



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería Informática

Gestión de Formación

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



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. José Ignacio Santos Martín, profesor del departamento de Ingeniería de Organización.

Exponen:

Que el alumno D. Víctor Manuel Vaquero Mesa, con DNI 52669729M, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Gestión de Formación.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, dd de enero de 2025

Vº. Bº. del Tutor:

D. José Ignacio Santos Martín

Resumen

La formación continua de los trabajadores de una empresa es de vital importancia para el desarrollo profesional, la mejora de competencias y la adaptación a los diferentes puestos de trabajo de la organización. Sin embargo, la gestión de las actividades formativas puede resultar compleja debido a la cantidad de datos involucrados, la necesidad de coordinar distintos perfiles de actuación y la supervisión del proceso en todas sus etapas.

Este trabajo tiene como objetivo desarrollar una aplicación web que facilite la gestión integral de los cursos de formación, desarrollando las actividades en 3 roles diferenciados: gestores, formadores y participantes.

Descriptores

Gestión de formación, Portal del participante, portal del gestor, portal del formador, asistencias, calificaciones, certificados, cursos, ediciones, symfony, php.

Abstract

The continuous training of a company's employees is of vital importance for professional development, skills improvement and adaptation to the different jobs in the organisation. However, the management of training activities can be complex due to the amount of data involved, the need to coordinate different profiles and the supervision of the process in all its stages.

This work aims to develop a web application that facilitates the integral management of training courses, developing the activities in 3 different roles: managers, trainers and participants.

Keywords

Training management, Participant portal, Manager portal, Trainer portal, Attendance, Qualifications, Certificates, Courses, Editions, Symfony, php.

Índice general

Tabla de contenido

<i>Índice general</i>	<i>III</i>
<i>Índice de figuras</i>	<i>V</i>
<i>Índices de tablas</i>	<i>VI</i>
1. Introducción	1
1.1. Estructura de la memoria	2
1.2. Materiales adjuntos	2
2. Objetivos del proyecto	3
2.1. Objetivos generales	3
2.2. Objetivos técnicos	3
2.3. Objetivos personales	4
3. Conceptos teóricos	6
3.1. Gestión de usuarios: roles y funcionalidades	6
3.2. Gestión de cursos y ediciones	6
3.3. Funcionalidades del formador	8
3.4. Reglas para los participantes	8
3.5. Funcionalidades del rol admin	8
4. Técnicas y herramientas	10
4.1. Metodologías de desarrollo	10
4.2. Herramientas de desarrollo	11
4.3. Gestión de dependencias	12
4.4. Control de versiones	13
4.5. Seguridad	13
4.6. Otras herramientas	13
5. Aspectos relevantes del desarrollo del proyecto	15
6. Trabajos relacionados	16
7. Conclusiones y Líneas de trabajo futuras	17

7.1.	Conclusiones	17
7.2.	Líneas de trabajo futuras	17
<i>Bibliografía</i>.....		18

Índice de figuras

Figura 3.1: Imagen modelo.....	9
--------------------------------	---

Índices de tablas

Tabla 4.1 Tabla modelo	14
------------------------------	----

1. Introducción

La formación continua es un elemento clave para el desarrollo profesional de los trabajadores y la mejora de la competitividad de las empresas en un mercado laboral cada vez más dinámico y exigente.

La capacidad de adaptación y actualización de los empleados resulta fundamental para cubrir las necesidades de los diferentes puestos de trabajo de una organización. No obstante, la gestión de las actividades formativas supone un reto importante debido a la cantidad de información involucrada y a la necesidad de coordinar distintos perfiles de actuación en todas las etapas del proceso: planificación, ejecución y evaluación.

Actualmente, en muchas organizaciones, la gestión de los cursos de formación se realiza mediante herramientas no especializadas. Esto puede derivar en problemas como duplicidad de datos, dificultades en la supervisión del avance de los cursos, pérdida de información relevante o un uso ineficiente de los recursos. La falta de una herramienta específica que permita organizar y automatizar estas tareas genera una sobrecarga de trabajo y ralentiza la toma de decisiones por parte de los responsables de formación.

Con este planteamiento, surge la necesidad de desarrollar una solución tecnológica que permita simplificar y optimizar la gestión de las actividades formativas.

En este trabajo se propone el desarrollo de una aplicación web que integre y organice todo el proceso de formación continua en una empresa, permitiendo la gestión eficiente de cursos, inscripciones, participantes y formadores.

La aplicación está estructurada en tres portales diferenciados que responden a las necesidades de los distintos perfiles involucrados en el proceso:

- **Gestores:** responsables de la planificación y administración de cursos y ediciones, gestión de inscripciones, supervisión del avance de la formación, gestión de la documentación de los formadores y sus retribuciones, así como de las certificaciones de los cursos impartidos.
- **Formadores:** encargados de registrar la asistencia, calificaciones y detalles relacionados con las sesiones de las ediciones asignadas.
- **Participantes:** trabajadores que pueden consultar el estado de su ficha formativa, inscribirse en cursos disponibles, visualizar su progreso y consultar los próximos cursos por comenzar en los que están inscritos.

La principal ventaja de esta herramienta es la automatización de tareas repetitivas y la centralización de la información, lo que permite un importante ahorro de tiempo y recursos, facilitando la supervisión del proceso formativo en tiempo real y ofreciendo funcionalidades adaptadas a cada rol, mejorando en ello la experiencia de uso y la eficacia del proceso de formación.

Con esta herramienta, se pretende resolver las dificultades actuales en la gestión de formación continua, ofreciendo una solución accesible, eficiente y escalable para organizaciones de cualquier tamaño.

1.1. Estructura de la memoria

La memoria sigue la siguiente estructura:

- **Introducción:** breve descripción del proyecto, estructura de la memoria y listado de materiales adjuntos.
- **Objetivos del proyecto:** descripción de los objetivos que persigue el proyecto.
- **Conceptos teóricos:** breve explicación de los conceptos necesarios para la realización del proyecto.
- **Técnicas y herramientas:** Descripción de las metodologías y herramientas que han sido utilizadas para llevar a cabo el proyecto.
- **Aspectos relevantes del desarrollo:** aspectos a destacar a lo largo de la realización del proyecto.
- **Trabajos relacionados:** resumen de trabajos y proyectos ya realizados en el campo del proyecto en curso.
- **Conclusiones y líneas de trabajo futuras:** Conclusiones obtenidas al finalizar el proyecto y posibles ideas de continuidad.

1.2. Materiales adjuntos

Los materiales adjuntos a la memoria son:

- Dirección del repositorio del proyecto.
- Dirección del portal de gestión ágil usado en el desarrollo.
- Dirección web del despliegue del proyecto.
- Dirección del video con demostración funcional de la aplicación.
- Dirección del video con descripción del proyecto.

2. Objetivos del proyecto

En este apartado abarcaremos los objetivos generales, técnicos y personales del proyecto

2.1. Objetivos generales

- a) Desarrollar una aplicación web que permita gestionar cursos de formación, facilitando la administración de ediciones, inscripciones, y recursos asociados.
- b) Implementar un solo portal con funcionalidades específicas para tres roles: gestor, participante y formador.
- c) Facilitar la inscripción de participantes en las ediciones de cursos y el seguimiento de su progreso académico.
- d) Proporcionar a los formadores una herramienta para gestionar sesiones de formación, asistencia y calificaciones.
- e) Garantizar un acceso seguro a la aplicación mediante autenticación y control de roles.
- f) Almacenar y organizar toda la información de los cursos, ediciones, participantes y formadores en una base de datos relacional.

2.2. Objetivos técnicos

- a) Desarrollar la aplicación utilizando el framework Symfony 7, que sigue el patrón arquitectónico Modelo-Vista-Controlador (MVC) para garantizar una separación clara de responsabilidades.
- b) Utilizar PHP 8.2 como lenguaje principal y MySQL 8 como sistema de gestión de bases de datos relacionales.
- c) Implementar la autenticación de usuarios utilizando un sistema de registro con envío de correos de alta, de seguridad con almacenamiento hash seguro de contraseñas y control de roles de usuario.

- d) Desarrollar formularios con validaciones y mensajes de confirmación para la gestión de participantes, formadores y ediciones de cursos.
- e) Asegurar la calidad del código utilizando herramientas como PHPUnit para la realización de pruebas unitarias y PHPStan para el análisis estático del código.
- f) Integrar un sistema de control de versiones mediante GitHub, aplicando buenas prácticas de desarrollo con ramas diferenciadas por issues, integración en rama developer y pull requests para la integración de producto final en rama main.
- g) Gestionar y automatizar procesos de despliegue mediante pipelines de CI/CD configurados en GitHub Actions.
- h) Utilización de Zube.io como herramienta de gestión ágil para planificar, organizar y dar seguimiento a las tareas del proyecto.
- i) Implementación de gestión de dependencias para la gestión de recursos frontend mediante AssetMapper.
- j) Implementar la base de datos relacional con relaciones como ManyToMany y OneToMany entre entidades clave.
- k) Despliegue de la aplicación en un entorno de pruebas para validar su funcionamiento.

2.3. Objetivos personales

- a) Consolidar los conocimientos adquiridos en el desarrollo de aplicaciones web con Symfony y en el uso del modelo MVC.
- b) Mejorar las habilidades en el diseño e implementación de bases de datos relacionales complejas con MySQL.
- c) Realización de pruebas unitarias y de integración al finalizar cada sprint, asegurando la funcionalidad del código implementado y mejorando su calidad mediante herramientas como PHPUnit y PHPStan.

- d) Familiarización con herramientas de gestión de proyectos como Zube.io y flujos de trabajo en GitHub.
- e) Desarrollo de un sistema web robusto y funcional que resuelva necesidades reales en la gestión de cursos de formación.
- f) Desplegar una aplicación funcional que pueda servir como base para futuros desarrollos.

3. Conceptos teóricos

En este apartado se exponen los conceptos teóricos del problema que han servido como base para el desarrollo del proyecto. Nos centraremos en la gestión de cursos de formación y sus ediciones, explicando los roles involucrados y las reglas funcionales que rigen la aplicación.

3.1. Gestión de usuarios: roles y funcionalidades

La aplicación distingue tres roles principales: **gestor**, **participante** y **formador**. Cada uno con funciones específicas y un portal propio que dan sentido a esta separación de roles entre los usuarios de la aplicación.

Participantes y formadores: La aplicación ofrece un **formulario de registro**, siendo esta la única forma de dar de alta a este tipo de usuarios con su rol específico, no siendo posible que el gestor dé de alta directamente a participantes ni formadores.

Activación de roles adicionales: En el caso de un usuario existente (participante o formador), el gestor tiene la capacidad de **activar el rol complementario** (es decir, convertir a un participante en formador o viceversa) para dotar al usuario de funcionalidades completas, pero no puede crearlo si no existe un alta previa.

3.2. Gestión de cursos y ediciones

Los **cursos** y sus **ediciones** son los elementos centrales en la gestión de la formación.

Definición de un curso.

Un curso es asimilable a la guía docente de una asignatura, compuesta por información general como:

- **Código del curso:** Cada curso cuenta con un código único de **5 dígitos**, donde:
 - Los **dos primeros dígitos** corresponden al **año del curso**.
 - Los **tres últimos dígitos** son generados de manera **incremental** (por ejemplo, 24001 para el primer curso del año 2024).

- Nombre del curso.
- Objetivos y justificación de la actividad formativa.
- Contenidos del curso.
- Duración y número de participantes.

Los cursos tienen 2 campos de gran importancia que definen su formato de impartición:

- Horas: Son las horas totales de duración del curso.
- Horas virtuales: Este campo es el que marca el **modelo de impartición** del curso.
 - Si el curso no tiene horas virtuales estamos declarando un curso **presencial**.
 - Si el número de horas virtuales es inferior a las horas del curso estamos hablando de un curso **semipresencial**.
 - Si las horas virtuales coinciden con las horas del curso estamos ante un curso **virtual**.

En el contexto de este proyecto, el formato de impartición influye significativamente en la naturaleza del curso. Por defecto, todos los cursos que incluyen **horas virtuales** serán **evaluables**, a diferencia de los cursos presenciales, que no lo son necesariamente. La activación de este campo, aunque **puede desactivarse posteriormente**, implica que en estos cursos se medirán los conocimientos adquiridos por los participantes mediante pruebas o actividades específicas.

Definición de una edición.

Las ediciones representan las distintas instancias o ejecuciones de un mismo curso. Todas las ediciones **comparten propiedades generales**, como el número de horas, las horas virtuales y su carácter evaluable. Además, cada edición incluye datos específicos, como:

- **Código de la edición:** Son generados de forma automática y correlativa, y está formado por un código único de **8 caracteres**, donde:
 - Los 5 primeros caracteres corresponden con el **código del curso**.
 - Le sigue una **barra inclinada** que separa el código del curso de la edición.
 - Los dos últimos caracteres son un **número incremental** que identifican la edición (24001/01, 24001/02, etc.).
- Fechas de inicio y fin.
- Lugar de impartición.
- Calendario y horario
- Máximo de participantes, que este si será un dato propio de cada edición.

En cada curso existe una edición especial con **código 00**. A esta edición no se le permite asignar formadores, fecha de inicio y fin, calendario...ya está pensada como una base para posibles **funcionalidades futuras**, como listas de espera de participantes o inscripciones para cursos de oficio.

3.3. Funcionalidades del formador

Los formadores desempeñan un papel clave en la gestión de las ediciones al tener el encargo de realizar las siguientes tareas:

- **Crear y gestionar las sesiones** (días de impartición de una edición) de las ediciones asignadas, utilizando como base el calendario y horario definidos previamente.
- **Registrar asistencia y calificaciones:** Los formadores deben garantizar la correcta introducción de los datos de sesiones, asistencias y calificaciones (en caso de cursos evaluables).
- **Remitir datos:** Una vez completada la información necesaria, el formador puede remitir los datos para cerrar la edición.

Cuando una edición ha sido cerrada por el formador, el gestor tiene la posibilidad de:

- **Certificar la edición**, validando los datos introducidos.
- **Emitir los certificados** correspondientes para los participantes que cumplan los criterios establecidos (porcentaje de asistencia y calificaciones si las hubiera).

3.4. Reglas para los participantes

Los participantes tienen ciertas **restricciones y condiciones** en la gestión de sus inscripciones:

- Solo pueden realizar una inscripción en aquellas **ediciones que aún no hayan comenzado**.
- Si está inscrito en una **edición que ya ha iniciado**, un participante no puede anular su inscripción. En su lugar, debe solicitar una **baja justificada** a través de los mecanismos definidos en la aplicación.
- Un participante no puede inscribirse en dos **ediciones del mismo curso** simultáneamente.
- La posibilidad de realizar inscripciones en las ediciones depende de que el campo "**unidad**" del participante esté relleno. Si este campo está **vacío**, el **usuario** se considera **inactivo** y no puede realizar inscripciones.

3.5. Funcionalidades del rol admin

El rol **admin**, desempeñado por el personal del departamento de formación, es el núcleo operativo de la aplicación.

Desde el **Portal del Gestor**, los administradores tienen acceso a las funcionalidades clave que permiten gestionar todos los aspectos relacionados con los cursos, ediciones, inscripciones, formadores y participantes. Su papel es esencial para garantizar el correcto funcionamiento del sistema y el cumplimiento de los objetivos de la aplicación. Por ejemplo:

- Los administradores son responsables de la creación y configuración de los cursos en el sistema, indicando si el curso será visible a los usuarios o no, si será calificable, su duración y modalidad.
- Deberán crear las distintas ediciones y asignarlas a los cursos a los que pertenecen. Indicar el número de participantes que se pueden inscribir, su calendario, horario y lugar entre otros datos.
- Gestionar las inscripciones de los participantes a las distintas ediciones, sus bajas justificadas y emitir las certificaciones que correspondan.
- Asignar los formadores a las distintas ediciones de los cursos creados, según su experiencia y disponibilidad. Registrando la recepción de documentación necesaria del formador, como datos bancarios o justificantes, las horas impartidas de clases y su retribución, si corresponde.
- Facilitando la actualización y el mantenimiento de los datos de los formadores y participantes.



Figura 3.1: Imagen modelo

4. Técnicas y herramientas

En esta sección se describen las principales metodologías y herramientas empleadas durante el desarrollo del proyecto, abarcando tanto aspectos técnicos como organizativos. Cada herramienta o técnica se detalla en función de su propósito y del uso específico dentro del proyecto.

4.1. Metodologías de desarrollo

- **Gestión ágil con Scrum:**

Se adoptó la metodología ágil Scrum para estructurar el trabajo del proyecto en ciclos iterativos llamados sprints, con una duración de 14 días cada uno. Al inicio de cada sprint, se definieron las historias de usuario a implementar, permitiendo establecer los objetivos considerados más importantes para cada iteración.

Esta metodología facilitó la adaptación a los cambios en los requisitos y la priorización de tareas según su importancia y transcendencia que tenían en el resto de las tareas que se debían implementar posteriormente.

La planificación y el seguimiento de las tareas se realizaron mediante la herramienta Zube.io, que incluye un tablero Kanban y se integra con el repositorio de GitHub, permitiendo un control eficiente del progreso.

Para estructurar y organizar las tareas dentro de cada sprint, se crearon épicas en Zube.io, agrupando las historias de usuario relacionadas. Esto permitió una visión clara de los objetivos de cada sprint y facilitó la priorización de tareas. Además, en GitHub se configuraron milestones para tener una representación visual y centralizada del progreso en cada sprint, alineando la gestión de código con la planificación ágil.

Al finalizar cada sprint, se llevaron a cabo las implementaciones de los tests unitarios y el análisis estático del código para evaluar la calidad del código implementado, así como una revisión del sprint para evaluar el avance de las historias de usuario completadas y reajustar las prioridades para los siguientes ciclos.

- **Integración y Despliegue Continuo (CI/CD):**

Se configuraron pipelines en GitHub Actions para garantizar la calidad del código mediante la ejecución automática de pruebas unitarias (PHPUnit). Estas herramientas aseguraron que cada cambio en el código fuera probado antes de ser integrado en la rama principal.

4.2. Herramientas de desarrollo

- **Symfony 7:**

El framework Symfony es el núcleo del desarrollo, utilizado para implementar las funcionalidades del backend y elegido por su robustez, escalabilidad y adherencia al patrón Modelo-Vista-Controlador (MVC). Simplificó la creación y validación de formularios, así como la gestión de entidades mediante Doctrine ORM. La autenticación de usuarios y asignación de roles se implementó de forma sencilla y segura utilizando sus herramientas nativas.

Symfony 7 adopta el patrón Modelo-Vista-Controlador (MVC), que permite separar las responsabilidades en la lógica de negocio (Modelo), la gestión de solicitudes (Controlador) y la presentación de la información (Vista). Este enfoque estructurado facilitó el mantenimiento del código y la implementación de nuevas funcionalidades

- **PHP 8.2:**

El proyecto utilizó PHP 8.2, aprovechando sus características modernas, como:

- Tipos estrictos: Asegurando la coherencia de los datos en funciones y métodos.
- Expresiones match: Simplificando decisiones condicionales.

PHP 8.2 contribuyó a mejorar el rendimiento y la legibilidad del código.

- **MySQL 8:**

MySQL se empleó como sistema de gestión de bases de datos relacional, ideal para manejar las entidades principales del proyecto (cursos, ediciones, participantes, formadores, entre otros).

- Esquema relacional: Se diseñaron relaciones OneToMany y ManyToMany para modelar las interacciones entre los datos.
- Optimización: La base de datos fue configurada para soportar consultas eficientes, clave en la gestión de grandes volúmenes de datos.

- **Doctrine ORM:**

Se empleó Doctrine para el mapeo objeto-relacional, facilitando la interacción con la base de datos, la definición de relaciones entre entidades y reduciendo la necesidad de escribir consultas SQL manuales.

Mediante las migraciones, Doctrine permitió gestionar cambios en el esquema de la base de datos de manera controlada.

- **PHPUnit:**

Se implementaron pruebas unitarias con PHPUnit para validar la funcionalidad de controladores, formularios y entidades, garantizando la estabilidad y fiabilidad del sistema.

PHPUnit fue una herramienta clave en el proyecto, integrándose con GitHub Actions para ejecutar automáticamente las pruebas en cada commit y en cada Pull Request hacia la rama principal.

- **PHPStan:**

El análisis estático de código con PHPStan fue clave para detectar errores e inconsistencias, contribuyendo a mejorar la calidad del código antes de su despliegue.

Se configuró un nivel alto de análisis, lo que permitió identificar problemas de tipo, métodos no utilizados y errores comunes. El feedback proporcionado por PHPStan ayudó a mantener un código limpio y robusto.

- **Sequel Ace:**

Sequel Ace se utilizó para administrar visualmente la base de datos.

- Pruebas manuales: Facilitó la validación y corrección de datos durante el desarrollo.
- Gestión de esquemas: Simplificó la creación y edición de tablas.

4.3. Gestión de dependencias

- **Composer:**

Se utilizó Composer para la gestión de dependencias del proyecto, asegurando que todas las librerías estuvieran actualizadas y compatibles con Symfony.

- **AssetMapper:**

En lugar de herramientas tradicionales como npm o yarn, se utilizó AssetMapper para la gestión de recursos frontend, integrando estilos y scripts de forma eficiente en el entorno de Symfony.

4.4. Control de versiones

- **Git y GitHub:**

El proyecto fue gestionado mediante Git, con un flujo de trabajo basado en ramas (developer para desarrollo y main para producción).

Cada historia de usuario se implementó en una rama propia, derivada de la rama developer. Una vez completado el desarrollo, los cambios realizados en la rama fueron integrados en developer.

Se realizaron pull requests para integrar cambios y asegurar revisiones de calidad antes de la fusión con la rama principal.

4.5. Seguridad

- **Autenticación y roles:**

El sistema de seguridad de Symfony se utilizó para implementar autenticación segura y asignación de roles (ROLE_USER, ROLE_ADMIN, ROLE_TEACHER).

Las contraseñas de los usuarios se almacenan de forma segura mediante hashing con algoritmos modernos como bcrypt.

4.6. Otras herramientas

- **PHPStorm:**

PHPStorm fue el IDE principal, elegido por sus características avanzadas:

- Soporte para Symfony: Atajos y herramientas específicas para trabajar con el framework.
- Depuración: Integración con Xdebug para identificar errores en tiempo de ejecución.

- **Xdebug:**

Xdebug se utilizó para depurar y generar reportes de cobertura de código en PHPUnit.

- Depuración en profundidad: Ayudó a identificar problemas específicos en la lógica de negocio.
- Cobertura de pruebas: Permitió visualizar qué partes del código estaban siendo probadas.

Tabla 4.1 Tabla modelo

5. Aspectos relevantes del desarrollo del proyecto

I

6.Trabajos relacionados

7. Conclusiones y Líneas de trabajo futuras

Texto general

7.1. Conclusiones

7.2. Líneas de trabajo futuras

Bibliografía
