



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería Informática

Gestión de Formación Documentación Técnica

Presentado por Víctor Manuel Vaquero Mesa
en Universidad de Burgos — dd de enero de 2025
Tutores: D. José Ignacio Santos Martín

Índice general

Índice general.....	III
Índice de figuras	V
Índices de tablas	VI
Apéndice A- Plan de proyecto Software	8
A.1. Introducción	8
A.2. Planificación temporal.....	8
A.3. Estudio de viabilidad	17
Viabilidad económica	17
Viabilidad legal.....	18
Apéndice B- Especificación de Requisitos	20
B.1. Introducción	20
B.2. Objetivos generales	20
B.3. Catálogo de requisitos	21
B.4. Especificación de requisitos	24
Apéndice C- Especificación de diseño	44
C.1. Introducción.....	44
C.2. Diseño de datos.....	44
C.3. Diseño procedimental	48
C.4. Diseño arquitectónico	54
Apéndice D- Documentación técnica de programación.....	59
D.1. Introducción	59
D.2. Estructura de directorios	59
D.3. Manual del programador.....	61
D.4. Pruebas del sistema	65
Apéndice E- Documentación de usuario	69
E.1. Introducción	69
E.2. Requisitos de usuario	69
E.3. Instalación	69
E.4. Manual del usuario	69
Apéndice F- Anexo de sostenibilización curricular	70

F.1. Introducción	70
F.2. Competencias en sostenibilidad adquiridas	70
F.3. Impacto social y ambiental	70
Bibliografía.....	71

Índice de figuras

Figura A.1: Burndown chart del Sprint 0.....	10
Figura A.2: Burndown chart del Sprint 1.....	11
Figura A.3: Burndown chart del Sprint 2.....	12
Figura A.4: Burndown chart del Sprint 3.....	13
Figura A.5: Burndown chart del Sprint 4.....	14
Figura A.6: Burndown chart del Sprint 5.....	15
Figura B.1: Diagrama de casos de uso	25
Figura C.1: Diagrama Entidad-Relación	47
Figura C.2: Diagrama de secuencia. Registro de usuario	48
Figura C.3: Diagrama de secuencia. Gestión cursos	49
Figura C.4: Diagrama de secuencia. Gestión ediciones	50
Figura C.5: Diagrama de secuencia. Ficha formativa	51
Figura C.6: Diagrama de secuencia. Actividades formativas.....	51
Figura C.7: Diagrama de secuencia. Registro de asistencias	52
Figura C.8: Diagrama de secuencia. Certificación de ediciones.....	53
Figura C.9: Diagrama de arquitectura	57

Índices de tablas

Tabla A.1: Estimación de Story Points.....	9
Tabla A.2: Costes de personal.....	17
Tabla A.3: Costes hardware y software	17
Tabla A.4: Costes totales	17
Tabla B.1: CU-01. Registro de usuario	26
Tabla B.2: CU-02. Login	26
Tabla B.3: CU-03. Logout	27
Tabla B.4: CU-04. Crear cursos	27
Tabla B.5: CU-05. Ver cursos.....	28
Tabla B.6: CU-06. Editar un curso.....	28
Tabla B.7: CU-07. Eliminar un curso	29
Tabla B.8: CU-08. Crear ediciones asociadas	29
Tabla B.9: CU-09. Ver ediciones.	30
Tabla B.10: CU-10. Editar ediciones	30
Tabla B.11: CU-11. Eliminar una edición	31
Tabla B.12: CU-12. Crear participantes.....	31
Tabla B.13: CU-13. Ver participantes	32
Tabla B.14: CU-14. Editar participante	32
Tabla B.15: CU-15. Eliminar participante	33
Tabla B.16: CU-16. Inscribir participante	33
Tabla B.17: CU-17. Baja completa	34
Tabla B.18: CU-18. Baja justificada.....	34
Tabla B.19: CU-19. Ver inscripciones	35
Tabla B.20: CU-20. Crear formadores.....	35
Tabla B.21: CU-21. Ver formadores	36
Tabla B.22: CU-22. Editar formador	36
Tabla B.23: CU-23. Baja de un formador	37
Tabla B.24: CU-24. Asignar formador	37
Tabla B.25: CU-25. Eliminar asignación	38
Tabla B.26: CU-26. Editar datos de asignación.....	38
Tabla B.27: CU-27. Ver ficha formativa	39
Tabla B.28: CU-28. Listar ediciones disponibles	39
Tabla B.29: CU-29. Editar datos de contacto del participante	40
Tabla B.30: CU-30. Ver ediciones asignadas	40
Tabla B.31: CU-31. Gestionar sesiones y asistencia	41
Tabla B.32: CU-32. Registrar calificaciones	41
Tabla B.33: CU-33. Editar datos de contacto del formador	42
Tabla B.34: CU-34. Certificar edición	42
Tabla B.35: CU-35. Enlace en el pie de página	43
Tabla B.36: CU-36. Página de política de protección de datos	43

Apéndice A- Plan de proyecto Software

A.1. Introducción

Este apéndice documenta la planificación y análisis realizado durante el desarrollo del proyecto "Gestión de formación". Su propósito es detallar la organización temporal y el análisis de viabilidad económica y legal que sustentan la realización del proyecto.

En la sección de Planificación temporal, se describen las fases de desarrollo, las tareas llevadas a cabo, y los plazos asignados a cada una. Este apartado incluye el desglose de los sprints realizados, siguiendo la metodología ágil Scrum, y el uso de herramientas de gestión como Zube.io y GitHub.

El apartado de Estudio de viabilidad se divide en dos secciones principales:

- Viabilidad económica: estima los costes necesarios para llevar a cabo un proyecto de desarrollo de este tipo, incluyendo recursos humanos, tecnológicos y operativos.
- Viabilidad legal: describe el marco normativo y las licencias aplicables, prestando especial atención a los aspectos relacionados con la protección de datos y los derechos de uso del software desarrollado.

Este análisis permite garantizar que el proyecto es factible tanto en términos económicos como legales, y establece las bases para su despliegue en un entorno real.

A.2. Planificación temporal

Metodología.

La planificación temporal se ha basado en una metodología ágil Scrum, adaptada a un entorno de desarrollo individual. Para ello:

- Se dividió el proyecto en sprints de dos semanas de duración.
- Se utilizó Zube.io como herramienta para gestionar las tareas y realizar un seguimiento de las historias de usuario.

- Se definieron objetivos específicos para cada sprint, priorizando las funcionalidades clave del sistema.
- A cada tarea se le definió un story points como medida de complejidad o duración de la tarea con la siguiente estimación temporal por nuestra parte:

Story Points	Estimación temporal
1	2 horas de trabajo
2	4 horas de trabajo
3	1 día de trabajo
5	2 días de trabajo
8	3 días de trabajo

Tabla A.1: Estimación de Story Points

- Para tareas con mayor Story Pains se entendió que lo correcto sería dividirla en varias historias de usuario.
- Se establecieron reuniones a la finalización de cada sprint para llevar a cabo la revisión de la iteración realizada y revisar la planificación de aquellas tareas a realizar en el próximo sprint.

Herramientas de planificación.

Zube.io: se utilizó para registrar las historias de usuario y organizar las tareas por sprint.

GitHub: se encargó del control de versiones mediante ramas propias para cada issue.

Burndown charts: permitieron monitorizar el progreso diario y evaluar la carga de trabajo restante.

Cronograma.

El proyecto se desarrolló a lo largo de 14 semanas, organizadas en tres fases clave: dos semanas iniciales dedicadas a la formación y preparación del entorno, ocho semanas destinadas al desarrollo intensivo de la aplicación, y cuatro semanas finales reservadas para la creación de la documentación, la resolución de bugs reportados y la implementación de funciones adicionales que pensamos le aportan valor al proyecto.

Este cronograma puede desglosarse de la siguiente manera:

- **Sprint 0 (7/10/24 – 21/10/24):**

El sprint 0 estuvo marcado por su importante componente formativo y de preparación del entorno de desarrollo.

Tras contactar con quien sería el tutor del proyecto, D. José Ignacio Santos Martín, y concretar los objetivos y requerimientos del proyecto, se dio comienzo al sprint 0.

La descomposición de las tareas llevadas a cabo a lo largo de este sprint se puede consultar en el siguiente enlace [Sprint 0](#).

El Sprint 0 se definió con la inclusión de 12 tareas y con un total de 24 Story Points.



Figura A.1: Burndown chart del Sprint 0.

Como se puede observar en el Burndown chart, este Sprint tuvo un fuerte marcado documental y formativo.

- **Sprint 1 (21/10/24 – 4/11/24):**

En la planificación del sprint 1 se estimó que el objetivo de este sprint debía pasar por crear el núcleo de este proyecto:

- Configurar Symfony para albergar los 3 portales que integran nuestro proyecto (Gestor, Participante y Formador).
- Desarrollar la gestión de cursos y ediciones mediante la implementación de las entidades Curso y Edición junto con sus métodos CRUD asociados.

La descomposición de las tareas llevadas a cabo a lo largo de este sprint se puede consultar en el siguiente enlace [Sprint 1](#).

El Sprint 1 se definió con la inclusión de 13 tareas y con un total de 35 Story Points.



Figura A.2: Burndown chart del Sprint 1.

Resultados del Sprint.

Al concluir el Sprint 1, se cumplieron todos los objetivos establecidos. Symfony quedó configurado con una estructura inicial para los tres portales, y las funcionalidades de gestión de cursos y ediciones fueron desarrolladas en su totalidad. Este sprint sentó las bases técnicas y funcionales para los sprints posteriores, consolidando la arquitectura del sistema.

- **Sprint 2 (4/11/24 – 18/11/24):**

En la planificación del sprint 2 se estimó que nuestro objetivo en este sprint debía ser la creación de las entidades y sus métodos necesarios para el avance en el desarrollo del Portal del Gestor: Participantes y Formadores:

- Implementar la entidad Participante con sus métodos CRUD y la entidad participante_edicion, que hará de tabla intermedia, para almacenar los datos de las inscripciones del participante en una edición concreta.
- Implementar la entidad Formador con sus métodos CRUD y la entidad formador_edicion, que hará de tabla intermedia, para almacenar los datos de las **asignaciones** del formador a una edición concreta.

La descomposición de las tareas llevadas a cabo a lo largo de este sprint se puede consultar en el siguiente enlace [Sprint 2](#).

El Sprint 2 se definió con la inclusión de 17 tareas y con un total de 41 Story Points.



Figura A.3: Burndown chart del Sprint 2.

Resultados del Sprint.

Al concluir el Sprint 2, se alcanzaron los objetivos establecidos para esta iteración. Se implementaron las entidades Participante y Formador junto con sus métodos CRUD, proporcionando una base sólida para la gestión de usuarios en el Portal del Gestor.

- **Sprint 3 (18/11/24 – 2/12/24):**

En la planificación del sprint 3 se estimó una vez terminado el núcleo del desarrollo del Portal de Gestor, que nuestro objetivo en este sprint debía ser:

- Implementar un sistema de autenticación en la plataforma, que permita el registro de un usuario (Participante) en la aplicación.
- La creación del Portal del Participante con sus distintas funcionalidades que le dan sentido a dicho portal (visualización de datos, ficha formativa del participante, actividades formativas, e inscripción en ediciones).
- La gestión de controles de acceso a los portales según los roles establecidos.

La descomposición de las tareas llevadas a cabo a lo largo de este sprint se puede consultar en el siguiente enlace [Sprint 3](#).

El Sprint 3 se definió con la inclusión de 6 tareas y con un total de 48 Story Points.



Figura A.4: Burndown chart del Sprint 3.

Resultados del Sprint.

Al concluir el Sprint 3, se lograron cumplir los objetivos establecidos, implementando funcionalidades clave para el sistema. Se desarrolló el sistema de autenticación de usuarios, permitiendo el registro y acceso seguro a la plataforma. Esta funcionalidad marcó un hito importante en el proyecto, al habilitar la gestión de usuarios y el control de acceso basado en roles.

- **Sprint 4 (2/12/24 – 16/12/24):**

En la planificación del Sprint 4, se estableció como objetivo principal cerrar el desarrollo del proyecto mediante la implementación del Portal del Formador, completando así la tercera y última sección funcional de la aplicación. Las tareas definidas para este sprint se enfocaron en los siguientes puntos:

- Implementar el Portal del Formador, incluyendo funcionalidades como la visualización de datos personales y la gestión de cursos asignados.
- Establecer controles de acceso al portal, asegurando que solo los usuarios con el rol de formador puedan acceder a esta sección.
- Diseñar y desarrollar la gestión de sesiones, asistencia y calificaciones para ediciones abiertas asignadas a los formadores.
- Implementar el envío de datos de asistencia y calificaciones para la certificación y cierre de ediciones.
- Completar el flujo de certificación, asignando el número de título a cada participante de la edición cerrada.

La descomposición de las tareas llevadas a cabo a lo largo de este sprint se puede consultar en el siguiente enlace [Sprint 4](#).

El Sprint 4 se definió con la inclusión de 11 tareas y con un total de 51 Story Points.



Figura A.5: Burndown chart del Sprint 4.

- **Sprint 5 (16/12/24 – 30/12/24):**

En la planificación del Sprint 5, se estableció como objetivo principal la creación de la memoria del proyecto, un componente esencial de la documentación a presentar junto con el desarrollo del programa. Este sprint marcó la fase final del proyecto, centrada en consolidar tanto los aspectos técnicos como los documentales.

Durante este sprint, también se llevaron a cabo actividades clave para completar el desarrollo de la aplicación. Se identificaron y resolvieron varios bugs en la aplicación. Además, se instaló y configuró el equipamiento necesario para el despliegue de la aplicación y su puesta en producción.

Paralelamente, se implementaron dos nuevas funcionalidades en el Portal del Gestor:

- Consultar la ficha formativa de un participante en el Portal del Gestor.
- Consultar ediciones asignadas a un formador en el Portal del Gestor.

La descomposición de las tareas llevadas a cabo a lo largo de este sprint se puede consultar en el siguiente enlace [Sprint 5](#).

El Sprint 5 se definió con la inclusión de 14 tareas y con un total de 53 Story Points.

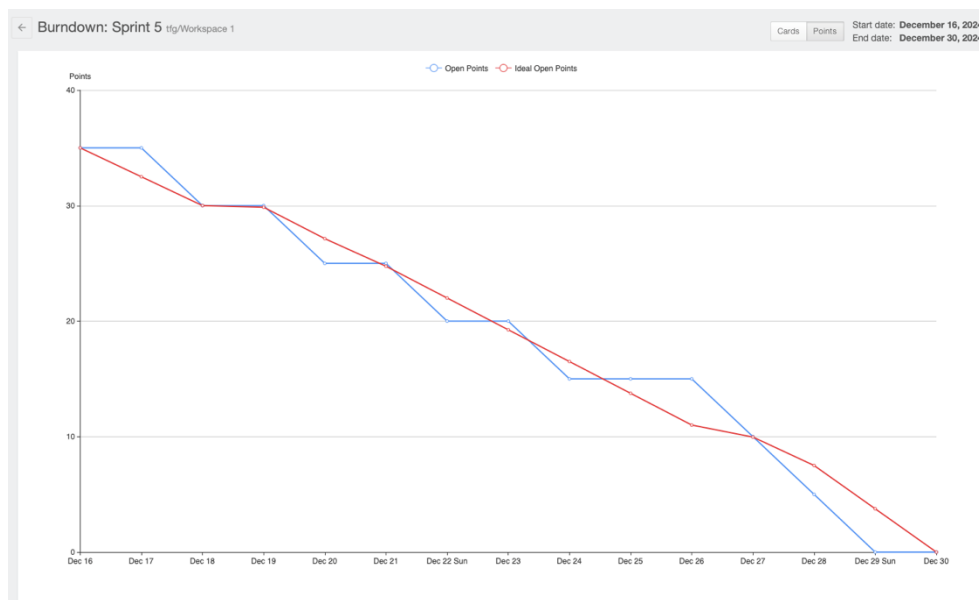


Figura A.6: Burndown chart del Sprint 5.

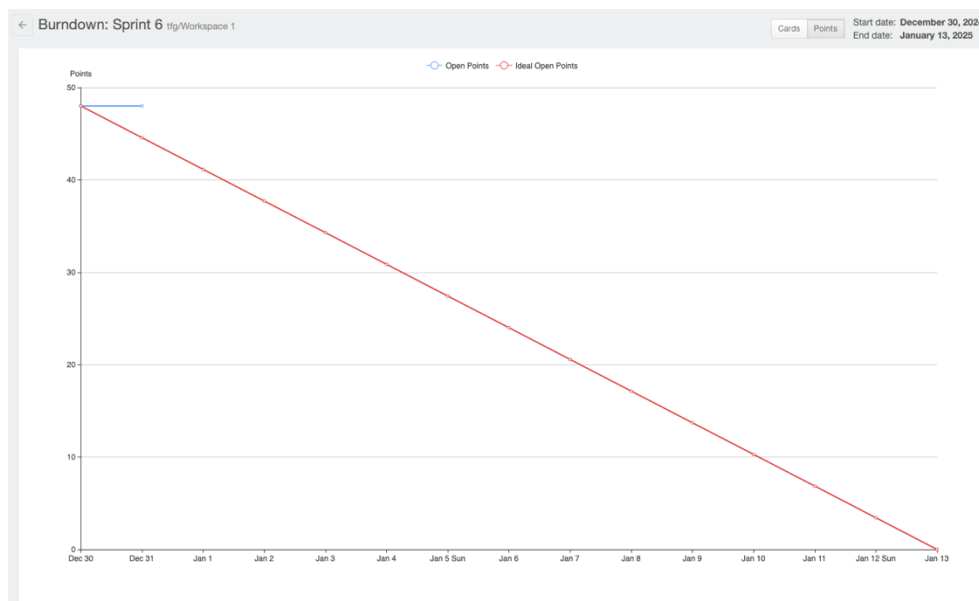
- **Sprint 6 (30/12/24 – 14/01/24):**

En la planificación del Sprint 6, se estableció como objetivo principal la creación de la documentación técnica del proyecto, un componente esencial y complementario a la memoria ya desarrollada. Este sprint marcó la conclusión definitiva del proyecto, previo a su entrega al tribunal evaluador.

Durante esta fase final, además de la documentación técnica, se elaboraron los videos de presentación y demostración del proyecto, con el objetivo de facilitar su comprensión y exhibir las principales funcionalidades del sistema.

La descomposición de las tareas llevadas a cabo a lo largo de este sprint se puede consultar en el siguiente enlace [Sprint 6](#).

El Sprint 6 se definió con la inclusión de 6 tareas y con un total de 48 Story Points.



A.3. Estudio de viabilidad

Viabilidad económica

En este apartado se analizan los costes y beneficios que podría haber supuesto el proyecto si se hubiera desarrollado en un marco empresarial real. A continuación, se presenta la estructura de costes del proyecto, desglosada en las siguientes categorías:

- **Costes de Personal**

Para el desarrollo del proyecto, se contó con un único trabajador a tiempo completo durante tres meses (de octubre de 2024 a enero de 2025). El desglose de costes es el siguiente:

Concepto	Coste mensual	Total 3 meses (€)
Salario bruto	2.000,00€	6.000,00€
Cuota Seguridad Social	598,00€	1.794,00€
Total mensual	2.598,00€	7.794,00€

Tabla A.2: Costes de personal

Cálculo de la Seguridad Social: La cuota de la Seguridad Social a cargo del empleador se calcula según la fórmula:

$$\text{Cuota SS} = \text{Sueldo bruto mensual} \times (0,236 + 0,055 + 0,002 + 0,006)$$

- **Costes de hardware y software**

A continuación, se detalla el coste del hardware y software utilizado, considerando una amortización a 6 años y un periodo de uso de 3 meses:

Concepto	Coste	Coste amortizado
Ordenador	900€	37,5€
PHPStorm	11,98€ mensual	35,94€

Tabla A.3: Costes hardware y software

- **Costes totales**

El coste total del proyecto se presenta a continuación:

Concepto	Coste
Personal	7.794,00€
Hardware	37,5€
Software	35,94€
TOTAL	7.573,44€

Tabla A.4: Costes totales

- **Beneficios**

Aunque el proyecto se ha desarrollado con fines académicos, su finalidad es ser implementado y utilizado en este mismo entorno, concretamente en la institución donde trabajo (Universidad de Sevilla). Por este motivo, no se esperan beneficios económicos directos de su explotación.

Viabilidad legal

El presente proyecto ha sido desarrollado teniendo en cuenta las normativas y requisitos legales aplicables, tanto en el ámbito de la protección de datos como en el uso de herramientas y software. A continuación, se detallan los aspectos legales más relevantes:

- **Protección de datos personales**

Dado que el proyecto incluye funcionalidades relacionadas con la gestión de usuarios (participantes y formadores), se ha tenido en cuenta el Reglamento General de Protección de Datos (RGPD), en vigor en la Unión Europea. Para garantizar el cumplimiento de esta normativa, se han implementado las siguientes medidas:

Consentimiento explícito: Los usuarios deben otorgar su consentimiento explícito para el tratamiento de sus datos personales durante el registro.

Finalidad específica: Los datos recopilados solo se utilizan para los fines indicados en la plataforma, como la gestión de inscripciones, calificaciones y certificados.

Derechos del usuario: Se garantiza que los usuarios puedan ejercer sus derechos de acceso, rectificación, supresión y oposición de sus datos personales.

- **Licencias de software**

El desarrollo del proyecto ha implicado el uso de herramientas y librerías bajo licencias específicas, principalmente MIT y BSD-3-Clause. A continuación, se detalla el tipo de licencia de las herramientas principales utilizadas:

Symfony Framework: Licencia MIT, que permite su uso, modificación y redistribución de forma libre.

PHPStan y PHPUnit: Licencia MIT y BSD-3-Clause, utilizadas para garantizar la calidad del código y realizar pruebas automatizadas.

PHPStorm: Licencia de software propietario, con una suscripción válida durante el periodo de desarrollo.

MySQL: Licencia GPL, que permite su uso gratuito en proyectos de código abierto.

Bootstrap (frontend): Licencia MIT, que permite la personalización del diseño del proyecto.

Todas las herramientas y librerías empleadas cumplen con las condiciones de uso establecidas por sus respectivos licenciarios, asegurando el respeto a los derechos de propiedad intelectual.

- **Propiedad intelectual**

El código desarrollado para este proyecto, así como la documentación técnica y la memoria asociada, son propiedad del autor. Se ha decidido aplicar una licencia abierta (MIT), que permite:

El uso, modificación y redistribución del código siempre que se conserve el aviso de copyright original y se atribuya al autor.

- **Normativa aplicable**

Además de las normativas mencionadas, se ha tenido en cuenta la legislación española y europea en materia de:

Protección de datos personales (RGPD y Ley Orgánica 3/2018 de Protección de Datos Personales y Garantía de los Derechos Digitales).

Derechos de propiedad intelectual: Asegurando la originalidad del contenido desarrollado y el respeto por los derechos de terceros.

Apéndice B- Especificación de Requisitos

B.1. Introducción

En este apéndice se presenta la Especificación de Requisitos del proyecto “Gestión de formación”. Su propósito es describir de manera clara y detallada las funcionalidades que el sistema debe cumplir, así como las restricciones y características técnicas que lo enmarcan.

Este documento tiene como objetivo servir de guía base durante todo el ciclo de vida del desarrollo, garantizando que las funcionalidades implementadas satisfagan las necesidades definidas.

B.2. Objetivos generales

Los objetivos generales de la aplicación “Gestión de formación” son:

Automatizar el proceso de gestión de la formación. Permitir a los gestores administrar cursos, ediciones, participantes y formadores de forma centralizada, reduciendo tareas manuales y el uso de múltiples herramientas.

Facilitar la interacción con los usuarios (participantes y formadores). Ofrecer un entorno web amigable y seguro para la consulta de su información personal, el acceso a las calificaciones, la inscripción en ediciones y la gestión de la asistencia.

Garantizar la trazabilidad de la información. Mantener un registro histórico de todos los cursos, ediciones, participantes y formadores, que permita su consulta y seguimiento.

Proporcionar mecanismos de evaluación y certificación. Ofrecer funcionalidades para gestionar la evaluación de la formación impartida, emitir certificados digitales y conservar la información de manera segura.

Estos objetivos se alinean con la necesidad de modernizar y agilizar el proceso formativo, contribuyendo a una mejor experiencia de todos los actores implicados.

B.3. Catálogo de requisitos

Los siguientes requisitos se derivan de los objetivos generales anteriormente dispuestos:

Requisitos funcionales

- **RF-1 Registro de usuario:** la aplicación debe permitir al usuario crear una cuenta con email y contraseña.
 - **RF-1.1 Login:** el usuario debe poder autenticarse introduciendo su email y contraseña.
 - **RF-1.2 Logout:** el usuario debe poder cerrar la sesión de forma segura.
- **RF-2 Gestión de la política de protección de datos:** la aplicación debe facilitar al usuario el acceso y la aceptación de la política de protección de datos, cumpliendo con los requisitos legales.
 - **RF-2.1 Aceptación del consentimiento en el registro:** la aplicación debe mostrar al usuario un checkbox obligatorio para que acepte la política de protección de datos al crear su cuenta.
 - **RF-2.2 Enlace en el pie de página:** la aplicación debe incluir un enlace a la política de protección de datos en el footer de todas las páginas del sitio.
 - **RF-2.3 Página de política de protección de datos:** la aplicación debe disponer de una página dedicada (por ej. /politica-datos) que muestre el contenido completo de la política de protección de datos.
- **RF-3 Portal del Gestor:** la aplicación debe proveer interfaces específicas según el rol.
 - **RF-3.1 Gestión de cursos:** la aplicación debe permitir gestionar los cursos de formación.
 - **RF-3.1.1 Crear cursos:** la aplicación debe permitir al gestor dar de alta nuevos cursos con información básica (código, nombre, descripción...).
 - **RF-3.1.2 Ver cursos:** el gestor podrá listar los cursos existentes y ver el detalle de cada curso (objetivos, duración, etc.).

- **RF-3.1.3 Editar un curso:** el gestor debe poder modificar los datos de un curso existente.
- **RF-3.1.4 Eliminar un curso:** el gestor debe poder eliminar un curso si no se necesita.
- **RF-3.2 Gestión de ediciones:** la aplicación debe permitir gestionar las ediciones asociadas.
 - **RF-3.2.1 Crear ediciones asociadas:** la aplicación debe permitir al gestor crear ediciones de un curso, definiendo fechas, plazas, modalidad, etc.
 - **RF-3.2.2 Ver ediciones:** el gestor podrá listar las ediciones asociadas a un curso y ver el detalle de cada edición.
 - **RF-3.2.3 Editar ediciones:** el gestor podrá modificar una edición existente.
 - **RF-3.2.4 Eliminar ediciones:** el gestor podrá eliminar una edición que ya no sea necesaria.
- **RF-3.3 Gestión de participantes:** la aplicación debe permitir administrar los participantes.
 - **RF-3.3.1 Crear participantes:** la aplicación debe permitir dar de alta un nuevo participante en la base de datos (datos personales, centro, etc.).
 - **RF-3.3.2 Ver participantes:** el gestor debe poder ver el listado de participantes dados de alta y los datos de cada uno.
 - **RF-3.3.3 Editar participante:** el gestor debe poder actualizar los datos de un participante.
 - **RF-3.3.4 Eliminar participante:** el gestor debe poder eliminar un participante para que no pueda inscribirse en los cursos.
- **RF-3.4 Gestión de inscripciones:** la aplicación debe permitir administrar las inscripciones de los participantes a las ediciones.
 - **RF-3.4.1 Inscribir participante:** el gestor debe poder inscribir manualmente a un participante en una edición.
 - **RF-3.4.2 Baja completa de una edición:** el gestor debe poder desinscribir a un participante de una edición (antes de iniciar la edición).
 - **RF-3.4.3 Baja justificada de una edición:** el gestor debe poder asignar una baja justificada a un participante en una edición si ha pasado la fecha de inicio.

- **RF-3.4.4 Ver inscripciones en una edición:** la aplicación debe permitir listar todos los inscritos en una determinada edición y los datos de su inscripción.
- **RF-3.5 Gestión de formadores:** la aplicación debe permitir administrar los formadores.
 - **RF-3.5.1 Crear formadores:** la aplicación debe permitir dar de alta un nuevo formador (datos de contacto, organización, etc.).
 - **RF-3.5.2 Ver formadores:** el gestor debe poder ver el listado de formadores dados de alta y los datos de cada uno.
 - **RF-3.5.3 Editar formador:** el gestor debe poder actualizar los datos de un formador.
 - **RF-3.5.4 Baja de formador:** el gestor debe poder dar de baja a un formador que ya no colaborará.
- **RF-3.6 Gestión de asignaciones:** la aplicación debe permitir administrar las asignaciones de los formadores a las ediciones.
 - **RF-3.6.1 Asignar formador a una edición:** el gestor debe poder vincular un formador a una edición concreta.
 - **RF-3.6.2 Eliminar asignación de formador:** el gestor debe poder eliminar dicha asignación si es necesario.
 - **RF-3.6.3 Editar datos de asignación:** el gestor debe poder modificar la retribución, horas impartidas, etc.
- **RF-4 Portal del Participante**
 - **RF-4.1 Ver ficha formativa:** el participante debe poder ver las ediciones en la que está inscrito (próximas, certificadas, etc.).
 - **RF-4.2 Listar actividades y ediciones disponibles:** el participante ve cursos/ediciones y debe poder inscribirse.
 - **RF-4.3 Mis datos de contacto:** el participante debe poder consultar o editar su información.
- **RF-5 Portal del Formador**
 - **RF-5.1 Ver ediciones asignadas:** el formador debe poder consultar las ediciones que tiene asignadas y están abiertas, y cuales están cerradas.

- **RF-5.2 Gestionar sesiones y asistencia:** el formador debe poder crear sesiones y registrar asistencias en las ediciones que tiene asignadas y estén abiertas.
- **RF-5.3 Registrar calificaciones:** el formador, si la edición es calificable y está abierta, debe poder asignar calificación.
- **RF-5.4 Editar mis datos de contacto:** el formador debe poder actualizar su información personal.
- **RF-6 Certificación y cierre de ediciones:** la aplicación debe permitir la certificación de participantes y el cierre de ediciones, validando asistencia, calificaciones, etc.

Requisitos no funcionales

- **RNF-1 Usabilidad:** la aplicación debe ser intuitiva, con una curva de aprendizaje baja, y ofrecer mensajes de error claros para el usuario.
- **RNF-2 Compatibilidad y disponibilidad:** la aplicación debe funcionar correctamente en los navegadores modernos (Chrome, Firefox, Edge) y, en la medida de lo posible, en versiones recientes de dispositivos móviles.
- **RNF-3 Rendimiento:** el sistema debe ofrecer tiempos de respuesta aceptables en las operaciones críticas. Se recomienda que las pantallas no queden bloqueadas durante más de 2-3 segundos en condiciones normales de carga.
- **RNF-4 Seguridad y privacidad:** todos los datos sensibles (contraseñas, tokens de acceso, datos personales, etc.) deben almacenarse y transmitirse de forma cifrada. Además, la aplicación deberá cumplir con las leyes de protección de datos vigentes (RGPD, LOPDGDD), asegurando la privacidad del usuario.
- **RNF-5 Escalabilidad:** el diseño del sistema debe permitir la incorporación de nuevas funcionalidades y la ampliación de la base de datos sin requerir cambios drásticos en la arquitectura.
- **RNF-6 Mantenibilidad:** el proyecto debe tener un código estructurado y documentado para facilitar el mantenimiento y las mejoras futuras.

B.4. Especificación de requisitos

En esta sección se muestra el diagrama de casos de uso y el detalle de cada uno de ellos.

Actores

Los actores que interactuarán con el sistema son los siguientes:

Visitante: Actor que aún no ha iniciado sesión en el sistema.

Participante: Actor que ha iniciado sesión en el sistema y tiene asignado el rol "ROLE_USER".

Gestor: Actor que ha iniciado sesión en el sistema y tiene asignado el rol "ROLE_ADMIN".

Formador: Actor que ha iniciado sesión en el sistema y tiene asignado el rol "ROLE_TEACHER".

Diagrama de casos de uso

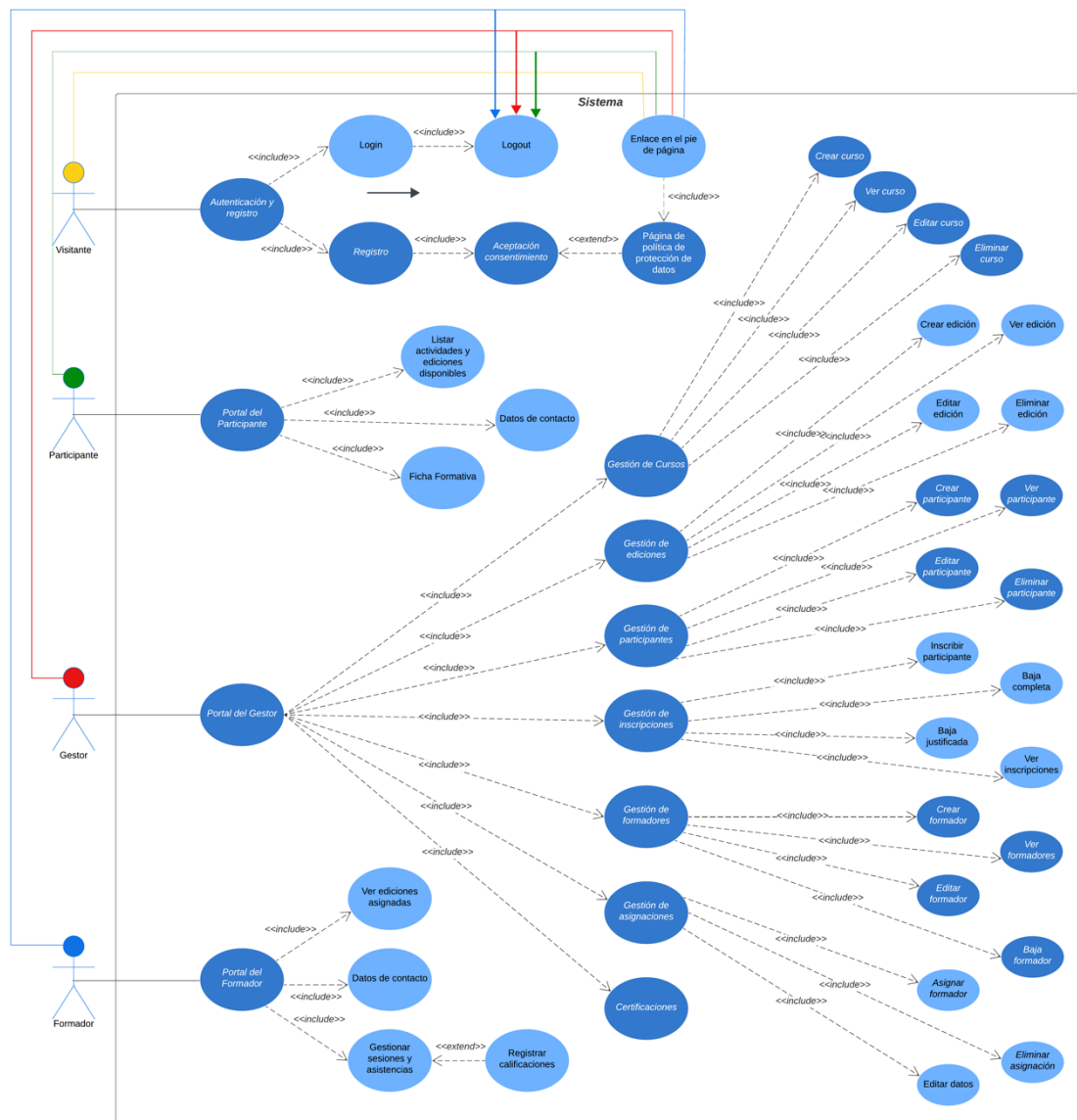


Figura B.1: Diagrama de casos de uso

Casos de Uso

CU-01	Registro de usuario
Actor	Usuario
Requisitos asociados	RF-1, RF-2.1, RF-2.3
Descripción	Permite al usuario crear una cuenta con nombre de usuario y contraseña.
Precondiciones	El usuario no tiene una cuenta existente en el sistema.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la página de registro. 2. Completa el formulario con sus datos. 3. Acepta los términos y condiciones. 4. Confirma el registro.
Postcondición	Se crea una cuenta asociada al username del usuario.
Excepciones	<p>El email ya está registrado (mensaje).</p> <p>El usuario de acceso ya está registrado (mensaje).</p> <p>El nif ya está registrado (mensaje).</p> <p>El campo Acepto el tratamiento de mis datos personales no está seleccionado(mensaje)</p>
Importancia	Alta.

Tabla B.1: CU-01. Registro de usuario

CU-02	Login
Actor	Usuario
Requisitos asociados	RF-1.1
Descripción	Permite al usuario autenticarse en el sistema introduciendo su nombre de usuario y contraseña.
Precondiciones	El usuario debe tener una cuenta activa en el sistema.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de sesión. 2. Introduce su nombre de usuario y contraseña. 3. Confirma el inicio de sesión.
Postcondición	El usuario accede a su portal según su rol.
Excepciones	Credenciales incorrectas (mensaje).
Importancia	Alta.

Tabla B.2: CU-02. Login

CU-03	Logout
Actor	Gestor, Participante, Formador
Requisitos asociados	RF-1.2
Descripción	Permite al usuario autenticado cerrar su sesión de manera segura.
Precondiciones	El usuario debe estar autenticado en el sistema y tener uno de los roles establecidos (gestor, participante o formador).
Acciones	1. El usuario selecciona la opción de cerrar sesión desde el menú.
Postcondición	Se cierra la sesión y se redirige a la página de login.
Excepciones	
Importancia	Media.

Tabla B.3: CU-03. Logout

CU-04	Crear cursos
Actor	Gestor
Requisitos asociados	RF-3.1.1
Descripción	Permite al gestor crear un nuevo curso en el sistema, proporcionando la información requerida.
Precondiciones	El gestor debe estar autenticado.
Acciones	1. El gestor accede al portal del gestor y selecciona "Crear curso". 2. Completa el formulario con los datos del curso (código, nombre, descripción, etc.). 3. Confirma la acción de crear el curso.
Postcondición	El curso es creado y registrado en el sistema.
Excepciones	El código ya está en uso (mensaje). Campos obligatorios sin rellenar (mensaje). Horas totales y horas virtuales negativas (mensaje). Horas virtuales mayores que horas totales (mensaje).
Importancia	Alta.

Tabla B.4: CU-04. Crear cursos

CU-05	Ver cursos
Actor	Gestor
Requisitos asociados	RF-3.1.2
Descripción	Permite al gestor visualizar los detalles de un curso.
Precondiciones	El gestor debe estar autenticado. Deben existir cursos en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona "Gestionar cursos". 2. Elige un curso de la lista para ver sus detalles.
Postcondición	Se muestra la información detallada del curso seleccionado.
Excepciones	No hay cursos registrados en el año seleccionado (mensaje).
Importancia	Media.

Tabla B.5: CU-05. Ver cursos

CU-06	Editar un curso
Actor	Gestor
Requisitos asociados	RF-3.1.3
Descripción	Permite al gestor modificar los datos de un curso existente
Precondiciones	El gestor debe estar autenticado. Debe existir al menos un curso registrado en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar cursos". 2. Elige un curso de la lista. 3. Realiza los cambios necesarios en el formulario. 4. Confirma la edición.
Postcondición	Los cambios realizados se guardan y el curso se actualiza.
Excepciones	Campos obligatorios sin rellenar (mensaje). Horas totales y horas virtuales negativas (mensaje). Horas virtuales mayores que horas totales (mensaje).
Importancia	Media.

Tabla B.6: CU-06. Editar un curso

CU-07	Eliminar un curso
Actor	Gestor
Requisitos asociados	RF-3.1.4
Descripción	Permite al gestor eliminar un curso del sistema.
Precondiciones	El gestor debe estar autenticado. Debe existir al menos un curso registrado en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar cursos". 2. Elige un curso de la lista. 3. Confirma la acción de eliminar el curso.
Postcondición	El curso seleccionado es eliminado del sistema.
Excepciones	El curso tiene ediciones asociadas y no puede ser eliminado (mensaje)
Importancia	Alta.

Tabla B.7: CU-07. Eliminar un curso

CU-08	Crear ediciones asociadas
Actor	Gestor
Requisitos asociados	RF-3.2.1
Descripción	Permite al gestor crear una edición asociada a un curso.
Precondiciones	El gestor debe estar autenticado. Debe existir al menos un curso registrado en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar cursos". 2. Elige un curso de la lista. 3. Completa el formulario con los datos de la edición (fechas, plazas, modalidad, etc.). 4. Confirma la creación de la edición.
Postcondición	La edición se crea y se asocia al curso seleccionado.
Excepciones	Campos obligatorios sin rellenar (mensaje). Fecha de inicio posterior a la fecha de fin (mensaje).
Importancia	Alta.

Tabla B.8: CU-08. Crear ediciones asociadas

CU-09	Ver ediciones
Actor	Gestor
Requisitos asociados	RF-3.2.2
Descripción	Permite al gestor visualizar los detalles de una edición.
Precondiciones	El gestor debe estar autenticado. Deben existir ediciones registradas en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona "Gestionar ediciones". 2. Elige una edición de la lista para ver sus detalles.
Postcondición	Se muestra la información detallada de la edición seleccionada.
Excepciones	No hay ediciones registradas (mensaje)
Importancia	Media.

Tabla B.9: CU-09. Ver ediciones.

CU-10	Editar ediciones
Actor	Gestor
Requisitos asociados	RF-3.2.3
Descripción	Permite al gestor modificar los datos de una edición existente.
Precondiciones	El gestor debe estar autenticado. Debe existir al menos una edición registrada en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar ediciones". 2. Elige una edición de la lista. 3. Realiza los cambios necesarios en el formulario. 4. Confirma la edición.
Postcondición	Los cambios realizados se guardan y la edición se actualiza.
Excepciones	Campos obligatorios sin rellenar (mensaje). Fecha de inicio posterior a la fecha de fin (mensaje).
Importancia	Media.

Tabla B.10: CU-10. Editar ediciones

CU-11	Eliminar una edición
Actor	Gestor
Requisitos asociados	RF-3.2.4
Descripción	Permite al gestor eliminar una edición del sistema.
Precondiciones	El gestor debe estar autenticado. Debe existir al menos una edición registrada en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar ediciones". 2. Elige una edición de la lista. 3. Confirma la acción de eliminar la edición.
Postcondición	La edición seleccionada es eliminada del sistema.
Excepciones	La edición tiene inscripciones asociadas y no puede ser eliminada (mensaje)
Importancia	Alta.

Tabla B.11: CU-11. Eliminar una edición

CU-12	Crear participantes
Actor	Gestor
Requisitos asociados	RF-3.3.1
Descripción	Permite al gestor dar de alta un nuevo participante en la base de datos.
Precondiciones	El gestor debe estar autenticado. Debe existir al menos un formador registrado en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona "Gestionar participantes". 2. En el listado de formadores existentes, selecciona cual quiere dar de alta como participante.
Postcondición	El participante se registra en la base de datos.
Excepciones	El formador seleccionado ya tiene un perfil de participante (mensaje).
Importancia	Alta.

Tabla B.12: CU-12. Crear participantes

CU-13	Ver participantes
Actor	Gestor
Requisitos asociados	RF-3.3.2
Descripción	Permite al gestor visualizar los detalles de un participante registrado.
Precondiciones	El gestor debe estar autenticado. Deben existir participantes registrados en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar participantes". 2. Elige un participante de la lista para ver sus detalles.
Postcondición	Se muestra la información detallada del participante seleccionado.
Excepciones	No hay participantes registrados (mensaje)
Importancia	Media.

Tabla B.13: CU-13. Ver participantes

CU-14	Editar participante
Actor	Gestor
Requisitos asociados	RF-3.3.3
Descripción	Permite al gestor modificar los datos de un participante existente.
Precondiciones	El gestor debe estar autenticado. Debe existir al menos un participante registrado en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar participantes". 2. Elige un participante de la lista. 3. Realiza los cambios necesarios en el formulario. 4. Confirma la edición.
Postcondición	Los cambios realizados se guardan y el participante se actualiza
Excepciones	Campos obligatorios sin rellenar (mensaje).
Importancia	Media.

Tabla B.14: CU-14. Editar participante

CU-15	Eliminar participante
Actor	Gestor
Requisitos asociados	RF-3.3.4
Descripción	Permite al gestor eliminar un participante del sistema.
Precondiciones	El gestor debe estar autenticado. Debe existir al menos un participante registrado en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar participante". 2. Elige un participante de la lista. 3. Confirma la acción de eliminar al participante.
Postcondición	El participante seleccionado es eliminado del sistema.
Excepciones	El participante tiene inscripciones asociadas y no puede ser eliminado (mensaje).
Importancia	Alta.

Tabla B.15: CU-15. Eliminar participante

CU-16	Inscribir participante
Actor	Gestor
Requisitos asociados	RF-3.4.1
Descripción	Permite al gestor inscribir manualmente a un participante en una edición.
Precondiciones	El gestor debe estar autenticado. Deben existir participantes, ediciones registradas y participantes que no estén ya inscritos en ese curso.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar ediciones". 2. Elige una edición de la lista. 3. Selecciona un participante. 4. Confirma la inscripción.
Postcondición	El participante queda inscrito en la edición seleccionada.
Excepciones	
Importancia	Alta.

Tabla B.16: CU-16. Inscribir participante

CU-17	Baja completa
Actor	Gestor
Requisitos asociados	RF-3.4.2
Descripción	Permite al gestor desinscribir completamente a un participante de una edición antes de que inicie.
Precondiciones	El gestor debe estar autenticado. El participante debe estar inscrito en la edición. La edición no debe haber iniciado.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar ediciones". 2. Elige una edición de la lista. 3. Selecciona un participante. 4. Confirma la acción.
Postcondición	El participante es desinscrito de la edición seleccionada.
Excepciones	La edición ya ha iniciado (mensaje)
Importancia	Media.

Tabla B.17: CU-17. Baja completa

CU-18	Baja justificada
Actor	Gestor
Requisitos asociados	RF-3.4.3
Descripción	Permite al gestor asignar una baja justificada a un participante en una edición después de que esta haya comenzado.
Precondiciones	El gestor debe estar autenticado. El participante debe estar inscrito en la edición. La edición debe haber iniciado.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar ediciones". 2. Elige una edición de la lista. 3. Selecciona un participante. 4. Introduce la fecha en la que se justifica la baja. 5. Confirma la acción.
Postcondición	El participante queda marcado con baja justificada en la edición seleccionada.
Excepciones	
Importancia	Media.

Tabla B.18: CU-18. Baja justificada

CU-19	Ver inscripciones
Actor	Gestor
Requisitos asociados	RF-3.4.4
Descripción	Permite al gestor listar todos los participantes inscritos en una edición y los datos de sus inscripciones.
Precondiciones	El gestor debe estar autenticado. La edición debe existir y tener participantes inscritos.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar ediciones". 2. Elige una edición de la lista.
Postcondición	Se muestra el listado de participantes inscritos en la edición seleccionada.
Excepciones	La edición no tiene participantes inscritos (mensaje)
Importancia	Media.

Tabla B.19: CU-19. Ver inscripciones

CU-20	Crear formadores
Actor	Gestor
Requisitos asociados	RF-3.5.1
Descripción	Permite al gestor dar de alta un nuevo formador en la base de datos.
Precondiciones	El gestor debe estar autenticado. Debe existir al menos un participante registrado en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona "Gestionar formadores". 2. En el listado de participantes existentes, selecciona cual quiere dar de alta como formador.
Postcondición	El formador se registra en la base de datos.
Excepciones	El participante seleccionado ya tiene un perfil de formador (mensaje).
Importancia	Alta.

Tabla B.20: CU-20. Crear formadores

CU-21	Ver formadores
Actor	Gestor
Requisitos asociados	RF-3.5.2
Descripción	Permite al gestor visualizar el listado de formadores registrados y consultar sus datos.
Precondiciones	El gestor debe estar autenticado. Deben existir formadores registrados en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona "Gestionar formadores". 2. Consulta la lista de formadores registrados. (Opcional) Selecciona un formador para ver sus datos en detalle.
Postcondición	Se muestran los datos de los formadores registrados en el sistema.
Excepciones	No hay formadores registrados (mensaje)
Importancia	Media.

Tabla B.21: CU-21. Ver formadores

CU-22	Editar formador
Actor	Gestor
Requisitos asociados	RF-3.5.3
Descripción	Permite al gestor modificar los datos de un formador existente.
Precondiciones	El gestor debe estar autenticado. Debe existir al menos un formador registrado en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar formadores". 2. Elige un formador de la lista. 3. Realiza los cambios necesarios en el formulario. 4. Confirma la edición.
Postcondición	Los cambios realizados se guardan y el formador se actualiza
Excepciones	Campos obligatorios sin rellenar (mensaje).
Importancia	Media.

Tabla B.22: CU-22. Editar formador

CU-23	Baja de un formador
Actor	Gestor
Requisitos asociados	RF-3.5.4
Descripción	Permite al gestor eliminar el registro de un formador que ya no colaborará.
Precondiciones	El gestor debe estar autenticado. El formador debe estar registrado en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar formadores ". 2. Elige un formador de la lista. 3. Confirma la baja del formador.
Postcondición	El formador se elimina del sistema.
Excepciones	El formador tiene ediciones asociadas y no puede ser eliminado (mensaje).
Importancia	Alta.

Tabla B.23: CU-23. Baja de un formador

CU-24	Asignar formador
Actor	Gestor
Requisitos asociados	RF-3.6.1
Descripción	Permite al gestor vincular un formador a una edición específica.
Precondiciones	El gestor debe estar autenticado. El formador y la edición deben estar registrados en el sistema.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar ediciones". 2. Elige una edición de la lista. 3. Selecciona un formador. 4. Confirma la asignación.
Postcondición	El formador queda asignado en la edición seleccionada.
Excepciones	
Importancia	Alta.

Tabla B.24: CU-24. Asignar formador

CU-25	Eliminar asignación
Actor	Gestor
Requisitos asociados	RF-3.6.2
Descripción	Permite al gestor eliminar la asignación de un formador a una edición específica.
Precondiciones	El gestor debe estar autenticado. El formador debe estar asignado a una edición.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar ediciones". 2. Elige una edición de la lista. 3. Selecciona un formador. 4. Confirma la acción.
Postcondición	La asignación del formador a la edición se elimina del sistema.
Excepciones	
Importancia	Media.

Tabla B.25: CU-25. Eliminar asignación

CU-26	Editar datos de asignación
Actor	Gestor
Requisitos asociados	RF-3.6.3
Descripción	Permite al gestor modificar los datos relacionados con la asignación de un formador a una edición, como retribución y horas impartidas.
Precondiciones	El gestor debe estar autenticado. El formador debe estar asignado a una edición.
Acciones	1. El gestor accede al portal del gestor y selecciona " Gestionar formadores". 2. Elige una edición de la lista. 3. Selecciona un formador. 4. Modifica los datos necesarios. 5. Confirma los cambios.
Postcondición	Los datos de la asignación se actualizan en el sistema.
Excepciones	Campos obligatorios sin rellenar (mensaje).
Importancia	Media.

Tabla B.26: CU-26. Editar datos de asignación

CU-27	Ver ficha formativa
Actor	Participante
Requisitos asociados	RF-4.1
Descripción	Permite al participante consultar las ediciones en las que está inscrito (próximas, certificadas, etc.).
Precondiciones	El participante debe estar autenticado. Deben existir ediciones en las que el participante esté inscrito.
Acciones	1. El participante accede al portal del participante. 2. Selecciona "Ver ficha formativa".
Postcondición	Se muestra la ficha formativa del participante, dividida en tres pestañas (próximas, certificadas, otras).
Excepciones	No hay ediciones inscritas (mensaje)
Importancia	Media.

Tabla B.27: CU-27. Ver ficha formativa

CU-28	Listar actividades y ediciones disponibles
Actor	Participante
Requisitos asociados	RF-4.2
Descripción	Permite al participante visualizar las actividades y ediciones existentes, así como la acción permitida para en cada una.
Precondiciones	El participante debe estar autenticado. Deben existir actividades formativas disponibles.
Acciones	1. El participante accede al portal del participante. 2. Selecciona la opción de actividades a consultar.
Postcondición	Se muestra la lista de actividades y ediciones disponibles con su respectiva acción permitida.
Excepciones	No hay actividades disponibles (mensaje)
Importancia	Media.

Tabla B.28: CU-28. Listar ediciones disponibles

CU-29	Editar datos de contacto del participante
Actor	Participante
Requisitos asociados	RF-4.3
Descripción	Permite al participante actualizar su información de contacto
Precondiciones	El participante debe estar autenticado.
Acciones	<ol style="list-style-type: none"> 1. El participante accede al portal del participante. 2. Selecciona "Mis datos de contacto". 3. Modifica los campos necesarios en el formulario. 4. Confirma los cambios.
Postcondición	Los datos del participante se actualizan en el sistema.
Excepciones	
Importancia	Media.

Tabla B.29: CU-29. Editar datos de contacto del participante

CU-30	Ver ediciones asignadas
Actor	Formador
Requisitos asociados	RF-5.1
Descripción	Permite al formador visualizar las ediciones que tiene asignadas, tanto abiertas como cerradas.
Precondiciones	El formador debe estar autenticado en el sistema. Deben existir ediciones asignadas al formador.
Acciones	<ol style="list-style-type: none"> 1. El formador accede al portal del formador. 2. Selecciona "Ediciones asignadas". 3. Elige entre las pestañas de ediciones abiertas o cerradas.
Postcondición	Se muestra el listado de ediciones asignadas al formador.
Excepciones	No hay ediciones asignadas (mensaje)
Importancia	Alta

Tabla B.30: CU-30. Ver ediciones asignadas

CU-31	Gestionar sesiones y asistencia
Actor	Formador
Requisitos asociados	RF-5.2
Descripción	Permite al formador gestionar las sesiones y registrar la asistencia de los participantes en una edición asignada.
Precondiciones	El formador debe estar autenticado en el sistema. Debe haber ediciones abiertas asignadas al formador. Deben existir participantes inscritos en la edición.
Acciones	1. El formador accede al portal del formador y selecciona una edición abierta. 2. Visualiza las sesiones creadas o crea nuevas sesiones. 3. Registra la asistencia de los participantes para cada sesión. 4. Confirma los cambios.
Postcondición	La asistencia queda registrada en el sistema.
Excepciones	Campos obligatorios sin rellenar (mensaje).
Importancia	Alta

Tabla B.31: CU-31. Gestionar sesiones y asistencia

CU-32	Registrar calificaciones
Actor	Formador
Requisitos asociados	RF-5.3
Descripción	Permite al formador registrar calificaciones para los participantes de una edición calificable.
Precondiciones	El formador debe estar autenticado en el sistema. La edición debe estar marcada como calificable. Deben existir participantes inscritos en la edición.
Acciones	1. El formador accede al portal del formador y selecciona una edición calificable. 2. Introduce las calificaciones para cada participante. 3. Confirma los cambios.
Postcondición	Las calificaciones quedan registradas en el sistema.
Excepciones	
Importancia	Alta

Tabla B.32: CU-32. Registrar calificaciones

CU-33	Editar datos de contacto del formador
Actor	Formador
Requisitos asociados	RF-5.4
Descripción	Permite al formador actualizar su información de contacto.
Precondiciones	El formador debe estar autenticado en el sistema.
Acciones	<ol style="list-style-type: none"> 1. El formador accede al portal del formador. 2. Selecciona "Mis datos de contacto". 3. Modifica los campos necesarios en el formulario. 4. Confirma los cambios.
Postcondición	Los datos del formador se actualizan en el sistema.
Excepciones	
Importancia	Media

Tabla B.33: CU-33. Editar datos de contacto del formador

CU-34	Certificar edición
Actor	Gestor
Requisitos asociados	RF-6
Descripción	Permite al gestor certificar una edición al validar asistencia y calificaciones.
Precondiciones	<p>El gestor debe estar autenticado.</p> <p>La edición debe estar cerrada.</p> <p>La asistencia y calificaciones deben estar completas.</p>
Acciones	<ol style="list-style-type: none"> 1. El gestor accede al portal del gestor y selecciona "Ediciones para certificar". 2. Selecciona una edición de la lista. 3. Revisa la información de asistencia y calificaciones. 4. Confirma la certificación.
Postcondición	La edición queda certificada y los certificados son asignados.
Excepciones	
Importancia	Alta

Tabla B.34: CU-34. Certificar edición

CU-35	Enlace en el pie de página
Actor	Usuario, Gestor, Participante, Formador
Requisitos asociados	RF-2.2
Descripción	Permite a cualquier actor acceder al enlace de la política de protección de datos desde el footer de cualquier página.
Precondiciones	El actor debe estar navegando por la aplicación. El footer debe estar visible en la interfaz.
Acciones	1. El actor visualiza el footer en la página que está navegando. 2. Selecciona el enlace "Política de protección de datos".
Postcondición	El actor es redirigido a la página que muestra la política de protección de datos.
Excepciones	
Importancia	Baja

Tabla B.35: CU-35. Enlace en el pie de página

CU-36	Página de política de protección de datos
Actor	Usuario, Gestor, Participante, Formador
Requisitos asociados	RF-2.3
Descripción	Permite a cualquier actor visualizar el contenido completo de la política de protección de datos en una página dedicada.
Precondiciones	El actor debe tener acceso a la página /politica-datos mediante un enlace o URL directa.
Acciones	1. El actor accede a la página /politica-datos mediante el enlace proporcionado en el footer o escribiendo la URL en el navegador. 2. Visualiza el contenido completo de la política de protección de datos.
Postcondición	El actor puede leer y revisar el contenido de la política de protección de datos.
Excepciones	
Importancia	Baja

Tabla B.36: CU-36. Página de política de protección de datos

Apéndice C- Especificación de diseño

C.1. Introducción

En este apéndice se presenta el diseño de la aplicación “Gestión de formación”. El propósito es detallar, desde una perspectiva técnica, cómo se implementarán los requisitos previamente especificados. Se explicará la estructura de los datos, la lógica de negocio y la arquitectura general del sistema.

A continuación, se describen los apartados que cubren el diseño de datos, el diseño procedimental y el diseño arquitectónico.

C.2. Diseño de datos

En esta sección se detalla el modelo de datos que soportará las funcionalidades de la aplicación. Para ello, se parte de los requisitos funcionales que involucran la creación, lectura, actualización y eliminación (CRUD) de la información.

A grandes rasgos, el sistema “Gestión de formación” maneja las siguientes entidades principales:

Usuario.

Un usuario dispone de un nombre de usuario (que se utilizará para iniciar sesión), una contraseña cifrada, una dirección de correo electrónico principal, y roles asociados (almacenados en formato JSON). Además, cuenta con campos adicionales como un indicador de verificación de cuenta, una fecha de creación de la cuenta y una fecha de última actualización de la contraseña.

Curso.

Un curso dispone de un código único para identificarlo, un nombre, una descripción, y un conjunto de datos específicos: la duración en horas totales, el número de horas virtuales (que determina la modalidad del curso: presencial, semipresencial o virtual), los objetivos del curso, los contenidos a tratar, los destinatarios, los requisitos de participación, y una justificación para su creación.

Adicionalmente, un curso incluye información sobre el coordinador responsable, el número estimado de participantes por edición, el número de ediciones previstas inicialmente, un plazo de solicitud para inscripciones, observaciones adicionales, y si el curso será visible en el Portal del Participante y calificable.

Edición.

Una edición se asocia directamente a un curso y dispone de datos específicos, incluyendo: un código único derivado del curso principal, las fechas de inicio y fin, el calendario detallado, el horario de las sesiones, el lugar de impartición, el número de sesiones que se van a impartir, el número máximo de plazas disponibles, y un estado que puede indicar si la edición está abierta, cerrada o en proceso de certificación.

Participante.

Un participante es una persona registrada en la plataforma y asociada a un usuario del sistema. Dispone de datos personales y profesionales, incluyendo: un NIF único, su nombre y apellidos, detalles sobre su puesto de trabajo como el grupo, nivel, puesto, subunidad, unidad y centro de destino, y otros datos administrativos como el tipo de régimen jurídico, la situación administrativa y el código de plaza. Además, se incluye información adicional como su titulación (nivel, fecha y descripción), su organización, el turno asignado, y datos de contacto, como teléfonos (particular, móvil y del trabajo) y un correo auxiliar. También se registran datos personales como su fecha de nacimiento y sexo.

Formador.

Un formador es una persona registrada en la plataforma que puede impartir cursos de formación. Dispone de datos personales y profesionales, incluyendo: un NIF único, su nombre y apellidos, la organización a la que pertenece, un correo auxiliar para contacto, un número de teléfono, y un campo para registrar observaciones adicionales. También se incluye un dato específico sobre el régimen jurídico del formador, indicado mediante un código (por ejemplo, empleado público, autónomo, empresa, etc.).

Inscripciones (participante_edicion).

Representa la relación entre un participante y una edición de curso. Incluye información específica sobre la participación, como la fecha de solicitud, posibles bajas justificadas con la fecha correspondiente, y resultados obtenidos como la nota de la prueba final (si aplica), el estado del certificado (si se emitió), el número de libro y título asignados, y si el participante fue considerado apto o no en la edición. Además, se registran observaciones adicionales y la dirección IP desde la que se realizó la inscripción para esta edición específica o el username del gestor que llevó a cabo la inscripción.

Asignaciones (formador_edicion).

Representa la relación entre un formador y una edición de curso. Incluye información específica sobre la colaboración, como las horas impartidas, la retribución prevista y la retribución ejecutada, además de datos administrativos como la fecha de entrega de la hoja de firma, de envío de los datos bancarios, y del documento de incompatibilidades. También se registran indicadores como si el pago está asociado a fondos FEDAP, si se va a realizar una evaluación del formador, si el trámite de su retribución fue grabado en Sorolla, o si la colaboración se llevó a cabo sin coste económico. Adicionalmente, se incluye información sobre coincidencias de turno de impartición del curso con turno de trabajo y observaciones relacionadas, así como las fechas de envío y recepción del control de personal.

Sesión.

Una sesión representa una clase o evento individual dentro de una edición de curso. Incluye información específica como la fecha de realización, la hora de inicio, y la duración en minutos. Además, se registra el tipo de sesión (presencial, virtual, u otra modalidad definida), y observaciones adicionales relacionadas con su ejecución.

Asistencia.

La asistencia representa el registro de la presencia de un participante en una sesión específica. Incluye información sobre si el participante asistió, si justificó su ausencia en caso de no asistir, y observaciones adicionales relacionadas con la asistencia.

Diagrama Relacional

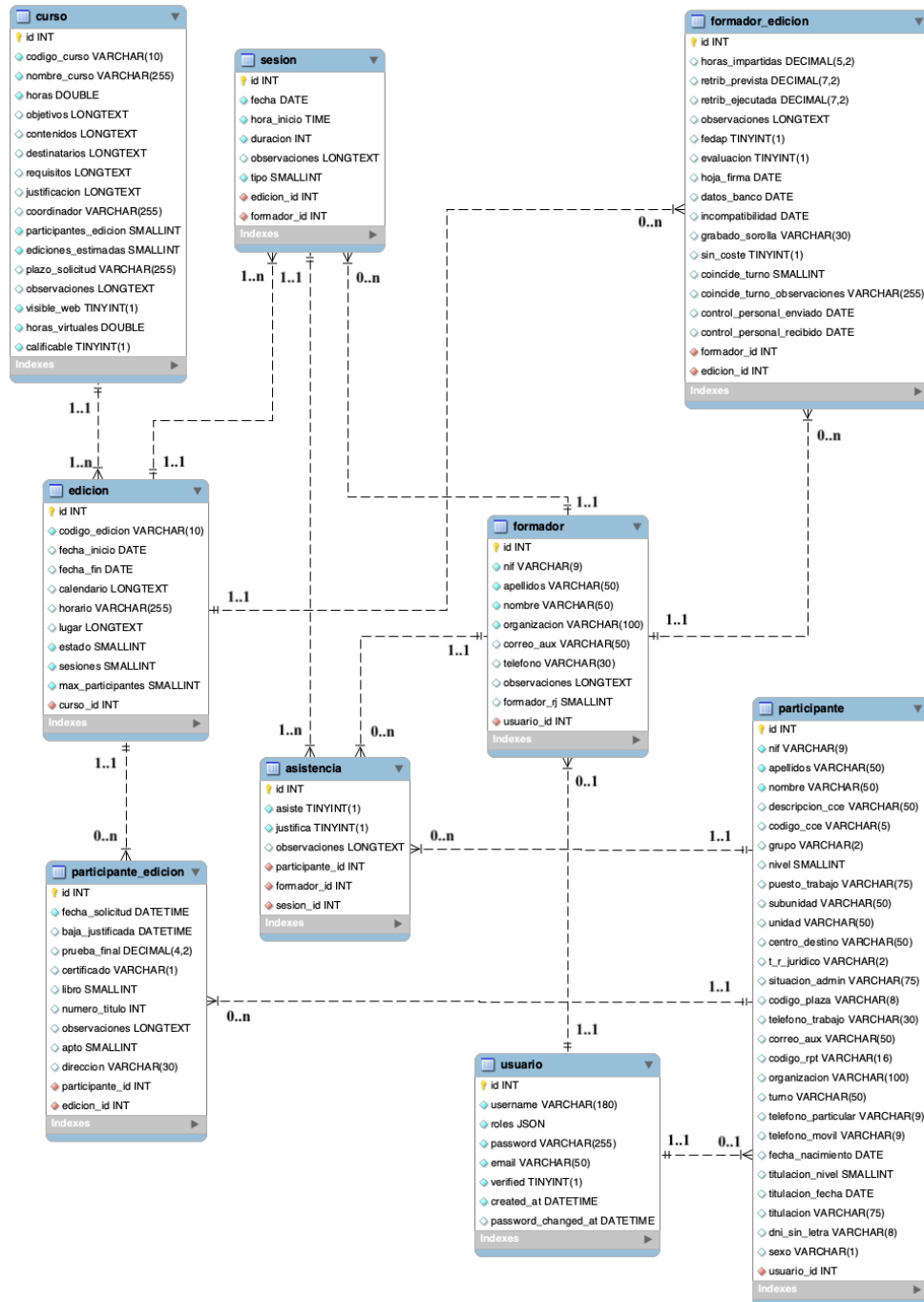


Figura C.1: Diagrama Entidad-Relación

C.3. Diseño procedimental

En este apartado se presentan los diagramas de secuencia correspondientes a los procesos clave del sistema. Estos diagramas ilustran de manera detallada cómo interactúan los diferentes actores, componentes del sistema y la base de datos para implementar las funcionalidades principales de la aplicación.

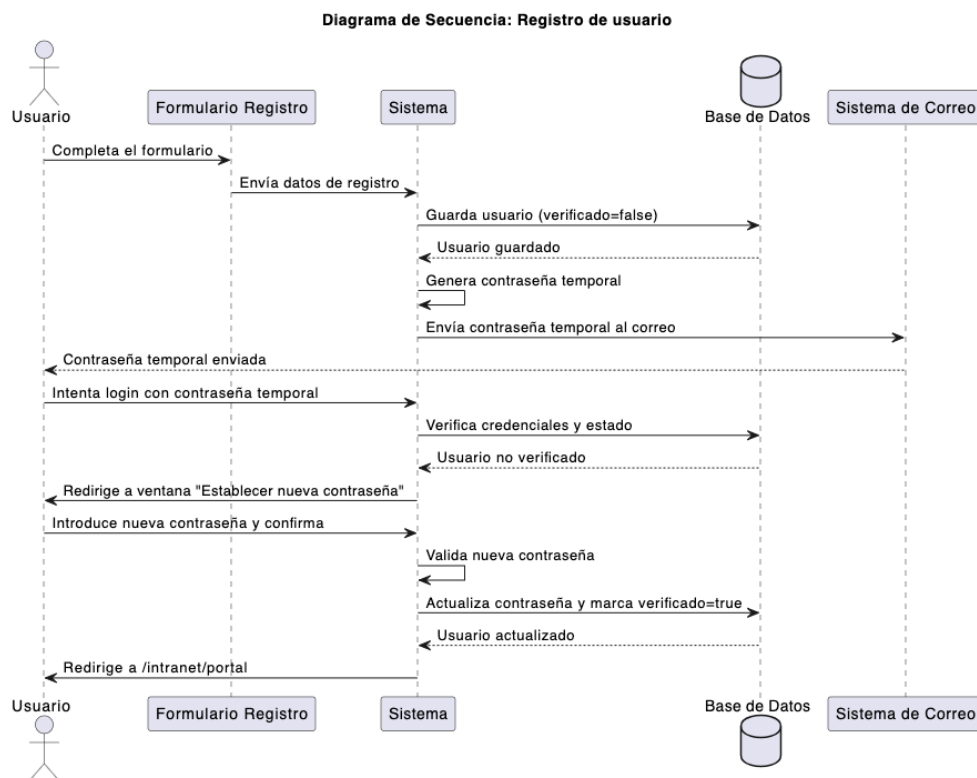


Figura C.2: Diagrama de secuencia. Registro de usuario

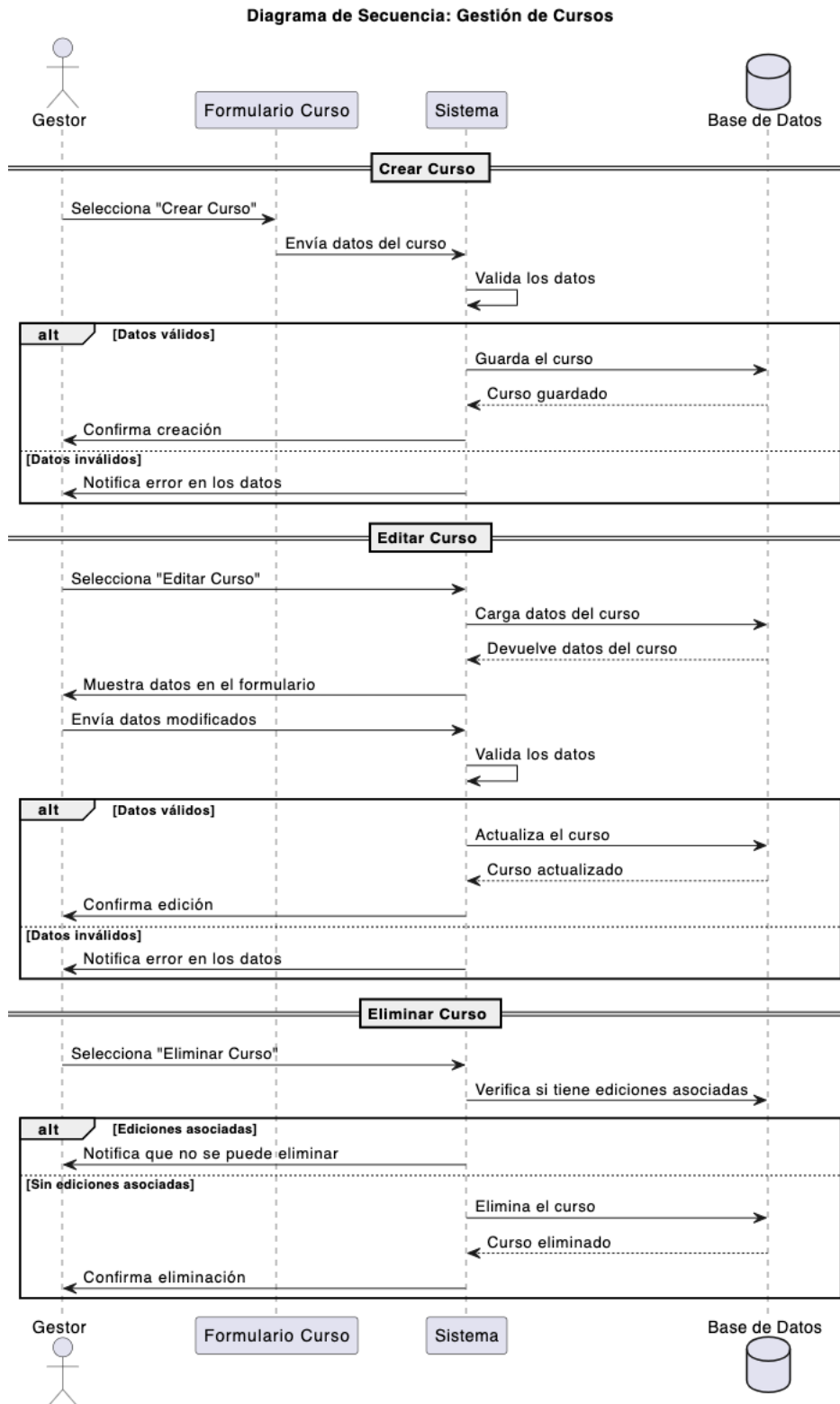


Figura C.3: Diagrama de secuencia. Gestión cursos

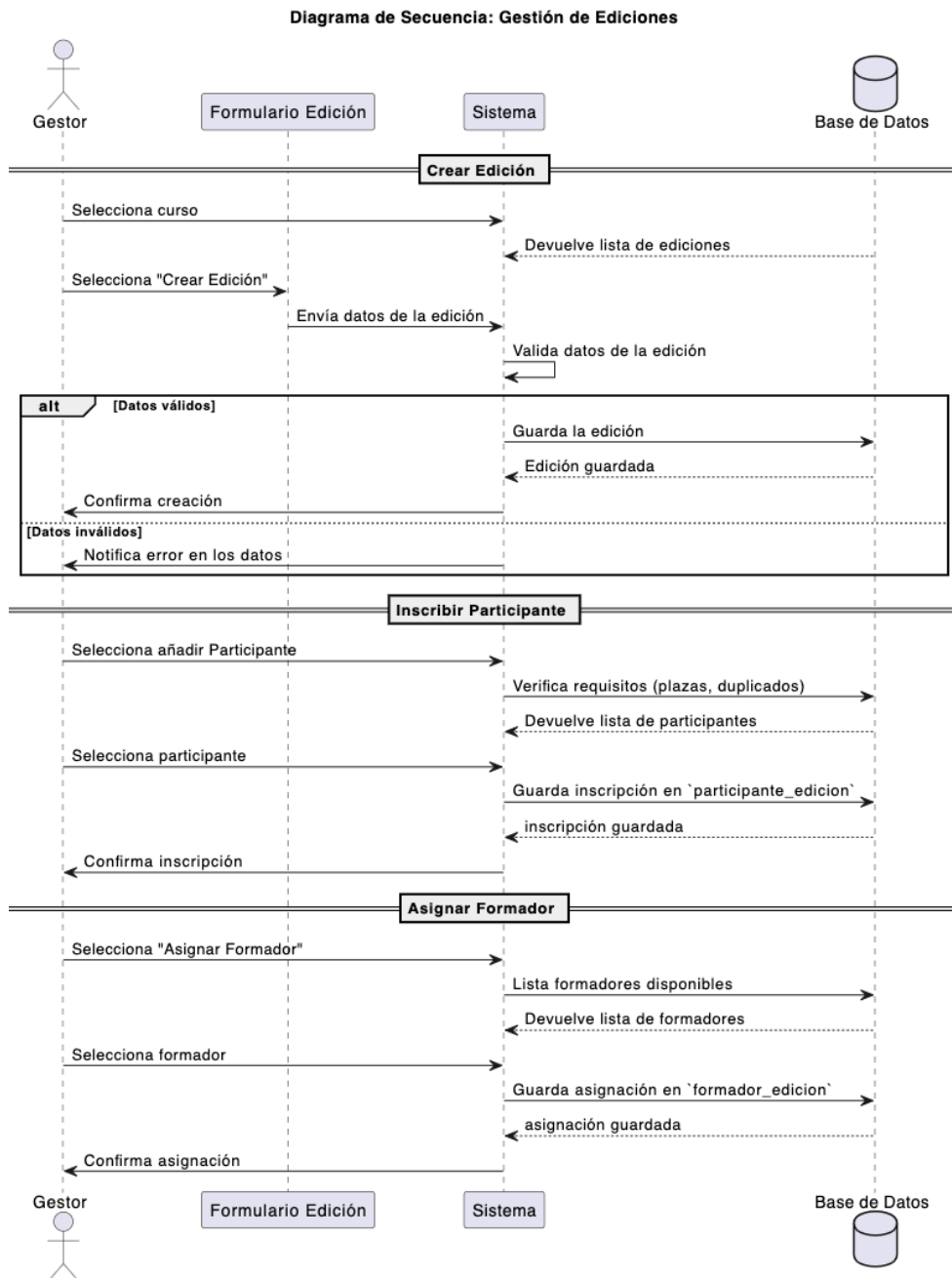


Figura C.4: Diagrama de secuencia. Gestión ediciones

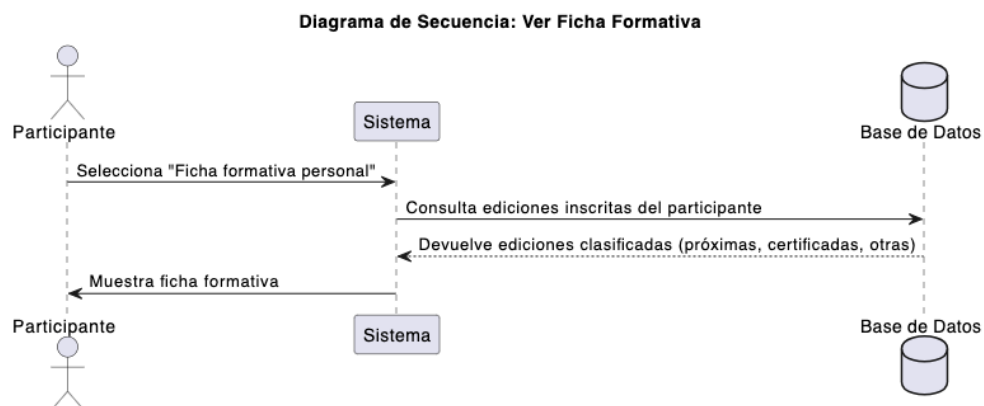


Figura C.5: Diagrama de secuencia. Ficha formativa

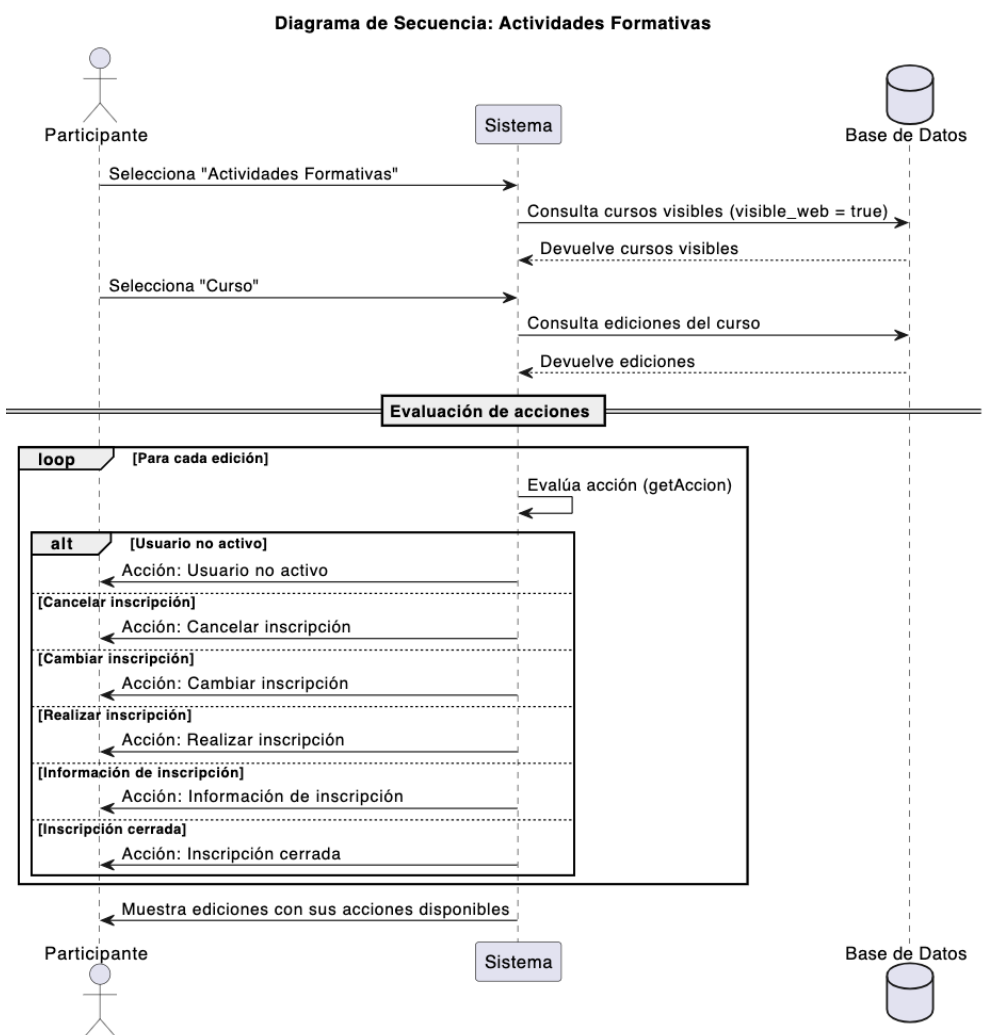


Figura C.6: Diagrama de secuencia. Actividades formativas

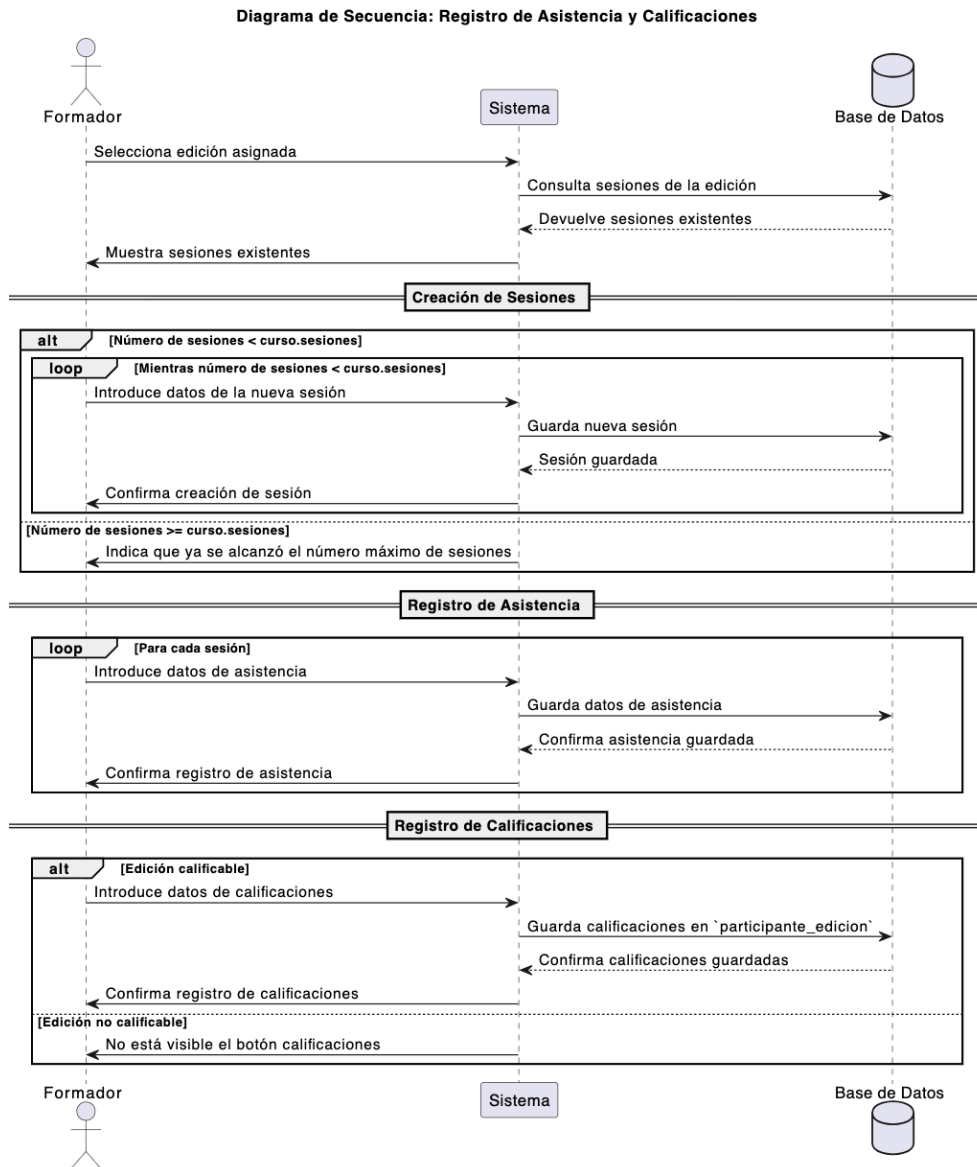


Figura C.7: Diagrama de secuencia. Registro de asistencias

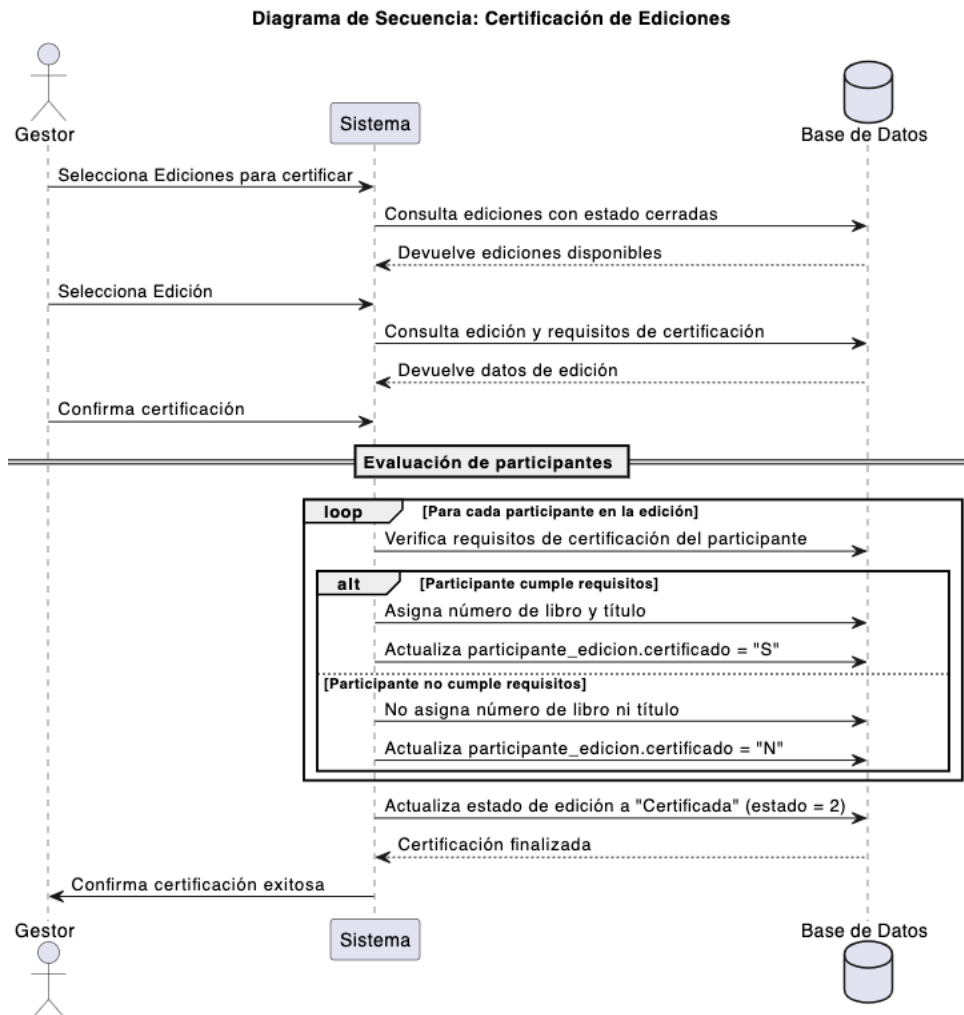


Figura C.8: Diagrama de secuencia. Certificación de ediciones

C.4. Diseño arquitectónico

En esta sección se describe la arquitectura del sistema "Gestión de formación", destacando los principios fundamentales, el patrón arquitectónico adoptado y los principales componentes que lo conforman. Este diseño asegura que el sistema sea escalable, mantenible y cumpla con los requisitos funcionales y no funcionales definidos en etapas anteriores.

C.4.1. Patrón arquitectónico

El sistema sigue una arquitectura basada en el patrón Modelo-Vista-Controlador (MVC), ampliamente utilizado en aplicaciones web modernas debido a su capacidad para desacoplar responsabilidades y facilitar el mantenimiento del código.

Modelo:

- Representa los datos y la lógica de negocio.
- Se implementa mediante entidades gestionadas por Doctrine ORM, que reflejan la estructura de la base de datos.
- Los repositorios asociados a estas entidades permiten realizar consultas y operaciones CRUD de manera eficiente.

Vista:

- Se encarga de la interacción con el usuario, generando las interfaces dinámicas mediante el motor de plantillas Twig.
- Las vistas reciben los datos procesados por los controladores y los presentan en formato HTML, con compatibilidad para navegadores modernos.

Controlador:

- Actúa como intermediario entre la vista y el modelo.
- Procesa las solicitudes HTTP entrantes, valida los datos y delega las operaciones a los servicios y repositorios correspondientes. Finalmente, genera una respuesta adecuada para el usuario.

La implementación de este patrón en Symfony asegura una separación clara entre la lógica de presentación, la lógica de negocio y la lógica de acceso a datos, permitiendo un desarrollo modular y escalable.

C.4.2. Arquitectura general

El sistema está diseñado en tres capas principales, cada una con responsabilidades claramente definidas:

1) Capa de presentación (Frontend):

- Encargada de la interacción con el usuario final mediante una interfaz web.
- Utiliza el motor de plantillas Twig para renderizar vistas dinámicas, asegurando una experiencia responsiva y accesible.
- Gestiona recursos frontend (CSS, JavaScript) mediante Symfony AssetMapper, eliminando dependencias externas como npm o yarn.

2) Capa de lógica de negocio (Backend):

- Implementada con Symfony 7, incluye los controladores y servicios que encapsulan las reglas de negocio del sistema.
- Incorpora mecanismos de seguridad configurados en security.yaml, como control de acceso basado en roles (ROLE_ADMIN, ROLE_TEACHER, ROLE_USER).
- Implementa eventos y listeners para automatizar procesos clave, como notificaciones o actualizaciones.

3) Capa de datos (Base de datos):

- Gestionada con MySQL 8, contiene tablas que reflejan el modelo E/R definido previamente.
- Doctrine ORM se utiliza para mapear las entidades de la capa de negocio a las tablas de la base de datos, permitiendo consultas eficientes y un manejo robusto de transacciones.

C.4.3. Componentes principales

El sistema se organiza en varios componentes fundamentales que interactúan para implementar las funcionalidades:

Controladores:

- Procesan las solicitudes HTTP entrantes y llaman a los servicios para ejecutar la lógica de negocio.

- Ejemplos:

CursoController: Gestiona la creación, edición y eliminación de cursos.

EdicionController: Gestiona ediciones de cursos y permite asignar formadores y participantes.

ParticipanteController: Gestiona inscripciones de participantes.

Servicios:

- Implementan la lógica de negocio que se reutiliza en diferentes controladores.
- Ejemplo: Servicio para generación de certificados o validación de inscripción.

Repositorios:

- Son responsables de las consultas a la base de datos relacionadas con entidades específicas.
- Ejemplo:
CursoRepository con métodos como findByYear() para buscar cursos de un año específico.

Configuraciones de Seguridad:

- Definidas en el archivo security.yaml.
- Controlan el acceso a las rutas según los roles del usuario (ROLE_ADMIN, ROLE_TEACHER, ROLE_USER).

Eventos y Listeners:

- Automatizan procesos como el envío de notificaciones tras inscripciones exitosas o cambios de estado en una edición.

Vistas:

- Renderizan la información procesada por los controladores. Usan Twig para generar páginas dinámicas que permiten la interacción del usuario con las funcionalidades del sistema.

C.4.4. Diagrama de arquitectura

Este diagrama ilustra cómo interactúan las capas y componentes principales del sistema, mostrando el flujo de datos desde el usuario hasta la base de datos y viceversa.

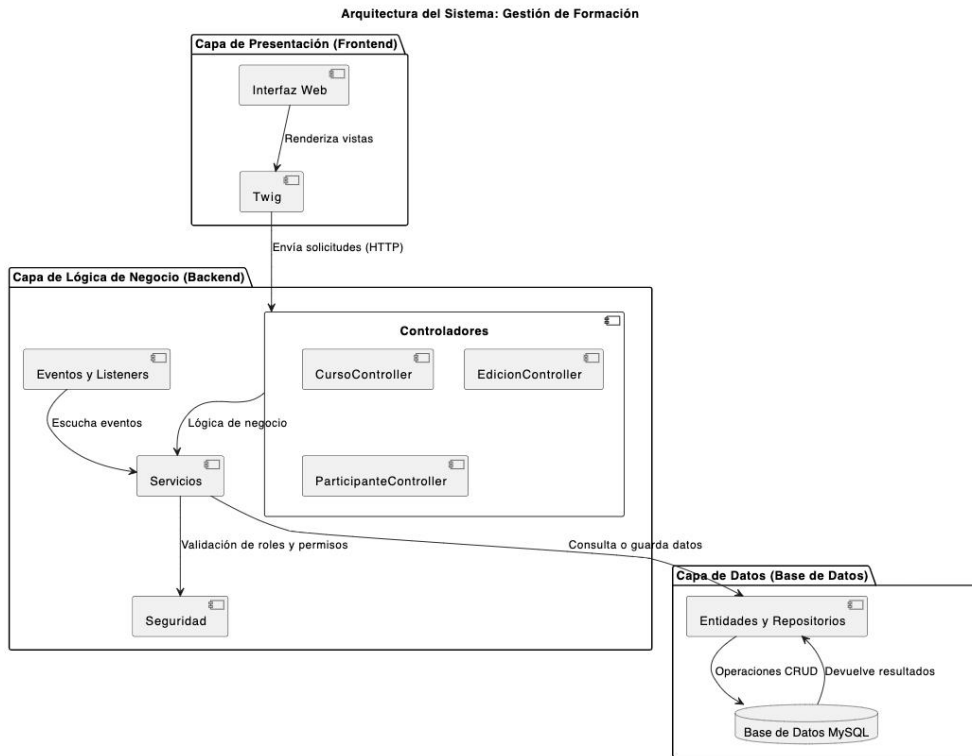


Figura C.9: Diagrama de arquitectura

Capa de Presentación (Frontend):

- **Interfaz Web:** Es el navegador donde los usuarios interactúan con el sistema.
- **Twig:** Genera las vistas dinámicas basadas en datos provenientes del backend.

Capa de Lógica de Negocio (Backend):

- **Controladores:** Reciben las solicitudes desde el frontend y llaman a los servicios necesarios.
- **Servicios:** Implementan la lógica de negocio.
- **Seguridad:** Valida roles y permisos según las reglas definidas.
- **Eventos y Listeners:** Automatizan acciones en función de eventos del sistema.

Capa de Datos (Base de Datos):

- Entidades y Repositorios: Representan los datos y manejan las operaciones CRUD.
- Base de Datos MySQL: Almacena la información persistente del sistema.

C.4.5. Interacciones entre componentes

El flujo típico de una solicitud en el sistema es el siguiente:

- 1) El usuario realiza una solicitud HTTP desde la interfaz web (ejemplo: crear un curso).
- 2) La solicitud llega al controlador correspondiente, que valida los datos y delega las operaciones necesarias a los servicios.
- 3) Los servicios interactúan con los repositorios para consultar o modificar datos en la base de datos.
- 4) Finalmente, el controlador genera una respuesta (HTML o JSON) que se presenta al usuario a través de la vista (navegador del usuario).

C.4.6. Tecnologías utilizadas

El sistema utiliza las siguientes tecnologías clave:

Symfony 7: Framework PHP para el backend, que proporciona la base estructural del sistema y facilita la creación de aplicaciones web modernas.

PHP 8.2: Lenguaje de programación utilizado para desarrollar el backend del sistema.

Doctrine ORM: Mapeador objeto-relacional (ORM) que permite interactuar con la base de datos utilizando entidades de PHP, reduciendo el uso de consultas SQL manuales y facilitando un manejo más intuitivo de los datos.

MySQL 8: Sistema de gestión de bases de datos relacional utilizado para almacenar de manera persistente la información del sistema.

Twig: Motor de plantillas que permite generar vistas dinámicas y responsivas para el frontend.

AssetMapper: Herramienta integrada en Symfony para la gestión de recursos frontend como CSS y JavaScript.

Apéndice D- Documentación técnica de programación

D.1. Introducción

Este apéndice proporciona la documentación técnica relacionada con la programación del proyecto "Gestión de formación".

Se incluye información detallada sobre la estructura del proyecto, los pasos para configurar el entorno de desarrollo, y las pruebas realizadas para garantizar la calidad del software.

Su objetivo es servir como referencia para futuros desarrolladores interesados en extender o mantener el proyecto.

D.2. Estructura de directorios

El repositorio del proyecto "Gestión de formación" sigue una estructura organizada para facilitar el mantenimiento, la escalabilidad y la colaboración.

A continuación, se describe cada carpeta y su contenido:

/.github: Contiene los workflows de GitHub Actions para la integración continua. Incluye configuración para ejecutar pruebas automatizadas y verificar la calidad del código.

/assets: Almacena los recursos de frontend como archivos SCSS, JavaScript o imágenes que se gestionan mediante Symfony AssetMapper.

/bin: Incluye los scripts ejecutables relacionados con Symfony, como console, utilizado para ejecutar comandos del framework.

/config: Contiene los archivos de configuración del proyecto.

- **/packages:** Configuración de paquetes específicos utilizados en el proyecto, como Doctrine, Twig y Security.
- **/routes:** Definición de rutas organizadas por módulos o funcionalidades.
- Archivos principales como services.yaml y security.yaml.

/docs: Incluye documentación adicional del proyecto, como memoria o anexo de especificaciones técnicas.

/migrations: Contiene los scripts generados por Doctrine para gestionar los cambios en el esquema de la base de datos.

/public: Directorio raíz accesible públicamente, donde se encuentran archivos como index.php, que sirve como punto de entrada para la aplicación, y recursos estáticos compilados.

/src: Contiene el código fuente principal del proyecto, organizado en las siguientes carpetas:

- **/Controller:** Controladores que gestionan las peticiones HTTP y orquestan las respuestas. Ejemplo:
 - **/Forpas:** Controladores relacionados con las funcionalidades del Portal del Gestor, Participante y Formador.
 - **/Intranet:** Controladores generales del sistema (login, registro, LOPD...)
- **/Entity:** Define las entidades del modelo de datos mapeadas a las tablas de la base de datos mediante Doctrine ORM.
- **/EventSubscriber:** Contiene suscriptores a eventos que permiten ejecutar lógica personalizada en eventos específicos de Symfony.
- **/Form:** Define los formularios utilizados para interactuar con los usuarios, como el registro o la edición de datos.
- **/Repository:** Repositorios que gestionan las consultas avanzadas y personalizadas a la base de datos.
- **/Security:** Contiene clases relacionadas con la seguridad del sistema, como el autenticador para la gestión de inicio de sesión.
- **/Twig:** Extensiones personalizadas para el motor de plantillas Twig.

/templates: Contiene las plantillas HTML renderizadas mediante Twig para construir las vistas de la aplicación. La organización de esta carpeta es la siguiente:

- **/form:** Plantillas relacionadas con los formularios del sistema.
- **/intranet:** Plantillas del conjunto de la intranet separada en:
 - **/forpas:** Plantillas de los distintos portales del proyecto. Estructurada en una subcarpeta por cada portal.
 - **/sistema:** Plantillas relacionadas con funcionalidades genéricas del sistema (registro, login, LOPD...).

- **/layout:** Plantillas del layout y de los mensajes flash.

/tests: Contiene las pruebas unitarias y funcionales, así como los reportes de cobertura del código y de análisis estático del código.

/translations: Archivos para la internacionalización (i18n) y traducción del sistema.

/var: Almacena archivos generados en tiempo de ejecución, como logs y cachés.

/vendor: Contiene las dependencias instaladas mediante Composer. Esta carpeta no debe modificarse manualmente.

D.3. Manual del programador

El manual del programador tiene como objetivo proporcionar una guía completa sobre la configuración, desarrollo, pruebas y mantenimiento del proyecto "Gestión de Formación". Está dirigido a desarrolladores que deseen colaborar, extender o mantener el sistema.

Requisitos Previos

Antes de trabajar en este proyecto, asegúrate de cumplir con los siguientes requisitos:

Entorno de desarrollo:

- PHP \geq 8.2
- MySQL 8.0
- Symfony 7.1
- Symfony-CLI
- PHPUnit 9.6
- PHPStan 1.12
- Composer

Herramientas recomendadas:

- IDE: PHPStorm (opcional, pero altamente recomendado).
- Administrador de base de datos: Sequel Ace, Adminer o phpMyAdmin.
- Git para el control de versiones.

Configuración Inicial

La configuración inicial asegura que el proyecto esté listo para ser ejecutado en tu entorno de desarrollo local. Al final de estos pasos, tendrás el sistema configurado con todas las dependencias necesarias, una base de datos lista para usar, y un servidor de desarrollo operativo para realizar pruebas y modificaciones.

Paso 1: Clonar el repositorio

Clona el repositorio desde GitHub a tu máquina local:

```
git clone https://github.com/VictorVaqueroUBU/TFG.git gestion-formacion
```

```
cd gestion-formacion
```

Paso 2: Instalar dependencias

Ejecuta Composer para instalar todas las dependencias necesarias del proyecto:

```
composer install
```

Paso 3: Instalar componentes JavaScript

```
bin/console importmap:install
```

```
bin/console sass:build
```

Paso 4: Crear base de datos

```
CREATE DATABASE IF NOT EXISTS formacion;
```

Paso 5: Crear usuario de la base de datos

```
CREATE USER IF NOT EXISTS 'Usuario'@'localhost' IDENTIFIED BY 'Clave';
```

Paso 6: Dar acceso al usuario creado

```
GRANT ALL PRIVILEGES ON formacion.* TO 'Usuario'@'localhost';
```

Paso 7: Configurar las variables de entorno en el archivo .env

DATABASE_URL="mysql://usuario:contraseña@127.0.0.1:3306/formacion?serverVersion=8.0"

MAILER_DSN=native://default

MAILER_SENDER=usuario@correo.es

Paso 8: Realizar las migraciones de la base de datos:

php bin/console doctrine:migrations:migrate

Paso 9: Levantar el servidor de desarrollo:

symfony server:start

Paso 10: Acceder a la aplicación en tu navegador en la dirección:

https://localhost:8000/

Desarrollo

Creación de nuevas funcionalidades:

1) Crear una rama para la funcionalidad:

- Crea una rama nueva basada en developer:

git checkout -b <nombre_de_la_funcionalidad>

2) Desarrollar la funcionalidad:

- Crea o actualiza las entidades en src/Entity.
- Configura los repositorios en src/Repository para las consultas personalizadas.
- Define los controladores en src/Controller.
- Crea formularios en src/Form si la funcionalidad lo requiere.
- Diseña las vistas en templates.

3) Pruebas locales:

- **Ejecución de pruebas unitarias y funcionales.** Para garantizar que el sistema funciona correctamente, utiliza PHPUnit:

```
php bin/phpunit
```

- **Verificación de la calidad del código.** Utiliza PHPStan para analizar el código estático y detectar posibles errores:

```
php vendor/bin/phpstan analyse --memory-limit=512M  
--configuration=phpstan.dist.neon > tests/phpstan_report.txt
```

4) Subir cambios al repositorio remoto:

```
git add .
```

```
git commit -m "Descripción breve de la funcionalidad"
```

```
git push origin <nombre_de_la_funcionalidad>
```

```
git checkout developer
```

```
git merge <nombre_de_la_funcionalidad>
```

```
git push origin developer
```

5) Creación de migraciones de base de datos

Si realizas cambios en las entidades, genera una migración. Esto asegura que la estructura de la base de datos esté sincronizada con las entidades actualizadas y evita errores al interactuar con el sistema.:

```
php bin/console make:migration
```

```
php bin/console doctrine:migrations:migrate
```

Seguridad

La configuración de seguridad se gestiona en el archivo `config/packages/security.yaml`. Los roles principales son:

- **ROLE_ADMIN:** Acceso completo al Portal del Gestor.
- **ROLE_USER:** Acceso al Portal del Participante.
- **ROLE_TEACHER:** Acceso al Portal del Formador.

Recomendaciones:

Almacena las contraseñas utilizando hashing seguro (Symfony PasswordHasher).

Verifica que todas las rutas están protegidas adecuadamente mediante `access_control`.

D.4. Pruebas del sistema

Introducción

El apartado de pruebas del sistema describe el enfoque, las herramientas y los resultados obtenidos durante la validación del proyecto "Gestión de Formación". Estas pruebas son fundamentales en el desarrollo de software, ya que permiten verificar la funcionalidad, detectar errores tempranamente y garantizar que el sistema sea fiable y escalable.

Estrategia de Pruebas

Para garantizar la calidad del sistema, se implementó un enfoque basado en pruebas automatizadas utilizando PHPUnit y análisis estático con PHPStan. La combinación de estas herramientas permite verificar la funcionalidad del código y, al mismo tiempo, identificar problemas potenciales como incompatibilidades de tipo o métodos no utilizados, asegurando así un desarrollo robusto y fiable. Este enfoque incluyó:

- 1) **Pruebas unitarias:** Validan cada componente del sistema de manera independiente, como entidades, controladores y formularios.
- 2) **Pruebas funcionales:** Verifican el comportamiento de funcionalidades completas, como la gestión de cursos, ediciones y usuarios.
- 3) **Pruebas de integración:** Aseguran que los diferentes componentes del sistema funcionan correctamente en conjunto.
- 4) **Cobertura de código:** Se evaluó el alcance de las pruebas mediante la generación de informes de cobertura.

Herramientas Utilizadas

PHPUnit: Framework principal para la ejecución de pruebas unitarias y funcionales.

PHPStan: Herramienta para el análisis estático del código, configurada en el nivel 8 para asegurar la detección de incompatibilidades de tipos, métodos no utilizados y errores en anotaciones.

GitHub Actions: Automatización de las pruebas en cada commit o pull request para garantizar la integridad del sistema y verificar que los cambios no introduzcan errores.

Cobertura de código: Se generó un informe utilizando PHPUnit, el cual permite identificar índice de cobertura que abarcan las pruebas implementadas.

Resultados de las Pruebas

Cobertura General

El último informe generado indica los siguientes niveles de cobertura:

- **Clases:** 71.74% (33/46)
- **Métodos:** 91.77% (379/413)
- **Líneas:** 96.13% (2288/2380)

Estos resultados reflejan un alto grado de cobertura, especialmente en las líneas de código, lo cual garantiza que la mayoría de las funcionalidades han sido validadas. Por ejemplo, el módulo `CursoController` fue probado exhaustivamente con cobertura del 100% en sus 5 métodos y 42 líneas, validando operaciones críticas como creación, edición y eliminación de cursos. Además, las pruebas en `ParticipantePortalController` aseguraron la correcta gestión de vistas personalizadas para los participantes.

Cobertura por Módulos

- Controladores:

Todos los controladores cuentan con una cobertura del 100% en métodos y líneas, asegurando que las operaciones CRUD y la gestión de flujo del sistema han sido probadas exhaustivamente. Ejemplos destacados:

- `App\Controller\Forpas\CursoController`:
100% de cobertura en sus 5 métodos y 42 líneas.
- `App\Controller\Forpas\ParticipantePortalController`:
100% en 9 métodos y 107 líneas.

- Entidades:

Las entidades principales, como Curso, Edicion y Participante, muestran niveles de cobertura elevados. Ejemplo:

- App\Entity\Forpas\Curso:
100% en 38 métodos y 64 líneas.
- App\Entity\Forpas\Participante:
95.31% en métodos y 91.35% en líneas.

- Formularios:

Formularios como CursoType y ParticipanteType cuentan con cobertura completa del 100%, validando los procesos de gestión de datos y validación.

- Repositorios:

Los repositorios más utilizados, como CursoRepository y EdicionRepository, cuentan con una cobertura funcional de más del 80%.

Análisis Estático

El análisis estático del código se realizó con PHPStan configurado en un nivel alto de análisis (nivel 8). Esta herramienta permitió identificar y corregir:

- Incompatibilidades de tipo.
- Métodos no utilizados.
- Errores comunes en las anotaciones de tipo en entidades y formularios.

Tras las correcciones, PHPStan no detectó errores significativos, reflejando un código robusto y bien estructurado.

Pruebas de Integración con GitHub Actions

Las pruebas automatizadas fueron integradas en el flujo de trabajo de desarrollo utilizando GitHub Actions. Esta configuración permitió:

- Ejecutar las pruebas unitarias y funcionales automáticamente en cada commit o pull request.
- Asegurar la consistencia y calidad del código en las ramas principales del repositorio.
- Generar reportes inmediatos sobre el estado de las pruebas y la cobertura de código.

Conclusiones

Los resultados obtenidos en las pruebas evidencian un alto nivel de confianza en la funcionalidad y la calidad del código. Algunos puntos destacados son:

- 1) **Cobertura alta en módulos críticos:** Los controladores y formularios, que contienen gran parte de la lógica de la aplicación, tienen una cobertura cercana al 100%, asegurando que las funcionalidades principales han sido ampliamente probadas.
- 2) **Cobertura moderada en entidades:** La cobertura de métodos y líneas en las entidades podría incrementarse mediante pruebas adicionales que contemplen casos límite o menos comunes.
- 3) **Análisis estático satisfactorio:** La ausencia de errores detectados por PHPStan refleja la calidad y consistencia del código.

En general, el proyecto presenta una base sólida tanto en su funcionalidad como en su calidad técnica, gracias al enfoque riguroso en pruebas y validación del sistema.

Apéndice E- **Documentación de usuario**

E.1. Introducción

E.2. Requisitos de usuario

E.3. Instalación

E.4. Manual del usuario

|

***Apéndice F-* Anexo de sostenibilización curricular**

F.1. Introducción

F.2. Competencias en sostenibilidad adquiridas

F.3. Impacto social y ambiental

Bibliografía
