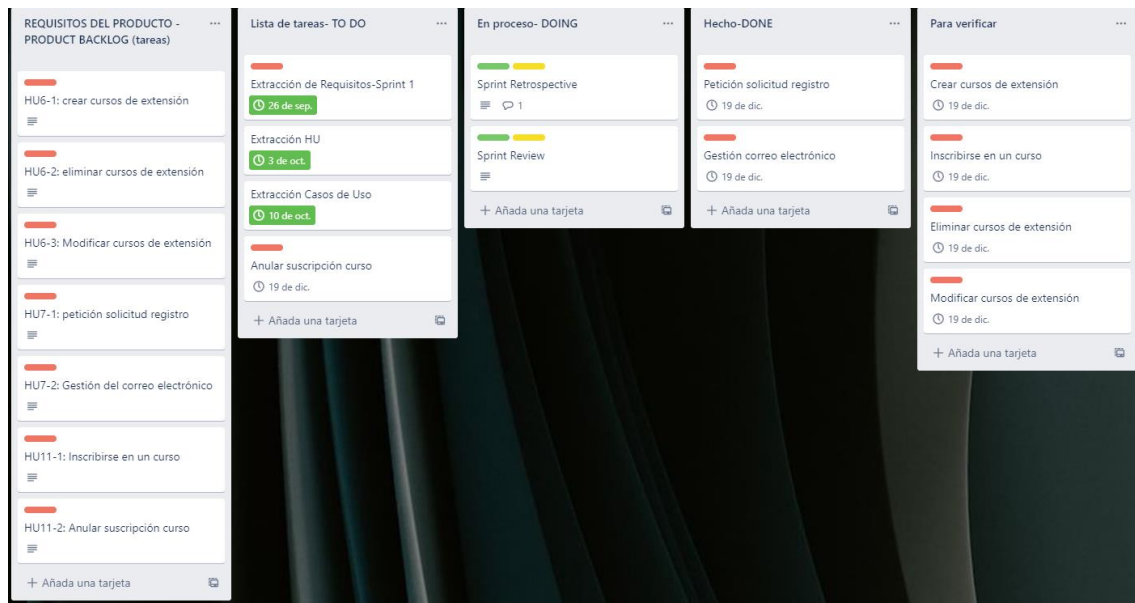


Figura 5: Estado del tablero durante el sprint 2.

Finalmente, al terminar el sprint 3, la visión global del tablero ha quedado como muestra la figura 6.

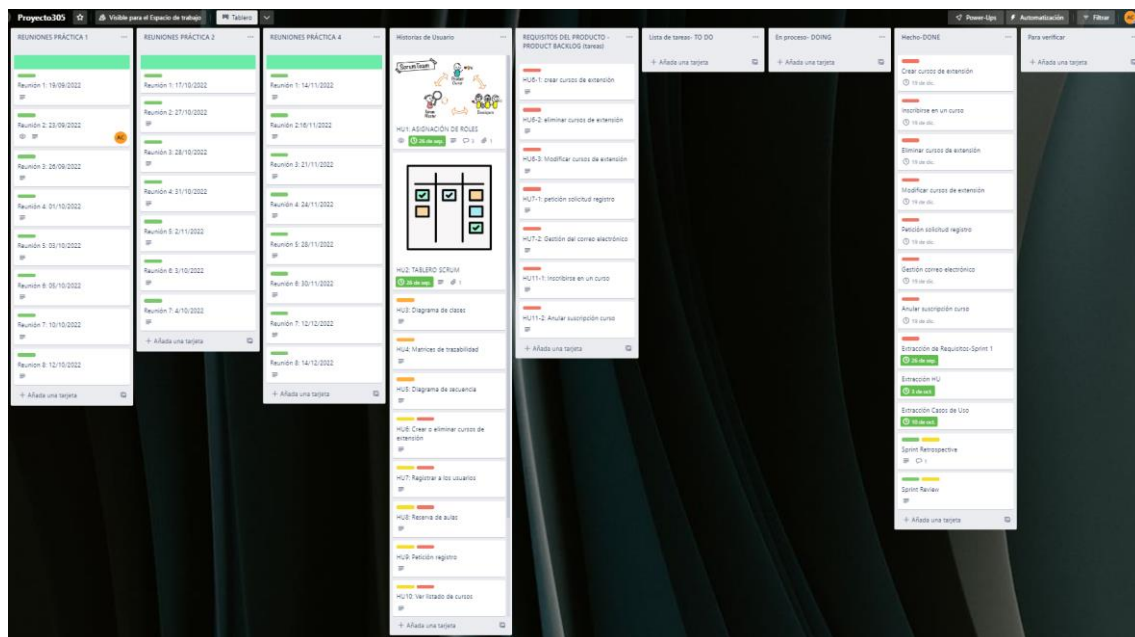


Figura 6: Estado final del tablero durante el sprint 3.

3. Especificación de Requisitos

Durante la especificación de requisitos, realizada mediante una entrevista con el cliente, hemos extraído y clasificado estos atendiendo a requisitos funcionales, requisitos de información y requisitos no funciones.

3.1. Requisitos funcionales

- **RF1:** El sistema deberá mostrar a los usuarios visitantes los cursos existentes.
- **RF2:** El sistema deberá permitir a los usuarios participantes la inscripción en un determinado curso, siempre y cuando éste tenga plazas suficientes.
- **RF3:** El sistema deberá mostrar el porcentaje de usuarios inscritos en un determinado curso.
- **RF4:** El sistema deberá asegurarse que a cada curso de extensión le corresponde un ID y que debe ser único.
- **RF5:** El sistema deberá permitir al administrador la gestión de los usuarios.
- **RF6:** El sistema deberá permitir a los usuarios registrados la modificación de la contraseña.
- **RF7:** El sistema deberá permitir al administrador modificar el estado (alta/baja) del curso.
- **RF8:** El sistema debe permitir a los usuarios registrados ver los cursos en los que se ha inscrito.
- **RF9:** El sistema deberá permitir a los usuarios la inscripción en un determinado curso, aunque éste tenga el número máximo de inscripciones.
- **RF10:** El sistema deberá crear una lista de espera para los cursos que completen su número máximo de inscripciones.
- **RF11:** El sistema deberá dar de baja a un curso cuando éste no llegue al número mínimo de inscripciones.
- **RF12:** El sistema mostrará únicamente los cursos de extensión disponibles según la fecha.
- **RF13:** El sistema deberá procesar la solicitud de reserva de aulas, según los recursos que se necesiten para impartir el curso.
- **RF14:** El sistema deberá mostrar los datos referentes a los cursos de extensión que se van a impartir.
- **RF15:** El sistema permitirá a los usuarios visitantes ponerse en contacto con el administrador a través de correo electrónico.
- **RF16:** El sistema permitirá al administrador la consulta del correo electrónico.

El RF07 de la práctica 1 ha sido eliminado y se han añadido los RF13, RF14, RF15, RF16.

3.2. Requisitos de Información

- **RI1:** El sistema deberá almacenar información referente a los cursos; el nombre, las fechas de inicio y fin, el número máximo de inscripciones, el ponente, recursos necesarios, precio, fecha límite de inscripción y descripción general.
- **RI2:** El sistema deberá almacenar información referente al registro de usuarios, para ello requiere el usuario de la UCO, nombre, apellidos, DNI y e-mail.
- **RI3:** El sistema deberá almacenar quién es el coordinador de un curso y la información referente al mismo.
- **RI4:** El sistema deberá almacenar información referente a las aulas, como puede ser la clase en la que se imparte el curso o la existencia o no de recursos informáticos.
- **RI5:** El sistema deberá mostrar una dirección de correo electrónico para contactar con el administrador.
- **RI6:** El sistema deberá mostrar una dirección de correo electrónico para contactar con el coordinador de cursos.
- **RI7:** El sistema deberá informar a los usuarios inscritos en un determinado curso de la anulación de este por no cumplir el mínimo de inscripciones.

3.3. Requisitos No Funcionales

- **RNF1:** El sistema debe ser capaz de garantizar la seguridad del sistema y para ello encriptará la clave.
- **RNF2:** El sistema deberá ser capaz de almacenar toda la información introducida por los usuarios registrados.
- **RNF3:** El sistema deberá realizar copias de seguridad periódicas.
- **RNF4:** El sistema no contará con interfaz de recursos para la pasarela de pagos.

4. Análisis de Requisitos

Para el análisis de requisitos, hemos creado historias de usuarios, que han quedado reflejadas en el tablero de trabajo de *Trello*, *Proyecto 305*. Las HUs creadas han seguido la siguiente estructura: como <usuario> quiero <objetivo> para <beneficio>.

Tras la realización de las HUs, hemos extraído los casos de uso (CUs).

4.1. Historias de Usuario

Podemos verlas en el tablero *Trello*, cuyo enlace es el siguiente: <https://trello.com/b/ZC32cHQO/proyecto305>.

Son las siguientes:

HU1: Crear o eliminar cursos de extensión.

Título: Como administrador quiero crear o eliminar los cursos de extensión para la visualización actualizada de los mismos.

Descripción: El administrador del sistema deberá acceder al listado de cursos vigentes para la visualización de estos en la aplicación y así permitir a los usuarios poder inscribirse en ellos. Del mismo modo, el administrador deberá eliminar de la lista de cursos aquellos que en los que la fecha de inscripción ha pasado.

Validación:

- El administrador debe tener una cuenta con los permisos requeridos para la creación de cursos.
- Poder crear el listado de cursos.

HU2: Registrar a los usuarios

Título: Como administrador quiero registrar a los usuarios para que puedan interactuar con los cursos de extensión.

Descripción: El administrador de la aplicación debe crear una cuenta para los usuarios solicitantes para que estos, una vez registrados puedan tener acceso a la inscripción en los cursos. Del mismo modo, creará usuarios para los coordinadores de cursos y coordinadores de recursos, que podrán ver la información de cada curso.

Validación:

- Creación de usuario.
- Asignación de contraseña.
- Enviar mail para confirmar el registro del usuario.

HU3: Reserva de aulas

Título: Como administrador tengo que realizar la reserva de las aulas para que el curso pueda impartirse.

Descripción: El administrador, tras recibir la solicitud por parte del coordinador de recursos, se encarga de la reserva de las aulas correspondientes a cada curso.

Validación:

- Asignación de un aula válida.
- Recibir la solicitud del coordinador de recursos.

HU4: Petición registro

Título: Como usuario visitante podrá ponerse en contacto con el administrador para solicitar un usuario.

Descripción: El usuario visitante podrá ver el listado de cursos disponibles en la aplicación y en caso de estar interesado en la inscripción de alguno de ellos, podrá ponerse en contacto con el administrador para la solicitud de un usuario.

Validación:

- Recibir el correo de la solicitud.
- Usuario disponible.
- Modificar la contraseña ofrecida por el administrador.

HU5: Ver listado de cursos

Título: Como usuario participante quiero visualizar el listado de cursos para ver en los que me he inscrito.

Descripción: El usuario participante podrá ver el listado de cursos en los que se ha inscrito, incluyendo toda la información referida a éste.

Validación:

- Ser un usuario participante registrado.

HU6: Inscripción en cursos

Título: Como usuario participante registrado quiero poder inscribirme en los cursos disponibles para acceder a su contenido.

Descripción: El usuario participante registrado podrá acceder al listado de los cursos disponibles con el fin de poder inscribirse a los que esté interesado.

Validación:

- El curso debe de estar disponible.

HU7: Información del curso

Título: Como coordinador de curso, quiero notificar información referente al curso para la inscripción de los interesados.

Descripción: El coordinador de curso, deberá informar al administrador de quién es el ponente del curso, el número máximo de participantes, la fecha de inicio y fin de este y el precio del curso.

Validación:

- El coordinador de curso debe estar registrado.
- Posibilidad de ponerse en contacto con el administrador.

HU8: Información de recursos

Título: Como coordinador de recursos quiero informar de los recursos necesarios para la correcta impartición de cada curso.

Descripción: El coordinador de recursos informará al administrador de todas las aulas disponibles con proyector, ordenador o las aulas normales para cada uno de los cursos disponibles.

Validación:

- Usuario registrado como coordinador de recursos.
- Contactar con el administrador.

4.2. Casos de Uso

Caso de Uso	Actualizar cursos de extensión
Identificador	CU01
Objetivo	El administrador creará o eliminará los cursos de extensión que soliciten los coordinadores de cursos.
Contexto	El administrador debe estar registrado y recibir una petición de modificación de un curso.
Actor principal	Usuario administrador
Escenario principal	<ol style="list-style-type: none"> 1. El administrador se identifica con su usuario y contraseña. 2. El sistema muestra una lista de cursos existentes. 3. El sistema permite la creación de un nuevo curso. 4. El sistema muestra la lista de cursos existentes actualizada.
Extensiones	2a. El sistema informa de errores

Tabla 1: CU01

Caso de Uso	Solicitud de Registro
Identificador	CU02

Objetivo	El usuario visitante mandará una solicitud de registro mediante correo electrónico al administrador.
Contexto	El usuario visitante se pondrá en contacto a través de correo electrónico con el administrador del sistema para registrar al usuario.
Actor principal	Usuario visitante
Escenario principal	1. Usuario visitante se pone en contacto con el administrador mediante correo electrónico solicitando un usuario.

Tabla 2: CU02

Caso de Uso	El administrador consulta las solicitudes
Identificador	CU03
Objetivo	El administrador podrá consultar las solicitudes recibidas mediante correo electrónico.
Contexto	El administrador consulta las peticiones de registro de los usuarios visitantes.
Actor principal	Usuario administrador
Escenario principal	1. El administrador recibe una solicitud de registro de usuario.

Tabla 3: CU03

Caso de Uso	Registrar a los usuarios
Identificador	CU04
Objetivo	El administrador registrará a los usuarios que lo hayan solicitado.
Contexto	El administrador recibe una petición de registro por correo electrónico y este crea un usuario y una contraseña para cada uno de ellos.
Actor principal	Usuario administrador
Escenario principal	1. El administrador recibe una solicitud de registro de usuario. 2. El administrador procesa la solicitud de inscripción, creando un usuario y contraseña. 3. El administrador envía un correo de confirmación al usuario. 4. El usuario cambia la contraseña.
Extensiones	2a. El sistema detecta que el usuario ya existe y el caso de uso termina

Tabla 4: CU04

Caso de Uso	Reserva de aulas
--------------------	-------------------------

Identificador	CU05
Objetivo	El administrador reservará el aula solicitada por el coordinador de recursos.
Contexto	El administrador recibe una petición de reserva de aula, y dependiendo de los recursos necesarios, se asignará un aula u otra.
Actor principal	Coordinador de recursos.
Escenario principal	<ol style="list-style-type: none"> 1. El coordinador de recursos envía una solicitud de reserva de aulas. 2. El administrador procesa la solicitud de reserva, y busca las aulas disponibles con los recursos solicitados. 3. El administrador envía un correo de confirmación al coordinador de recursos.
Extensiones	2a. El sistema informa de errores

Tabla 5: CU05

Caso de Uso	Ver listado de cursos
Identificador	CU06
Objetivo	El usuario visitante podrá acceder la lista de cursos de extensión en los que se ha inscrito.
Contexto	El usuario participante podrá acceder a ver el total de cursos de extensión en los que se ha inscrito y el contenido del curso.
Actor principal	Usuario participante.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario se loguea con su usuario y contraseña. 2. Accede al listado de cursos inscritos.

Tabla 6: CU06

Caso de Uso	Inscripción de cursos
Identificador	CU07
Objetivo	El usuario participante podrá inscribirse en los cursos de extensión en los que está interesado.
Contexto	El usuario participante podrá ver el listado en el que aparecen todos los cursos de extensión y podrá inscribirse en el que desee
Actor principal	Usuario participante
Escenario principal	<ol style="list-style-type: none"> 1. El usuario participante accede al listado de cursos existentes. 2. Podrá ver la información referente al curso. 3. Podrá inscribirse
Extensiones	

	<p>3a. Si el curso está completo, el sistema permitirá al usuario inscribirse.</p> <p>3b. Si el curso no llega al mínimo de inscripciones, se cancela la suscripción</p>
--	--

Tabla 7: CU07

Caso de Uso	Información del curso
Identificador	CU08
Objetivo	El coordinador del curso facilitará información al administrador referente a los cursos.
Contexto	El coordinador de cursos proporcionará información que considere importante referente a los cursos para que puedan ser visibles a los usuarios participantes y visitantes.
Actor principal	Coordinador de cursos.
Escenario principal	<ol style="list-style-type: none"> 1. El coordinador de cursos debe acceder con usuario y contraseña. 2. Debe ponerse en contacto con el administrador. 3. El administrador hará visible la información facilitada.

Tabla 8: CU08

Caso de Uso	Información de recursos
Identificador	CU09
Objetivo	El coordinador de recursos informará de los recursos necesarios para la correcta impartición de los cursos.
Contexto	El coordinador de recursos se pondrá en contacto con el administrador para poder indicarle los recursos necesarios para cada curso.
Actor principal	Coordinador de recursos.
Escenario principal	<ol style="list-style-type: none"> 1. El coordinador de recursos debe acceder con su usuario y contraseña. 2. Se pondrá en contacto con el administrador. 3. El administrador asignará a cada curso los recursos necesarios solicitados por el coordinador de recursos.

Tabla 9: CU09

Caso de Uso	Mostrar los cursos disponibles
Identificador	CU10
Objetivo	El usuario participante o visitante podrá acceder al listado de cursos disponibles.
Contexto	El usuario participante y/o visitante podrá acceder a ver el total de cursos de extensión y el contenido del curso según las fechas disponibles.
Actor principal	Usuario participante/visitante
Escenario principal	1. El usuario participante o visitante accede a la lista de cursos disponibles.

Tabla 10: CU10

5.- Diseño del sistema

En esta sección veremos cuál ha sido el diseño estructural que se ha realizado, mediante diagrama de clases y su especificación. También encontraremos el diseño del comportamiento, utilizando diagramas de secuencia.

5.1. Diseño estructural – diagrama de clases

Para la realización de los diagramas de clases, hemos trabajado con la herramienta *Visual Paradigm*. En este diagrama de clases de UML, podremos encontrar toda la interacción que realizan los usuarios con el sistema.

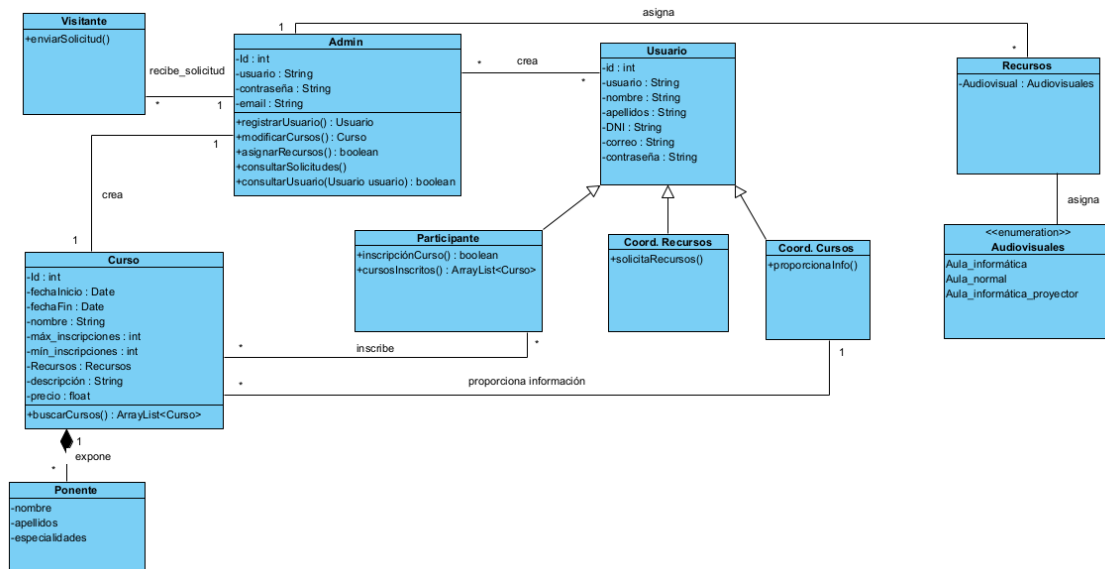


Figura 7: Diagrama de clases.

Como se puede observar en la figura 7, hemos obtenido el diagrama de clases mostrado, incluyendo la especificación de cada una de las clases.

5.2. Diseño del comportamiento – diagramas de secuencia

Para el diseño de los diagramas de secuencia, hemos utilizado la herramienta *Visual Paradigm*.

Hemos considerado oportuno dividir en dos los diagramas de secuencia, mostrando en uno de ellos el comportamiento realizado durante la acción de solicitud y registro de usuarios y el otro que hemos realizado ha sido la facilitación de la información de recursos de un determinado curso.

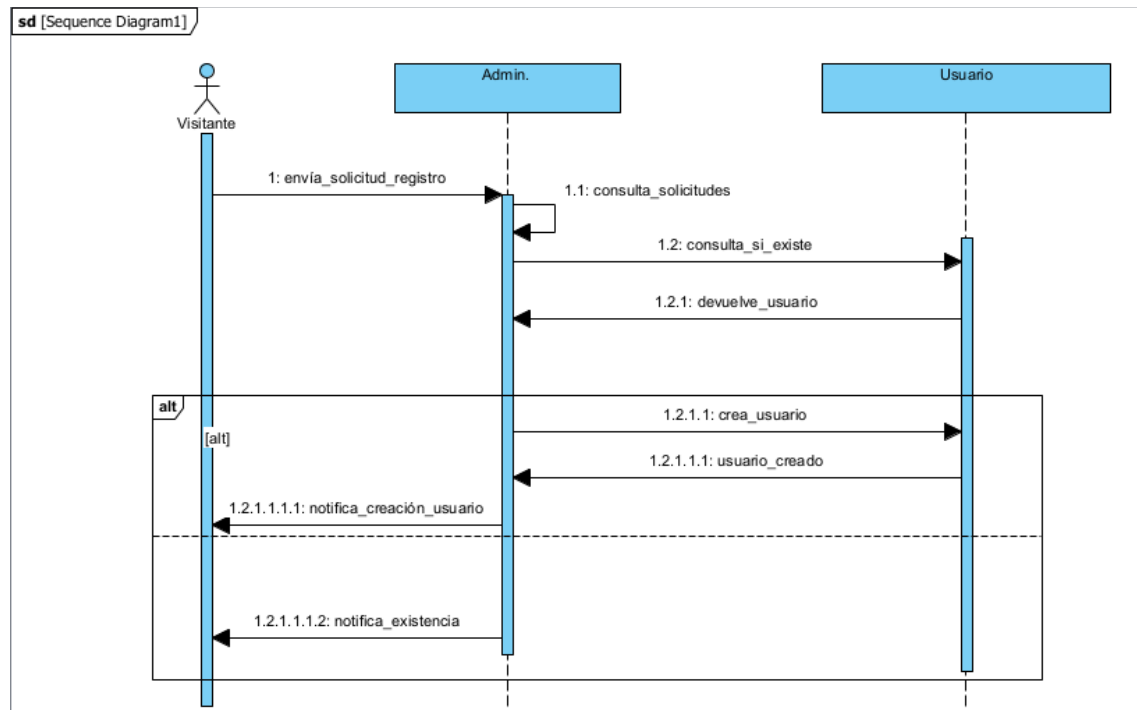


Figura 8: Diagrama de secuencia. Solicitud y registro de usuarios.

Como podemos observar en la figura 8, el usuario visitante envía la solicitud de registro al administrador, que este comprueba la solicitud recibida en su correo. Al verificar la existencia de la solicitud, manda al usuario su usuario y contraseña.

Encontramos una alternativa en caso de que el usuario haya sido creado con anterioridad.

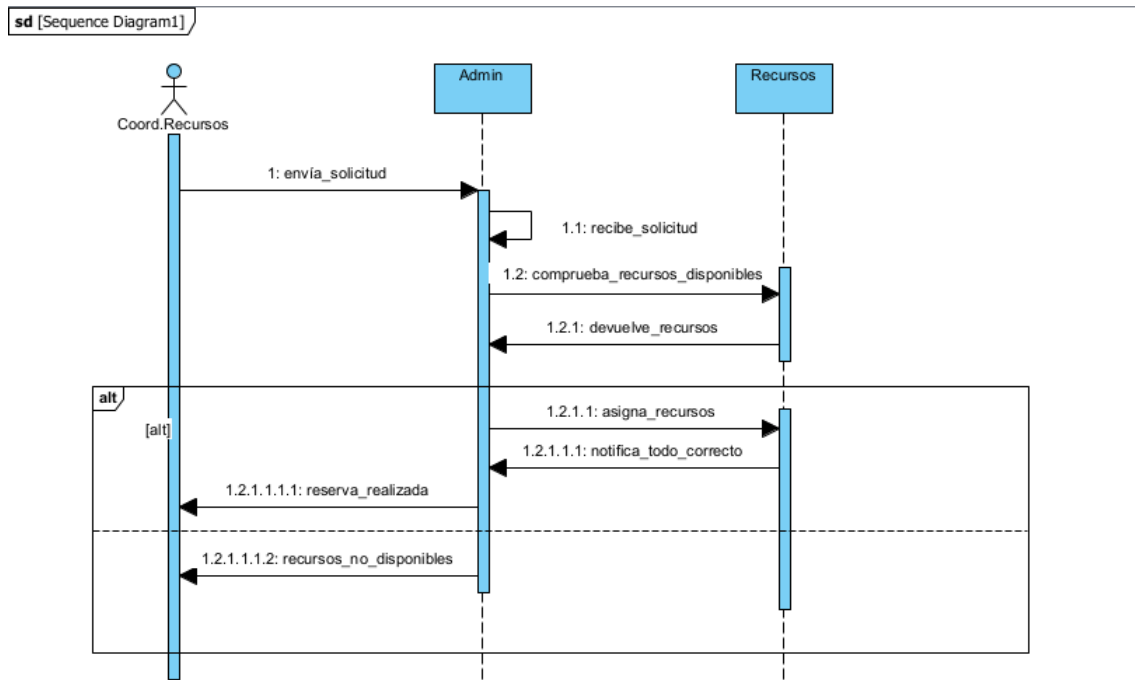


Figura 9: Diagrama de secuencia. Información de recursos para un curso.

En la figura 9, encontramos otro diagrama de secuencia. Este indica la interacción del sistema para ofrecer la información de recursos que tiene un determinado curso.

Como actor principal encontramos al coordinador de recursos, que para ponerse en contacto con el administrador envía una solicitud. El administrador comprueba en su bandeja de entrada la existencia de un correo nuevo, en caso de que exista, comprueba la disponibilidad de los recursos, asignando uno.

Se plantea una alternativa, en caso de que no haya recursos disponibles para asignar, informado al coordinador de recursos del estado.

6.- Validación del sistema

Para hacer una validación del sistema, hemos utilizado matrices de validación. En nuestro caso, hemos realizado dos, una de ellas indica la matriz de trazabilidad de RF frente a los CU. La otra enfrenta los CU con los diagramas de clases.

6.1. Matriz de trazabilidad – RF/CU

En este caso, enfrentaremos en la matriz de trazabilidad los requisitos funcionales con los casos de uso.

Fue preciso realizar modificaciones de los RF presentados en la práctica 1, ya que no estaban cubiertos o eran redundantes.

	CU01	CU02	CU03	CU04	CU05	CU06	CU07	CU08	CU09	CU10
RF01						X				
RF02							X			
RF03							X			
RF04	X									
RF05				X						
RF06				X						
RF07	X									
RF08						X				
RF09							X			
RF10							X			
RF11							X			
RF12										X
RF13									X	
RF14								X		
RF15		X								
RF16			X							

Tabla 11: Matriz de trazabilidad RF/CU.

6.2. Matriz de trazabilidad – CU/Diagrama de clases

La tabla 12 muestra la matriz de trazabilidad que enfrenta a casos de uso y diagrama de clases.

	Admin.	Visitante	Participante	Coord. Recursos	Coord. Cursos	Recursos	Curso
CU01	X						X
CU02		X					
CU03	X						
CU04	X		X	X	X		
CU05	X			X		X	
CU06			X				X
CU07			X				X
CU08					X		X
CU09	X			X		X	
CU10		X	X				X

Tabla 12: Matriz de trazabilidad CU/Diagrama de clases.

7. Implementación del sistema

Aplicando el marco de trabajo de *Scrum*, hemos llevado a cabo la implementación del sistema. Para ello, hemos decidido dividir la práctica 4 en 3 sprints diferentes, y para ello, hemos elegido varias HUs que desarrollar en cada sprint.

Las HUs han sido priorizadas por valor, es decir, hemos decidido implementar primero las HUs que hacen que el sistema sea funcionar y que genere beneficios para nuestro cliente.

Durante el sprint 1, se decidió que lo más importante era que el administrador pudiese crear el listado de los cursos de extensión, por lo que trabajamos en ella. Implementar primero esta función era la base del sistema, ya que es la funcionalidad principal de nuestro sistema.

Para el sprint 2 decidimos que la siguiente historia de usuario a implementar era que el administrador pudiese dar de alta a los usuarios visitantes, convirtiéndose en usuarios participantes y teniendo acceso al listado de los cursos creados durante el sprint 1.

En el sprint 3 implementamos la funcionalidad de que los usuarios participantes pudiesen inscribirse en los cursos de extensión existentes, siendo ésta la acción que aporte valor económico al cliente.

Para la implementación, hemos decidido desarrollar el proyecto en el entorno de *Visual Studio Code*, utilizando el lenguaje de programación *javascript*.

Hemos dividido el proyecto en dos partes, el backend que contiene toda la lógica de control de usuario, mientras que el frontend es la parte accesible al usuario y con la que interactuará.

Hemos creado una base de datos para el almacenamiento de toda la información. Para ello, se debe de crear una base de datos con el nombre *cursos_extension* en *Mysql* con el puerto por defecto: 3306.

Para poder realizar el despliegue del backend, se debe hacer un git clone desde <https://github.com/i72gocao/curso-extension-backend/tree/final-job> .

Es necesario que el dispositivo disponga de *nodejs*, que es el entorno de servidor de código abierto de javascript. Una vez realizado el clone, se debe abrir una terminal de git bash, o desde la propia terminal de *VSCode*, e insertar el comando *npm install*. Este proceso descargará todas las dependencias necesarias para que el proyecto funcione correctamente.

Para comenzar a realizar el despliegue de la aplicación, es necesario utilizar el comando *npm run watch*, haciendo el despliegue de la base de datos, generará los end-points APIs rest.

Para el despliegue del front-end, hay que hacer un git clone desde <https://github.com/i72gocao/cursos-extension-frontend/tree/final-test> .

Del mismo modo que anteriormente, hay que hacer el comando *npm install* desde *VSCode* o desde la powershell. En caso de que el sistema operativo sea Linux, se debe abrir la terminal.

Por último, se debe ejecutar el comando *npm start*, que permite visualizar la página web creada.

***NOTA IMPORTANTE:** El backend utiliza el puerto 3001, mientras que el frontend, el puerto 3000.

8. Pruebas del sistema

Para la realización de las pruebas, hemos creado 3 test.

PASS	src/App.test.js				
PASS	src/components/login/Form.test.js				
PASS	src/components/contact/Contactar.test.js				
File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	77.08	40	61.53	77.08	
src	0	100	0	0	
App.js	0	100	0	0	27-65
src/components/contact	90.9	50	66.66	90.9	
Contactar.js	90.9	50	66.66	90.9	103
src/components/header	64.28	50	50	64.28	
LinkAllUsers.js	64.28	50	50	64.28	20-34
src/components/login	100	100	100	100	
Form.js	100	100	100	100	
src/context	78.57	20	66.66	78.57	
AuthContext.js	50	0	0	50	17-21
MessageContext.js	100	50	100	100	9
src/routes	100	100	100	100	
routesConfig.js	100	100	100	100	
Test Suites: 3 passed, 3 total					
Tests: 6 passed, 6 total					
Snapshots: 0 total					
Time: 3.095 s					
Ran all test suites related to changed files.					

Figura 10: Test

Como se puede observar en la figura 10, el primer test es sobre el formulario de petición de registro. El test simula la escritura en los campos de formulario y al clicar sobre el botón *submit*, se espera que se obtengan dichos valores en los campos del formulario, y la posterior llamada al evento del envío del formulario.

El siguiente test hace lo mismo, pero sobre el botón de *login*.

Dentro de la vista de petición de registro de usuario, también se realizó un test en el que debe aparecer el encabezado del formulario.

Por último, se realizó un test sobre las rutas que debe de seguir la página. Por ejemplo, la ruta hacia el home redirige la página hacia el home y muestra el logo de la misma. También, en la ruta para hacer login, mostrando su encabezado.

Las dificultades encontradas fueron que, al utilizar la librería *Reactjs*, se hizo uso del *hooks useContexts*, que permite el envío de datos desde un nodo html a otro nodo html, sin tener que pasar por los nodos padre de cada uno. El problema fue que, al hacer los test, se debía simular que se les pasaba información a los contextos utilizados. La solución encontrada fue crear una plantilla.

9. Referencias

- [1] *Dev.* (s.f.). Obtenido de <https://dev.to/alexandprivate/smoothly-test-react-components-with-multiple-contexts-453f>
- [2] *Dev.* (s.f.). Obtenido de <https://daily-dev-tips.com/posts/testing-library-and-react-context/>
- [3] *GitHub Backend.* (s.f.). Obtenido de <https://github.com/i72gocao/curso-extension-backend>
- [4] *GitHub FrontEnd.* (s.f.). Obtenido de <https://github.com/i72gocao/cursos-extension-frontend>
- [5] *React Hooks.* (s.f.). Obtenido de <https://react-hooks-testing-library.com/usage/advanced-hooks>
- [6] *Stackoverflow.* (s.f.). Obtenido de <https://stackoverflow.com/questions/67749630/react-testing-library-testing-hooks-with-react-context>
- [7] *Stackoverflow.* (s.f.). Obtenido de <https://stackoverflow.com/questions/74399490/how-to-test-routing-logic-with-react-router-v6-and-testing-library>
- [8] *Trello.* (s.f.). Obtenido de <https://trello.com/b/ZC32cHQO/proyecto305>
- [9] *Visual Paradigm.* (s.f.). Obtenido de <https://online.visual-paradigm.com/es/diagrams/solutions/free-class-diagram-tool/>
- [10] *Wanago.* (s.f.). Obtenido de <https://wanago.io/2022/08/01/javascript-testing-rest-api-mock-service-worker/>