UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Migración de grupos y permisos de seguridad de usuarios para un ERP del sector sociosanitario

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Aas Alas, Mohamed

Tutor/a: Letelier Torres, Patricio Orlando

Cotutor/a externo: SUAREZ GRUESO, FRANCISCO MANUEL

CURSO ACADÉMICO: 2022/2023

Dedicatoria

A mis queridos padres y a mi adorada abuela, cuyo amor y apoyo trascienden la distancia geográfica. Aunque estamos lejos, su cariño y creencia en mí siempre ha sido una fuente inagotable de motivación y fortaleza.

Agradecimientos

Primero que nada, quiero expresar mi más profundo agradecimiento a aquellas personas que, de una forma u otra, han formado parte de esta maravillosa aventura que culmina hoy con la presentación de este Trabajo de Fin de Grado.

Quiero comenzar agradeciendo a mis padres, pilares fundamentales en mi vida. Gracias por vuestro amor incondicional, por vuestra paciencia y por inculcarme valores tan importantes como el respeto, la disciplina y el trabajo duro. A mis hermanos, gracias por ser mi apoyo y mi motivación constante, por creer en mí incluso cuando yo no lo hacía. Vuestra fe en mí ha sido mi mayor impulso para llegar hasta aquí.

En segundo lugar, quiero agradecer a mis amigos de la carrera, compañeros de desvelos, trabajos en equipo y también de risas y buenos momentos. Sin vuestro apoyo y compañía, este camino habría sido mucho más difícil. Habéis sido un elemento esencial en mi crecimiento personal y académico, y no tengo palabras suficientes para agradeceros por ello.

Deseo dar las gracias especialmente a mi profesor, Patricio Letelier. Su dedicación, paciencia y sabiduría han sido una luz guía en mi proceso de aprendizaje. Gracias por inspirarme a buscar siempre más allá, por fomentar mi curiosidad y por impartir conocimientos con pasión. Su apoyo ha sido crucial para la finalización de este Trabajo de Fin de Grado.

Finalmente, me gustaría agradecer a mis compañeros de la empresa ADD Informática. Gracias por acogerme en vuestro equipo, por enseñarme y guiarme. Vuestra experiencia y conocimientos han sido fundamentales para mi formación, y los valores que promovéis en el día a día me han ayudado a crecer como profesional.

En resumen, gracias a todos vosotros he conseguido finalizar este Trabajo de Fin de Grado. Este logro no es solo mío, es también vuestro, porque sin vuestra ayuda, apoyo y cariño, no habría sido posible. A todos vosotros, mi más sincero agradecimiento.

Resumen

Este proyecto se enmarca en el contexto de unas prácticas en empresa, y se enfoca en el desarrollo de un sistema de planificación de recursos empresariales (ERP) para la gestión y administración en el sector sociosanitario. El trabajo forma parte del departamento de investigación, desarrollo e innovación (I+D+i) y está orientado hacia la creación de una versión mejorada del producto, con nuevas funcionalidades y mejoras respecto a la versión existente.

La finalidad de este trabajo es la de implementar la migración de la seguridad de un ERP dentro del microservicio de migración de datos integrado dentro de la estructura del nuevo producto software compuesto por diferentes microservicios. La migración de la seguridad se encargará de trasladar los grupos y permisos de seguridad de los usuarios desde la versión actual del ERP a la nueva versión.

Los grupos de seguridad son colecciones de usuarios que tienen permisos comunes en el sistema. Por ejemplo, un grupo de seguridad podría ser el de los gerentes, que tienen acceso a información y funciones específicas dentro del ERP. Los permisos de seguridad, por su parte, definen qué acciones pueden realizar los usuarios dentro del sistema. Por ejemplo, un permiso podría ser el de "crear facturas de venta".

Durante el proceso de migración de seguridad, se identifican los grupos y permisos de seguridad existentes en la versión actual del ERP y se transfieren a la nueva versión. Esto asegura que los usuarios tengan los mismos niveles de acceso y permisos que tenían antes de la actualización, y que no pierdan la capacidad de realizar acciones importantes en el sistema.

Este proyecto se ha desarrollado dentro del marco del enfoque de Desarrollo Dirigido por Modelos (MDD) lo que ha simplificado el proceso de trabajo. Además, se ha empleado una metodología ágil y se han realizado pruebas simultáneamente al proceso de implementación. Para asegurar su correcto funcionamiento, se han llevado a cabo diversas pruebas con datos reales en relación a esta migración de seguridad.

Se ha trabajado con la tecnología ASP.NET Core, y para la creación de los modelos se ha utilizado una herramienta de DSL Tools. El lenguaje de programación que se ha usado es C#.

Palabras clave: ERP, seguridad en el software, migración, microservicios, MDD.

Resum

Aquest projecte s'emmarca en el context d'unes pràctiques en empresa, i s'enfoca en el desenvolupament d'un sistema de planificació de recursos empresarials (ERP) per a la gestió i administració en el sector sociosanitari. El treball forma part del departament d'investigació, desenvolupament i innovació (I+D+i) i està orientat cap a la creació d'una versió millorada del producte, amb noves funcionalitats i millores respecte a la versió existent.

La finalitat d'aquest treball és la d'implementar la migració de la seguretat d'un ERP dins del microservei de migració de dades integrat dins de l'estructura del nou producte software compost per diferents microserveis. La migració de la seguretat s'encarregarà de traslladar els grups i permisos de seguretat dels usuaris des de la versió actual del ERP a la nova versió.

Els grups de seguretat són col·leccions d'usuaris que tenen permisos comuns en el sistema. Per exemple, un grup de seguretat podria ser el dels gerents, que tenen accés a informació i funcions específiques dins del ERP. Els permisos de seguretat, per part seua, defineixen quines accions poden realitzar els usuaris dins del sistema. Per exemple, un permís podria ser el de "crear factures de venda".

Durant el procés de migració de seguretat, s'identifiquen els grups i permisos de seguretat existents en la versió actual del ERP i es transfereixen a la nova versió. Això assegura que els usuaris tinguen els mateixos nivells d'accés i permisos que tenien abans de l'actualització, i que no perden la capacitat de realitzar accions importants en el sistema.

Aquest projecte s'ha desenvolupat dins del marc de l'enfocament de Desenvolupament Dirigit per Models (MDD) el que ha simplificat el procés de treball. A més, s'ha emprat una metodologia àgil i s'han realitzat proves simultàniament al procés d'implementació. Per a assegurar el seu correcte funcionament, s'han dut a terme diverses proves amb dades reals en relació a aquesta migració de seguretat.

S'ha treballat amb la tecnologia ASP.NET Core, i per a la creació dels models s'ha utilitzat una eina de DSL Tools. El llenguatge de programació que s'ha usat és C#.

Paraules clau: ERP, seguretat en el software, migració, microserveis, MDD.

Abstract

This project is part of the context of an internship in a company, and focuses on the development of an enterprise resource planning (ERP) system for management and administration in the social and healthcare sector. The work is part of the research, development and innovation (R+D+i) department and is oriented towards the creation of an improved version of the product, with new features and improvements compared to the current version.

The purpose of this work is to implement the security migration of an ERP within the integrated data migration microservice within the structure of the new software product composed of different microservices. The security migration will take care of transferring the groups and security permissions of the users from the current version of the ERP to the new version.

Security groups are collections of users who have common permissions on the system. For example, a security group could be the managers, who have access to specific information and functions within the ERP. Security permissions, for their part, define what actions users can perform within the system. For example, a permission could be to "create sales invoices".

During the security migration process, existing security groups and permissions in the current version of the ERP are identified and transferred to the new version. This ensures that users have the same levels of access and permissions that they had before the upgrade, and that they do not lose the ability to perform important actions on the system.

This project has been developed within the framework of the Model Driven Development (MDD) approach, which has simplified the work process. In addition, an agile methodology has been used and tests have been carried out simultaneously with the implementation process. To ensure its correct operation, various tests have been carried out with real data in relation to this security migration.

We have worked with ASP.NET Core technology, and a DSL Tools tool has been used to create the models. The programming language that has been used is C#.

Keywords : ERP, security in software, migration, microservices, MDD.

Tabla de contenidos

Índice de Figuras

Índice de Tablas

CAPÍTULO PRIMERO

1. Introducción

1.1 Motivación

Este TFG se desarrolla en el marco de una práctica en una empresa dedicada al desarrollo y mantenimiento de un ERP del sector socio-sanitario, específicamente en el departamento de I+D+i donde se está desarrollando una nueva versión de su ERP. La seguridad en las aplicaciones es esencial para garantizar la protección de los datos y la privacidad de los usuarios, así como para evitar posibles ciberataques que puedan afectar a la integridad del sistema. En el caso del desarrollo de una nueva versión de un ERP, esta preocupación es aún más importante, ya que los ERP contienen una gran cantidad de información confidencial y sensible de la empresa y sus clientes, como datos financieros y de recursos humanos.

La migración de datos, en concreto los de seguridad es una tarea crítica y necesaria cuando se actualiza o se cambia a una nueva versión del sistema ERP. Esta tarea toma especial relevancia en nuestra empresa, que cuenta con más de 2300 clientes. En cada uno de ellos, están definidos sus grupos de seguridad y los usuarios dentro de estos con elementos de seguridad asociados y los permisos que indican si se tiene o no acceso a un determinado elemento de seguridad, estos elementos constituyen nuestro sistema de seguridad. La migración de la seguridad, por tanto, implica transferir estos datos a la nueva versión del ERP para garantizar una consistencia de la seguridad. A medida que la nueva versión del ERP esté disponible, debemos asegurarnos de que este sistema de seguridad de cada cliente se migre eficazmente, preservando la integridad de los datos de seguridad en todo el proceso.

Dentro de un sistema ERP, los grupos son conjuntos de usuarios que tienen acceso a ciertas funciones o recursos del sistema, mientras que los elementos de seguridad son los objetos o recursos a los que se puede acceder.

Los permisos de seguridad son las reglas que se establecen para permitir o denegar el acceso a los elementos de seguridad. Estos permisos pueden ser configurados a nivel de grupo o de usuario, lo que significa que un usuario o grupo puede tener acceso a un elemento de seguridad mientras que otro no lo tiene. Los permisos de seguridad se utilizan para garantizar que los datos empresariales estén seguros y que solo los usuarios autorizados tengan acceso a ellos.

1.2 Objetivos

El objetivo principal de este trabajo es desarrollar el proceso para la migración de los datos de seguridad (grupos y permisos de seguridad de usuarios) dentro del microservicio de migración de un sistema de planificación de recursos empresariales (ERP).

Un objetivo específico que hay que desarrollar para el correcto funcionamiento de la migración de seguridad, es el registro de las migraciones de los elementos de seguridad en un histórico de migraciones, ya que la migración es un proceso que puede tardar días o incluso más, por el gran volumen de datos a migrar. Este histórico nos permite gestionar cualquier cambio en la seguridad durante el proceso de migración, es decir, permite identificar cualquier discrepancia en los datos migrados y detectar si algún elemento ha sido modificado o eliminado después de haber sido migrado anteriormente.

Es importante destacar que el proceso de migración está diseñado para lanzarse una vez y migrar todos los elementos de la ERP a la nueva ERP. Luego, se realizan ciclos en los que se comprueba si hay cambios respecto a lo registrado en el histórico, y en caso de que existan, se migra de nuevo dicho elemento. Es por ello que el registro de cada elemento de seguridad migrado en el histórico de migraciones debe realizarse cada vez que se lleve a cabo una migración exitosa. Al hacerlo de esta manera, podemos monitorizar cualquier cambio en la seguridad durante el proceso de migración. Además, el seguimiento continuo de los elementos migrados en el histórico nos permitirá detectar si se han modificado después de la migración. Esta es una tarea crucial ya que cualquier cambio en la ERP actual debe ser replicado en la nueva versión, y cualquier elemento eliminado en la ERP actual después de su migración debe ser eliminado también en la nueva versión. De esta manera, se garantiza la integridad y consistencia de los datos en ambas versiones de la ERP.

Otro objetivo específico importante de la migración de seguridad de un sistema ERP es que el cliente no ha de notar ningún cambio en su experiencia. La migración de seguridad se ha de llevar a cabo en segundo plano, lo que significa que no habrá interrupciones en el servicio ni afectará a la calidad de los productos o servicios ofrecidos por el sistema. Esto se hace para que los clientes puedan usar ambas versiones de la ERP hasta que un determinado momento decidan usar únicamente el nuevo ERP. Mientras aún se utilice la versión actual, el proceso de migración debe estar activo para garantizar la consistencia entre las dos versiones de los datos de los clientes. De esta manera, se asegura una transición suave y sin problemas a la nueva versión del sistema ERP por parte del cliente.

1.3 Estructura de la memoria

Esta memoria se divide en los siguientes capítulos:

- En este primero se introduce el trabajo a realizar explicando la motivación y los objetivos a cumplir del mismo, junto con la estructura que seguirá el documento.
- El segundo capítulo es una introducción al tema, en la que se explicarán una serie de conceptos y mecanismos fundamentales para poder entender el trasfondo y la base del proyecto en cuestión
- El tercer capítulo presenta de forma breve el estado del arte, referente a las tecnologías que se utilizan actualmente para la gestión y manipulación de datos, también se hablará de otras herramientas que podrían haberse utilizado para llevar a cabo un proyecto similar.
- El cuarto capítulo comenta las tecnologías que se han empleado para desarrollar este proyecto
- El quinto capítulo explica el procedimiento para el desarrollo de la solución. Este capítulo empieza explicando la metodología seguida durante el proceso, continua con un planteamiento generalista del proyecto a realizar, siguiendo por una especificación de requisitos, junto con el diseño y estructura del mismo, también contiene un apartado que habla de la programación, continúa explicando las pruebas realizadas y acaba comentando la cronología del proceso.
- El sexto y último capítulo ofrece las conclusiones de este trabajo, así como el trabajo futuro que queda por realizar para la finalización del proyecto hasta su posible despliegue.

CAPÍTULO SEGUNDO

2. Introducción a la seguridad y migraciones

Para entender este proyecto, es importante tener en cuenta dos conceptos clave: seguridad y migración de aplicaciones. La seguridad se refiere a la protección de los sistemas y datos frente a amenazas externas o internas que pueden comprometer su integridad, confidencialidad o disponibilidad. Por su parte, la migración de aplicaciones hace referencia al proceso de trasladar una aplicación de un entorno a otro, por ejemplo, de un servidor a la nube. Además, proporcionaremos una explicación del concepto de microservicios junto con sus beneficios para entender mejor su uso en un contexto determinado.

2.1 Seguridad en aplicaciones

Para entender bien la seguridad en las aplicaciones esta se va a definir de la siguiente manera: La seguridad de las aplicaciones es el proceso de desarrollar, añadir y probar funciones de seguridad dentro de las aplicaciones para evitar vulnerabilidades de seguridad contra amenazas como el acceso y la modificación no autorizados [1].

La seguridad en las aplicaciones se refiere a las precauciones y medidas implementadas para proteger la información y el código dentro de una aplicación, protegiéndola contra posibles robos o ataques informáticos. Estas medidas abarcan consideraciones de seguridad durante el desarrollo y diseño de la aplicación, así como sistemas y enfoques posteriores a la implementación para protegerla. Los métodos de seguridad pueden ser de hardware, software o procedimientos, y se enfocan en identificar y minimizar vulnerabilidades de seguridad. Por ejemplo, un enrutador que oculta la dirección IP de una computadora es una medida de seguridad hardware, mientras que un firewall de aplicación que define estrictamente qué actividades están permitidas y prohibidas es una medida de seguridad software. Los procedimientos de seguridad pueden incluir rutinas de pruebas periódicas para garantizar que la aplicación esté continuamente protegida.

Es crucial considerar la seguridad de las aplicaciones debido a que en la actualidad éstas se encuentran disponibles a través de varias redes y conectadas a la nube, lo que las hace más vulnerables a amenazas, como ataques informáticos. Además, hay una creciente presión y

motivación para asegurar no solo la seguridad en la red, sino también dentro de las aplicaciones en sí. Esto se debe a que los piratas informáticos están enfocando más sus ataques en las aplicaciones. Mediante la realización de pruebas de seguridad en las aplicaciones podemos descubrir debilidades a nivel de la aplicación y así prevenir estos ataques.

Además, la seguridad es importante por el principio del "Privilegio mínimo" se refiere a que los empleados solo deben tener los privilegios necesarios para cumplir con sus funciones laborales, y nada más. Esto limita el potencial daño que puede causar un empleado, ya sea accidentalmente o a propósito [2].

2.1.1 Tipos de seguridad de las aplicaciones

Existen diversas categorías de tipos de seguridad para aplicaciones, entre las que se encuentran la autenticación, autorización, cifrado, registro y pruebas de seguridad de aplicaciones [1].

- **Autenticación**: medidas que permiten restringir el acceso únicamente a los usuarios que están autorizados. Para ello, se utilizan procedimientos de autenticación que verifican la identidad del usuario que desea ingresar al sistema. Una forma de autenticación común consiste en solicitar al usuario que proporcione un nombre de usuario y una contraseña al momento de iniciar sesión. Por otro lado, la autenticación multifactor implica el uso múltiples formas de autenticación, como por ejemplo algo que el usuario conoce (como una contraseña), algo que el usuario posee (como un dispositivo móvil) y algo que es característico de ese usuario (como una huella digital o el reconocimiento facial), esto último se conoce como autenticación biométrica. De esta manera, se aumenta el nivel de seguridad y se reduce el riesgo de acceso no autorizado al sistema.
- **Autorización**: Después de que un usuario ha sido autenticado, puede obtener la autorización necesaria para acceder y utilizar la aplicación. Para comprobar si un usuario tiene permiso para utilizar la aplicación, el sistema puede contrastar su identidad con una lista de usuarios autorizados. Es importante que la autenticación se realice antes de la autorización, para asegurarse de que la aplicación sólo confirme las credenciales de usuario que hayan sido validadas con la lista de usuarios autorizados. Podemos ver una diferencia entre la Autorización y la Autenticación en la *figura 1*:

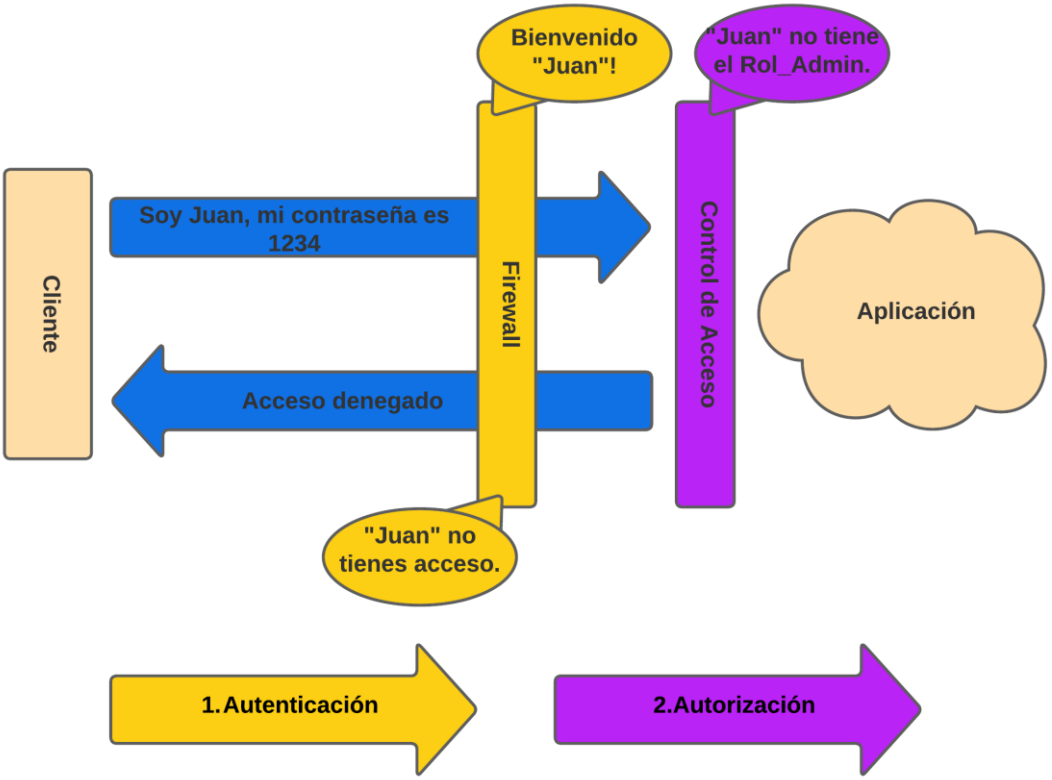


Figura 1. Esquema Autenticación y Autorización

- **Cifrado**: Una vez que un usuario ha iniciado sesión y está haciendo uso de la aplicación, existen otras medidas de seguridad que pueden proteger la información confidencial de ser visualizada o utilizada por un ciberdelincuente. En el caso de aplicaciones basadas en la nube, donde los datos confidenciales viajan desde el usuario hasta la nube, es posible cifrar el tráfico para garantizar la seguridad de los datos. El emisor de los datos cifra los datos con una clave privada, los datos cifrados pasan por un "canal inseguro", una vez llegan al receptor este descifra los datos con la clave privada.
- **Registro**: Si una aplicación tiene una vulnerabilidad de seguridad, los registros pueden servir para identificar a la persona que logró acceder a los datos y la forma en que lo hizo. Los archivos de registro de la aplicación registran el momento en que se accedió a las distintas partes de la aplicación y por quién.
- **Pruebas de seguridad de aplicaciones**: Es esencial llevar a cabo pruebas de seguridad en la aplicación para identificar posibles debilidades y corregirlas, con el fin de asegurarse de que todos los controles de seguridad operen adecuadamente.

2.2 Migración de aplicaciones

Para entender bien la migración en las aplicaciones ésta se va a definir de la siguiente manera: La migración de aplicaciones es el proceso de mover una aplicación de software de un entorno informático a otro, como cambiarla de un centro de datos a otro, de un servidor local a un proveedor de servicio en la nube, o de la nube pública a un entorno de nube privada [3]. En esta descripción, cabe resaltar que se está llevando a cabo un trabajo que involucra la migración de datos. Este proceso es una tarea crucial y una subparte integral de la migración de aplicaciones. La migración de datos implica la transferencia de datos desde sistemas antiguos a sistemas nuevos.

La creación de aplicaciones para sistemas operativos y arquitecturas de red específicas o para una única plataforma en la nube puede presentar dificultades al momento de trasladarlas a otro entorno. En este sentido, suele ser más complicado migrar aplicaciones que se ejecutan en hardware sin sistema operativo en comparación con aquellas que están basadas en servicios o virtualizadas.

Para llevar a cabo una migración exitosa, es fundamental considerar las dependencias y requerimientos técnicos de cada aplicación en particular, así como también considerar las restricciones de seguridad y presupuesto de la empresa. De esta forma, se podrá determinar una estrategia general adecuada para la migración de aplicaciones.

2.2.1 Tipos de migraciones de las aplicaciones

Existen diferentes estrategias de migración de aplicaciones, las cuales se pueden adaptar según las necesidades y objetivos de cada organización [4].

- **Rehost**: El "lift-and-shift", también conocido como traslado directo, es una táctica común que consiste en trasladar una aplicación desde un servidor local a una máquina virtual en la nube sin realizar modificaciones importantes. Esta estrategia suele ser más rápida que otras estrategias de migración y puede reducir significativamente los costes involucrados. Sin embargo, la desventaja es que, al no realizar cambios, las aplicaciones no aprovechan las características nativas de la nube, lo que puede generar costos más elevados a largo plazo para su funcionamiento en la nube [5].

- **Refactorizar o rediseñar**: La refactorización implica realizar modificaciones significativas en una aplicación con el fin de mejorar su rendimiento o escalabilidad en un entorno de nube. Estos cambios pueden incluir la reescritura de secciones cruciales de la aplicación para aprovechar las características nativas de la nube, como la transformación de una aplicación monolítica en un conjunto de microservicios o la actualización del sistema de gestión de bases de datos de SQL a NoSQL.
- **Cambiar de plataforma**: Modificar la plataforma de una aplicación se refiere a realizar ajustes pequeños en la misma, que permitan aprovechar mejor las ventajas de la arquitectura de la nube, y puede entenderse como una especie de punto medio entre las dos estrategias de migración anteriores. Por ejemplo, puede implicar actualizar la aplicación para que sea compatible con una base de datos nativa en la nube, cambiar los sistemas operativos o el middleware utilizados, o incluso utilizar contenedores para su ejecución.
- **Retirar/reemplazar**: En ocasiones, puede ser más lógico eliminar la aplicación debido a su utilidad limitada, porque sus funcionalidades se encuentran en otras partes del entorno o porque resulta mejor a nivel económico sustituirla por una nueva alternativa, como una plataforma de software como servicio (SaaS), en lugar de realizar la migración de la aplicación.

CAPÍTULO TERCERO

3. Estado del arte

En este capítulo se abordarán temas relacionados con la migración de la seguridad de un sistema ERP. Para ello, es necesario considerar el contexto actual en el que se encuentra la versión del ERP, lo que permitirá comprender las posibles soluciones para migrar la seguridad al nuevo sistema. En este sentido, se analizará cómo funciona la seguridad en el ERP actual y en el nuevo ERP, lo que será fundamental para establecer posteriormente el algoritmo de migración de los elementos de seguridad del ERP actual a la nueva versión. Para ello, se tendrán en cuenta aspectos como la gestión de usuarios, los permisos y las restricciones de acceso, y las herramientas de control y auditoría. De esta manera, se buscará brindar una visión amplia y detallada sobre cómo se puede llevar a cabo una migración de seguridad eficiente y exitosa.

3.1 Seguridad Actual vs Seguridad objetivo

3.1.1 Seguridad del producto actual

El sistema ERP actual implementa un modelo de seguridad basado en grupos de permisos y permisos hijos, así como asociaciones de usuario-grupo. Este es un enfoque común en la gestión de seguridad en sistemas de software, especialmente aquellos que necesitan manejar accesos a una variedad de recursos y funciones.

Aquí detallamos los conceptos de seguridad:

- **Grupos con permisos**: Un grupo es un conjunto de usuarios que comparten un conjunto común de permisos para los elementos de seguridad. Los permisos asignados a un grupo se aplican a todos los usuarios que son miembros de ese grupo. Por ejemplo, podríamos tener un grupo llamado "Contabilidad" con permisos para acceder a funciones financieras y de contabilidad en el sistema ERP.
- **Permisos hijos**: Este concepto hace referencia a un modelo jerárquico de elementos de seguridad, donde algunos elementos de seguridad pueden tener "elementos de seguridad hijos" bajo ellos, como podemos ver en la *figura 2*. Por ejemplo, un

permiso de "Gestión Financiera" podría tener permisos hijos como "Ver Informes Financieros", "Crear Informes Financieros", etc. Esto permite una gran flexibilidad y granularidad en la gestión de permisos, permitiendo asignar derechos de acceso muy específicos a diferentes grupos o usuarios.

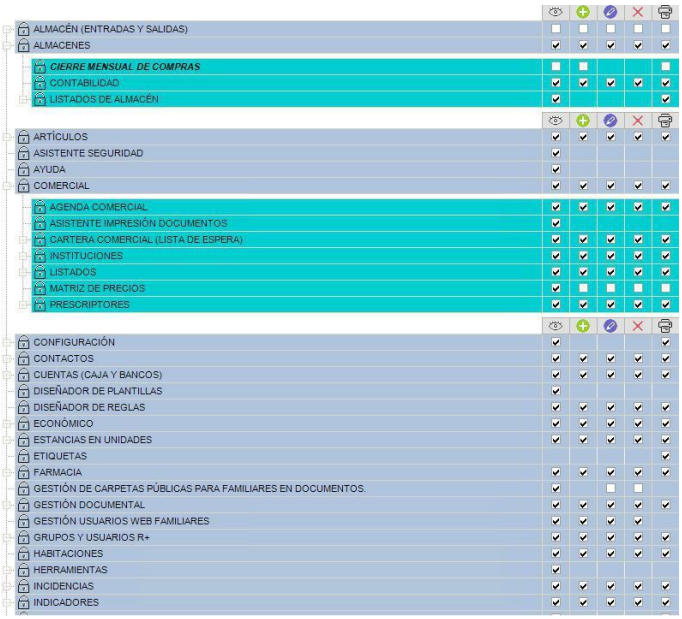


Figura 2. Permisos y Permisos hijos

- **Usuarios y grupos**: Cada usuario en el sistema puede ser miembro de un solo grupo. Los permisos del usuario se derivan del grupo al que pertenecen. En el sistema actual, un usuario solo puede pertenecer a un grupo. Esto significa que los permisos de un usuario están claramente definidos por su afiliación grupal, lo que simplifica la gestión de permisos. La *figura 3* Muestra esta distribución:

Figura 3. Usuarios y Grupos de Seguridad

- **Elementos de seguridad y vistas**: Las vistas en un sistema ERP hacen referencia a las diferentes páginas o secciones de la interfaz de usuario. Los elementos de seguridad están asociados a estas vistas, lo que significa que, si un grupo tiene un permiso para un elemento de seguridad, en consecuencia, tiene el permiso para una determinada vista, y los usuarios de ese grupo pueden acceder a esa vista. Los permisos pueden también controlar qué acciones se pueden realizar en una vista (por ejemplo, ver, crear, editar o eliminar datos). También haciendo referencia a lo explicado anteriormente, los elementos de seguridad son contenedores de otros elementos de seguridad(hijos) para poder asignar los permisos de forma grupal de forma más rápida.

Aunque en general un usuario hereda los permisos del grupo al que pertenece, en algunos casos, un usuario puede tener permisos “especiales”. Esto significa que dicho usuario tendría acceso a más o menos funciones o vistas en el ERP que los demás miembros de su grupo.

3.1.2 Seguridad Objetivo en la nueva versión del producto

En la nueva versión del ER, la seguridad está estructurada con un enfoque basado en roles y permisos, que está diseñada para proporcionar un control flexible y efectivo del acceso a los recursos y funcionalidades del sistema.

Aquí detallamos los conceptos de seguridad:

- **Roles**: Los roles son definiciones de las responsabilidades o funciones dentro de una organización. Por ejemplo, podrías tener roles como "Gerente", "Contador" o "Administrador de Inventarios". Cada rol se configura con una serie de permisos que definen lo que los usuarios asignados a ese rol pueden hacer dentro del sistema ERP.
- **Usuarios**: Los usuarios son las personas que usan el sistema ERP. Cada usuario puede ser asignado a uno o más roles, y los permisos de esos roles determinan lo que el usuario puede hacer en el sistema. Por ejemplo, si un usuario está asignado a los roles de "Contador" y "Administrador de Inventarios", tendría todos los permisos asociados a ambos roles.
- **Permisos y Entidades**: En el contexto del ERP, las entidades se refieren a los diversos tipos de datos o recursos que el sistema maneja, como los registros de clientes, las órdenes de compra o las facturas. Los permisos se definen en relación con estas

entidades. Por ejemplo, un rol podría tener permiso para "ver" y "editar" entidades de facturas, pero solo "ver" entidades de clientes. En esta versión del ERP, la seguridad está enfocada a los datos y no a los elementos de seguridad (vistas). la principal implicación de esto es que si un dato aparece en varias vistas y un usuario de la aplicación no tienen acceso a ese dato, no va a poder ver ese dato en ninguna de las vistas donde aparece, sin embargo, en la versión actual del ERP, la persona gestora de la seguridad le tiene que quitar el acceso al usuario de la aplicación explícitamente en cada una de las vistas en las que aparece el dato, con el consiguiente aumento de probabilidad de fallo y aumento de tiempo en la gestión de los permisos.

En contraste con la seguridad en la versión actual del ERP, la nueva versión del ERP opta por una estructura de permisos más simplificada en donde los roles solo poseen permisos positivos. A diferencia de la versión actual, donde los grupos pueden tener tanto permisos positivos (autorizaciones para hacer algo) como negativos (restricciones para hacer algo), la nueva versión únicamente permite asignar capacidades, evitando las restricciones explícitas.

3.2 Propuestas de servicios para migrar datos del ERP

La migración de datos es un componente esencial en el proceso de migración de aplicaciones, ya que se trata del traslado de datos de un entorno a otro. Conscientes de esta importancia, nuestra propuesta de servicios de migración de aplicaciones incluirá un conjunto de herramientas dedicadas a la migración de datos.

En esta sección, se explicarán algunos de los servicios de migración de aplicaciones que proporcionan herramientas especializadas para la tarea de migración de datos. Estas herramientas facilitan el traslado eficiente y seguro de la información entre diferentes sistemas.

3.2.1 Docker y kubernetes

Docker es una plataforma abierta que permite a los desarrolladores y administradores de sistemas construir, empaquetar, y distribuir aplicaciones de manera eficiente. Docker utiliza tecnología de contenedores para permitir que las aplicaciones se empaqueten junto con sus bibliotecas y dependencias, en un objeto estándar que puede ser ejecutado de manera consistente en cualquier sistema que soporte Docker [6].

Un contenedor Docker es como una versión ligera de una máquina virtual, pero en lugar de empaquetar un sistema operativo completo, solo incluye los componentes necesarios para ejecutar la aplicación específica. Esto significa que los contenedores son mucho más eficientes en términos de recursos que las máquinas virtuales.

Ahora bien, ¿cómo puede Docker ayudarnos a migrar una aplicación?

1. **Empaquetado de la aplicación**: Puede empaquetar la aplicación y todas sus dependencias en un contenedor Docker. Esto se realiza utilizando un archivo llamado `Dockerfile`, que especifica qué software, bibliotecas y dependencias se requieren para ejecutar la aplicación.
2. **Consistencia en diferentes entornos**: Una vez que la aplicación está empaquetada en un contenedor Docker, puedes ejecutarla en cualquier sistema que soporte Docker, ya sea la máquina local, un servidor en la nube o un cluster de servidores. Esto elimina problemas comunes como "en mi máquina funciona" y facilita la migración de la aplicación entre diferentes entornos.
3. **Migración de la aplicación**: Para migrar la aplicación, simplemente necesitas instalar Docker en el nuevo sistema y ejecutar el contenedor Docker que contiene la aplicación. Esto se puede hacer descargando el contenedor desde un repositorio de Docker como Docker Hub, o transfiriéndolo directamente desde un sistema actual.
4. **Escalado y orquestación**: Si necesitas migrar la aplicación a un entorno de alta disponibilidad o de alta escala, Docker también puede ayudar. Herramientas como Docker Compose[1], Docker Swarm[2] o Kubernetes[3] (que también es compatible con Docker) pueden gestionar múltiples contenedores Docker, permitiéndote escalar la aplicación a medida que la necesidad crece.

3.2.2 Amazon Web Services (AWS) Migration Hub

Amazon Web Services (AWS) es una plataforma de servicios de computación en la nube que proporciona una variedad de servicios de infraestructura como servicio (IaaS), plataforma como servicio (PaaS) y software como servicio (SaaS). Ofrece servicios en diversas áreas como computación, almacenamiento, redes, base de datos, análisis, aplicaciones y despliegue [7].

Los servicios de AWS pueden ser utilizados para migrar una aplicación existente a la nube. AWS proporciona la herramienta "AWS Migration Hub" que ayuda a planificar y monitorear el proceso de migración [8].

[1] Docker Compose: Gestiona apps multi-contenedor Docker. Más info
[2] Docker Swarm: Orquesta clústeres de nodos Docker. Más info
[3] Kubernetes: Automatiza implementación y gestión de apps contenerizadas. Más info

AWS ofrece varios servicios para soportar diferentes tipos de aplicaciones. Por ejemplo, si la aplicación se ejecuta en un servidor, se puede elegir EC2 (Elastic Compute Cloud). Si es una aplicación sin servidor, se puede elegir AWS Lambda. Si se necesita una base de datos, se puede elegir entre varios tipos como RDS (Relational Database Service), DynamoDB (NoSQL), etc [9].

AWS Database Migration Service (DMS) puede ayudarnos a migrar una base de datos a la nube con tiempo de inactividad mínimo. Para grandes cantidades de datos, se puede hacer uso de AWS Snowball, un servicio de transferencia de datos petabyte-scale [10].

Una vez que los datos estén en AWS, podemos comenzar a migrar la aplicación. Podemos utilizar AWS Server Migration Service (SMS) para migrar servidores enteros a la nube.

Una vez que la aplicación está en la nube, AWS ofrece herramientas para la optimización de la aplicación, para su seguridad, asi como para la monitorización y gestión de esta.

3.2.3 Google Cloud Platform (GCP) Migration Tools

Google Cloud Platform (GCP) ofrece una variedad de herramientas para facilitar la migración de aplicaciones a su plataforma. Estas herramientas pueden ayudar a mover aplicaciones, bases de datos y otros componentes de infraestructura desde entornos locales, otros servicios en la nube o incluso entre proyectos y servicios dentro de GCP [11].

Aquí se describe algunas de las herramientas de migración más destacadas de GCP:

1. **Migrate for Compute Engine**: Esta herramienta puede ayudarnos a mover nuestras máquinas virtuales (VM) a Google Cloud. Podemos migrar VM desde sistemas físicos, Compute Engine, AWS, Azure y VMware.
2. **Transfer Service**: Es un servicio en línea para mover datos a y desde Google Cloud Storage. Este servicio admite transferencias de datos desde fuentes en línea, como Amazon S3, HTTP/HTTPS, Google Cloud Storage y buckets de datos de la nube de Azure, y desde fuentes de datos locales.
3. **Database Migration Service (DMS)**: Permite migrar bases de datos MySQL y PostgreSQL a Cloud SQL. DMS ofrece una migración de datos segura, fácil de usar y confiable.
4. **BigQuery Data Transfer Service**: Este servicio ayuda a mover datos desde aplicaciones SaaS a Google BigQuery de forma programada y administrada.
5. **Migrate for Anthos**: Permite migrar aplicaciones desde sistemas físicos o VM a contenedores en Google Kubernetes Engine (GKE).

3.2.4 Microsoft Azure Migrate

Microsoft Azure Migrate es una solución de Microsoft diseñada para simplificar el proceso de migración de servidores, aplicaciones, bases de datos y otras cargas de trabajo a la nube de Azure [12].

Selección de un escenario de migración: Dentro del proyecto de Azure Migrate, se puede seleccionar el escenario de migración que se desea utilizar, dependiendo del tipo de datos o servicios que se desea migrar. Las opciones incluyen la migración de máquinas y cargas de trabajo a Azure (Evaluar y migrar servidores), la migración de bases de datos locales (Evaluar y migrar bases de datos), la migración de aplicaciones web locales, y la migración de datos a Azure mediante Data Box.

Selección de una herramienta de migración: Se pueden agregar herramientas de migración a su proyecto de Azure Migrate. Por ejemplo, si creó un proyecto a través de la opción "Evaluar y migrar servidores", la herramienta "Migración y modernización" se agrega automáticamente al proyecto.

Aquí están algunas de las herramientas principales que forman parte de Azure Migrate: [13]

- **Server Assessment**: Esta herramienta permite hacer una evaluación de tus servidores antes de migrarlos. Proporciona detalles sobre las dependencias entre aplicaciones y servidores, tamaño adecuado de las VMs en Azure, estimaciones de costos, etc.
- **Server Migration**: Esta herramienta proporciona un medio de migrar servidores físicos y virtuales hacia Azure. Permite la migración de máquinas virtuales de VMware, servidores físicos y máquinas virtuales de Hyper-V hacia Azure.
- **Database Assessment**: Esta herramienta proporciona una evaluación detallada de las bases de datos que planeas migrar a Azure. Da detalles sobre la compatibilidad, el rendimiento y las recomendaciones de destino.
- **Database Migration**: Esta herramienta ayuda a migrar tus bases de datos a Azure. Permite la migración de bases de datos de SQL Server a Azure SQL Database, y otras migraciones de bases de datos.
- **Web App Migration**: Esta herramienta proporciona una forma de evaluar y migrar aplicaciones web a Azure.
- **Data Box:** Para migrar grandes cantidades de datos a Azure, se puede solicitar una instancia de Azure Data Box para la transferencia de datos sin conexión. Se puede especificar la suscripción y el grupo de recursos que quiere usar para solicitar una

instancia de Data Box, así como el país en el que residen los datos y la región de Azure a la que quiere transferir los datos.

3.4 Comparativa

Se ha llevado a cabo una comparativa entre las diferentes tecnologías de migración estudiadas: Docker, AWS Migration Hub, GCP Migration Tools, Microsoft Azure Migrate y el microservico de migración ya desarrollado por la empresa con el objetivo de seleccionar la más adecuada para el proceso de migración de la seguridad de nuestro ERP. Los criterios evaluados en esta comparativa como se puede ver en la *tabla 1* incluyen el precio, el destino de almacenamiento final, la personalización y el tiempo de desarrollo. Al sopesar estos factores, se busca entender las ventajas y desventajas de cada tecnología, lo que permitirá tomar una decisión fundamentada sobre cuál utilizar en función de nuestras necesidades y limitaciones específicas.

Tool	Precio	Destino de almacenamiento final	Personalización	Tiempo de desarrollo
Docker	Gratis. Las soluciones empresariales tienen costos adicionales.	Contenedor Docker	Requiere conocimientos técnicos	Corto a Largo
AWS Migration Hub	Gratis a Pago	AWS	Personalización limitada	Corto a mediano
GCP migration Tools	Gratis a Pago	GCP	Personalización limitada	Corto a mediano
Microsoft Azure Migrate	Gratis a Pago	Azure o proveedor externo	Personalización limitada	Corto a mediano
Miroservicio de Migración	Gratis	Nuestros servidores	Requiere conocimientos técnicos	El tiempo de desarrollo del microservicio

Tabla 1. Tabla Comparativa de Tecnologías para migración

3.5 Conclusiones

Después de una evaluación de los servicios y herramientas de migración de aplicaciones disponibles en el mercado que se han explicado a lo largo de los apartados anteriores, hemos decidido emplear el microservicio ya desarrollado por nuestra empresa para la migración de datos. Este microservicio será utilizado para migrar la seguridad de la aplicación a la nueva versión del ERP. Ésta es la decisión lógica y fijándonos en la tabla comparativa es la más eficiente, tanto en términos de tiempo como de costes. Al aprovechar las capacidades existentes, optimizaremos el proceso de migración y garantizaremos una transición suave y segura hacia el nuevo sistema. También tiene la ventaja que, al utilizar la propia arquitectura de la empresa, se aprovecha el uso de las DSL Tools y la generación automática del código de migración que se explicaran en el siguiente capitulo o al menos de parte de él. Además, se tiene un control total sobre cómo se debe hacer el proceso de migración para que la herramienta pueda ser usada por los clientes finales.

CAPÍTULO CUARTO

4. Tecnologías utilizadas

En el desarrollo de nuestro proyecto, hemos usado una serie de tecnologías robustas y modernas para garantizar la eficiencia y la escalabilidad de nuestra solución. Estas tecnologías, seleccionadas por su capacidad para abordar los desafíos específicos del proyecto son: ASP.NET Core como framework para el proyecto.

El lenguaje de programación utilizado es C#, el IDE seleccionado para el desarrollo del proyecto es Visual Studio (VS), se ha hecho uso de las Domain Specific Language (DSL) Tools para el modelado y la generación de código y por último Azure DevOps para gestionar el proyecto y poder trabajar en diferentes ramas de forma paralela entre los departamentos. A continuación, explicamos las características de algunas de estas tecnologías:

4.1 DSL Tools

Las DSL Tools (Domain-Specific Language Tools) son un conjunto de herramientas que permiten a los desarrolladores crear lenguajes de dominio específico (DSL) personalizados para abordar problemas particulares en un contexto específico. Estos lenguajes especializados se diseñan para ser utilizados por expertos en el dominio y aportan una representación más natural y específica del problema a resolver [14]. Una vez definida la DSL, esta se puede utilizar para crear modelos abstractos que representen un problema o sistema particular en el dominio elegido. Estos modelos, a su vez, sirven como base para generar código fuente en un lenguaje de programación de propósito general, automatizando y agilizando el proceso de desarrollo de software.

Hemos utilizado las DSL desarrolladas en la empresa para crear modelos que representan de manera precisa el dominio de nuestras aplicaciones. Las DSL nos han permitido generar automáticamente el código base necesario para desarrollar la solución de forma más rápida y eficiente. Al utilizar las DSL, hemos logrado acelerar el proceso de desarrollo al evitar la necesidad de escribir manualmente grandes cantidades de código repetitivo. Además, las DSL también nos han proporcionado la capacidad de modelar y generar pruebas automatizadas para verificar el correcto funcionamiento de todas las funciones implementadas.

Cabe destacar los elementos más representativos y utilizados al modelar con las DSL que encontramos en la toolbox que podemos observar en la *figura 4*. Para modelar las pruebas tenemos los siguientes elementos:

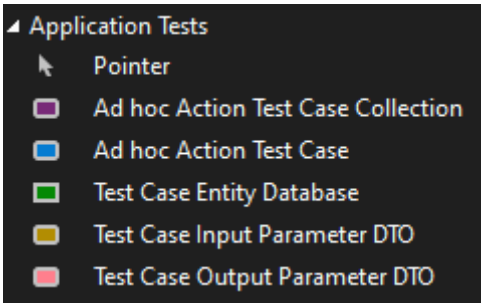
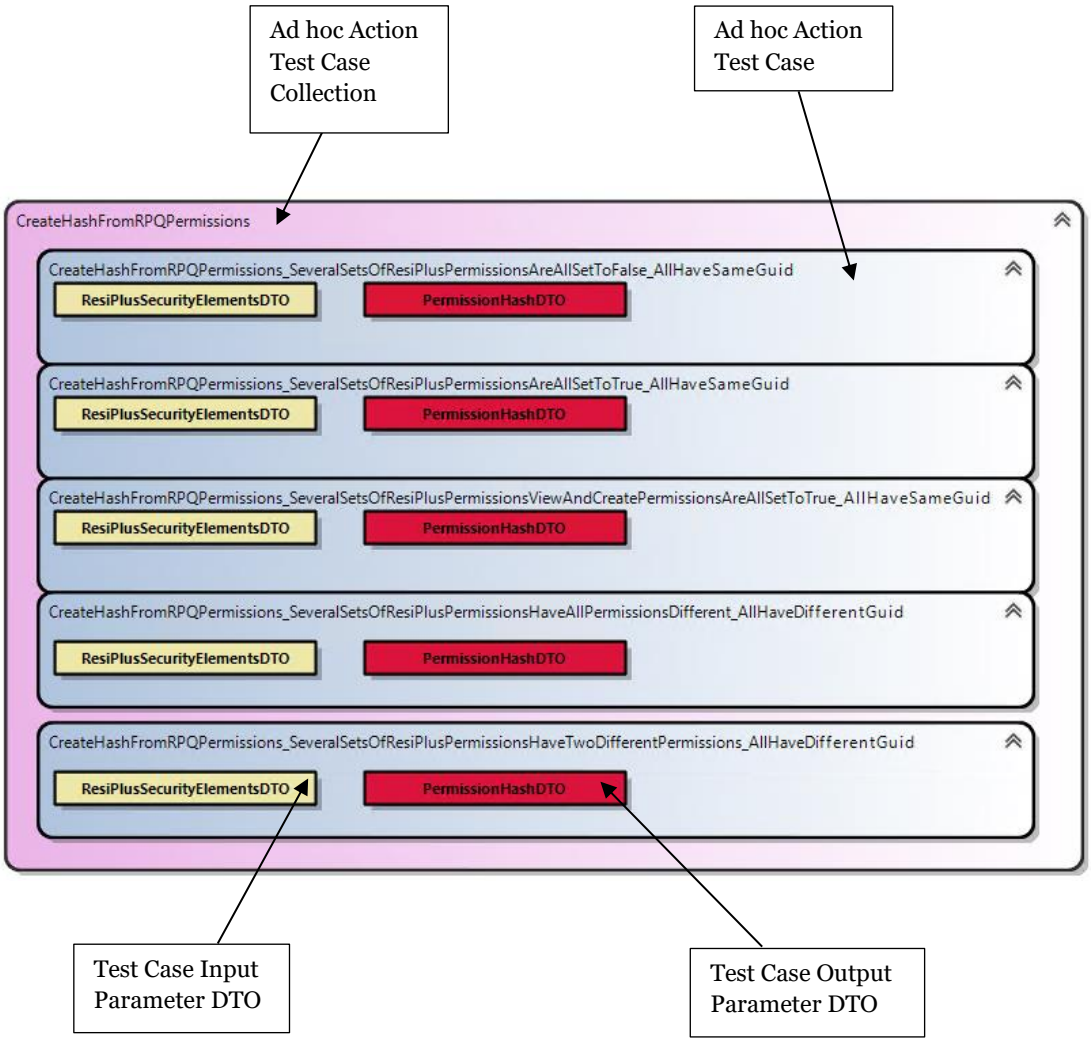


Figura 4. DSL Tool box Application Test

- **Ad hoc Action Test Case Collection**: Son agrupaciones o clases de pruebas. Sirven como contenedores para agrupar múltiples pruebas que se relacionan entre sí.
- **Ad hoc Action Test Case**: Estos se encuentran dentro de las colecciones de Ad hoc Action Test Case y representan pruebas individuales dentro de dicha clase.
- **Test Case Entity Database**: Este elemento permite crear datos ficticios/falsos de entrada para la base de datos especificada.
- **Test Case Input Parameter DTO**: Representa los parámetros de entrada para un caso de prueba específico. Estos parámetros son los que se utilizan para ejecutar el caso de prueba.
- **Test Case Output Parameter DTO**: Este elemento representa la salida esperada para un caso de prueba Ad hoc Action Test. Esencialmente, es lo que se compara con el resultado real obtenido después de ejecutar el caso de prueba para determinar si la prueba fue exitosa o no.

La siguiente figura proporciona un ejemplo concreto de un diseño de una clase test usando un modelo de las DSL. Dicho ejemplo muestra la estructura, los atributos y los métodos específicos para una clase test en el contexto de las DSL. De esta manera, sirve como guía para entender los modelos de test que hemos elaborado con nuestras DSL.



Hay que mencionar que al modelar un Ad hoc Action Test Case, los DTOs tanto de entrada como de salida son componentes fundamentales. Estos DTOs, que sirven para transferir datos entre procesos, deben estar previamente modelados en las DSL tools. Por lo tanto, cualquier elemento que se requiera referenciar para los DTOs en el contexto de un Ad hoc Action Test Case debe ser modelado y definido previamente en esta herramienta, asegurando así una adecuada coherencia y funcionalidad en el proceso de modelado y ejecución de pruebas.

Para modelar las clases y métodos tenemos otra variedad de elementos que vemos en la *figura 5*, pero solo comentaremos los más importantes, aquellos que se han utilizado más en el desarrollo de la solución del proyecto:

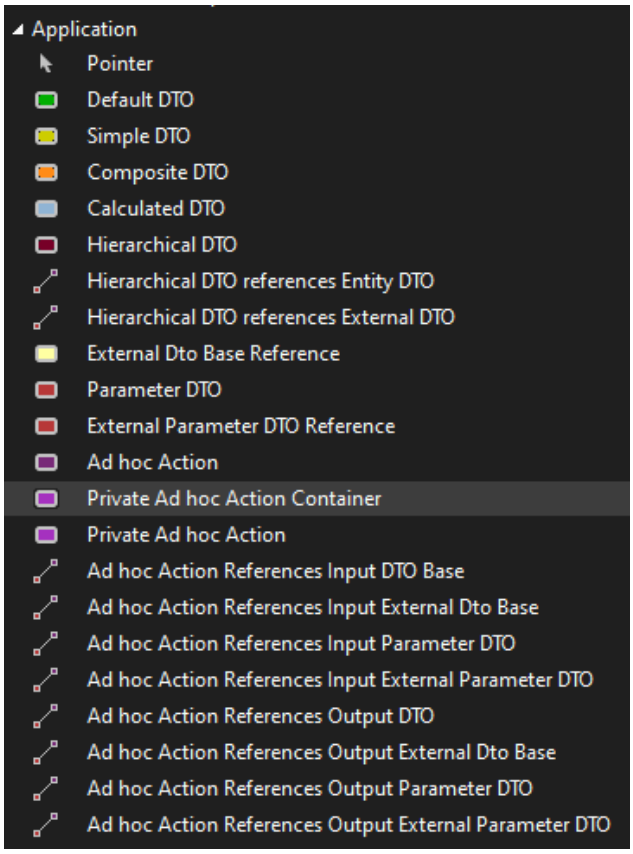
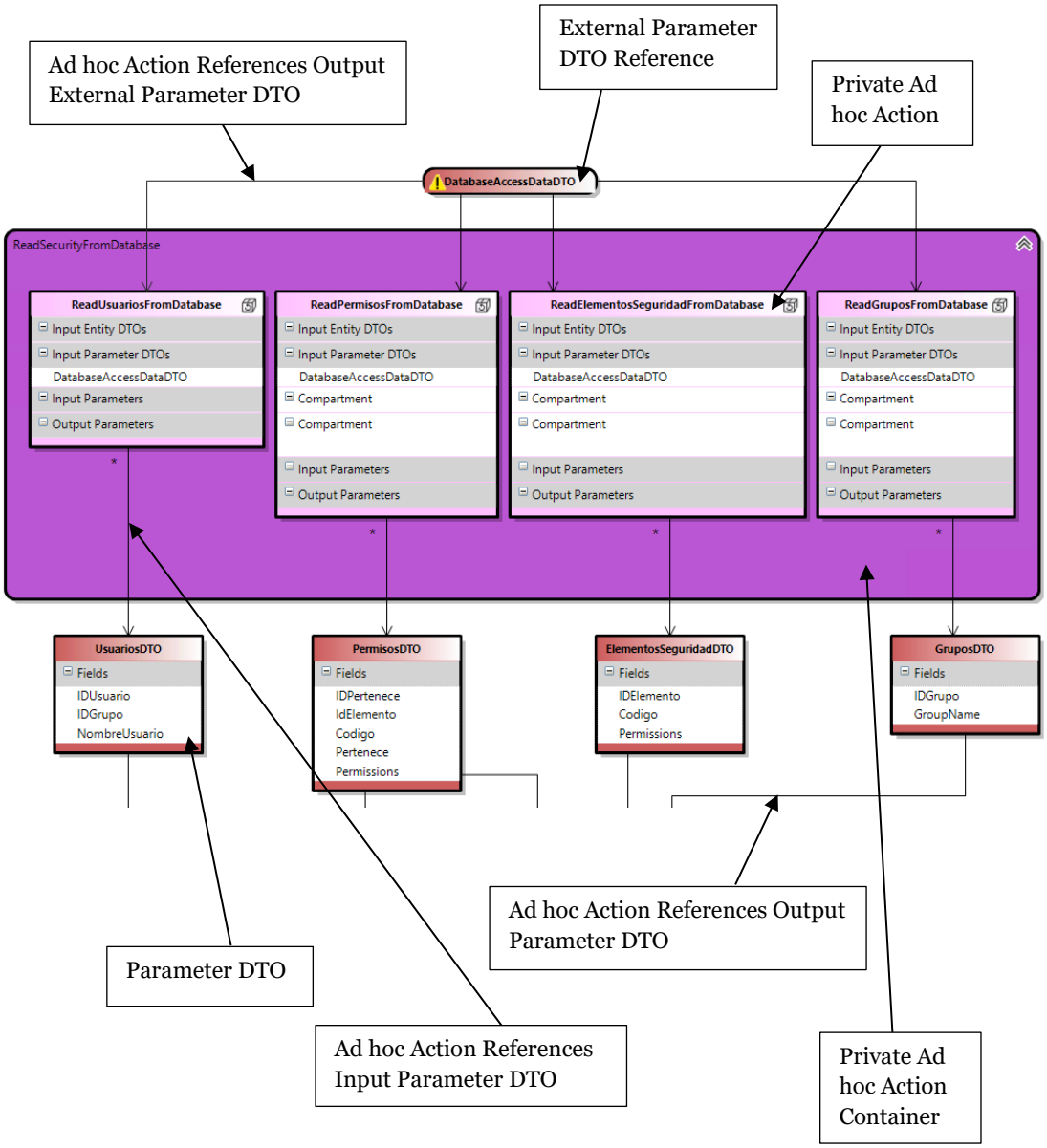


Figura 5. DSL Tool box Application

- **Parameter DTO**: Este elemento es crucial para definir los parámetros de los datos que serán transferidos entre diferentes partes de la aplicación (Ad Hoc Actions).
- **External Parameter DTO Reference**: Este componente permite hacer referencia a los parámetros de los DTOs que se encuentran fuera del modelo actual.
- **Private Ad Hoc Action Container**: Este es un contenedor especializado para agrupar las acciones ad hoc privadas de un modelo específico, serían las clases.
- **Private Ad Hoc Action**: Estas son acciones que se pueden diseñar y aplicar a un modelo de manera personalizada y privada, serían los métodos de la clase.
- **Ad Hoc Action References Input External Parameter DTO Reference**: Este elemento permite que una acción ad hoc haga referencia a un parámetro de entrada de un DTO externo.
- **Ad Hoc Action References Output External Parameter DTO Reference**: Similar al anterior, este permite que una acción ad hoc haga referencia a un parámetro de salida de un DTO externo.

- **Ad Hoc Action References Output Parameter DTO**: Este componente posibilita que una acción ad hoc haga referencia a un parámetro de salida de un DTO dentro del mismo modelo.
- **Ad Hoc Action References Input Parameter DTO**: Este permite que una acción ad hoc haga referencia a un parámetro de entrada de un DTO dentro del mismo modelo.

La siguiente figura proporciona un ejemplo concreto de un diseño de una clase usando un modelo de las DSL. Dicho ejemplo muestra la estructura, los atributos y los métodos específicos para una clase en el contexto de las DSL. De esta manera, sirve como guía para entender los modelos de clases que hemos elaborado con nuestras DSL.



Cabe destacar que los modelos que creamos utilizando nuestra herramienta DSL se guardan en archivos con la extensión .dsl. Después, mediante un script batch, estos archivos .dsl son procesados para generar los correspondientes archivos en C#, que serán utilizados para diversas funcionalidades dentro de nuestra infraestructura de software.

4.2 Azure DevOps Server

Azure DevOps es una plataforma de colaboración y gestión del ciclo de vida de desarrollo de software que proporciona herramientas y servicios para planificar, desarrollar, probar y entregar aplicaciones de software de manera eficiente. Permite a los equipos de desarrollo trabajar juntos de manera más efectiva, automatizar los procesos de entrega de software y garantizar la calidad del código [15].

Se hace uso de una de las características clave de Azure DevOps, su capacidad para gestionar el flujo de trabajo de desarrollo utilizando repositorios de código fuente basados en Git. Los equipos pueden crear ramas (*branches*) para trabajar en nuevas funcionalidades. Una vez que se completa el trabajo en una rama, se crea una *Pull Request* (PR) para fusionar los cambios en la rama principal o en otra rama.

Cuando se crea una PR, los revisores revisan los cambios realizados y dejar comentarios para proporcionar *feedback*. Esto permite una colaboración efectiva entre los miembros del equipo, ya que los revisores pueden señalar problemas, sugerir mejoras o hacer preguntas directamente en la PR para aclarar cualquier cosa. Los comentarios en una PR pueden ser los *commits* realizados sobre el código, comentarios identificando algún problema que se ha de arreglar o incluso puede proporcionar orientación adicional como una forma de hacer las cosas o algo que todo el equipo que trabaja en dicha PR ha de tener en cuenta.

Además de los comentarios, en el desarrollo de este proyecto se ha hecho uso de la descripción de la PR como se puede ver en la *figura 6*, esta es útil para proporcionar contexto adicional sobre el trabajo que se está realizando. Puede incluir una lista de tareas realizadas y pendientes por implementar, fecha de inicio y de integración prevista para la PR, o cualquier otro tipo de información relevante. Esto ayuda a mantener a todos los miembros del equipo informados y al tanto de los avances asociados con la PR.

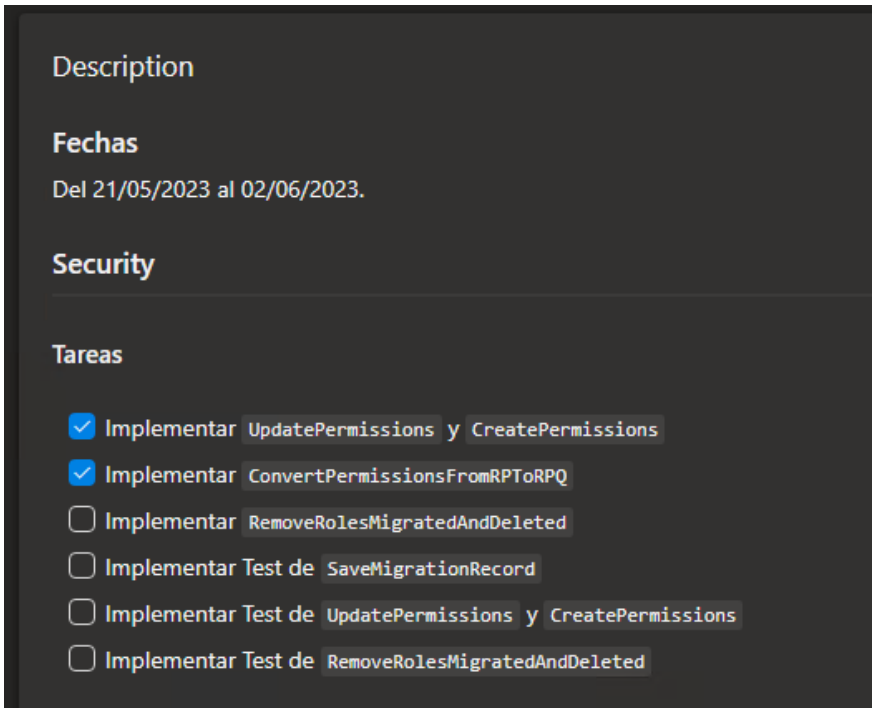


Figura 6. Descripción PR Azure DevOps

Azure DevOps también ofrece la posibilidad de ejecutar automatizaciones, como pruebas de integración o compilaciones, antes de fusionar una PR a la rama Master. Esto se puede lograr utilizando servicios de compilación y despliegue continuo integrados en la plataforma. Además, se pueden configurar BOTs o tareas automatizadas para comprobar que todo funciona correctamente antes de aceptar los cambios.

En el desarrollo de este proyecto, se ha creado una rama por cada sprint que se ha querido abordar. Una vez concluido el sprint, se ha tenido que integrar dicho sprint en la rama Master. Para ello, hemos tenido que pasar el proceso de validación con el BOT, lo que garantiza que todos los cambios son seguros y funcionales antes de ser fusionados.

Además, en este proceso, se ha llevado a cabo un 'rebase sobre Master', que consiste en traer los cambios más recientes desde la rama Master a la rama del sprint. Este paso adicional asegura que cada integración está al día con los últimos cambios en el código base, minimizando así los posibles conflictos.

4.3 Aplicaciones basadas en microservicios

La arquitectura de microservicios se compone de varios servicios pequeños y autónomos que tienen la capacidad de llevar a cabo funciones específicas de negocio dentro de un contexto definido. Cada uno de estos servicios es independiente y se enfoca en una tarea individual. El contexto delimitado se refiere a una sección natural de la empresa que define un modelo de dominio con un límite explícito [16]. Cabe mencionar este sistema ya que el desarrollo del proyecto se lleva a cabo utilizando un sistema de microservicios, el cual presenta numerosas ventajas y contribuye significativamente a la eficacia del proyecto.

En una arquitectura de microservicios convencional se pueden encontrar elementos adicionales como vemos en la *figura 7* además de los propios servicios:

- **Administración (orquestador)**: Este elemento se encarga de la ubicación de servicios en los nodos, la detección de fallos, la redistribución de servicios entre nodos, entre otras funciones. Usualmente, se utiliza una tecnología estándar como Kubernetes en lugar de desarrollar algo a medida.
- **Servicio**: Los servicios de un microservicio se refieren a las diferentes funciones independientes dentro de una arquitectura de microservicios. Cada servicio es un componente de software autónomo y desacoplado que realiza una función específica dentro de un sistema más grande. Los microservicios se comunican entre sí a través de APIs y pueden ser desarrollados, implementados y escalados independientemente, lo que permite flexibilidad y robustez en sistemas complejos.
- **Puerta de enlace de API**: La entrada para que los clientes accedan a los servicios es la puerta de enlace de la API. En lugar de contactar directamente con los servicios, los clientes hacen la solicitud a la puerta de enlace, la cual redirige la solicitud a los servicios correspondientes en el back-end.

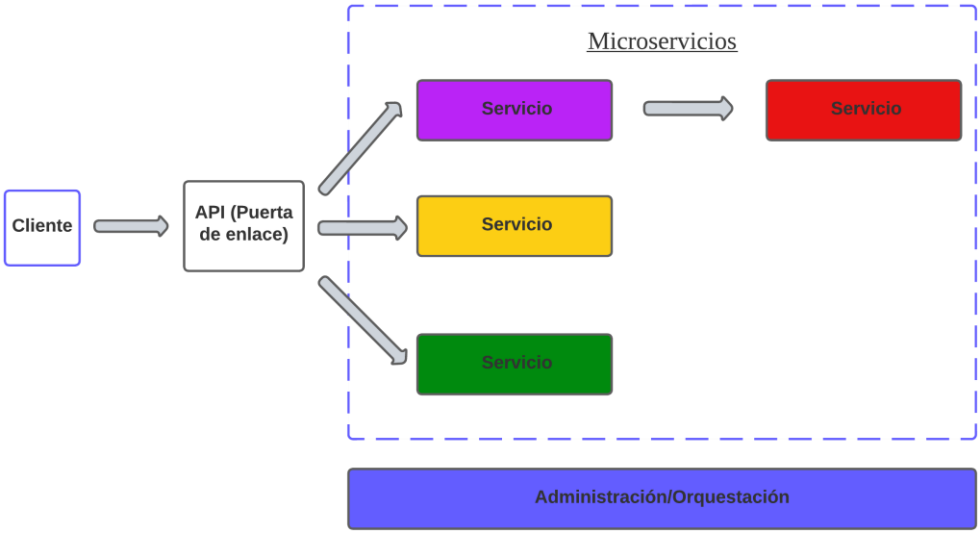


Figura 7. Arquitectura basada en Microservicios

Los microservicios aparecen como una opción para solucionar los inconvenientes de las aplicaciones monolíticas, pero conllevan nuevos desafíos y, al mismo tiempo, ofrecen varias ventajas:

- **Agilidad**: Debido a que los microservicios se despliegan de manera autónoma, resulta menos complicado manejar las actualizaciones y correcciones de errores. Es posible mejorar un servicio sin necesidad de reinstalar completamente la aplicación, y también se puede deshacer una actualización en caso de algún problema.
- **Aislamiento de errores y datos**: Si un componente de un microservicio no se encuentra operativo, no afectará el funcionamiento del conjunto de la aplicación en su totalidad.
- **Base de código pequeña**: Estos son de menor tamaño gracias a una base de código reducida, lo que los hace más sencillos de alterar o sustituir si se utiliza la misma interfaz o acuerdo.
- **Mezcla de tecnologías**: No tienen una dependencia excesiva de la tecnología y pueden optar por la más adecuada para cada situación en particular.
- **Escalabilidad**: Si los sistemas son más pequeños, es más sencillo expandirlos horizontalmente sin preocuparse por su estado, ya que solo es necesario escalar el servicio que lo requiere.
- **Equipos pequeños y centrados**: es posible que cada servicio pueda ser creado por un equipo específico que se enfoque en ese servicio en particular.
- Arrancan más rápido y son más fáciles de desplegar.

4.4 T4 Text Templates

Las plantillas T4 (Text Template Transformation Toolkit) son una funcionalidad de Visual Studio que permite la generación de código basada en plantillas. Las plantillas T4 se utilizan para generar código fuente, documentos de texto o cualquier otro tipo de contenido basado en texto. Se pueden utilizar para crear una amplia gama de archivos, desde código fuente hasta archivos de configuración, y son especialmente útiles para situaciones en las que necesitas generar código repetitivo [17].

Una plantilla T4 consta de dos partes principales: texto directo y bloques de control. El texto directo se escribe en el archivo de salida tal como está, mientras que los bloques de control se utilizan para controlar la lógica de la plantilla. Podemos observar un ejemplo muy sencillo de plantilla T4 en la *figura 8*:

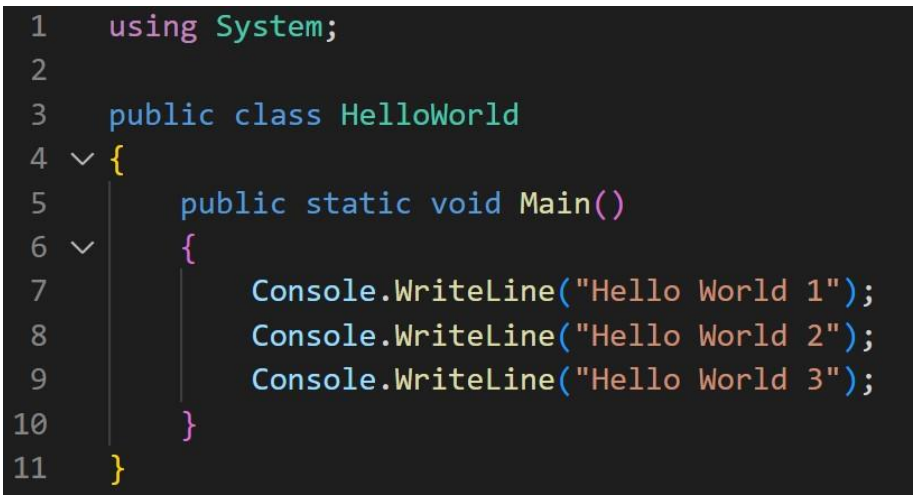
```
<#@ template debug="false" hostspecific="false" language="C#" #>
<#@ output extension=".cs" #>
using System;

public class HelloWorld
{
    public static void Main()
    {
<#
        for (int i = 1; i <= 3; i++)
        {
#>
        Console.WriteLine("Hello World <#=i#>");
<#
        }
#>
    }
}
```

Figura 8. Ejemplo Plantilla T4

En este ejemplo, primero definimos algunas directivas de plantilla (líneas que comienzan con `<#@`). Estas son necesarias para indicar el lenguaje de programación que se utilizará en la plantilla (en este caso, C#), así como para importar cualquier espacio de nombres necesario. Luego, especificamos que la salida de esta plantilla será un archivo `.cs`.

Finalmente, tenemos dos líneas de texto directo que definen la clase y el main() ,y dos bloques de control que se utilizan para el for que hace un bucle de 3 iteraciones ejecutando una línea que es una combinación de texto directo y bloque de control, en la que se generara una

línea con la instrucción Console.Writeline que imprime “Hello World” seguido del número de iteración actual. Los bloques de control se delimitan con `<#= #>`. El código generado se puede apreciar en la *figura 9*:



```
using System;

public class HelloWorld
{
    public static void Main()
    {
        Console.WriteLine("Hello World 1");
        Console.WriteLine("Hello World 2");
        Console.WriteLine("Hello World 3");
    }
}
```

Figura 9. Resultado de la generación de la plantilla T4

Más adelante, en la sección de programación de este proyecto, detallaremos específicamente cómo hemos implementado esta plantilla T4 en nuestra solución.

CAPÍTULO QUINTO

5. Desarrollo de la solución

5.1 Metodología

Para realizar el proyecto, se utilizará una metodología ágil. El trabajo se definirá y abordará en incrementos. Estos incrementos o tareas serán organizadas en Sprints de 1 a 2 semanas de duración, lo que permitirá avanzar en el proyecto de manera iterativa e incremental. Además, se llevarán a cabo reuniones entre Sprints con el fin de hacer un seguimiento del progreso, compartir los avances y reestimar el alcance de dichos Sprints en caso necesario. Esto permitirá adaptar el proyecto a medida que avanza, lo que aumentará la probabilidad de éxito del mismo y permitirá realizar ajustes en tiempo real [18].

Para la gestión del proyecto, se ha elegido la herramienta OneNote, la cual permitirá registrar las tareas en un backlog de manera organizada y eficiente. Asimismo, se utilizará Azure DevOps para ir creando ramas para cada Sprint y en la descripción de estas se añadirán las tareas correspondientes. Además, para el registro del tiempo se empleará la aplicación SAPI[4], la cual facilitará el seguimiento y control de las horas trabajadas.

Durante nuestro trabajo, hemos implementado diversas prácticas de la metodología ágil, como Scrum y Kanban. Estos enfoques ágiles nos ofrecen varias ventajas, como la adaptabilidad a los cambios rápidos, la mejora continua a través de la iteración y la retroalimentación, y la capacidad de entregar productos o soluciones de alta calidad a un ritmo más rápido y sostenible. A continuación, detallaremos estas prácticas y cómo las hemos aplicado en nuestro proyecto:

- **Scrum:** Utilizamos Scrum como marco de trabajo principal, organizando nuestro trabajo en sprints de 1 a 2 semanas. Durante cada sprint, el equipo se enfocaba en un conjunto específico de tareas para completar, lo que nos permitió entregar incrementos de producto de manera regular y adaptable.
- **Tablero kanban**: Para gestionar nuestro flujo de trabajo de manera visual y transparente, creamos un tablero kanban en Trello. Este tablero constaba de varias columnas que representaban las distintas etapas de nuestro proceso, desde

[4] SAPI: es una herramienta de gestión de proyectos que permite asignar y organizar tareas en un backlog, y además ofrece una función de registro de tiempo para monitorear cuánto se tarda en completar cada tarea.

"pendiente" hasta "en progreso" y "terminado". Este método nos permitió ver rápidamente el estado de cada tarea y gestionar nuestra carga de trabajo de manera eficaz.

- **Técnicas de desarrollo**: En cuanto a las técnicas de desarrollo, hicimos de **Test-Driven Development (TDD)**. En algunos sprints, seguimos el enfoque TDD, escribiendo primero las pruebas antes de implementar la funcionalidad. Esta práctica nos ayudó a definir claramente los requisitos de la función antes de su implementación y asegurar que el código resultante cumpliera con los criterios de aceptación. Esta flexibilidad en nuestras prácticas de desarrollo nos permitió adaptarnos a las necesidades específicas de cada sprint y maximizar nuestra eficiencia y productividad.
- **Pruebas de aceptación y de regresión**: Las pruebas de aceptación fueron esenciales para confirmar que el software cumple con los criterios de aceptación definidos por el cliente o los usuarios. Estas pruebas se realizaron al final de cada sprint para asegurarnos de que cada nueva funcionalidad cumplía con las especificaciones y funcionaba como se esperaba.

  Las pruebas de regresión son las mismas que las de aceptación, solo que estas se realizaron para asegurar que las nuevas características o cambios en el software no hayan afectado negativamente a las funcionalidades existentes. Estas pruebas son fundamentales para mantener el correcto funcionamiento del software a medida que evoluciona y se desarrolla.
- **Entrega incremental**: Adoptamos la práctica de entrega incremental, que es una característica fundamental de las metodologías ágiles. En lugar de esperar a que todas las funcionalidades estuvieran completas para entregar el software, entregamos pequeños incrementos de funcionalidad en cada sprint. Esta práctica nos permitió, como desarrolladores, verificar gradualmente el funcionamiento de cada nueva función a medida que estuviera disponible. Al hacerlo, pudimos evaluar y validar progresivamente nuestro proyecto en función de la retroalimentación real, lo que nos dio la oportunidad de adaptarlo y mejorarlo de manera continua. Al mismo tiempo es una ventaja para los clientes, ya que los clientes o usuarios finales también pueden utilizar y aportar feedback, lo que puede ser enriquecedor para el desarrollo del proyecto.

5.2 Planteamiento general

Antes de abordar detalladamente las especificidades técnicas del proceso de migración de la seguridad, procederemos a examinar el estado actual del ERP, analizando qué datos maneja y cómo estos son persistidos en el sistema.

Esto nos ayudará a entender la especificación de requisitos que se realizará posteriormente, asi como las decisiones tomadas a la hora de desarrollar el proceso de migración de seguridad.

5.2.1 Estado del Producto actual

La seguridad en el ERP actual como hemos explicado en el apartado 3.1.1 implementa un modelo de seguridad basado en grupos de permisos y permisos hijos, así como asociaciones de usuario-grupo, sin olvidarnos de que existen usuarios con permisos diferentes a los de su grupo, a estos llamamos usuarios con permisos especiales. Este modelo de seguridad esta persistido de la siguiente forma en tablas SQL:

Tabla Elementos de Seguridad: Estos son los permisos que se asignan a todos los grupos por defecto. Cuando se crea un grupo, estos permisos se definen por defecto y pueden modificarse posteriormente según las necesidades de cada grupo.

En la *tabla 2* se pueden ver todas las columnas de la tabla y lo que significa cada una:

Elementos de Seguridad		
Columna	**Significado**	**Tipo de Dato**
IDElemento	ID del Elemento de seguridad	INT
Codigo	Código del Elemento de Seguridad	STRING
AplicarVer	Si es igual a 1 tiene permisos para ver, si es igual a 0 no.	INT
AplicarImprimir	Si es igual a 1 tiene permisos para Imprimir, si es igual a 0 no.	INT
AplicarAnadir	Si es igual a 1 tiene permisos para Añadir, si es igual a 0 no.	INT
AplicarBorrar	Si es igual a 1 tiene permisos para Borrar, si es igual a 0 no.	INT

Tabla 2. Tabla SQL Elementos de Seguridad

Tabla Permisos: Son todos los permisos especiales modificados para cada Grupo. Cuando un Grupo es creado y modificamos uno de los Elementos de seguridad, este grupo pasa a tener una entrada en la tabla de permisos.

En la *tabla 3* se pueden ver todas las columnas de la tabla y lo que significa cada una:

Permisos		
Columna	**Significado**	**Tipo de Dato**
IDPertenece	ID del Grupo al que pertenecen los permisos	INT
IdElemento	ID del Elemento de Seguridad al que el permiso hace referencia	INT
Codigo	Código del Permiso	STRING
Ver	Si es igual a 1 tiene permisos para Ver, si es igual a 0 no.	INT
Anadir	Si es igual a 1 tiene permisos para Añadir, si es igual a 0 no.	INT
Modificar	Si es igual a 1 tiene permisos para Modificar, si es igual a 0 no.	INT
Borrar	Si es igual a 1 tiene permisos para Borrar, si es igual a 0 no.	INT
Imprimir	Si es igual a 1 tiene permisos para Imprimir, si es igual a 0 no.	INT
Pertenece	Si es igual a 1 es un permiso de grupo, si es igual a 2 es un permiso de usuario	INT

Tabla 3. Tabla SQL Permisos

Tabla Grupos: Son todos los grupos creados en el ERP.

En la *tabla 4* se pueden ver todas las columnas de la tabla y lo que significa cada una:

Grupos		
Columna	**Significado**	**Tipo de Dato**
IDGrupo	ID del Grupo	INT
Descripcion	Nombre del Grupo	STRING

Tabla 4. Tabla SQL Grupos

Tabla Usuarios: Son todos los Usuarios que pertenecen a alguno de los grupos.

En la *tabla 5* se pueden ver todas las columnas de la tabla y lo que significa cada una:

Usuarios		
Columna	**Significado**	**Tipo de Dato**
IDUsuario	ID del Usuario	INT
IDGrupo	ID del Grupo al que pertenece	INT

	el Usuario	
NombreUsuario	Nombre del Usuario	STRING

Tabla 5. Tabla SQL Usuarios

Relaciones entre las tablas:

Las relaciones entre tablas son fundamentales para obtener información y emparejar datos. En las bases de datos relacionales, estas conexiones nos permiten cruzar información de diferentes tablas a través de claves comunes. Cuando se realiza el "matching" de datos, estamos buscando coincidencias o correspondencias entre registros en diferentes tablas. En la *figura 10* se pueden ver las relaciones entre las tablas mencionadas anteriormente:

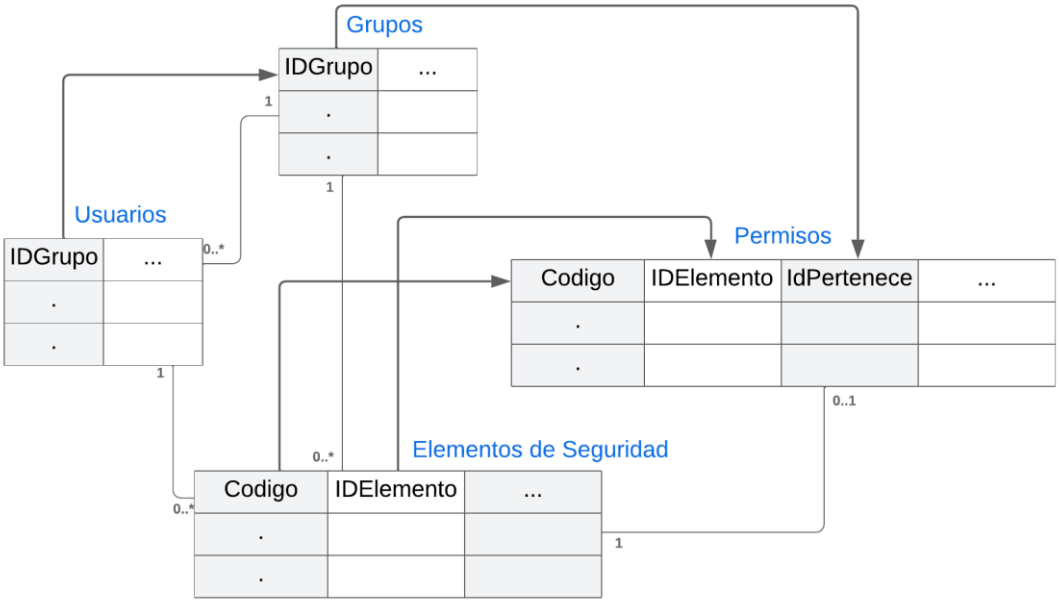


Figura 10. Relaciones entre las tablas SQL

5.3 Especificación de requisitos

5.3.1 Casos de Uso

A continuación, se expondrán las diversas funcionalidades que la migración de seguridad debería ser capaz de realizar, ilustradas mediante Casos de Uso. Cada uno de estos Casos de Uso se identificará con un identificador único, lo que simplificará su referencia, además de tener un nombre y una descripción.

En la *figura 11* vemos el diagrama de los casos de uso. Los casos de uso que están coloreados de gris no forman parte directamente del proceso de migración de la seguridad, sino que pertenecen al microservicio de migración, sobre el cual está implementado dicho proceso:

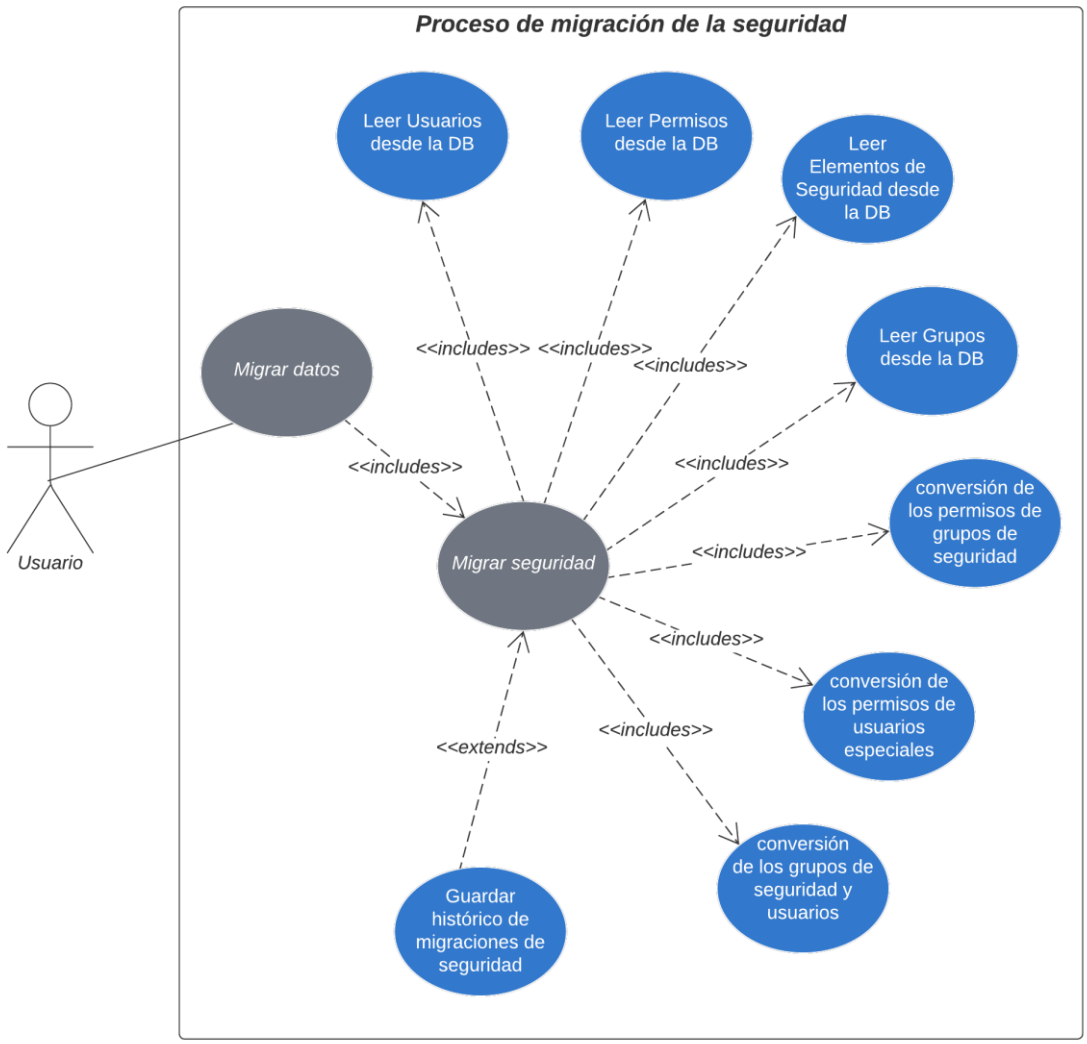


Figura 11. Diagrama de Casos de Uso

Identificador	CU01
Figura	11
Nombre	Lectura de los usuarios desde la DB
Actor	Usuario de la aplicación
Descripción	El proceso de migración de seguridad implicará acceder a la base de datos donde se encuentran persistidos los datos de los clientes. Es importante tener en cuenta que este acceso se requerirá para leer la información de los usuarios. Esto es crucial ya que estos datos son necesarios para el CU06.

Identificador	CU02
Figura	11
Nombre	Lectura de los Permisos desde la DB
Actor	Usuario de la aplicación
Descripción	El proceso de migración de seguridad implicará acceder a la base de datos donde se encuentran persistidos los datos de los clientes. Es importante tener en cuenta que este acceso se requerirá para leer la información de los permisos. Esto es crucial ya que estos datos son necesarios para el CU05 y CU06.

Identificador	CU03
Figura	11
Nombre	Lectura de los Elementos de Seguridad desde la DB
Actor	Usuario de la aplicación
Descripción	El proceso de migración de seguridad implicará acceder a la base de datos donde se encuentran persistidos los datos de los clientes. Es importante tener en cuenta que este acceso se requerirá para leer la información de los Elementos de Seguridad. Esto es crucial ya que estos datos son necesarios para el CU05 y CU06.

Identificador	CU04
Figura	11
Nombre	Lectura de los Grupos de Seguridad desde la DB
Actor	Usuario de la aplicación

Descripción	El proceso de migración de seguridad implicará acceder a la base de datos donde se encuentran persistidos los datos de los clientes. Es importante tener en cuenta que este acceso se requerirá para leer la información de los Grupos de Seguridad. Esto es crucial ya que estos datos son necesarios para el CU05 y CU06.

Identificador	CU05
Figura	11
Nombre	conversión de los permisos de grupos de seguridad
Actor	Usuario de la aplicación
Descripción	El proceso de migración de seguridad implicará crear un conjunto de permisos de seguridad para cada grupo de seguridad con los datos obtenidos de grupos de seguridad, permisos y elementos de seguridad. Esto es crucial ya que estos datos son necesarios para el CU07.

Identificador	CU06
Figura	11
Nombre	conversión de los permisos de usuarios especiales
Actor	Usuario de la aplicación
Descripción	El proceso de migración de seguridad implicará crear un conjunto de permisos de seguridad para cada usuario especial con el conjunto de permisos del grupo de seguridad creado en el CU05 y con los datos obtenidos de usuarios. Esto es crucial ya que estos datos son necesarios para el CU07.

Identificador	CU07
Figura	11
Nombre	conversión de los grupos de seguridad y usuarios especiales de RP a RPQ
Actor	Usuario de la aplicación
Descripción	El proceso de migración de seguridad implicará convertir los conjuntos de permisos de usuarios especiales y grupos de seguridad, en Roles en la nueva versión del ERP.

Identificador	CU08
Figura	11
Nombre	Guardar histórico de migraciones de seguridad
Actor	Usuario de la aplicación
Descripción	El proceso de migración de seguridad implicará guardar un registro de los permisos de grupos de seguridad y permisos especiales ya migrados a la nueva versión del ERP.

Identificador	CU09
Figura	11
Nombre	Crear y actualizar roles
Actor	Usuario de la aplicación
Descripción	El proceso de migración de seguridad implicará Crear y actualizar los roles en la nueva versión del ERP.

Identificador	CU10
Figura	11
Nombre	Eliminar Roles migrados pero eliminados
Actor	Usuario de la aplicación
Descripción	El proceso de migración de seguridad implicará eliminar los roles que una vez creados en la nueva versión del ERP han sido eliminados en la versión actual de este.

5.3.2 Requisitos no funcionales

Este proceso de migración de seguridad debe cumplir con ciertas propiedades esenciales, como la eficiencia, confiabilidad y robustez, para proporcionar a los clientes un sistema de alta calidad.

ISO/IEC 25010 es un estándar internacional que proporciona directrices para evaluar la calidad de un sistema de software, focalizándose en los requisitos no funcionales. Se divide en ocho características principales: funcionalidad, eficiencia, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad, cada una de las cuales se desglosa en subcaracterísticas específicas, permitiendo así un análisis detallado y completo de la calidad del software [19].

Usaremos este ISO 25010 para analizar los RNF (Requisitos no funcionales) de nuestro proceso de migración de la seguridad:

Identificador	RNF1
Característica	Eficiencia de Desempeño
Descripción	El proceso de migración de seguridad debe ser capaz de manejar grandes volúmenes de datos, lo que implica la capacidad para gestionar Gigabytes de información de manera eficiente. Y realizar la migración en un tiempo razonable, debe completar el proceso dentro de un periodo establecido.

Identificador	RNF2
Característica	Compatibilidad
Descripción	El proceso de migración de seguridad debe ser capaz de interactuar correctamente con el nuevo y el antiguo ERP. Además, dicho proceso de migración no debe interferir en otros procesos del mismo microservicio o de otro dentro del ERP.

Identificador	RNF3
Característica	Fiabilidad
Descripción	El proceso de migración de seguridad debe ser capaz de manejar y recuperarse de errores durante la migración de seguridad sin pérdida de datos. Debido a esto es relevante el tener actualizado un registro histórico de las migraciones de seguridad.

Identificador	RNF4
Característica	Seguridad
Descripción	El proceso de migración de seguridad debe ser capaz de migrar los datos de manera segura para evitar la pérdida o el compromiso de los datos.

Identificador	RNF5
Característica	Mantenibilidad
Descripción	El proceso de migración de seguridad debe ser capaz de mantener consistencia entre los datos en ambas versiones del ERP. Es por esto que el proceso de migración deberá eliminar los roles migrados pero eliminados en el ERP actual.

5.4 Diseño

La solución se divide en cuatro clases: en primer lugar, la clase orquestadora que coordina las demás; en segundo lugar, la clase que interpreta los elementos de seguridad de SQL, tal y como se describió en la sección 5.2.1; en tercer lugar, la clase dedicada a transformar los permisos; y finalmente, la clase que comprende las funciones relacionadas a la generación de los elementos de seguridad apropiados en el nuevo sistema ERP, así como a la gestión del registro histórico de migraciones previamente mencionado. Cada una de estas clases desempeña acciones ad hoc, las cuales se desglosarán a continuación para su mejor comprensión:

1. La clase orquestadora **SecurityMigrator** alberga una acción ad hoc que es fundamental *StartSecurityMigration*: es la responsable de invocar secuencialmente las acciones ad hoc del resto de las clases, asegurando así un orden lógico y eficiente en la ejecución de las tareas. En la *figura 12* vemos un diagrama de flujo de esta acción ad hoc:

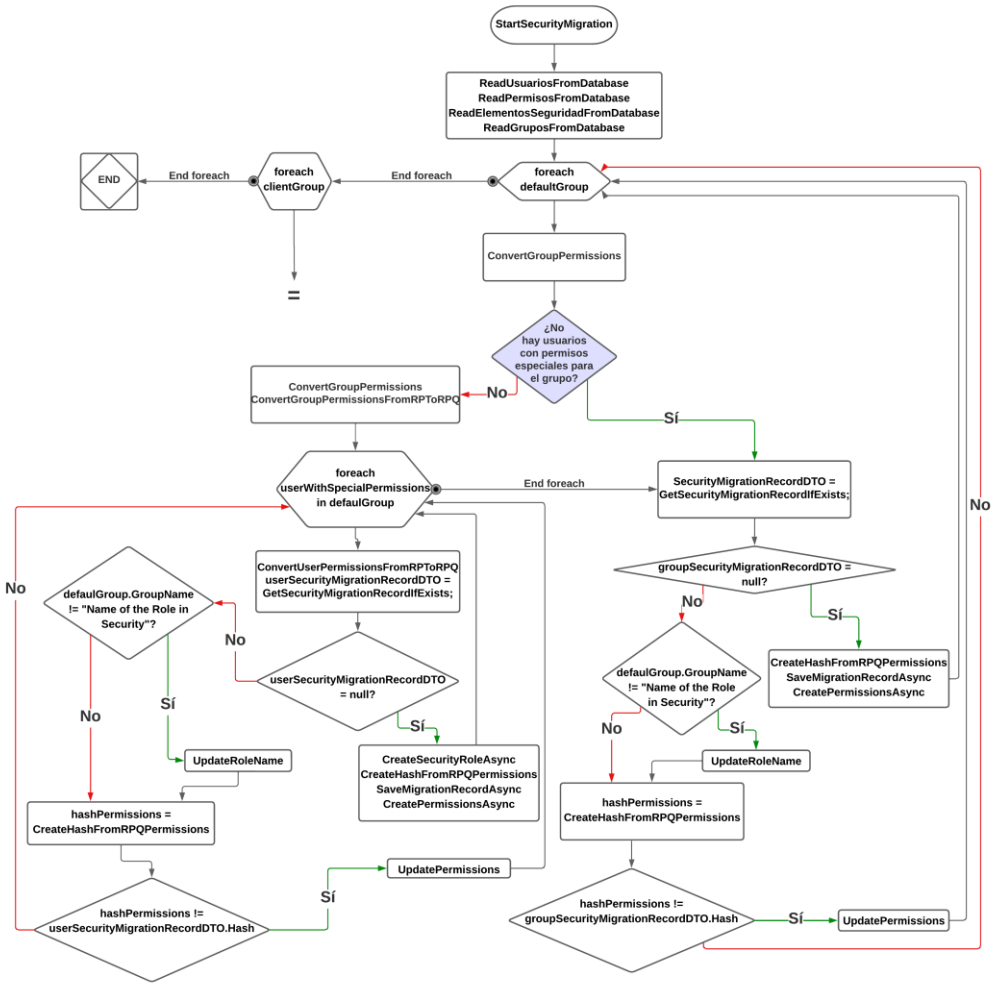


Figura 12. Diagrama Nuevo StartSecurityMigration

2. La clase **ReadSecurityFromDatabase** engloba cuatro acciones Ad Hoc: *ReadUsuariosFromDatabase*, *ReadPermisosFromDatabase*,

ReadElementosSeguridadFromDatabase y *ReadGruposFromDatabase*. Estas acciones extraen información de las tablas Usuarios, Permisos, Elementos Seguridad y Grupos respectivamente, desde la base de datos SQL, a través del DTO DatabaseAccesDTO. Los resultados obtenidos de cada lectura se almacenan en los DTOs correspondientes: UsuarioDTO, PermisosDTO, ElementosSeguridadDTO y GruposDTO. Los campos de cada uno de estos DTOs corresponden a las columnas de interés en cada tabla SQL, tal como se expuso en el apartado 5.2.1. En la *figura 13* vemos el modelo DSL de esta clase:

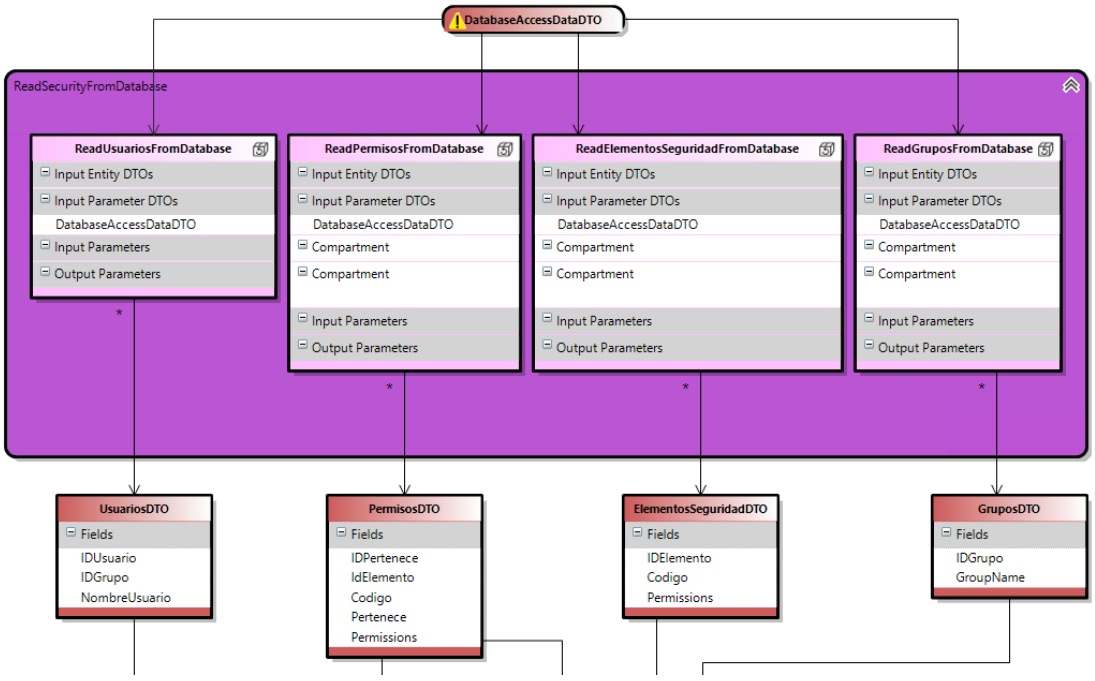


Figura 13. Modelo de la clase ReadSecurityFromDatabase

3. La clase **SecurityConversions** incluye tres acciones ad hoc. Dos de estas acciones, a partir de los DTOs generados por las acciones ad hoc de la clase *ReadSecurityFromDatabase*, producen una lista de *ResiPlusSecurityElementDTO* para cada grupo o usuario. Esta lista contiene todos los elementos de seguridad y los valores de sus respectivos permisos CRUD (Create, Read, Update, Delete) para cada uno de ellos. En la *figura 14* vemos el modelo DSL de esta clase.
    a. La acción *ConvertGroupPermissions* genera una lista de *ResiPlusSecurityElementDTO* para cada grupo de seguridad,
    b. La acción *ConvertUserPermissions* hace lo mismo, pero para cada usuario con permisos especiales.

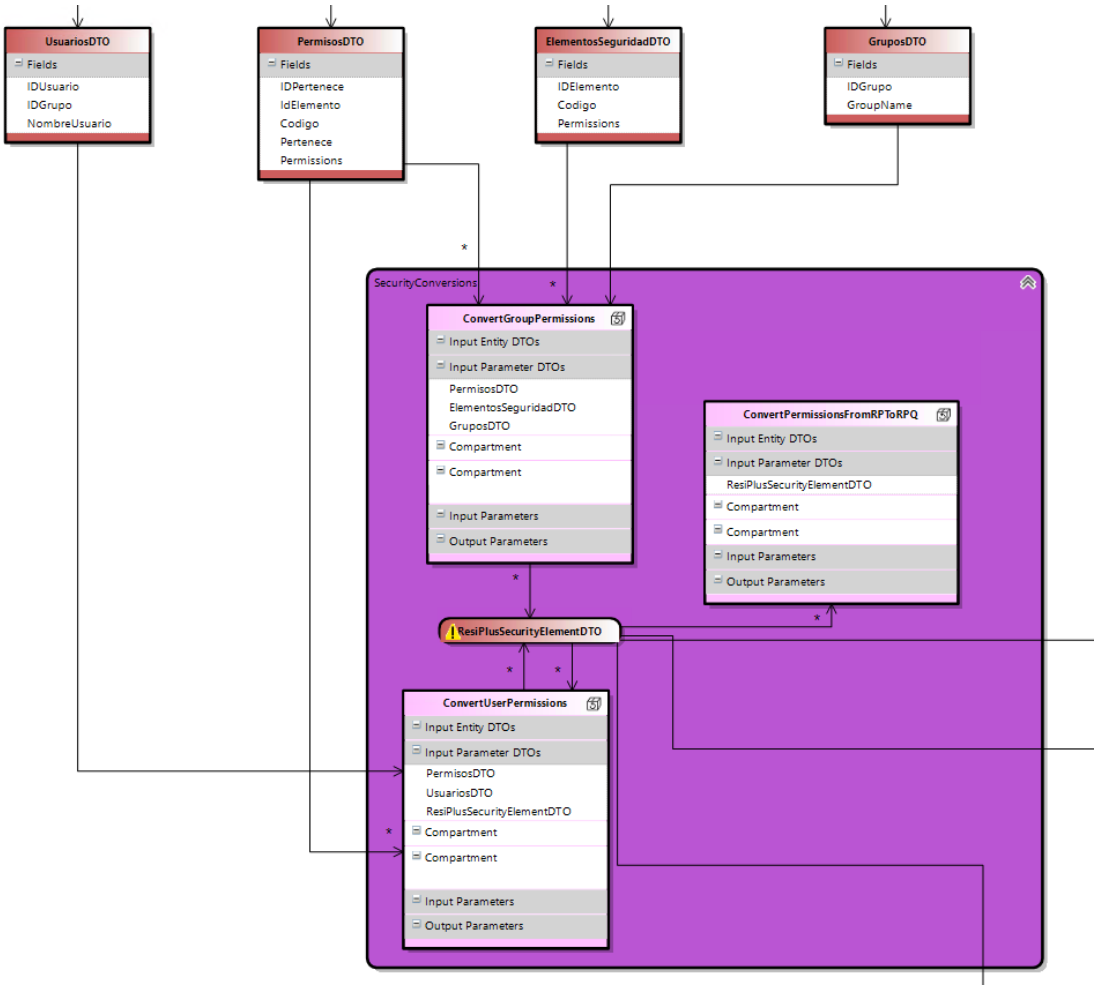


Figura 14. Modelo de la Clase SecurityConversions

Finalmente, abordamos la acción ad hoc denominada *ConvertPermissionFromRPToRPQ*, que representa un método crucial. Este se encarga de transformar los elementos de seguridad actuales, que contienen tanto permisos positivos como negativos, en entidades del dominio que únicamente poseen permisos positivos. Este cambio es esencial debido a que la estructura de permisos para los roles ha sido configurada de esta manera en el nuevo sistema ERP. En la *figura 14* vemos una esquematización de la conversión:

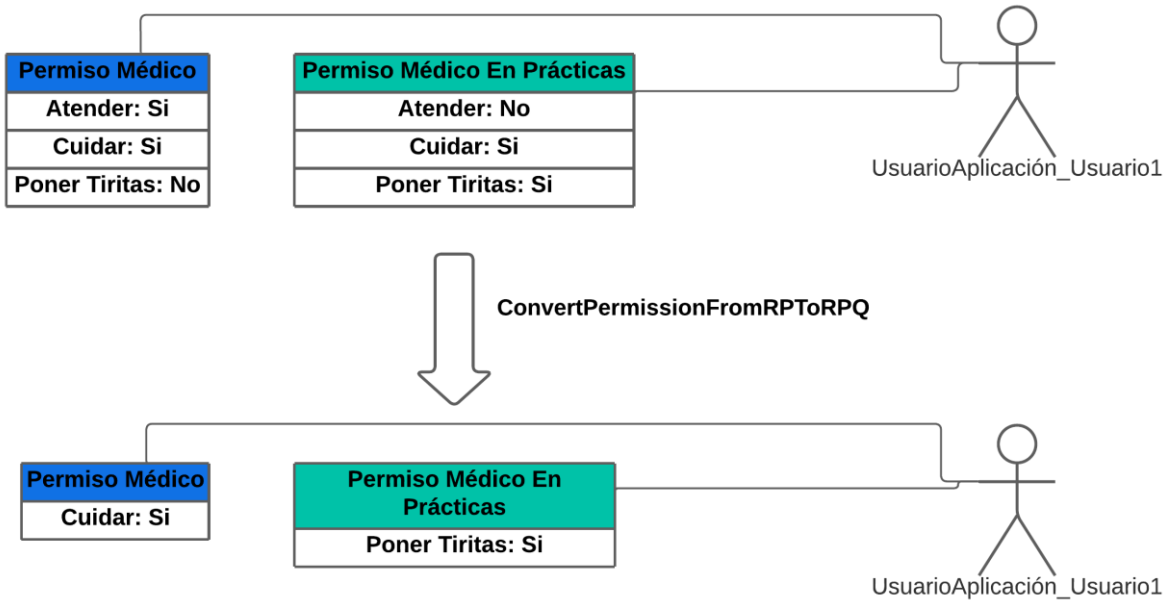


Figura 15. Esquema ConvertPermissionFromRPToRPQ

4. Las acciones ad hoc de la clase **SecurityMigratorActions** pueden agruparse en dos categorías: aquellas relacionadas con el registro histórico de migraciones y las encargadas de la creación de los elementos de seguridad en el nuevo ERP:
    a. En el primer grupo, destaca la acción ad hoc *CreateHashFromRPQPermissions*, que toma como entrada la lista de *ResiPlusSecurityElementDTO* y genera un *PermissionHashDTO*, que incluye un hash para la lista de elementos de seguridad, es relevante señalar que el hash se utiliza para verificar cambios en los elementos de seguridad, tal como se ilustra en el diagrama de la clase orquestadora, esto se debe a que si los permisos experimentan alguna modificación, el hash asociado a estos también cambiará, proporcionando así una vía eficaz para la detección y gestión de las alteraciones en los permisos. Seguidamente, la acción ad hoc *SaveMigrationRecord* se encarga de añadir una entrada al registro de migraciones realizadas, almacenando el hash previamente calculado.
    b. En el segundo grupo, encontramos acciones ad hoc como *CreateSecurityRole* y *UpdateRoleName*, que crean y actualizan un rol respectivamente, así como *CreatePermissions* y *UpdatePermissions*, que establecen y actualizan los permisos de un rol. Finalmente, la acción *RemoveRolesMigratedAndDeleted* elimina los roles que se migraron pero que fueron posteriormente eliminados en el ERP actual, garantizando de esta manera la consistencia entre ambas versiones del ERP, tal como se mencionó anteriormente.

5.5 Programación

A lo largo del desarrollo del proyecto hemos seguido un clean code predeterminado en Visual Studio como encabezados y saltos de línea y así como nombres expresivos para métodos y variables que los revisores han ido comentando. En esta sección, abordaremos las herramientas que se han usado asi como refactorings y los patrones que han sido más relevantes durante el desarrollo de la solución.

Se ha usado una técnica *Model-Driven Development* (MDD) que es un enfoque de desarrollo de software que se centra primero en la creación de modelos abstractos de un sistema, esto se consigo gracias a las DSL que se han mencionado anteriormente. Luego, estos modelos son transformados automáticamente en código, permitiendo a los desarrolladores concentrarse en el diseño y minimizando errores en la implementación.

5.5.1 Desarrollo y refactorings

Plantilla T4

En primer lugar, los analistas proporcionan un archivo de Excel con un mapeo de seguridad. Este mapeo incluye un código para cada elemento de seguridad y a qué entidad, campo, etc. se refiere en la nueva versión del ERP. Esta información es crucial para asignar correctamente los permisos a las entidades correspondientes. En la *figura 16* vemos el Excel:

codigo	Entidad	Campo	Acción
COM-LIS-PLL	CommercialContact		Print
COM-LIS-PRE	CommercialRequest		Print
MDC-VAR-BST	CurrentMedicationStock		Delete
RES-LIS-DEP	BowellMovement		Print

Figura 16. Excel del Mapeo de los elementos de seguridad

Sin embargo, para que esta información pueda ser utilizada en el proceso de migración, necesita estar en formato de código. Para convertir los datos de Excel a código, se utilizan las plantillas de generación de código T4 que se han explicado en el apartado 4.3.

En este caso, las plantillas T4 se utilizan para leer el archivo CSV generado a partir del Excel y a partir de este, generan una clase con un diccionario. En este diccionario, el código del elemento de seguridad se utiliza como clave, y los elementos (entidades, campos, etc.) a los que se refiere se utilizan como valor.

De este modo, la información de mapeo de seguridad proporcionada por los analistas puede ser convertida en un formato que puede ser utilizado directamente en el proceso de migración, facilitando así la asignación de permisos a las entidades correspondientes en el nuevo sistema ERP.

Cuando se agrega más información de mapeo al archivo de Excel, no es necesario añadir manualmente estos nuevos mapeos a la clase del diccionario en el código. En lugar de eso, simplemente se necesita volver a ejecutar la plantilla de generación de código T4.

Este proceso leerá automáticamente el nuevo archivo CSV actualizado (obtenido del Excel) y actualizará la clase del diccionario para incluir los nuevos mapeos de seguridad. Esto significa que la actualización de los permisos de seguridad se convierte en un proceso mucho más eficiente y menos propenso a errores, ya que no requiere una intervención manual para añadir los nuevos mapeos al código. En lugar de eso, todo se maneja automáticamente por la plantilla T4, asegurando que todas las actualizaciones de seguridad se reflejen de manera precisa en el nuevo sistema ERP.

Cambio del DTO de Elementos de seguridad

Figura 17. ResiPlusPermissionsDTO

Anteriormente, se utilizaba el DTO *ResisPlusPermissionsDTO* como podemos ver en la *figura 17* para almacenar los elementos de seguridad de cada grupo de seguridad o de usuarios especiales. Sin embargo, el DTO que se muestra en la imagen no está completo, ya que poseemos más de 2500 elementos de seguridad y el plan original consistía en tener tantos campos en el DTO como elementos de seguridad. Esto generaba dos problemas principales. El primero era el manejo de un DTO excesivamente grande, lo cual resultaba en una eficiencia bastante baja. El segundo problema era que para modificar los permisos de cada elemento de seguridad, debíamos recurrir a la reflexión de C#, lo que podía tener un impacto negativo en la eficiencia de ejecución.

La reflexión en C# es un mecanismo que permite a los programas inspeccionar y manipular el código en tiempo de ejecución. Es una característica de la biblioteca de clases de .NET Framework y se usa en el espacio de nombres System.Reflection [20].

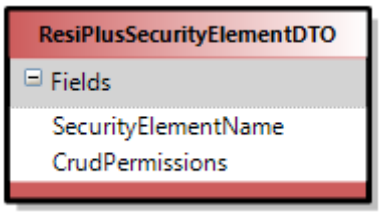


Figura 18.ResiPlusSecurityElementDTO

Considerando los problemas mencionados, se optó por reemplazar *ResisPlusPermissionsDTO* por una lista de *ResiPlusSecurityElementDTO* que podemos ver en la *figura 18*. Cada DTO en esta lista hace referencia a cada elemento de seguridad, y los campos serían el nombre (código del elemento de seguridad) y sus permisos CRUD. De esta manera, estamos gestionando una lista de DTOs mucho más simples. Evitamos la necesidad de la reflexión, ya que ahora simplemente podemos crear un ResiPlusSecurityElementDTO a nuestra preferencia y añadirlo a la lista.

Cambio de la implementación de la Acción Ad Hoc Orquestadora

Se ha refactorizado la implementación de la acción ad hoc orquestadora *StartSecurityMigration* ya que antes la acción orquestadora gestionaba a los grupos por defecto, a los grupos creados por el cliente y a los usuarios con permisos especiales por separado en 3 for each diferentes. En la *figura 19* se puede ver cómo era la implementación anterior:

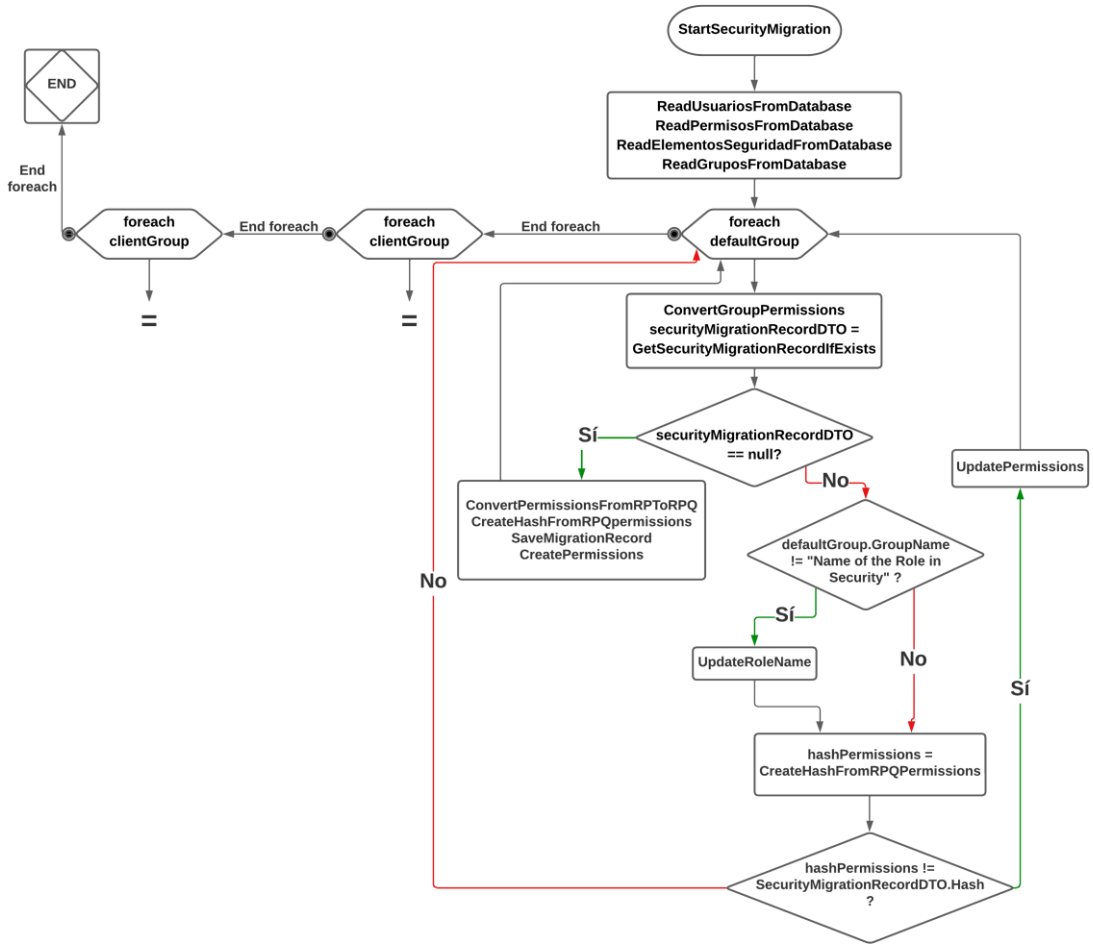


Figura 19. Diagrama Antiguo StartSecurityMigration

Pero para añadir el *ConvertPermissionFromRPToRPQ* se tiene que gestionar a los usuarios con permisos especiales dentro de los foreach de los grupos por defecto y grupos creados por el cliente, ya que para convertir los permisos de los grupos y usuarios en los respectivos roles en la nueva versión del ERP se ha de poder comparar unos con otros, grupos con usuarios.

Además, cabe destacar que al principio se hacía uso de las listas de *ResiPlusSecurityElementDTO* para gestionar los permisos de un grupo o usuario siempre, pero en algunos de esos casos se ha pasado a usar diccionarios que contienen los campos del *ResiPlusSecurityElementDTO* respectivamente como clave y valor, ya que usar diccionarios en ciertos casos reduce mucho el coste.

Por último, hay que comentar que el método *ConvertPermissionFromRPToRPQ* se ha dividido en 2 diferentes para usuarios y grupos, *ConvertUserPermissionFromRPToRPQ* y *ConvertGroupPermissionFromRPToRPQ* respectivamente.

5.5.2 Patrones de diseño

En el desarrollo de este proyecto, se han aplicado varios patrones de diseño que han resultado esenciales para su implementación de forma eficaz. A continuación, comentamos los más relevantes: [21]

Patron creacionales (Patrón Builder)

Este patrón se ha utilizado en la creación gradual de la lista de *ResiPlusSecurityElementDTO* para los usuarios con permisos especiales o los grupos de seguridad.

Este proceso de construcción se ha llevado a cabo de la siguiente manera:

1. Inicialmente, se asignan a cada grupo los permisos por defecto correspondientes a todos los grupos de la tabla SQL de elementos de seguridad.
2. En un segundo paso, se asignan a cada grupo sus permisos específicos de grupo, los cuales se obtienen de la tabla SQL de permisos.
    a. En un tercer paso, se asignan a cada usuario sus permisos específicos de usuario, los cuales se obtienen de la tabla SQL de permisos (este paso es solo para los usuarios con permisos especiales).
3. Finalmente, estos permisos son transformados mediante el método *ConvertPermissionsFromRPToRPQ*. Esta transformación asegura que los permisos cumplen con la estructura de seguridad del nuevo ERP. El objetivo de este paso es garantizar que los grupos o usuarios solo tengan permisos positivos.

El patrón Builder ha permitido un proceso de construcción más limpio y organizado de los objetos ResiPlusSecurityElementDTO, facilitando así su manipulación y mantenimiento. Ha sido esencial para gestionar la complejidad asociada con la construcción de estos objetos, especialmente dada la necesidad de interactuar con múltiples tablas SQL y realizar transformaciones de datos. En la *figura 20* vemos una esquematización del patrón builder:

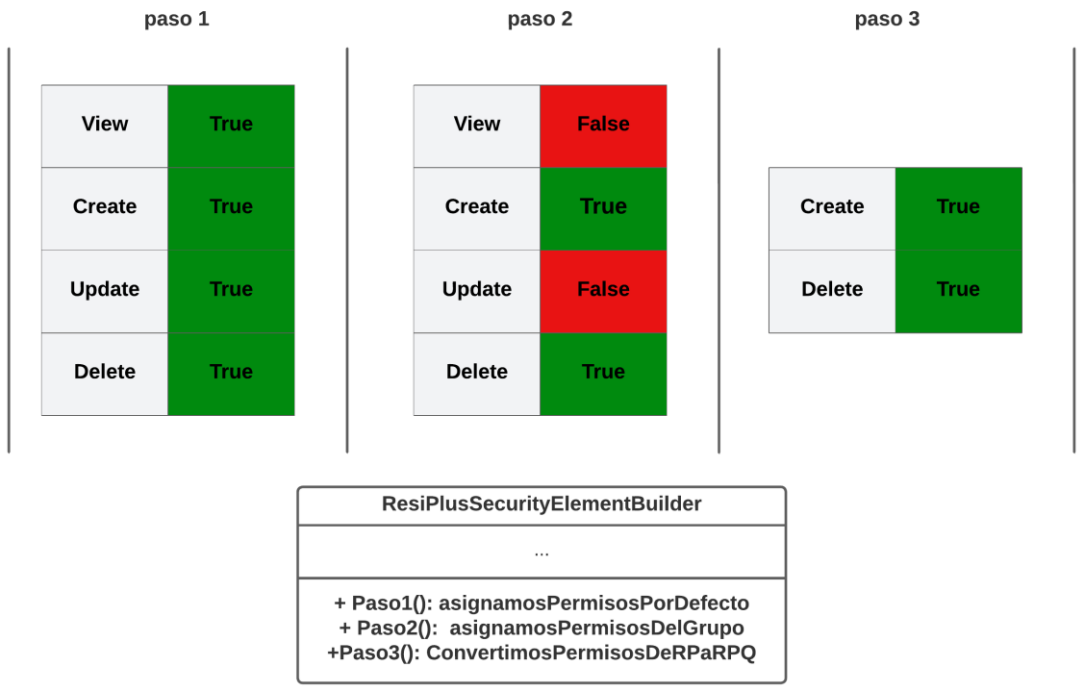


Figura 20. Patrón Builder

Patron estructurales (Patrón Adapter)

También hemos implementado patrones de diseño estructurales, en particular, el patrón Adapter. Este patrón es de gran utilidad cuando necesitamos que dos interfaces diferentes sean compatibles, permitiendo que las clases que de otro modo no podrían trabajar juntas debido a interfaces incompatibles, puedan hacerlo.

En el contexto de este proyecto, el patrón Adapter ha jugado un papel crucial durante el proceso de migración de seguridad entre el antiguo ERP y el nuevo ERP. En esencia, este proceso de migración ha funcionado como una especie de 'puzle' intermedio, proporcionando la compatibilidad necesaria entre los dos sistemas.

El Adapter ha permitido transformar la estructura y la representación de los datos de seguridad del antiguo ERP a la estructura requerida por el nuevo ERP, garantizando así que los datos de seguridad se conserven y funcionen correctamente en el nuevo entorno.

Este patrón de diseño ha sido fundamental para facilitar este proceso de transición y garantizar una migración de datos de seguridad eficiente y efectiva entre los dos sistemas ERP. En la *figura 21* vemos una esquematización del patrón Adapter:

Figura 21. Patrón Adapter

Patron de comportamiento (Patrón Template)

Para desarrollar nuestro histórico de migraciones que se explicó antes, fue necesario heredar de la clase MigrationRecord(diseñada para guardar entradas de los datos migrados), resultando en la creación de nuestra propia clase llamada SecurityMigrationRecord. Esta nueva clase, al ser una extensión de MigrationRecord, referencia a un RTableRPQEntity como se ve en la *figura 22*. Esta última es la tabla donde se almacenan todas las instancias de MigrationRecord. En nuestro caso, durante el proceso de migración de la seguridad, es la tabla donde se almacenan todas las instancias de SecurityMigrationRecord. Estas representaciones de registros de migraciones de seguridad se almacenan en la misma tabla RTableRPQEntity, permitiendo un seguimiento eficiente y organizado de todos los elementos migrados de la seguridad en el sistema.

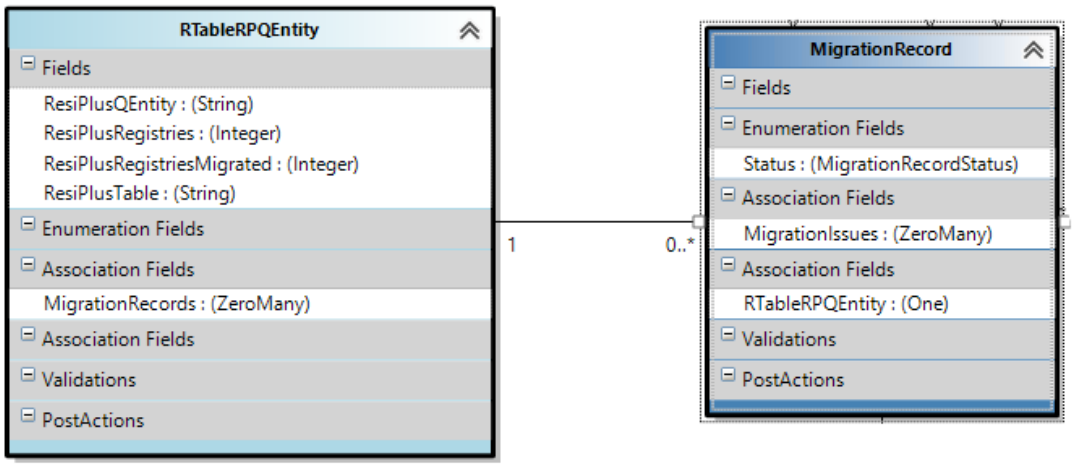


Figura 22. Migration Record

Además de la migración de la seguridad, existen otros procesos que se encargan de migrar diferentes tipos de datos, cada uno de ellos tiene su propia versión personalizada de la clase MigrationRecord. En nuestro caso, para crear nuestra versión del MigrationRecord, el SecurityMigrationRecord que se puede ver en la *figura 23*, hemos recurrido al uso del patrón de diseño de plantilla. Este patrón nos ha permitido agregar 4 nuevos campos a la clase: GroupId (INT), Hash (Guid), RoleId (Guid) y RoleType (enum: Group o User). Estos campos adicionales nos facilitan la identificación de los elementos de seguridad que se han migrado, permitiéndonos determinar si un registro en particular hace referencia a un grupo o a un usuario, si ha sido migrado, modificado o eliminado.

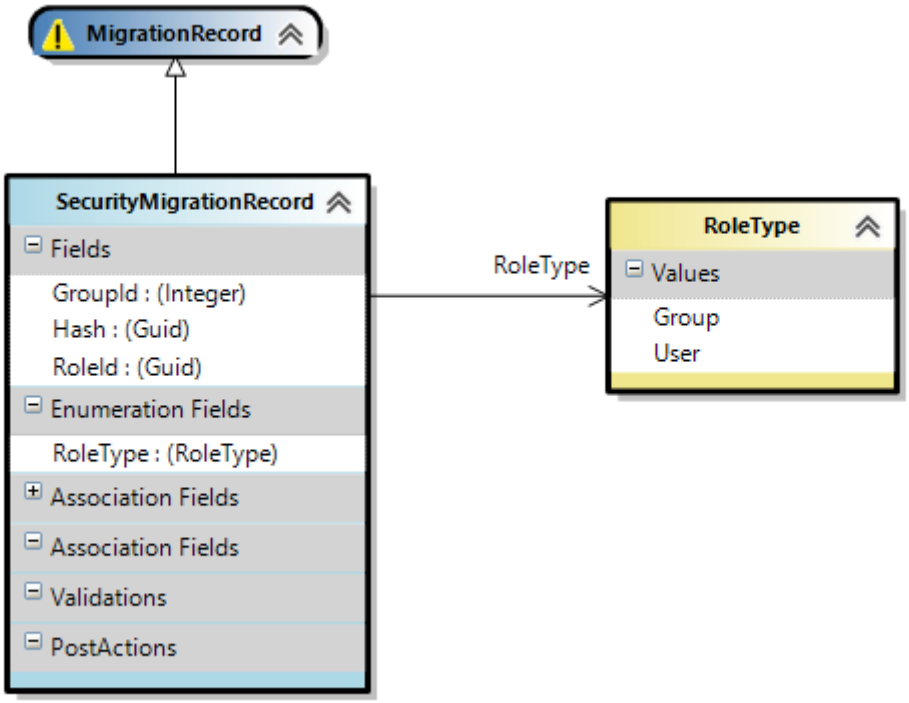


Figura 23. SecurityMigrationRecord

La plantilla sería la clase MigrationRecord. Esta clase proporciona una estructura general para registrar las migraciones, independientemente del tipo específico de datos que se esté migrando. Cada versión específica de MigrationRecord, como SecurityMigrationRecord en nuestro caso, hereda de la clase MigrationRecord y redefine ciertos aspectos de la misma para adaptarse a sus necesidades específicas.

La parte adicional, en nuestro caso, serían los 4 nuevos campos añadidos a la clase SecurityMigrationRecord: GroupId, Hash, RoleId, y RoleType. Estos campos representan aspectos específicos de la migración de seguridad que no están presentes en la clase de plantilla

MigrationRecord. Estos campos adicionales nos permiten adaptar el esqueleto general de MigrationRecord para gestionar de manera eficiente y eficaz las migraciones de seguridad.

5.6 Pruebas

Las pruebas son fundamentales para garantizar la calidad y funcionalidad del código. Cuando se desea integrar una Pull Request, es necesario pasar el BOT. Para que este proceso se complete sin problemas, todos los test deben ser exitosos, confirmando así el correcto funcionamiento del sistema con los nuevos cambios aplicados.

5.6.1 Pruebas automatizadas

Las pruebas automatizadas son un proceso de verificación y validación de software para realizar pruebas en un sistema, sin la intervención manual. Estas pruebas incluyen la ejecución de casos de prueba predefinidos, la comparación de los resultados obtenidos con los esperados, y la generación de resultados de pruebas, que pueden ser exitosas o no. Este método ofrece varias ventajas: rapidez y eficiencia (las pruebas se realizan mucho más rápido que las pruebas manuales), precisión (las herramientas de automatización pueden realizar pruebas con un alto grado de precisión), y reutilización de los casos de prueba (los casos de prueba pueden ser reutilizados para futuras pruebas, ahorrando tiempo y esfuerzo).

En el proceso de migración, las pruebas automatizadas han sido utilizadas para comprobar todas las "Ad hoc Actions". La automatización permite verificar que estas acciones se ejecuten correctamente y que el sistema migrado funcione según lo esperado. Además, las pruebas automatizadas proporcionan una verificación constante y consistente en cada etapa del proceso de migración, lo que ayuda a minimizar los riesgos y a garantizar que el sistema sea robusto y confiable.

Estas pruebas automatizadas han sido modeladas con las DSL tools que se mencionaron anteriormente, podemos ver los modelos en la *figura 24*. Con el uso de estas DSL, se han generado pruebas específicas para cada Ad hoc Action, asegurándonos de que se cubren todos los casos posibles. Esta estrategia meticulosa e intensiva es fundamental para garantizar la integridad del proceso de migración de seguridad. Al modelar y generar estas pruebas de forma tan específica y detallada, podemos tener confianza en que el proceso de migración de seguridad

no fallará, ya que cada acción ad hoc ha sido sometida a pruebas exhaustivas para identificar y corregir cualquier posible fallo.

Figura 24. Test de ConvertGroupPermissions

Aprovechamos para explicar varios componentes clave que en la imagen anterior donde se ve el modelo de los test de la Ad Hoc Action ConvertGroupPermissions. En azul, se visualizan los 3 Ad hoc Action Test Case, que representan cada uno de los casos particulares incluidos en la Ad hoc Action Test Case Collection, mostrada en color morado. Esta colección sería la clase test en la que se agrupan los casos. Adicionalmente, se observan los Test Case Input Parameter DTO, marcados en amarillo, que representan las entradas para cada uno de estos casos de prueba. Finalmente, los Test Case Output Parameter DTO, destacados en rojo, son los resultados que se esperan obtener de cada test.

Aunque la mayoría de las pruebas automatizadas no requieren una clase parcial, hay casos específicos en los que es necesario. Un ejemplo de esto es la Ad Hoc Action 'CreateHashFromRPQPermissions' que podemos ver en la *figura 25*, en la que se necesita crear manualmente un parámetro de entrada adicional. Este parámetro adicional se utiliza para validar si dos permisos tienen o no el mismo GUID (Identificador Único Global). En tales escenarios, la creación de una clase parcial proporciona la flexibilidad para añadir funcionalidades adicionales a la clase de prueba existente y facilitar la realización de pruebas más complejas. Así, el proceso de prueba se vuelve más completo y robusto, aumentando la probabilidad de descubrir cualquier posible fallo en el sistema antes de que se complete el proceso de migración de seguridad.

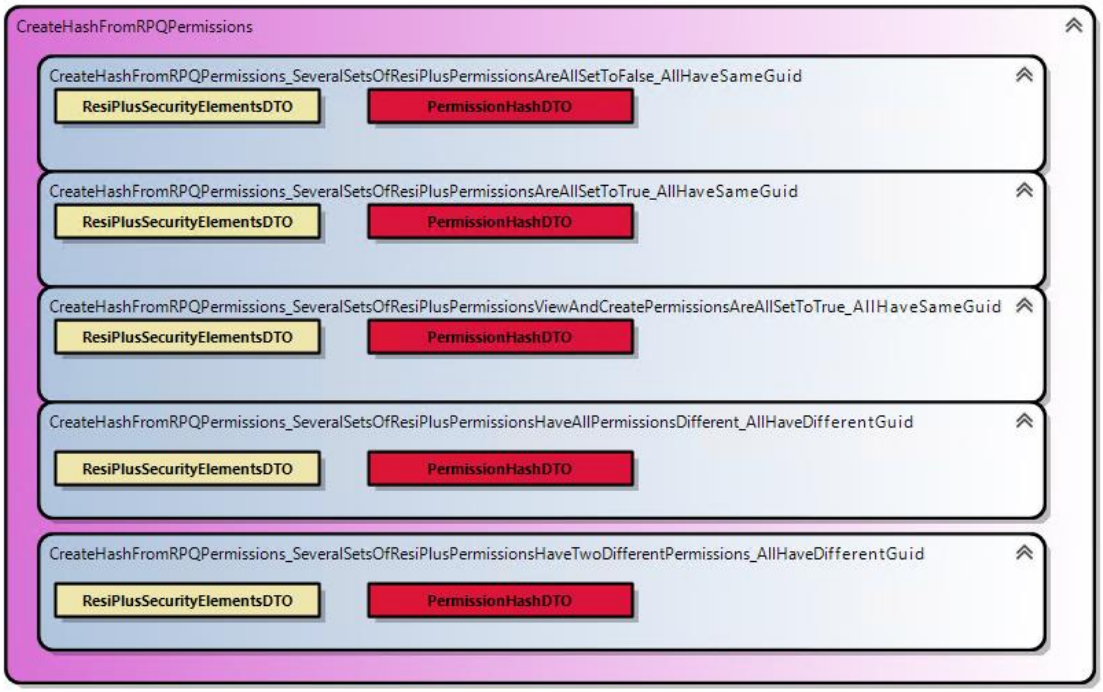


Figura 25. Test de CreateHashFromRPQPermissions

En el siguiente fragmento de código de la figura 26, puedes ver cómo se invoca al método CreateHashFromRPQPermissions para el nuevo parámetro de entrada adicional. Posteriormente, se realiza una verificación para asegurar que este valor es igual al output para el primer parámetro, que en este caso sería result.Hash.

```
1  CalculateHashFromTemplateContent(inputResiPlusSecurityElementDtoCollection).Hash.Should().Be(result.Hash);
2
3              return Task.CompletedTask;
4
5              PermissionHashDTO CalculateHashFromTemplateContent(List<ResiPlusSecurityElementDTO> inputResiPlusSecurityElementDtoCollection)
6              {
7                  ISecurityMigratorActions securityMigratorActions = assertServiceProvider.GetService<ISecurityMigratorActions>();
8
9                  return securityMigratorActions.CreateHashFromRPQPermissions(inputResiPlusSecurityElementDtoCollection);
10             }
```

Figura 26. Código de la clase parcial del Test CreateHashFromRPQPermissions

Test con Mock

Los Mock son objetos que simulan el comportamiento de objetos reales, usados principalmente en pruebas unitarias para aislar y reproducir comportamientos específicos [22]. Permiten simular interacciones con dependencias externas, permitiendo así un control preciso sobre el flujo de datos. En el test de la *figura 27* "Create Permission", se usa para simular respuestas de servicios en concreto el proceso de creación de permisos, y así asegurar que el código se comporta como se espera, creando dichos permisos.

Figura 27. Modelo del test CreatePermissions

Esta línea de código de la *figura 28* está creando un objeto Mock de un tipo *ICoktailSecurityAuthorizationInfraAuthenticationManager*. Después de crear la objeto, se obtiene la instancia del objeto simulado con la propiedad .Object.

```
private static void PermissionsAreCreatedAdditionalRegistrations(IServiceCollection services)
{
    services.AddSingleton(new Mock<ICocktailSecurityAuthorizationInfraAuthenticationManager>().Object);
}
```

Figura 28. Creación del Mock para el Test CreatePermissions

En el siguiente código de la *figura 29*, se comprueba que el método *UpdateEntityPermissionsAsync* es llamado con un *EntityCrudPermissionsDTO* como argumento y que los campos de este DTO son los que se esperan (los previamente definidos en la lista *expectedEntities*)

```
IEnumerable<EntityCrudPermissionsDTO> expectedEntities = new List<EntityCrudPermissionsDTO>
{
    new EntityCrudPermissionsDTO()
    {
        EntityName = "CurrentMedicationStock",
        ViewPermission = true,
        CreatePermission = true,
        UpdatePermission = true,
        DeletePermission = true,
    },
    new EntityCrudPermissionsDTO()
    {
        EntityName = "Institution",
        ViewPermission = true,
        CreatePermission = false,
        UpdatePermission = true,
        DeletePermission = false,
    },
    new EntityCrudPermissionsDTO()
    {
        EntityName = "ProcedurePriority",
        ViewPermission = false,
        CreatePermission = false,
        UpdatePermission = false,
        DeletePermission = false,
    },
};

foreach (EntityCrudPermissionsDTO expectedEntity in expectedEntities)
{
    Mock.Get(assertServiceProvider.GetService<ICocktailSecurityAuthorizationInfraAuthenticationManager>())
        .Verify(
            m => m.UpdateEntityPermissionsAsync(
                It.Is<EntityCrudPermissionsDTO>(en =>
                    en.EntityName == expectedEntity.EntityName
                        && en.ViewPermission == expectedEntity.ViewPermission
                        && en.CreatePermission == expectedEntity.CreatePermission
                        && en.UpdatePermission == expectedEntity.UpdatePermission
                        && en.DeletePermission == expectedEntity.DeletePermission)),
            Times.Once);
}
```

Figura 29. Uso del Mock en el Test CreatePermissions

5.6.2 Pruebas sobre datos reales

Una vez que se ha terminado el desarrollo del proyecto, se han llevado a cabo varias pruebas sobre una base de datos de prueba que la empresa tiene a su disposición para este tipo de pruebas. El propósito de estas pruebas es verificar el correcto funcionamiento del sistema antes de llevarlo a un entorno de producción.

El enfoque de prueba en particular que hemos seguido en esta instancia implica verificar la migración de la seguridad. Los microservicios de seguridad (Authorization) y de migración (Migrator) son partes fundamentales de este proceso.

A continuación se describen los pasos que se han seguido en detalle:

- **Lanzar el microservicio de seguridad (Authorization)**: Este microservicio actúa como el destino final para los datos migrados. Al lanzarlo primero, nos aseguramos de que esté listo y en funcionamiento para recibir y manejar los datos de seguridad que se migrarán. Este microservicio no solo maneja las credenciales de autenticación y autoriza las solicitudes a la API, sino que también es el repositorio central para los datos de seguridad tras la migración.
- **Lanzar el microservicio Migrator**: Una vez que el microservicio de seguridad está en marcha y funcionando correctamente, se lanza el microservicio Migrator. Este servicio es responsable de gestionar la migración de datos de seguridad de un sistema a otro. El Migrator puede mover, transformar y sincronizar los datos según sea necesario.
- **Atacar la API del microservicio de migración con POSTMAN**: "Atacar" aquí significa hacer solicitudes a la API para probar su funcionalidad. En este caso, POSTMAN[5] se utiliza para iniciar la migración a través de la API del microservicio de migración.

Una vez que se inicia la migración, el sistema comenzará a mover los datos según las reglas definidas en el microservicio Migrator. Este proceso se supervisa para asegurar que se complete sin errores y que todos los datos se transfieran correctamente.

En resumen, estos pasos permiten a los equipos de desarrollo garantizar que el proceso de migración de seguridad funciona como se espera. De esta manera, podemos asegurarnos de que el sistema está preparado y listo para ser desplegado en producción.

---

[5] POSTMAN: Es una plataforma popular que permite a los desarrolladores y testers enviar solicitudes HTTP a una API y recibir las respuestas.

5.6.3 Resultado de la aplicación de las pruebas

los resultados de las pruebas automatizadas, así como las pruebas sobre datos reales, han sido satisfactorios. En las figuras 30 y 31 se pueden ver la lista de grupos y de usuarios respectivamente.

Results | Messages

	IDGrupo	Codigo	Descripcion	Observaciones	RowGUID	EsGrupoFamiliar	EsGrupoEnlace	PerteneceAplicacion	EsGrupoVinculado
1	-999	000	Aplicación	Grupo de ResiPlus	5C486A7F-AA9A-44CB-9C61-E7DB517E8134	0	0	1	NULL
2	1	001	Dirección	Administración del centro	433FBF56-7B38-4E6A-8829-5165F0955EBF	0	0	0	0
3	2	002	Recepción	Grupo encargado de tareas de facturación, provee...	A7323CD3-C586-4C4D-BA29-0D4FF6AD9FFC	0	0	0	0
4	3	003	Médicos	Grupo formado por los médicos de la residencia.	6256E961-8977-4CF3-93C0-553AA56C0F20	0	0	0	0
5	5	005	Enfermeras		87F16D6C-986F-4E01-BC77-DD7947239857	0	0	0	0
6	7	007	Trabajo Social		69C2E194-C734-4F06-8015-FA1A606E23D3	0	0	0	0
7	8	008	Administración		E348C290-CCA6-41A1-80C9-52DEFFB286CC	0	0	0	0
8	9	009	Fisioterapeuta		1526D6C4-0361-4897-B21D-BDA2D6AD5D9C	0	0	0	0
9	10	010	Animador Sociocultural		04774E5B-2FEB-4AEC-B69E-D00AA6814A83	0	0	0	0
10	11	011	Terapia Ocupacional		9076212C-67EF-4E22-A1AE-7FBA4954769E	0	0	0	0
11	12	012	Familiares		86C2F409-1452-43DC-99D2-847235C0020C	1	0	0	NULL
12	13	000	Links		F08DB394-EDFC-4BC1-826B-D50E7DC7E5AD	0	1	0	NULL
13	15	015	Nuevo Grupo 015		FD4129A5-AEA1-4D5D-BED3-D9C7881E7657	0	0	0	0

Figura 30. Tabla SQL Grupos

Results | Messages

	IDUsuario	IDGrupo	Codigo	NombreUsuario	Contrasena	IDPersonal	IDMedico	NombreApellidos
1	-999	-999	000	La Aplicación	E3AFED0047B08059D0FADA10F400C1E5	-999	-999	La Aplicación
2	1	1	001	Administrador	E3AFED0047B08059D0FADA10F400C1E5	-999	-999	Administrador
3	2	12	002	usu01	B489B4014A8381B33B5C091D3DF0C8AA	-999	-999	usu01
4	3	12	003	usu02	B489B4014A8381B33B5C091D3DF0C8AA	-999	-999	usu02
5	4	1	004	Nuevo Usuario 004	$2a$10$vI8aWBnW3fID.....XgZ2gHnBLLh9CdiGCv5...	-999	-999	Nuevo Usuario 004
6	5	3	005	Nuevo Usuario 005	$2a$10$vI8aWBnW3fID.....XgZ2gHnBLLh9CdiGCv5...	-999	-999	Nuevo Usuario 005

Figura 31. Tabla SQL Usuarios

Fijándonos en las figuras anteriores vemos como se migran los datos tras lanzar el microservicio de migración, en la figura 32 se puede ver como se ha migrado el usuario con permisos especiales *Administrador* del grupo *Dirección* y se ha creado el respectivo Rol (Role_Dirección_Administrador) en la seguridad de la nueva versión del ERP.

Results | Messages

	Id	Name	ConcurrencyToken
1	D35D0E66-2A23-4DB9-964B-261A69D7B4E2	Infrastructure Admin	00000000-0000-0000-0000-000000000000
2	91C80AF9-7779-425A-A0DC-42EE840368FC	Role_Dirección_Administrador	792047D4-74AD-4AA5-B3F1-C01BA3FBA14F

Figura 32. Tabla SQL Roles

5.7 Cronología del proyecto

A lo largo del desarrollo del proyecto, hemos dividido y organizado todas las tareas que se han llevado a cabo, identificando en qué Sprint se realizaron. Estas tareas abarcan principalmente la implementación y el testeo de las acciones ad hoc que se han discutido anteriormente, estas forman nuestro backlog. Cada Sprint, como una etapa del proceso de desarrollo, ha sido crucial para avanzar en el proyecto y garantizar la efectividad de las acciones implementadas, así como para proporcionar posibilidades de ajuste y mejora continuos.

SPRINT 0: 01/03/2023-10/03/2023(10 días)

- Modelar la solución con las DSL Tools

SPRINT 1: 10/03/2023-17/03/2023(7 días)

- Implementar *ReadGruposFromDatabase*, *ReadElementosSeguridadFromDatabase* y *ReadPermisosFromDatabase*
- Implementar *ConvertGroupPermissions* y *ConvertUserPermissions*
- Implementar el esqueleto Acción Ad hoc orquestadora

SPRINT 2: 17/03/2023-31/03/2023(14 días)

- Crear el *SecurityMigrationRecord*
- Implementar Test de la clase *SecurityConversion,* para los métodos *ConvertUserPermissions* y *ConvertGroupPermissions*
- Revisar el uso de nombre en español y ver cuales se pueden pasar a inglés.

SPRINT 3: 31/03/2023-14/04/2023(14 días)

- Implementar método para calcular Hash desde resiPlusPermissionsDTO *CreateHashFromRPQPermissions*.
- Implementar Test de la clase *SecurityMigratorActions*, Para el método `CreateHashFromRPQPermissions`.
- Implementar plantilla T4 para la transformación del Excel en la estructura de datos diccionario.

SPRINT 4: 14/04/2023-23/05/2023(39 días)

- Implementar *SaveMigrationRecord.*
- Implementar *CreateSecurityRole*.
- Implementar *UpdateRoleName*

SPRINT 5: 23/05/2023-10/06/2023(17 días)

- Implementar *UpdatePermissions* y *CreatePermissions*
- Implementar *ConvertPermissionsFromRPToRPQ*
- Implementar Test de *SaveMigrationRecord*

SPRINT 6: 10/06/2023-Presente

- Implementar Test de *UpdatePermissions* y *CreatePermissions*
- Implementar Test de *UpdateRoleName* y *CreateSecurityRole*
- Implementar Test de *RemoveRolesMigratedAndDeleted*
- Refactorizar las clases y la clase orquestadora

En la *figura 27* vemos la línea temporal que muestra los sprints de nuestro proyecto, identificando claramente cuándo comienza cada uno y su duración, incluyendo además información sobre la formación y estudio inicial, la elaboración de la memoria y la revisión de esta, facilitando así el seguimiento del progreso del proyecto.

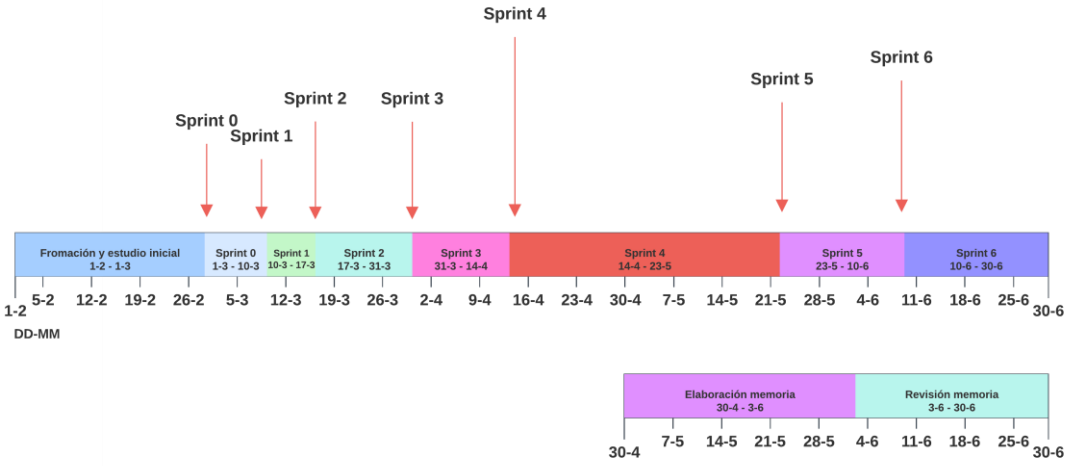


Figura 33. Línea del tiempo Sprints

Para concluir, es relevante indicar que, hasta la fecha, hemos invertido aproximadamente 320 horas dedicadas a este proyecto. Durante este periodo, se han elaborado cerca de 4000 líneas de código, sin contar las clases base generadas por las DSL Tools, reflejando nuestro compromiso y trabajo constante en su desarrollo.

CAPÍTULO SEXTO

6. Conclusiones y trabajo futuro

La realización del proyecto ha cumplido con los objetivos propuestos, logrando un proceso de migración de seguridad de un ERP eficiente y seguro con el uso de tecnologías como ASP.NET Core y DSL Tools.

El proyecto, a día de hoy, se encuentra finalizado. Se ha realizado una demostración en la que ha demostrado su correcto funcionamiento y eficacia, satisfaciendo las necesidades y expectativas del usuario. Esto representa un hito importante en nuestro camino, marcando el fin de esta fase de desarrollo y abriendo la puerta a futuras mejoras y ampliaciones.

El desarrollo de este proyecto ha sido una valiosa experiencia a nivel profesional y personal. Hemos adquirido conocimientos en el uso de múltiples tecnologías y metodologías de desarrollo. A nivel personal, hemos aprendido a trabajar de manera efectiva en equipo, interactuar con colegas de otros departamentos, gestionar retos y cambios, y a mantener un ritmo de trabajo sostenible. Esta experiencia ha contribuido a nuestro crecimiento y desarrollo integral, fortaleciendo nuestras habilidades técnicas y personales.

En cuanto a la formación recibida durante la carrera, las asignaturas de Proyecto de Software (PSW) y Diseño de Software (DDS) han sido de gran utilidad en la ejecución de este proyecto. Además, la experiencia y conocimientos adquiridos durante el intercambio en la Universidad de Leicester, especialmente en las asignaturas de Advanced C++ Programming y Software Measurement and Quality Assurance, han sido también importantes. Estos conocimientos han proporcionado una base sólida que ha facilitado la comprensión y manejo de las tecnologías y técnicas utilizadas en el proyecto.

En cuanto al trabajo futuro, una posible línea de desarrollo sería la implementación de una función multicentro en el proceso de migración, para atender a los clientes de la empresa que tienen varios centros con distintas bases de datos que requieren migración. Esta nueva implementación supone un reto interesante que permitiría expandir la funcionalidad y aplicabilidad de nuestro proyecto. Otra tarea pendiente a tener en cuenta será comprobar el correcto funcionamiento de los requisitos no funcionales RNF5 y RNF6.

Referencias

[1] VMWARE, «What is application security?,» [En línea]. Available: https://www.vmware.com/topics/glossary/content/application-security.html. [Último acceso: 5 Mayo 2023].

[2] CYBERARK, «Principio del Mínimo Privilegio PoLP,» [En línea]. Available: https://www.cyberark.com/es/what-is/least-privilege/. [Último acceso: 17 Junio 2023].

[3] IBM, «What is application migration?,» [En línea]. Available: https://www.ibm.com/topics/application-migration. [Último acceso: 5 Mayo 2023].

[4] CloudSoft, «A Practical Guide to Understanding the 6Rs for Migration to AWS,» [En línea]. Available: https://cloudsoft.io/blog/a-practical-guide-to-understanding-the-6rs-for-migration-to-aws. [Último acceso: 10 Mayo 2023].

[5] CloudSoft, «Migrating to AWS Method 2 – Rehosting, aka Lift-and-Shift,» [En línea]. Available: https://cloudsoft.io/blog/migrating-to-aws-method-2-rehosting-aka-lift-and-shift. [Último acceso: 10 Mayo 2023].

[6] O'REILLY, «Docker Migration Guide,» [En línea]. Available: https://www.oreilly.com/library/view/enterprise-docker/9781491994986/ch04.html. [Último acceso: 14 Mayo 2023].

[7] AWS, «AWS Migration Hub,» [En línea]. Available: https://aws.amazon.com/es/migration-hub/. [Último acceso: 14 Mayo 2023].

[8] AWS, «What Is AWS Migration Hub?,» [En línea]. Available: https://docs.aws.amazon.com/migrationhub/latest/ug/whatishub.html. [Último acceso: 14 Mayo 2023].

[9] Wikipedia, «Amazon EC2,» [En línea]. Available: https://es.wikipedia.org/wiki/Amazon_EC2. [Último acceso: 14 Mayo 2023].

[10] Amazon, «AWS Database Migration Service,» [En línea]. Available: https://aws.amazon.com/es/dms/. [Último acceso: 14 Mayo 2023].

[11] Gogle Cloud, «Migración a Google Cloud,» [En línea]. Available: https://cloud.google.com/products/cloud-migration?hl=es-419. [Último acceso: 14 Mayo 2023].

[12] AZURE, «Azure Migrate,» [En línea]. Available: https://azure.microsoft.com/es-es/products/azure-migrate#overview. [Último acceso: 14 Mayo 2023].

[13] Microsoft, «Incorporación de herramientas de migración,» [En línea]. Available: https://learn.microsoft.com/es-es/azure/migrate/how-to-migrate. [Último

acceso: 20 Mayo 2023].

[14] Microsoft, «Overview of Domain-Specific Language Tools,» [En línea]. Available: https://learn.microsoft.com/en-us/visualstudio/modeling/overview-of-domain-specific-language-tools?view=vs-2022. [Último acceso: 20 Mayo 2023].

[15] Microsoft, «¿Qué es Azure DevOps?,» [En línea]. Available: https://learn.microsoft.com/es-es/azure/devops/user-guide/what-is-azure-devops?bc=%2Fazure%2Fdevops%2Fget-started%2Fbreadcrumb%2Ftoc.json&view=azure-devops. [Último acceso: 20 Mayo 2023].

[16] Microsoft, «Estilo de arquitectura de microservicios,» [En línea]. Available: https://learn.microsoft.com/es-es/azure/architecture/guide/architecture-styles/microservices. [Último acceso: 11 Mayo 2023].

[17] Microsoft, «Code Generation and T4 Text Templates,» [En línea]. Available: https://learn.microsoft.com/en-us/visualstudio/modeling/code-generation-and-t4-text-templates?view=vs-2022. [Último acceso: 20 Mayo 2023].

[18] Red Hat, «¿Qué es la metodología ágil?,» [En línea]. Available: https://www.redhat.com/es/devops/what-is-agile-methodology. [Último acceso: 24 Mayo 2023].

[19] ISO 25000, «ISO/IEC 25010,» [En línea]. Available: https://iso25000.com/index.php/normas-iso-25000/iso-25010. [Último acceso: 24 Mayo 2023].

[20] Microsoft, «Reflexión en .NET,» [En línea]. Available: https://learn.microsoft.com/es-es/dotnet/framework/reflection-and-codedom/reflection. [Último acceso: 28 Mayo 2023].

[21] REFACTORING GURU, «Patrones de diseño,» [En línea]. Available: https://refactoring.guru/es/design-patterns. [Último acceso: 28 Mayo 2023].

[22] KEEPCODING, «¿Qué es mock y fake en pruebas unitarias?,» [En línea]. Available: https://keepcoding.io/blog/que-es-mock-y-fake-en-pruebas-unitarias/#Que_es_mock_y_fake_en_pruebas_unitarias. [Último acceso: 23 Junio 2023].

ANEXO I

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	**Alto**	**Medio**	**Bajo**	**No Procede**
ODS 1. **Fin de la pobreza.**				**x**
ODS 2. **Hambre cero.**				**x**
ODS 3. **Salud y bienestar.**	**x**			
ODS 4. **Educación de calidad.**				**x**
ODS 5. **Igualdad de género.**				**x**
ODS 6. **Agua limpia y saneamiento.**				**x**
ODS 7. **Energía asequible y no contaminante.**				**x**
ODS 8. **Trabajo decente y crecimiento económico.**				**x**
ODS 9. **Industria, innovación e infraestructuras.**			**x**	
ODS 10. **Reducción de las desigualdades**.				**x**
ODS 11. **Ciudades y comunidades sostenibles**.				**x**
ODS 12. **Producción y consumo responsables.**				**x**
ODS 13. **Acción por el clima.**				**x**
ODS 14. **Vida submarina.**				**x**
ODS 15. **Vida de ecosistemas terrestres.**				**x**
ODS 16. **Paz, justicia e instituciones sólidas.**				**x**
ODS 17. **Alianzas para lograr objetivos.**				**x**

La migración de la seguridad de la versión actual a la nueva de la aplicación del ERP para residencias de ancianos ha tenido un impacto significativo en la consecución de los Objetivos de Desarrollo Sostenible (ODS) de Salud y Bienestar e Industria, Innovación e Infraestructura.

Con respecto al ODS de Salud y Bienestar, este proyecto ha mejorado la calidad de vida y el bienestar de los residentes ancianos al ofrecer un sistema de gestión más seguro y eficaz. Las mejoras de seguridad en la aplicación ERP no solo protegen la información personal y médica de los residentes, sino que también aseguran un acceso y una entrega eficientes de servicios de

atención médica y apoyo. El fortalecimiento de la seguridad de los datos también ayuda a prevenir errores médicos o problemas de tratamiento, garantizando un cuidado más adecuado y personalizado para cada residente. Además, la tranquilidad de los residentes y sus familias se ve reforzada al saber que su información está segura.

En cuanto al ODS de Industria, Innovación e Infraestructura, este proyecto ha promovido la innovación en la industria del software ERP al implementar soluciones de seguridad avanzadas en la nueva versión de la aplicación. Este avance tecnológico fomenta un entorno de innovación, lo que ayuda a mejorar la competitividad de la industria y el desarrollo económico. Al mismo tiempo, la mejora de la infraestructura digital de la residencia de ancianos a través de la migración a una aplicación ERP más segura y moderna, también se alinea con este objetivo, ya que implica la modernización de las infraestructuras y el fomento de la innovación.

En resumen, este proyecto de migración de seguridad ha contribuido de manera significativa a los ODS al proporcionar una solución de software ERP más segura y eficiente que mejora el bienestar de los ancianos y promueve la innovación en la industria.