

DEPARTAMENTO DE INFORMÁTICA  
CARRERA: INGENIERÍA INFORMÁTICA  
TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO INFORMÁTICO

**APLICACIÓN DE LA METODOLOGÍA PRACTICAL DATA  
MIGRATION PARA LA MIGRACIÓN DEL SISTEMA  
PESCASPIR PARA LA DETERMINACIÓN DE FORTALE-  
ZAS Y DEBILIDADES**

**APPLICATION OF THE PRACTICAL DATA MIGRATION  
METHODOLOGY FOR THE MIGRATION OF THE PES-  
CASPIR SYSTEM FOR THE DETERMINATION OF STRENGTHS  
AND WEAKNESSES**

**Autora: Rosangela Benitez Castillo**

Tutora: Msc. Yunet Lorenzo Vega

Este documento es Propiedad Patrimonial de la Universidad de Sancti Spíritus «José Martí Pérez», y se encuentra depositado en los fondos del Centro de Recursos para el Aprendizaje y la Investigación «Raúl Ferrer Pérez», subordinado a la Dirección General de Desarrollo 3 de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

**Atribución- No Comercial- Compartir Igual**



Para cualquier información, contacte con:

Centro de Recursos para el Aprendizaje y la Investigación “Raúl Ferrer Pérez”. Comandante Manuel Fajardo s/n, esquina a Cuartel, Olivos 1. Sancti Spíritus. Cuba. CP. 60100

Teléfono: **41-334968**

A mi hermano con mucho amor y cariño le dedico todo mi esfuerzo y trabajo puesto por la realización de esta tesis.

A mis padres, ángeles guardianes, permanentes centinelas de mis pasos en la vida escolar.

A mis abuelos eternos y sabios guías.

A mi tía por ser mi Hada Madrina.

A mi novio por ser el mejor compañero de vida que pudiera desear.

A mi pequeña familia que es lo más valioso que Dios me ha dado.

## AGRADECIMIENTOS

Hoy después de 5 largos años se cumple este sueño compartido por mí y mi familia, haciéndome así capaz de creer en las segundas oportunidades.

Quisiera comenzar agradeciendo a la persona que me dió la vida, a mi mamá, por apoyarme en cada decisión, por creer en mí cuando ni yo misma lo hacía, por ser madre, pero ante todo por ser amiga.

A mi padre por siempre instarme a ser una mejor persona aunque la mayoría de las veces no pensemos igual se que siempre estará ahí para mí.

A mis abuelos del alma, a mi abuelo por todos sus consejos y su ayuda incondicional y a mi abuela por ser esa segunda madre que siempre está ahí para mí, que siempre vive pendiente de todos mis logros y fracasos, sin ellos yo no sería la persona que soy hoy.

A mi tía-mamá que a pesar de la distancia ha estado pendiente de mí todos los días, dándome consejos, ánimos cada vez que sentía que no podía con algo, creyendo en mí y tratándome como su hija.

A mi Rami hermoso que más que un primo es un hermano, gracias por ayudarme en todos los proyectos que he tenido.

A mi novio quien ha sido un apoyo incondicional durante todo este tiempo, el que se ha pasado madrugadas enteras a mi lado estudiando, me ha cuidado cada vez que lo he necesitado, ha sido mi arco-iris en mis días grises.

A mis suegros por abrirme las puertas de su casa y tratarme como una hija.

A mi tutora Yunet, por todo el tiempo, esfuerzo y paciencia que me ha dedicado en estos meses, no lo olvidaré.

---

A profesores como Luis, Reinides, Jarvin y Lydia por todo su amor y dedicación en todo este tiempo.

A mis compañeros de aula por hacer de la escuela un lugar agradable.

## **RESUMEN**

La presente investigación centra su estudio en determinar fortalezas y debilidades tras la aplicación de la metodología de migración de datos Practical Data Migration.

Luego de la aplicación de la metodología y a partir de los criterios establecidos, los cuales fueron tomados de una investigación previa en el departamento, se procedió a realizar la comparación entre las metodologías RUP y XP con la metodología de estudio con el fin de determinar fortalezas y debilidades.

**Palabras clave:**

migración de datos, fortalezas, debilidades.

## **ABSTRACT**

This research focuses its study on determining strengths and weaknesses after the application of the Practical Data Migration data migration methodology.

After the application of the methodology and based on the established criteria, which were taken from a previous investigation in the department, a comparison was made between the RUP and XP methodologies with the study methodology in order to determine strengths and weaknesses.

**Keywords:**

data migration, strengths, weaknesses.

# ÍNDICE

<b>FIGURAS</b>	<b>XII</b>
<b>TABLAS</b>	<b>XIV</b>
<b>INTRODUCCIÓN</b>	<b>1</b>
<b>1. Fundamentos teórico y metodológicos de las metodologías de migración de datos</b>	<b>5</b>
1.1. Metodologías de migración de datos . . . . .	5
1.1.1. Metodología de desarrollo de software . . . . .	6
1.1.2. Migración de datos . . . . .	6
1.1.3. Necesidad de una migración de datos . . . . .	6
1.1.4. Estrategias de migración de datos . . . . .	7
1.1.5. Desafíos de la migración de datos . . . . .	7
1.1.6. Los 10 principales retos de migración de datos . . . . .	8
1.1.7. Importancia de una estrategia de migración de datos . . . . .	10
1.2. Practical Data Migration . . . . .	11
1.2.1. Análisis del panorama . . . . .	12
1.2.2. Análisis de brechas y mapeo . . . . .	12
1.2.3. Diseño y ejecución de migración . . . . .	12



1.2.4.	Desmontaje del legado . . . . .	13
1.2.5.	Reglas de calidad de datos . . . . .	13
1.2.6.	Gestión de los interesados claves de los datos . . . . .	13
1.2.7.	Plan de retiro del sistema . . . . .	14
1.2.8.	Estrategia de migración y gobernanza . . . . .	14
1.2.9.	Zona desmilitarizada . . . . .	14
1.2.10.	Reglas de Oro . . . . .	15
1.3.	Herramientas y tecnologías . . . . .	15
1.3.1.	Definición de Base de datos . . . . .	15
1.3.1.1.	Base de Datos Relacional . . . . .	16
1.3.1.2.	Base de Datos No Relacional . . . . .	16
1.3.1.3.	Comparación entre Base de Datos Relacional y Base de Datos No Relacional . . . . .	16
1.3.2.	Sistemas Gestores de base de datos . . . . .	19
1.3.2.1.	MySQL . . . . .	19
1.3.2.2.	MongoDB . . . . .	19
1.3.2.3.	Comparación entre MySQL y MongoDB . . . . .	20
1.3.3.	Lenguaje de programación . . . . .	21
1.3.3.1.	PHP . . . . .	21
1.3.3.2.	JavaScript . . . . .	21
1.3.3.3.	Comparación entre JavaScript y PHP . . . . .	22
1.3.3.4.	Symfony . . . . .	23
1.3.3.5.	Node.js . . . . .	24
1.3.3.6.	Comparación entre Node.js y Symfony . . . . .	24
1.3.3.7.	Front-end . . . . .	24

1.3.3.8. Back-end . . . . .	27
1.4. Criterios . . . . .	28
1.5. Conclusiones parciales . . . . .	29
<b>2. Aplicación de la metodología Practical Data Migration en el Sistema PESCASPIR y determinación de fortalezas y debilidades</b>	<b>30</b>
2.1. Descripción general del proyecto . . . . .	31
2.2. Alcance del proyecto de migración de datos . . . . .	31
2.3. Presupuesto del proyecto de migración de datos . . . . .	32
2.4. Organización formal de proyectos y programas . . . . .	32
2.5. Los módulos de la metodología Practical Data Migration . . . . .	32
2.5.1. Gestión de datos clave de las partes interesadas . . . . .	32
2.5.2. Análisis del paisaje . . . . .	32
2.5.3. Análisis de brechas y mapeo . . . . .	33
2.5.4. Reglas de calidad de datos (DQR) . . . . .	33
2.5.5. Diseño y ejecución de la migración . . . . .	37
2.5.5.1. Diseño Físico . . . . .	37
2.5.5.2. Pruebas . . . . .	39
2.5.6. Desmantelamiento heredado . . . . .	39
2.6. Políticas . . . . .	40
2.6.1. Metodología del proyecto . . . . .	40
2.6.1.1. Requisitos Funcionales . . . . .	41
2.6.1.2. Descripción de requisitos . . . . .	45
2.6.2. Arquitectónico . . . . .	46
2.7. Descomposición del proyecto . . . . .	48

2.7.1. Forma de la migración . . . . .	49
2.7.2. Plan de migración inicial . . . . .	49
2.8. Criterios Comunes . . . . .	51
2.8.1. Ciclos de Vida . . . . .	51
2.8.2. Enfoque de la migración . . . . .	51
2.8.3. Comparación de Metodologías de Migración basado en los criterios establecidos anteriormente . . . . .	52
2.9. Resultados y discusión . . . . .	59
2.9.1. Determinación de fortalezas y debilidades . . . . .	61
2.10. Conclusiones Parciales . . . . .	62
<b>CONCLUSIONES</b>	<b>63</b>
<b>RECOMENDACIONES</b>	<b>64</b>
<b>REFERENCIAS</b>	<b>65</b>
<b>ANEXOS</b>	<b>69</b>
<b>A. Análisis del Panorama</b>	<b>70</b>
A.1. Análisis del Panorama . . . . .	70
<b>B. Análisis de Brechas y Mapeo y Diseño Físico</b>	<b>77</b>
B.1. Brechas y mapeo . . . . .	77
B.2. Diseño Físico de la base de datos . . . . .	80
<b>C. Requisitos funcionales 1</b>	<b>88</b>
<b>D. Requisitos funcionales 2</b>	<b>96</b>

**E. Requisitos funcionales 3**

**105**

# FIGURAS

1.1. Representación de PDM2 . . . . .	11
2.1. Perfilado de datos de la tabla pescadería . . . . .	33
2.2. Mapeo de datos de pescadería . . . . .	33
A.1. Perfilado de datos de la tabla oferta . . . . .	70
A.2. Perfilado de datos de la tabla oferta_pescadería_producto . . . . .	71
A.3. Perfilado de datos de la tabla producto . . . . .	71
A.4. Perfilado de datos de la tabla receta . . . . .	71
A.5. Perfilado de datos de la tabla categoría de receta . . . . .	72
A.6. Perfilado de datos de la tabla servicio . . . . .	72
A.7. Perfilado de datos de la tabla categoría de servicio . . . . .	72
A.8. Perfilado de datos de la tabla opción . . . . .	73
A.9. Perfilado de datos de la tabla grupo . . . . .	73
A.10. Perfilado de datos de la tabla usuario . . . . .	73
A.11. Perfilado de datos de la tabla contacto . . . . .	74
A.12. Perfilado de datos de la tabla contacto_interés . . . . .	74
A.13. Perfilado de datos de la tabla sobre_nosotros . . . . .	74
A.14. Perfilado de datos de la tabla publicidad . . . . .	75
A.15. Perfilado de datos de la tabla petición_contacto . . . . .	75
A.16. Perfilado de datos de la tabla información_complementaria . . . . .	75
A.17. Perfilado de datos de la tabla personal . . . . .	76
A.18. Perfilado de datos de la tabla cliente . . . . .	76
B.1. Mapeo de datos de producto . . . . .	78

## FIGURAS

---

B.2. Mapeo de datos de usuario . . . . .	78
B.3. Mapeo de datos quienes somos . . . . .	79
B.4. Mapeo de datos personal . . . . .	79
B.5. Mapeo de datos receta . . . . .	80
B.6. Mapeo de datos servicios . . . . .	81
B.7. Mapeo de datos pregunta . . . . .	82

# TABLAS

1.1. Comparativa entre Base de Datos Relacionales y Base de Datos No Relacionales	17
1.1. Comparativa entre Base de Datos Relacionales y Base de Datos No Relacionales	18
1.2. Comparación entre MongoDB y MySQL	20
1.3. Comparación entre JavaScript y PHP	23
1.4. Comparación entre Node.js y Symfony	25
2.1. Regla de calidad de datos	34
2.1. Regla de calidad de datos	35
2.1. Regla de calidad de datos	36
2.1. Regla de calidad de datos	37
2.2. Tabla de equivalencias	39
2.3. Casos Válidos	40
2.4. Casos no válidos	40
2.5. Requisitos funcionales	41
2.5. Requisitos funcionales	42
2.5. Requisitos funcionales	43
2.5. Requisitos funcionales	44
2.6. RF1 Crear Usuario	45
2.5. Requisitos funcionales	45
2.7. Descripción de los requisitos no funcionales	46
2.7. Descripción de los requisitos no funcionales	47
2.7. Descripción de los requisitos no funcionales	48
2.8. Iteraciones y tiempo.	49
2.9. Módulos y tiempo	50

2.10. Comparación atendiendo al ciclo de vida . . . . .	51
2.11. Comparación para el enfoque de la migración . . . . .	52
2.12. Comparación de Metodologías . . . . .	53
2.12. Comparación de Metodologías . . . . .	54
2.12. Comparación de Metodologías . . . . .	55
2.12. Comparación de Metodologías . . . . .	56
2.12. Comparación de Metodologías . . . . .	57
2.12. Comparación de Metodologías . . . . .	58
2.12. Comparación de Metodologías . . . . .	59
C.1. RF2 Cambiar Usuario . . . . .	89
C.2. RF3 Borrar Usuario. . . . .	89
C.3. RF4 Enseñar Usuario. . . . .	90
C.4. RF5 Crear Producto. . . . .	90
C.5. RF6 Cambiar Producto . . . . .	91
C.6. RF7 Borrar Producto . . . . .	91
C.7. RF8 Enseñar Producto . . . . .	92
C.8. RF9 Crear Pescadería . . . . .	92
C.9. RF10 Cambiar Pescadería . . . . .	93
C.10. RF11 Borrar Pescadería. . . . .	93
C.11. RF12 Enseñar Pescadería. . . . .	94
C.12. RF13 Crear Receta . . . . .	94
C.13. RF14 Cambiar Receta . . . . .	95
C.14. RF15 Borrar Receta . . . . .	95
D.1. RF16 Enseñar Receta . . . . .	97
D.2. RF17 Crear Personal . . . . .	97
D.3. RF18 Cambiar Personal . . . . .	98
D.4. RF19 Borrar Personal . . . . .	98
D.5. RF20 Enseñar Personal . . . . .	99
D.6. RF21 Crear Servicio . . . . .	99
D.7. RF22 Cambiar Servicio . . . . .	100
D.8. RF23 Borrar Servicio . . . . .	100
D.9. RF24 Enseñar Servicio . . . . .	101



D.10.RF25 Crear Quienes Somos. . . . .	101
D.11.RF26 Cambiar Quienes Somos. . . . .	102
D.12.RF27 Borrar Sobre Nosotros. . . . .	102
D.13.RF28 Enseñar Quienes Somos . . . . .	103
D.14.RF29 Crear Embalse. . . . .	103
D.15.RF30 Cambiar Embalse . . . . .	104
D.16.RF31 Borrar Embalse. . . . .	104
E.1. RF32 Enseñar Embalse. . . . .	106
E.2. RF33 Crear Especies. . . . .	106
E.3. RF34 Cambiar Especies. . . . .	107
E.4. RF35 Borrar Especies. . . . .	107
E.5. RF36 Enseñar Especies. . . . .	108
E.6. RF7 Crear Pregunta Frecuente. . . . .	108
E.7. RF38 Cambiar Preguntas Frecuentes. . . . .	109
E.8. RF39 Borrar Pregunta Frecuente . . . . .	109
E.9. RF40 Enseñar Preguntas Frecuentes. . . . .	110
E.10. RF41 Autenticarse. . . . .	110
E.11. RF42 Cerrar Sesión . . . . .	110
E.12. RF43 Cambiar Contraseña . . . . .	111
E.13. RF44 Visualizar Información . . . . .	111

# INTRODUCCIÓN

Antiguamente, cuando la información se manejaba de forma manual, la "migración de datos" era una tarea sencilla. Si deseaba pasar de archivadores grises de cuatro cajones a otros de cinco cajones de color tostado, simplemente levantaba las carpetas de los archivadores viejos y las dejaba caer en los nuevos (ahaapps, 2014)

Actualmente la cantidad de información producida y utilizada por individuos y empresas por igual es continuamente creciente. Para manejar grandes cantidades de datos, se han desarrollado diferentes sistemas. Este proceso puede ser arriesgado y muy complejo según los sistemas involucrados o los datos que se tiene que migrar (Hofer, 2016).

La razón más común para la migración de datos es la necesidad de mover datos de un servidor a otro o a un nuevo sistema para escalar y acomodar el crecimiento del volumen. También existen otras circunstancias que pueden incitar a las organizaciones a optar por un proyecto de migración (Naeem, 2019)

Sin embargo, muchas empresas tratan incluso una migración masiva de datos como una tarea de dos clics de bajo nivel. Tal subestimación inicial se traduce en gastar más tiempo y recursos (AltexSoft, 2020)

Este proceso consta de las siguientes fases: descubrimiento, análisis, planeamiento, ejecución y de verificación, las cuales proporcionan una metodología probada (Netapp, 2021)

Es importante el trabajo con las metodologías de migración ya que permite reducir el nivel de dificultad, organizar las etapas, agilizar el proceso y mejorar el resultado final (Castillo Oporto

y Figueredo, 2021)

Cuba transita por un proceso que se ha definido como informatización de la sociedad: uno de los tres pilares que respalda la gestión gubernamental (Meneses, 2021)

El Primer Secretario del Comité Central del Partido Comunista de Cuba (PCC) y Presidente de la República, Miguel Díaz-Canel Bermúdez, exhortó a la sociedad espirituana a potenciar aún más el rol de la Universidad de Sancti Spíritus José Martí Pérez (UNISS) en la implementación del sistema de gestión de gobierno basado en ciencia e innovación (Escambray, 2022a)

Es una institución que se caracteriza por los aportes a las ciencias, obteniendo resultados de mayor impacto en lo referente al desarrollo de la ciencia, la técnica y la innovación que están relacionados con la ejecución de disímiles proyectos de investigación articulados con programas y proyectos de desarrollo económico (Escambray, 2022b)

Dentro de la UNISS se encuentra el departamento de Ingeniería Informática, que pertenece a la facultad de Ciencias Técnicas y Económicas, donde se desarrollan varios trabajos de diploma como ejercicio de culminación de estudio de la carrera Ingeniería Informática. Dichas tesis provienen en su gran mayoría de empresas empleadoras de esos futuros profesionales por lo que el estudiante se enfrenta a diversos temas, pero en ocasiones la investigación gira sobre la migración de un software que puede estar o no en uso en ese momento en la empresa, donde se evidencia la necesidad del trabajo exhaustivo y minucioso de las metodologías de migración.

Luego de un estudio bibliográfico sobre el tema se determinaron tres tesis donde se utilizaron metodologías ágiles y tradicionales para asumir un caso de migración de datos.

El Sistema de Explotación de los Medios Técnicos de Transporte comienza en la provincia de Sancti-Spíritus con un sistema para el control de los vehículos en la unidad de transporte del MININT en la provincia, luego se acordó automatizar y ampliar este sistema para que abarcara toda el área de trabajo. Surge para potenciar el trabajo de explotación de los vehículos y de control de los mismos. Para llevar a cabo la documentación del análisis, diseño e implementación del sistema se utilizó el lenguaje de modelado UML, siguiendo lo establecido por el Proceso de Desarrollo de Software (RUP) (Pérez Bravo, 2010)

Expósito Yanes (2020) desarrolló un software, que permitió una mayor compatibilidad al momento de crearse y que se integrara de manera correcta al sistema integral de gestión de redes (SIGERE). Para el desarrollo de la aplicación se utilizó la metodología XP (programación

extrema) y RUP; y como lenguaje de modelación el UML (Lenguaje Unificado de Modelado); lo cual posibilitó la adecuada documentación del análisis, diseño e implementación de la solución.

La Empresa de Tecnologías de la Información y la Automática de Sancti Spíritus brinda servicios técnicos con garantía de la calidad, en correspondencia con los requisitos internos, de los clientes y de otras partes interesadas, en las actividades de informática, automática y comunicaciones, para ello desarrolla el Sistema Integral de Gestión de Redes (SIGERE), dentro del cual se insertó en el subsistema de Explotación, El Módulo de transformadores. Este módulo está desarrollado en Delphi Object Pascal 2010, el cual no ha sido actualizado a una nueva versión ni mejorado con respecto a su creación, siendo una de las principales dificultades que este presentaba, la enorme cantidad de código legado debido a que los desarrolladores del mismo ya no se encuentran en la entidad y una gran cantidad de personas trabajaron en este, necesiándose la migración a C# con el objetivo de deshacerse de este código legado y la obsolescencia que presenta las herramientas utilizadas con respecto a los avances tecnológicos. La presente investigación se centra fundamentalmente en migrar a una aplicación web el sistema informático de transformadores eléctricos del sistema integral de gestión de redes de la UNE (Yero Grillo, 2020)

Tras esta situación es necesario que se desarrollen las investigaciones en torno a lo que realmente es petición del cliente que es la migración y no la reingeniería, temas que suelen mezclarse con mucha facilidad, hasta el momento en dicho departamento se ha aplicado para estos casos las metodologías de desarrollo de software tradicionales o ágiles, pero ninguna enfocada en la migración de datos en sí, lo que lleva a un mal uso de la misma, retraso en el desarrollo de los sistemas, pérdida de tiempo por parte de los desarrolladores y clientes con la reingeniería de los sistemas.

Castillo Oporto y Figueredo (2021) desarrollan una investigación enfocada en la detección de metodologías de migración de sistemas heredados y migración de datos, en la que se detectan cinco metodologías, dos para los Sistemas Heredados y tres para la migración de datos que pueden ser llevadas hasta la migración del sistema en sí, dicha investigación concluye que las metodologías MADIISH y la metodología Practical Data Migration son las más completas para realizar una migración.

Una de las recomendaciones planteadas en esa investigación es poner en marcha estas metodo-

logías con casos de estudios para compararlas con las metodologías usadas anteriormente en el departamento y así determinar sus fortalezas y debilidades, por este motivo se plantea como **problema de investigación** ¿Cómo contribuir a la determinación de fortalezas y debilidades de la metodología Practical Data Migration para asumir un proceso de migración de datos?

Para dar solución al problema anteriormente planteado se define como **objetivo general**: Aplicar la metodología Practical Data Migration para la migración del sistema PESCASPIR para determinar fortalezas y debilidades.

### **Preguntas de investigación:**

1- ¿Cómo fundamentar teórica y metodológicamente el uso de las metodologías de migración de datos?

2- ¿Cómo aplicar la metodología Practical Data Migration en el caso de estudio del Sistema PESCASPIR y determinar las fortalezas y debilidades obtenidas tras su aplicación?

### **Tareas de investigación:**

1. Fundamentar teórica y metodológicamente el uso de las metodologías de migración de datos.
2. Aplicar la metodología Practical Data Migration en el caso de estudio del Sistema PESCASPIR y Plantear las fortalezas y debilidades obtenidas tras su aplicación.

El presente trabajo está estructurado en introducción, dos capítulos de contenido, conclusiones, recomendaciones, referencias bibliográficas y anexos. El orden y contenido de los capítulos se describe a continuación:

**Capítulo 1. “Fundamentos teórico y metodológicos de las metodologías de migración de datos”**, se realiza un análisis sobre la migración de datos, haciendo énfasis en la metodología Practical Data Migration, con el objetivo de determinar todas sus fases. Se definieron las tecnologías y herramientas seleccionadas para el desarrollo del proyecto, y los criterios de comparación a tener en cuenta para determinar fortalezas y debilidades.

**Capítulo 2. “Aplicación de la metodología Practical Data Migration en el Sistema PESCASPIR y determinación de fortalezas y debilidades”**, se realiza una descripción general del proyecto, y se procede a ejecutar cada uno de los módulos propuestos por la metodología, generando así sus artefactos correspondientes y luego se procede a la comparación de esta metodología con otras dos para así obtener sus fortalezas y debilidades.

# CAPÍTULO 1

## Fundamentos teórico y metodológicos de las metodologías de migración de datos

El presente capítulo comprende el estudio de varios conceptos y características generales asociados a las metodologías de migración de datos, haciendo énfasis en la metodología Practical Data Migration y analizando cada una de sus fases. También, se caracterizará y se realizará una comparación entre los lenguajes de programación, las tecnologías y herramientas del sistema legado y el destino, donde se escogerá el más apropiado para el desarrollo de este proyecto, además se plasman los criterios de comparación que se tendrán en cuenta para determinar fortalezas y debilidades.

### 1.1. Metodologías de migración de datos

Para conseguir una migración acorde a las necesidades de las empresas es necesario organizar, planificar, diseñar cada operación a realizar; para esto se emplean las metodologías de desarrollo de software (Castillo Oporto y Figueredo, 2021)

### 1.1.1. Metodología de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. El objetivo de las distintas metodologías es el de intentar organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible (Santander, 2020)

Dentro de estas se encuentran las metodologías de migración de datos.

### 1.1.2. Migración de datos

En general, la migración de datos significa mover información digital. La transferencia de esa información a una ubicación, formato de archivo, entorno, sistema de almacenamiento, base de datos, centro de datos o aplicación diferente se ajusta a la definición de migración de datos. Para definir la migración de datos más específicamente: La migración de datos es el proceso de seleccionar, preparar, extraer y transformar datos y transferirlos permanentemente de un sistema de almacenamiento informático a otro (Microsoft, 2022)

### 1.1.3. Necesidad de una migración de datos

Los motivos habituales para utilizar la migración de bases de datos son:

- Actualización a la última versión del software de base de datos para mejorar la seguridad y el cumplimiento.
- Mover datos existentes a una nueva base de datos para reducir costos, mejorar el rendimiento y lograr escalabilidad.
- Pasar de una base de datos local a una base de datos basada en la nube para una mejor escalabilidad y menores costos.
- Combine datos de varias bases de datos en una sola base de datos para obtener una vista de datos unificada después de la fusión.

(Fatima, 2022)

### 1.1.4. Estrategias de migración de datos

Al planear una migración de datos, debe considerarse la estrategia más adecuada de acuerdo con las necesidades de la organización. La elección puede realizarse dependiendo de los requisitos del proyecto y las ventanas de procesamiento disponibles. Existen dos enfoques principales de migración (Software, 2022)

#### **Migraciones en bigbang:**

Implican completar toda la migración en una pequeña ventana de procesamiento definida. La consecuencia de este enfoque es la inactividad del sistema mientras los datos se extraen de la fuente, se procesan, se cargan en el destino, y luego se redirecciona el procesamiento al nuevo entorno.

Este enfoque resulta atractivo, en la medida en que permite completar la migración en el menor tiempo posible, pero conlleva riesgos, pues pocas organizaciones pueden vivir con un sistema central no disponible por mucho tiempo, por lo que existe una intensa presión sobre la migración. La verificación y aprobación de datos se encuentran en la ruta crítica.

#### **Migraciones incrementales:**

En este caso, en lugar de intentar completar todo el evento en una ventana de tiempo breve, la migración implica tener ejecutándose los sistemas antiguo y nuevo en paralelo y migrar los datos en fases.

Este método proporciona un tiempo de inactividad cero como lo requieren las aplicaciones de misión crítica que operan 24/7. Este enfoque agrega cierta complejidad al diseño, ya que debe ser posible rastrear qué datos se han migrado y qué datos no, para redireccionar las peticiones cuando los sistemas operan en paralelo

### 1.1.5. Desafíos de la migración de datos

Pese a que la migración de datos puede ser un proceso sencillo, su implementación puede encontrarse con desafíos a los que habrá que hacer frente (PowerData, 2018)

- Descubrir que el código fuente de la aplicación de origen no está disponible y el fabricante de dicha aplicación ya no se encuentra en el mercado ya.



## 1.1 Metodologías de migración de datos

---

- Encontrar tipos o formatos de datos de origen que no tienen correspondencia en destino: números, fechas, sub-registros.
- Problemas de codificación que afecten a determinados conjuntos de datos.
- Existencia de optimizaciones en el formato de almacenamiento de datos, tales como almacenamiento binario decimal codificado, almacenamiento no estándar de los valores numéricos positivos / negativos o los tipos de almacenamiento de los que se excluyen mutuamente sub-registros dentro de un registro.
- Cuestiones relativas a la aparición de redundancias y duplicidades cuando, al tiempo que se llevaba a cabo la migración de datos, diferentes tipos de usuario empleaban el sistema o aplicación antiguo y el nuevo

### 1.1.6. Los 10 principales retos de migración de datos

Aunque la migración de datos lleva décadas siendo una realidad en la tecnología, seguimos conociendo historias terroríficas cada año. A continuación, se presentan los 10 principales retos que las empresas encuentran en el movimiento de datos (Netapp, 2021)

**1. No contactar a los interesados clave:** No importa el tamaño de la migración, hay alguien, en algún lugar que se preocupa por los datos que se están moviendo. Contacta con ellos y explícales la necesidad de este proyecto y su impacto antes de continuar con la tarea. Si no lo haces, ten por seguro que tendrás noticias de ellos en algún momento, muy probablemente para interrumpir tus plazos.

**2. Falta de comunicación con los responsables:** Una vez que hayas explicado el proyecto a los principales interesados, asegúrate de que se mantienen al tanto de tu progreso. Lo mejor es proporcionar un informe de estado el mismo día cada semana, sobre todo si hay cambios. La comunicación regular facilita en gran medida la confianza de las partes afectadas.

**3. Falta de gobierno de datos:** Asegúrate de que queda claro quién tiene los derechos para crear, aprobar, editar o eliminar datos del sistema de origen y dejarlo por escrito como parte del plan del proyecto.

**4. Falta de experiencia:** Aunque se trata de una tarea sencilla, hay mucha complejidad en el movimiento de datos. Tener un profesional experimentado con excelentes referencias ayuda a

que el proceso se desarrolle sin problemas.

**5. Falta de planificación:** De media, las familias pasan de 10 a 20 horas planificando sus vacaciones, mientras que los equipos de TI pueden pasar tan sólo la mitad de ese tiempo planificando una pequeña migración de datos. Las horas de planificación no siempre garantizan el éxito, pero el hecho de tener un plan de migración de datos sólido ahorra horas cuando se trata de mover los datos.

**6. Software y habilidades de preparación de datos insuficientes:** Si se trata de una gran migración (millones de registros o cientos de tablas), invierte en un buen software de calidad de datos y contempla la posibilidad de contratar a una empresa especializada para que te ayude. La buena noticia es que una empresa externa probablemente te alquile el software para ayudarte a ahorrar costes.

**7. Esperar las especificaciones perfectas del objetivo:** Si el equipo de implementación está resolviendo los criterios de diseño, haz presión en los pasos 2 y 3. Prepararse para el objetivo será importarte en fases posteriores del proyecto, pero eso no debe detenerte ahora.

**8. Metodología de migración no probada:** Investiga un poco para asegurarte de que el procedimiento de movimiento de datos ha funcionado bien para otras empresas como la tuya. No caigas en la tentación de aceptar el procedimiento genérico que te ofrezca un proveedor.

**9. Gestión de proveedores y proyectos:** Los proveedores y proyectos deben gestionarse. Si todavía estás con tu trabajo cotidiano, asegúrate de que tienes tiempo para gestionar el proyecto y cualquier proveedor relacionado.

**10. Dependencias entre objetos:** Con la tecnología y las funcionalidades de las herramientas de gestión de datos disponibles en la actualidad, sigue resultando sorprendente detectar un conjunto de datos dependiente que no se incluyó en el plan original. Dado que las dependencias entre objetos a menudo no se descubren hasta muy tarde en el proceso de migración, asegúrate de contar con un plan de contingencia para ellos de modo que la fecha de entrega no se vea afectada.

### 1.1.7. Importancia de una estrategia de migración de datos

Independientemente de la finalidad exacta de una migración de datos, el objetivo suele ser mejorar el rendimiento y la competitividad. Las migraciones menos exitosas pueden dar lugar a datos inexactos que contienen redundancias e incógnitas. Esto puede ocurrir incluso cuando los datos de origen son totalmente utilizables y adecuados. Además, cualquier problema que existiera en los datos de origen puede verse amplificado cuando se introducen en un sistema nuevo y más sofisticado. Una estrategia completa de migración de datos evita una experiencia deficiente que acaba creando más problemas de los que resuelve. Además de incumplir los plazos y sobrepasar los presupuestos, los planes incompletos pueden hacer que los proyectos de migración fracasen por completo. A la hora de planificar y elaborar la estrategia del trabajo, los equipos deben prestar toda su atención a las migraciones, en lugar de subordinarlas a otro proyecto de gran alcance.

Un plan estratégico de migración de datos debe incluir la consideración de estos factores críticos:

- **Conocer los datos:** antes de la migración, los datos de origen deben someterse a una auditoría completa. Pueden surgir problemas inesperados si se ignora este paso.
- **Limpieza :** Una vez identificados los problemas con los datos de origen, hay que resolverlos. Esto puede requerir herramientas de software adicionales y recursos de terceros debido a la magnitud del trabajo.
- **Mantenimiento y protección :** Los datos se degradan tras un periodo de tiempo, lo que hace que no sean fiables. Esto significa que debe haber controles para mantener la calidad de los datos.
- **Gobernanza :** El seguimiento y la presentación de informes sobre la calidad de los datos es importante porque permite comprender mejor la integridad de los mismos. Los procesos y las herramientas utilizadas para producir esta información deben ser altamente utilizables y automatizar las funciones cuando sea posible.

Además de un procedimiento estructurado paso a paso, un plan de migración de datos debe incluir un proceso para incorporar el software y las herramientas adecuadas para el proyecto (Curran, 2021)

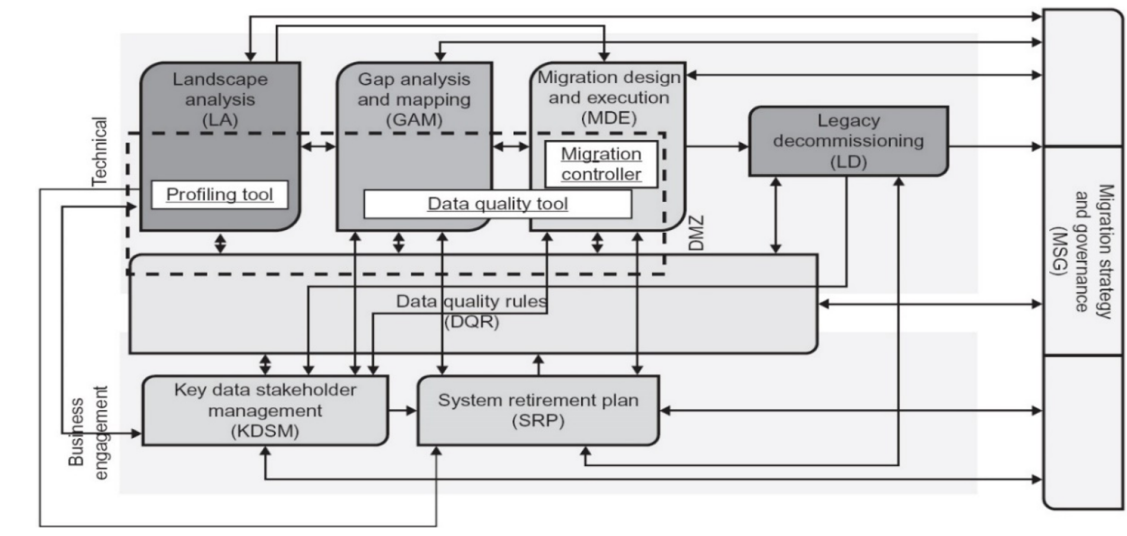


Figura 1.1: Representación de PDM2

## 1.2. Practical Data Migration

La metodología Practical Data Migration propone, en primera instancia, que la migración de datos debe manejarse como un proyecto independiente en sí mismo debido a que tiene sus entregables y reglas específicas, requiere un grado de negociación entre lo funcional y lo técnico en la organización, precisa de habilidades especializadas y no puede encasillarse dentro de la estructura de proyectos estándar. El modelo es denominado “Practical Data Migration v2” (en adelante PDM2) y propone un enfoque integral que consta de varios módulos que cubren el alcance completo de la migración de datos, desde sus inicios hasta la desconexión de los sistemas legados. También presenta un resumen de los tipos de tecnología disponibles para soportar la migración de datos (Software, 2022)

Los módulos considerados por el autor se encuentran representados en el modelo PDM2 de la Figura 1.1 y se encuentran organizados en dos flujos o streams: el primero es el de negocio y el segundo el de tecnología. La idea es integrar ambas partes del proyecto.

### 1.2.1. Análisis del panorama

Este módulo usa varias técnicas para descubrir y catalogar almacenamientos de datos legados (legacy data stores – LDS) y su relación entre ellos. Se analizan los sistemas de almacenamiento de datos de los legados para ver cómo funcionan, qué datos tienen y qué “sorpresas” pueden contener. Este perfilamiento de datos se puede realizar mediante el uso de herramientas de software disponibles y de forma manual. Es necesario buscar conscientemente todos los legados disponibles, no solo los oficiales de empresa. Una importante ventaja de este método es que el análisis del panorama puede comenzar antes del diseño o incluso la selección del sistema de destino (Castillo Oporto y Figueredo, 2021)

### 1.2.2. Análisis de brechas y mapeo

Se realiza el mapeo de datos una vez que el sistema de destino está disponible. El mapeo de datos es la vinculación de campos en los legados con los campos en el destino, además de definir la lógica de transformación que se necesita para dividir datos y fusionar campos. Un ejemplo clásico de esto es reformatear nombre y direcciones donde quizás una base de datos fuente tenga el nombre en un solo campo, pero el destino tiene el nombre y el apellido por separado y solo contiene la primera línea de la dirección, el resto se deriva de un archivo postal nacional basado en un código postal (Castillo Oporto y Figueredo, 2021)

### 1.2.3. Diseño y ejecución de migración

Es donde el diseño físico, pruebas y ejecución de la migración se llevan a cabo. La migración de datos es más que simplemente mover bits y bytes. Se debe conocer las limitaciones de negocio, tiempos, requisitos de auditoría, linaje de datos, plan de vuelta atrás (rollback), requisitos de informes, de gestión y control, etc. Este módulo integra también los requisitos comerciales para la desconexión y retiro de los sistemas legados (Castillo Oporto y Figueredo, 2021)

### 1.2.4. Desmontaje del legado

Cubre la eliminación física o lógica de bases de datos legadas, hardware y software. También cubre la entrega de almacenamiento de datos legados para componentes de datos que deben retenerse por algún tiempo pero que no se deben migrar al destino. También hay procesos de cierre del proyecto, incluido el traspaso de problemas de calidad de datos (que no fue posible ajustar dentro del tiempo del proyecto y por restricciones de presupuesto) a los equipos de calidad de datos del día a día (donde existan) (Castillo Oporto y Figueredo, 2021)

### 1.2.5. Reglas de calidad de datos

Es la pieza central que hace único el modelo PDM2. Siguiendo el principio de Super SMART Tasks, este módulo gestiona toda la calidad de los datos. Integra a los expertos en sistemas técnicos heredados, los especialistas del sistema destino y los expertos en dominios de negocio para priorizar, administrar y completar todos los problemas de datos, incluida la selección y exclusión de fuentes de datos. Es “Súper Inteligente” porque integra el equipo al hacer que se relacione con los recursos del resto de la empresa, lo que implica la creación de un solo equipo virtual; empodera a los colegas del negocio y les da las habilidades y la oportunidad de hacer una contribución positiva al proyecto. Se completa la tarea al incorporar conocimiento empresarial muy necesario en un marco colaborativo (Castillo Oporto y Figueredo, 2021)

### 1.2.6. Gestión de los interesados claves de los datos

PDM2 tiene sus propias definiciones de roles específicos para cada interesado clave de datos. La gestión de los interesados clave de los datos gestiona el descubrimiento, la información y la gestión de estos individuos. PDM2 está muy centrado en el negocio, por lo que hay tantos roles de negocio como técnicos. Los dos interesados más importantes son los propietarios de los datos y los expertos en el dominio empresarial. Los propietarios de datos son todas las personas dentro o fuera de una organización que tiene el poder legítimo de detener una migración (Castillo Oporto y Figueredo, 2021)

### 1.2.7. Plan de retiro del sistema

El objetivo final de una migración de datos es desactivar los sistemas legados. Es por esta razón, que bajo el modelo PDM2 propone iniciar la conversación con el negocio por este tema. Se trata de una serie de preguntas estructuradas para obtener una visión empresarial de la migración, buscando elementos necesarios como requisitos de:

- Auditoría de la migración.
- Linaje de los datos.
- Retención de datos.
- Restricciones de migración.
- Requisitos de prueba de aceptación del usuario.
- Respaldo de datos.

(Castillo Oporto y Figueredo, 2021)

### 1.2.8. Estrategia de migración y gobernanza

Cubre todas las funciones de gerencia estándar que se esperan en un proyecto bien gestionado, más algunas actividades exclusivas que PDM2 propone. Se realiza especial énfasis en la creación de una estrategia de migración de datos que debe completarse con la participación de la alta gerencia (Castillo Oporto y Figueredo, 2021)

### 1.2.9. Zona desmilitarizada

Hace las veces de interfaz entre el trabajo del proveedor de tecnología y las responsabilidades de sus clientes. La DMZ es un componente clave de PDM2 que, en cierta medida, se definirá formalmente en el contrato con el proveedor. Sin embargo, la DMZ es más amplia que el contrato, y su definición formal ayudará a ambas partes a comprender y gestionar sus dependencias.

El modelo expone además la importancia de la tecnología en el desarrollo de los proyectos de migración de datos, considerando las herramientas subyacentes y de apoyo a procesos como la calidad de datos, el perfilamiento de datos y controladores para la migración. Sugiere, además, que el uso de software especializado para estos fines reduce el riesgo del proyecto, en lugar de realizar desarrollos propios, pues han sido probadas por más tiempo y con la lógica específica de la migración (Castillo Oporto y Figueredo, 2021)

### 1.2.10. Reglas de Oro

A continuación, se presentan las cuatro reglas de oro de la migración de datos, que estarán presentes en todas las decisiones que se vayan a tomar. Estos son la base de todas las demás actividades de esta metodología y son el conjunto de conceptos más importante dentro de él (Morrays, 2012)

1. La migración de datos es un negocio, no un problema técnico.
2. El negocio sabe mejor.
3. Ninguna organización necesita, quiere o pagará por datos de calidad perfecta.
4. Si no puedes contarlo, no cuenta.

## 1.3. Herramientas y tecnologías

Para el desarrollo de un software se deben tener en cuenta ciertos elementos, como son las tecnologías y las herramientas básicas para su construcción. A continuación, se describen las principales utilizadas.

### 1.3.1. Definición de Base de datos

Una base de datos es una herramienta para recopilar y organizar información. Las bases de datos pueden almacenar información sobre personas, productos, pedidos o cualquier otra cosa.



Muchas bases de datos comienzan como una lista en un programa de procesamiento de texto o una hoja de cálculo. A medida que la lista crece, comienzan a aparecer redundancias e inconsistencias en los datos (Support, 2022)

### 1.3.1.1. Base de Datos Relacional

Las bases de datos relacionales son un tipo de base de datos que almacena y organiza puntos de datos con relaciones definidas para un acceso rápido. Con una base de datos relacional, los datos se organizan en tablas que contienen información sobre cada entidad y representan categorías predefinidas a través de filas y columnas. La estructuración de los datos de esta manera hace que el acceso sea eficaz y flexible, motivo por el que las bases de datos relacionales son más comunes. Las bases de datos relacionales también se crean para comprender Lenguaje de consulta estructurado (SQL), un lenguaje de programación estandarizado que se usa para almacenar, manipular y recuperar datos (Azure, 2022)

### 1.3.1.2. Base de Datos No Relacional

Una base de datos no relacional es aquella que no usa el esquema tabular de filas y columnas que se encuentra en la mayoría de los sistemas de base de datos más tradicionales. En su lugar, las bases de datos no relacionales usan un modelo de almacenamiento que está optimizado para los requisitos específicos del tipo de datos que se almacena. Por ejemplo, los datos se pueden almacenar como pares clave/valor simple, como documentos JSON o como un grafo que consta de bordes y vértices (EdPrice-MSFT, 2022)

### 1.3.1.3. Comparación entre Base de Datos Relacional y Base de Datos No Relacional

Tabla Comparativa entre Base de Datos Relacionales y Base de Datos No Relacionales ver Tabla1.1

Tabla 1.1: Comparativa entre Base de Datos Relacionales y Base de Datos No Relacionales

<b>Base</b>	<b>Bases de datos relacionales</b>	<b>Bases de datos no relacionales</b>
SQL	Las bases de datos relacionales son la base de datos que almacena los datos de forma estructurada. Por lo tanto, las bases de datos relacionales o RDBMS también se denominan SQL.	Las bases de datos no relacionales se basan en documentos. No definen ninguna relación y mantienen datos en el formulario del documento. Por lo tanto, estas bases de datos también se denominan NoSQL.
Lenguaje de consulta	En esta base de datos se utiliza el lenguaje de consulta estructurado o SQL.	Pero en las bases de datos no relacionales, entra en uso el lenguaje de consulta no declarativo. En resumen, estas bases de datos utilizan NoSQL o un lenguaje de consulta no estructurado.
Idoneidad	Estas son las bases de datos que no son ideales para el almacenamiento jerárquico de datos. Por lo tanto, si la empresa necesita almacenar sus datos jerárquicamente, debe evitar las bases de datos relacionales.	Estas bases de datos son adecuadas para el almacenamiento jerárquico de los datos de la empresa porque aquí, estas bases de datos admiten el método clave-valor. Y al utilizar esta función, la empresa puede almacenar fácilmente sus datos de forma jerárquica.

Tabla 1.1: Comparativa entre Base de Datos Relacionales y Base de Datos No Relacionales

<b>Base</b>	<b>Bases de datos relacionales</b>	<b>Bases de datos no relacionales</b>
Esquema	El RDBMS viene con la función de esquema predefinido.	Pero los Non-RDBMS utilizan el esquema dinámico para almacenar los datos no estructurados.
Uso ideal	Si la empresa necesita validar los datos de sus clientes de manera eficiente. Entonces, las bases de datos relacionales pueden ser la opción ideal.	Estas bases de datos son excelentes si la empresa necesita una experiencia de datos rápida en lugar de los datos correctos del cliente.
Escalabilidad	Estas bases de datos son escalables verticalmente, o agregando más filas, la base de datos se puede expandir.	Considerando que estos son el tipo de bases de datos que utiliza el documento y, por lo tanto, puede ser escalable horizontalmente.
Apoyo	Esta base de datos utiliza ACID, la forma abreviada de atomicidad, consistencia, aislamiento y durabilidad. Estas propiedades están incorporadas en las bases de datos relacionales.	Las bases de datos no relacionales utilizan BASE para almacenar los datos ideales, que representan el marco de consistencia disponible, de estado suave y eventualmente.
Ejemplos de	MySQL, Oracle y Microsoft SQL Server son algunos ejemplos de RDBMS.	Algunos ejemplos de bases de datos no relacionales son MongoDB, HBase y Cassandra.

Entonces, ambos tipos de bases de datos tienen varias ventajas y desventajas. En base a estos, la empresa decide qué base de datos debe optar. Sin embargo, si la empresa necesita una ex-

perencia de datos rápida, las bases de datos no relacionales son la opción ideal (CursosGratis, 2021)

### 1.3.2. Sistemas Gestores de base de datos

Un Sistema Gestor de Base de Datos (SGBD) o DGBA (Data Base Management System) es un conjunto de programas no visibles que administran y gestionan la información que contiene una base de datos. Los gestores de base de datos o gestores de datos hacen posible administrar todo acceso a la base de datos ya que tienen el objetivo de servir de interfaz entre ésta, el usuario y las aplicaciones (Data, 2019)

#### 1.3.2.1. MySQL

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto. Desarrollado originalmente por MySQL AB, fue adquirida por Sun Microsystems en 2008 y esta su vez comprada por Oracle Corporation en 2010, la cual ya era dueña de un motor propio InnoDB para MySQL.

Es una opción razonable para ser usado en ámbito empresarial. Al estar basado en código abierto permite a pequeñas empresas y desarrolladores disponer de una solución fiable y estandarizada para sus aplicaciones (OpenWebinars, 2019)

#### 1.3.2.2. MongoDB

MongoDB es una aplicación de gestión de base de datos NoSQL. Los sistemas de base de datos NoSQL ofrecen una alternativa a las bases de datos relacionales tradicionales que utilizan SQL (lenguaje de consulta estructurado). Los datos se almacenan en tablas, filas y columnas en una base de datos relacional, con relaciones entre entidades. En MongoDB, los datos se almacenan en documentos utilizando una estructura similar a JSON para representar e interactuar con los datos (Ubuntu, 2022)

## 1.3 Herramientas y tecnologías

Tabla 1.2: Comparación entre MongoDB y MySQL

	MongoDB	MySQL
Mayor disponibilidad	MongoDB replica automáticamente sus datos en nodos adicionales para una alta disponibilidad y durabilidad. En caso de falla del sistema, la conmutación por error se completa automáticamente, generalmente en menos de 5 segundos.	Si bien MySQL puede replicar datos en otro nodo, la conmutación por error entre nodos es un proceso manual complejo que aumenta el tiempo de inactividad de la aplicación.
Desarrollo más rápido	El modelo de datos de documentos de MongoDB se asigna de forma natural a los objetos en el código de la aplicación, lo que simplifica el aprendizaje y el uso por parte de los desarrolladores.	Si bien muchos desarrolladores están familiarizados con SQL y el modelo relacional que utiliza MySQL, imponen restricciones en el esquema de la base de datos y el modelado de datos que ralentizan el desarrollo.
Escale infinitamente y de forma económica	MongoDB incluye soporte nativo para distribuir o fragmentar una base de datos en cualquier cantidad de máquinas básicas de manera transparente para la aplicación.	Escalar MySQL requiere comprar un servidor más robusto o implementar una solución de fragmentación más compleja en la aplicación.

### 1.3.2.3. Comparación entre MySQL y MongoDB

Tabla Comparativa entre MongoDB y MySQL ver Tabla 1.2

El rendimiento de la base de datos puede variar ampliamente según varios factores: el diseño de la base de datos, los patrones de consulta de la aplicación y la carga en la base de datos son solo algunos. Dado que el modelo de documentos de MongoDB almacena datos relacionados juntos, a menudo es más rápido recuperar un solo documento de MongoDB que UNIR datos en varias tablas en MySQL. Muchos clientes han evaluado y seleccionado MongoDB en lugar de MySQL, tanto por su mejor rendimiento a escala como por las mejoras radicales en la productividad de los desarrolladores (MongoDB, 2022)

### 1.3.3. Lenguaje de programación

En informática, se conoce como lenguaje de programación a un programa destinado a la construcción de otros programas informáticos. Su nombre se debe a que comprende un lenguaje formal que está diseñado para organizar algoritmos y procesos lógicos que serán luego llevados a cabo por un ordenador o sistema informático, permitiendo controlar así su comportamiento físico, lógico y su comunicación con el usuario humano (Conceptode, 2021)

#### 1.3.3.1. PHP

PHP (acrónimo recursivo de PHP: HypertextPreprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML (PHP, 2022)

#### 1.3.3.2. JavaScript

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web (Amazon, 2022)

**Ventajas de JavaScript** (freeCodeCamp, 2021)

**Velocidad :** JavaScript tiende a ser muy rápido porque a menudo se ejecuta inmediatamente en el navegador. Entonces mientras no requiera recursos externos, JavaScript no tiene permitido retrasarse por llamados del servidor backend.

**Simplicidad:** La sintaxis de JavaScript está inspirada por Java y es relativamente sencillo de aprender comparado a otros lenguajes de programación populares como C++.

**Popularidad :** JavaScript esta por todas partes de la web, y con la llegada de Node.js, se ha incrementado su uso en backend. Hay incontables recursos para aprender JavaScript. Tanto StackOverflow como GitHub muestran un creciente número de proyectos que usan JavaScript, y la popularidad que ha alcanzado en los recientes años se espera que siga creciendo.

**Compatibilidad:** A diferencia de PHP u otros lenguajes scripting, JavaScript puede ser usado en cualquier página web. JavaScript puede ser usado en diferentes tipos de aplicaciones gracias al soporte en otros lenguajes como Pearl y PHP.

**Server Load:** JavaScript es client-side, entonces esto reduce la demanda de servidores en general, y las aplicaciones sencillas puede que no necesiten el servidor del todo.

Interfaces sencillas- JavaScript puede ser usado para crear características como arrastrar y soltar, y componentes tales como las diapositivas, lo cual mejora enormemente la interfaz de usuario y la experiencia del sitio.

**Funcionalidad extendida:** Los desarrolladores pueden extender la funcionalidad de las páginas web mediante fragmentos de JavaScript para un tercer grupo de extensiones como Greasemonkey.

**Versatilidad:** Hay muchos métodos para usar JavaScript mediante servidores Node.js. Si tú estas en bootstrap Node.js con Express, puedes usar un documento database como MongoDB, y usar JavaScript en el frontend para clientes. Es posible desarrollar una aplicación completa de JavaScript desde principio a fin solo usando JavaScript.

**Actualizaciones:** Desde que la llegada de ECMAScript 5 (la especificación escrita en que se basa JavaScript), ECMA International se ha dedicado a actualizar JavaScript anualmente. Hasta el momento, hemos recibido soporte para el navegador ES6 en 2017 y esperamos que ES7 sea soportado en un futuro.

### 1.3.3.3. Comparación entre JavaScript y PHP

Tabla comparativa entre JavaScript y PHP ver Tabla 1.3

Ambos lenguajes son bastante buenos en términos de soporte comunitario, extensibilidad y aplicaciones para las que son adecuados. JavaScript es ciertamente más eficiente en términos de velocidad y universalidad. Mientras tanto, pierde frente a PHP en términos de curva de aprendizaje, aunque la sintaxis, como concluimos, es solo una cuestión de preferencia personal (freeCodeCamp, 2020)

## 1.3 Herramientas y tecnologías

Tabla 1.3: Comparación entre JavaScript y PHP

	<b>JavaScript</b>	<b>PHP</b>
Breve descripción	Un lenguaje Full Stack. Lanzado en 1995.	Lenguaje Back-end. Lanzado en 1994.
Rendimiento y velocidad	Rápida.	Lenta.
Extensibilidad	Puede ser combinado con HTML, XML y Ajax. Varios frameworks disponibles.	Puede ser combinado con HTML
Universalidad.	Cross-Platform Full Stack	Cross-Platform Back-end
Comunidad	Grande	Grande
Aprendizaje	Un poco difícil.	Simple.
Sintaxis	Simple y compacta	Simple y compacta

### 1.3.3.4. Symfony

Symfony es un proyecto PHP de software libre que permite crear aplicaciones y sitios web rápidos y seguros de forma profesional (Symfony, 2022)

#### Las ventajas de Symfony (Diligent, 2017)

- Es extremadamente flexible: se adapta a casi cualquier necesidad, permitiendo instalar únicamente las piezas requeridas para el proyecto en vez de todo el framework.
- Es tremendamente ampliable: según GitHub, es el proyecto php más activo, lo que te garantiza encontrar paquetes para prácticamente cualquier funcionalidad.
- Un sistema rápido y que consume poca memoria: Symfony ha sido desarrollado con la idea de ofrecer aplicaciones de alto rendimiento, es más rápido que otros frameworks usando la mitad de la memoria.
- Facilidad de uso: Gracias a la gran cantidad de documentación y tutoriales que se pueden encontrar en la web, cualquier profesional o usuario avanzado puede aprender rápidamente los conceptos más básicos de Symfony. Para ello ha tomado las mejores ideas de sus competidores, como Ruby on Rails o Django



### 1.3.3.5. Node.js

Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables (Node.js, 2022)

#### **Beneficios de utilizar Node.js** (Platzi, 2018)

- La compilación de Node.js se realiza en tiempo de ejecución, Just In Time (JIT), esto trae consigo una mayor optimización a las funciones que más veces sean llamadas.
- Mediante clusters permite tener una escalabilidad alta.
- Podemos expandir nuestro código añadiendo módulos de forma fácil gracias al Node Package Manager (NPM).
- Un alto rendimiento en proyectos donde necesitemos ejecución en tiempo real.
- En startups o equipos pequeños podremos realizar front-end, back-end y hasta una aplicación móvil con un mismo lenguaje

### 1.3.3.6. Comparación entre Node.js y Symfony

Tabla comparativa entre Node.js y Symfony ver Tabla 1.4

Para resumir Node.js puede procesar múltiples funciones y eventos en paralelo. Es ideal para crear cualquier solución personalizada. Utiliza JavaScript, por lo que se usa un solo idioma en toda la aplicación, incluidos el backend y el front-end, lo que hace que las operaciones de código sean muy convenientes. Debido a la contenedorización o los microservicios, la escalabilidad de la arquitectura se puede lograr sin problemas. Sus llamadas de servicio son más ligeras que las llamadas de PHP. Permite el establecimiento de una conexión cliente-servidor en tiempo real. Proporciona la facilidad de trabajar con JSON (Almeida, 2022)

### 1.3.3.7. Front-end

Angular es un framework aplicativo JavaScript desarrollado por Google, que permite crear aplicaciones web mono página (es decir, completamente cargadas en el cliente antes de que el usuario las utilice). Estas aplicaciones solo hacen peticiones al servidor para intercambiar

## 1.3 Herramientas y tecnologías

Tabla 1.4: Comparación entre Node.js y Symfony

	<b>Node.js</b>	<b>PHP</b>	<b>Ganador</b>
Rendimiento	Lenguaje compilado	Lenguaje interpretado	Node.js
Desarrollo	Un solo paquete con funcionalidad. Utiliza JavaScript, que puede ser un lenguaje completo. Puede integrarse con multitud de sistemas.	Más rápido para codificar.	Empate
Base de datos	SQL No SQL Graph	SQL NoSQL (difícil de integrar)	Node.js
Hosting	SmartOS Nodejitsu Heroku	Compatible con la mayoría de los proveedores Se puede integrar en la pila de tecnología LAMP	PHP
Servicios prácticos	Servicios de transmisión Mensajeros instantáneos Educación en línea Redes sociales Aplicaciones Fintech Herramientas de gestión de proyectos Aplicaciones de juegos Chats en vivo	Se puede integrar con CMS populares como WordPress o Joomla Mercados de comercio electrónico sitios web Plataformas de mercadeo blogs	Depende del proyecto
Velocidad de ejecución	Procesos concurrentes	Procesos secuenciales	Node.js

los datos (y no para cambiar las páginas web), a través de los servicios web. En el marco frecuente de una arquitectura MEAN, Angular está acoplado con un servidor Node.js (él mismo enriquecido con el framework Express) y en unión con un sistema de gestión básico de datos NoSQL MongoDB (ENI, 2022)

### **Ventajas de usar Angular** (Coppola, 2022):

1. Alta calidad de la aplicación: Si bien Angular es una plataforma compleja y difícil de aprender para algunos desarrolladores web principiantes, esto, a la vez, también representa una enorme ventaja, pues el éxito del producto está prácticamente asegurado. Puedes crear cualquier cosa que se te ocurra y hacerla realidad con las funciones integradas del marco de trabajo.
2. Desarrollo multiplataforma: Angular ofrece soluciones de aplicaciones web progresivas angulares que puedes ejecutar en plataformas móviles, pero el marco puede utilizarse muy bien en aplicaciones móviles nativas.
3. Proceso de desarrollo web más rápido: Angular te permite crear aplicaciones de forma más rápida y eficiente, pues goza de ventajas técnicas como las siguientes:

Documentación detallada. La documentación de Angular está cuidadosamente escrita y dotada de una gran variedad de ejemplos de código para tener mayor claridad en el proceso y permitir que los desarrolladores encuentren soluciones rápidas a cualquier problema conforme crean una aplicación.

Interfaz de línea de comandos de Angular. La CLI (por sus siglas en inglés) facilita el trabajo de los desarrolladores al ofrecer un conjunto de herramientas de codificación. Además, esta línea de comandos puede ampliarse con bibliotecas de terceros para resolver problemas de software inusuales o muy complicados.

Enlace de datos bidireccional. Como ya lo mencionamos, Angular cuenta con esta característica, lo que permite el ahorro de tiempo al automatizar algunos procesos de generación de código.

Soporte de Google. Cuando definimos qué es Angular también destacamos que lo mantiene Google, por lo que puedes estar seguro de que recibirás soporte en tiempo y forma a cualquier duda o problema que presentes.

4. Aplicaciones web más ligeras: Versiones anteriores de Angular (las cuales te contaremos de

ellas más adelante) presentaban problemas en el tamaño de las aplicaciones, lo que no permitía una carga rápida. Actualmente, se han hecho mejoras en los módulos de carga diferida y con el renderizador de Ivy, que permite hacer paquetes más pequeños para acelerar la aplicación.

5. Código legible y comprobable: Podemos resaltar que Angular es un marco de trabajo consistente gracias a sus elementos estructurales como módulos, componentes, directivas, tuberías y servicios; así como la posibilidad de escribir aplicaciones con la arquitectura tradicional MVVM (Model-View-ViewModel) y MVC (Model-View-Controller). Ambas sirven para el mejoramiento de reutilización de código

### 1.3.3.8. Back-end

Nest es un marco para crear aplicaciones del lado del servidor Node.js eficientes y escalables. Utiliza JavaScript moderno, está construido con TypeScript (preserva la compatibilidad con JavaScript puro) y combina elementos de OOP (Programación orientada a objetos), FP (Programación funcional) y FRP (Programación reactiva funcional).

Debajo del capó, Nest utiliza Express, pero también proporciona compatibilidad con una amplia gama de otras bibliotecas, como por ejemplo Fastify, lo que permite un uso fácil de la mirada de complementos de terceros que están disponibles (NestJS, 2022)

**Algunas de las ventajas de utilizar Nest.js son** (Kinsta, 2022)

**Potente pero fácil de usar:** El framework es fácil de usar para los desarrolladores, lo suficiente como para que incluso las funciones más complejas y potentes sean fáciles de usar. El equipo de Nest.js ha diseñado el framework para que los desarrolladores puedan empezar a trabajar rápidamente y centrarse únicamente en escribir la lógica de negocio mientras el framework se encarga de otros aspectos importantes del desarrollo, como la seguridad.

**Sintaxis de estilo Angular (backend):** Angular es un framework frontend muy popular que se centra en la arquitectura y la estructuración. Nest.js actúa como Angular para el backend porque utiliza el estilo y la sintaxis de Angular para ayudarte a estructurar tu proyecto empresarial.

**TypeScript:** Nest.js es compatible con TypeScript desde el principio, y esto resuelve el rendimiento y la escritura de aplicaciones mantenibles rápidamente al proporcionar errores de

compilación y advertencias. TypeScript está bien integrado en VSCode para un entorno de desarrollo accesible.

**Documentación exhaustiva:** Nest.js cuenta con una de las mejores documentaciones para cualquier framework, que es muy fácil de entender. Se ahorra tiempo de depuración al ojear rápidamente la documentación y encontrar la solución al problema.

**Buena arquitectura y desarrollo rápido:** Nest.js ahorra tiempo a la hora de construir tu aplicación, tanto si se está creando tu primer MVP como la aplicación real, porque te da una estructura y arquitectura sólidas con las que trabajar desde el principio, mejorando así tus procesos de desarrollo

## 1.4. Criterios

Estos criterios fueron tomados de una investigación previa en el departamento con el objetivo de utilizarlos para comparar la metodología en estudio con las metodologías usadas para estos tipos de casos de estudio, RUP y XP, con el objetivo de que a partir de los resultados obtenidos determinar sus fortalezas y debilidades.

**Criterios** (Castillo Oporto y Figueredo, 2021)

- 1-Estructura iterativa e incremental.
- 2-Procesos ágiles.
- 3-Estudio del caso y obtención de datos del sistema heredado.
- 4-Definición de módulos que integran el sistema a migrar.
- 5-Proceso de selección de módulo a migrar.
- 6-Análisis de procesos de negocio y proceso de comprensión de funciones.
- 7-Modelado, diseño de interfaces y generación de prototipos funcionales.
- 8-Ambiente de pruebas y generación de resultados.
- 9-Implementación e interoperabilidad.
- 10-Continuidad de la migración y generación de documentación.

11-Ciclo de Vida

12-Enfoque de la migración

### **1.5. Conclusiones parciales**

Al finalizar el presente capítulo se arribó a las siguientes conclusiones parciales:

El análisis y la fundamentación teórica de los principales conceptos asociados a las metodologías de migración de datos, permitieron lograr una mayor comprensión del alcance de este proyecto. Se caracterizó la metodología Practical Data Migration analizando cada una de sus fases, se realizaron comparaciones entre las herramientas, tecnologías y lenguajes de programación del sistema legado con respecto al de destino obteniéndose como resultado que este proyecto se hará utilizando JavaScript , Node.js y MongoDB, también se analizaron los factores de comparación que permitirán determinar las fortalezas y debilidades de la metodología.

## **CAPÍTULO 2**

# **Aplicación de la metodología Practical Data Migration en el Sistema PESCASPIR y determinación de fortalezas y debilidades**

En el presente capítulo se realiza la descripción general del proyecto, se determinan la gestión de datos clave de las partes interesadas, se realiza un perfilamiento de datos a la base de datos legada y un mapeo de datos entre la base de datos legada y la de destino donde se compararán , eliminarán, editarán y crearán nuevas colecciones y conjuntamente de determinan las reglas de calidad de datos, luego se lleva a cabo el diseño físico, pruebas y ejecución de la migración, también se procede a la eliminación física o lógica de bases de datos legadas, hardware y software, después se realiza la comparación de las metodologías RUP, XP y Practical Data Migration, atendiendo a los criterios de comparación propuestos para determinar fortalezas y debilidades.

### 2.1. Descripción general del proyecto

En el año 2020 la Empresa PESCASPIR realiza un contrato con la Universidad de Sancti Spíritus José Martí Pérez, el cuál consistía en la realización de una página web que gestionara todo lo referente al proceso de promoción de esta empresa.

El proceso de promoción, antes de la realización de la página web, de la Empresa Pesquera de Sancti Spíritus, se basaba en la oferta de productos para la venta. Actualmente existen dos vías de venta: en pescaderías y mediante las ventas online. Por lo que se divide en dos subprocesos:

•**Venta en Pescadería:** La comercializadora (UEB COMESPIR) elabora la planificación de distribución diaria para cada pescadería teniendo en cuenta el inventario existente en la industria. Ya terminada se procede a la distribución y se realiza la oferta en cada una de las pescaderías.

•**Venta Online:** Esta venta se realiza mediante dos páginas (Bazar Regalo y Katapulck). La comercializadora (UEB COMESPIR) define el inventario a ofertar, luego se envía a CITMATEL quien se encarga de ofertarlos en cada una de las páginas.

Luego de la realización de la página web los clientes podían entrar a la página y ver los productos que se estaban produciendo y las diferentes pescaderías que los vendían.

Actualmente la empresa necesita que se realice una actualización que permita aún más la interacción con el cliente.

### 2.2. Alcance del proyecto de migración de datos

Este proyecto tiene como objetivo la migración de la base de datos del sistema PESCASPIR haciendo uso de la metodología Practical Data Migration.

Actualmente la base de datos funciona bajo un marco relacional con MySQL y se quiere migrar hacia una base de datos no relacional en MongoDB.



### 2.3. Presupuesto del proyecto de migración de datos

El presupuesto para la realización de este proyecto es de 195 000 cup.

### 2.4. Organización formal de proyectos y programas

El departamento de Ingeniería Informática será el Product Owner encargado de transmitir lo que realmente quiere la empresa Pesquera de Sancti Spíritus por orden de prioridad.

### 2.5. Los módulos de la metodología Practical Data Migration

#### 2.5.1. Gestión de datos clave de las partes interesadas

Una parte interesada en los datos es cualquier persona dentro o fuera de una organización que tiene un interés legítimo en los resultados de la migración de datos. (Morrays, 2012)

**Product Owner:** Departamento de Informática.

**Administrador:** Es el trabajador de la empresa que utilizará el software.

**Cliente:** Es una persona que podrá visualizar cierta información en la página web.

#### 2.5.2. Análisis del paisaje

En este módulo se realiza un perfilado de datos a la base de datos legada. La Figura 2.1 muestra el perfilado de datos de la tabla pescadería, los demás análisis se encuentran en el Anexo 1

## 2.5 Los módulos de la metodología Practical Data Migration

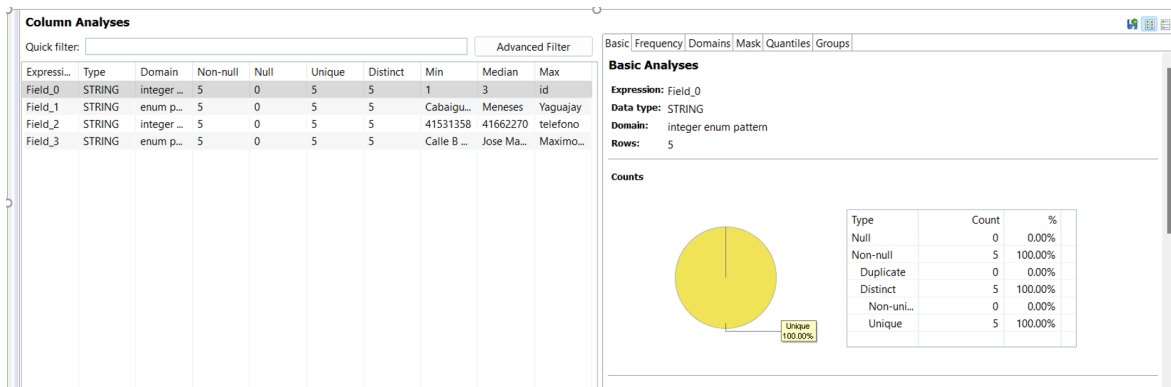


Figura 2.1: Perfilado de datos de la tabla pescadería

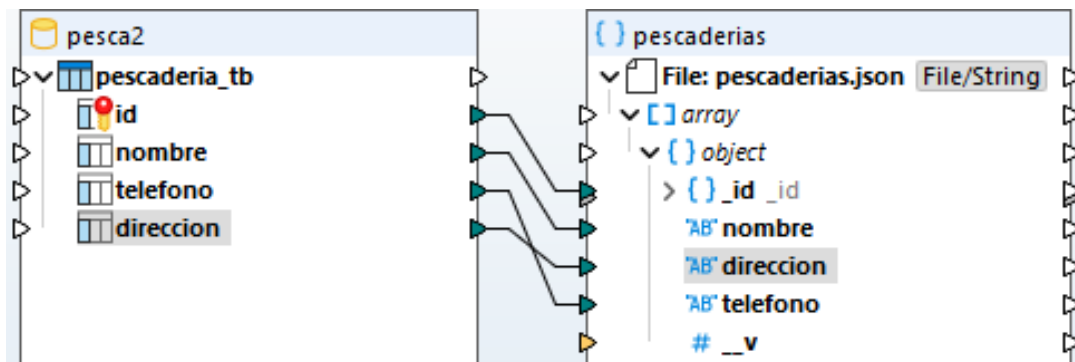


Figura 2.2: Mapeo de datos de pescadería

### 2.5.3. Análisis de brechas y mapeo

En este módulo se procede a la eliminación de los datos no necesarios, la actualización de algunos y la creación de nuevos. Se realiza una comparación entre la base de datos legada y la de destino. La Figura 2.2 muestra el mapeo de datos de la colección pescadería, los demás mapeos se encuentran en el Anexo 2B.1

#### Colección Pescadería

La entidad pescadería se convierte en colección sin sufrir cambios.

### 2.5.4. Reglas de calidad de datos (DQR)

## 2.5 Los módulos de la metodología Practical Data Migration

Tabla 2.1: Regla de calidad de datos

No	Nombre de Colección	Nombre del campo	Tipo de campo	Descripción	Validación
RCD1	Usuario	Nombre	String	Campo Alfabético	Es obligatorio
		Usuario	String	Campo Alfanumérico	Es obligatorio y debe ser único
		Password	String	Campo Alfanumérico	Es obligatorio
		Imagen	String	Se insertará la imagen que el usuario desee	Es obligatorio
		Email	String	Campo Alfanumérico	Es obligatorio y solo aceptará direcciones de correo electrónico
		Teléfono	Number	Campo Numérico	Es obligatorio
		Dirección	String	Campo Alfanumérico	Es obligatorio
RCD2	Producto	Nombre	String	Campo Alfanumérico	Es obligatorio
		Imagen	String	Se insertará la imagen que el usuario desee	
		Información Nutricional	String	Campo Alfanumérico	Es obligatorio

## 2.5 Los módulos de la metodología Practical Data Migration

Tabla 2.1: Regla de calidad de datos

No	Nombre de Colección	Nombre del campo	Tipo de campo	Descripción	Validación
		Calorías	Number	Campo Numérico	Es obligatorio
		Precio	Number	Campo Numérico	Es obligatorio
RCD3	Pescadería	Nombre	String	Campo Alfanumérico	Es obligatorio
		Teléfono	Number	Campo Numérico	Es obligatorio
		Dirección	String	Campo Alfanumérico	Es obligatorio
RCD4	Receta	Nombre	String	Campo Alfabético	Es obligatorio
		Descripción	String	Campo Alfanumérico	Es obligatorio
		Categoría	String	Campo Alfabético	Es obligatorio
		Imagen	String	Se insertará la imagen que el usuario desee	Es obligatorio
RCD5	Personal	Nombre	String	Campo Alfabético	Es obligatorio
		Cargo	String	Campo Alfabético	Es obligatorio
		Email	String	Campo Alfabético	Es obligatorio
		Imagen	String	Se insertará la imagen que el usuario desee	Es obligatorio
		Teléfono	Number	Campo Numérico	Es obligatorio
RCD6	Servicios	Nombre	String	Campo Alfabético	Es obligatorio
		Categoría	String	Campo Alfabético	Es obligatorio

## 2.5 Los módulos de la metodología Practical Data Migration

Tabla 2.1: Regla de calidad de datos

No	Nombre de Colección	Nombre del campo	Tipo de campo	Descripción	Validación
		Descripción	String	Campo Alfanumérico	Es obligatorio
RCD7	Quienes Somos	Misión	String	Campo Alfanumérico	Es obligatorio
		Visión	String	Campo Alfanumérico	Es obligatorio
		Valor	String	Campo Alfanumérico	Es obligatorio
		Historia	String	Campo Alfanumérico	Es obligatorio
		Correo	String	Campo Alfanumérico	Es obligatorio y solo aceptará direcciones de correo electrónico
RCD8	Embalses	Nombre	String	Campo Alfabético	Es obligatorio
		Descripción	String	Campo Alfanumérico	Es obligatorio
		Imagen	String	Se insertará la imagen que el usuario desee	Es obligatorio
RCD9	Especies	Nombre	String	Campo Alfabético	Es obligatorio
		Descripción	String	Campo Alfanumérico	Es obligatorio

## 2.5 Los módulos de la metodología Practical Data Migration

Tabla 2.1: Regla de calidad de datos

No	Nombre de Colección	Nombre del campo	Tipo de campo	Descripción	Validación
		Imagen	String	Se insertará la imagen que el usuario desee	Es obligatorio
RCD10	Preguntas Frecuentes	Pregunta	String	Campo Alfanumérico	Es obligatorio
		Respuesta	String	Campo Alfanumérico	Es obligatorio

### 2.5.5. Diseño y ejecución de la migración

Es donde el diseño físico, pruebas y ejecución de la migración se llevan a cabo.

#### 2.5.5.1. Diseño Físico

Para el Sistema PESCASPIR se creó una base de datos llamada pesca la cuál posee 10 colecciones:

Colección ConsejoDir (en esta colección se almacenan todos los documentos relacionados con el personal del consejo de dirección)

Colección Pescadería (en esta colección se almacenan todos los documentos relacionados con las pescaderías)

Colección Usuarios (en esta colección se almacenan todos los documentos relacionados con los usuarios que utilizarán este sistema)

Colección Pregunta (en esta colección se almacenan todos los documentos relacionados con las preguntas frecuentes)

## 2.5 Los módulos de la metodología Practical Data Migration

---

Colección Producto (en esta colección se almacenan todos los documentos relacionados con los productos que allí se ofertan)

Colección QuienesS (en esta colección se almacenan todos los documentos relacionados con los datos de la empresa)

Colección Receta (en esta colección se almacenan todos los documentos relacionados con las recetas)

Colección Servicios (en esta colección se almacenan todos los documentos relacionados con los servicios que realiza la empresa)

Colección Especies (en esta colección se almacenan todos los documentos relacionados con las especies que allí se cultivan)

Colección Embalses (en esta colección se almacenan todos los documentos relacionados con los embalses)

A continuación, se mostrará el esquema de cada colección:

### **Colección Pescadería:**

```
export type PescaderiaDocument = Pescaderia & Document;
```

```
@Schema()
```

```
export class Pescaderia {
```

```
  @Prop()
```

```
  nombre: string;
```

```
  @Prop()
```

```
  direccion: string;
```

```
  @Prop()
```

```
  telefono: string;}
```

```
export const PescaderiaSchema = SchemaFactory.createClass(Pescaderia);
```

Los demás diseños físicos se encuentran en el Anexo 2B.2

## 2.5 Los módulos de la metodología Practical Data Migration

Tabla 2.2: Tabla de equivalencias

Código de entrada	Tipo	Clases Válidas	Clases no válidas	Regla
•Nombre	Valor	A-Z , a-z 0-9 (1)	Otro(2)	Charfield
•Dirección	Valor	A-Z , a-z, 0-9 (3)	Otro(4)	Charfield
•Teléfono	Valor	0-9(5)	Otro(6)	Charfield

### 2.5.5.2. Pruebas

Para evaluar la calidad del sistema que se está desarrollando y verificar el cumplimiento de los objetivos trazados, se aplicó la prueba de aceptación del usuario usando el método de la caja negra definida por la metodología.

La Interfaz “Pescadería” tomó como entradas a validar las siguientes:

- Nombre
- Dirección
- Teléfono

Donde:

- Nombre: es un campo alfanumérico
- Dirección: es un campo alfanumérico
- Teléfono: es un campo numérico

### 2.5.6. Desmantelamiento heredado

Cubre la eliminación física o lógica de bases de datos legadas, hardware y software.



Tabla 2.3: Casos Válidos

No Caso	Clase de equivalencia	Nombre	Dirección	Teléfono
1	1,3,5	Trinidad	Jesús Menéndez # 204	41998542
2	1,3,5	Cabaiguán1	José Martí #586	41665821

Tabla 2.4: Casos no válidos

No Caso	Clase de equivalencia	Nombre	Dirección	Teléfono
1	2,4,6	Jatibonico5	Pto Guinart #256	476223gg
2	2,4,6	Trini7//	Calle B #258	475336vhjv

## 2.6. Políticas

Las políticas son los impulsores subyacentes explícitos o tácitos para un proyecto y para el entorno circundante en el que opera un proyecto (Morrays, 2012)

### 2.6.1. Metodología del proyecto

Scrum es una metodología de desarrollo ágil utilizada en el desarrollo de Software basada en un proceso iterativo e incremental. Scrum es un marco ágil adaptable, rápido, flexible y eficaz que está diseñado para ofrecer valor al cliente durante todo el desarrollo del proyecto. El objetivo principal de Scrum es satisfacer la necesidad del cliente a través de un entorno de transparencia en la comunicación, responsabilidad colectiva y progreso continuo. El desarrollo parte de una idea general de lo que hay que construir, elaborando una lista de características ordenadas por prioridad (product backlog) que el propietario del producto quiere obtener. (Digite, 2021)

### 2.6.1.1. Requisitos Funcionales

Requerimientos funcionales son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema.

Los Requerimientos funcionales del sitio web para la promoción de los servicios, actividades comerciales y productos de la Empresa Pesquera de Sancti Spíritus se muestran en la siguiente Tabla 2.5:

Tabla 2.5: Requisitos funcionales

<b>Agrupación Funcional</b>	<b>No</b>	<b>Nombre</b>	<b>Complejidad</b>	<b>Prioridad</b>	<b>Iteración</b>
Usuario	RF1	Crear Usuario.	Alta	Alta	1
	RF2	Cambiar Usuario.	Alta	Alta	1
	RF3	Borrar Usuario.	Alta	Alta	1
	RF4	Enseñar Usuario.	Alta	Alta	1
Producto	RF5	Crear Producto.	Alta	Alta	1
	RF6	Cambiar Producto.	Alta	Alta	1
	RF7	Borrar Producto.	Alta	Alta	1
	RF8	Enseñar Producto.	Alta	Alta	1

Tabla 2.5: Requisitos funcionales

<b>Agrupación Funcional</b>	<b>No</b>	<b>Nombre</b>	<b>Complejidad</b>	<b>Prioridad</b>	<b>Iteración</b>
Pescadería.	RF9	Crear Pescadería.	Alta	Alta	1
	RF10	Cambiar Pescadería.	Alta	Alta	1
	RF11	Borrar Pescadería.	Alta	Alta	1
	RF12	Enseñar Pescadería.	Alta	Alta	1
Receta.	RF13	Crear Receta.	Alta	Alta	2
	RF14	Cambiar Receta.	Alta	Alta	2
	RF15	Borrar Receta.	Alta	Alta	2
	RF16	Enseñar Receta.	Alta	Alta	2
Personal	RF17	Crear Personal.	Alta	Alta	2
	RF18	Cambiar Personal.	Alta	Alta	2
	RF19	Borrar Personal.	Alta	Alta	2
	RF20	Enseñar Personal.	Alta	Alta	2

Tabla 2.5: Requisitos funcionales

<b>Agrupación Funcional</b>	<b>No</b>	<b>Nombre</b>	<b>Complejidad</b>	<b>Prioridad</b>	<b>Iteración</b>
Servicio.	RF21	Crear Servicio.	Alta	Alta	3
	RF22	Cambiar Servicio.	Alta	Alta	3
	RF23	Borrar Servicio.	Alta	Alta	3
	RF24	Enseñar Servicio.	Alta	Alta	3
Quienes Somos.	RF25	Crear Quienes Somos.	Media	Media	3
	RF26	Cambiar Quienes Somos.	Media	Media	3
	RF27	Borrar Quienes Somos.	Media	Media	3
	RF28	Enseñar Quienes Somos.	Media	Media	3
Embalse.	RF29	Crear Embalse.	Media	Media	3
	RF30	Cambiar Embalse.	Media	Media	3

Tabla 2.5: Requisitos funcionales

<b>Agrupación Funcional</b>	<b>No</b>	<b>Nombre</b>	<b>Complejidad</b>	<b>Prioridad</b>	<b>Iteración</b>
	RF31	Borrar Embalse.	Media	Media	3
	RF32	Enseñar Embalse.	Media	Media	3
Especies.	RF33	Crear Especies.	Media	Media	4
	RF34	Cambiar Especies.	Media	Media	4
	RF35	Borrar Especies.	Media	Media	4
	RF36	Enseñar Especies.	Media	Media	4
Preguntas Frecuentes	RF37	Crear Preguntas Frecuentes.	Media	Media	4
	RF38	Cambiar Preguntas Frecuentes.	Media	Media	4
	RF39	Borrar Preguntas Frecuentes.	Media	Media	4
	RF40	Enseñar Preguntas Frecuentes.	Media	Media	4
Autenticarse.	RF41	Autenticarse.	Alta	Baja	4

Tabla 2.6: RF1 Crear Usuario

<b>Descripción de requisito.</b>	
<b>Número: RF1</b>	<b>Nombre:</b> Crear Usuario.
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador.
<b>Prioridad:</b> Alta.	<b>Complejidad:</b> Alta.
<b>Descripción:</b> El requisito permite crear un usuario. El usuario selecciona en el menú lateral la opción Usuario. En el área de íconos flotantes selecciona la opción Crear Usuario. Se muestran los datos a llenar: Ver 2.1. Además, la opción: Listar en el área de íconos flotantes.	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD12.1

Tabla 2.5: Requisitos funcionales

<b>Agrupación Funcional</b>	<b>No</b>	<b>Nombre</b>	<b>Complejidad</b>	<b>Prioridad</b>	<b>Iteración</b>
Cerrar Sesión.	RF42	Cerrar Sesión.	Alta	Baja	4
Cambiar contraseña.	RF43	Cambiar contraseña.	Alta	Baja	4
Visualizar información.	RF44	Visualizar información.	Baja	Baja.	4

### 2.6.1.2. Descripción de requisitos

Con el objetivo de evitar errores y representar los requisitos funcionales de una manera consistente y detallada para el desarrollo del componente, se hace una descripción de requisitos. Para ello se establecen la prioridad, la complejidad, una breve descripción de las acciones principales que permitirán el desarrollo de la funcionalidad para su posterior implementación.

Las demás descripciones de requisitos se encuentran en el Anexo 3C, Anexo 4D y Anexo 5E

## 2.6.2. Arquitectónico

Los requisitos no funcionales del sistema a tener en cuenta en el desarrollo del sitio web se describen en la siguiente Tabla 2.7 :

Tabla 2.7: Descripción de los requisitos no funcionales

<b>Interfaz externa:</b>	
RNF1	La interfaz estará diseñada de modo tal que el usuario pueda tener en todo momento el control de la aplicación, lo que le permitirá ir de un punto a otro dentro de ella con gran facilidad. La aplicación será lo más interactiva posible, además la interfaz será uniforme logrando identificarla como un mismo sistema.
RNF2	Diseño encuadrado para la resolución 1280x1024, y preparado para verse en otras resoluciones.
RNF3	La interfaz del sistema debe ser a través de páginas web dinámicas y personalizadas de acuerdo al tipo de usuario que acceda al sistema.
<b>Usabilidad:</b>	
RNF4	El producto está orientado para ser usado por diferentes tipos de usuarios, que posean conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general, dependiendo de su funcionalidad.
<b>Rendimiento:</b>	
RNF5	Tiempo de respuesta rápido.
RNF6	Velocidad de procesamiento de la información rápida.
<b>Soporte:</b>	
RNF7	Se requiere un servidor de bases de datos con soporte para grandes volúmenes de información, velocidad de procesamiento y tiempo de respuesta rápido en accesos concurrentes. Se documentará la aplicación para garantizar su soporte.

Tabla 2.7: Descripción de los requisitos no funcionales

<b>Interfaz externa:</b>	
RNF8	El sistema estará expuesto a un período de prueba para garantizar la funcionalidad y ventajas que este ofrece.
RNF9	El sistema debe propiciar su mejoramiento y la incorporación de otras opciones.
<b>Seguridad:</b>	
RNF10	Identificar al usuario antes de que pueda realizar cualquier acción.
RNF11	En el diseño de la aplicación debe tenerse en cuenta la existencia de regulaciones y/o restricciones en la manipulación de la información.
RNF12	Además, a los usuarios autorizados se les garantizará el acceso a la información y los mecanismos utilizados para lograr la seguridad no retrasarán a dichos usuarios para obtener los datos deseados en un momento dado.
RNF13	Establecer permisos para los diferentes tipos de usuarios en y garantizar que cada usuario tenga disponibles solamente las opciones asociadas a sus permisos.
RNF14	Proteger contra acciones no autorizadas o que puedan afectar la integridad de los datos.
RNF15	Utilizar algoritmos y protocolos para la autenticación.
<b>Confiabilidad:</b>	
RNF16	La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.
RNF17	Se realizará una programación en capas en la que el servidor de datos esté separado de las aplicaciones, para garantizar la integridad de los datos.
<b>Software:</b>	



## 2.7 Descomposición del proyecto

Tabla 2.7: Descripción de los requisitos no funcionales

<b>Interfaz externa:</b>	
RNF18	Cualquier navegador web.
RNF19	Servidor Web Apache.
RNF20	Servidor de Base de Datos MongoDB.
<b>Hardware:</b>	
RNF21	Procesador PENTIUM IV
RNF22	512 Mbyte de RAM
RNF23	20 GB de HDD
RNF24	Tarjeta de red de 100 MB.
RNF25	UPS o fuente de corriente ininterrumpida.
<b>Portabilidad:</b>	
RNF26	La plataforma seleccionada para desarrollar la aplicación fue Windows, pero puede ser ejecutada desde otras plataformas que soporten el lenguaje JavaScript y MongoDB, por ejemplo, Linux.
<b>Ayuda y documentación:</b>	
RNF27	El producto debe incluir una ayuda bien detallada sobre las principales opciones del sistema para orientar al usuario en el uso del mismo.

## 2.7. Descomposición del proyecto

La elección de la descomposición del proyecto tiene una relación recíproca con la forma de migración, que se analizará a continuación.

## 2.7 Descomposición del proyecto

Tabla 2.8: Iteraciones y tiempo.

<b>Iteraciones</b>	<b>Tiempo Estimado</b>	<b>Tiempo Real</b>
Iteración 1	4 semanas	4 semanas
Iteración 2	3 semanas	2 semanas
Iteración 3	3 semanas	2 semanas
Iteración 4	3 semanas	2 semanas
Tiempo Total	13 semanas	10 semanas

### 2.7.1. Forma de la migración

La forma de migración que se usará en este trabajo es Big bang, donde todos los datos se mueven de una sola vez y el legado se retira de inmediato, pues el software actualmente no se encuentra en funcionamiento y la empresa no sufriría ninguna afectación.

### 2.7.2. Plan de migración inicial

La realización de este proyecto comenzó el 26 de julio del 2022 y su fecha de entrega es el 28 de noviembre.

El primer paso para la realización de este proyecto fue el desarrollo de la aplicación de destino, para esto se realizó una entrevista al Product Owner creando así un Product Backlog en dependencia de sus prioridades, luego de esto cada día se realizaba una reunión con el Product Owner mostrándole los avances realizados. A continuación se muestra una Tabla 2.8 con el tiempo de desarrollo de cada iteración.

Esta parte se desarrolló con éxito en 10 semanas. El segundo paso es desarrollar los módulos de la metodología Practical Data Migration para así llevar a cabo la migración de la base de datos. A continuación se muestra una Tabla 2.9 con el tiempo empleado en cada módulo.

Esta parte se desarrolló exitosamente en 4 semanas y 5 días. En total el proyecto se desarrolló en 14 semanas y 5 días.

## 2.7 Descomposición del proyecto

Tabla 2.9: Módulos y tiempo

<b>Módulos</b>	<b>Tiempo Estimado</b>	<b>Tiempo Real</b>
Análisis del Paisaje (LA)	1 semana	5 días
Análisis de brechas y mapeo (GAM)	1 semana	1 semana
Diseño y ejecución de la migración (MDE)	2 semana	1 semana
Desmantelamiento Heredado (LD)	1 semana	3 días
Reglas de calidad de datos (DQR)	0(se realizó en conjunto con los demás módulos)	0
Gestión de datos clave de las partes interesadas (KDSH)	2 días	1 día
Plan de retiro del sistema (SRP)	1 semana	3 días
Estrategia de migración y gobernanza – Migrationstrategy and governance en adelante (MSG)		
	1 semana	Tiempo Total 7 semanas y 2 días 4 semanas y 5 días

Tabla 2.10: Comparación atendiendo al ciclo de vida

Metodologías	Ciclo de Vida				
	Especificación	Análisis	Diseño	Codificación	Prueba y Mantenimiento
Practical Data Migration	X	X	X	X	X
RUP	X	X	X	X	X
XP	X	X	X	X	X

## 2.8. Criterios Comunes

En este epígrafe se abarcarán criterios que permitan la comparación de las metodologías RUP y XP, abordadas frecuentemente en el departamento para la realización de migraciones de datos, con la metodología Practical Data Migration, descrita anteriormente.

### 2.8.1. Ciclos de Vida

El ciclo de vida del software describe el desarrollo de software desde la fase inicial hasta la fase final. Se refiere al conjunto de fases por las que pasa el sistema que se está desarrollando (Castillo Oporto y Figueredo, 2021)

En la Tabla 2.10 se muestran las metodologías y se realiza la comparación atendiendo al ciclo de vida que estas poseen, para así saber su alcance a lo largo del desarrollo del proyecto.

En esta Tabla 2.10 se puede apreciar que las metodologías analizadas no realizan completamente el ciclo de vida.

### 2.8.2. Enfoque de la migración

El éxito en un proceso de migración de datos dependerá en gran medida del nivel de comprensión que se llegue a alcanzar acerca del proceso y sus implicaciones. Conocer los retos que implica una iniciativa de este tipo es el primer paso (Castillo Oporto y Figueredo, 2021)

Tabla 2.11: Comparación para el enfoque de la migración

Metodologías	Enfoque de la migración			
	Migración de Almacenamiento	Migración de Base de Datos	Migración de aplicaciones	Codificación de caracteres
Practical Data Migration	X	X	X	X
RUP		X	X	X
XP		X	X	

En la Tabla 2.11 se muestran las metodologías y se realiza la comparación atendiendo al enfoque de la migración que estas poseen, para así saber su alcance a lo largo del desarrollo del proyecto.

### 2.8.3. Comparación de Metodologías de Migración basado en los criterios establecidos anteriormente

En la Tabla 2.12 se establece la comparación de Metodologías basados en los criterios establecidos anteriormente.

Tabla 2.12: Comparación de Metodologías

<b>Actividad</b>	<b>XP</b>	<b>Practical Data Migration</b>	<b>RUP</b>
<b>Estructura iterativa e incremental</b>	Se caracteriza por ser iterativo e incremental.	Propone un enfoque integral que consta de varios módulos que cubren el alcance completo de la migración de datos, desde sus inicios hasta la desconexión de los sistemas legados.	Se caracteriza por ser iterativo e incremental.
<b>Procesos ágiles</b>	Es una metodología ágil.	No declara un proceso ágil.	No tiene un proceso ágil.
<b>Estudio del caso y obtención de datos del sistema heredado</b>	Forma parte de la fase de análisis.	Análisis del panorama – Landscape Analysis (LA): Este módulo usa varias técnicas para descubrir y catalogar almacenamientos de datos legados (legacy data stores – LDS) y su relación entre ellos. Se analizan los sistemas de almacenamiento de datos de los legados para ver cómo funcionan, qué datos tienen y qué “sorpresas” pueden contener	Forma parte de la fase de análisis.

Tabla 2.12: Comparación de Metodologías

<b>Actividad</b>	<b>XP</b>	<b>Practical Data Migration</b>	<b>RUP</b>
<b>Definición de módulos que integran el sistema a migrar</b>	No se contempla porque no está enfocada en una migración.	Análisis de brechas y mapeo – Gap analysis and mapping (GAM) Se realiza el mapeo de datos una vez que el sistema de destino está disponible. El mapeo de datos es la vinculación de campos en los legados con los campos en el destino, además definirá la lógica de transformación que se necesita para dividir datos y fusionar campos	No se contempla porque no está enfocada en una migración.
<b>Proceso de selección de módulo a migrar</b>	No se contempla porque no está enfocada en una migración.	Diseño y ejecución de migración – Migration Design and Execution (MODE)	No se contempla porque no está enfocada en una migración.

Tabla 2.12: Comparación de Metodologías

<b>Actividad</b>	<b>XP</b>	<b>Practical Data Migration</b>	<b>RUP</b>
<b>Análisis de procesos de negocio y proceso de comprensión de funciones</b>	En esta etapa, se identificarán las historias de usuario, se determinan basándose en las necesidades del cliente. Cada función o historia de usuario se divide según su prioridad y se descompone en versiones.	Desmontaje del legado – Legacy Decommissioning (LD) Cubre la eliminación física o lógica de bases de datos legadas, hardware y software. También cubre la entrega de almacenamiento de datos legados para componentes de datos que deben retenerse por algún tiempo pero que no se deben migrar al destino	Se define el alcance del proyecto con los clientes, se identifican los riesgos asociados al proyecto, se elabora el plan de las fases y el de la iteración posterior, se detalla de manera general la arquitectura del software.



Tabla 2.12: Comparación de Metodologías

Actividad	XP	Practical Data Migration	RUP
<b>Modelado, diseño de interfaces y generación de prototipos funcionales</b>	En este paso se intentará trabajar con un código sencillo, haciendo lo mínimo imprescindible para que funcione. Se obtendrá el prototipo. Además, para el diseño del software orientado a objetos, se crearán tarjetas CRC (Clase-Responsabilidad-Colaboración).	Zona desmilitarizada – Demilitarized zone (DMZ) Hace las veces de interfaz entre el trabajo del proveedor de tecnología y las responsabilidades de sus clientes. La DMZ es un componente clave de PDM2 que, en cierta medida, se definirá formalmente en el contrato con el proveedor. Sin embargo, la DMZ es más amplia que el contrato, y su definición formal ayudará a ambas partes a comprender y gestionar sus dependencias.	Se diseña la solución preliminar, se selecciona los casos de uso que permiten definir la arquitectura base del sistema y se desarrollara el primer análisis del dominio del problema.

Tabla 2.12: Comparación de Metodologías

Actividad	XP	Practical Data Migration	RUP
<b>Ambiente de pruebas y generación de resultados</b>	Se deben realizar pruebas automáticas continuamente. Al tratarse normalmente de proyectos a corto plazo, este testeo automatizado y constante es clave. Además, el propio cliente puede hacer pruebas, proponer nuevas pruebas e ir validando las mini-versiones.	Plan de retiro del sistema: el objetivo final de una migración de datos es desactivar los sistemas de legados, es por esta razón que bajo el modelo PDMv2 propone iniciar la conversación con el negocio por este tema. El modelo PDMv2 propone una serie de preguntas estructuradas para obtener una visión empresarial de la migración, buscando elementos necesarios como requisitos de auditoría de migración, requisitos de linaje de datos, requisitos de retención de datos, restricciones de migración, requisitos de prueba de aceptación del usuario, requisitos de respaldo de datos, entre otros.	Las actividades relacionadas a las pruebas se realizan en todo el ciclo de vida del proyecto, siendo esta una de las características relevantes respecto a la metodología estructurada.

Tabla 2.12: Comparación de Metodologías

Actividad	XP	Practical Data Migration	RUP
<b>Implementación e interoperabilidad</b>	En este punto, se han probado todas las historias de usuario o mini-versiones con éxito, ajustándonos a los requerimientos del cliente. Tenemos un software útil y podemos incorporarlo en el producto.	Diseño y ejecución de migración – Migration Design and Execution (MODE) Es donde el diseño físico, pruebas y ejecución de la migración se llevan a cabo. La migración de datos es más que simplemente mover bits y bytes. Se debe conocer las limitaciones de negocio, tiempos, requisitos de auditoría, linaje de datos, plan de vuelta atrás (rollback), requisitos de informes, de gestión y control, etc. MODE integra también los requisitos comerciales para la desconexión y retiro de los sistemas legados.	La función de esta fase es completar la funcionalidad del sistema, se clarifican los requisitos pendientes, se administran los cambios de acuerdo a las evaluaciones realizadas por los usuarios, y se realizan las mejoras para el proyecto.

Tabla 2.12: Comparación de Metodologías

Actividad	XP	Practical Data Migration	RUP
<b>Continuidad de la migración y generación de documentación</b>	No posee una completa documentación.	El modelo expone además la importancia de la tecnología en el desarrollo de los proyectos de migración de datos, considerando las herramientas subyacentes y de apoyo a procesos como: -la calidad de datos -el perfilamiento de datos -controladores para la migración. -el uso de software especializado para estos fines reduce el riesgo del proyecto.	Está muy bien documentado.

## 2.9. Resultados y discusión

Para ganar en entendimiento de las tablas anteriores se hará un resumen por cada una:

En el caso de la Tabla 2.10 que se hace referencia al ciclo de vida de las metodologías estudiadas se puede apreciar como las tres metodologías comienzan el ciclo en la etapa de análisis llevando al software hasta las pruebas dejando fuera la etapa de mantenimiento. Por lo que se puede arribar a la conclusión de que en forma general cumplen en grandes rangos con el ciclo de vida de un software.

Analizando la Tabla 2.11 con el fin de evaluar el enfoque de la migración se puede observar que solo la metodología Practical Data Migration cubre todo el enfoque de la migración, mientras que las otras dos metodologías no consideran la migración de almacenamiento.

El análisis de la Tabla 2.12 que aborda el tema de comparación entre las metodologías RUP, XP y Practical Data Migration basado en los criterios establecidos anteriormente se realizará por separado en cada uno de estos criterios para ganar en profundidad.

- **Estructura iterativa e incremental:** RUP y XP tienen como característica ser iterativa e incremental durante todo su desarrollo, mientras que Practical Data Migration no .
- **Procesos ágiles:** solo la metodología XP declara tener un proceder ágil, mientras que Practical Data Migration y RUP no se declaran ágiles.
- **Estudio del caso y obtención de datos del sistema heredado:** todas las metodologías lo declaran durante su fase de análisis.
- **Definición de módulos que integran el sistema a migrar:** la metodología Practical Data Migration tiene en sus fases una mayor profundización en el tema mientras que las otras dos no la contemplan porque no están enfocadas en una migración.
- **Proceso de selección de módulo a migrar:** la metodología Practical Data Migration tiene en sus fases una mayor profundización en el tema mientras que las otras dos no la contemplan porque no están enfocadas en una migración.
- **Análisis de procesos de negocio y proceso de comprensión de funciones:** a pesar de que en todos los casos toman en cuenta el proceso de negocio RUP se centra más la atención en ese aspecto y lo hace más detallado.
- **Modelado, diseño de interfaces y generación de prototipos funcionales:** a pesar de que en todos los casos se toma en cuenta este proceso, RUP y XP centran más la atención en ese aspecto y lo hacen más detallado.
- **Ambiente de pruebas y generación de resultados:** las tres metodologías transitan por el proceso de pruebas durante el ciclo de vida de las mismas siendo RUP la que más detallado y profundo lo trata.
- **Implementación e interoperabilidad:** las tres metodologías cumplen con la fase de implementación.

• **Continuidad de la migración y generación de documentación:** solo la metodología Practical Data Migration cumple con la continuidad de la migración , mientras que RUP es la que mejor documentación posee.

### 2.9.1. Determinación de fortalezas y debilidades

#### Fortalezas

1. Realiza un análisis a profundidad con vistas a una migración de datos del sistema heredado en el módulo Análisis del Panorama mientras que las metodologías RUP y XP no realizan su análisis basados en una migración sino en una reingeniería.
2. Se centra en definir y escoger los módulos que integran el sistema a migrar en los módulos de Análisis de Brechas y Mapeo y Diseño y ejecución de la migración con mayor énfasis en comparacion con la metodología RUP y XP.
3. Comienza con la fase de implementación desde edades tempranas pues permite mantener la base de datos legada y la base de datos de destino trabajando paralelamente hasta que en el módulo desmantelamiento heredado la legada es eliminada mientras que RUP tiene que esperar a que pase todo el ciclo de vida del software para ver la implementación donde pueden aparecer errores que pudieran haberse prevenido.
4. Continuidad de la migración y generación de documentación pues posee un módulo documentativo que integra todos los módulos generando un alto nivel de especificidad.

#### Debilidades

- 1.. No posee como característica ser iterativa e incremental durante todo su desarrollo a pesar de los beneficios que trae ser iterativa.
2. No declara un proceso ágil pero por sus características sí pudiera ser considerada .
3. No profundiza en el proceso de negocio en comparación con las metodologías RUP y XP que en su fase de análisis se enfocan en su proceso de negocio mientras que la metodología Practical Data Migration ] lo realiza directamente sobre el trabajo con los datos.
4. Para la parte de Sistema necesita apoyarse en otra metodología para documentar su migración.

### 2.10. Conclusiones Parciales

La confección del capítulo anterior propone las siguientes conclusiones:

A partir de la aplicación de la metodología Practical Data Migration se fueron desarrollando cada uno de los módulos que la componen , realizando un perfilamiento de datos a la base de datos legada , un mapeo de datos entre la base de datos legada y la de destino, se llevó a cabo el diseño físico y las pruebas del sistema de destino, se establecieron los principales actores del negocio y del sistema.

La metodología Scrum cubrió la parte del sistema declarando los principales requisitos funcionales y no funcionales y una descripción detallada de cada uno de ellos.

Luego de la aplicación de la metodología se procedió a realizar una comparación entre la metodología RUP , XP y la estudiada ,basados en unos criterios utilizados en una investigación previa en el departamento, con el fin de que a partir de los resultados obtenidos se procedió a determinar las fortalezas y debilidades de la metodología Practical Data Migration.

# CONCLUSIONES

Una vez finalizada la presente investigación se dió solución a la problemática planteada, contribuyendo a determinar las fortalezas y debilidades de la metodología estudiada, por lo que se arriban a las siguientes conclusiones:

Luego del estudio de los fundamentos teóricos y metodológicos enfocados en las metodologías de migración de datos, se evidenció la necesidad de conocer su manejo y uso en los procesos de migración para así evitar caer en procesos de reingeniería. Se hizo énfasis en la metodología Practical Data Migration analizando cada uno de sus módulos.

La aplicación de la metodología Practical Data Migration permitió realizar la migración de la base de datos hacia un nuevo sistema y después se procedió a realizar una comparación entre las metodologías RUP y XP , basados en unos criterios antes trabajados en el departamento, para así determinar sus fortalezas y debilidades. Siendo su mayor fortaleza que realiza un análisis a profundidad con vistas a una migración en casi todos sus módulos y su mayor debilidad es que necesita apoyarse en otra metodología para la migración del sistema.



# RECOMENDACIONES

Se recomienda realizar la evaluación de la implementación de la metodología con el fin de comparar, a partir de las fortalezas y debilidades detectadas, con la otra metodología de migración en estudio.

Además, se recomienda a partiendo del análisis anterior realizar una nueva propuesta de una metodología de migración.

# REFERENCIAS

- ahaapps (2014). 9 Best Practices and Strategies for Data Migration. Section: Salesforce.  
**URL:** <https://www.ahaapps.com/9-best-practices-and-strategies-for-data-migration/>
- Almeida, J. (2022). Node js vs. PHP: An In-depth Comparison Guide for Web Development.  
**URL:** <https://distantjob.com/blog/nodejs-vs-php/>
- AltexSoft (2020). Data Migration: Process, Types, and Golden Rules to Follow.  
**URL:** <https://www.altexsoft.com/blog/data-migration/>
- Amazon (2022). ¿Qué es JavaScript? - JavaScript explicado - AWS.  
**URL:** <https://aws.amazon.com/es/what-is/javascript/>
- Azure, M. (2022). ¿Qué es una base de datos relacional? | Microsoft Azure.  
**URL:** <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-a-relational-database/>
- Castillo Oporto, R. y Figueredo, L. (2021). *Estudio comparativo de metodologías de desarrollo software con base a la migración de sistemas heredados*, PhD thesis.
- Conceptode (2021). Lenguaje de Programación - Concepto, tipos y ejemplos.  
**URL:** <https://concepto.de/lenguaje-de-programacion/>
- Coppola, M. (2022). ¿Qué es Angular? Características y ventajas.  
**URL:** <https://blog.hubspot.es/website/que-es-angular>
- Curran, T. (2021). Data Migration: A Roadmap To Success.  
**URL:** <https://www.actian.com/blog/data-integration/a-roadmap-to-a-successful-data-migration-project/>

CursosGratis (2021). Base de datos relacional vs no relacional: el programador loco \* Aprende a Programar Gratis.

**URL:** <https://aprendiendoaprogramar.es/blog/base-de-datos-relacional-vs-no-relacional-el-programador-loco/>

Data, P. (2019). ¿Qué es un gestor de datos y para qué sirve?

**URL:** <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-es-un-gestor-de-datos-y-para-que-sirve>

Digite (2021). ¿Qué Es La Metodología Scrum? Y Gestión De Proyectos Scrum.

**URL:** <https://www.digite.com/es/agile/que-es-scrum/>

Diligent (2017). Qué es el Framework Symfony php: características y ventajas - diligent.

**URL:** <https://www.diligent.es/framework-symfony-php/>

EdPrice-MSFT (2022). Datos no relacionales y NoSQL - Azure Architecture Center.

**URL:** <https://learn.microsoft.com/es-es/azure/architecture/data-guide/big-data/non-relational-data>

ENI, E. (2022). Angular y Node.js - Optimice el desarrollo de sus aplicaciones web con una arquitectura MEAN - Presentación de Angular | Ediciones ENI.

**URL:** <https://www.ediciones-eni.com/open/mediabook.aspx?idR=f66793d58696ff7a7c64081c9f8ec013>

Escambray (2022a). Llama Díaz-Canel a potenciar aún más el rol de universidad espirituana en la implementación del sistema de gestión de gobierno basado en ciencia e innovación. Publisher: Escambray Sancti Spiritus Section: Escambray.

**URL:** <http://www.escambray.cu/2022/llama-diaz-canel-a-potenciar-aun-mas-el-rol-de-universidad-espirituana-en-la-implementacion-del-sistema-de-gestion-de-gobierno-basado-en-ciencia-e-innovacion/>

Escambray (2022b). Un nuevo cumpleaños de ciencia e innovación para la UNISS. Publisher: Escambray Sancti Spiritus Section: Escambray.

**URL:** <http://www.escambray.cu/2022/un-nuevo-cumpleanos-de-ciencia-e-innovacion-para-la-uniss/>

Expósito Yanes, R. (2020). *Migración del Sistema Informático para la Planificación de Proyectos Eléctricos de la UNE*, PhD thesis.

Fatima, N. (2022). Migración de base de datos: ¿Cómo hacer una migración de base de datos.

freeCodeCamp (2020). PHP vs JavaScript: How to Choose the Best Language for Your Project.

**URL:** <https://www.freecodecamp.org/news/php-vs-javascript-which-technology-will-suit-your-business-better/>

freeCodeCamp (2021). Ventajas y Desventajas de JavaScript.

**URL:** <https://www.freecodecamp.org/espanol/news/ventajas-y-desventajas-de-javascript/>

Hofer, C. M. (2016). Data migration to oracle, p. 110.

Kinsta (2022). ¿Qué es Nest.js? Un Vistazo al Framework Ligero de JavaScript.

**URL:** <https://kinsta.com/es/base-de-conocimiento/nestjs/>

Meneses, Y. P. (2021). De la informatización de la sociedad a la transformación digital en Cuba. Section: Noticias.

**URL:** <https://www.presidencia.gob.cu/es/noticias/de-la-informatizacion-de-la-sociedad-a-la-transformacion-digital-en-cuba/>

Microsoft (2022). What is Data Migration? Migrating Data to the Cloud | Microsoft Azure.

**URL:** <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-data-migration/>

MongoDB (2022). MongoDB Vs MySQL.

**URL:** <https://www.mongodb.com/es/compare/mongodb-mysql>

Morrys, J. (2012). *Practical Data Migration*.

Naeem, T. (2019). Data Migration Software - The Why, The What, and The How.

**URL:** <https://www.astera.com/type/blog/data-migration-software/>

NestJS (2022). NestJS - A progressive Node.js framework.

**URL:** <https://nestjs.com/>

Netapp (2021). Data migration methodology.

**URL:** [https://docs.netapp.com/us-en/ontap-fli/san-migration/concept\\_data\\_migration\\_methodology.html](https://docs.netapp.com/us-en/ontap-fli/san-migration/concept_data_migration_methodology.html)

Node.js (2022). Acerca.

**URL:** <https://nodejs.org/es/about/>

OpenWebinars (2019). Qué es MySQL: Características y ventajas.

**URL:** <https://openwebinars.net/blog/que-es-mysql/>

Pérez Bravo, D. (2010). *Sistema de explotación de los medios técnicos de transporte en el Ministerio del Interior de Sancti Spíritus*, PhD thesis.

PHP (2022). PHP: ¿Qué es PHP? - Manual.

**URL:** <https://www.php.net/manual/es/intro-what-is.php>

Platzi (2018). Beneficios de utilizar Node.js.

**URL:** <https://platzi.com/blog/beneficios-de-node/>

PowerData (2018). ¿Qué es la migración de datos?

**URL:** <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/397387/qu-es-la-migracion-de-datos>

PowerData, G. (s.f.). Migración de datos: definición, desafíos y mejores prácticas para afrontarla.

**URL:** <https://www.powerdata.es/migracion-de-datos>

Santander, B. (2020). Metodologías de desarrollo de software: ¿qué son?

**URL:** <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>

Software, E. (2022). Tipos de migración de datos - Evaluando Software.

**URL:** <https://www.evaluandosoftware.com/tipos-migracion-datos/>

Support, M. (2022). Database basics - Microsoft Support.

**URL:** <https://support.microsoft.com/en-au/office/database-basics-a849ac16-07c7-4a31-9948-3c8c94a7c204>

Symfony (2022). ¿Qué es Symfony?

**URL:** <https://symfony.es/pagina/que-es-symfony/>

Ubuntu (2022). What is MongoDB and why use it for modern web applications?

**URL:** <https://ubuntu.com/blog/what-is-mongodb>

## REFERENCIAS

---

Yero Grillo, Y. (2020). *Sistema Informático para la Gestión de la Información del Módulo Transformadores del SIGERE*, PhD thesis.

# ANEXO A

## Análisis del Panorama

### A.1. Análisis del Panorama

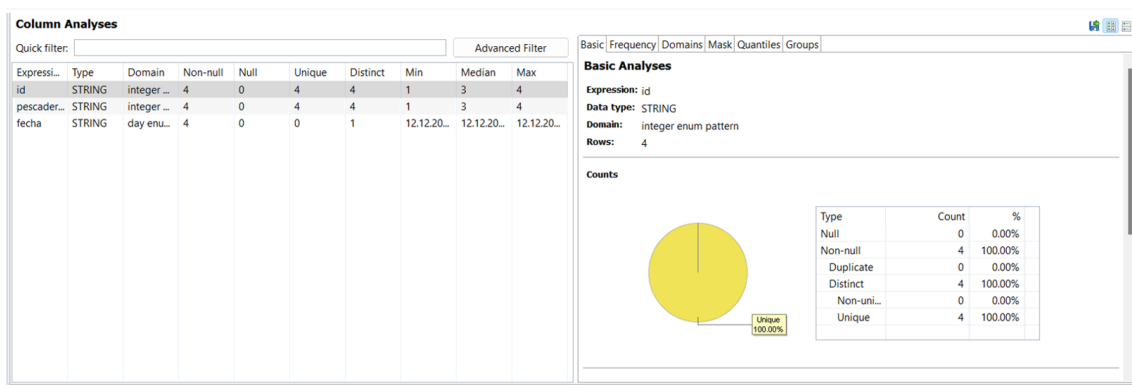


Figura A.1: Perfilado de datos de la tabla oferta

## A.1 Análisis del Panorama

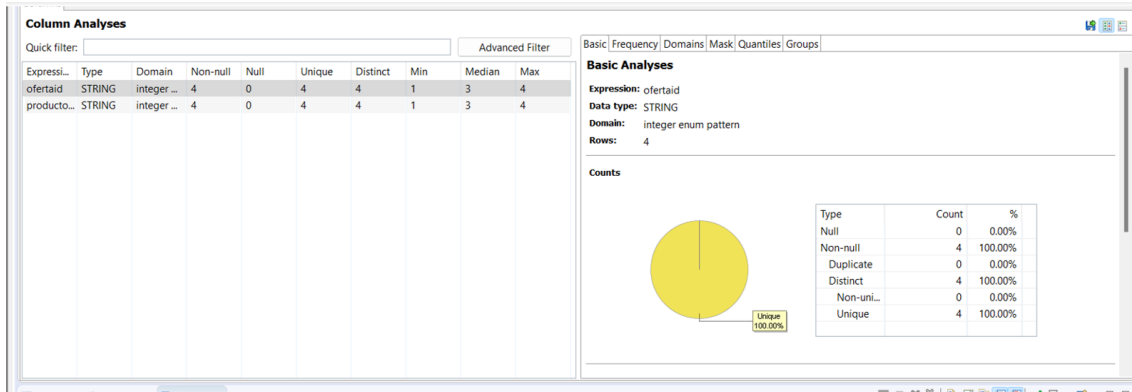


Figura A.2: Perfilado de datos de la tabla oferta\_pescadería\_producto

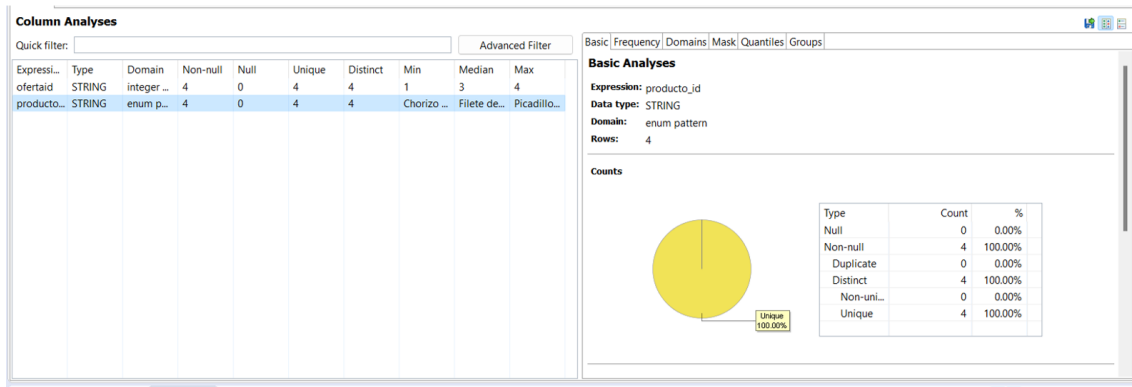


Figura A.3: Perfilado de datos de la tabla producto

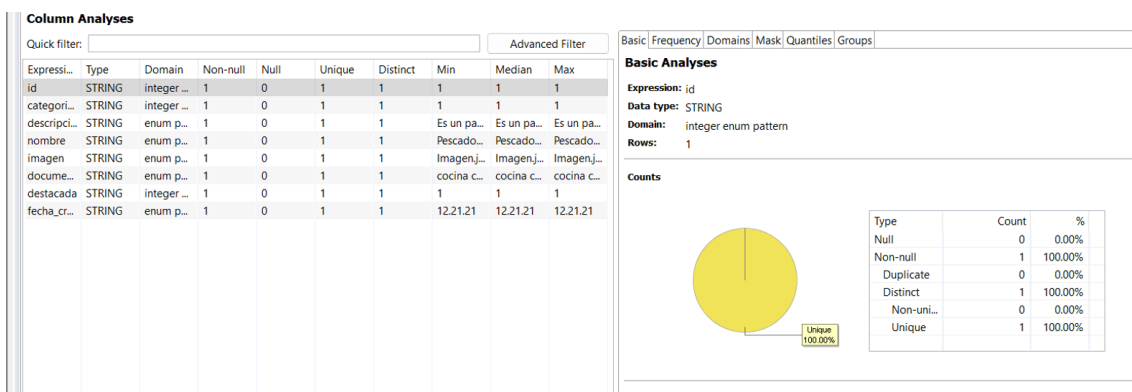


Figura A.4: Perfilado de datos de la tabla receta



## A.1 Análisis del Panorama

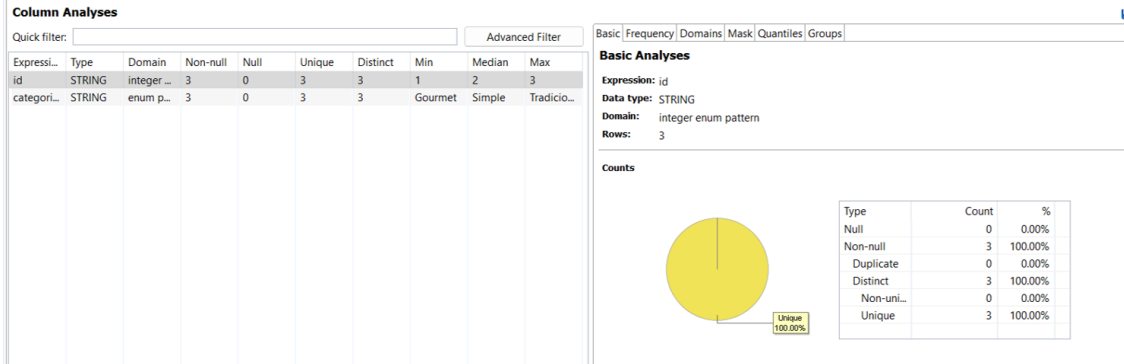


Figura A.5: Perfilado de datos de la tabla categoría de receta

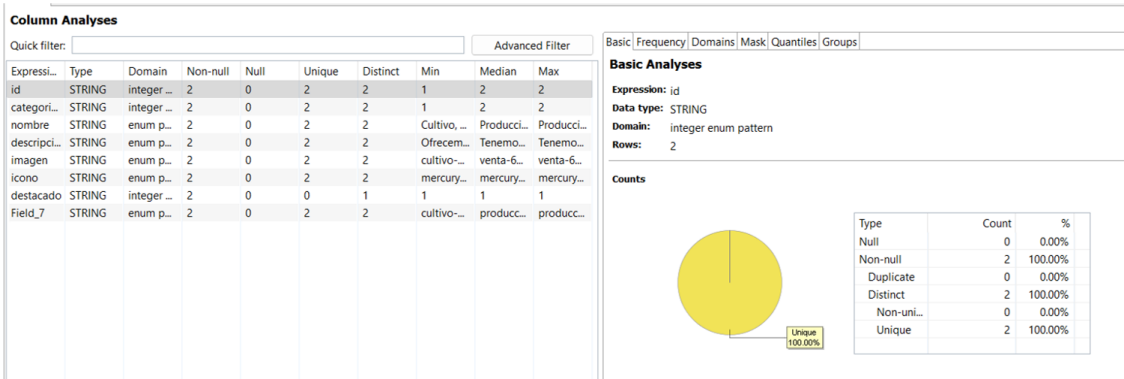


Figura A.6: Perfilado de datos de la tabla servicio

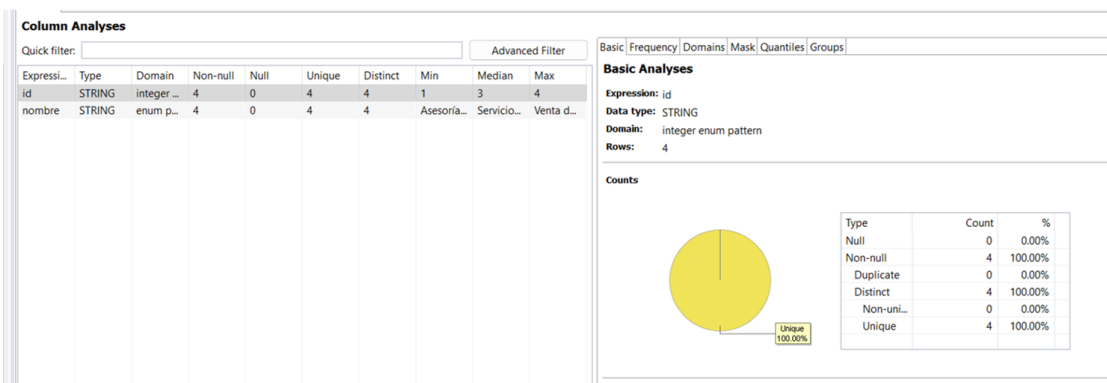


Figura A.7: Perfilado de datos de la tabla categoría de servicio

## A.1 Análisis del Panorama

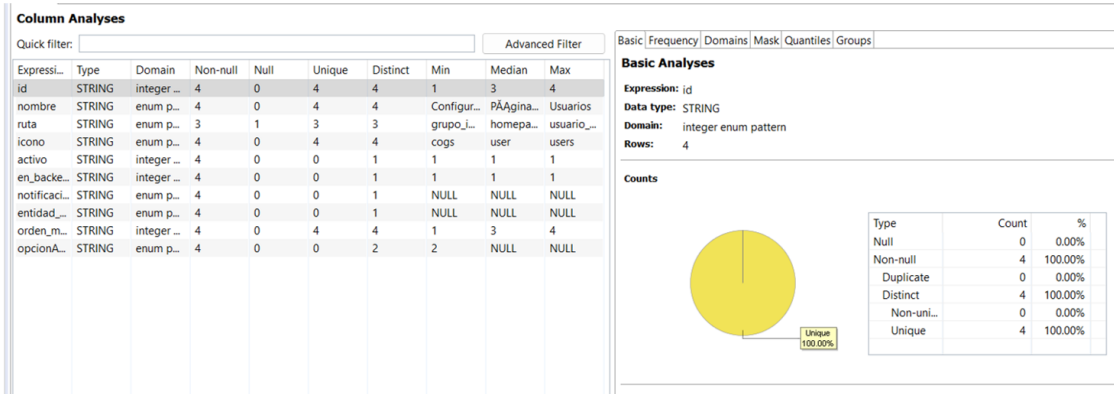


Figura A.8: Perfilado de datos de la tabla opción

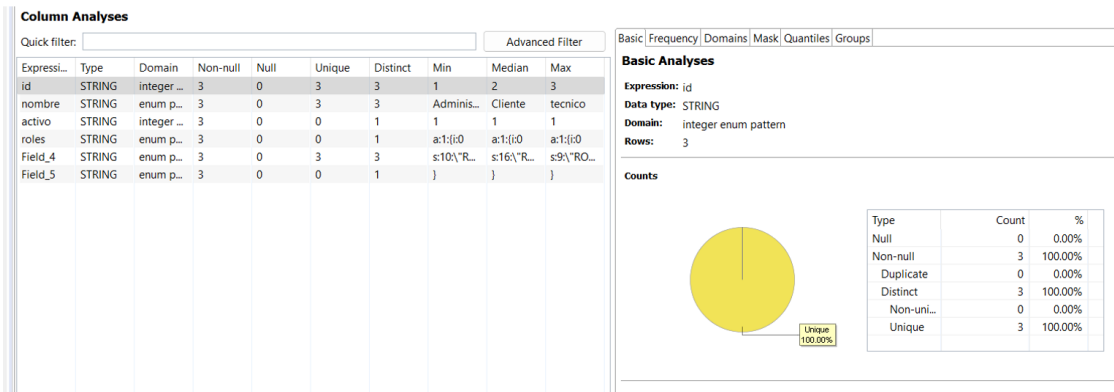


Figura A.9: Perfilado de datos de la tabla grupo

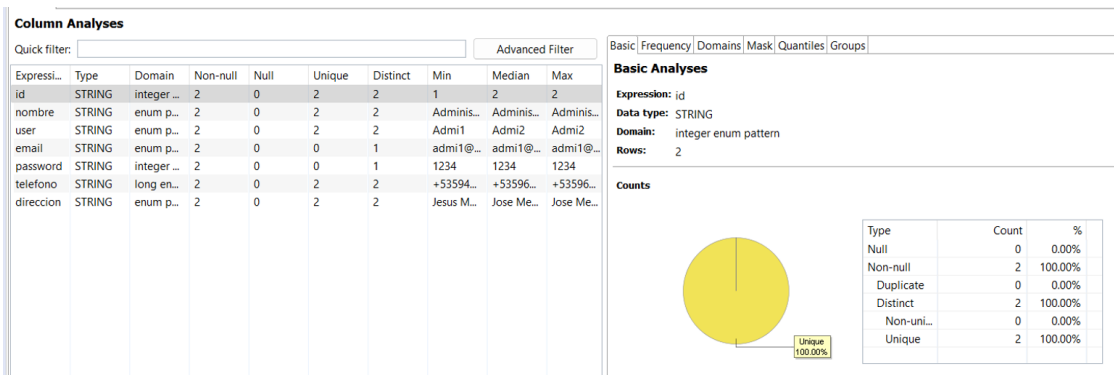


Figura A.10: Perfilado de datos de la tabla usuario

## A.1 Análisis del Panorama

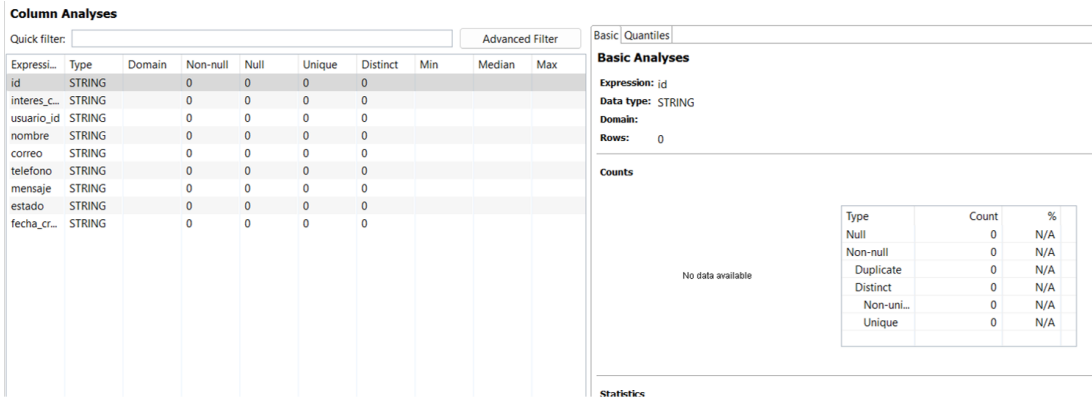


Figura A.11: Perfilado de datos de la tabla contacto

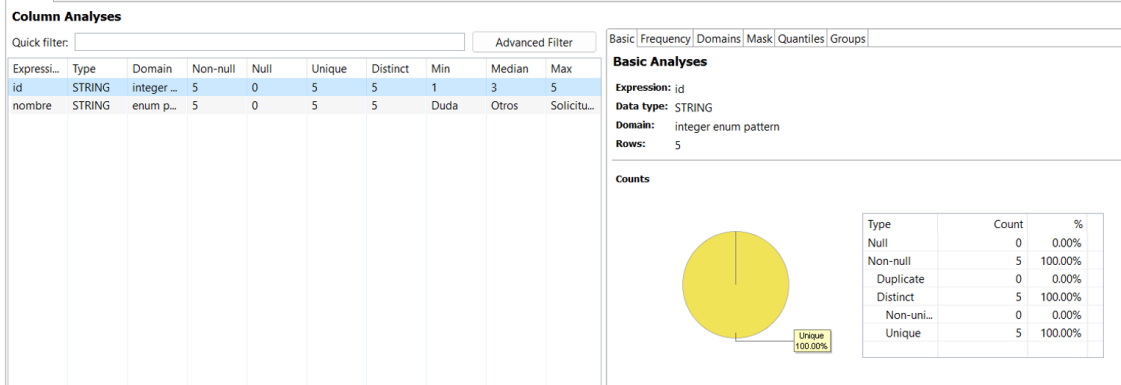


Figura A.12: Perfilado de datos de la tabla contacto\_interés

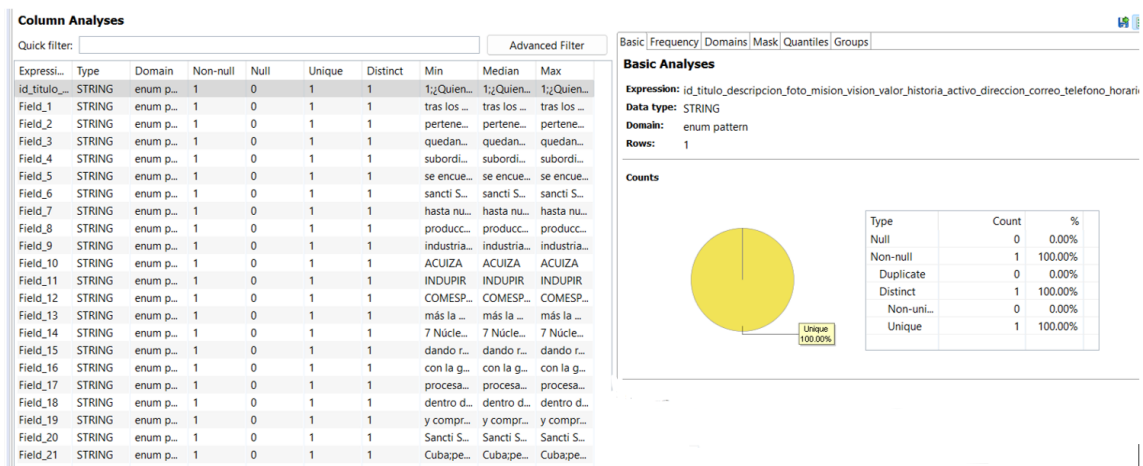


Figura A.13: Perfilado de datos de la tabla sobre\_nosotros

## A.1 Análisis del Panorama

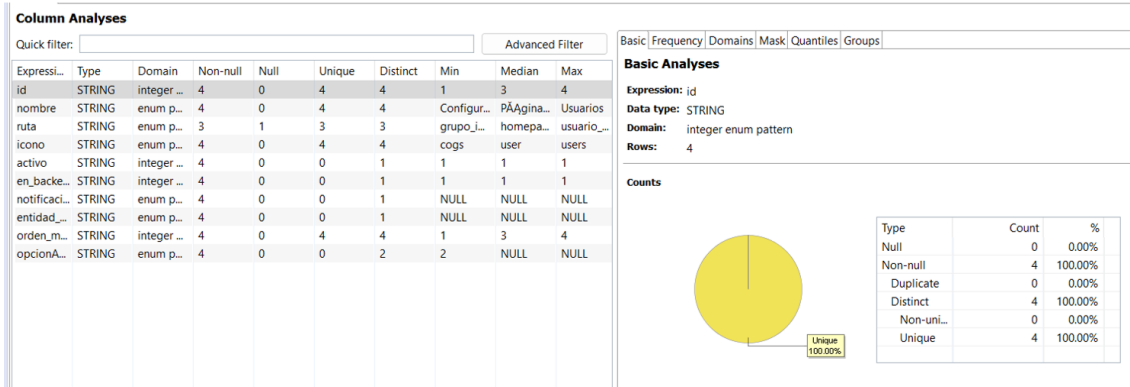


Figura A.14: Perfilado de datos de la tabla publicidad

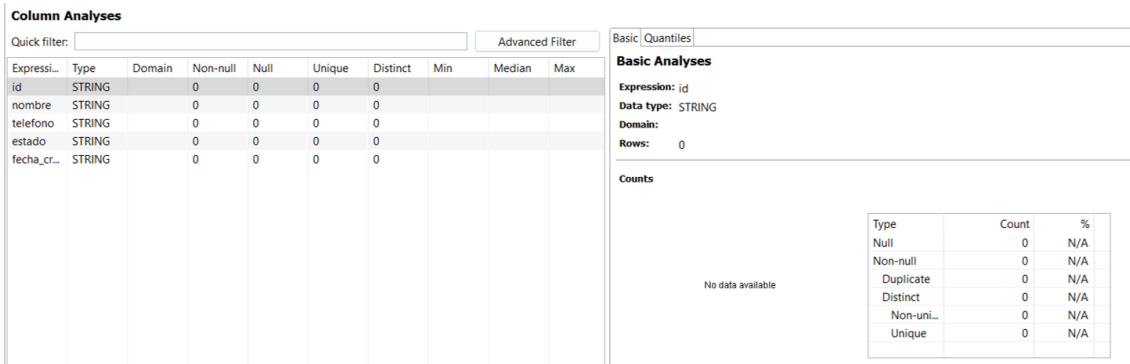


Figura A.15: Perfilado de datos de la tabla petición\_contacto

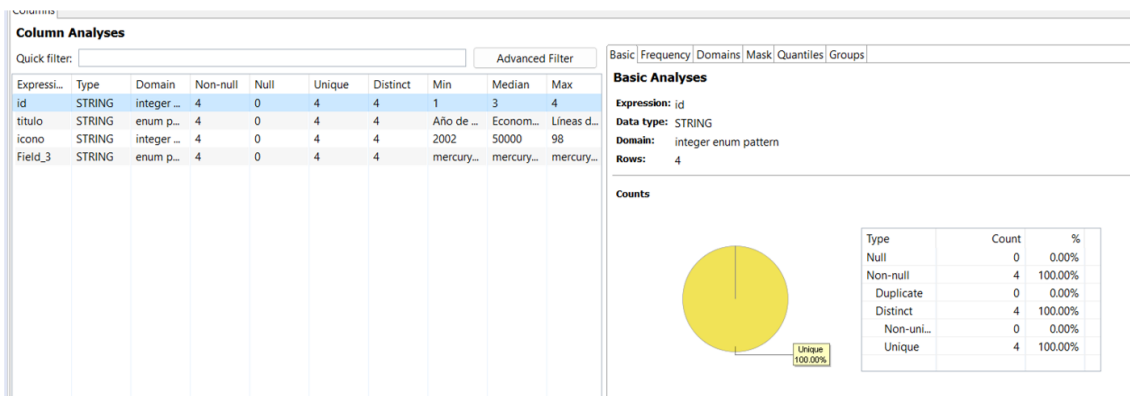


Figura A.16: Perfilado de datos de la tabla información\_complementaria

## A.1 Análisis del Panorama

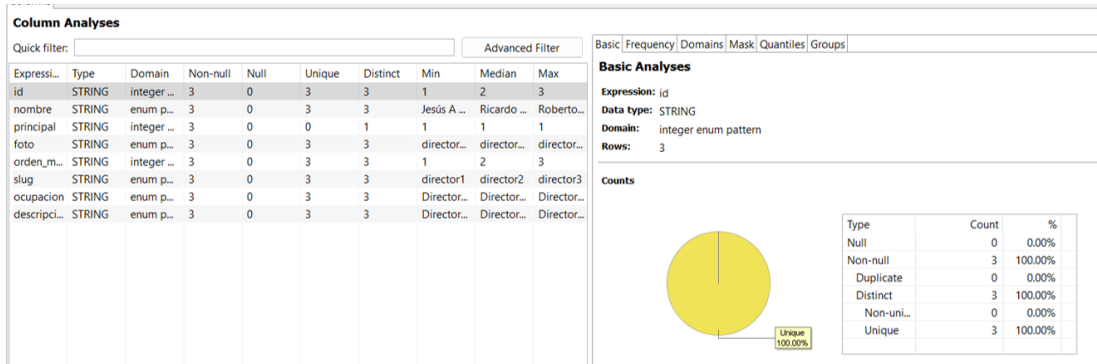


Figura A.17: Perfilado de datos de la tabla personal

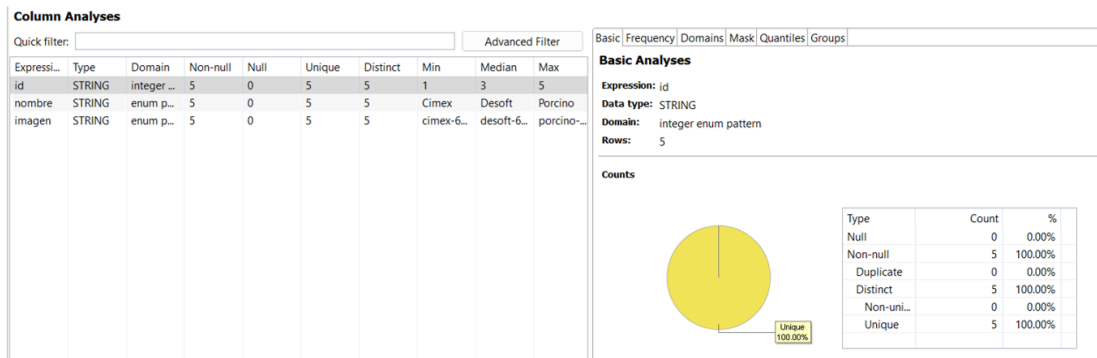


Figura A.18: Perfilado de datos de la tabla cliente

# **ANEXO B**

## **Análisis de Brechas y Mapeo y Diseño Físico**

### **B.1. Brechas y mapeo**

#### **Colección Producto**

La entidad producto se convierte en colección y se le agregan dos campos nuevos, calorías y precio, para aumentar el nivel de detalle.

#### **Colección Usuario**

La entidad usuario se convierte en colección con algunos cambios pues algunos de sus atributos son eliminados pues son considerados innecesarios.

#### **Colección QuienesS**

La entidad sobre nosotros se convierte en colección con el nombre de quienes somos y también se le realizan algunos cambios pues algunos de sus atributos son eliminados porque son considerados innecesarios.

#### **Colección ConsejoDir**

La entidad personal se convierte en colección con el nombre de ConsejoDir y también se le realizan algunos cambios pues algunos de sus atributos son eliminados porque son considerados

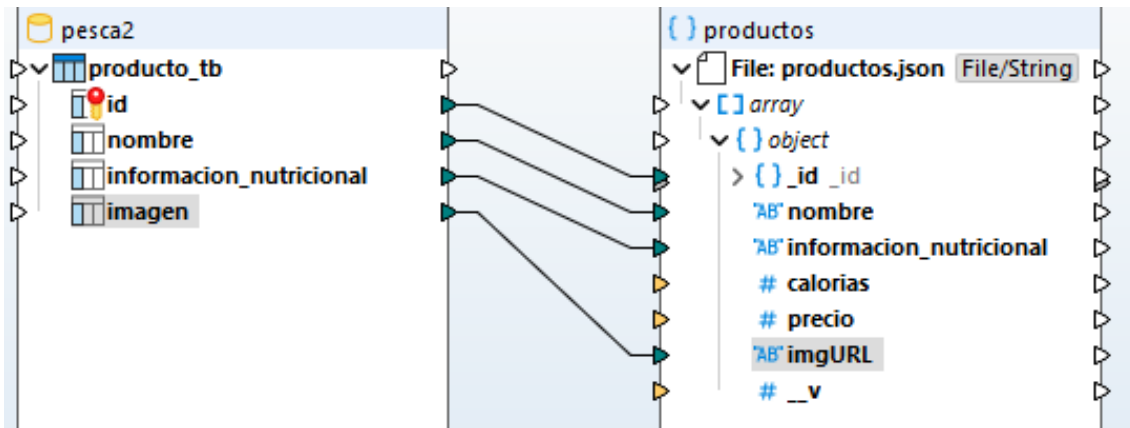


Figura B.1: Mapeo de datos de producto

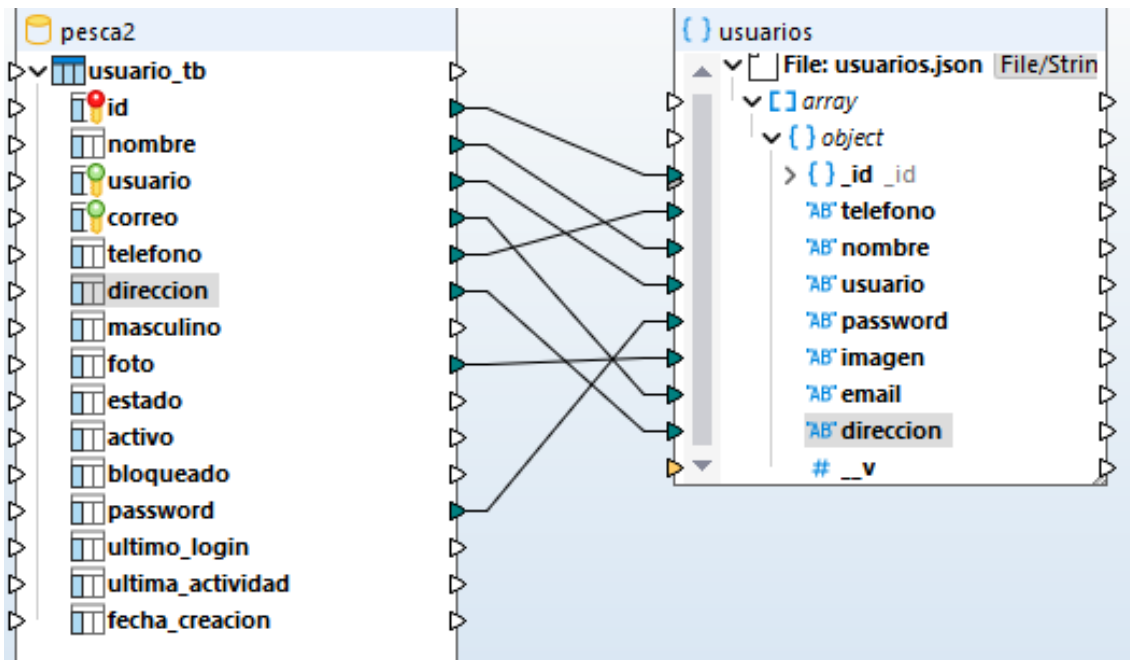


Figura B.2: Mapeo de datos de usuario

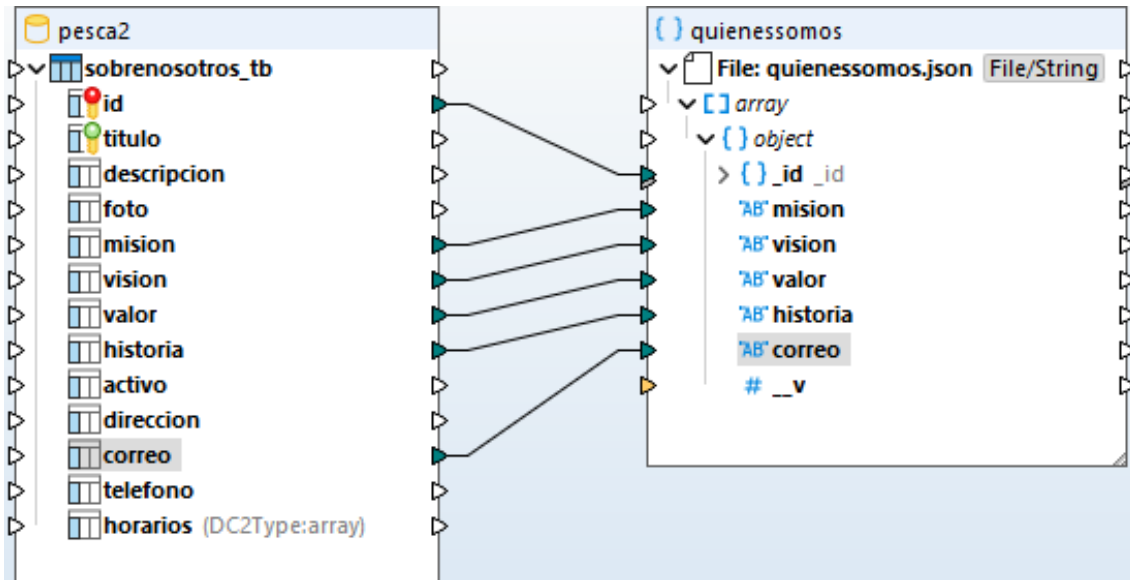


Figura B.3: Mapeo de datos quienes somos

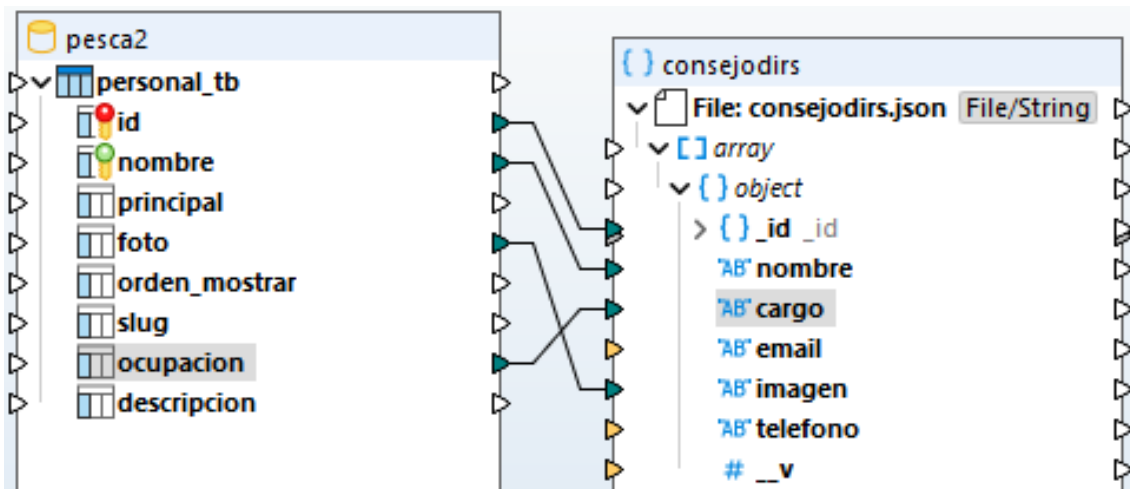


Figura B.4: Mapeo de datos personal



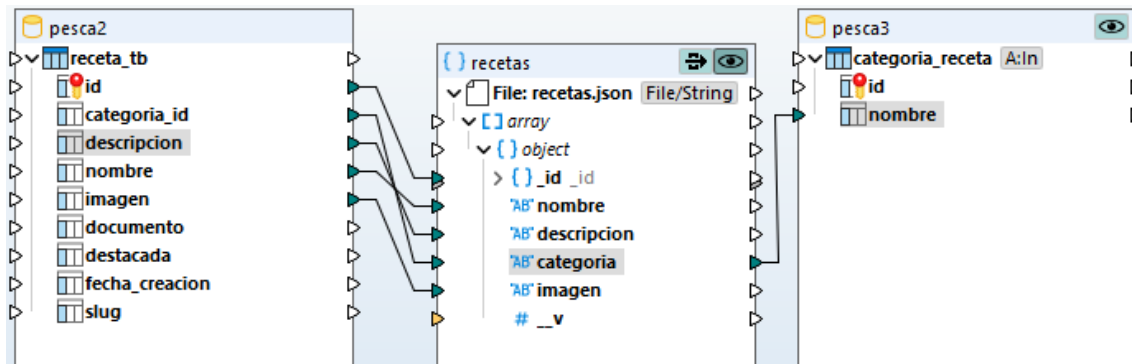


Figura B.5: Mapeo de datos receta

innecesarios y se agregan unos.

### Colección Receta

Se unen las entidades receta y categoría de receta en una misma colección, se mantienen algunos atributos de receta mientras que los otros se consideran innecesarios y se eliminan, en el caso de categoría de receta solo se tomará el nombre de la categoría.

### Colección Servicios

Se unen las entidades servicio y categoría de servicio en una misma colección, se mantienen algunos atributos de servicio mientras que los otros se consideran innecesarios y se eliminan, en el caso de categoría de servicio solo se tomará el nombre de la categoría.

### Colección Pregunta

La entidad pregunta frecuente se convierte en colección sin sufrir cambios.

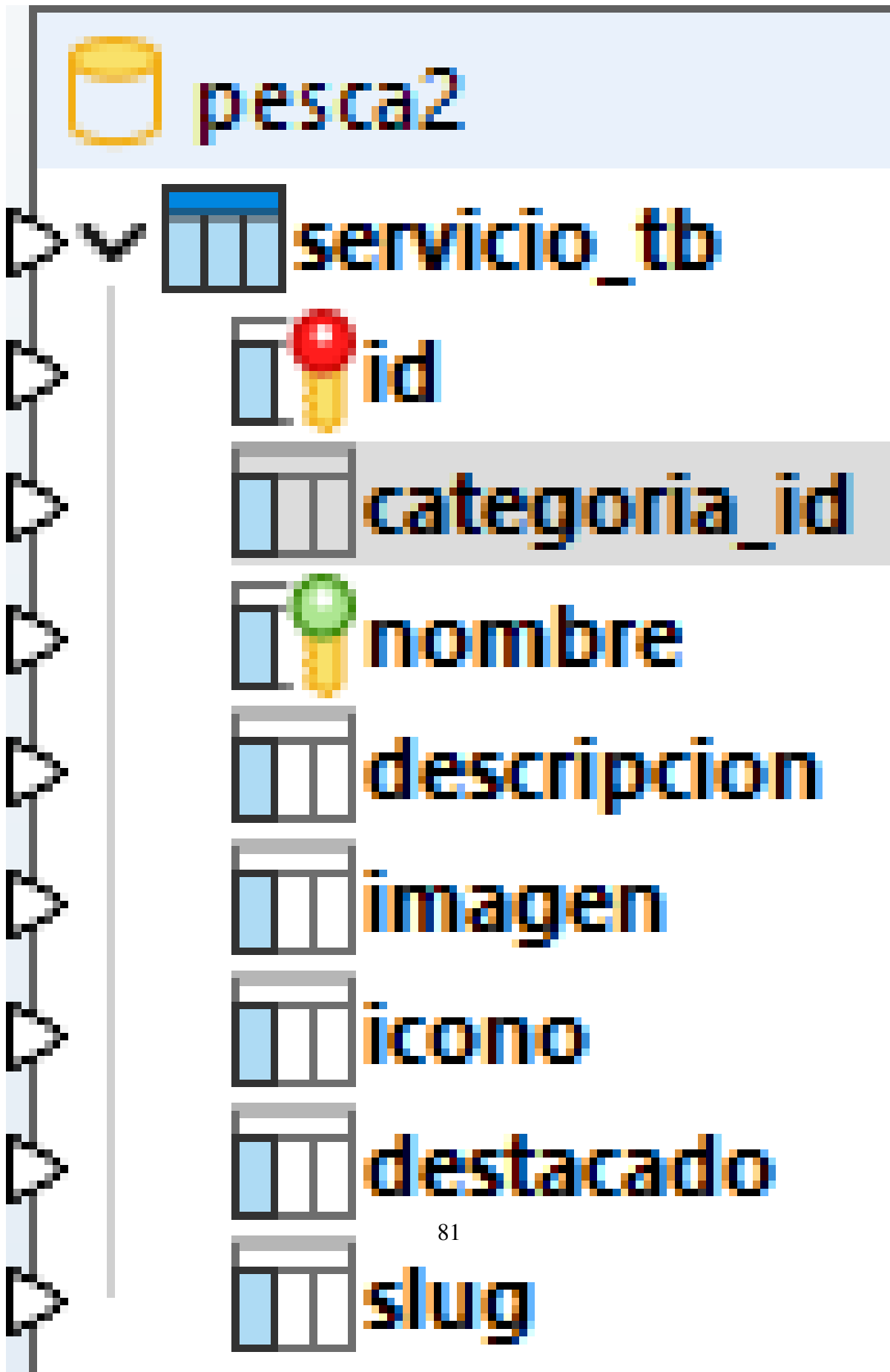
## B.2. Diseño Físico de la base de datos

### Colección Usuarios:

```
export type UsuarioDocument = Usuario & Document;
```

```
@Schema()
```

```
export class Usuario {
```



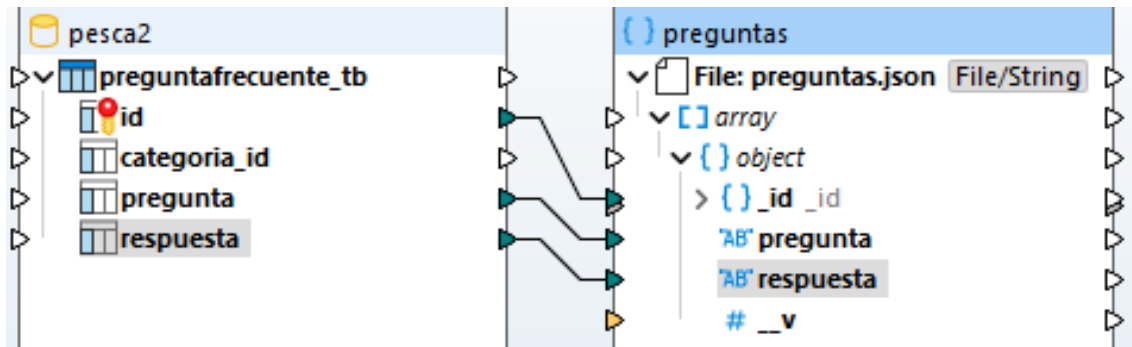


Figura B.7: Mapeo de datos pregunta

@Prop()

telefono: string;

@Prop()

nombre: string;

@Prop()

usuario: string;

@Prop()

password: string;

@Prop()

imagen: string;

@Prop()

email: string;

@Prop()

direccion: string;}

export const UsuarioSchema = SchemaFactory.createForClass(Usuario);

**Colección Pregunta:**

export type PreguntaDocument = Pregunta & Document;

```
@Schema()
export class Pregunta {
  @Prop()
  pregunta: string;
  @Prop()
  respuesta: string;}
export const PreguntaSchema = SchemaFactory.createClass(Pregunta);
```

### **Colección Producto:**

```
export type ProductoDocument = Producto & Document;
@Schema()
export class Producto {
  @Prop()
  nombre: string;
  @Prop()
  informacion_nutricional: string;
  @Prop()
  calorias: number;
  @Prop()
  precio: number;
  @Prop()
  imgURL: string;}
export const ProductoSchema = SchemaFactory.createClass(Producto);
```

### **Colección QuienesS:**

```
export type QuienesSomoDocument = QuienesSomo & Document;
```

```
@Schema()
export class QuienesSomo {
  @Prop()
  mision: string;
  @Prop()
  vision: string;
  @Prop()
  valor: string;
  @Prop()
  historia: string;
  @Prop()
  correo: string;}
export const QuienesSomoSchema = SchemaFactory.createForClass(QuienesSomo);
```

### **Colección Receta:**

```
export type RecetaDocument = Receta & Document;
@Schema()
export class Receta {
  @Prop()
  nombre: string;
  @Prop()
  descripcion: string;
  @Prop()
  categoria: string;
  @Prop()
```

```
imagen: string;}
```

```
export const RecetaSchema = SchemaFactory.createClass(Receta);
```

### **Colección Servicios:**

```
export type ServicioDocument = Servicio & Document;
```

```
@Schema()
```

```
export class Servicio {
```

```
@Prop()
```

```
nombre: string;
```

```
@Prop()
```

```
categoria: string;
```

```
@Prop()
```

```
descripcion: string;}
```

```
export const ServicioSchema = SchemaFactory.createClass(Servicio);
```

### **Colección Especies:**

```
export type EspeciesDocument = Especies & Document;
```

```
@Schema()
```

```
export class Especies {
```

```
@Prop()
```

```
_id_especies: string;
```

```
@Prop()
```

```
nombre: string;
```

```
@Prop()
```

```
descripcion: string;
```

```
@Prop()
```

```
imagen: string;}
```

```
export const EspeciesSchema = SchemaFactory.createClass(Especies);
```

### **Colección Embalses :**

```
export type EmbalseDocument = Embalse & Document;
```

```
@Schema()
```

```
export class Embalse {
```

```
@Prop()
```

```
_id_embalse: string;
```

```
@Prop()
```

```
nombre: string;
```

```
@Prop()
```

```
descripcion: string;
```

```
@Prop()
```

```
imagen: string;}
```

```
export const EmbalseSchema = SchemaFactory.createClass(Embalse);
```

### **Colección ConsejoDir:**

```
export type ConsejoDirDocument = ConsejoDir & Document;
```

```
@Schema()
```

```
export class ConsejoDir {
```

```
@Prop()
```

```
nombre: string;
```

```
@Prop()
```

```
cargo: string;
```

```
@Prop()
```

email: string;

@Prop()

imagen: string;

@Prop()

telefono: string;}

export const ConsejoDirSchema = SchemaFactory.createClass(ConsejoDir);



# **ANEXO C**

## **Requisitos funcionales 1**

Tabla C.1: RF2 Cambiar Usuario

<b>Descripción de requisito</b>	
<b>Número: RF2</b>	<b>Nombre:</b> Cambiar Usuario
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite cambiar un usuario. El usuario selecciona en el menú lateral la opción Usuario. En el área de íconos flotantes selecciona la opción Modificar. Se muestran los datos: Teléfono, Nombre, Usuario, Password, Imagen, Email, Dirección, y el usuario decidirá cuál o cuáles cambiar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD12.1

Tabla C.2: RF3 Borrar Usuario.

<b>Descripción de requisito</b>	
<b>Número: RF3</b>	<b>Nombre:</b> Borrar Usuario
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite borrar un usuario. El usuario selecciona en el menú lateral la opción Usuario. En el área de íconos flotantes selecciona la opción Eliminar. Se muestra un mensaje de confirmación donde el usuario podrá seguir con la opción de borrar el usuario o cancelar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD12.1

Tabla C.3: RF4 Enseñar Usuario.

<b>Descripción de requisito</b>	
<b>Número: RF4</b>	<b>Nombre: Enseñar Usuario</b>
<b>Programadora: Rosangela Benitez</b>	<b>Rol: Administrador</b>
<b>Prioridad: Alta</b>	<b>Complejidad: Alta</b>
<b>Descripción:</b> El requisito permite enseñar un listado con los usuarios existentes. El usuario selecciona en el menú lateral la opción Usuario. Se enseña un listado de los usuarios con los datos: Teléfono, Nombre, Usuario, Password, Imagen, Email, Dirección. Se mostrarán las opciones en el área de íconos internos: Eliminar y Modificar. Además, la opción: Crear Usuario en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD12.1

Tabla C.4: RF5 Crear Producto.

<b>Descripción de requisito</b>	
<b>Número: RF5</b>	<b>Nombre: Crear Producto</b>
<b>Programadora: Rosangela Benitez</b>	<b>Rol: Administrador</b>
<b>Prioridad: Alta</b>	<b>Complejidad: Alta</b>
<b>Descripción:</b> El requisito permite crear un producto. El usuario selecciona en el menú lateral la opción Productos. En el área de íconos flotantes selecciona la opción Crear Producto. Se muestran los datos a llenar: Nombre, Información Nutricional , Imagen, Precio y Calorías. Además, la opción: Listar en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD22.1

Tabla C.5: RF6 Cambiar Producto

<b>Descripción de requisito</b>	
<b>Número: RF6</b>	<b>Nombre:</b> Cambiar Producto
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite cambiar un producto. El usuario selecciona en el menú lateral la opción Productos. En el área de íconos flotantes selecciona la opción Modificar. Se muestran los datos: Nombre, Información Nutricional , Imagen, Precio y Calorías y el usuario decidirá cuál o cuáles cambiar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD22.1

Tabla C.6: RF7 Borrar Producto

<b>Descripción de requisito</b>	
<b>Número: RF7</b>	<b>Nombre:</b> Borrar Producto
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite borrar un producto. El usuario selecciona en el menú lateral la opción Productos. En el área de íconos flotantes selecciona la opción Eliminar. Se muestra un mensaje de confirmación donde el usuario podrá seguir con la opción de borrar el producto o cancelar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD22.1

Tabla C.7: RF8 Enseñar Producto

<b>Descripción de requisito</b>	
<b>Número: RF8</b>	<b>Nombre:</b> Enseñar Producto
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite enseñar un listado con los productos existentes. El usuario selecciona en el menú lateral la opción Productos. Se enseña un listado de los productos con los datos: Nombre, Información Nutricional , Imagen, Precio y Calorías. Se mostrarán las opciones en el área de íconos internos: Eliminar y Modificar. Además, la opción: Crear Producto en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD22.1

Tabla C.8: RF9 Crear Pescadería

<b>Descripción de requisito</b>	
<b>Número: RF9</b>	<b>Nombre:</b> Crear Pescadería.
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite crear una pescadería. El usuario selecciona en el menú lateral la Gestión del contenido y allí escoge la opción pescadería. En el área de íconos flotantes selecciona la opción Crear Pescadería. Se muestran los datos a llenar: Nombre, Teléfono y Dirección. Además, la opción: Listar en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD32.1

Tabla C.9: RF10 Cambiar Pescadería

<b>Descripción de requisito</b>	
<b>Número: RF10</b>	<b>Nombre:</b> Cambiar Pescadería.
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite cambiar una pescadería. El usuario selecciona en el menú lateral la Gestión del contenido y allí escoge la opción pescadería. En el área de íconos flotantes selecciona la opción Modificar. Se muestran los datos: Nombre, Teléfono y Dirección y el usuario decidirá cuál o cuáles cambiar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD32.1

Tabla C.10: RF11 Borrar Pescadería.

<b>Descripción de requisito</b>	
<b>Número: RF11</b>	<b>Nombre:</b> Cambiar Pescadería
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite borrar una pescadería. El usuario selecciona en el menú lateral la Gestión del contenido y allí escoge la opción pescadería. En el área de íconos flotantes selecciona la opción Eliminar. Se muestra un mensaje de confirmación donde el usuario podrá seguir con la opción de borrar la pescadería o cancelar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD32.1

Tabla C.11: RF12 Enseñar Pescadería.

<b>Descripción de requisito</b>	
<b>Número:</b> RF12	<b>Nombre:</b> Enseñar Pescadería
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite enseñar un listado con las pescaderías existentes. El usuario selecciona en el menú lateral la Gestión del contenido y allí escoge la opción pescadería. Se enseña un listado de las pescaderías con los datos: Nombre, Teléfono y Dirección. Se mostrarán las opciones en el área de íconos internos: Eliminar y Modificar. Además, la opción: Crear Pescadería en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD32.1

Tabla C.12: RF13 Crear Receta

<b>Descripción de requisito</b>	
<b>Número:</b> RF13	<b>Nombre:</b> Crear Receta
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite crear una receta. El usuario selecciona en el menú lateral la opción Receta. En el área de íconos flotantes selecciona la opción Crear Receta. Se muestran los datos a llenar: Categoría, Descripción, Nombre e Imagen. Además, la opción: Listar en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD42.1

Tabla C.13: RF14 Cambiar Receta

<b>Descripción de requisito</b>	
<b>Número: RF14</b>	<b>Nombre:</b> Cambiar Receta
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite cambiar una receta. El usuario selecciona en el menú lateral la opción Receta. En el área de íconos flotantes selecciona la opción Modificar. Se muestran los datos: Categoría, Descripción, Nombre e Imagen, y el usuario decidirá cuál o cuáles cambiar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD42.1

Tabla C.14: RF15 Borrar Receta

<b>Descripción de requisito</b>	
<b>Número: RF15</b>	<b>Nombre:</b> Borrar Receta
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta.	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite borrar una receta. El usuario selecciona en el menú lateral la opción Receta. En el área de íconos flotantes selecciona la opción Eliminar. Se muestra un mensaje de confirmación donde el usuario podrá seguir con la opción de borrar la receta o cancelar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD42.1



# **ANEXO D**

## **Requisitos funcionales 2**

Tabla D.1: RF16 Enseñar Receta

<b>Descripción de requisito</b>	
<b>Número: RF16</b>	<b>Nombre: Enseñar Receta</b>
<b>Programadora: Rosangela Benitez</b>	<b>Rol: Administrador</b>
<b>Prioridad: Alta</b>	<b>Complejidad: Alta</b>
<b>Descripción:</b> El requisito permite enseñar un listado con las recetas existentes. El usuario selecciona en el menú lateral la opción Receta. Se enseña un listado de las recetas con los datos: Categoría, Descripción, Nombre e Imagen. Se mostrarán las opciones en el área de íconos internos: Eliminar y Modificar. Además, la opción: Crear Receta en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD42.1

Tabla D.2: RF17 Crear Personal

<b>Descripción de requisito</b>	
<b>Número: RF17</b>	<b>Nombre: Crear Personal</b>
<b>Programadora: Rosangela Benitez</b>	<b>Rol: Administrador</b>
<b>Prioridad: Alta.</b>	<b>Complejidad: Alta</b>
<b>Descripción:</b> El requisito permite crear un personal. El usuario selecciona en el menú lateral la opción Personal. En el área de íconos flotantes selecciona la opción Crear Personal. Se muestran los datos a llenar: Nombre, Imagen, Cargo, Email y Teléfono. Además, la opción: Listar en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD52.1

Tabla D.3: RF18 Cambiar Personal

<b>Descripción de requisito</b>	
<b>Número: RF18</b>	<b>Nombre:</b> Cambiar Personal
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite cambiar un personal. El usuario selecciona en el menú lateral la opción Personal. En el área de iconos flotantes selecciona la opción Modificar. Se muestran los datos: Nombre, Imagen, Cargo, Email y Teléfono, y el usuario decidirá cuál o cuáles cambiar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD52.1

Tabla D.4: RF19 Borrar Personal

<b>Descripción de requisito</b>	
<b>Número: RF19</b>	<b>Nombre:</b> Borrar Personal
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador.
<b>Prioridad:</b> Alta.	<b>Complejidad:</b> Alta.
<b>Descripción:</b> El requisito permite borrar un personal. El usuario selecciona en el menú lateral la opción Personal. En el área de íconos flotantes selecciona la opción Eliminar. Se muestra un mensaje de confirmación donde el usuario podrá seguir con la opción de borrar el personal o cancelar.	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD52.1

Tabla D.5: RF20 Enseñar Personal

<b>Descripción de requisito</b>	
<b>Número: RF20</b>	<b>Nombre:</b> Enseñar Personal
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<p><b>Descripción:</b> El requisito permite enseñar un listado con los personales existentes. El usuario selecciona en el menú lateral la opción Personal. Se enseña un listado de los personales con los datos: Nombre, Imagen, Cargo, Email y Teléfono. Se mostrarán las opciones en el área de íconos internos: Eliminar y Modificar. Además, la opción: Crear Personal en el área de íconos flotantes</p>	<p><b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD52.1</p>

Tabla D.6: RF21 Crear Servicio

<b>Descripción de requisito</b>	
<b>Número: RF21</b>	<b>Nombre:</b> Crear Servicio
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<p><b>Descripción:</b> El requisito permite crear un servicio. El usuario selecciona en el menú lateral la opción Servicios. En el área de íconos flotantes selecciona la opción Crear Servicios. Se muestran los datos a llenar: Categoría, Nombre y Descripción. Además, la opción: Listar en el área de íconos flotantes</p>	<p><b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD62.1</p>

Tabla D.7: RF22 Cambiar Servicio

<b>Descripción de requisito</b>	
<b>Número: RF22</b>	<b>Nombre:</b> Cambiar Servicio
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite cambiar un servicio. El usuario selecciona en el menú lateral la opción Servicios. En el área de íconos flotantes selecciona la opción Modificar. Se muestran los datos: Categoría, Nombre y Descripción, y el usuario decidirá cuál o cuáles cambiar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD62.1

Tabla D.8: RF23 Borrar Servicio

<b>Descripción de requisito</b>	
<b>Número: RF23</b>	<b>Nombre:</b> Borrar Servicio
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite borrar un servicio. El usuario selecciona en el menú lateral la opción Servicios. En el área de íconos flotantes selecciona la opción Eliminar. Se muestra un mensaje de confirmación donde el usuario podrá seguir con la opción de borrar el servicio o cancelar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD62.1

Tabla D.9: RF24 Enseñar Servicio

<b>Descripción de requisito</b>	
<b>Número: RF24</b>	<b>Nombre:</b> Enseñar Servicio
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Alta	<b>Complejidad:</b> Alta
<b>Descripción:</b> El requisito permite enseñar un listado con los servicios existentes. El usuario selecciona en el menú lateral la opción Servicios. Se enseña un listado de los servicios con los datos: Categoría, Nombre y Descripción. Se mostrarán las opciones en el área de iconos internos: Eliminar y Modificar. Además, la opción: Crear Servicio en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD62.1

Tabla D.10: RF25 Crear Quienes Somos.

<b>Descripción de requisito</b>	
<b>Número: RF25</b>	<b>Nombre:</b> Crear Quienes Somos
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite crear una información sobre quienes somos. El usuario selecciona en el menú lateral la opción Quienes Somos. En el área de íconos flotantes selecciona la opción Crear. Se muestran los datos a llenar: Misión, Visión, Valor, Historia y Correo. Además, la opción: Listar en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD72.1

Tabla D.11: RF26 Cambiar Quienes Somos.

<b>Descripción de requisito</b>	
<b>Número: RF26</b>	<b>Nombre:</b> Cambiar Quienes Somos
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite cambiar la información sobre quienes somos. El usuario selecciona en el menú lateral la opción Quienes Somos. En el área de íconos flotantes selecciona la opción Modificar. Se muestran los datos a modificar: Misión, Visión, Valor, Historia y Correo, y el usuario decide cuál o cuáles cambiar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD72.1

Tabla D.12: RF27 Borrar Sobre Nosotros.

<b>Descripción de requisito</b>	
<b>Número: RF27</b>	<b>Nombre:</b> Borrar Quienes Somos
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite borrar una información sobre quienes somos. El usuario selecciona en el menú lateral la opción Quienes Somos. En el área de íconos flotantes selecciona la opción Eliminar. Se muestra un mensaje de confirmación donde el usuario podrá seguir con la opción de borrar la información sobre quienes somos o cancelar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD72.1

Tabla D.13: RF28 Enseñar Quienes Somos

<b>Descripción de requisito</b>	
<b>Número: RF28</b>	<b>Nombre:</b> Enseñar Quienes Somos.
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite enseñar un listado con la información sobre quienes somos. El usuario selecciona en el menú lateral la opción Quienes Somos. Se enseña un listado de las informaciones sobre nosotros con los datos: Misión, Visión, Valor, Historia y Correo. Se mostrarán las opciones en el área de íconos internos: Eliminar y Modificar. Además, la opción: Crear en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD72.1

Tabla D.14: RF29 Crear Embalse.

<b>Descripción de requisito</b>	
<b>Número: RF29</b>	<b>Nombre:</b> Crear Embalse
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite crear embalses. El usuario selecciona en el menú lateral la opción Embalse. En el área de íconos flotantes selecciona la opción Crear Embalse. Se muestran los datos a llenar: Nombre, Foto y Descripción. Además, la opción: Listar en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD82.1



Tabla D.15: RF30 Cambiar Embalse

<b>Descripción de requisito</b>	
<b>Número: RF30</b>	<b>Nombre:</b> Cambiar Embalse
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite cambiar un embalse. El usuario selecciona en el menú lateral la opción Embalse. En el área de íconos flotantes selecciona la opción Modificar. Se muestran los datos: Nombre, Foto y Descripción, y el usuario decidirá cuál o cuáles cambiar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD82.1

Tabla D.16: RF31 Borrar Embalse.

<b>Descripción de requisito</b>	
<b>Número: RF31</b>	<b>Nombre:</b> Borrar Embalse
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite borrar embalses. El usuario selecciona en el menú lateral la opción Embalse. En el área de íconos flotantes selecciona la opción Eliminar. Se muestra un mensaje de confirmación donde el usuario podrá seguir con la opción de borrar el embalse o cancelar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD82.1

# **ANEXO E**

## **Requisitos funcionales 3**

Tabla E.1: RF32 Enseñar Embalse.

<b>Descripción de requisito</b>	
<b>Número: RF32</b>	<b>Nombre:</b> Enseñar Embalse
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite enseñar un listado con los embalses existentes. El usuario selecciona en el menú lateral la opción Embalse. Se enseña un listado de los embalses con los datos: Nombre, Foto y Descripción. Se mostrarán las opciones en el área de iconos internos: Eliminar y Modificar. Además, la opción: Crear en el área de iconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD82.1

Tabla E.2: RF33 Crear Especies.

<b>Descripción de requisito</b>	
<b>Número: RF33</b>	<b>Nombre:</b> Crear Especies
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite crear especies. El usuario selecciona en el menú lateral la opción Especies. En el área de íconos flotantes selecciona la opción Crear. Se muestran los datos a llenar: Nombre, Foto y Descripción. Además, la opción: Listar en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD92.1

Tabla E.3: RF34 Cambiar Especies.

<b>Descripción de requisito</b>	
<b>Número: RF34</b>	<b>Nombre:</b> Cambiar Especies
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite cambiar una especie. El usuario selecciona en el menú lateral la opción Especies. En el área de íconos flotantes selecciona la opción Modificar. Se muestran los datos: Nombre, Foto y Descripción, y el usuario decidirá cuál o cuáles cambiar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD92.1

Tabla E.4: RF35 Borrar Especies.

<b>Descripción de requisito</b>	
<b>Número: RF35</b>	<b>Nombre:</b> Borrar Especies
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite borrar especies. El usuario selecciona en el menú lateral la opción Especies. En el área de íconos flotantes selecciona la opción Eliminar. Se muestra un mensaje de confirmación donde el usuario podrá seguir con la opción de borrar la especie o cancelar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD92.1

Tabla E.5: RF36 Enseñar Especies.

<b>Descripción de requisito</b>	
<b>Número: RF36</b>	<b>Nombre:</b> Enseñar Especies
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media.	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite enseñar un listado con las especies existentes. El usuario selecciona en el menú lateral la opción Especies. Se enseña un listado de las especies con los datos: Nombre, Foto y Descripción. Se mostrarán las opciones en el área de iconos internos: Eliminar y Modificar. Además, la opción: Crear en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD92.1

Tabla E.6: RF7 Crear Pregunta Frecuente.

<b>Descripción de requisito</b>	
<b>Número: RF37</b>	<b>Nombre:</b> Crear Pregunta Frecuente
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite crear preguntas frecuentes. El usuario selecciona en el menú lateral la opción Preguntas Frecuentes. En el área de íconos flotantes selecciona la opción Crear. Se muestran los datos a llenar: Pregunta y Respuesta. Además, la opción: Listar en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD102.1

Tabla E.7: RF38 Cambiar Preguntas Frecuentes.

<b>Descripción de requisito</b>	
<b>Número: RF38</b>	<b>Nombre:</b> Cambiar Pregunta Frecuente
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite cambiar una pregunta frecuente. El usuario selecciona en el menú lateral la opción Preguntas Frecuentes. En el área de íconos flotantes selecciona la opción Modificar. Se muestran los datos a cambiar: pregunta y respuesta	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD102.1

Tabla E.8: RF39 Borrar Pregunta Frecuente

<b>Descripción de requisito</b>	
<b>Número: RF39</b>	<b>Nombre:</b> Borrar Pregunta Frecuente
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite borrar una pregunta frecuente. El usuario selecciona en el menú lateral la opción Pregunta Frecuente. En el área de íconos flotantes selecciona la opción Eliminar. Se muestra un mensaje de confirmación donde el usuario podrá seguir con la opción de borrar la pregunta frecuente o cancelar	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD102.1

Tabla E.9: RF40 Enseñar Preguntas Frecuentes.

<b>Descripción de requisito</b>	
<b>Número: RF40</b>	<b>Nombre:</b> Enseñar Pregunta Frecuente
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Media	<b>Complejidad:</b> Media
<b>Descripción:</b> El requisito permite mostrar una lista de las preguntas frecuentes. El usuario selecciona en el menú lateral la opción Pregunta Frecuente. Se enseña un listado de las preguntas frecuentes con los datos: pregunta y respuesta. Se mostrarán las opciones en el área de íconos internos: Eliminar y Modificar. Además, la opción: Crear en el área de íconos flotantes	<b>Observaciones:</b> El usuario debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD102.1

Tabla E.10: RF41 Autenticarse.

<b>Descripción de requisito</b>	
<b>Número: RF41</b>	<b>Nombre:</b> Autenticarse
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Baja	<b>Complejidad:</b> Baja
<b>Descripción:</b> El requisito permite autenticarse en el sitio web. El usuario introduce los datos usuario y contraseña para poder entrar al sistema	<b>Observaciones:</b> El usuario debe estar añadido con anterioridad en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD12.1

Tabla E.11: RF42 Cerrar Sesión

<b>Descripción de requisito</b>	
<b>Número: RF42</b>	<b>Nombre:</b> Cerrar Sesión
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Baja	<b>Complejidad:</b> Baja
<b>Descripción:</b> El requisito permite autenticarse en el sitio web. El usuario introduce los datos usuario y contraseña para poder entrar al sistema	<b>Observaciones:</b> El usuario debe estar añadido con anterioridad en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD12.1

Tabla E.12: RF43 Cambiar Contraseña

<b>Descripción de requisito</b>	
<b>Número:</b> RF43	<b>Nombre:</b> Cambiar Contraseña
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Administrador
<b>Prioridad:</b> Baja	<b>Complejidad:</b> Baja
<b>Descripción:</b> El requisito permite cambiar la contraseña. El usuario selecciona en el menú la opción Cambiar Contraseña. Se muestra el dato Contraseña	<b>Observaciones:</b> El usuario debe estar añadido con anterioridad en el sistema con los permisos necesarios para acceder a la funcionalidad. Ver RCD12.1

Tabla E.13: RF44 Visualizar Información

<b>Descripción de requisito</b>	
<b>Número:</b> RF44	<b>Nombre:</b> Visualizar Información
<b>Programadora:</b> Rosangela Benitez	<b>Rol:</b> Cliente
<b>Prioridad:</b> Baja.	<b>Complejidad:</b> Baja
<b>Descripción:</b> El requisito permite mostrar a los clientes la información	<b>Observaciones:</b> El usuario no debe estar autenticado en el sistema con los permisos necesarios para acceder a la funcionalidad