



Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Departamento de Estadística, Informática y Matemáticas

Trabajo de fin de máster

Ecosistema de Aprendizaje: Innovación en la formación corporativa

Autor:
Yasin Ez-ziani Chakir

Supervisor:
Ugaitz Amozarrain Perez

Pamplona, 2022

Tu dedicatoria, opcional

Índice general

Índice de figuras	III
Índice de tablas	V
Listings	VI
1. Introducción	1
2. Análisis de requisitos	3
2.1. Requisitos funcionales	3
2.2. Requisitos no funcionales	4
2.3. Roles	4
3. Tecnologías y herramientas utilizadas	7
3.1. Lenguaje de programación y ecosistema .NET	7
3.2. Interfaz de usuario	8
3.3. Gestión de datos: Entity Framework y SQL Server	9
3.4. Sistema de notificaciones y comunicaciones	10
4. Estructura del proyecto	11
4.1. Presentación	12
4.2. Aplicación	14
4.3. Infraestructura	15
4.4. Dominio	15
5. Desarrollo del Ecosistema de Aprendizaje	17
5.1. Acceso a plataforma de aprendizaje	18
5.2. Acceso directo a la plataforma de ciberseguridad	19

5.3. Solicitar nueva acción formativa	19
5.4. Solicitar incluirse en una acción formativa ya creada	28
5.5. Clonación y valoraciones	29
5.5.1. Información para responsables y directores	29
5.6. Planes de formación y propuestas	31
6. Funcionamiento interno	37
6.1. Arquitectura técnica y organización del proyecto	37
6.2. Implementación de consultas (Query Handlers)	38
6.3. Implementación de comandos (Command Handlers)	40
6.4. Acceso a datos e infraestructura	43
6.5. Sistema de permisos y filtros de autorización	46
6.6. Sistema de notificaciones y enlaces únicos	48
6.7. Procesamiento de encuestas y valoraciones	49
6.8. Conclusión técnica	49
Bibliografía	51
Índice alfabético	53
Apéndice	53

Índice de figuras

4.1. Diagrama de una arquitectura por capas.	11
5.1. Pantalla home con los accesos directos.	18
5.2. Pantalla plataforma de aprendizaje.	18
5.3. Plataforma de ciberseguridad.	19
5.4. Cuestionario INTAKE para solicitar formación.	20
5.5. Formulario de datos generales a llenar.	20
5.6. Ventana para seleccionar propuesta a asignar a la formación.	22
5.7. Pestañas que cargan después de crear una formación.	22
5.8. Pestaña de los docentes de la formación.	23
5.9. Seleccionar asistentes para enviarles envío de convocatoria.	23
5.10. Página para llenar encuesta de eficacia formativa.	24
5.11. Pestaña de valoraciones de una formación.	24
5.12. Resultado de una encuesta realizada a una formación.	25
5.13. Pestaña de calendario de una formación.	25
5.14. Pestaña con el histórico de cambios en una formación.	26
5.15. Pestaña de gestión de una formación.	26
5.16. Listado de pedidos para asignar a una formación.	27
5.17. Pestaña de gestión de una formación con pedidos asignados.	27
5.18. Pestaña de solicitudes de inscripción pendientes de aprobar.	28
5.19. Pantalla de HOME con las formaciones ya tramitadas sin comenzar.	28
5.20. Botones de clonación de la formación.	29
5.21. Flujo de aprobación y notificaciones de las acciones formativas, incluyendo rutas de rechazo.	31
5.22. Pantalla con el listado de planes de formación anual.	32
5.23. Pantalla con el formulario para crear un plan de formación anual.	32
5.24. Pantalla de edición del plan de formación anual con todas las pestañas.	33

5.25. Pantalla de departamentos del plan de formación anual.	33
5.26. Usuarios con permiso de un departamento del plan de formación anual.	33
5.27. Pantalla con el listado de las propuestas de plan de formación anual.	34
5.28. Pantalla con el formulario para crear una propuesta del plan de formaciones.	34
5.29. Pantalla con el listado de las propuestas de plan de formación anual con una fila seleccionada.	35
5.30. Pantalla con el listado de las propuestas de plan de formación anual con dos filas seleccionadas.	35
5.31. Pestaña de asistentes dentro de la propuesta del plan de formaciones.	36
5.32. Pestaña de propuestas unificadas dentro de la propuesta del plan de formaciones.	36
5.33. Pestaña de formaciones creadas que tienen asignada la propuesta del plan de formaciones.	36
 ApéndiceA ApéndiceB ApéndiceC ApéndiceD ApéndiceE ApéndiceF ApéndiceG ApéndiceH ApéndiceI ApéndiceJ ApéndiceK ApéndiceL ApéndiceM ApéndiceN ApéndiceO ApéndiceP ApéndiceQ ApéndiceR ApéndiceS ApéndiceT ApéndiceU ApéndiceV ApéndiceW ApéndiceX ApéndiceY ApéndiceZ	 53 53 54 54 54 54 55 55 55 55 55 55 56

Índice de tablas

Listings

4.1.	Controlador del proyecto WebUI con métodos de selección de formaciones almacenados en base de datos.	12
4.2.	Interfaz de formaciones con métodos para llevar a cabo las operaciones sobre la base de datos.	14
6.1.	Clase Handler que realiza una consulta para obtener formaciones.	38
6.2.	Clase DTO para filtrar las formaciones.	40
6.3.	Clase Handler para crear una formación.	40
6.4.	Clase Validator para crear una formación.	43
6.5.	Mapeo de tablas mediante Fluent API.	44
6.6.	Mapeo de tablas mediante Fluent API.	44
6.7.	Clase que valida el permiso del usuario para poder consultar vistas o realizar acciones.	46

1

Introducción

En la actualidad, las organizaciones se encuentran inmersas en un entorno empresarial en constante transformación, donde la gestión del conocimiento y el desarrollo de competencias profesionales se han convertido en factores estratégicos para garantizar la competitividad y la sostenibilidad a largo plazo. La capacidad de formar a los equipos de manera ágil, estructurada y alineada con los objetivos corporativos es un desafío clave para cualquier empresa que aspire a mantener una posición destacada en su sector.

En este contexto, surge el presente Trabajo de Fin de Máster, que tiene como propósito el diseño, desarrollo e implementación de la aplicación web Ecosistema de Aprendizaje en colaboración con Tracasa Global [1] y Tracasa Instrumental [2]. Esta solución tecnológica se concibe como una plataforma integral destinada a optimizar los procesos de gestión del aprendizaje, permitiendo a los trabajadores acceder a un entorno digital centralizado desde el cual pueden solicitar nuevas acciones formativas, inscribirse en programas ya existentes y acceder a diferentes recursos de capacitación externos.

La aplicación no solo buscará facilitar el acceso a la formación, sino también incorporar un sistema de validación y aprobación por parte de responsables y directores, garantizando así la coherencia entre las necesidades individuales de los empleados y las prioridades estratégicas de la empresa. Además, se prevé que incluya funcionalidades avanzadas como la planificación de actividades, el seguimiento del progreso y la valoración de la eficacia formativa, con el fin de contribuir a una mayor trazabilidad y transparencia en los procesos de desarrollo profesional.

El desarrollo del proyecto se estructurará en distintas fases claramente definidas. En primer lugar, se llevará a cabo una fase de análisis de requisitos, en la que se recopilarán las necesidades funcionales y técnicas de la empresa, identificando los procesos clave a digitalizar y los distintos perfiles de usuario que interactuarán con la aplicación (empleados, responsables, directores y administradores del sistema). A partir de esta información se definirán los requisitos funcionales principales, como la creación y gestión de acciones formativas, la validación de solicitudes, la integración con plataformas externas de aprendizaje, la generación de informes de seguimiento y la automatización de notificaciones por correo.

Posteriormente, se realizará la fase de diseño, en la que se definirá la arquitectura de la solución siguiendo un modelo por capas, garantizando la separación de responsabilidades y la escalabilidad del sistema. En esta etapa se elaborarán los diagramas de flujo, los modelos de datos y la estructura de componentes que servirán como base para la fase de implementación.

Durante la fase de desarrollo, se procederá a construir los distintos módulos de la aplicación, que se verá en la sección de *Desarrollo del Ecosistema de Aprendizaje*, utilizando tecnologías modernas del ecosistema .NET [3] y C# [4], junto con TypeScript [5] y HTML[6] / CSS[7] para la capa de presentación. Se aplicará el patrón arquitectónico *Command Query Responsibility Segregation (CQRS)* [8] para separar las operaciones de lectura y escritura, asegurando así un código más limpio y mantenible.

Una vez implementada la base del sistema, se llevará a cabo la fase de pruebas, en la que se realizarán tanto pruebas unitarias como de integración [9], además de validaciones funcionales con usuarios de la empresa colaboradora. Este proceso permitirá detectar posibles errores, ajustar la experiencia de usuario y garantizar la fiabilidad del sistema antes de su despliegue.

Finalmente, la fase de despliegue y entrega contemplará la puesta en marcha de la aplicación en el entorno corporativo, la configuración de las bases de datos y servidores, así como la formación de los usuarios en el uso de la herramienta.

Se estima que el desarrollo completo del proyecto se extenderá a lo largo de aproximadamente 1 año, combinando sesiones de planificación y seguimiento con el equipo técnico de Tracasa y el área de gestión de proyectos tecnológicos (PyT). Para ello, se adoptará una metodología ágil de trabajo, que permitirá iterar de manera continua sobre los avances, incorporar mejoras basadas en el feedback de los usuarios y priorizar las funcionalidades más relevantes para cada perfil.

En definitiva, este proyecto representa una apuesta firme por la innovación y la transformación digital en el ámbito de la formación corporativa. La implementación del Ecosistema de Aprendizaje permitirá a Tracasa Global y Tracasa Instrumental fortalecer su modelo de gestión del talento, fomentar el aprendizaje continuo y consolidar una cultura organizativa orientada al crecimiento y a la excelencia.

2

Análisis de requisitos

Antes de abordar el desarrollo y la estructura técnica de la aplicación, resulta fundamental realizar un análisis de los requisitos funcionales y no funcionales que definirán el alcance del Ecosistema de Aprendizaje. Este análisis tiene como objetivo identificar las necesidades específicas de la organización, los usuarios involucrados y las funcionalidades esenciales que el sistema deberá cubrir para cumplir con los objetivos planteados.

2.1. Requisitos funcionales

Los requisitos funcionales describen las acciones y procesos que la aplicación deberá permitir realizar a sus usuarios. A partir de las reuniones iniciales con Tracasa Global y Tracasa Instrumental, y del estudio de los flujos actuales de gestión de la formación, se han definido las siguientes funcionalidades principales:

Gestión de acciones formativas: permitir la creación, edición, eliminación y clonación de acciones formativas, distinguiendo entre formaciones planificadas y extraordinarias.

Gestión de solicitudes: permitir que los empleados soliciten nuevas formaciones o su participación en acciones ya existentes, con validación por parte de responsables, directores y el área de PyT (Proyectos y Tecnología).

Flujo de aprobación y notificaciones: automatizar el envío de correos electrónicos para informar a los distintos roles del estado de las solicitudes, aprobaciones y valoraciones pendientes.

Gestión de docentes y asistentes: permitir asociar docentes internos o externos a una formación, así como gestionar los asistentes y sus convocatorias.

Encuestas y valoraciones: generar encuestas de eficacia formativa y registrar valoraciones asociadas a cada acción formativa.

Planes de formación anual: posibilitar la creación, edición y seguimiento de planes de formación, con la asignación de propuestas por departamento y la gestión de permisos para cada usuario.

Integración con plataformas externas: permitir el acceso directo a la plataforma de aprendizaje GoodHabitz y a la plataforma de ciberseguridad corporativa desde la propia aplicación.

Histórico y trazabilidad: registrar los cambios y actualizaciones realizados sobre las acciones formativas para mantener un seguimiento completo de la evolución de cada proceso.

2.2. Requisitos no funcionales

Además de las funcionalidades descritas, la aplicación deberá cumplir con una serie de condiciones relacionadas con su rendimiento, seguridad y usabilidad:

Usabilidad: la interfaz debe ser intuitiva, clara y accesible para usuarios con distintos niveles de experiencia técnica.

Seguridad: el acceso a la aplicación deberá estar autenticado y controlado mediante un sistema de roles (solicitante, responsable, director, PyT o administrador), garantizando la protección de los datos personales y formativos.

Rendimiento: el sistema debe ser capaz de gestionar múltiples solicitudes y consultas simultáneas sin afectar la velocidad de respuesta.

Escalabilidad: la arquitectura debe permitir la incorporación de nuevas funcionalidades o módulos sin afectar el funcionamiento existente.

Mantenibilidad: el código debe estar bien estructurado y documentado, siguiendo principios de modularidad y reutilización.

Compatibilidad: la aplicación debe ser accesible desde los navegadores corporativos más utilizados y adaptarse correctamente a diferentes resoluciones de pantalla.

2.3. Roles

Durante la fase de análisis se identificaron los diferentes tipos de usuarios que interactuarán con el sistema, cada uno con permisos y responsabilidades específicas:

Solicitante: puede solicitar nuevas acciones formativas o inscribirse en formaciones ya creadas.

Responsable de área: recibe y valida las solicitudes de los empleados de su equipo.

Director de departamento: revisa y aprueba las solicitudes validadas por los responsables de las áreas que forman parte de su departamento.

PyT/PyV (Proyectos y Tecnología): gestiona las solicitudes aprobadas, tramita las formaciones y supervisa el cumplimiento de los procesos.

Administrador del sistema: mantiene la aplicación, gestiona usuarios y configura los parámetros globales de funcionamiento.

3

Tecnologías y herramientas utilizadas

La elección de las tecnologías empleadas en el desarrollo del Ecosistema de Aprendizaje responde a la necesidad de construir una solución robusta, segura y con capacidad de crecer a largo plazo dentro del entorno corporativo de Tracasa Global y Tracasa Instrumental. En este sentido, se van a seleccionado herramientas consolidadas en el sector y ampliamente utilizadas en el desarrollo de aplicaciones empresariales, garantizando así un mantenimiento eficiente, una curva de aprendizaje asumible y una larga vida útil del sistema.

3.1. Lenguaje de programación y ecosistema .NET

Antes de profundizar en las tecnologías utilizadas en el desarrollo del backend, resulta pertinente introducir brevemente el concepto de **framework**, ya que constituye la base sobre la que se construye gran parte de la solución implementada.

Un *framework* [10] puede entenderse como un conjunto de herramientas, componentes y estructuras previamente definidas que facilitan la creación de aplicaciones de manera más ágil y eficiente. Su propósito es evitar que los desarrolladores deban partir desde cero en cada proyecto, proporcionando elementos reutilizables y mecanismos estandarizados para resolver problemas comunes. Gracias a ello, se reduce el tiempo de desarrollo, se mejora la calidad del software y se garantiza una mayor consistencia en la implementación.

En este contexto, la plataforma **.NET** constituye uno de los frameworks más completos y ampliamente adoptados en el ámbito empresarial. Se trata de una plataforma de código abierto y multiplataforma diseñada para compilar y ejecutar aplicaciones de todo tipo, desde soluciones web hasta servicios distribuidos o herramientas de escritorio. .NET destaca por su alto rendimiento, su sistema avanzado de ejecución y su amplio conjunto de bibliotecas optimizadas, que permiten trabajar de forma segura, productiva y escalable. Entre sus características clave se encuentran la administración automática de memoria, la seguridad de tipos, la protección de la memoria y la existencia de un extenso ecosistema de herramientas y marcos de trabajo especializados.

Sobre esta plataforma se apoya el lenguaje **C#**, utilizado como lenguaje principal en el desarrollo del Ecosistema de Aprendizaje. C# es un lenguaje moderno, fuertemente

tipado y orientado a objetos, diseñado para ofrecer productividad y claridad en el código, además de funciones avanzadas como la concurrencia integrada, el manejo automático de recursos y una sintaxis elegante fácil de mantener a largo plazo. Su interoperabilidad con otros lenguajes del ecosistema .NET y su alta fiabilidad lo han convertido en la opción predilecta dentro del entorno profesional cuando se requieren aplicaciones robustas, seguras y de gran rendimiento.

La elección conjunta de C# y .NET proporciona una base tecnológica sólida para el proyecto, especialmente en un entorno corporativo donde la seguridad, la escalabilidad y la mantenibilidad son requisitos fundamentales.

Entre los motivos que justifican el uso de esta tecnología destacan los siguientes:

- **Madurez y estabilidad:** .NET es una plataforma ampliamente consolidada y en constante evolución, lo que garantiza fiabilidad incluso en proyectos de larga duración.
- **Rendimiento elevado:** el entorno de ejecución de .NET ofrece tiempos de respuesta competitivos, ideales para sistemas corporativos que gestionan un gran volumen de solicitudes simultáneas.
- **Seguridad integrada:** la plataforma incorpora mecanismos nativos de protección, validación y control de acceso, esenciales en aplicaciones que manejan datos sensibles como los relativos a formación y gestión interna.
- **Productividad y facilidad de mantenimiento:** gracias al uso de C# y al acceso a bibliotecas optimizadas, el desarrollo es más ágil y el código más fácil de mantener.
- **Interoperabilidad y ecosistema amplio:** .NET permite integrar la solución con otros servicios y tecnologías corporativas, reduciendo riesgos y facilitando futuras ampliaciones.

En conjunto, el uso del ecosistema .NET y del lenguaje C# constituye una opción altamente adecuada para el desarrollo del Ecosistema de Aprendizaje, proporcionando un equilibrio óptimo entre rendimiento, seguridad, mantenibilidad y capacidad de evolución tecnológica.

3.2. Interfaz de usuario

La capa de presentación del Ecosistema de Aprendizaje se ha desarrollado combinando **TypeScript**, el motor de componentes **Blazor** y estilos definidos mediante **CSS global**. Este conjunto tecnológico permite construir interfaces modernas, dinámicas y fáciles de mantener, aprovechando tanto el ecosistema web tradicional como las capacidades integradas de .NET para el desarrollo de aplicaciones interactivas.

- **TypeScript** aporta tipado estático, una organización más clara del código y la detección temprana de errores durante el desarrollo. Esto reduce fallos en ejecución y mejora significativamente la mantenibilidad de la interfaz.

- **Blazor** facilita la creación de componentes interactivos empleando código familiar para el equipo de desarrollo, permitiendo integrar lógica de presentación de forma estructurada sin necesidad de depender exclusivamente de JavaScript. Su capacidad para trabajar con datos y eventos de manera fluida mejora la consistencia de la experiencia de usuario.
- **CSS** se utiliza para definir estilos unificados en toda la aplicación, garantizando coherencia visual, facilidad de personalización y una apariencia profesional adaptada al entorno corporativo.

Este conjunto de tecnologías permite construir una interfaz intuitiva, ligera y adaptada a los estándares actuales de usabilidad, asegurando que la experiencia de usuario resulte coherente para todos los perfiles que interactúan con la aplicación.

3.3. Gestión de datos: Entity Framework y SQL Server

El acceso a datos se ha implementado mediante **Entity Framework**, un *ORM* que simplifica la comunicación con la base de datos. Sus principales ventajas incluyen:

- **Reducción del código repetitivo**: permite abstraer consultas comunes sin necesidad de escribir SQL manual, disminuyendo la probabilidad de errores.
- **Sincronización continua del modelo de datos**: gracias al sistema de migraciones, es posible actualizar la base de datos de forma segura a medida que evoluciona el proyecto.
- **Facilidad de mantenimiento**: su integración natural con C# permite trabajar de forma coherente con las entidades del sistema.

Como motor de almacenamiento se ha utilizado **SQL Server** [11], una tecnología ampliamente extendida en entornos corporativos y que presenta ventajas significativas:

- **Potentes capacidades de administración**: SQL Server facilita la gestión de volúmenes grandes de información, el control de transacciones y la seguridad.
- **Escalabilidad**: permite afrontar un crecimiento progresivo del sistema sin necesidad de realizar cambios disruptivos en la infraestructura.
- **Integración con herramientas empresariales existentes**: lo que agiliza los procesos de despliegue y mantenimiento.

Estas características hacen que la combinación de Entity Framework con SQL Server sea particularmente adecuada para aplicaciones internas de gestión, donde la consistencia, seguridad y trazabilidad de los datos son elementos clave.

3.4. Sistema de notificaciones y comunicaciones

La plataforma incorpora un sistema automatizado de **envío de correos electrónicos**, fundamental para coordinar las aprobaciones, convocatorias, valoraciones y avisos entre usuarios. Su importancia radica en que:

- Reduce los tiempos de gestión.
- Evita pérdidas de información entre departamentos.
- Asegura una comunicación constante y centralizada.

La integración del sistema de notificaciones dentro de la lógica de la aplicación permite que el usuario reciba información en tiempo real, lo que incrementa la eficiencia de los flujos de trabajo relacionados con la formación.

En conjunto, todas estas tecnologías permiten construir una aplicación robusta, segura, fácil de mantener y preparada para evolucionar en el futuro. Su uso está alineado con las tendencias actuales en desarrollo de software corporativo, ofreciendo garantías de fiabilidad y sostenibilidad a largo plazo dentro del entorno empresarial de Tracasa.

4

Estructura del proyecto

El proyecto se estructura siguiendo una arquitectura por capas¹, lo que permite organizar el código y separar las responsabilidades de forma clara y eficiente. Cada capa cumple un papel distinto. La presentación se encarga de la interfaz y la interacción con el usuario, la aplicación contiene la lógica de negocio, la infraestructura gestiona el acceso a recursos externos como la base de datos y la capa de dominio define las reglas y entidades principales del modelo. La elección de esta arquitectura se debe a que proporciona una base sólida y escalable para el desarrollo, favorece la mantenibilidad del sistema y facilita la incorporación de nuevas funcionalidades sin afectar al resto de componentes. Además, permite una mayor reutilización del código y mejora la calidad del software al fomentar un diseño más limpio, modular y fácilmente testeable.

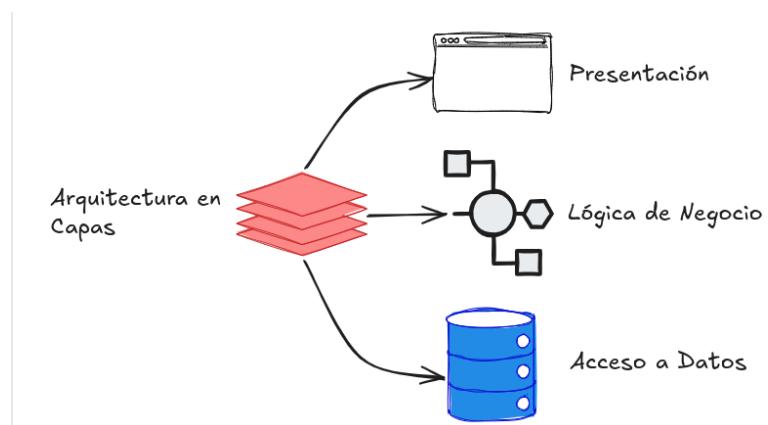


Figura 4.1: Diagrama de una arquitectura por capas.

¹<https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/capas>

4.1. Presentación

La capa de presentación constituye el nivel superior de la arquitectura y se encarga de gestionar la interacción directa con el usuario. En esta capa se encuentra el proyecto Interfaz de Usuario Web (*WebUI*), que corresponde a la aplicación web visible para los usuarios finales. Dentro de *WebUI* se definen las vistas, es decir, los elementos visuales que conforman la interfaz, junto con los estilos *CSS* que determinan su apariencia.

Asimismo, en esta capa se desarrollan las clases *TypeScript*, responsables de manejar los distintos eventos del usuario y de definir las acciones que se ejecutan como respuesta a dichos eventos. Los controladores también forman parte de esta capa y su función principal es invocar a las clases de niveles inferiores para realizar operaciones sobre la base de datos. Estas clases, situadas en capas más profundas, se explicarán más adelante. Se incluyen también los *ViewModels*, que son modelos diseñados específicamente para las vistas. Estos modelos facilitan la organización y presentación de los datos de forma estructurada y coherente, permitiendo que la información se muestre correctamente en la interfaz web. En esencia, actúan como una plantilla que define tanto la apariencia como el comportamiento de los datos dentro de la interfaz de usuario.

Un ejemplo ilustrativo puede observarse en el Código 4.1, donde se presentan varios métodos pertenecientes al controlador de la clase *Formación*. El primer método, que se invoca mediante una petición *POST*, devuelve un conjunto de formaciones almacenadas en la base de datos, permitiendo además aplicar filtros de búsqueda a través del parámetro *filtros*, en caso de que ocurra algún error y no se devuelven las formaciones se muestra un mensaje al usuario en pantalla. En cambio, el segundo método se ejecuta mediante una petición *GET*, que solicita un recurso específico del servidor y recupera una formación concreta identificada por su *ID* pasado como parámetro en la línea 27 (*FormacionID*), aquí también se hace una validación para comprobar el tipo de respuesta ya sea un error para informar al usuario, un aviso de que esa persona no está autorizada a ver los detalles de esa formación ó hay éxito que en ese caso se genera el *ViewModel* correspondiente y se le pasa los datos de la formación para poder mostrarlos por pantalla al usuario.

Además de los métodos de consulta, el controlador incluye también operaciones para crear, modificar y eliminar registros de formaciones. Otra nota importante aquí es cuando ponemos a los métodos `[TienePermisosFilter(EAutorizacionPermisos.SOLICITANTE)]`, esto nos valida el rol del usuario el cual llama al método. En este caso definimos que solamente aquellos que mínimo el rol de *Solicitante* podrán acceder a este método, aunque más tarde explicaremos con más detalle el tema de los roles y como funciona la validación de permisos.

```
1 [HttpPost("[action]")]
2 [TienePermisosFilter(EAutorizacionPermisos.SOLICITANTE)]
3 public async Task<IActionResult>
4     GetListadoFormaciones([DataSourceRequest]
5         DataSourceRequest request, FormacionFiltrosDTO filtros)
6     {
7         logger.LogInformation("Obtener el listado de formaciones")
```

```
      filtrado");

6     var formacionesRespuesta = await Mediator.Send(new
7         GetAllFormacionesQuery(filtros));
8
9     if (formacionesRespuesta.Tipo == TipoRespuesta.Error)
10    {
11        ModelState.AddModelError("ListadoGridError", "Ha
12            ocurrido un error al obtener el listado de
13            formaciones.");
14    }
15
16    return Json(formacionesRespuesta.Data.ToDataSourceResult
17        (request, ModelState));
18}

19 [HttpGet("[action]/[{FormacionID}]")]
20 [TienePermisosFilter(EAutorizacionPermisos.SOLICITANTE)]
21 public async Task<IActionResult> Detalle(int formacionID)
22 {
23     logger.LogInformation($"Pantalla de detalle de la
24         formacion: {formacionID}");
25
26     var formacion = await Mediator.Send(new
27         GetSingleFormacionQuery(formacionID));
28
29     if (formacion.Tipo == TipoRespuesta.Error)
30    {
31         TempData["NotificacionMensaje"] =
32             NotificacionMensaje.FormacionObtenerError
33             .ToString();
34         return RedirectToAction("Index");
35     }
36     else if (formacion.Tipo == TipoRespuesta.NoAutorizado)
37    {
38         return RedirectToAction("HttpStatusCodeHandler",
39             "Error", new { statusCode = 403 });
39     }
39
40     var viewModel = new FormacionViewModel();
41     viewModel.FormacionGeneral =
42         viewModel.FormacionGeneral.GetViewModel
```

```

40     (formacion.Data);
41
42     return View(viewModel);
43 }
```

Listing 4.1: Controlador del proyecto WebUI con métodos de selección de formaciones almacenados en base de datos.

4.2. Aplicación

La capa de aplicación representa el nivel en el que se concentra la lógica de negocio del sistema. En este nivel se implementa el principio de *Command Query Responsibility Segregation* (CQRS), mediante el cual cada proceso o acción empresarial se modela como un comando o una consulta independiente. Este patrón arquitectónico permite diferenciar las operaciones que modifican el estado del sistema (comandos) de aquellas que únicamente recuperan información (consultas), promoviendo así una estructura más ordenada, clara y escalable.

Dentro de esta capa también se definen las clases conocidas como *Data Transfer Objects* (DTO). Estas entidades sirven como vehículos de intercambio de información entre las diferentes capas, evitando que los modelos internos se expongan directamente. Su propósito es agrupar y transportar los datos necesarios para ejecutar una acción concreta, garantizando una comunicación sencilla, coherente y desacoplada entre componentes de la aplicación.

Finalmente, las operaciones relacionadas con los comandos y las consultas, como se muestra en el Código 4.2, se declaran mediante interfaces localizadas en esta misma capa, mientras que su implementación específica se desarrolla en la capa inferior.

```

1 public interface IFormacionesDataService
2 {
3     Task<List<FormacionListDTO>>
4         SelectAll(FormacionFiltrosDTO filtros,
5             CancellationToken cancellationToken);
6     Task<List<int>> SelectAreas(List<string> areas,
7         CancellationToken cancellationToken);
8     Task<bool> PerteneceFormacionArea(int formacionId,
9         List<string> areas, CancellationToken
10            cancellationToken);
11    Task<FormacionDTO> SelectSingle(int formacionID,
12        CancellationToken cancellationToken);
13    Task<bool> ExisteCodigo(string codigo, CancellationToken
14       cancellationToken);
15    Task<bool> ComprobarValoracionCompletada(int formacionID);
16    Task<FormacionDTO> Insert(FormacionDTO formacionDTO,
17        CancellationToken cancellationToken);
```

```
10     Task<FormacionDTO> Update(FormacionDTO formacionDTO,
11         CancellationToken cancellationToken);
12     Task<bool> Delete(int formacionID, CancellationToken
13         cancellationToken);
14 }
```

Listing 4.2: Interfaz de formaciones con métodos para llevar a cabo las operaciones sobre la base de datos.

4.3. Infraestructura

La capa de infraestructura constituye el nivel responsable de la interacción con los recursos externos y de proporcionar los servicios necesarios para que las demás capas puedan operar correctamente. En esta capa se ubican las clases encargadas de acceder a sistemas externos, tales como bases de datos, sistemas de archivos, servicios web o cualquier otro recurso externo que el sistema requiera.

Las clases que pertenecen a este nivel implementan las interfaces definidas en la capa de aplicación, cumpliendo así el principio de inversión de dependencias y garantizando una arquitectura más modular y desacoplada. Gracias a este enfoque, el resto del sistema no necesita conocer los detalles técnicos de cómo se realizan las operaciones externas, sino únicamente la interfaz que las define, lo que facilita la sustitución o actualización de componentes sin afectar al funcionamiento general.

Otro elemento importante presente en esta capa son los procesos de migración, que hacen referencia al conjunto de procedimientos destinados a trasladar o actualizar la estructura y los datos de una base de datos. Las migraciones permiten mantener la coherencia entre el modelo de datos y el código fuente del proyecto, especialmente durante las fases de desarrollo o actualización del sistema. Mediante estas operaciones se pueden aplicar cambios de esquema, incorporar nuevas tablas o modificar relaciones existentes, asegurando la integridad de la información y la compatibilidad entre versiones.

4.4. Dominio

La capa de dominio constituye el núcleo esencial del sistema, ya que en ella se define la lógica de negocio que modela los procesos y reglas fundamentales de la aplicación. En este nivel se ubican las entidades, que representan los elementos principales del modelo de negocio y encapsulan tanto sus atributos como el comportamiento asociado.

El propósito de esta capa es reflejar con fidelidad la realidad del problema que la aplicación pretende resolver, garantizando la coherencia interna y la integridad de los datos mediante las operaciones que se realizan sobre dichas entidades.

Una característica clave del dominio es su independencia respecto a la infraestructura y a los elementos externos, como bases de datos o servicios. Esta separación asegura que las reglas de negocio permanezcan estables y no se vean afectadas por los cambios tecnológicos, facilitando así la evolución y el mantenimiento del sistema a largo plazo.

Desarrollo del *Ecosistema de Aprendizaje*

La aplicación *Ecosistema de Aprendizaje* es un entorno digital que nos va a permitir, desde la organización, gestionar el aprendizaje y el desarrollo profesional de una manera más ágil y dinámica. A través de esta aplicación, se busca dar respuesta a la necesidad de disponer de un entorno digital unificado que permita a Tracasa Global y Tracasa Instrumental gestionar de forma eficiente los procesos de formación y desarrollo profesional de sus empleados. Este proyecto se enmarca en una estrategia de transformación digital más amplia, orientada a optimizar la gestión del talento y fomentar el aprendizaje continuo dentro de la organización. Desde el punto de vista funcional, la aplicación ofrecerá una interfaz intuitiva y accesible, compuesta por diferentes pantallas y formularios que permitirán a los usuarios interactuar con el sistema de manera sencilla. La herramienta persigue no solo digitalizar los procesos de solicitud y seguimiento de la formación, sino también mejorar la trazabilidad y la comunicación entre empleados, *managers* y responsables del área de proyectos tecnológicos. De este modo, se potencia una gestión más ágil, transparente y alineada con los objetivos estratégicos de la empresa.

La aplicación nos ofrece varias funcionalidades:

1. Posibilidad de cada persona pueda solicitar formación, que será validada por el *manager* y *PyT*.
2. Posibilidad de conocer qué acciones formativas se van a hacer en la empresa y solicitar incluirse en ella. También será validado por el *manager* y *PyT*.
3. Acceder a una plataforma de aprendizaje donde cada persona podrá desarrollarse en aquellas materias que puedan resultar de interés. Aquellas acciones que sean por interés personal y no estén relacionadas con el desempeño de su puesto, se realizarán fuera de jornada laboral.

Cuando accedemos a la aplicación, visualizamos la pantalla *HOME* que sale en la Figura 5.1, donde podremos elegir la acción que vayamos a realizar. En la Figura 5.1 aparecen marcadas con círculo rojo las acciones que hay.

1. Acceso a la plataforma de aprendizaje *GoodHabitZ*.
2. Acceso a la plataforma de ciberseguridad.
3. Solicitar una nueva acción formativa.
4. Visualizar las acciones previo a su comienzo y solicitar unirse a ella.

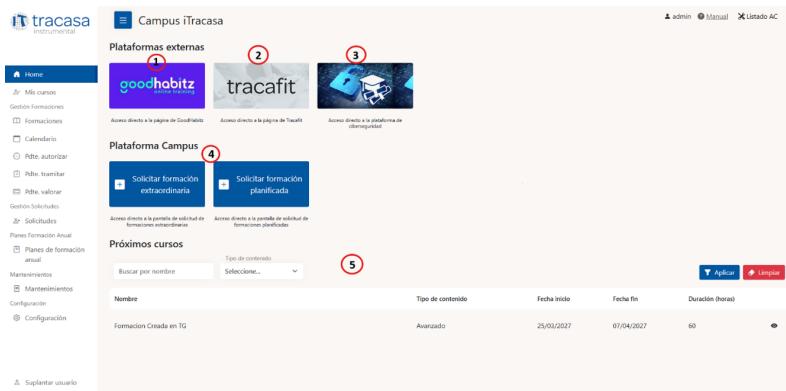


Figura 5.1: Pantalla home con los accesos directos.

5.1. Acceso a plataforma de aprendizaje

Para acceder a la plataforma de aprendizaje, que vemos en la Figura 5.2, debemos posicionarnos en la pantalla Home y seleccionar en GoodHabitZ Al acceder a la plataforma, por defecto, y sólo la primera vez, aparecerá un mensaje para aceptar la política de privacidad y acuerdo de usuario.

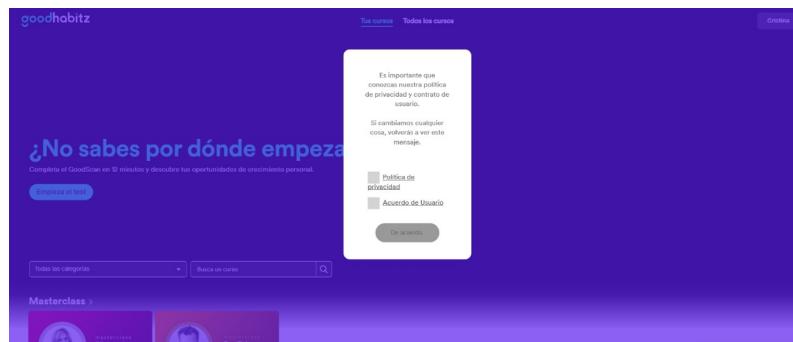


Figura 5.2: Pantalla plataforma de aprendizaje.

A continuación, encontraremos, en la parte superior de la pantalla, dos opciones a las que podremos acceder:

- Tus cursos: donde podremos encontrar aquellas acciones formativas que cada usuario haya realizado, esté realizando en ese momento o, también podemos encontrar aquellas formaciones que la empresa haya sugerido realizar
- Todos los cursos: podremos ver el catálogo de acciones formativas a las que podemos acceder y cursar.

5.2. Acceso directo a la plataforma de ciberseguridad

Acceso directo a contenidos para mejorar las capacidades en ciberseguridad tal y como se observa en la Figura 5.3

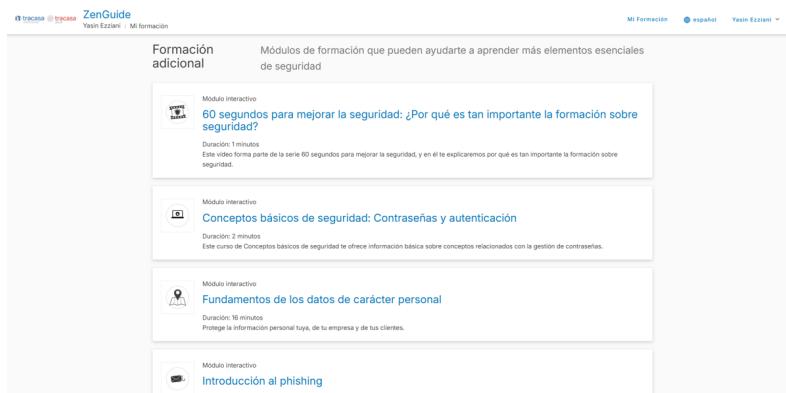


Figura 5.3: Plataforma de ciberseguridad.

5.3. Solicitar nueva acción formativa

Se proporcionan dos accesos para poder solicitar una formación. Podemos crear una formación extraordinaria o planificada. La diferencia que hay es que en el caso de la planificada debemos seleccionar una propuesta de plan de formación (que explicaremos más tarde) que asignaríamos a la formación que vamos a crear y para el caso de extraordinaria no se le asigna.

Al acceder a la opción de *Solicitar formación extraordinaria* encontraremos la pantalla mostrada en la Figura 5.4:

La captura de pantalla muestra la interfaz de usuario de la plataforma tracasa. En la parte superior izquierda, hay un menú lateral con opciones como 'Home', 'Mis cursos', 'Gestionar formaciones', 'Formaciones' (que está resaltado), 'Calendario', 'Pestaña autorizar', 'Pestaña transferir', 'Mis roles', 'Verificar documentación', 'Solicitudes', 'Planes formativos Anual', 'Planes de formación actual', 'Mantenimientos', 'Mantenimientos Configuración', 'Configuración' y 'Soporte al usuario'. La barra superior tiene iconos para 'admin', 'Manual' y 'Logout AC'. El título de la pantalla es 'Nueva formación' y el sub-título es 'Cuestionario INTAKE'. La sección 'Datos generales' incluye 10 preguntas que deben ser respondidas. Las preguntas son:

1. Esta oportunidad de mejora o necesidad, ¿Es algo que ha surgido como nuevo o ya existía y ha ido creciendo con el tiempo?
2. ¿Por qué esto supone una oportunidad/necesidad? ¿Afecta a tu desempeño?
3. ¿Cuál es el objetivo que perseguimos? ¿Cómo sabemos que lo hemos conseguido?
4. ¿Con la adquisición de estos conocimientos, queremos conseguir hacer que ahora no lo hacemos o lo hacemos de forma defectuosa o insuficiente? ¿Qué no estamos consiguiendo hacer ahora mismo y deberíamos ser capaces de hacer correctamente?
5. ¿Qué ocurren en nuestro rendimiento individual o de equipo, si no hacemos nada?
6. ¿A qué personas o equipos afecta que no se aproveche esta oportunidad de aprendizaje? ¿Es algo que afecta a todo el equipo, aunque sea parcialmente, (conocimiento necesario incompleto) o solo a una parte del mismo? ¿Cómo afecta?
7. ¿Se ha hecho algo para aprovechar la oportunidad o responder a la necesidad desde que se viene detectando (aunque no sea necesariamente una actividad de aprendizaje)?
8. ¿Qué tipo de recursos necesitamos para aprovechar la oportunidad o responder a la necesidad?
9. ¿Qué tipo de solución formativa podría mejorar mejor a priori según tu experiencia? ¿Qué formato podría ser más apropiado?
10. Existen barreras e influencias que se comportan como impedimentos a que se aproveche esta oportunidad o necesidad. ¿Qué barreras existen? ¿Qué influencias existen?

Figura 5.4: Cuestionario INTAKE para solicitar formación.

Un cuestionario de 10 preguntas para recoger información sobre la necesidad formativa u oportunidad de mejora. Este cuestionario solo aparece para el caso de formación extraordinaria.

Nos encontramos con otra pestaña *Datos generales* que sí aparece para ambos tipos de formación. que nos muestra la pantalla de la Figura 5.5:

La captura de pantalla muestra la interfaz de usuario de la plataforma tracasa. En la parte superior izquierda, hay un menú lateral con las mismas opciones que en la Figura 5.4. La barra superior tiene los mismos iconos. El título de la pantalla es 'Nueva formación' y el sub-título es 'Datos generales'. La sección 'Datos generales' contiene campos para rellenar la información de la acción formativa. Los campos incluyen:

- Nombre acción formativa: Un campo para introducir el nombre de la acción formativa.
- Área: Un campo para seleccionar el área.
- Aprobador: Un campo para seleccionar al aprobador.
- Validador: Un campo para seleccionar al validador.
- Tipo de formación: Un campo para seleccionar el tipo de formación.
- Selección del tipo de contenido: Un campo para seleccionar el tipo de contenido.
- Lugar: Un campo para indicar el lugar.
- Tipo de coste: Un campo para seleccionar el tipo de coste.
- Unidades: Un campo para indicar las unidades.
- Importe: Un campo para indicar el importe.
- Tasa: Un campo para indicar la tasa.
- Fecha inicio: Un campo para indicar la fecha de inicio.
- Fecha fin: Un campo para indicar la fecha de fin.
- Hora formación: Un campo para indicar la hora de formación.
- Habitación: Un campo para indicar la habitación.
- Plano de formación: Un campo para seleccionar el plano de formación.
- Mostrar en la home: Una opción para mostrar la acción formativa en la página principal.
- Es bonificable: Una opción para marcar si es bonificable.
- Contenido y/o programa: Un campo para introducir el contenido y/o programa.
- Vehículo: Un campo para introducir el vehículo.
- Observaciones: Un campo para introducir observaciones.

Figura 5.5: Formulario de datos generales a llenar.

Cumplimentar los datos de la segunda pantalla:

Nombre: Nombre de la acción formativa.

Área: Área a la que pertenece el creador de la solicitud.

Aprobador/a: Responsable directo del/los asistentes. En caso de asistir personal de varias áreas, identificar como aprobador al responsable del área de la que asistan más personas.

Validador/a: Director de departamento al que pertenecen los asistentes. En caso de asistir

personal de varios departamentos, identificar como aprobador al director del departamento del que asistan más personas.

Valorador/a: Encargado/a de valorar la formación. Puede haber más de un/a valorador/a para cada formación. Por defecto se asigna como valorador/a el aprobador/a.

Planificación: Ubicarla en el trimestre en el que se imparte la acción.

Modalidad de la formación: Modalidad en la que se imparte (Online, Streaming, presencial, mixta...).

Tipo de formación: Si la formación es interna (el docente es de plantilla), externa (el docente es una empresa/persona ajena a la empresa), mixta (se combinan ambas modalidades).

Medio pedagógico: Comunidades en prácticas (aprendizaje en comunidad o social), aprendizaje experiencial (Learning by doing), aprendizaje tradicional (formación tradicional).

Tipo de contenido: Si se trata de conocimientos técnicos, Soft Skills, Idiomas, Agile...

Lugar: Dónde tiene lugar la acción formativa.

Tipo de coste: Coste (si la acción implica un coste en la matriculación) o gratuita (no supone coste en la inscripción).

Código: Aparece cumplimentada por defecto. No rellenable.

Unidades: Número de personas para asistir.

Importe: En caso de que suponga coste, indicar cuantía por persona sin IVA.

Total: Cuantía total sin IVA. No rellenable.

Fecha Inicio: Fecha en la que tiene lugar el inicio de la acción formativa.

Fecha Fin: Fecha en la que finaliza la acción formativa.

Horas de formación: Número de horas que dura la acción formativa.

Sala: La sala donde se impartirá la acción formativa.

Días aviso valoración: Aparece cumplimentada por defecto. No rellenable.

Tipo de plan de formación: Planificado o Extraordinario. Según qué tipo de solicitud hemos solicitado tendrá un valor u otro, con la posibilidad de poder cambiarlo.

Plan de formación anual: Plan para formar en el año actual. En el caso de solicitud extraordinaria este campo estará deshabilitado y, para el caso de planificada, tendrá como valor la propuesta del plan de formación seleccionada, también con la opción de poder cambiarla.

Mostrar en la home: Opción para mostrar o no en la pestaña home.

Es bonificable: Opción para indicar si es bonificable o no la acción formativa.

Contenido y/o programa: Espacio destinado a alojar el link del programa de contenidos, en caso de disponer de él, o adjuntarlo en caso de tenerlo como documento.

Temario: Espacio destinado a adjuntar el temario en caso de tenerlo como documento.

Observaciones: Observaciones opcionales para indicar en la acción formativa.

Si hemos seleccionado la opción de *Solicitar formación planificada* nos aparece lo que vemos en la Figura 5.6

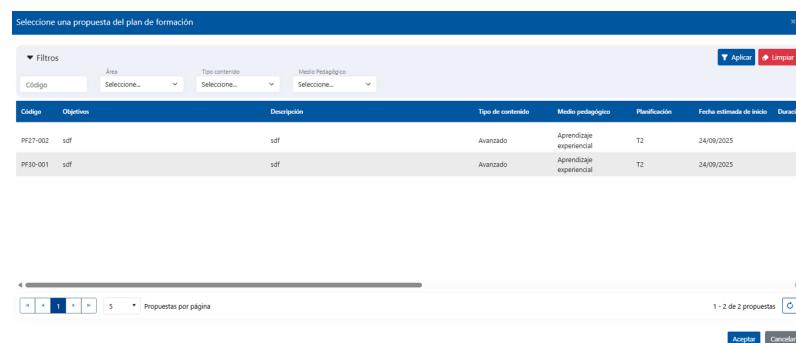


Figura 5.6: Ventana para seleccionar propuesta a asignar a la formación.

Es un listado de propuestas aceptadas de los planes de formación que estén aprobados y activos. Se selecciona una propuesta y haciendo clic en el botón *Aceptar* nos carga solo la pestaña de *Datos generales* con los campos *Tipo de plan de formación* y *Plan de formación* automáticamente llenados.

Una vez cumplimentada la pantalla de Datos generales, debemos guardar y continuar la creación de la solicitud y a continuación aparecerá más pestañas en la parte superior, tal y como se muestra en la Figura 5.7



Figura 5.7: Pestañas que cargan después de crear una formación.

La pestaña Docencia es el espacio para crear el/los docentes e introducir toda la información relativa a la docencia como vemos en la Figura 5.8:

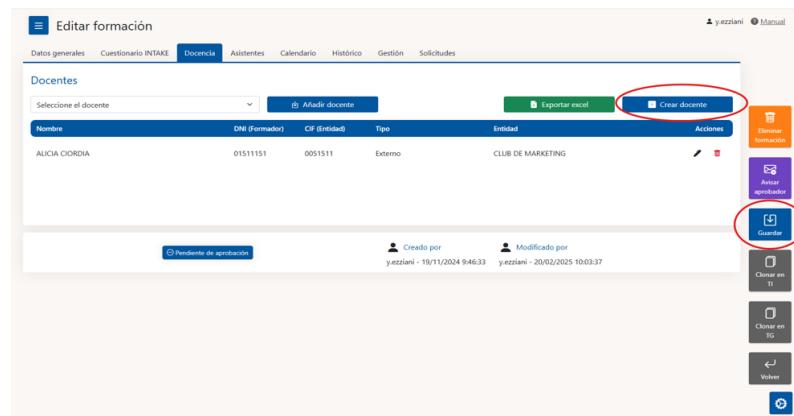


Figura 5.8: Pestaña de los docentes de la formación.

Pulsamos Crear docente y aparece una línea para introducir los datos correspondientes (Nombre, DNI, CIF, Tipo de docente, si es ajeno a la empresa o interno, entidad a la que pertenece...).

Cuando la acción formativa está pendiente de tramitación se permite seleccionar asistentes y nos aparece el botón de enviar convocatoria y lo podemos ver en la Figura 5.9:

Asistentes								
	Nombre	Estado convocatoria	Fecha envío convocatoria	Fecha respuesta convocatoria	Código	Área	Horas laborables	Acciones
<input type="checkbox"/>	Daniel Pérez Cámpo	Pendiente	03/03/2025		3000/25-Z-1013-0027	1013	1	
<input type="checkbox"/>	David Laredo Díaz				3000/25-Z-1013-0027	1013	1	
<input type="checkbox"/>	Yasín Ezziani				3000/25-Z-1013-0027	1013	1	

Figura 5.9: Seleccionar asistentes para enviarles envío de convocatoria.

Al hacer clic se envía un correo a los asistentes seleccionados. En este correo se encuentra un archivo adjunto que contiene los eventos creados en la pestaña Calendario (que veremos más adelante) para agregarlos automáticamente en el Outlook personal de cada asistente.

Por otro lado, en el cuerpo del correo aparece un enlace el cual nos dirige a una página donde se muestra el nombre de la acción formativa y podemos aceptar o rechazar la asistencia con la posibilidad de agregar comentario opcional. Se puede aceptar o rechazar solo una vez, por lo que si se intenta acceder de nuevo después de contestar salta acceso denegado.

Cuando la formación pasa a ser tramitada, automáticamente a todos los asistentes que están pendientes de responder o que no se les mandó correo antes se les envía el correo. Hay dos casos a considerar al enviar un correo a uno o varios asistentes:

- Si el correo se envía el mismo día de la fecha de finalización de la acción formativa, la asistencia se rechaza automáticamente y se notifica a los asistentes por correo electrónico.

2. Si quedan menos de 10 días para la fecha de finalización de la acción formativa, la asistencia se acepta automáticamente y no se envía ningún correo a los asistentes.

Cuando la formación pasa a ser pendiente de valoración, al seleccionar los asistentes aparece un nuevo botón para enviar correo para encuesta de eficacia formativa y en el correo se manda una url a la pantalla que se ve en la Figura 5.10 para llenar la encuesta.

Figura 5.10: Página para llenar encuesta de eficacia formativa.

No se puede llenar más de una vez excepto si se vuelve a enviar al asistente de nuevo el correo como comentamos en convocatoria. En este estado se agrega una pestaña más que es Valoraciones como vemos en la Figura 5.11:

Figura 5.11: Pestaña de valoraciones de una formación.

Nos indica tanto las valoraciones de la formación, que explicaremos más tarde, como las encuestas realizadas, como vemos se muestra marcado en rojo la valoración media total de todas las encuestas. Si desplegamos alguna encuesta podemos ver la respuesta del asistente y la valoración total de dicha encuesta, tal y como se ve en la Figura 5.12.

Encuesta 1

1. Se ha alcanzado el objetivo de la formación. *
★★★ 4 / 4

2. El contenido de la formación se ajusta a nuestras necesidades.*
★★☆ 3 / 4

3. Los asistentes han desarrollado nuevas habilidades.*
★★★ 4 / 4

4. Se han puesto en práctica los conocimientos adquiridos en el curso.*
★★★ 4 / 4

Valoración total: 3,8

Figura 5.12: Resultado de una encuesta realizada a una formación.

En la pestaña Calendario, que vemos en la Figura 5.13, se incluirán los datos relativos a días y horas que abarca la formación.

Editar formación

Datos generales Cuestionario INTAKE Docencia Asistentes Calendario Histórico Gestión Solicitudes

Calendario - (21/01/2025 - 28/04/2025)

Hoy	20/01	21/01	22/01	23/01	24/01	25/01	26/01
todo el día							
7:00							
8:00							
9:00							
10:00							
11:00							
12:00							
13:00							
14:00							
15:00							
16:00							

Formación

Eliminar formación
Avistar aprobador
Guardar
Cerrar en TI
Cerrar en TG
Volver

Pendiente de aprobación

Creado por: y.ezziani - 19/11/2024 9:46:33 Modificado por: y.ezziani - 20/02/2025 10:03:37

Figura 5.13: Pestaña de calendario de una formación.

Es un sistema flexible que permite personalizar las jornadas y días. El funcionamiento es similar al de la herramienta *Outlook*.

En la pestaña Histórico, que vemos en la Figura 5.14, se guarda un historial de cambios de los estados de la acción formativa, cambios de valores de los campos de los datos generales (importe, horas, fecha inicio, fecha fin), cuando se añade/elimina un docente y cuando se añade/elimina un asistente.

Figura 5.14: Pestaña con el histórico de cambios en una formación.

En la pestaña Gestión, que vemos en la Figura 5.15, se pueden enviar avisos por correo, llenar opcionalmente el campo de cantidad bonificada y/o asignar pedidos, creando uno nuevo o asignando uno existente.

Figura 5.15: Pestaña de gestión de una formación.

Al querer asignar pedido nos muestra el listado de pedidos para seleccionar, tal y como nos aparece en la Figura 5.16:

5.3 Solicitar nueva acción formativa

27

Pedido	Nombre	Ref Interna	Ref Compra	Proveedor	Área	Fecha
TI-2022-12-010	Servicios generales - Formación Adeloz - febrero 2023		AD-DA10-79-2022	0027 - CLUB MARKETING	1014	2024-12-20
TI-EX09/2022-12-002	PLATAFORMA DE CONTENIDOS APRENIZAJE	20221216		0122 - GOODHABIT SPAIN, S.L.	1013	2024-12-16
TI-2022-12-009	Jurídico - curso inglés jurídico-Tolos		AD-DA10-78-2022	0028 - EDITORIAL ARANZADI S.L.	1014	2024-12-12
TI-2022-12-006	Paula Moreno Complej v2	20221212	AD-255-2022	0001 - ACREDITORES VARIOS	1013	2024-12-12
TI-2022-11-027	Formación AZURE A2-900	SIT 2022 AZURE	AD-DA10-75-2022	0001 - ACREDITORES VARIOS	5050	2024-11-30
TI-2022-12-001	PROGRAMA ACTIVIDAD FISICA	20231123		0198 - MERCER BIENESTAR CORPORATIVO	1013	2024-11-23
TI-2022-12-001	genero - inglés - vaughan - enero - abril 2023		AD-DA10-74-2022	0088 - VAUGHAN SYSTEMS, SLU	1011	2024-11-21
TI-2021-11-015	formación power bi	AHN_2021	AD-DA10-01-2021	0015 - AIN ASOC. INDUSTRIA NAVARRA	5050	2024-11-15

Figura 5.16: Listado de pedidos para asignar a una formación.

Se pueden filtrar y al asignar un pedido nos muestra la referencia interna y el nº del pedido vinculado con la posibilidad de tanto ver el pedido como desvincularlo.

Número factura	Fecha factura	Fecha entrada	Importe	Proveedor	Ref. compra	Estado	Acción
2022-13182	30/11/2022	14/12/2022	810,00 €	0007 - CHAI	FPC-GRUPO-39-2022	<input checked="" type="checkbox"/> Pagada	<input type="checkbox"/> Marcar Pagada
2022-13146	31/10/2022	16/11/2022	607,00 €	0007 - CHAI	FPC-GRUPO-39-2022	<input type="checkbox"/> Pendiente	<input type="checkbox"/> Marcar Pagada

Figura 5.17: Pestaña de gestión de una formación con pedidos asignados.

Vemos en la Figura 5.17 que aparece también el listado de facturas del pedido vinculado. A parte de los datos de las facturas se nos proporciona el estado de cada factura (pagada o pendiente). Si una factura pendiente se ha cambiado de estado a pagada podemos actualizar las facturas de estado usando el botón *Marcar Pagada* que aparece en la columna de Acción para las facturas pendientes señalado con un cuadrado rojo.

En la pestaña *Solicitudes*, que vemos en la Figura 5.18, podemos ver todas las solicitudes de inscripción pendientes de aprobar

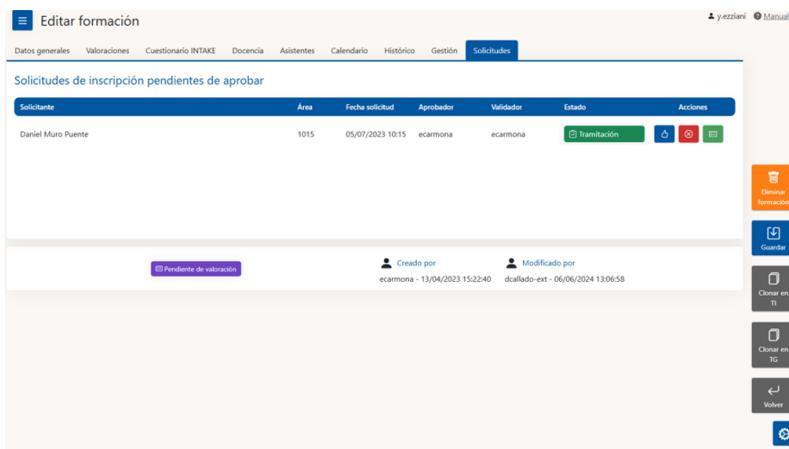


Figura 5.18: Pestaña de solicitudes de inscripción pendientes de aprobar.

Finalizamos la solicitud de acción formativa pulsando *Avisar al aprobador* y automáticamente llega un aviso a los responsables y *PyT* para validar y tramitar la acción, si es el caso. En caso de que, una vez solicitada una acción formativa, surja cualquier tipo de modificación, de procederá a hacer los cambios oportunos en la aplicación y, tras guardar esos cambios, debemos volver a *Avisar al aprobador*.

5.4. Solicitar incluirse en una acción formativa ya creada

En la pantalla Home, tal y como podemos observar en la Figura 5.19, podremos visualizar las acciones formativas que han sido solicitadas, aprobadas y tramitadas pero que aún no han dado comienzo, dando la posibilidad de que otras personas, que no han sido convocadas y que pueden estar interesadas en asistir, puedan solicitar su incorporación.

Nombre	Tipo de contenido	Fecha inicio	Fecha fin	Duración (horas)
Formación Creada en TG	Avanzado	25/03/2027	07/04/2027	60

Figura 5.19: Pantalla de HOME con las formaciones ya tramitadas sin comenzar.

Pulsamos en cualquier parte de la línea de la formación en la que estamos interesados. Mostrará la información de la acción y para finalizar la solicitud de incorporación a la acción, debemos pulsar Solicitar inscripción y llegará a responsables y *PyT* para validar y tramitar la solicitud, si es el caso.

5.5. Clonación y valoraciones

Otra funcionalidad que tenemos es la posibilidad de clonar una acción formativa como se ve en la Figura 5.20.

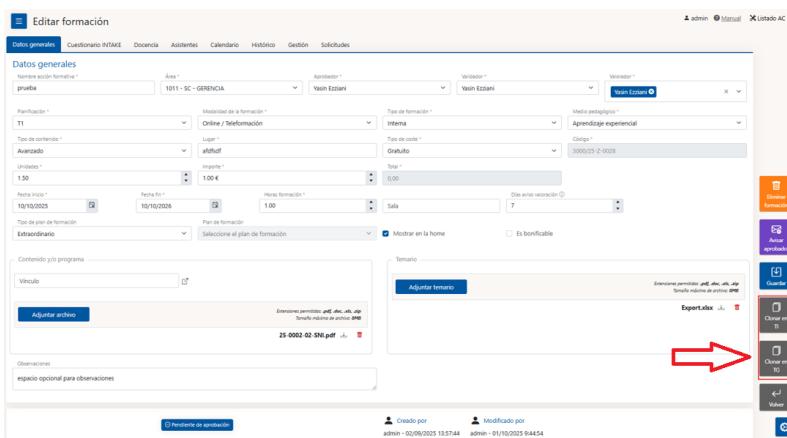


Figura 5.20: Botones de clonación de la formación.

Podemos clonar tanto en instrumental como en global. Al hacer clic se realiza un proceso similar a cuando solicitamos formación con algunos cambios:

1. La pestaña Cuestionario INTAKE se rellena por defecto con la información que había en la acción formativa que se desea clonar.
2. La pestaña Datos generales se rellena por defecto con la información que había en la acción formativa que desea clonar, excepto los campos Fecha inicio y Fecha fin. En el caso de clonar en la empresa contraria los campos que son selectores se cargan con los datos de dicha empresa.
3. Se muestra la pestaña Docencia con los docentes que tenía la acción formativa que se va a clonar sin la posibilidad de agregar o crear nuevos docentes.

5.5.1. Información para responsables y directores

En el momento en que un trabajador crea una acción formativa, la solicitud generada se envía automáticamente al **responsable de área** correspondiente para su revisión inicial. Este será el encargado de analizar la propuesta y valorar si la acción solicitada se ajusta a las necesidades del departamento o si, por el contrario, no resulta procedente.

- En caso de ser rechazada, el sistema genera de forma automática una notificación al creador de la solicitud, informándole tanto del estado de la misma como del motivo del rechazo, garantizando así una comunicación directa y transparente.
- Si la solicitud es aprobada, esta se remite al **director del departamento** (cuando corresponda) para que realice un segundo nivel de validación y análisis antes de su tramitación definitiva.

El proceso se repite en este punto, siguiendo la misma dinámica de aprobación y notificación entre los distintos niveles jerárquicos.

- En caso de que el director rechace la solicitud, el sistema volverá a generar automáticamente un aviso dirigido tanto al responsable de área como al creador de la acción formativa, informando del estado actualizado y del motivo de la decisión.
- Si, por el contrario, la solicitud es aprobada, se enviará un correo electrónico al área de *PyT*, que será la encargada de revisar, aprobar y tramitar la formación de manera definitiva.

De esta forma, el flujo de aprobación garantiza que todas las solicitudes sean revisadas por los distintos niveles de responsabilidad antes de ser ejecutadas, manteniendo un control exhaustivo sobre el proceso y asegurando la coherencia entre las necesidades individuales de los empleados y los objetivos estratégicos de la empresa. Podemos ver este flujo en el diagrama que aparece en la Figura 5.21

Además, los **responsables de área** y los **directores de departamento** recibirán de forma continua notificaciones sobre las solicitudes pendientes de revisión o cualquier otra tarea relacionada con su gestión. En caso de que estas notificaciones no se localicen en el buzón de correo, los usuarios podrán consultar directamente las tareas pendientes a través del menú lateral de la aplicación, donde se ha habilitado un acceso específico para este fin.

Al acceder al sistema, dispondrán de dos nuevas secciones que no están visibles para el resto de trabajadores:

- La opción *Pdte. Autorizar*, que muestra el listado de solicitudes que se encuentran pendientes de aprobación. Desde este apartado los responsables y directores podrán acceder al detalle de cada acción formativa, revisar su información completa y, en función de su criterio, aprobar o rechazar la solicitud directamente desde la interfaz.
- La opción *Pdte. Valorar*, destinada a la gestión de las valoraciones de la eficacia de la formación. En este apartado se mostrarán las formaciones finalizadas que están pendientes de valoración, y al acceder a una de ellas aparecerá una nueva pestaña en la parte superior con el formulario correspondiente para registrar la evaluación.

Gracias a este sistema de comunicación y control, la aplicación centraliza todos los procesos relacionados con la validación, aprobación y valoración de las acciones formativas, reduciendo los tiempos de gestión y evitando pérdidas de información entre los distintos niveles organizativos.

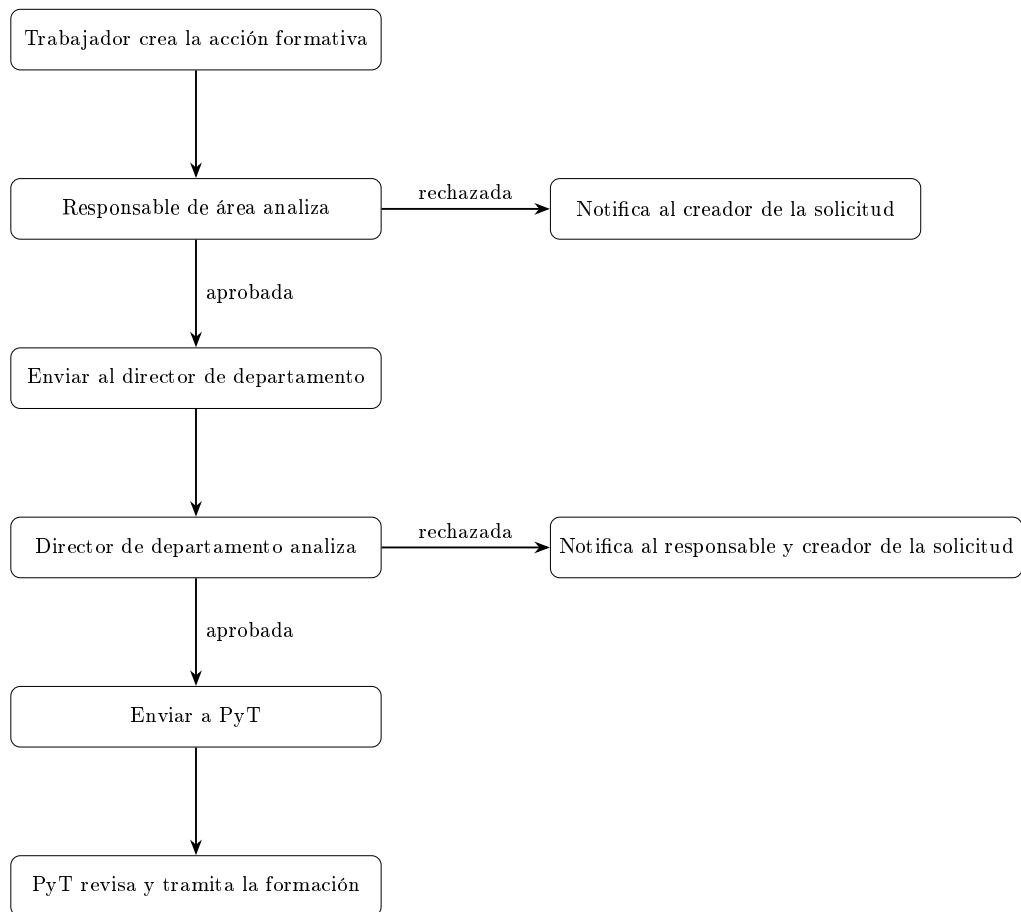


Figura 5.21: Flujo de aprobación y notificaciones de las acciones formativas, incluyendo rutas de rechazo.

5.6. Planes de formación y propuestas

En el menú lateral tenemos una sección de planes de formación anula, si hacemos clic nos aparece la siguiente Figura 5.22:

Acciones	Año	Presupuesto	Horas anuales	Estado	Descripción	Acción
<input checked="" type="checkbox"/> <input type="checkbox"/>	2030	200.000,00 €	1000	Acceptado	Plan 2025	<input checked="" type="checkbox"/> Activo
<input checked="" type="checkbox"/> <input type="checkbox"/>	2029	99.999.999,00 €	153	Pendiente PdT	señalizadaad	<input checked="" type="checkbox"/> Activo
<input checked="" type="checkbox"/> <input type="checkbox"/>	2028	80.000,00 €	200	Pendiente PdT		<input checked="" type="checkbox"/> Activo
<input checked="" type="checkbox"/> <input type="checkbox"/>	2027	200.000,00 €	1000	Acceptado	Plan 2025	<input checked="" type="checkbox"/> Activo
<input checked="" type="checkbox"/> <input type="checkbox"/>	2026	200.000,00 €	1000	Pendiente Gerencia	Plan 2025	<input checked="" type="checkbox"/> Activo
<input checked="" type="checkbox"/> <input type="checkbox"/>	2025	200.000,00 €	1000	Pendiente PdT	Plan 2025	<input checked="" type="checkbox"/> Activo

Figura 5.22: Pantalla con el listado de planes de formación anual.

Un listado de planes con la opción de poder filtrar los activos o inactivos. Para crear un plan de formación hacemos clic en el botón *Crear plan de formación anual* que nos muestra la página de la Figura 5.23:

Figura 5.23: Pantalla con el formulario para crear un plan de formación anual.

Al crear el plan aparecen más pestañas, tal y como podemos observar y la Figura 5.24 y en la parte inferior, se nos muestra el estado del plan junto al año del plan.

Figura 5.24: Pantalla de edición del plan de formación anual con todas las pestañas.

En la pestaña Departamentos vemos la siguiente Figura 5.25

Nombre	Total usuarios	Acciones
SC	2	
Usuarios con permisos		
(admin)	<input checked="" type="checkbox"/>	
Yasin Ezziani (y.ezziani)	<input checked="" type="checkbox"/>	
SIC	1	
GTSI	0	
SIASS	1	
SIT	0	
SUUD	0	
SIAA	0	
GEDESA	0	

Figura 5.25: Pantalla de departamentos del plan de formación anual.

Aquí, por cada departamento se podrá indicar que usuarios tendrán permisos para subir propuestas de formación para el departamento indicado. Para cada usuario hay una columna Enviar plan a PyT para poder indicar a que usuario dar los permisos, tal y como vemos en la Figura 5.26

Usuarios con permisos	Enviar plan a PyT
Yasin Ezziani (y.ezziani)	<input checked="" type="checkbox"/>
Xabier Dendaneta Sarrés (xdendarreta)	<input type="checkbox"/>

Figura 5.26: Usuarios con permiso de un departamento del plan de formación anual.

En la pestaña de propuestas se encuentra el listado de las propuestas para el plan de formación como podemos ver en la Figura 5.27

Figura 5.27: Pantalla con el listado de las propuestas de plan de formación anual.

Deabajo del listado nos aparecen los estados que podría tener una propuesta y cada estado con un color diferente. En el listado al inicio de cada propuesta aparece el color que representa su estado. En el caso de la Figura 5.27, hay 2 propuestas unificadas, 4 aceptadas y 1 rechazada.

Si nos fijamos en la parte superior hay una serie de botones:

Crear propuesta de formación: haciendo clic nos dirige a la pantalla, que vemos en la Figura 5.28, donde podemos crear una nueva propuesta rellenando el formulario y haciendo clic en *Guardar y continuar*.

Exportar excel: nos permite descargar un excel con las propuestas que hay en el listado.

Cargar excel: nos permite cargar un excel que tengamos con propuestas y agregarlos al plan automáticamente sin tener que crearlas manualmente.

Figura 5.28: Pantalla con el formulario para crear una propuesta del plan de formaciones.

El resto de los botones aparecen deshabilitados por defecto.

Aprobar: nos permite aprobar propuestas. Para ello se debe seleccionar una o varias propuestas y hacer clic al botón para aprobarlas como vemos en la Figura 5.29.

Rechazar: nos permite rechazar propuestas. Para ello se debe seleccionar una o varias propuestas y hacer clic al botón para rechazarlas como vemos en la Figura 5.29.

Combinar propuestas: A parte de poder crear una propuesta rellenando el formulario, tenemos la opción de generar una nueva propuesta combinando dos o más propuestas. Para este caso se deben seleccionar mínimo dos propuestas como vemos en la Figura 5.30

Acciones	Código	Objetivos	Descripción	Tipo de contenido	Medio pedagógico
	PF25-001	a	sadas	IDIOMAS	Aprendizaje tradicor
	PF25-002	asdas	dsadasdas	COMPETENCIAS TÉCNICAS	Aprendizaje tradicor
	PF25-003	asd	sadas	IDIOMAS	Aprendizaje tradicor
	PF25-004	asdasd	sadas	PR/ SALUD LABORAL	Aprendizaje tradicor
	PF25-005	sdffsf	sadas	IDIOMAS	Aprendizaje tradicor
	PF26-006	qqq	sadas	IDIOMAS	Aprendizaje tradicor
	PF26-007	hhh	sadas	IDIOMAS	Aprendizaje tradicor

Figura 5.29: Pantalla con el listado de las propuestas de plan de formación anual con una fila seleccionada.

Acciones	Código	Objetivos	Descripción	Tipo de contenido	Medio pedagógico
	PF25-001	a	sadas	IDIOMAS	Aprendizaje tradicor
	PF25-002	asdas	dsadasdas	COMPETENCIAS TÉCNICAS	Aprendizaje tradicor
	PF25-003	asd	sadas	IDIOMAS	Aprendizaje tradicor
	PF25-004	asdasd	sadas	PR/ SALUD LABORAL	Aprendizaje tradicor
	PF25-005	sdffsf	sadas	IDIOMAS	Aprendizaje tradicor
	PF26-006	qqq	sadas	IDIOMAS	Aprendizaje tradicor
	PF26-007	hhh	sadas	IDIOMAS	Aprendizaje tradicor

Figura 5.30: Pantalla con el listado de las propuestas de plan de formación anual con dos filas seleccionadas.

Podemos desplegar las propuestas y nos aparecerán 3 pestañas:

Asistentes: podemos ver el listado de asistentes de la propuesta, lo podemos ver en la Figura 5.31.

Propuestas unificadas: solo si la propuesta ha sido creada combinando otras, nos aparecerá el código de esas propuestas que hemos combinado, lo podemos ver en la Figura 5.32.

Formaciones creadas: nos aparecen las formaciones que tienen asignada esa propuesta, lo podemos ver en la Figura 5.33.

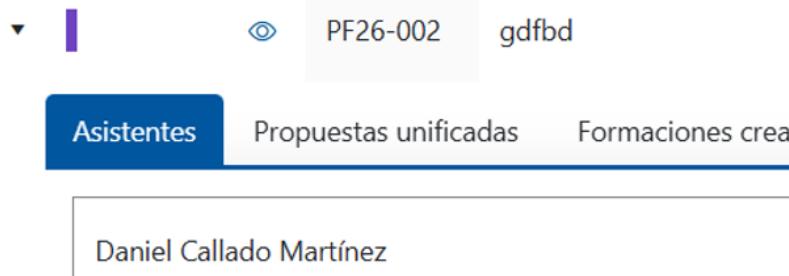


Figura 5.31: Pestaña de asistentes dentro de la propuesta del plan de formaciones.

Acciones	Código	Objetivos	Descripción	Tipo de contenido	Medio pedagógico
•	PF25-003	asd	sadas	IDIOMAS	Aprendizaje tradicior
•	PF25-004	asdasd	sadas	PRJ/ SALUD LABORAL	Aprendizaje tradicior
•	PF26-008	asd	sadas	IDIOMAS	Aprendizaje tradicior

Asistentes Propuestas unificadas Formaciones creadas

PF25-003
PF25-004

Figura 5.32: Pestaña de propuestas unificadas dentro de la propuesta del plan de formaciones.



Figura 5.33: Pestaña de formaciones creadas que tienen asignada la propuesta del plan de formaciones.

6

Funcionamiento interno

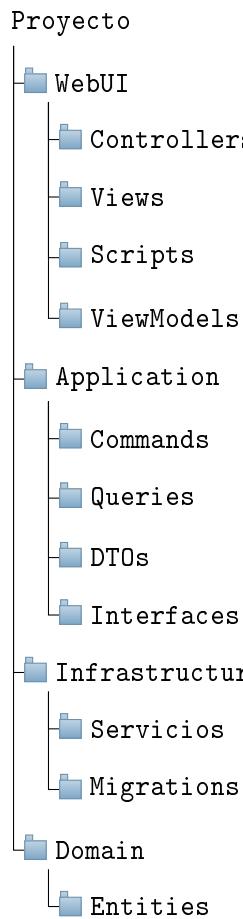
En el capítulo 5 se ha descrito el funcionamiento del Ecosistema de Aprendizaje desde el punto de vista del usuario, detallando las pantallas, los flujos y las interacciones. En esta sección se aborda el sistema desde la perspectiva del desarrollador, explicando la arquitectura interna, los patrones aplicados, el funcionamiento del backend y diversos fragmentos de código relevantes para comprender cómo se ha construido la solución.

El objetivo de esta sección es aportar una visión técnica del proyecto, profundizando en la estructura del código, la implementación de la lógica de negocio, la comunicación entre las capas y las decisiones técnicas que han guiado el desarrollo.

6.1. Arquitectura técnica y organización del proyecto

El sistema se ha implementado siguiendo una arquitectura por capas combinada con el patrón *Command Query Responsibility Segregation* (CQRS). Esta combinación proporciona una estructura clara, mantenible y fácilmente escalable. La separación entre comandos y consultas permite dividir de forma explícita las operaciones que modifican estado de aquellas que únicamente lo leen.

La estructura general del proyecto puede verse en el siguiente esquema:



Cada una de estas áreas cumple un rol específico:

WebUI: interacción con el usuario, controladores, vistas y modelos de presentación.

Application: lógica de negocio, implementación de CQRS, validaciones y DTOs.

Infrastructure: acceso a datos, repositorios, migraciones y servicios externos.

Domain: entidades principales y reglas del dominio de negocio.

6.2. Implementación de consultas (Query Handlers)

El controlador mostrado en el capítulo 4 de formaciones llama internamente a consultas modeladas como *Handlers*. Estas consultas se encuentran en la capa *Application* y permiten recuperar datos de manera estructurada.

A continuación se muestra un ejemplo simplificado del *Handler* correspondiente a la obtención del listado de formaciones:

```
1 | public class GetAllFormacionesQueryHandler :
```

```
2 IRequestHandler<GetAllFormacionesQuery ,  
3     ApiRespuesta<List<FormacionListDTO>>>  
4 {  
5     private readonly IFormacionesDataService  
6         _formacionesDataService ;  
7  
8     public  
9         GetAllFormacionesQueryHandler(IFormacionesDataService  
10            formacionesDataService)  
11     {  
12         _formacionesDataService = formacionesDataService;  
13     }  
14  
15     public async Task<ApiRespuesta<List<FormacionListDTO>>>  
16         Handle(GetAllFormacionesQuery request ,  
17             CancellationToken cancellationToken)  
18     {  
19         var apiRespuesta = new  
20             ApiRespuesta<List<FormacionListDTO>> {  
21                 Data = new List<FormacionListDTO>(), Tipo =  
22                     TipoRespuesta.Error };  
23  
24         List<FormacionListDTO> formaciones = await  
25             Task.Run(() => _formacionesDataService  
26                 .SelectAll(request.Filtros , cancellationToken));  
27         if (formaciones != null)  
28         {  
29             apiRespuesta.Tipo = TipoRespuesta.Success;  
30             apiRespuesta.Data = formaciones;  
31         }  
32  
33         return apiRespuesta;  
34     }  
35 }  
36 }
```

Listing 6.1: Clase Handler que realiza una consulta para obtener formaciones.

Este fragmento de código define una clase llamada *GetAllFormacionesQueryHandler*. Esta clase se encarga de manejar una solicitud para obtener todas las formaciones en el sistema, basándose en los filtros proporcionados en la consulta. En el constructor, se inyecta una dependencia del servicio *IFormacionesDataService*, que se utiliza para acceder a los datos relacionados con las formaciones. El método *Handle* es el encargado de ejecutar la consulta. Recibe como parámetros el objeto *request*, que contiene los filtros para la consulta, y un *cancellationToken* para permitir la cancelación de la operación si es necesario.

sario. Dentro del método, se inicializa una *apiRespuesta* con un estado de error. Luego, se ejecuta de forma asíncrona el método *SelectAll* del servicio *formacionesDataService*, que realiza la consulta a la base de datos utilizando los filtros y el token de cancelación. Si la consulta devuelve una lista de formaciones, el tipo de respuesta se actualiza a *Success*, y la lista obtenida se asigna al campo *Data* de la respuesta. Finalmente, el método retorna la respuesta que puede contener tanto los datos de las formaciones como un posible error si no se encuentran datos o si ocurre alguna otra falla. La clase *FormacionListDTO* que en este caso representa una formación contiene:

```

1 public class FormacionFiltrosDTO
2 {
3     public string FiltroTexto { get; set; }
4     public string FiltroArea { get; set; }
5     public string FiltroTipo { get; set; }
6     public string FiltroModalidad { get; set; }
7     public string FiltroTipoConocimiento { get; set; }
8     public string FiltroAsistente { get; set; }
9     public int? FiltroFormacionEstado { get; set; }
10    public int? FiltroPedido { get; set; }
11    public DateTime? FiltroFechaInicioDesde { get; set; }
12    public DateTime? FiltroFechaInicioHasta { get; set; }
13    public bool? FiltroBonificable { get; set; }
14    public bool? FiltroAccionCorrectiva { get; set; }
15    public bool? FiltroPendienteFactura { get; set; }
16 }
```

Listing 6.2: Clase DTO para filtrar las formaciones.

En cada *Query* tenemos un parámetro de filtrado.

6.3. Implementación de comandos (Command Handlers)

Las operaciones que modifican el estado de la aplicación, como crear o actualizar una formación, se implementan mediante *Command Handlers*. Su diseño es similar al de las consultas, pero incluye validaciones y lógica de negocio adicional.

```

1 public class CreateFormacionCommandHandler :
2 IRequestHandler<CreateFormacionCommand ,
3                         ApiRespuesta<FormacionDTO>>
4 {
5     private readonly IFormacionesDataService
6             _formacionesDataService;
7     public
8         CreateFormacionCommandHandler(IFormacionesDataService
```

```
      formacionesDataService)
6    {
7      _formacionesDataService = formacionesDataService;
8    }
9
10   public async Task<ApiRespuesta<FormacionDTO>>
11     Handle(CreateFormacionCommand request,
12       CancellationToken cancellationToken)
13   {
14     var formacionDTO = request.FormacionDTO;
15     var apiRespuesta = new ApiRespuesta<FormacionDTO>()
16       {
17         Tipo = TipoRespuesta.Error
18         , Data = formacionDTO };
19
20     if (formacionDTO.ListaErrores.Count == 0) {
21
22       InicializarCampos(formacionDTO);
23
24       formacionDTO = await Task.Run(() =>
25         _formacionesDataService
26         .Insert(formacionDTO, cancellationToken));
27
28       if (formacionDTO != null)
29       {
30         apiRespuesta.Tipo = TipoRespuesta.Success;
31         apiRespuesta.Data = formacionDTO;
32       }
33
34     return apiRespuesta;
35   }
36
37   private void InicializarCampos(FormacionDTO formacionDTO)
38   {
39     formacionDTO.FormacionID = 0;
40     formacionDTO.FormacionEstadoID =
41       (int)EFormacionEstados.PendienteAprobacion;
42     formacionDTO.CreadoPor =
43       Utility.UsuarioActual.NombreSinDominio;
44     formacionDTO.CreadoEn = DateTime.Now;
45     formacionDTO.ModificadoPor = formacionDTO.CreadoPor;
46     formacionDTO.ModificadoEn = formacionDTO.CreadoEn;
47   }
48 }
```

```

43     formacionDTO.CuestionarioDTO.CuestionarioInicialID =
44         0;
45     formacionDTO.CuestionarioDTO.CreadoPor =
46         formacionDTO.CreadoPor;
47     formacionDTO.CuestionarioDTO.CreadoEn =
48         formacionDTO.CreadoEn;
49     formacionDTO.CuestionarioDTO.ModificadoPor =
50         formacionDTO.CreadoPor;
51     formacionDTO.CuestionarioDTO.ModificadoEn =
52         formacionDTO.CreadoEn;
53
54     if (formacionDTO.AdjuntoFormFile != null
55     && formacionDTO.AdjuntoFormFile.Length > 0)
56     {
57         using (var memoryStream = new MemoryStream())
58         {
59             formacionDTO.AdjuntoFormFile
60                 .CopyTo(memoryStream);
61             byte[] archivoBytes = memoryStream.ToArray();
62
63             formacionDTO.Adjunto = archivoBytes;
64             formacionDTO.AdjuntoNombre =
65                 formacionDTO.AdjuntoFormFile.FileName;
66         }
67     }
68 }

```

Listing 6.3: Clase Handler para crear una formación.

Este fragmento de código define una clase llamada *CreateFormacionCommandHandler*. Su propósito es manejar la creación de una nueva formación. En el constructor, se inyecta el servicio *IFormacionesDataService*, utilizado para insertar la nueva formación en la base de datos.

El método *Handle* recibe como parámetros el objeto *request*, que contiene los datos de la formación a crear, y un *cancellation Token*. Inicialmente, se crea una respuesta *apiRespuesta* con un estado de error y los datos de la formación. Si la lista de errores de la formación está vacía, se procede a inicializar varios campos del objeto *formacionDTO* mediante el método *IniciarCampos*, que establece valores predeterminados como el estado de la formación y los datos de creación. Luego, se ejecuta la inserción de la formación de manera asíncrona utilizando el método *Insert* del servicio *formacionesDataService*. Si la inserción es exitosa, se actualiza la respuesta a *Success* y se asignan los datos de la formación insertada. Finalmente, el método *Handle* retorna una respuesta que puede contener la

formación recién creada o un error si ocurre algún problema en el proceso.

Este tipo de comandos garantiza que todas las modificaciones sigan las reglas de dominio establecidas en la capa *Domain*. Además, para los comandos, tenemos clases validator por cada comando donde validamos los parámetros de los comandos. Es decir, validar que no sean nulos cuando son parámetros obligatorios, validar longitud de cadenas si tienen un máximo para guardar en base de datos....

```

1   public class CreateFormacionCommandValidator :
2       AbstractValidator<CreateFormacionCommand>
3   {
4       public CreateFormacionCommandValidator()
5       {
6           RuleFor(v => v.FormacionDTO.Nombre)
7               .NotEmpty()
8               .NotNull()
9               .MaximumLength(256);
10
11          RuleFor(v => v.FormacionDTO.Lugar)
12              .NotEmpty()
13              .NotNull()
14              .MaximumLength(256);
15
16          RuleFor(v => v.FormacionDTO.FechaInicio)
17              .LessThanOrEqualTo(l => l.FormacionDTO.FechaFin);
18      }
19  }
```

Listing 6.4: Clase Validator para crear una formación.

En este caso, validamos que:

- El valor de *Nombre* de la formación no sea nulo y que no tenga más de 256 caracteres.
- El valor de *Lugar* de la formación no sea nulo y que no tenga más de 256 caracteres.
- El valor de *FechaInicio* de la formación no sea posterior al valor de *FechaFin* de la formación.

Si alguna regla no se cumple se guarda el error y se le muestra al usuario para que rellene el formulario correctamente antes de poder crear la formación.

6.4. Acceso a datos e infraestructura

La capa *Infrastructure* implementa las interfaces definidas en *Application* mediante Entity Framework. Las entidades del dominio se mapean a tablas mediante Fluent API:

```

1 protected override void OnModelCreating(ModelBuilder
2     modelBuilder)
3 {
4     modelBuilder.Entity<Formacion>(entity =>
5     {
6         entity.ToTable("Formaciones");
7         entity.Property(f => f.Nombre)
8             .HasMaxLength(200)
9             .IsRequired();
10    });
}

```

Listing 6.5: Mapeo de tablas mediante Fluent API.

Este fragmento de código define una sobrecarga del método *OnModelCreating* en una clase que hereda de *DbContext*. En este método, se configura el modelo de la entidad *Formacion* utilizando el objeto *modelBuilder*. La entidad *Formacion* se mapea a la tabla *Formaciones* en la base de datos mediante *entity.ToTable("Formaciones")*. Luego, se configura la propiedad *Nombre* de la entidad para que sea obligatoria (*IsRequired()*) y tenga una longitud máxima de 200 caracteres (*HasMaxLength(200)*). Esta configuración garantiza que la propiedad *Nombre* sea requerida y tenga un límite de caracteres en la base de datos.

Asimismo, cada cambio en el modelo genera una migración que se aplica sobre SQL Server, manteniendo sincronizada la estructura de la base de datos con el código fuente. Los métodos que implementan las interfaces tienen de lógica solamente el mapeo y la interacción con las tablas de base de datos.

```

1 public async Task<FormacionDTO> Insert(FormacionDTO
2     formacionDTO,
3     CancellationToken cancellationToken)
4 {
5     using (var transaction =
6         _context.Database.BeginTransaction())
7     {
8         try
9         {
10             CuestionarioInicial cuestionario =
11                 _mapper.Map<CuestionarioInicial>
12                 (formacionDTO.CuestionarioDTO);
13
14             _context.CuestionarioInicial.Add(cuestionario);
15             await
16                 _context.SaveChangesAsync(cancellationToken);
17
18             formacionDTO.CuestionarioInicialID =
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
305
306
307
307
308
309
309
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
13
```

```
15         cuestionario.CuestionarioInicialID;
16     Formacion formacion =
17         _mapper.Map<Formacion>(formacionDTO);
18
19     _context.Formacion.Add(formacion);
20     await
21         _context.SaveChangesAsync(cancellationToken);
22
23     formacionDTO.FormacionID = formacion.FormacionID;
24     formacionDTO.ValoracionUsuario.ForEach(valorador
25         =>
26     {
27         Valoracion valoracionEntity = new Valoracion
28         {
29             FormacionID = formacion.FormacionID,
30             ValoracionUsuario = valorador
31         };
32         _context.Valoracion.Add(valoracionEntity);
33     });
34     await
35         _context.SaveChangesAsync(cancellationToken);
36     transaction.Commit();
37     _logger.LogInformation("Exito al crear la
38         formación");
39
40     }
41 }
42
43     return formacionDTO;
44 }
```

Listing 6.6: Mapeo de tablas mediante Fluent API.

Este fragmento de código define un método asíncrono *Insert* que inserta una nueva formación en la base de datos. Primero, se inicia una transacción de base de datos. Luego, se mapea el objeto *formacionDTO.CuestionarioDTO* a una entidad *CuestionarioInicial* y se guarda en la base de datos. Después, se mapea *formacionDTO* a una entidad *Formacion*

y se agrega a la base de datos. A continuación, se insertan las valoraciones de los usuarios asociadas a la formación. Si todas las operaciones son exitosas, se confirma la transacción. En caso de error, se captura la excepción, se deshace la transacción y se registra el error. Finalmente, el método retorna el objeto *formacionDTO*.

6.5. Sistema de permisos y filtros de autorización

El control de acceso a los distintos métodos del controlador se realiza mediante un filtro personalizado que valida los roles del usuario autenticado. Esto permite proteger acciones como ver detalles de una formación o tramitar solicitudes.

```
1 public class TienePermisosFilterAttribute :  
2     AuthorizeAttribute, IAuthorizationFilter  
3 {  
4  
5     private EAutorizacionPermisos AutorizacionPermisos { get;  
6         set; }  
7     public TienePermisosFilterAttribute(EAutorizacionPermisos  
8         autorizacionPermisos)  
9     {  
10         AutorizacionPermisos = autorizacionPermisos;  
11     }  
12  
13     public void OnAuthorization(AuthorizationFilterContext  
14         context)  
15     {  
16  
17         var usuario = Utility.UsuarioActual;  
18  
19         if (context.HttpContext.User  
20             .HasClaim("NotAuthorized", "NotAuthorized")  
21             || usuario == null)  
22         {  
23             context.Result = new RedirectToActionResult(  
24                 "HttpStatusCodeHandler",  
25                 "Error",  
26                 new { statusCode = 403 });  
27         }  
28         else  
29         {  
30             switch (AutorizacionPermisos) {  
31                 case EAutorizacionPermisos.NINGUNO:  
32                     if ((usuario.AutorizacionPermisos &
```

```
29             EAutorizacionPermisos.NINGUNO)
30         != EAutorizacionPermisos.NINGUNO)
31         context.Result = new RedirectToActionResult(
32             "HttpStatusCodeHandler",
33             "Error",
34             new { statusCode = 403 });
35         break;
36
37         case EAutorizacionPermisos.SOLICITANTE:
38         if ((usuario.AutorizacionPermisos &
39             EAutorizacionPermisos.SOLICITANTE)
40         != EAutorizacionPermisos.SOLICITANTE)
41         context.Result = new RedirectToActionResult(
42             "HttpStatusCodeHandler",
43             "Error",
44             new { statusCode = 403 });
45         break;
46
47         case EAutorizacionPermisos.PYV:
48         if ((usuario.AutorizacionPermisos &
49             EAutorizacionPermisos.PYV)
50         != EAutorizacionPermisos.PYV)
51         context.Result = new RedirectToActionResult(
52             "HttpStatusCodeHandler",
53             "Error",
54             new { statusCode = 403 });
55         break;
56
57         case EAutorizacionPermisos.ADMINISTRADOR:
58         if (((usuario.AutorizacionPermisos &
59             EAutorizacionPermisos.ADMINISTRADOR)
60         != EAutorizacionPermisos.ADMINISTRADOR) &&
61             !usuario.EsUsuarioSuplantado)
62         context.Result = new RedirectToActionResult(
63             "HttpStatusCodeHandler",
64             "Error",
65             new { statusCode = 403 });
66         break;
67
68         case EAutorizacionPermisos.RESPONSABLENEGOCIO:
69         if (((usuario.AutorizacionPermisos &
70             EAutorizacionPermisos.RESPONSABLENEGOCIO)
71         != EAutorizacionPermisos.RESPONSABLENEGOCIO)
```

```

66             && !usuario.EsUsuarioSuplantado)
67             context.Result = new RedirectToActionResult(
68                 "HttpStatusCodeHandler",
69                 "Error",
70                 new { statusCode = 403 });
71             break;
72
73         default:
74             context.Result = new RedirectToActionResult(
75                 "HttpStatusCodeHandler",
76                 "Error",
77                 new { statusCode = 403 });
78             break;
79         }
80     }
81 }
82 }
```

Listing 6.7: Clase que valida el permiso del usuario para poder consultar vistas o realizar acciones.

Este fragmento de código define una clase *TienePermisosFilterAttribute* que hereda de *AuthorizeAttribute* e implementa la interfaz *IAuthorizationFilter*. Su función es verificar si el usuario tiene los permisos necesarios para acceder a un recurso. En el constructor, se recibe un parámetro *AutorizacionPermisos*, que indica el nivel de permiso requerido. El método *OnAuthorization* comprueba si el usuario está autorizado para realizar la acción solicitada, evaluando su rol mediante un *switch* según el tipo de permiso. Si el usuario no tiene el permiso adecuado o está marcado como *NotAuthorized*, se redirige a una página de error con el código de estado 403.

El uso de este filtro en los controladores permite centralizar la gestión de permisos y evitar duplicación de lógica.

6.6. Sistema de notificaciones y enlaces únicos

El envío automático de correos electrónicos es una de las piezas clave del flujo interno. Tenemos un comando encargado de envío de correos después de que se aplique algún trámite, como por ejemplo al agregar un asistente a una formación le llega un correo para poder aceptar o rechazar la convocatoria. También tenemos otro controlador que es un recordatorio que envía diferentes correos según las fechas. Por ejemplo, cuando un asistente es agregado a una formación y queda una semana para que la formación comience, si ese asistente aún no ha aceptado ni rechazado la convocatoria automáticamente le llega un correo de recordatorio ya que cuando quede un solo día para que comience la formación, si no confirmó entonces se rechaza automáticamente.

6.7. Procesamiento de encuestas y valoraciones

Al finalizar una acción formativa, el sistema envía automáticamente una encuesta de eficacia. Cada respuesta queda asociada a un asistente y a una formación, y el promedio se calcula dinámicamente. Este proceso se refleja posteriormente en la pestaña *Valoraciones*, visible para los roles autorizados.

6.8. Conclusión técnica

El diseño interno del Ecosistema de Aprendizaje combina una arquitectura moderna con patrones avanzados y una estricta separación por capas. Esto ha permitido desarrollar un sistema robusto, mantenible y fácilmente ampliable. La modularidad del código y el uso de Entity Framework, servicios desacoplados y filtros de permisos garantizan que la aplicación pueda evolucionar en el futuro incorporando nuevas funcionalidades sin comprometer su estabilidad.

Bibliografía

- [1] (2025) Página de tracasa global. [Online]. Available: <https://tracasa.es>
- [2] (2025) Página de tracasa instrumental. [Online]. Available: <https://itracasa.es>
- [3] (2025) ¿qué es .net? [Online]. Available: <https://dotnet.microsoft.com/es-es/learn/dotnet/what-is-dotnet>
- [4] (2025) ¿qué es c#? [Online]. Available: <https://dotnet.microsoft.com/es-es/languages/csharp>
- [5] (2025) ¿qué es typescript? [Online]. Available: <https://codigofacilito.com/articulos/typescript>
- [6] (2025) ¿qué es html? [Online]. Available: <https://developer.mozilla.org/es/docs/Web/HTML>
- [7] (2025) ¿qué es css? [Online]. Available: <https://lenguajecss.com/css/introduccion/que-es-css/>
- [8] (2025) ¿qué es cqrs? [Online]. Available: <https://learn.microsoft.com/es-es/azure/architecture/patterns/cqrs>
- [9] (2025) ¿qué son las pruebas de integración? [Online]. Available: <https://www.ibm.com/es-es/think/topics/integration-testing>
- [10] (2025) ¿qué es un framework? [Online]. Available: <https://www.ibm.com/es-es/think/topics/integration-testing>
- [11] (2025) ¿qué es sql server? [Online]. Available: <https://www.microsoft.com/es-es/sql-server>

Apéndice

Modalidades		
<input type="checkbox"/> Filtros		
Nombre	Estado	
Aula Virtual (Streaming)	<input checked="" type="checkbox"/> Activo	<input type="checkbox"/> <input type="checkbox"/>
Mista	<input checked="" type="checkbox"/> Activo	<input type="checkbox"/> <input type="checkbox"/>
Presencial	<input checked="" type="checkbox"/> Activo	<input type="checkbox"/> <input type="checkbox"/>
Tel formación (Online)	<input checked="" type="checkbox"/> Activo	<input type="checkbox"/> <input type="checkbox"/>

Figura Apéndice.1: Listado de mantenimiento de modalidad.

Tipos de contenido		
<input type="checkbox"/> Filtros		
Nombre	Estado	
COMPETENCIAS TECNICAS	<input checked="" type="checkbox"/> Activo	<input type="checkbox"/> <input type="checkbox"/>
IDIOMAS	<input checked="" type="checkbox"/> Activo	<input type="checkbox"/> <input type="checkbox"/>
IGUALDAD, INCLUSIÓN Y DIVERSIDAD	<input checked="" type="checkbox"/> Activo	<input type="checkbox"/> <input type="checkbox"/>
METODOLOGIAS AGILE	<input checked="" type="checkbox"/> Activo	<input type="checkbox"/> <input type="checkbox"/>
PR/ SALUD LABORAL	<input checked="" type="checkbox"/> Activo	<input type="checkbox"/> <input type="checkbox"/>
SEGURIDAD DE LA INFORMACION	<input checked="" type="checkbox"/> Activo	<input type="checkbox"/> <input type="checkbox"/>
SISTEMAS DE GESTION	<input checked="" type="checkbox"/> Activo	<input type="checkbox"/> <input type="checkbox"/>
SOFT SKILLS	<input checked="" type="checkbox"/> Activo	<input type="checkbox"/> <input type="checkbox"/>

Figura Apéndice.2: Listado de tipos de contenido.

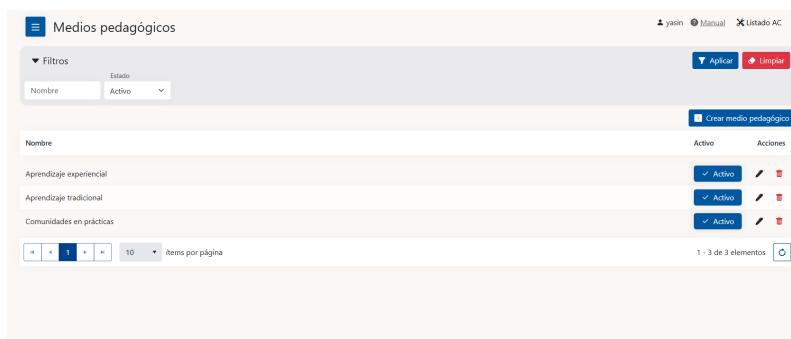


Figura Apéndice.3: Listado de mantenimiento de medios pedagógicos.

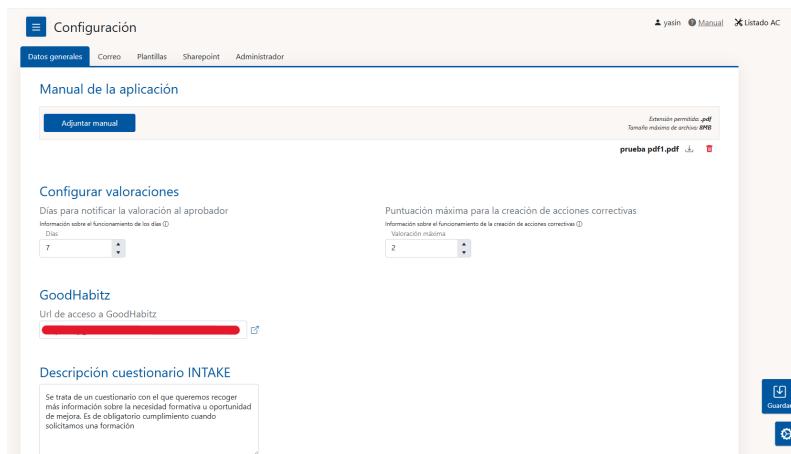


Figura Apéndice.4: Pestaña datos generales de configuración.

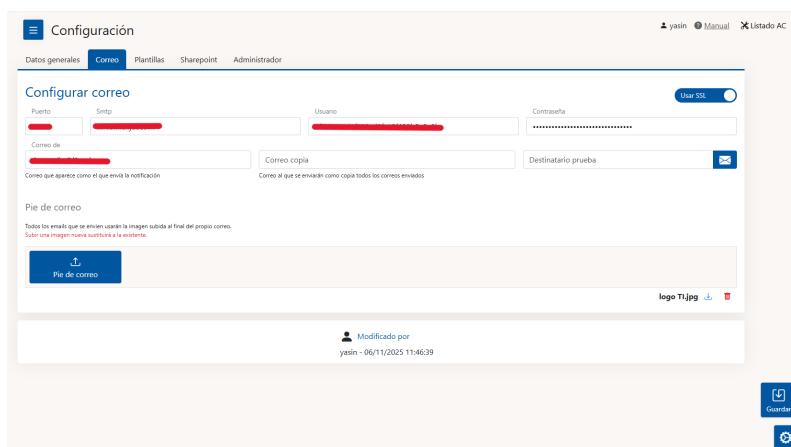


Figura Apéndice.5: Pestaña de configuración de correo.

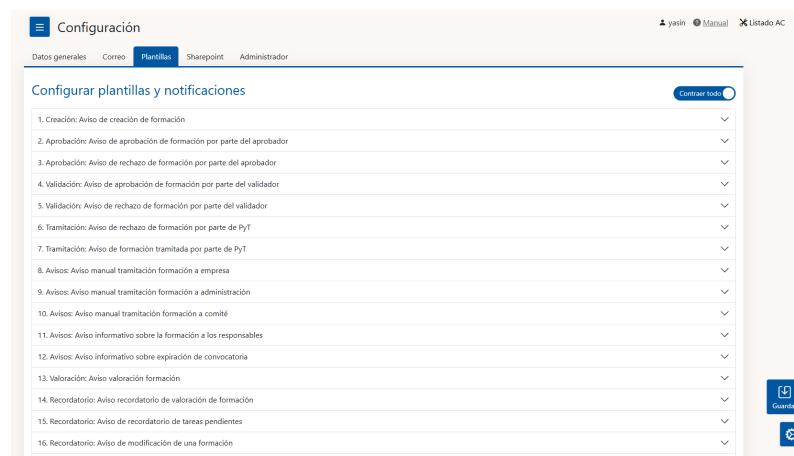


Figura Apéndice.6: Pestaña con las plantillas de los correos.

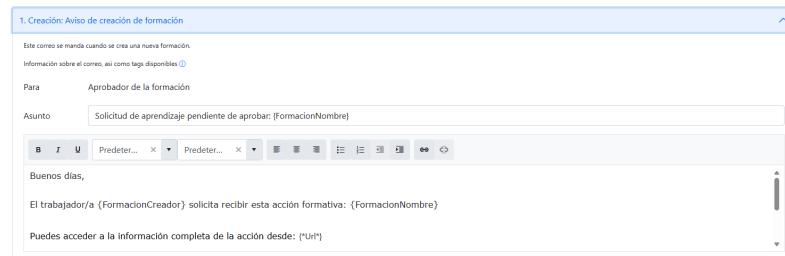


Figura Apéndice.7: Ejemplo con una plantilla de correo.

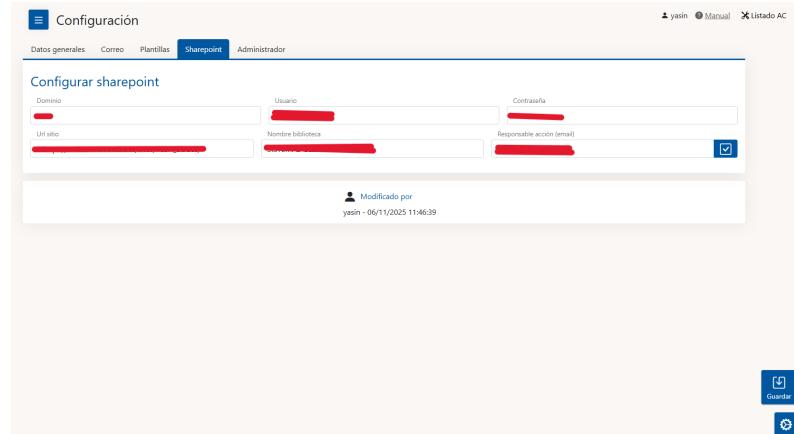


Figura Apéndice.8: Pestaña sharepoint de configuración.

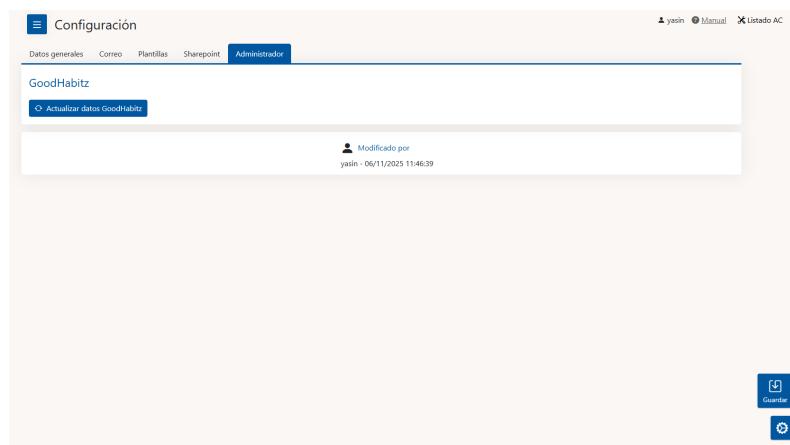


Figura Apéndice.9: Pestaña administrador de configuración.