



MÓDULO DE SISTEMA DE GESTIÓN DOCUMENTAL PARA ARGOGPL

Autor(es):

**Diego Fernando Garnica Ortiz
1214747653**

**Boris Nicolas Leon Vargas
1192778554**

**Mateo Alejandro Losada Ordoñez
1000506338**

**UNIVERSIDAD EL BOSQUE
PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
Bogotá, 2023**

MÓDULO DE SISTEMA DE GESTIÓN DOCUMENTAL PARA ARGOGPL

Autor(es):

**Diego Fernando Garnica Ortiz
Boris Nicolas León Vargas
Mateo Alejandro Losada Ordoñez**

**Trabajo de Grado presentado como requisito para optar al título de
INGENIERO DE SISTEMAS**

**Modalidad de Grado:
DESARROLLO TECNOLÓGICO**

Director(a)

**Carlos Ignacio Delgado Román
Ingeniero de sistemas
Magíster en gestión de proyectos
Magíster en gestión financiera
Maestría en administración**

**UNIVERSIDAD EL BOSQUE
PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
Bogotá, 2023**

DEDICATORIA

Le quiero dedicar este proyecto a mi hermana, mis amigos y mi director por todo el apoyo, que fueron los que estuvieron durante el proceso del mismo y saben lo que fue para mí llevar a cabo este proyecto.

Boris León

A mis papás, por el apoyo en todo aspecto.

Mateo Losada

A mi familia y amigos por el apoyo incondicional durante todo el proceso.

Diego Garnica

AGRADECIMIENTOS

Quiero agradecer a mi abuela, que sin ella no habría sido capaz de llegar a la culminación de este proyecto y que de verdad estuvo presente en cada etapa del mismo apoyándome y ayudándome en lo que podía.

Boris León

Gracias a nuestro director, a los interesados en la organización y al equipo que estuvo en el proceso desde su concepción, y dieron forma al sistema tanto en lo funcional como en lo técnico para que este fuera posible.

Mateo Losada

Quiero agradecer a nuestro director y la organización, en especial al equipo de desarrollo por el empeño que nos ayudó a dar vida a este proyecto.

Diego Garnica

Tabla de contenido

1. Introducción	4
2. Descripción Contexto y Justificación del Problema Desde el Modelo Biopsicosocial y Cultural	5
3. Marco Referencial	8
3.1. Antecedentes y Estado Del Arte	8
3.2. Marco Teórico	12
4. Descripción de la Solución Desde el Modelo Biopsicosocial y Cultural	13
4.1. Objetivos del Proyecto	13
4.2. Descripción del Artefacto	13
4.3. Componente de Análisis: Descripción de la Transformación Esperada del Contexto	14
5. Diseño Metodológico	16
6. Resultados y Discusión	20
7. Conclusiones	24
8. Lecciones Aprendidas y Trabajo Futuro	25
9. Referencias	27
Anexos	28
1. Anexo No. 1: As Is	28
2. Anexo No. 2: To Be	28
3. Anexo No. 3: Diagrama de arquitectura	28
4. Anexo No. 4: Página principal Argo	28
5. Anexo No. 5: Bandeja Argo	28
6. Anexo No. 6: Radicación Argo	28
7. Anexo No. 7: Lista Multiempresa Argo	28
8. Anexo No. 8: Mockup Módulo divipola Argo	29
9. Anexo No. 9: Release Plan	29
10. Anexo No. 10: Resultados Pruebas estáticas Argo Origen	29
11. Anexo No. 11: Manual técnico del sistema	29
12. Anexo No. 12: Requerimientos funcionales del sistema	29
13. Anexo No. 13: Documento de Arquitectura del Sistema (SAD)	29
14. Anexo No. 14: Reuniones de Sprint Planning	29
15. Anexo No. 15: Reuniones de Sprint Review	29
16. Anexo No. 16: Entrevista pruebas técnicas desarrolladores	29
17. Anexo No. 17: Manual de usuario del sistema	29

MÓDULO DE SISTEMA DE GESTIÓN DOCUMENTAL PARA ARGOGPL

DOCUMENT MANAGEMENT SYSTEM MODULE FOR ARGOGPL

Diego Fernando Garnica Ortiz, Boris Nicolas León Vargas, y Mateo Alejandro Losada Ordoñez
dgarnicao@unbosque.edu.co, bleonv@unbosque.edu.co, mlosada@unbosque.edu.co
Universidad El Bosque, Colombia

Resumen—El objetivo de este proyecto fue el proceso de reingeniería del primer módulo para el inicio de la actualización del Sistema de Gestión Documental Electrónico de Archivos ArgoGPL. Este proceso incorpora el cambio de tecnologías, donde se incluye la transición a una arquitectura orientada a microservicios, la implementación de buenas prácticas y el cumplimiento de la legislación Colombiana actual; permaneciendo cómo un software libre que permitirá mejorar la escalabilidad, usabilidad y mantenibilidad del sistema. El presente proyecto, que fue llevado a cabo bajo una metodología SCRUM, se trabajó muy de la mano con la fundación Correlibre para poder diseñar y desarrollar un sistema que fuera apto para poder desempeñar las labores del anterior, supliendo las necesidades de parametrización y diseño con las que se contaban en el sistema antiguo, de tal forma que se pueda contar con mejoras significativas tanto para el desarrollador y el usuario final, como para el cliente mismo.

Abstract— The objective of this project was the reengineering process of the first module for the start of the update of the ArgoGPL Electronic Document Management System. This redesign process incorporates the change of technologies, including the transition to a microservices oriented architecture, the implementation of best practices and compliance with current Colombian legislation; remaining as a free software that will improve the scalability, usability and maintainability of the system. This project, which was carried out under a SCRUM methodology, worked hand in hand with the Correlibre Foundation to design and develop a system that was able to perform the work of the previous one, meeting the needs of parameterization and design that were in the old system, so that you can have significant improvements for both the developer and the end user, as for the client itself.

Palabras Clave— Document management, Information flow controls, Dataflow architectures, Workflow management

----- ◆ -----

1. INTRODUCCIÓN

En el contexto de este proyecto, se encontró con el desafío de crear un sistema a la vanguardia que sea capaz de responder a las necesidades personalizadas de cada cliente, sin dejar de cubrir las exigencias gubernamentales y empresariales a las que se debe someter los sistemas de gestión documental. Dichas necesidades son cambiantes y de gran exigencia, por lo que este proyecto se concibe cómo una solución integral, flexible, escalable y mantenible para suplir a su antecesor, OrfeoGPL, y además, continuar vigente en el panorama futuro.

Reconociendo la importancia de los sistemas de gestión documental (SGD) como herramientas transversales dentro de las organizaciones para la organización, acceso, flujo y almacenamiento de

la información, se busca crear un sistema que no solo cumpla con las tareas básicas, sino que permita la implementación de elementos como BPM e IA, además de una personalización, fácil integración con sistemas externos y usabilidad notoria para que sea una herramienta más completa y competente en el mercado actual.

Del mismo modo, aunque se tienen varias opciones de SGD dentro del mercado, son pocas las que se pueden conseguir como opciones de software libre que presenten las mismas características y posibilidades de sus contrapartes privatizadas. Este proyecto permitirá crear las bases para futuramente liberar una opción robusta y accesible dentro del mercado.

Los SGD actuales, como OrfeoGPL cuentan con varias falencias en cuanto a procesos de ingeniería, así como la posibilidad de ser sistemas altamente escalables que permitan la implementación de nuevas funcionalidades acordes a las necesidades de cada cliente de una forma sencilla y eficaz, razón por la cual este proyecto busca crear un sistema altamente parametrizable, y con niveles de mantenibilidad y escalabilidad mucho más óptimos. Del mismo modo implementar mejoras enfocadas en la usabilidad para mejorar la experiencia del usuario dentro del sistema.

En el presente documento se ofrece una contextualización y análisis del proceso de gestión documental, su aplicación en el entorno actual y el impacto esperado del nuevo artefacto. Se abordan diversos temas, entre ellos la historia de los sistemas de gestión documental en Colombia, la regulación que estos tienen a nivel jurídico, y las necesidades que se presentaron al momento de realizar un proceso de reingeniería sobre un sistema desactualizado, para poder mejorar sobre estos los aspectos de escalabilidad, mantenibilidad y usabilidad, y convertirlo en un sistema base para cualquier sistema de gestión de documentos electrónicos que se requiera. Posteriormente se presenta la forma en la que fue desarrollado el proyecto y los resultados que se obtuvieron a partir de la finalización del mismo.

2. DESCRIPCIÓN CONTEXTO Y JUSTIFICACIÓN DEL PROBLEMA DESDE EL MODELO BIOPICOSOCIAL Y CULTURAL

OrfeoGPL es un Sistema de Gestión Documental Electrónico de Archivos (SGDEA) inicialmente desarrollado para la Superintendencia de Servicios Públicos Domiciliarios (SSPD) en Colombia, publicado con licencia GNU/GPL para compartir el conocimiento y mantener la creación colectiva. La empresa Infometrika en conjunto con la fundación Correlibre ha venido trabajando a lo largo de los años para poder adaptar este sistema a las diversas necesidades del mercado y de los clientes, creando un proyecto cada vez más robusto que pueda satisfacer dichas demandas. Argo GPL nace como la evolución de OrfeoGPL versión Correlibre, dado que este sistema ideado desde el año 2002 y desarrollado en su totalidad para el 2006, ha sido introducido por múltiples entidades, lo que implicó cambios en el transcurso del tiempo, en su mayoría estas entidades hacen parte del sector público del gobierno Colombiano.

El actual sistema de gestión documental Argo/OrfeoGPL generó cambios beneficiosos en el medio a la hora de reducir el consumo de papel [1], [2], puesto que permitió digitalizar y manejar documentos con una disposición y gestión totalmente digital en diversas empresas en las que fue introducido. Del mismo modo, este trajo beneficios para los actores de las compañías que lo implementan, puesto que los trámites o procesos documentales se llevaron a cabo de manera más rápida, ordenada y eficaz, reduciendo costos en tiempo y dinero para la compañía. Sin embargo, esta promesa de mejora respecto al impacto en el medio, puede verse afectada por los hábitos de las personas dentro de las empresas que finalmente deciden usar o no el software, dado que su percepción del mismo puede verse trastornada [3] por los errores y deficiencias que posee actualmente; además teniendo en cuenta su creación hace más de 20 años, se genera un escepticismo adicional, y por tanto, una oposición al uso frecuente del sistema, que a su vez reduce la magnitud del impacto deseado.

Por otra parte, se encuentra otro tipo de problema debido a las tecnologías usadas en su momento para la creación del sistema y las necesidades divergentes que se presentan a la empresa actualmente. Entre estas se encuentra la adaptación del software a las nuevas normas que se generan con respecto al manejo de los documentos y la información, como las establecidas por el Archivo General de la Nación y MinTIC, donde se encuentran algunas como el Decreto 2609 de 2012, que entre otras cosas define al mantenimiento como el “establecimiento de los requisitos que permitan mantener la integridad técnica, estructural y relacional de los documentos en el sistema de gestión documental” [4]. Por su parte, esto afecta directamente los desarrollos para mantener actualizado el sistema, ya que, al no ser un sistema modular, los desarrolladores enfrentan problemas como la difícil integración con otros sistemas, al igual que impedimentos presentes a nivel de escalabilidad y mantenibilidad. [Ver Anexo No.1]

Así, en la Tabla 1 se identifican los principales elementos que conformarán el análisis BioPsicoSocial y Cultural por medio del cual se llevó a cabo este estudio de contexto.

Tabla 1. Análisis del contexto por el Modelo BPSC (Biopsicosocial y cultural)

MBPSC	ArgoGPL
Medio	<ul style="list-style-type: none"> ● El proyecto en su forma actual está enfocado a empresas que requieren de una gestión documental y control de flujo de documentos, que esté siendo manejada en su mayor parte en papel o en físico. ● Ya que se trabaja dentro de la comunidad de <i>software libre</i>, se incluyen las personas que pueden modificar el sistema y las otras herramientas con que trabaja, puesto que estas siguen lineamientos y licencias similares ● El sistema debe ser adaptado a las necesidades de cada uno de los clientes. <p>[3]</p>
Creencias	<ul style="list-style-type: none"> ● El sistema es complejo de usar, es muy robusto y poco intuitivo. ● No hay un proceso de gestión de cambio adecuado que permita introducir el sistema de forma amigable dentro de las organizaciones que lo implementan. ● Inseguridad frente a quien tiene acceso a determinados documentos. ● Insuficiente disponibilidad de documentos a causa de su difícil ubicación. ● La reputación del sistema y de la organización se vio afectada por los desarrollos independientes y desinformados realizados sobre versiones creadas a partir de la base inicial (versión publicada oficialmente por Correlibre).

Hábitos	<ul style="list-style-type: none"> ● Resistencia al cambio para el uso del sistema actual. ● Demoras en procesos burocráticos y flujo de documentos físicos en empresas. ● Falta de prácticas de ingeniería en los procesos de desarrollo del sistema (buenas prácticas, arquitectura de software, etc). ● Continuación de malas prácticas de ingeniería para el desarrollo e integración de nuevas funcionalidades. <p>[3]</p>
Artefactos	<ul style="list-style-type: none"> ● Argo/ OrfeoGPL es un sistema con una arquitectura monolítica, que ha sido desarrollado a lo largo de los años por múltiples equipos de desarrollo y con múltiples funcionalidades. ● Las herramientas y tecnologías usadas en el Argo/ OrfeoGPL actual son un framework denominado <i>Caliope</i>, que utiliza el lenguaje de programación <i>Php</i>, un servidor <i>Apache</i> y la base de datos relacional <i>PostgreSQL</i>. Además, se han incluido otras herramientas como <i>Docker</i>, <i>Nginx</i> e incluso adiciones de <i>JBPM</i>.

Según el análisis que se llevó a cabo con un punto de vista de problemática u oportunidad, se establecieron las siguientes correlaciones entre los elementos presentados en la **Tabla 1**.

En primer lugar, los usuarios presentan una alta resistencia al cambio en el momento en que se intenta introducir esta clase de sistemas dentro de una entidad como causa de su confianza en los procesos antiguos, en el papel y el funcionamiento actual de la compañía. De la misma manera, al presentar el sistema ante ellos, se muestra como un sistema bastante robusto y difícil de usar, el cual podría volver más complicados los procesos de flujo de control de documentos, su accesibilidad, y a causa del desconocimiento del propio sistema, se genera desconfianza frente a la seguridad de los archivos y quien tiene acceso a estos [5].

Con esta resistencia al cambio presente, se genera la necesidad de que el sistema sea adaptado para cada una de las empresas que deciden implementarlo, de tal forma que sea más personalizado y los procesos sean moldeables dependiendo de las necesidades de la compañía. A esta necesidad de adaptaciones, se le interpone la arquitectura monolítica actual del sistema junto con las herramientas que este usa, mencionadas anteriormente en la **Tabla 1**. Esto genera complicaciones a la hora de realizar cambios, puesto que, al ser un sistema tan robusto, y al haber tenido procesos de desarrollo poco estandarizados, cuenta con un nivel de complejidad y acoplamiento bastante alto, lo cual lo vuelve menos escalable, menos mantenible y, por lo tanto, menos personalizable para cada empresa. Dentro de esto, también hay que tener en cuenta que cada uno de los desarrollos tomaría más tiempo del debido a la ralentización que implica tener que entender el sistema y encontrar formas de agregar nuevas funcionalidades sin dañar las ya existentes con nuevos desarrollos, afectando tanto al desarrollador como al cliente [6].

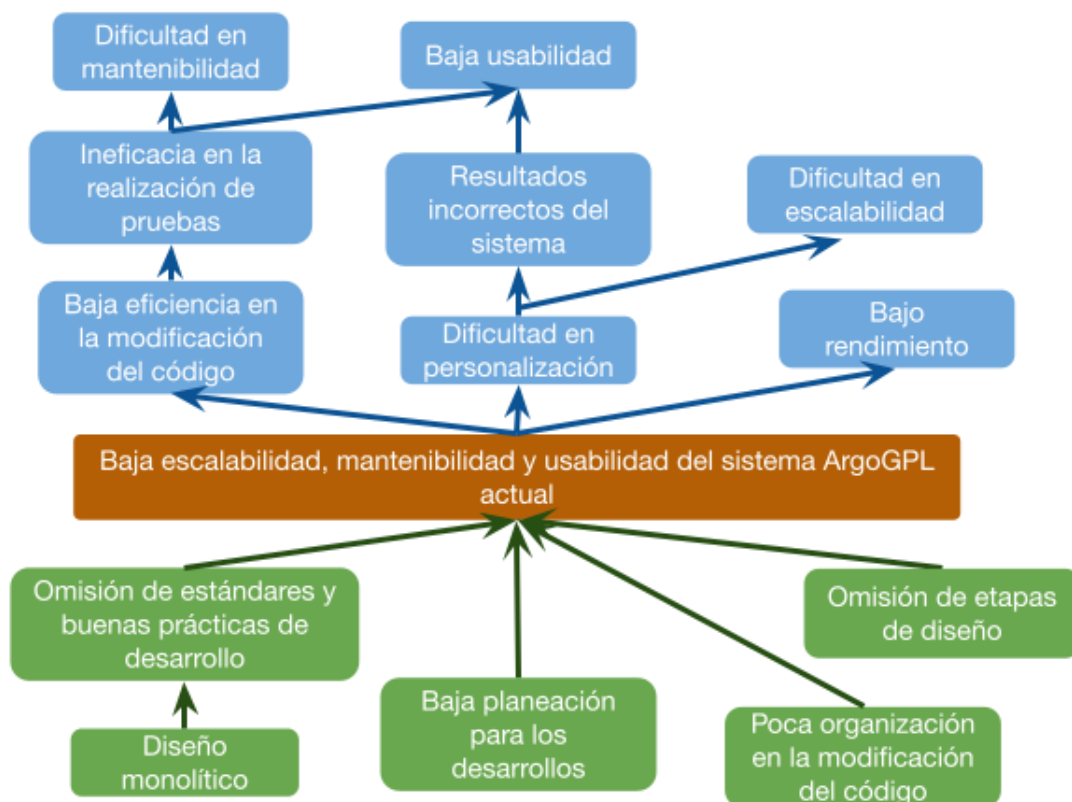
Dicho estado actual de la tecnología, convierte al sistema finalmente en uno con grandes falencias tanto en diseño cómo en la construcción del mismo, dejándolo con defectos tales como tiempos de

respuesta largos y una usabilidad deficiente, poca maleabilidad y difícil mantenibilidad, reafirmando las creencias de un sistema poco efectivo y difícil de usar, aumentando así el hábito de rehusarse a usar el sistema, o al menos tener cierta intransigencia a la hora de hacer uso del mismo [3,7].

Finalmente, esto va a tener múltiples repercusiones en el medio, puesto que el cliente no va a poder llevar a cabo una correcta actualización de su flujo y control documental, lo que conlleva una pérdida de tiempo y dinero para el mismo. De igual forma, al tener ya múltiples falencias en su diseño y construcción, las versiones que se quieran crear a partir de este, al contar con una licencia de software libre, poseerá los mismos defectos y es probable que se mantengan las malas prácticas de desarrollo, la falta de estándares en su construcción y, por lo tanto, que se siga empeorando la percepción tanto del sistema en general, como de la fundación que publicó el sistema inicial, dentro del mercado y entorno en que se mueven.

De esta manera, en el Gráfico 1 se plantea un árbol de problemas con las causas y consecuencias ya descritas anteriormente.

Gráfico 1. Árbol de problemas



Como se presenta en el Gráfico 1, y junto con lo descrito a lo largo de esta sección, se identifican tres factores principales que engloban dichas falencias del sistema. Estos factores se consideran el punto de partida para identificar las oportunidades de mejora en el proceso de diseño del nuevo sistema. Tomar en cuenta estas variables permite atacar problemas específicos como lo pueden ser el incremento en costos de desarrollo y mantenimiento generados por la alta cohesión en el código, o la flexibilidad deficiente frente a las necesidades actuales del mercado.

3. MARCO REFERENCIAL

3.1. Antecedentes y Estado Del Arte

La IEEE, en el glosario de ingeniería de software, define la mantenibilidad como la facilidad con la que un sistema puede ser modificado para corregir fallos, mejorar el rendimiento o adaptarse a un cambio en su entorno.

La usabilidad la define como la facilidad con la que un usuario puede aprender a operar, preparar las entradas e interpretar las salidas de un sistema.

La fiabilidad es la capacidad de un sistema o componente para realizar las funciones requeridas en las condiciones establecidas durante un periodo de tiempo determinado.

La disponibilidad se define como el grado en que un sistema o componente es operativo y accesible cuando es necesario para su uso.

La tolerancia a fallos es la capacidad de un sistema o componente para seguir funcionando con normalidad a pesar de la presencia de fallos de hardware o software. [8]

Documento Electrónico

Para poder dar la definición de este concepto se toma una adaptación a partir de lo establecido por el decreto 1080 de 2015 ARTÍCULO 2.8.2.7.2. , el material proporcionado por el minTIC y la ley 1564 de 2012 de El Código General del Proceso, en donde se toma Documento Electrónico como la información generada, enviada, recibida, almacenada y comunicada por medios electrónicos, ópticos o similares en donde estos archivos dependen más de su estructura lógica que de su estructura física, esto es que cuentan con un contenido estable, forma documental fija, un vínculo archivístico y un equivalente funcional, con el fin de que se pueda validar su existencia e integridad. Así, se destaca que un documento puede abarcar objetos tales como documentos escritos, impresos, planos, dibujos, cuadros, mensajes de datos, fotografías, cintas cinematográficas, discos, grabaciones magnetofónicas, videograbaciones, radiografías, talones, contraseñas, cupones, etiquetas, sellos y, en general, todo objeto mueble que tenga carácter representativo o declarativo, y las inscripciones en lápidas, monumentos, edificios o similares.

Sistema de información

Para la definición de Sistema de información se toma la definición planteada por el Proyecto Educativo del Programa de Ingeniería de Sistemas de la Universidad El Bosque que dice que se entiende a un Sistema como una entidad (un todo) y no como un conjunto de partes, conformado por "personas, datos e información, infraestructura tecnológica, software y procesos de negocio que recopila, almacena, protege, transforma y distribuye información en entornos sociales" [9].

Gestión Documental

Existen diversas definiciones acerca de la gestión documental, dado que este ejercicio no se puede definir en una serie de tareas globales, sino que actúa de manera transversal y personalizada en cada organización, formando así parte vital del camino de digitalización para las instituciones donde se facilita la utilización y conservación de documentos al punto de llegar a convertirse en un eje importante para la administración de la información de forma adecuada [10].

Así, una perspectiva muy importante sobre la gestión documental es definida en la Ley General de

Archivos para Colombia como un “conjunto de actividades administrativas y técnicas tendientes a la planificación, manejo y organización de la documentación producida y recibida por las entidades, desde su origen hasta su destino final, con el objeto de facilitar su utilización y conservación” (Ley 594, 2000) [11].

De la misma manera, para Patricia Russo la gestión documental es un “conjunto de actividades que permiten coordinar y controlar los aspectos relacionados con creación, recepción, organización, almacenamiento, preservación, acceso y difusión de documentos” [12].

Normativa Colombiana sobre Gestión Documental

La Constitución Política de Colombia define principios esenciales tales como el derecho a la información y el acceso a documentos públicos por medio de Leyes y Acuerdos, dando pie al nacimiento del Decreto 103 de 2015 que reglamenta la gestión documental en la administración pública y mencionando al Archivo General de la Nación (AGN), en el artículo 32, como una entidad encargada del establecimiento de la Política Pública de acceso a la información.

En Colombia se tiene una amplia reglamentación vigente acerca de la gestión documental, en las que cabe resaltar la construida por el AGN, la Política Nacional de Archivos, la cual es un conjunto de directrices para salvaguardar el patrimonio documental, así como la modernización de los archivos públicos. Así mismo, el gobierno nacional fomenta iniciativas de gobierno electrónico y adopción de tecnologías de la información y la comunicación (TIC) en la administración pública en dicha normativa. [13]

Dentro de esta Política Nacional de Archivos se encuentra la política de gestión documental en el Estado Colombiano, que se define en el artículo 6 del Decreto 2609 de 2012 como “el conjunto de directrices establecidas por una entidad para tener un marco conceptual claro para la gestión de la información física y electrónica, un conjunto de estándares para la gestión de la información en cualquier soporte, una metodología general para la creación, uso, mantenimiento, retención, acceso y preservación de la información” [14].

Sistemas de Gestión Documental

En cuanto a los sistemas de gestión documental (SGD), en 2016 Berdugo Báez [15] presenta un análisis de dos (SGD) que existían para ese entonces en Colombia y los cuales cumplían con la legislación vigente en aquel momento para empresas públicas y lo establecido en la Ley General de Archivos. El principal objetivo de dicho documento es comparar los dos sistemas de información que se encontraban debidamente reglamentados, con el fin de definir si cumplían la normatividad vigente, esto a base de la pregunta planteada por Báez [15]: “¿Los sistemas de gestión documental que existen actualmente en Colombia cumplen con la legislación vigente para empresas públicas de acuerdo con el programa de gestión documental?”. De esta manera se analizaron los sistemas Orfeo y Alfresco, puesto que eran sistemas que para la época habían realizado avances significativos en el área con tecnologías novedosas y que permitía una buena implementación para los SGDEA, llegando a la conclusión de que los programas de computador que se implementaron en las entidades públicas cumplían con la mayoría de requisitos establecidos por el Archivo General de la Nación.

Sin embargo, en contraposición con lo anterior, comunicaciones más recientes con el cliente [16] denotaron que los sistemas de información que se ofrecen en Colombia implementados en las entidades públicas de las que tiene conocimiento no cumplen con la mayoría de requisitos exigidos por el ya mencionado ente rector (AGN), por lo cual se debe hacer un replanteamiento para actualizar el software Argo y que este cumpla con las nuevas normas exigidas a nivel nacional e internacional. Es de ver que el AGN se ha basado internacionalmente en normas Moreq2 [17].

Siguiendo con la misma línea, Puentes Poloche [18] para 2017 muestra una propuesta metodológica que permitiera precisamente articular los procesos de gestión documental con un cumplimiento total de la Ley General de Archivo, con el fin último de que cualquier organización en Colombia pueda implementar sistemas de este estilo sin que en la práctica surjan inconvenientes que puedan resultar en problemas e impedimentos para la organización en algún momento. Dentro de los parámetros para una buena implementación se encuentra que es necesario llevar un análisis de madurez y estructura de la organización, tener en cuenta algunas buenas prácticas en los procesos documentales y de gestión de archivos internos, y además de esto entender el impacto a nivel técnico y humano que esta implementación puede tener dentro de la misma, comprendiendo e identificando las ventajas que puede traer y los principales puntos que hay que tener en cuenta a la hora de tomar la decisión final.

Sin embargo, su implementación inicial cuenta aún con muchas barreras que se deben visualizar con anterioridad para poder sortearlas con estrategias y decisiones tomadas al respecto. Este es el caso que Ahmad y compañía [19] presentan para el mismo año en donde se encuentra que, aunque las organizaciones están tomando medidas y pasos para poder llevar a cabo un proceso de digitalización dentro de la empresa, aún existe la necesidad de implementaciones efectivas de sistemas como los de gestión documental con el fin de lograr resultados exitosos dentro de la organización. Principalmente, a partir del estudio, se encontró que entre las principales barreras existentes para el correcto y exitoso proceso de implementación de estos sistemas se encuentra que esta clase de procesos requieren una gran inversión de tiempo, esfuerzo y dinero, en comparación con los beneficios que tomaran un tiempo un tanto mayor en ser vistos por parte de la compañía. Del mismo modo, se encontró que los empleados se mostraban reacios a aprender estos nuevos métodos y procedimientos para llevar a cabo sus actividades habituales dentro de estos sistemas, por lo cual se debe tener presente los usuarios finales al momento de tomar la decisión de aplicar los nuevos sistemas dentro de la compañía y las medidas que se tomarán al respecto.

Con lo anterior, se analizan las ventajas que se puede llegar a tener al implementar esta clase de SGD, no solo en las organizaciones de manera individual, sino incluso a nivel nacional y el gran impacto es, en su mayoría positivo. Así es como en 2017 la Facultad de Administración y Negocios de la Universidad Argentina de la Empresa publicó un artículo investigativo sobre los efectos de la digitalización del sector público [20]. Dentro de este análisis a la implementación de estos sistemas dentro de los órganos públicos a nivel nacional en Argentina, en palabras del autor le define cómo: “un aumento notorio en la transparencia administrativa y ahorros de tiempo, espacio físico y traslado no solo a la Administración Pública Nacional sino también a la ciudadanía”.

Y es precisamente en 2018 cuando Cerrillo Martínez y Casadesús de Míngo [21] presentan un documento cuya premisa principal es: “La gestión documental tiene un impacto significativo en la mejora de la transparencia de la Administración pública” donde se examinan cada uno de los cambios que los órganos públicos deberían implementar en sus SGD para poder contribuir de manera significativa a la mejora de la transparencia pública, incorporando las obligaciones de transparencia presentes en la legislación directamente desde el diseño del sistema y garantizando así el acceso de la ciudadanía a la información pública.

Microservicios

Así mismo, se ha visto la necesidad que el sistema de información, además de cumplir con normas expedidas por AGN, permita tener una mejor gestión y tratamiento de la información implementando una estrategia de gestión de procesos de negocio (BPM, Business process management) así como herramientas de inteligencia artificial (AI). Con esto presente, el trabajo a nivel de microservicios permitirá implementar y/o escalar estas tecnologías de manera más óptima.

Mateus-Coelho y compañía [22] hace un breve resumen de las propiedades de micro-servicios al momento del desarrollo del documento, donde resalta que son de gran importancia al momento del

desarrollo de software tales como la facilidad en el proceso de escalado, la independencia en la actualización de características por parte del equipo desarrollador y el simple mantenimiento de cada una de estas características, necesidades presentes actualmente en el sistema objeto del documento. Así mismo se evita las fallas en cascada en el sistema porque un servicio falle, es decir, solo este último se ve afectado.

También resalta otros dos puntos importantes en la arquitectura de micro-servicios; la heterogeneidad y los múltiples lenguajes que se pueden usar en un mismo sistema y que se acomode a cada uno de los servicios que se desarrollan, incluso permitiendo una reconstrucción de los mismos sin afectar el sistema, así como que cada equipo pueda estar enfocado en un solo objetivo permitiendo tener una mayor robustez.

3.2. Marco Teórico

Es necesario realizar un cambio a la arquitectura del sistema de la cual se puedan aprovechar características de maleabilidad y mantenibilidad y sobre la cual se puedan llevar a cabo múltiples desarrollos. El sistema original tiene una arquitectura monolítica, la cual se plantea cambiar hacia una orientada a microservicios para impulsar el funcionamiento modular. La modularización de las funcionalidades permite realizar cambios más fácilmente en el sistema, teniendo en cuenta que la adición de funcionalidades tiene menor impacto en el funcionamiento de los demás desarrollos al momento de desplegar los cambios en el sistema principal [23].

Una arquitectura distribuida es una de las mejores opciones disponibles enfocada al sistema que se quiere conseguir. La arquitectura de microservicios consiste en definir una aplicación como la colección de múltiples servicios, donde estos se comunican constantemente para obtener la funcionalidad completa, pero presentando una alta cohesión dentro de cada módulo o servicio y así permitiendo un encapsulamiento mayor para mitigar el acoplamiento [6]. De esta manera, se aumenta la cohesión del sistema, reduciendo en gran medida el acoplamiento y permitiendo una mayor escalabilidad en la medida en la que el sistema será más maleable y más tolerante a fallos.

Con ello, también es importante la actualización de las tecnologías. Entre ellas, en primer lugar, se encuentran los frameworks Django y Vue, con los respectivos lenguajes de programación Python y Javascript, escogidos por su facilidad de implementación y su capacidad de ser utilizados dentro del modelo de arquitectura escogido. Por su parte, cabe destacar que estos frameworks fueron elegidos porque permiten una alta reutilización de componentes (entre sus diversas características) y reduciendo líneas, permitiendo crear un código más limpio y facilitando mantener un estándar en la codificación de forma transversal en el equipo de desarrollo. Luego, también se tiene la posibilidad de integrar herramientas como Docker, Kubernetes o Keycloak, que ayudan con procesos para el despliegue y la seguridad, que facilita el trabajo al desarrollador y asegura el funcionamiento para el usuario final. Todo esto, como parte de recomendación y elección del cliente.

Para la conexión de estos microservicios se encuentra la API gateway Kong, el cual tiene entre sus tareas la de gestionar cada una de las solicitudes y accesos; para esto último se refuerza con el sistema de autenticación y autorización Keycloak, consiguiendo así un gran manejo del sistema ArgoGPL y una fuerte seguridad, entre otros aspectos [25, 26].

Luego, con el fin de mantener una estandarización del código, refiriéndose a esto como el seguimiento de buenas prácticas de desarrollo y la creación de un código limpio, se implementaron herramientas de análisis de código estático, que realizan revisiones constantes sobre el código, exigiendo el cumplimiento de estándares y buenas prácticas sobre la codificación del sistema. Por medio de estas herramientas se pueden detectar fácilmente no solo malas prácticas de desarrollo, sino falencias en cuanto a optimización y seguridad dentro del sistema. Dentro del **Anexo. 11** en la sección de componentes y estándares se puede profundizar un poco más sobre lo que fueron los estándares seguidos dentro del proceso de desarrollo del sistema.

En el diseño y construcción de cada uno de estos microservicios se ven reflejadas las tecnologías de vanguardia, para obtener un desempeño más adecuado en el desarrollo y conllevando a una mayor mantenibilidad y escalabilidad para futuros desarrollos. Esto se puede ver siendo que los Frameworks de desarrollo elegidos permiten reducir la cantidad de código necesario para la implementación de diversas funcionalidades, recortando la cantidad de código escrito, pero duplicando la funcionalidad del mismo; así se logra no solo satisfacer las exigencias contemporáneas sino permitir que el sistema pueda conservar su sostenibilidad a lo largo del tiempo y los cambios que deban ser aplicados.

4. DESCRIPCIÓN DE LA SOLUCIÓN DESDE EL MODELO BIOPSIOSOCIAL Y CULTURAL

El sistema Argo/OrfeoGPL, al igual que diversos sistemas actuales, representa un cambio en la forma cómo se entiende y aplica la gestión documental, siendo una herramienta útil que permite una evolución respecto al manejo de registros en físico, sin embargo, se convirtió en algo desactualizado y dispendioso con relación a múltiples aspectos, entre los que destacan la usabilidad y fiabilidad [24]. Actualmente, exhibe carencias respecto al desarrollo e implementación esperado en las organizaciones, implicando así crecientes inconformidades relacionadas principalmente con la baja escalabilidad y mantenibilidad del sistema. De igual manera, existen falencias en su nivel de usabilidad, limitando el impacto previsto debido a la disparidad de tecnologías, metodologías y diseño que existían entre la construcción inicial y el estado actual. ArgoGPL es un proceso de reingeniería concebido a partir de la identificación de oportunidades de mejora en los sistemas actuales, con tecnologías y prácticas de vanguardia, un diseño parametrizable y mucho más moderno, para una mayor escalabilidad, personalización, mantenibilidad y usabilidad, para potenciar las labores que el sistema cumple actualmente, y las nuevas que se desean integrar.

4.1. Objetivos del Proyecto

Objetivo General

Desarrollar un prototipo de software libre para el módulo principal de un Sistema de Gestión Documental (SGD) bajo una arquitectura de microservicios para el almacenamiento, creación y manejo del flujo de los documentos electrónicos dentro del sistema ArgoGPL para la fundación Correlibre, con el fin de brindar a la comunidad de software libre una herramienta que haga más eficiente el manejo de procesos respecto al anterior sistema de gestión en cuanto usabilidad, escalabilidad y mantenibilidad .

Objetivos específicos

1. Identificar y establecer los requerimientos funcionales y arquitectónicos del módulo correspondiente del sistema de gestión documental mediante historias de usuario para formular el diseño del sistema.
2. Definir una arquitectura basada en microservicios a partir de los requerimientos establecidos para el módulo y su integración al SGD, al igual que su interacción con el sistema a partir de los requerimientos identificados.
3. Construir un prototipo funcional del módulo diseñado para el sistema de gestión documental, mediante las tecnologías elegidas, con el fin de probar su implementación y sus características en cuanto a usabilidad, escalabilidad y tiempos de respuesta.
4. Evaluar la mantenibilidad, la usabilidad y la escalabilidad del módulo construido, en comparación con la versión actual del SGD Argo/OrfeoGPL, mediante el desarrollo de un plan de pruebas controladas para validar una mejora significativa en el mismo.

4.2. Descripción del Artefacto

El diseño del sistema se planteó inicialmente con el fin de contar con una estructura que sirva de base para cualquier sistema de gestión. En pocas palabras, que su arquitectura, relacionamiento y distribución sean fácilmente escalables y parametrizables para cualquier tipo de entidad que se quiera gestionar. Esto, entre otras cosas, se decidió teniendo en cuenta que la ley define documento electrónico como cualquier tipo de archivo digital (incluyendo además PDF, archivos de texto, archivos multimedia, correos electrónicos, etc.). Con ese concepto, se comenzó a diseñar tres módulos que contarán con funciones básicas pero necesarias en la mayoría de procesos de gestión documental presentes actualmente.

Inicialmente, se cuenta con un módulo organizacional, encargado de manejar la estructura en la que se encuentra la organización en la que se implementa el sistema. Aquí se describen los tipos de usuarios, estructuras jerárquicas, permisos con los que cuenta cada uno y datos de la organización (u organizaciones) que harán uso del mismo sistema.

Como núcleo del sistema se encuentra el módulo de registros, encargado de todos los procesos de radicación y gestión de archivos electrónicos del sistema. Cada registro (radicado) cuenta con la información necesaria por cada cliente, esto gracias a un módulo de metadatos diseñado específicamente con este propósito.

El módulo de metadatos es un módulo que le permitirá a cada usuario (cliente) definir los datos correspondientes a las entidades (documentos electrónicos) que va a manejar dentro de su organización. Para esto, y teniendo en cuenta la definición de documento electrónico dada en secciones anteriores, el módulo de metadatos es un motor de formularios que permite la creación de formularios personalizados a partir de las necesidades del cliente, así, por medio de estos, cada persona podrá definir los campos con los que se contará en los formularios para radicación, y se podrá controlar fácilmente y de forma mucho más parametrizada los tipos de datos y la información que se manejara en cada registro.

Entrando al detalle de este proceso, se destaca que el módulo de registros cuenta con una serie de submódulos que también definen la identificación de los mismos. Esto quiere decir que hay una definición de los tipos de registro, que haciendo un paralelismo pueden ser un tipo de registro de entrada, salida o comunicados oficiales, por ejemplo; Unos trámites y servicios, o procesos, que ayudarán con el uso de ciertos registros para unas u otras situaciones; Y la modificación de la máscara, para guardar personalizadamente los identificadores de los registros.

4.3. Componente de Análisis: Descripción de la Transformación Esperada del Contexto

La actualización del artefacto tendrá un efecto en los hábitos y en las prácticas, tanto de los desarrolladores de la empresa productora del software, como en el usuario final y los directivos que deseen implementar estos sistemas en sus compañías. En primer lugar, al tener una actualización del sistema con las características descritas anteriormente permitirá reducir los tiempos de desarrollo al tener mejores herramientas para programar, lenguajes más actualizados, de más alto nivel, y arquitecturas más sofisticadas, mejorando las condiciones para los desarrolladores, así como para quienes usan el software, puesto que este se convertirá en uno más mantenible y escalable a largo plazo. Del mismo modo, esta reducción en tiempos de desarrollo y mantenimiento mejoraría los tiempos de construcción de nuevos requerimientos en el sistema. Esto para los clientes que tengan la necesidad de realizar cambios y personalizaciones dentro del mismo implican una gran ventaja a la hora de implementarlo dentro de sus empresas, puesto que para el cliente no solo será una reducción en sus tiempos de espera sino en el tiempo para la puesta en producción y será un beneficio

más pronto para su organización al poder integrar todas las ventajas que este sistema implica [15]. Así mismo, el mantenimiento del sistema se verá afectado positivamente, trayendo no solo más facilidad a la hora de realizarlo por parte de los desarrolladores, sino que para el cliente implica menores costos en mantenimiento y menor probabilidad de fallos en el medio y largo plazo. Finalmente, con un cambio de arquitectura se abrirá aún más la posibilidad de integrar con otros sistemas como lo pueden ser los de seguridad, los JBPM y hasta el uso bases de datos no relacionales. **[Ver Anexo No. 2]**

La solución propuesta tendrá un gran impacto en la alta resistencia al cambio presente en las organizaciones al intentar introducir esta clase de sistemas dentro de una empresa, afianzará la confianza en los procesos actualizados, haciendo uso de las nuevas tecnologías y sistemas propuestos, como el presente. De la misma manera, al presentar el sistema ante ellos, se mostraría como uno simple e intuitivo, sin dejar de ser robusto y afín a las necesidades de la organización. Esto simplificará los procesos de flujo de control de documentos, su accesibilidad, y generará confianza frente a la seguridad de los archivos y quien tiene acceso a estos [3, 9].

De la misma manera, como solución a una necesidad de personalización del sistema para atacar directamente la alta reticencia al cambio, esta propuesta permitiría adaptar el sistema más fácilmente, al contar con una arquitectura distribuida en microservicios, reduciendo el acoplamiento de la aplicación y permitiendo no solo desarrollar la aplicación de forma procedural cumpliendo con principios de bajo acoplamiento y alta cohesión en una mayor medida, sino a su vez, abriendo la posibilidad a una fácil integración con otras aplicaciones al sistema. Dentro de esto, también hay que tener en cuenta que cada uno de los desarrollos tomará menos tiempo, como se mencionó anteriormente, del que requeriría, teniendo en cuenta que, al ser un sistema modularizado, permite hacer desarrollos pequeños y rápidos, sin tener gran impacto directo en el todo. Esto en conjunto con procesos de desarrollo mucho más estandarizados, organizados y estructurados, lo volvería uno mucho más eficiente.

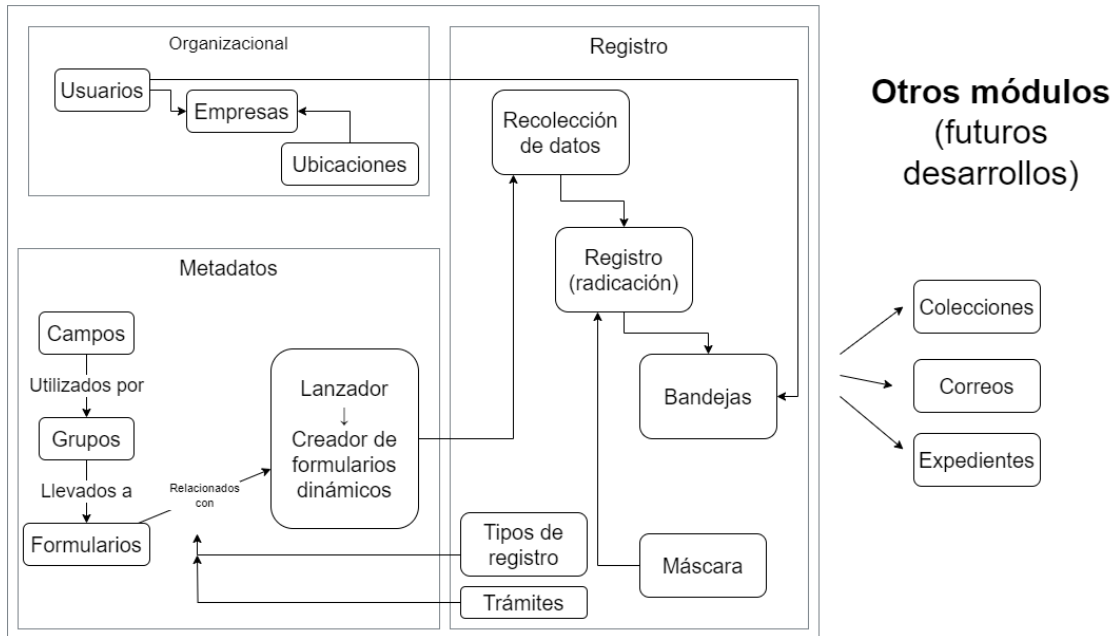
Además, teniendo en cuenta la amplia presencia del sistema en entidades públicas, se hace mucho mayor la necesidad garantizar la transparencia y seguridad con los documentos especialmente para cumplir las necesidades del ciudadano, quien además tiene el derecho a obtener este tipo de informaciones [5], al igual que el cumplimiento de las diversas leyes dictaminadas por el Archivo General de la Nación, entre otras.

Finalmente, esto va a tener múltiples repercusiones en el medio, puesto que la empresa podrá llevar a cabo una correcta actualización de su flujo y control documental, y este proceso que decidió emprender sería una correcta inversión de su tiempo y dinero. De igual medida, este cambio representa una apuesta a poder mejorar la reputación que tanto su sistema como la misma organización distribuidora del software (Fundación Correlibre) poseen frente al mercado en el que se mueven.

En el Gráfico 2 se mostrará esta estructura descrita, teniendo en cuenta otros módulos planteados para la continuación del sistema, como lo son las colecciones, que entre otras cosas trabajarán con los metadatos creados y documentos asociados al registro.

Gráfico 2. Descripción del módulo

Core (Módulo desarrollado)



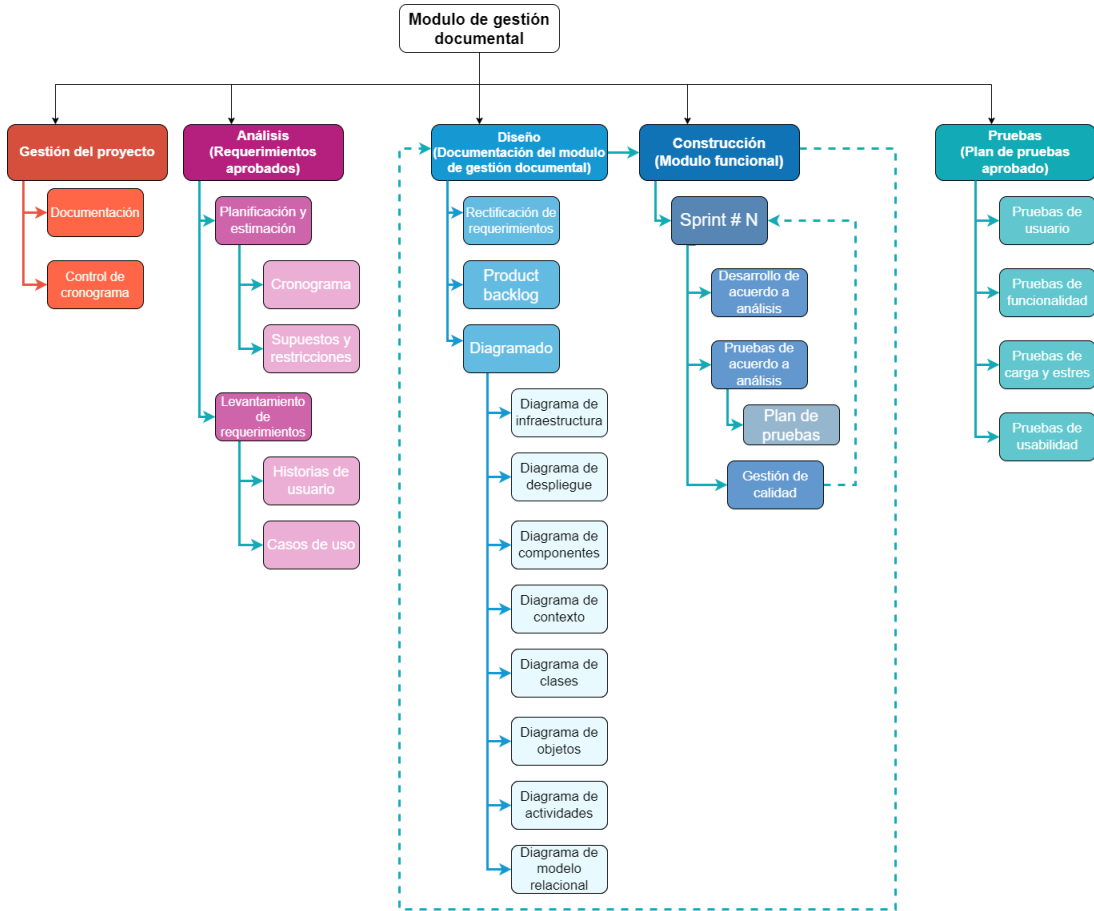
5. DISEÑO METODOLÓGICO

En esta sección se describe el Esquema de Desglose de Trabajo que se siguió durante el proyecto, el marco de trabajo bajo el cual se llevó a cabo el proyecto que fue SCRUM y el cronograma que se siguió desde el inicio del mismo.

Esquema de Desglose de Trabajo (WBS)

En el Gráfico 3 se encuentra la Estructura de Descomposición de Trabajo (EDT) diseñado para el Módulo de Gestión Documental para ArgoGPL donde se organizan los paquetes de trabajo planeados hasta el momento para el desarrollo del proyecto.

Gráfico 3. Esquema de Desglose de Trabajo



La metodología de trabajo será basada en SCRUM, ya que éste es el modelo de desarrollo iterativo e incremental implementado en la organización como parte del proceso de certificación Capability Maturity Model Integration (CMMI), y se requiere para llevar a cabo el desarrollo del proyecto. Así mismo se resalta la comunicación con el cliente como una piedra fundamental en el desarrollo de los productos para cumplir con las expectativas planteadas, dando un completo seguimiento al funcionamiento del software, pues el desarrollo deberá ser escalable y parametrizable para suplir las necesidades específicas de cada uno de sus clientes.

El marco de trabajo SCRUM da las herramientas para la construcción, desarrollo y despliegue de un producto tecnológico donde se interactúa constantemente con el ambiente y sus necesidades; en este proyecto se tomará en cuenta el conocimiento de temáticos y desarrolladores que tienen amplia experiencia en la implementación de SGD en varias entidades a nivel nacional, esto combinado con las nuevas tecnologías y arquitecturas, permitirá no solo suplir las necesidades de un SGD sino evolucionar para hacerlo más parametrizable y dirigido al usuario [27].

En la parte del desarrollador se tienen múltiples ventajas:

La definición de Sprint con el cual se busca que todo el equipo esté consciente de lo que se necesita específicamente en cada lapso de tiempo durante el proyecto, esto se logra implementando un modelo de reuniones de planeación semanal para definir las tareas a resolver en esa semana y de reportes diarios para hacer un control de seguimiento al avance en los desarrollos, donde se

responden las tres preguntas claves de ¿Que hice ayer? ¿Qué haré hoy? ¿Tengo algún impedimento?, con el fin de apoyar el desarrollo constante del producto y la gestión del proyecto.

Otro elemento claro es la comunicación con el cliente, esto impulsa que el objetivo de cada sprint se cumpla satisfactoriamente, y al mejorar la relación con el mismo, se puede entender con mayor certeza sus requerimientos aumentando la precisión con que se desarrollan estos con respecto a sus gustos e intereses.

Cronograma

Así, dentro de la ejecución del proyecto y haciendo uso de la metodología de trabajo se llevó a cabo una serie de Sprints en los cuales se realizó el desarrollo de los elementos que componen el núcleo del sistema, o en otras palabras el módulo principal que sirve de base para el sistema general. En la Tabla 3 se muestra una versión general de este cronograma en el cual se listan dichos elementos.

Tabla 3. Cronograma del proyecto

# Sprint	Nombre del Sprint	Fecha de inicio
Sprint 0	Parametrización inicial y creación de interfaz base	25 de Septiembre de 2022
Sprint 1	Módulo multiempresa	19 de Octubre de 2022
Sprint 2	Módulo Divipola y Complementos	17 de Noviembre de 2022
Sprint 3	Ajustes y afinamiento	13 de Diciembre de 2022
Sprint 4	Estructura Orgánica y Tipos de Registro	27 de Diciembre de 2022
Sprint 5	Creación de interfaz de Módulo de usuarios, spikes JBPM y Formularios dinámicos	23 de Enero de 2023
Sprint 6	Trámites y Servicios, Metadatos y Generador de Secuencias	1 de Febrero de 2023
Sprint 7	Servicio de Registro, Módulo máscara y Módulo de Secuencias, Spike KeyCloak	22 de Febrero de 2023

Sprint 8	Actualización a Vue3 y Spikes (Swagger y KeyCloack)	8 de Marzo de 2023
Sprint 9	Bandeja de Tareas y correcciones de actualización a Vue3	22 de Marzo de 2023
Sprint 10	Banco de Campos, Módulo de Metadatos y Librería Enmascaramiento	11 de Abril de 2023
Sprint 11	Módulo de Formularios y ajustes del sistema	24 de Abril de 2023

Para información más detallada del cronograma por favor remítase a ver **Anexo No. 9**

Variables a medir

En la Tabla 2 se puede observar las variables a ser medidas en el proyecto para medir el impacto y satisfacción al concluir el mismo.

Tabla 2. Variables a medir

Variable	Definición	Medida	Impacto esperado
Escalabilidad	Refiere a la capacidad del sistema de adicionar funcionalidades y realizar cambios al sistema.	Análisis de código fuente estático por medio de herramientas analizadoras de calidad de código.	Reducción en los tiempos usados para la implementación de nuevas funcionalidades en comparación con los desarrollos en el sistema antiguo.
Usabilidad	Refiere a la capacidad de aprendizaje, satisfacción y tiempo de comprensión en el cómo se realiza una tarea según la definición de usabilidad de Walter Sanchez en [28].	Percepción del usuario, proveniente de encuestas en escala Likert	Existen usuarios que navegan frecuentemente por una misma página o que realizan las mismas acciones de manera recurrente; Se espera que con estos cambios se les pueda facilitar el acceso de forma más rápida y directa a esas acciones.
Mantenibilidad	Facilidad con la que un sistema puede ser modificado para corregir fallos, mejorar el rendimiento o adaptarse a un cambio en su entorno. [8]	Estimaciones de tiempo y esfuerzo (puntos de función) por parte del equipo desarrollador para implementar nuevas	Mejora en niveles de calidad en el código, lo cual permitiría menores tiempos de desarrollo por parte de los desarrolladores y ser más parametrizable para el

		funcionalidades.	usuario final.
--	--	------------------	----------------

Pruebas técnicas con desarrolladores

Estas pruebas se llevarán a cabo con el objetivo de determinar y comparar tiempo y esfuerzo que se emplea para un desarrollo en Orfeo/origen y Argo/OrfeoMSA.

Procedimiento

Se presentan los ambientes de desarrollo sobre los que se lleva a cabo la implementación de los requerimientos.

Se presentan 2 requerimientos diferentes. El primero de ellos es un requerimiento menor (Ajustes o adición de funciones sobre código ya construido) y el otro un requerimiento más amplio (Considerado como una nueva funcionalidad o módulo completo).

Se lleva a cabo una ceremonia de Planning Poker para poder determinar por medio de puntos de función la cantidad de esfuerzo y tiempo que toma cada uno de los requerimientos, en cada uno de los ambientes. De esta forma se obtienen 4 estimaciones, una para cada requerimiento en cada ambiente de desarrollo.

Finalmente, con los resultados obtenidos se realiza un análisis con las respectivas conclusiones de lo que se obtuvo.

Requerimientos

Requerimiento #1 (Requerimiento menor)

Yo como administrador requiero poder asignarle a un usuario una dependencia desde la interfaz de usuarios tanto en el proceso de creación como de edición. Debe haber una lista desplegable con las dependencias disponibles para poder elegir desde la interfaz.

Pre requisitos:

La dependencia ya debe haber sido creada.

Requerimiento #2 (Requerimiento general)

Yo como administrador del sistema, requiero de un módulo que me permita crear las dependencias pertenecientes a cada empresa creada en la herramienta, basados en sus estructuras orgánicas (Plana o Jerárquica), teniendo en cuenta los niveles de jerarquía.

El formulario de Creación de dependencias debe contar con los siguientes campos:

- Código de la dependencia
- Nivel de subordinación: (Lista desplegable. ejemplo: Gerencia/Subgerencia/Dirección/Subdirección, Coordinación, Jefatura)
- Nombre de la dependencia

- Sigla de la Dependencia. Alfanumérico.
- Dependencia Padre: (Lista de código de dependencias activas, no es obligatorio)
- Fecha de Cierre de la dependencia (No obligatorio)

Criterios de aceptación

- El sistema NO debe permitir que existan dependencias con información (nombre de la dependencia y/o código) duplicada
- El sistema debe informar cuando se intente ingresar información duplicada en un nuevo registro de dependencia.
- El sistema debe notificar al usuario el resultado de cualquier acción que este realice, creación, edición, activación e inactivación.
- Solo el usuario con permisos puede crear, editar, activar e inactivar los registros de las nuevas dependencias (administrador).
- Una vez creada la dependencia, se debe visualizar los campos capturados en una tabla o una lista.

Pruebas de usabilidad

Para la medición de los resultados en términos de la usabilidad percibida por parte de los usuarios finales se realizó un cuestionario con escalas Likert en el cual se presentan funcionalidades básicas de una interfaz del sistema origen en una de sus últimas versiones, y se compara una funcionalidad similar en el sistema construido dentro del proyecto, para así comparar las respuestas con unas mismas variables.

Diseño de las pruebas de usabilidad

En cuanto al diseño de las preguntas, para medir dicha percepción se formularon distintos aspectos basados en los principios de usabilidad definidos en el Usability Body of Knowledge [28] y en las normas ISO/IEC 9126 e ISO/IEC 9241. Estos aspectos incluyen las definiciones de estética, comodidad, coherencia, completitud y visibilidad, a los cuales se otorga un nivel según el encuestado considere dependiendo de la funcionalidad presentada para esa parte del cuestionario.

Además, para complementar los aspectos, es de resaltar que estos se tomaron y se le presentaron al usuario como: 1. Estética como su opinión respecto a la atractividad de la interfaz mostrada; 2. Comodidad, como la distribución de los componentes y la capacidad intuitiva que tuvo para encontrarlos en ella; Coherencia como la percepción de si considera si el estilo concuerda con el propósito del sistema; Completitud, como la disponibilidad de las opciones necesarias para dicho módulo; y la visibilidad como el diagnóstico sobre los posibles campos sin llenar, el informe de error en caso de haberlo, y la suficiencia de información sobre el estado del sistema y del proceso.

6. RESULTADOS Y DISCUSIÓN

Tras una planificación durante julio, el proyecto inició oficialmente a inicios de agosto de 2022, donde se llevó a cabo una configuración de las herramientas y componentes básicos a utilizar siguiendo la arquitectura descrita en el **Gráfico 4**, sobre la cual se inició el levantamiento de requerimientos para el sistema, requerimientos que puede ver en el **Anexo No. 12**. Con este proceso finalizado, la fase de diseño se inició en donde a grandes rasgos se empezó por los microservicios de backend, el componente que se encargaría del front y la base de datos.

Sin embargo, por motivos de tiempo y los objetivos definidos para este periodo, aún no se configuraron las conexiones con la API gateway, Keycloak y parte del despliegue con Docker. Los esfuerzos se enfocan en construir una base de personalización para cualquier sistema de gestión documental, permitiendo al usuario realizar cambios en el comportamiento del sistema sin la

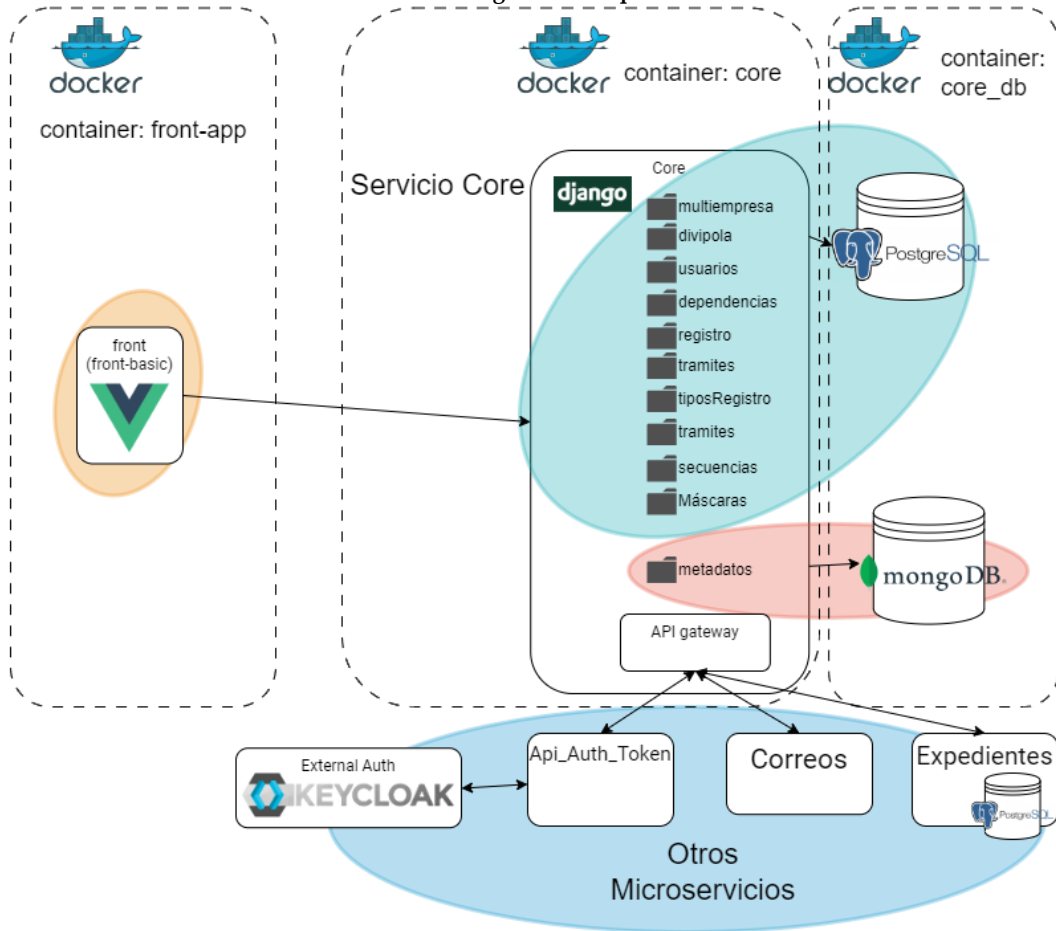
necesidad de la presencia de un desarrollador.

Con esto presente, la fase de diseño se fue llevando a cabo de manera iterativa en conjunto con la fase de desarrollo, y de la cual se obtuvo un Documento de Arquitectura [Ver Anexo No. 13] que permitiera documentar fácilmente las principales funcionalidades y decisiones a nivel de diseño y arquitectura que fueron tomadas a lo largo del proyecto. Dentro de este en las secciones 8 y 9 se encuentra documentación relacionada a diagramas del sistema y decisiones de tecnologías tomadas para el sistema.

Es de resaltar que luego de ello se continuó con los requerimientos básicos que dan forma al sistema web como los elementos básicos de la página. Estos incluyen una página de inicio, unos menús superiores y laterales que permitirían la navegación entre funcionalidades, acceso, visibilidad del usuario y un botón para cerrar sesión [Ver Anexo No. 4]. Asimismo, se realizó la construcción de los mismos teniendo en cuenta unas paletas de colores distribuidas y la capacidad de ajuste para diversos tipos de pantalla, de modo que se procura por el funcionamiento en distintos dispositivos y la clara diferenciación de las distintas opciones ofrecidas por el sistema. Esto incluyendo el redimensionamiento y redistribución de dichas partes, al tiempo que se generaron los respectivos manuales de usuario y técnico [Ver Anexo No. 11, y No. 17].

Una vez teniendo estos elementos, se procedió al avance sobre los módulos principales. En un inicio se propuso como objetivo principal el desarrollo del módulo de radicación, sobre el cual se desarrolló un formulario inicial para las pruebas de conexión con el módulo correspondiente en el *back-end* [Ver Anexo No. 5, y No. 6]. Sin embargo, al momento realizar este módulo se comprendió la necesidad de dar prioridad a otros que terminan siendo componentes principales del sistema, hasta el punto de ser parte del proceso de radicación. Dentro de estos componentes se realizaron desarrollos sobre el módulo de gestión de empresas [Ver Anexo No. 7] y ubicaciones y direcciones, al cual se le conoce como el nombre de *divipola* [Ver Anexo No. 8] al igual que todo el módulo de metadatos. Estas decisiones de comenzar a trabajar sobre los módulos de metadatos y organizacional fueron tomadas por el equipo de desarrollo en conjunto con el cliente de manera incremental y evolutiva, como lo establece el marco de trabajo, a lo largo de los diferentes sprints tanto en las ceremonias de *Sprint Review* como en las *Sprint Planning* que eran realizados [Ver Anexo No. 14, y No. 15]. Así fue como con el cliente se fueron estableciendo tanto las necesidades como la forma de proceder a partir del diseño que se conseguía del sistema.

Gráfico 4. Diagrama de arquitectura.



Finalmente, de manera similar se realizaron otros módulos que son el sustento de los datos que se usan en todos los procesos, como lo son las dependencias y tipos de trámites. Sin embargo, en este punto se tomó también la decisión de reestructurar el concepto de radicado para que abarcara lo ahora definido como documento electrónico, llamándolo un registro. Así, en cuanto al registro, sobre todo se agregó lo correspondiente a las máscaras (organización del número de referencia), el manejo de los tipos de registro, e incluso se empezó con un módulo de metadatos, el cual incluye administración de campos y formularios para que en un futuro estos registros aceptan todo tipo de documentos, con sus respectivas variables. Este proceso definido durante el avance del proyecto se puede ver en el *Release Plan* [Ver Anexo No.9].

Dentro de las validaciones y pruebas realizadas, una de las más importantes para poder medir la mantenibilidad del sistema, y compararla con el anterior, fue la implementación de Análisis de Código Estático, un proceso que permite medir el cumplimiento y seguimiento de buenas prácticas de desarrollo y estándares de codificación en el proyecto. Para este proyecto fueron usados PyLint para el back-end y ESLint y Prettier para el front-end, entre las múltiples herramientas existentes actualmente para este fin.

Paralelamente, se realizaron las mismas pruebas sobre una de las versiones de Orfeo anteriores haciendo uso de Yasca, como se describió, con el fin de comparar los resultados con la versión que se está construyendo. Para esta comparación, cabe destacar, se usó el criterio de que el sistema anterior es mucho más robusto y con muchas más funcionalidades (código), por lo que se realizó la prueba sobre ciertos módulos específicos, considerados como los más importantes como el módulo de radicación. Se encontraron falencias en el cumplimiento de estándares de codificación y buenas

prácticas como posibles vulnerabilidades de seguridad o problemas de optimización en el código [Ver Anexo No. 10] lo cual da evidencia de que este sistema, aun con sus años de desarrollo y su robustez, aún posee varios problemas causados por las razones mencionadas anteriormente.

Con esto presente, se realizaron las pruebas técnicas con dos desarrolladores, en donde, como se explicó en secciones pasadas, se pretendía conocer una posible mejoría en la cantidad de tiempo y esfuerzo necesario para realizar ciertas implementaciones sobre el sistema.

Tabla 3. Pruebas técnicas con desarrolladores.

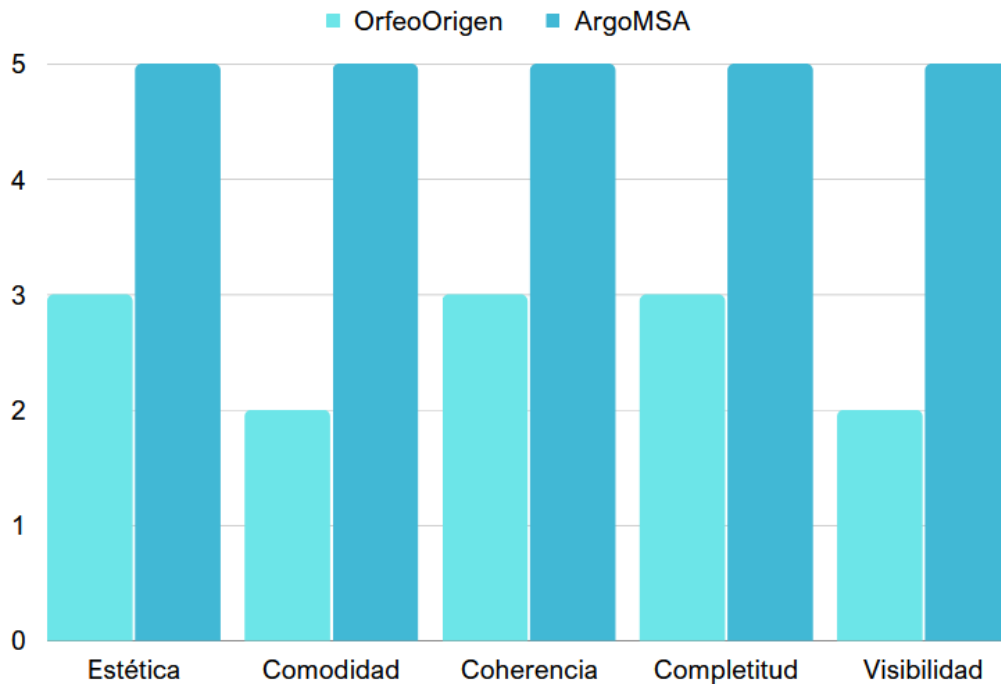
Sistema	Estimación	Requerimiento 1	Requerimiento 2	Desarrollador
Orfeo	Puntos de función	1 punto	89 puntos	Desarrollador 1
		4 puntos	89 puntos	Desarrollador 2
	Tiempo	3 días	6 semanas	Desarrollador 1
		2 días	5 semanas	Desarrollador 2
Argo MSA	Puntos de función	1 punto	89 puntos	Desarrollador 1
		1 punto	89 puntos	Desarrollador 2
	Tiempo	2 días	4 semanas	Desarrollador 1
		1 día	4 semanas	Desarrollador 2

En la Tabla 3 se pueden ver plasmadas las respuestas de cada uno de los desarrolladores respecto a los dos requerimientos tanto en el sistema Argo MSA cómo en el Orfeo, destacando que dentro de este ejercicio se tomaron las opiniones tras sus respuestas para sustentar los resultados, con las cuales se llegó a unos consensos en los cuales describieron, sobre todo, que no veían una gran diferencia en cuanto a dificultad con este tipo de requerimientos, pero sí la posibilidad de reducción de tiempos de desarrollo [Ver Anexo No. 16]. Esto teniendo en cuenta que actualmente muchas de las partes de dificultad son dadas, además de la construcción, por el diseño y pruebas que extienden propiamente la duración de los desarrollos.

Adicionalmente, dentro del espacio se propició la discusión en cuanto a las herramientas utilizadas tomando como base estos requerimientos. Principalmente, se notó que, aunque sí hay ventajas en lo escogido para el proyecto, se pueden todavía contemplar otras técnicas con las cuales un proyecto podría hacer mayor énfasis en las pruebas unitarias y en herramientas distintas a Django que no limiten los microservicios a unos modelos relacionales. Sin embargo, estas sesiones también permiten vislumbrar una tendencia a tratar enfocar muchos de los esfuerzos en el trabajo dentro del grupo, ya que el uso de estas propuestas es lo que como conjunto lleva a la mejora de los tiempos y estándares.

A su vez, se realizó la encuesta de usabilidad anteriormente nombrada, donde se aplicó a una persona.

Gráfico 5. Encuesta usabilidad.



El Gráfico 5 muestra los resultados de la encuesta aplicada al cliente, evidenciando un cambio en la percepción en comparación con el sistema anterior. Es importante destacar que estos resultados podrían estar influenciados por el hecho de que el cliente ha estado involucrado en cada una de las entregas parciales, donde estas han seguido las directrices establecidas por las necesidades identificadas. Por lo tanto, es posible que esta opinión sea sesgada hacia el producto y sea necesaria una futura evaluación con personas ajenas al contexto.

7. CONCLUSIONES

Durante el desarrollo del proyecto se llevaron a cabo cambios de framework como de arquitectura en el back-end en el ambiente de desarrollo, que a gran escala demostraron que el sistema presenta buenas características a nivel de escalabilidad y mantenibilidad, a raíz de la arquitectura y las prácticas implementadas a lo largo de su desarrollo. Por un lado, a nivel de escalabilidad, se compararon los tiempos de desarrollo de requerimientos en el sistema antiguo con los del sistema actual en pruebas validadas por desarrolladores de la organización que han trabajado sobre ambos sistemas, notando una reducción en los mismos. Del mismo modo, además de la eficiencia, también se nota una mayor facilidad tanto en la implementación de nuevas funcionalidades como en la integración con sistemas externos.

Adicionalmente, dentro de cambios realizados emergentes del propio desarrollo del proyecto, hubo algunos como la migración del framework *VueJS* de la versión 2 a la 3, el cual, aunque fue bastante extenso y requirió de una actualización de la librería *Vuetify*, no fue nada traumático para el sistema y para su desarrollo. Del mismo modo, por decisiones de comodidad en el ambiente de desarrollo, se probó hacer el cambio de que el back-end del sistema fuera montado en una arquitectura monolítica, cambio que no presentó mayor dificultad gracias, en gran parte, a la arquitectura que este sostenía.

De igual forma, en cuanto a la mantenibilidad, entre otras características, se encontró una mejoría en cuanto a estándares, buenas prácticas y uso de procesos de ingeniería correctos a comparación del

sistema antiguo. Esto se comprobó gracias a pruebas de Análisis de código estático realizadas sobre ambos sistemas, que permitieron determinar la calidad de código y la implementación de buenas prácticas de desarrollo y validar una mejoría en la mantenibilidad del sistema.

Dentro de los objetivos planteados, uno de los principales era el de la mejora en la usabilidad del sistema. Se hizo uso de ciertas herramientas de diseño como lo fue Figma, para poder llevar a cabo un proceso de prototipado y diseño centrado en el usuario. Aun así, no se percibió un cambio o una mejoría en la percepción del sistema por parte del cliente, se notaban las mismas interfaces y los mismos diseños, pero con diferentes colores o en otro orden. Para este punto se puede concluir que hace falta un proceso de diseño de interfaz de usuario y diseño de experiencia de usuario mucho más riguroso, definido y con personas con más experiencia que puedan brindar un efecto mucho más significativo en este ámbito.

8. LECCIONES APRENDIDAS Y TRABAJO FUTURO

Este proyecto dio paso a múltiples aprendizajes y crecimiento a distintos niveles. El marco de trabajo posibilita un desempeño particular entre el equipo, donde el conjunto de conocimientos permite resolver las tareas delegando responsabilidades autónomamente, mientras cada uno cumple con su papel. Esto ya que el equipo termina compenetrado a base de la comunicación y la responsabilidad.

Sin embargo, alcanzar los objetivos propuestos de la manera más óptima requiere un esfuerzo particular en estos aspectos. Dentro de este proyecto en el equipo de trabajo se pudo ver ese proceso evolutivo desde un primer momento, sin el nivel de comunicación y planeación óptima, hasta uno en el que empezó a ser una fortaleza el apoyo entre miembros y hubo acuerdos claros sobre el plan y resultados con el cliente. Esto, claramente se ve reflejado en una mejor formulación de requerimientos, desarrollos más pulidos y mejor llevados a cabo, tiempos de desarrollo menores y un trabajo de mucha más calidad.

De igual forma, la organización y la estandarización de procesos y procedimientos internos es importante. Durante el desarrollo que se hizo en conjunto con la empresa, se llevó a cabo un proceso de certificación de CMMi nivel 3, lo que dio paso a una evaluación de los procesos internos que se llevaban a cabo en la empresa. Hay que destacar que esto permitió también establecer de forma mucho más clara cada una de las responsabilidades, roles y papeles que cumplía cada uno dentro del equipo. El hecho de estandarizar, que se dejen claras las responsabilidades y asignaciones de cada uno, y que se tenga un orden para poder cumplir con cada una de las tareas, permite mejorar el desempeño del equipo, y del proyecto en general, convirtiéndolo en uno de mucha más calidad y efectividad.

Dentro de estos aprendizajes, una de las cosas a resaltar fue darse cuenta de que la etapa de planeación en verdad cumple un periodo importante dentro de un proyecto y requiere darle su tiempo respectivo. Una correcta etapa de planeación, permite asentar los cimientos de un proyecto estable, viable y óptimo. Procesos de levantamiento y refinación de requerimientos, planeación, estimaciones y gestión de riesgos, entre otros, es lo que permite que los demás procesos de “construcción” puedan salir adelante de la mejor manera posible. De la misma manera, la correcta documentación, así como aplicar un conjunto de buenas prácticas, es lo que hace que un proyecto de software se vuelva uno de más calidad y mucho más rentable tanto para el cliente como para la empresa que lo produce.

Es importante crear sistemas que no estén destinados únicamente a su objetivo principal, sino que sean fácilmente integrables con sistemas externos, que permitan crear una herramienta mucho más completa, a la vez que modularizada. Esta necesidad parte a partir del hecho de que hoy en día los sistemas tienden a exponer múltiples servicios consumibles que permiten crear sistemas más robustos y complejos, pero a su vez con más funcionalidades, dando solución a las necesidades, y sin tener que crear un sistema desde ceros, con una arquitectura mucho más acoplada, que da paso

a los múltiples problemas de complejidad que ya conocemos.

Ahora, desde otro punto de vista, existe la necesidad de cambiar el concepto que se tiene sobre los documentos, en tanto que estos incluyen desde correos hasta videos, y por lo cual se requiere poder tener un sistema que se adapte fácilmente o de mejor manera a las necesidades del medio. En lo que respecta a trabajos futuros, se encuentra la idea de crear un sistema fácilmente parametrizable, que permite no gestionar sus documentos electrónicos, sino cualquier tipo de material que se considere "insumo" para el cliente. Hablamos de pasar de un SGD (Sistema de Gestión Documental) a un SG (Sistema de Gestión) que, de una manera parametrizable, permite llevar a cabo un proceso de gestión sobre una serie de recursos, sin tener que realizar una serie de cambios considerables.

Incluso en este proyecto se vio la necesidad de expansión para otros intereses de la empresa, que finalmente integran herramientas en común con el SGDEA. Hablamos ya de que, al tener un sistema fácilmente integrable con otros, gracias a su arquitectura, puede formar parte de un sistema más grande, como un ERP, como posible producto para la empresa, uno más completo, que a nivel empresarial le brinda múltiples soluciones a las necesidades que los clientes encuentran hoy en día.

La implementación del análisis de código estático, por su parte, permitió identificar múltiples beneficios en la mantenibilidad de un proyecto de software, puesto que se tendrán estándares a seguir, validados por la herramienta, y que mantendrán un código limpio, fácil de leer, libre de errores de sintaxis, y que ahorrará tiempo proporcionando una guía de estilos.

Finalmente, cabe resaltar que este sistema fue diseñado para poder ser la base de un sistema mucho más completo y robusto. Dentro de los trabajos que se espera realizar con este está su integración con sistemas de diversas índoles que permitan conformar un ERP propio de la empresa y de software libre. De la misma manera, poder hacer uso de la gran cantidad de información que se posee para la automatización de procesos por medio de IA (Inteligencia Artificial) en conjunto con la integración de herramientas como BPM (Business Process Management) para el control de flujos de trabajo y la formación de una herramienta mucho más eficiente y completa. Cabe resaltar que el sistema seguirá siendo trabajado por parte del equipo desarrollador por al menos 6 meses más para poder finalizar la integración con el módulo de radicación y que el flujo pueda ser finalizado.

9. REFERENCIAS

[1] S. E. López Riquelme, "Document Management: good practice to reduce paper consumption in support of sustainable development," vol. 7, no. 1, 2019.

[2] S. R. Gonzales Sánchez, "Eficiencia de los sistemas de gestión documental para el cuidado del medio ambiente y el nivel de reducción del uso de papel en la Municipalidad Provincial del Callao," 2018.

[3] N. Castro (Entrevista), 2023

[4] Decreto 2609 de 2012 [Presidencia de la República]. Por el cual se reglamenta el Título V de la Ley 594 de 2000, parcialmente los artículos 58 y 59 de la Ley 1437 de 2011 y se dictan otras disposiciones en materia de Gestión Documental para todas las Entidades del Estado. 14 de diciembre de 2012. D.O. No. 48647.

[5] Tirenti, C. (2019) Maestría en Administración y Políticas Públicas, Repositorio Digital san andrés: Página de Inicio.

Available at: <https://repositorio.udesa.edu.ar/jspui/> (Accessed: November 21, 2022).

- [6] van Steen, M., & Tanenbaum, A. S. (2017). *Distributed systems*. Maarten van Steen Leiden, The Netherlands.
- [7] Argo/ORFEOGPL (no date) Correlibre. Available at: <https://correlibre.org/argo-orfeo/gpl/> (Accessed: November 21, 2022).
- [8] "IEEE Standard Glossary of Software Engineering Terminology," in IEEE Std 610.12-1990, vol., no., pp.1-84, 31 Dec. 1990, doi: 10.1109/IEEESTD.1990.101064.
- [9] Programa de Ingeniería de Sistemas, Proyecto Educativo del Programa. Universidad El Bosque, 2019. pp. 16
- [10] Sogamoso Gómez, J. P. (2019). La gestión documental en empresas multinacionales de Colombia. Retrieved from https://ciencia.lasalle.edu.co/sistemas_informacion_documentacion/319
- [11] Por medio de la cual se dicta la Ley General de Archivos y se dictan otras disposiciones. LEY 594 DE 2000. Art.3
- [12] P. Russo, Gestión documental en las organizaciones, *Gestión documental en las organizaciones* (2012) 0-0. Retrieved from https://books.google.com.co/books?hl=en&lr=&id=AOkL0OYJp54C&oi=fnd&pg=PA7&dq=gestion+documental&ots=XvVVSQyVcm&sig=OfJM0x7Q6DxqEfcN8YoxwflvQ50&redir_esc=y#v=onepage&q=gestion%20documental&f=false
- [13] Inicio | Archivo general de la Nación (2016). Available at: https://www.archivogeneral.gov.co/sites/default/files/Estructura_Web/2_Politica_archivistica/PoliticasyPublicasdeArchivo_V2.pdf (Accessed: November 22, 2022).
- [14] Políticas de Archivos (no date) Archivo General de la Nación | Política de archivos. Available at: <https://www.archivogeneral.gov.co/politica/politicas> (Accessed: November 21, 2022).
- [15] L. E. Berdugo Baez, "Análisis de los Sistemas De Gestión Documental que existen actualmente en Colombia que cumplen con la legislación vigente para empresas públicas de acuerdo con el programa de gestión documental," 2016.
- [16] J.H Losada (comunicación privada), 2022.
- [17] MoReq & DLM Forum. (n.d.). MoReq2010. MoReq. Retrieved May 30, 2022, from <https://www.moreq.info>
- [18] M. Puentes Poloche, "Propuesta metodológica para articular la gestión documental con los requisitos de la Ley General de Archivos y la norma técnica internacional ISO 9001:2015," *SIGNOS - Investigación en sistemas de gestión*, vol. 9, no. 2, pp. 81-95, Dec. 2017, doi: 10.15332/s2145-1389.2017.0002.05.
- [19] H. S. Ahmad, I. M. Bazlamit, and M. D. Ayoush, "Investigation of Document Management Systems in Small Size Construction Companies in Jordan," *Procedia Engineering*, vol. 182, pp. 3-9, Jan. 2017, doi: 10.1016/J.PROENG.2017.03.101.
- [20] Y. Huayapa Corrales, R. Pato, and N. Villalba Holzinger, "Digitalización Del Sector Público: Gestión Documental Electrónica Para La Desburocratización," 2017.
- [21] A. Cerrillo Martínez and A. Casadesús de Mingo, "El impacto de la gestión documental en la transparencia de las Administraciones públicas: la transparencia por diseño," *Gestión y Análisis de Políticas Públicas*, pp. 6-16, May 2018, doi: 10.24965/gapp.v0i19.10515.
- [22] Mateus-Coelho, N., Cruz-Cunha, M., & Ferreira, L. G. (2021). Security in Microservices Architectures. *Procedia Computer Science*, 181, 1225-1236. <https://doi.org/10.1016/I.PROCS.2021.01.320>
- [23] Villamizar, M., Garces, O., Castro, H., Verano, M., Salamanca, L., Casallas, R., & Gil, S. (2015). Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud. *2015 10th Computing Colombian Conference (10CCC)*, 583-590. <https://doi.org/10.1109/ColumbianCC.2015.7333476>
- [24] Sprehe, J. T. (2005). The positive benefits of electronic records management in the context of enterprise content management. *Government Information Quarterly*, 22(2), 297-303. <https://doi.org/10.1016/j.giq.2005.02.003>
- [25] "Kong Open-Source API Management Gateway for Microservices". Kong Inc. <https://konghq.com/products/api->

gateway-platform .

[26] "Keycloak". Keycloak. <https://www.keycloak.org>.

[27] Mahalakshmi, M., & Sundararajan, M. (2013). Traditional SDLC vs scrum methodology—a comparative study. *International Journal of Emerging Technology and Advanced Engineering*, 3(6), 192–196.

[28] Sánchez, Walter Ovidio y otros. (2015). La usabilidad en Ingeniería de Software: definición y características. Editorial Universidad Don Bosco

[29] User Experience Professionals' Association (2012). Welcome to the Usability Body of Knowledge. usabilitybok.org

ANEXOS

1. Anexo No. 1: As Is

Imagen del modelo biopsicosocial y cultural en su aspecto de análisis del problema (As Is) referente al sistema antiguo de Orfeo/Argo GPL.

2. Anexo No. 2: To Be

Imagen del modelo biopsicosocial y cultural en su aspecto de análisis de la solución (As Is) propuesta para la reingeniería Orfeo/Argo GPL.

3. Anexo No. 3: Diagrama de arquitectura

Imagen con el diagrama de arquitectura a grandes rasgos, con cada microservicio englobado por su tecnología.

4. Anexo No. 4: Página principal Argo

Página principal con menús y distribución planeada para el sistema.

5. Anexo No. 5: Bandeja Argo

Página con la bandeja de radicados en el sistema.

6. Anexo No. 6: Radicación Argo

Página con un formulario inicial para la creación de un radicado, sólo con unos campos necesarios para dicha funcionalidad.

7. Anexo No. 7: Lista Multiempresa Argo

Página principal para el módulo multiempresa, donde se ve el listado y algunas acciones sobre las empresas en el sistema.

8. Anexo No. 8: Mockup Módulo divipola Argo

Diseño propuesto para el módulo de divipola, el cual permitirá subir el archivo con departamentos y municipios para su uso en el resto del sistema.

9. Anexo No. 9: Release Plan

Entregas planeadas al cliente para esta fase de requerimientos esenciales del sistema (MVP).

10. Anexo No. 10: Resultados Pruebas estáticas Argo Origen

Tabla de resultados del análisis de código estático realizado sobre Orfeo Origen

11. Anexo No. 11: Manual técnico del sistema

Documento dispuesto con el objetivo ser una guía para el entendimiento, utilización y despliegue del sistema ArgoGPL

12. Anexo No. 12: Requerimientos funcionales del sistema

Enlace a la carpeta con los requerimientos funcionales levantados para el sistema.

13. Anexo No. 13: Documento de Arquitectura del Sistema (SAD)

Enlace al documento de arquitectura del sistema.

14. Anexo No. 14: Reuniones de Sprint Planning

Enlace a la carpeta con las grabaciones de los *Sprint Planning* del proyecto.

15. Anexo No. 15: Reuniones de Sprint Review

Enlace a la carpeta con las grabaciones de los *Sprint Review* del proyecto.

16. Anexo No. 16: Entrevista pruebas técnicas desarrolladores

Enlace a la carpeta con las grabaciones de los *Sprint Review* del proyecto.

17. Anexo No. 17: Manual de usuario del sistema

Documento dispuesto con el objetivo ser una guía para el entendimiento y utilización del sistema ArgoGPL