



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

Autora:

Supervisor:
Nombre Apellido

Pamplona, 2022

Tu dedicatoria, opcional

Índice general

Índice de figuras	III
Índice de tablas	IV
Listings	VI
1. Introducción	1
2. Estructura del proyecto	3
2.1. Presentación	3
2.2. Aplicación	5
2.3. Infraestructura	6
2.4. Dominio	7
3. Proyecto	9
3.1. Acceso a plataforma de aprendizaje	10
3.2. Acceso directo a la página de Tracafit	10
3.3. Acceso directo a la plataforma de ciberseguridad	11
3.4. Solicitar nueva acción formativa	11
4. Conclusiones y Trabajo Futuro	15
Bibliografía	17
Índice alfabético	19
Apéndice	19

Índice de figuras

3.1. Pantalla home con los accesos directos	10
3.2. Pantalla plataforma de aprendizaje	10
3.3. Plataforma de ciberseguridad	11
3.4. Cuestionario INTAKE para solicitar formación	11
3.5. Formulario de datos generales a rellenar	12
3.6. Ventana para seleccionar propuesta a asignar a la formación	14

Índice de tablas

Listings

2.1. Controlador del proyecto WebUI con métodos de selección de formaciones almacenados en base de datos.	4
2.2. Interfaz de formaciones con métodos para llevar a cabo las operaciones sobre la base de datos.	6
Apéndice A. Mostrar notas.	19

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 y Tracasa Instrumental. 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 facilita el acceso a la formación, sino que también incorpora 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. Asimismo, ofrece funcionalidades avanzadas como la planificación de actividades, el seguimiento del progreso y la valoración de la eficacia formativa, contribuyendo a una mayor trazabilidad y transparencia en los procesos de desarrollo profesional.

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.

Estructura del proyecto

El proyecto se estructura siguiendo una arquitectura por capas, 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.

2.1. 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 [2.1](#), 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
8         filtrado");
9
10    var formacionesRespuesta = await Mediator.Send(new
11        GetAllFormacionesQuery(filtros));
12
13    if (formacionesRespuesta.Tipo == TipoRespuesta.Error)
14    {
15        ModelState.AddModelError("ListadoGridError", "Ha
16            ocurrido un error al obtener el listado de
17            formaciones.");
18    }
19
20    return
21        Json(formacionesRespuesta.Data.ToDataSourceResult(request,
22            ModelState));
23 }
24
25 [HttpGet("[action]/{FormacionID}")]
26 [TienePermisosFilter(EAutorizacionPermisos.SOLICITANTE)]
27 public async Task<IActionResult> Detalle(int formacionID)
```



```
21 {  
22     logger.LogInformation($"Pantalla de detalle de la  
23         formacion: {formacionID}");  
24  
25     var formacion = await Mediator.Send(new  
26         GetSingleFormacionQuery(formacionID));  
27  
28     if (formacion.Tipo == TipoRespuesta.Error)  
29     {  
30         TempData["NotificacionMensaje"] =  
31             NotificacionMensaje.FormacionObtenerError.ToString();  
32         return RedirectToAction("Index");  
33     }  
34     else if (formacion.Tipo == TipoRespuesta.NoAutorizado)  
35     {  
36         return RedirectToAction("HttpStatusCodeHandler",  
37             "Error", new { statusCode = 403 });  
38     }  
39  
40     var viewModel = new FormacionViewModel();  
41     viewModel.FormacionGeneral =  
42         viewModel.FormacionGeneral.GetViewModel(formacion.Data);  
43  
44     return View(viewModel);  
45 }
```

Listing 2.1: Controlador del proyecto WebUI con métodos de selección de formaciones almacenados en base de datos.

2.2. 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 2.2— 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,
14                             CancellationToken cancellationToken);
15     Task<bool> ComprobarValoracionCompletada(int
16                                             formacionID);
17     Task<FormacionDTO> Insert(FormacionDTO formacionDTO,
18                              CancellationToken cancellationToken);
19     Task<FormacionDTO> Update(FormacionDTO formacionDTO,
20                              CancellationToken cancellationToken);
21     Task<bool> Delete(int formacionID, CancellationToken
22                     cancellationToken);
23 }
```

Listing 2.2: Interfaz de formaciones con métodos para llevar a cabo las operaciones sobre la base de datos.

2.3. 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.

2.4. 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.

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. Se trata de un entorno que irá evolucionando, totalmente adaptado a las necesidades de la empresa. La aplicación nos ofrece varias funcionalidades y ciertas novedades:

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 recibirán fuera de jornada laboral.

Cuando accedemos a la aplicación, visualizamos la pantalla HOME, donde podremos elegir la acción que vayamos a realizar:

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

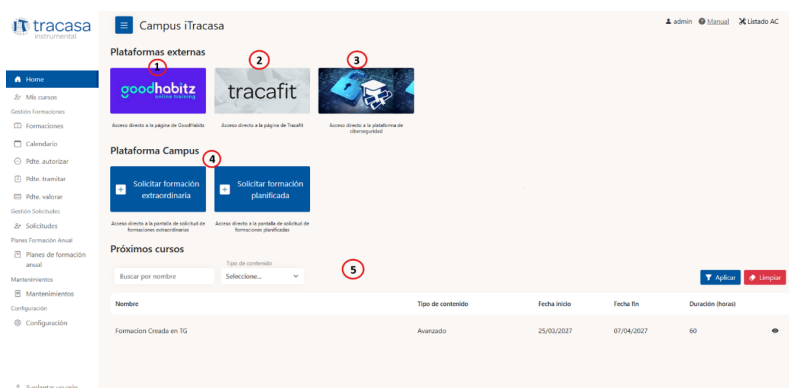


Figura 3.1: Pantalla home con los accesos directos¹

3.1. Acceso a plataforma de aprendizaje

Para acceder a la plataforma de aprendizaje debemos posicionarnos en la pantalla Home y seleccionar en GoodHabitiz 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. A

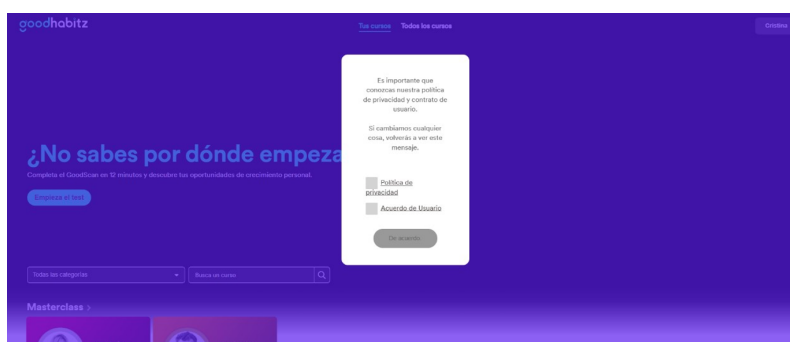


Figura 3.2: Pantalla plataforma de aprendizaje²

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.

3.2. Acceso directo a la página de Tracafit

Acceso directo al moodle de tracasa.

3.3. Acceso directo a la plataforma de ciberseguridad

Acceso directo a contenidos para mejorar las capacidades en ciberseguridad.

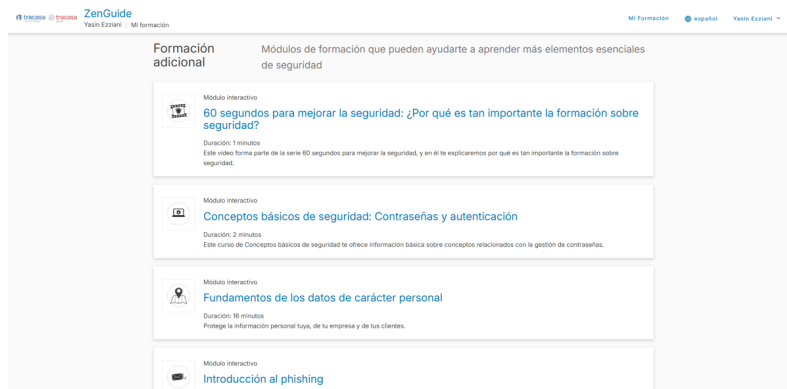


Figura 3.3: Plataforma de ciberseguridad³

3.4. Solicitar nueva acción formativa

Se proporciona 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 siguiente pantalla: Un cuestionario de 10 preguntas para recoger información sobre la necesidad

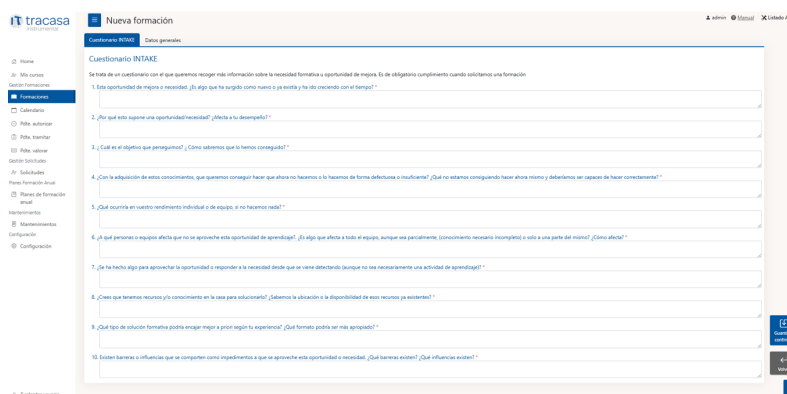


Figura 3.4: Cuestionario INTAKE para solicitar formación⁴

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 siguiente pantalla:

Figura 3.5: Formulario de datos generales a rellenar⁵

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

Es un listado de propuestas aceptadas de los planes de formación que estén aprobados y activos. Se selecciona una propuesta y haciendo click 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 rellenos.

Seleccione una propuesta del plan de formación

Filtros

Área: Seleccione... Tipo contenido: Seleccione... Nivel de formación: Seleccione...

Aplicar Limpiar

Código	Objetivo	Descripción	Tipo de contenido	Modalidad pedagógica	Planificación	Fecha estimada de inicio	Duración
PF27-002	sdf	sdf	Avanzado	Aprendizaje experiencial	T2	24/09/2023	
PF30-001	sdf	sdf	Avanzado	Aprendizaje experiencial	T2	24/09/2023	

1 - 2 de 2 propuestas

Propuestas por página

Aceptar Cancelar

Figura 3.6: Ventana para seleccionar propuesta a asignar a la formación⁶

Conclusiones y Trabajo Futuro

Bibliografía

Apéndice

```
1 public void mostrarTodasNotas() {  
2     int a = 1;  
3     System.out.println("Estas_son_todas_las_notas_que_hay_  
4         guardadas.");  
5     for(int j=0; j<lista_notas.size(); j++){  
6         System.out.print(a+"-");  
7         System.out.println(lista_notas.get(j).toString());  
8         a++;  
9     }  
}
```

Listing Apéndice.1: Mostrar notas.