

Gestor de seguimiento y control de versión de procesos de software en la etapa de mantenimiento.

Carlos Armando García Marroquín y Pedro Julián Vargas Mejía

Ingeniería de Software, Fundación Universitaria Compensar

Proyecto fin de grado para Facultad de Ingeniería

José de los Santos Solorzano Suarez

28 de abril de 2023



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.

Resumen

La etapa de mantenimiento es una parte crítica del ciclo de vida del software, que requiere una gestión eficiente para garantizar que el sistema de información continúe funcionando correctamente y se adapte a las necesidades cambiantes de los usuarios. En este contexto, este proyecto de investigación se centró en el desarrollo de un Gestor de seguimiento y control de versión de procesos de software en la etapa de mantenimiento, con el objetivo de mejorar la eficiencia y la calidad del proceso de mantenimiento. La metodología aplicada en este proyecto de investigación incluirá la revisión de la literatura existente sobre la gestión de proyectos de software, la identificación de los requisitos específicos de los equipos de mantenimiento de software y el diseño e implementación de un prototipo para el Gestor de seguimiento y control de versión. El prototipo de software se desarrolló utilizando tecnologías de vanguardia bajo el principio de arquitectura limpia y se sometió a pruebas para evaluar su eficacia y eficiencia. Así mismo, se logró una comprensión más profunda de las necesidades y requisitos específicos de los equipos de mantenimiento de software y una evaluación de la eficacia y eficiencia del Gestor de seguimiento y control de versión de procesos de software en la etapa de mantenimiento.

Palabras clave: Gestión de versiones, mantenimiento de software, novedades del ciclo de vida, arquitectura del software.

Contenido

Introducción 4

Problema 10

Objetivo general 11

Objetivos específicos 11

Método 12

Tipo de investigación 12

Hipótesis 15

Propuesta 15

Técnicas e instrumentos de recolección de información 17

Técnicas de Investigación Documental 17

Técnica de Encuesta 20

Metodología de desarrollo 23

Resultados 29

Resultados técnicos 30

Resultados visuales 31

Discusión 36

Referencias 38

Introducción

En la actualidad existe un crecimiento exponencial en la tecnología, en la cual se ve involucrado el Software, dado que es el que constituye y permite que una necesidad sea convertida en un producto intangible que cumple a cabalidad según las diferentes solicitudes planteadas.

Para ello, se abordarán conceptos claves y de gran importancia en lo que conlleva todo el proceso de construcción y elaboración del software. Tales conceptos nos guiarán sobre los estándares utilizados tanto para el análisis, diseño, implementación, pruebas y mantenimiento del software.

En cuanto a ingeniería de software es importante hablar de la Ingeniería de Software dado que es la ciencia que permite avanzar en todos los aspectos tecnológicos y esto, es parte esencial del mundo tecnológico en el que actualmente vivimos, por ende, es por esto por lo que es necesario citar a Pantaleo, quien tuvo en cuenta las disciplinas que se relacionan en la construcción de productos y la arquitectura de software, refiriéndose a la estructura propia de las aplicaciones y otro tipo de conceptos en cuestión.

También, Pantaleo G. (2015), afirma que el software puntualmente es todo aquello que no es el hardware. El software entre todo su amplio contexto tiene incluido, por supuesto, los programas que rigen el funcionamiento del sistema, pero adicionalmente incluye otros elementos fundamentales, como lo son bases de datos, documentos, o algo intangible, pero de gran importancia como son los procedimientos de operación o mantenimiento periódico.

Además de lo que el anterior autor afirma, es importante mencionar a otros autores que nos brinden claridad y traten de aterrizar un poco más el verdadero significado del concepto ingeniería de software y es por ello por lo que PRESSMAN, R. S., en su libro Ingeniería de Software de Pressman dice que la ingeniería de software se encuentra conformada por un proceso, un ligado de métodos (prácticas) y un arreglo de herramientas que presta y garantiza a los profesionales construir software de cómputo de alta calidad. Además de esto menciona que el software de la computadora es el resultado ejercido a partir de una necesidad que construyen los programadores profesionales y al que una vez cumple todo el ciclo de vida del software, proceden a darle mantenimiento durante un largo tiempo, según se requiera

(PRESSMAN, 2002, p. 1). Para el funcionamiento del software, se ven involucrados programas que se ejecutan en un computador sin importar como tal el tamaño o arquitectura (hardware), dando como resultado la ejecución de estos programas de cómputo y sistemas de información en formatos virtuales.

A raíz de los conceptos de los autores mencionados, se puede deducir que el software genera alto impacto en la tecnología y de todo lo que esto abarca, desde documentos sistematizados como procedimientos de operación, ejerciendo así todo un ciclo de vida del software, que se compone principalmente de las siguientes fases:

- Requerimientos.
- Análisis.
- Diseño.
- Desarrollo.
- Pruebas.
- Implementación.
- Mantenimiento.

Pressman, R. S. propuso que el ingeniero de software cuenta con un punto de vista para identificar que el producto final es la consolidación de los conjuntos de programas, contenido (datos) y otras funcionalidades de otros productos ya finalizados que constituyen el software de la computadora (PRESSMAN, 2002, p. 1). Sin embargo, para una persona en la cual su diario vivir no es la informática, el producto final es el resultado que emite el software elaborado por el ingeniero, generando así una solución óptima según la necesidad en el entorno de su realidad.

Al ser una disciplina muy robusta y con diferentes campos y temas a abarcar, se contextualizará acerca del desarrollo de software y lo que esto conlleva, dado que también es un campo amplio para estudiar.

Por lo tanto, lo que concierne a desarrollo de software, la ingeniería de software comprende varios nichos para el estudio e implementación correspondiente, y por otra parte el desarrollo de software

es la rama fundamental para la extensión del estudio, dado que comprende a aquellas actividades informáticas diligentes al proceso de creación, diseño y despliegue de software.

Es por esto por lo que para percibir aún más acerca de lo que es el desarrollo de software, se asociarán algunas enunciaciones según varios autores:

(1) Citamos a LAINEZ, F. J. quien dice que el software, así como las ideas, procede a madurar con el tiempo, dado que es complejo elaborar la idea de manera completa en el inicio del proyecto para solventar la necesidad planteada (LAINEZ, 2015, p. 3). De lo cual se puede deducir fácilmente que a medida que se vaya desarrollando el proyecto se ira resolviendo el problema.

(2) También es importante mencionar que un proceso de desarrollo de software según NORIEGA, M. R. es una estructura utilizada para la construcción, elaboración y desarrollo de un producto de software. Entre sus grandes rasgos se encuentran el "ciclo de vida" y "proceso de software" (NORIEGA, 2015, p. 6). Cabe decir que existen varios modelos que se utilizan para estos procesos los cuales tienen enfoques distintos para las diferentes tareas y actividades que pueden ser desarrolladas en el transcurso del proceso.

La importancia del desarrollo de software resalta a grandes rasgos, dado que este concepto comprende todo el proceso de elaboración hasta puesta en marcha de un programa, derivando este los diferentes manejos que se efectúan en el proceso:

- Arquitectura del software: Definir claramente las características necesarias con la que contará el software.
- Elaboración: Adaptar una solución adecuada al problema, siguiendo la estructura planteada en la arquitectura del software.
- Documentación técnica y funcional: Relacionar información sobre lo realizado, tanto de manera técnica como comprensible para cualquier persona.

- Versionamiento: Alimentación de versiones de las diferentes fuentes (recursos técnicos) elaborados.
- Pruebas unitarias: Garantizar que la solución cumple a cabalidad y solventa el problema originalmente propuesto.
- Implementación: Poner en marcha (en funcionamiento) el desarrollo de la solución propuesta.
- Mantenimiento: Efectuar correcciones o aplicación de nuevos ajustes sobre el software entregado inicialmente, esto, a raíz de nuevos requerimientos identificados post-implementación.

Se puede comprender que en el desarrollo del software es de suma importancia tener un excelente control en la elaboración de este, es decir, llevar un control de versiones sobre los ajustes implementados a las diferentes fuentes o recursos técnicos creados, esto, para contener una correcta trazabilidad del software para un futuro mantenimiento.

Se contextualizará acerca del control de versiones y lo que esto conlleva, dado que es un ítem para percibir aún más sobre los diferentes manejos que se realizan a las fuentes.

En lo que respecta a control de versión, hace referencia a la gestión que se le da a los diversos y diferentes tipos de cambios que se ejecutan sobre los recursos de un producto o una corrección al mismo. Las versiones, revisiones o las ediciones que se le dan a un producto, son las fases en la que este mismo producto se encuentra en un momento dado según el desarrollo o modificación.

A continuación, se informa acerca de uno de los varios manejos que se le pueden realizar al control de versiones:

- Recién se crea un recurso, este debe ser la versión 1 y debe contener una breve información de la funcionalidad del recurso.
- Cuando se efectúa el primer cambio sobre el recurso, se debe incrementar en 1 la versión, añadir la fecha de documentación y una breve documentación de los ajustes o nuevas implementaciones efectuadas.

- A medida que se ajuste el recurso, se deberá incrementar en 1 el número de la versión, deberá contener la breve documentación y la fecha de modificación, pero no se podrá borrar documentación existente o tomar versiones muy inferiores a la que antecede según el nuevo ajuste.

Pressman, R. S. afirma que todos los productos de trabajo se efectúan cuando el software se construye, cada uno debe analizarse, declararse e identificarse de manera única. Una vez logrado lo anterior, pueden establecerse diferentes mecanismos y métodos para control de versión y de cambio (PRESSMAN, 2002, p. 1). Esto con el fin de dar garantía que la calidad es constante mientras se efectúa mantenimiento en el software y se realice auditoria y reportes sobre los resultados obtenidos durante el proceso, esto para darlos a conocer e informar al respecto.

En la actualidad existen muchas herramientas que permiten tener y realizar un correcto control de versiones según la necesidad requerida, y esto es fundamental para la administración efectiva y concisa del cambio efectuado.

Según PRESSMAN, R. S. dice que el control de versión cuenta con la combinación e integración de procedimientos y herramientas para administrar diferentes versiones de objetos de desarrollo y configuración que se adelantan y finiquitan durante el proceso de software (PRESSMAN, 2002, p. 510).

Para finiquitar el contexto de control de versiones y según lo manifestado por PRESSMAN, R. S., un sistema de control de versión tiene directamente cuatro funcionalidades, las cuales se describen a continuación:

1) Una base de datos del software que almacena toda la información importante de la estructura de este.
2) Una capacidad de gestión de versión que aloja aquellos cambios efectuados en los recursos de desarrollo o lo que permite la construcción de cualquier versión usando diferencias de las versiones pasadas.
3) Un mecanismo ágil que permite la elaboración de la colección de todos los recursos de configuración, construcción e implementación de un correcto control de versiones específico

del software, conjuntamente, aquellos sistemas de control de cambio con periodicidad y de control de versión frecuentes.

4) Un identificador de errores que garantiza al equipo de trabajo o al equipo de desarrollo, registrar y efectuar seguimiento sobre todos aquellos estados relevantes de los conflictos asociados a los recursos de configuración.

GIT es una herramienta de código abierto robusto, el cual está en constante mantenimiento. Tal software fue desarrollado por Linus Torvalds, el ingeniero de software que en el 2005 creó el kernel del sistema operativo Linux.

Varios proyectos de software dependen de GIT para el control de versiones de sus recursos, incluyendo proyectos productivos y de código abierto. Por lo tanto, es prudente referenciar los siguientes autores para detallar un poco a cerca del concepto generalizado que se tiene del sistema de control de versiones GIT:

Según, Ram, K. 2013 & Blischak, J. D. 2016, mencionan que GIT es una herramienta avanzada sobre el control de versiones (estilo similar al "control de cambios" de Microsoft Word) distribuido.

Por su parte, FERNANDEZ dice que GIT es uno de los sistemas de control de versiones distribuidos que ha contado con un alto crecimiento de uso de este en los últimos años y es el utilizado y aplicado para gestionar el código del Kernel de Linux y otros importantes proyectos libres y de alta relevancia como X.org y Ruby on Rails (FERNANDEZ, 2008, p. 28).

Por otro lado, dice que, GIT permite efectuar un "rastreo" al progreso de un proyecto a lo largo del tiempo, dado que efectúa "capturas" del mismo a medida que se encuentra en evolución y los cambios se van aplicando según lo realizado (Astigarraga, Cruz-Alonzo, 2022, p. 1), esto hace que se pueda visualizar más fácilmente qué modificaciones se hicieron, quién y por qué se realizaron, suministrando buenamente aún más el trabajo.

Por último y no menos importante, BERUBE cuestiona al expresar que ¿Quién dice que el sistema de control de versiones GIT es sólo para proyectos de desarrollo software? (BERUBE, 2011, p.

57). Por el contrario, hace referencia a la protección de los datos con el “creativo” sistema de copias de seguridad GIT.

Además de lo anteriormente mencionado es indispensable saber que GIT se ha diseñado bajo algunas características que lo hacen ser la mejor alternativa al momento de emplear un control de versiones dentro de un proyecto. Dentro de las características se encuentran:

- El rendimiento: Los algoritmos implementados, el formato que se definió para los archivos del repositorio, la arquitectura distribuida, la ramificación, fusión, comprensión y, sobre todo, la herramienta que nos permite comparar versiones nos brinda múltiples ventajas en términos de rendimiento.
- La seguridad: La información de los archivos se encuentra protegida por un algoritmo que cuida todos los recursos (código) y el historial de versiones y cambios, frente a las modificaciones que se realizan de forma accidental o maliciosa, garantizando la conservación de la integridad.
- La flexibilidad: Se puede realizar diferentes proyectos de trabajo de desarrollo de fácil adaptación, realizando un nivel de seguimiento constante en su eficiencia.

Problema

Para llevar a cabo la creación de un software, se implementa el Systems Development Life Cycle (SDLC), y la última etapa del ciclo es la de mantenimiento, esta fase da por finalizado el proceso de desarrollo del software, pero habilita la creación de nuevos desarrollos o implementaciones que no estaban dentro del alcance inicial del proyecto, y estos son reportados por los usuarios funcionales del sistema de información. Asimismo, el desarrollo de software es un proceso continuo, por tanto, en la etapa de mantenimiento es crucial garantizar el correcto funcionamiento y la actualización constante del software, lo que implica la realización de cambios y actualizaciones en el código fuente existente. Sin embargo, este proceso puede ser complicado y propenso a errores si no se gestiona adecuadamente.

Una de las principales dificultades en la etapa de mantenimiento de software es el seguimiento y control de las versiones del código fuente. En muchos casos, el equipo de mantenimiento trabaja con múltiples versiones del código fuente, lo que puede generar confusión, redundancias y conflictos. Además, la falta de una herramienta adecuada de seguimiento y control de versión puede dificultar la colaboración y la comunicación entre los equipos de desarrollo y mantenimiento.

Por lo tanto, el problema abordado en este proyecto de investigación hace referencia a la carencia de una herramienta a la medida de seguimiento y control de versión de procesos de software adecuada para la etapa de mantenimiento, lo que puede generar errores, redundancias y conflictos en el proceso de mantenimiento de software. En consecuencia, es necesario diseñar y desarrollar un gestor de seguimiento y control de versión que facilite la gestión eficiente y efectiva de las versiones del código fuente en la etapa de mantenimiento, mejorando así la calidad y la eficiencia del proceso de mantenimiento de software.

Objetivo general

Desarrollar un sistema de información para la gestión de novedades y solicitudes reportadas en la etapa de mantenimiento del ciclo de vida del software.

Objetivos específicos

- Analizar los requerimientos y novedades de acuerdo con las necesidades del sistema de información.
- Aplicar los principios de la ingeniería de software para el modelamiento de datos y la estructuración lógica del sistema como fundamento principal en el desarrollo de la aplicación.
- Establecer pruebas unitarias de la persistencia y de la lógica implementada en el software.
- Verificar que las tecnologías y herramientas utilizadas garanticen una buena experiencia de usuario.

Método

A continuación, se relaciona la metodología utilizada para la elaboración de este proyecto de investigación, teniendo en cuenta los diferentes métodos y técnicas propias de cada una de las etapas que intervinieron en el estudio, incluyendo los procedimientos, recolección, procesamiento y análisis de la información.

Tipo de investigación

Tomando como base el problema planteado a lo largo del desarrollo del proyecto referido Gestor de seguimiento y control de versión de procesos de software en la etapa de mantenimiento, y en función de la formulación de los objetivos generales y específicos que la sustenta, el presente proyecto está enmarcado dentro del tipo de investigación aplicada, la cual consiste en "… la invención de artefactos para solucionar problemáticas y obtener beneficios económicos. Es de tipo experimental pues está integrada por un conjunto de actividades metódicas y técnicas que se realizan para recabar la información y datos necesarios sobre el tema a investigar y el problema a resolver" (García-Córdoba, 2005), en función de esta definición y enmarcado en el proyecto planteado bajo esta modalidad, en la cual se debe cumplir con el protocolo descrito por este tipo de investigación, se desarrollará el diagnóstico de la situación existente a fin de mejorar la atención, la toma de decisiones y el desarrollo de los beneficiarios de la corporación bajo un análisis de diagnóstico. Seguidamente, y atendiendo a los resultados del diagnóstico, se formulará el modelo operativo propuesto, donde se intenta dar respuestas al problema planteado.

Las investigaciones realizadas sobre diferentes modelos de gestión de procesos se encuentran abarcados en el campo y por experiencia laboral, dado que en el proyecto de software vinculado usan las herramientas de correo electrónico y Excel para llevar un control y seguimiento de solicitudes con respecto a las acciones de mejora o nuevos requerimientos que se deben implementar en el sistema de información.

Lo anterior genera falencias, las cuales se comunican a continuación:

- Falta de conocimiento por parte del usuario del estado real de las solicitudes.
- Inconsistencias en trazabilidad de correos.
- Desconocimiento por desarrolladores de asignaciones de solicitudes.
- Falencias en versiones de fuentes y/o en merge ocasionados.
- Documentación; formatos de control de cambios incompletos.

La intención de un sistema de información que genere automatización de seguimiento y control de versiones de procesos de software permite agilizar técnicas, optimizar tiempos y mejorar métodos de trabajo, generando una mejor interacción entre usuarios funcionales, clientes, sistema y equipo técnico.

Para llevar a cabo lo anterior, se desarrollarán y se implementarán las siguientes fases de la investigación en el actual estudio:

- **Análisis:**

  Se efectuaron estudios a los diferentes modelos de procesos que conlleva el soporte de un aplicativo o software, se identificaron falencias en ellos debido a su estructura organizacional, su trazabilidad y seguimiento de estos, ocasionando así generar un planteamiento para optimizar de manera sistémica dichos procesos.

- **Documentación:**

  Una vez identificadas las falencias y novedades con las que cuentan dichos modelos de procesos, se planteó documentar la estructura que tenía la gestión y modelo de estos procesos, con el fin de tener clara la secuencia y así plantear la posible versatilidad que podría contener la sistematización de un proyecto orientado a ello.

- **Hipótesis:**

  Al tener claro y estructurar la posible gestión de procesos, se elabora la hipótesis, planteando principalmente el desarrollo de un sistema de información que permita efectuar un manejo, control y seguimiento del modelo de procesos implementados en un software al cual se le está

dando soporte, es decir, el cual se encuentra en la fase de mantenimiento del ciclo de vida del software.

- **Diseño:**

  Se procede a efectuar el modelamiento y estructuración del marco de investigación del proyecto a elaborar, con el fin de definir aquellas todas variables que puedan interferir en el desarrollo de este, y adicionando, cada de uno de los ítems de investigación que se llevan a cabo para robustecer un proyecto.

- **Selección de la muestra:**

  Se realiza planteamiento del problema a líderes de proyectos de software, con el fin de corroborar si la necesidad identificada coincide con la que se encuentran inmersos los usuarios en el día a día sobre los procesos y validaciones efectuados.

- **Recolección de los datos:**

  Se realiza toma de datos a los usuarios funcionales sobre los procesos efectuados y las falencias que presentan, con el fin de ampliar la visión acerca de la problemática y así profundizar acerca de ella en la investigación a elaborar.

- **Análisis de los datos:**

  Se procede a efectuar revisión e interpretación de las indagaciones realizadas a los usuarios, en donde se corrobora que la necesidad de ellos puede quedar solventada de acuerdo con la hipótesis planteada, una vez esta sea elaborada.

- **Presentación de resultados:**

  La generación y exposición de los resultados obtenidos se podrán verificar una vez el desarrollo del software ya se encuentre consumado/finalizado.

Toda la información recopilada, tanto la cuantitativa como la cualitativa, harán parte esencial de la investigación y el resultado será clave para posteriormente utilizarlo en el desarrollo de los objetivos específicos.

Hipótesis

Existen modelos de gestión de procesos de sistemas de información que se encuentran implementados en diferentes compañías. Sin embargo, muchos de ellos tienen como base algunas herramientas que hacen que todo sea muy manual y ocasionando reprocesos en la atención, apropiación y secuencia de dichos procesos. Adicional a lo anterior, se ha evidenciado otra gran problemática que es en el control de versiones de los objetos del sistema y de la base de datos (componentes del software).

Propuesta

Se diseñará una solución de software, que resuelva las necesidades planteadas en esta problemática, donde se plantea la realización de un sistema de información, que automatice y cree una estructura de trabajo óptima para así agilizar procesos. El 100% del software se abarcará en 2 fases, es decir, el proyecto contará con una restricción hasta el 80% en la primera versión, el 20% faltante se trabajará en la segunda versión del software.

Los objetivos específicos del proyecto incluyen entregar la funcionalidad al 80% en un plazo de 3 meses, dando prioridad a toda la gestión de procesos y dejando como requerimientos futuros lo siguiente:

- Creación de módulo que efectuará Merge de diferentes tipos de fuentes (recursos), dado que en la primera versión se encontrará la generación automática del formato de control de cambios, más no se aplicará Merge para archivos con diferentes extensiones, ejemplo; .JAVA y .SQL.
- Creación de aplicación móvil para que los usuarios vinculados en el sistema de información puedan reportar las novedades y hacer un seguimiento de una manera más cómoda desde un smartphone.

Se realizó un análisis de los procesos por los que mínimamente debe pasar una historia de usuario (solicitud). Dichos procesos se exponen a continuación:

- Reporte de solicitud: El usuario informa su necesidad. Se hace un levantamiento de información.
- Catalogación de solicitud: Identificación de prioridad y si es algo que esté deteniendo el proceso o es un desarrollo nuevo, también se realiza la estimación.
- Asignación a equipo de trabajo: Se realiza la asignación de la solicitud a integrante del equipo, a partir de los conocimientos de este y la necesidad reportada.
- Desarrollo: La persona a la que se le asignó la solicitud da la solución a la misma, a partir de actualizaciones en los objetos del sistema o de la base de datos.
- Documentación y control de versiones: Se realiza un formato de control de cambios de software en el cual se exponga la nueva versión y los cambios realizados.
- Fase de pruebas unitarias y pruebas de certificación por analista de calidad: En esta fase se garantiza la calidad del desarrollo hecho. Se hacen pruebas unitarias y pruebas de QA.
- Instalación en ambientes correspondientes: Se envía a instalar en el ambiente de producción, después de tener la certificación de calidad.
- Cierre de la solicitud: Se da el cierre a la solicitud.

Teniendo en cuenta la secuencia anterior, se espera realizar un software que sea capaz de administrar cada uno de los ítems anteriores y también de combatir las principales y más graves problemáticas que se evidencian en los modelos actuales que son:

- Control de versiones de objetos: Se llevará control de cada objeto de la base de datos y externo a ella. Garantizando así que no haya desarrollos perdidos o que haya complique al momento de realizar pasos al ambiente de producción. Cabe resaltar que en la primera versión se realizará el formato de control de cambios más no funcionalidad de merge o control de versiones de archivos con diferentes extensiones.

- Documentación manual: A partir de algunas entradas y condicionales, el sistema podrá generar un formato de control de cambios digital. También, se podrá generar el certificado de calidad de esta misma manera. Por lo que la documentación ya será automatizada.

Sin embargo, dado al tiempo que conlleva todo el ciclo de vida del software sobre el proyecto actual, se plantea desde ya una segunda versión sobre el actual proyecto, en donde se implementarán los siguientes requerimientos:

- Control de versiones de objetos: Se implementará módulo para efectuar un control de cada objeto de base de datos y externo a ella, para garantizar un correcto merge para todos los tipos de fuentes que pueda contener el aplicativo, el cual será complemento e insumo para integrar y consolidar el formato de control de cambios.
- Desarrollo APP: Creación de aplicación móvil Android, que permitirá reportar los respectivos requerimientos a tratar para la corrección de estos. Adicionalmente, permitirá conocer la trazabilidad y estado en el cual se encuentra la solicitud reportada.

Técnicas e instrumentos de recolección de información

De acuerdo con la naturaleza del estudio, cada parte de este se analizará en pro de los datos necesarios para su desarrollo, tanto del punto de vista teórico, como metodológico y de resultados, por lo que orden se usarán:

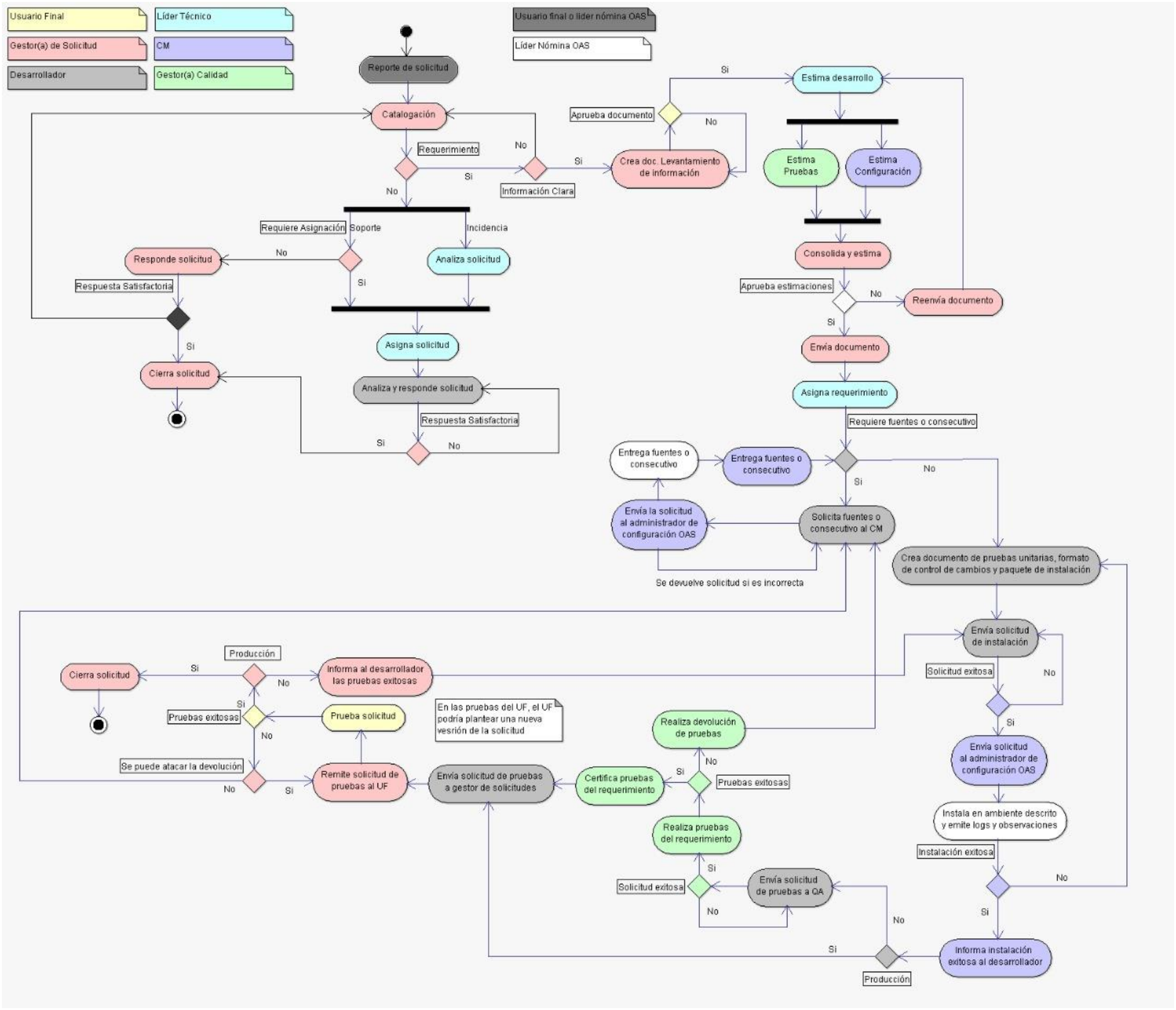
- Técnicas de Investigación Documental:

  Se procedió a efectuar el seguimiento correspondiente a la operación de procesos que realizaba el equipo técnico de la compañía Inetum Colombia SAS para brindar soporte a la herramienta Páoyer del Ministerio de Defensa (aplicativo de nómina que se encontraba en fase de mantenimiento, según el ciclo de vida del software), con la finalidad de identificar cada una de sus etapas, la gestión que se efectuaba, el manejo correspondiente a una solicitud y la trazabilidad que esta conllevaba.Por ende, a partir del análisis e interpretación realizado

sobre la operación de procesos de dicha entidad, se creó gráfico como objeto de estudio para proceder con la documentación correspondiente, plantear problema y avanzar con la respectiva gestión del actual proyecto de investigación.

Figura 1

Mapeo y gráfico de trazabilidad de procesos.

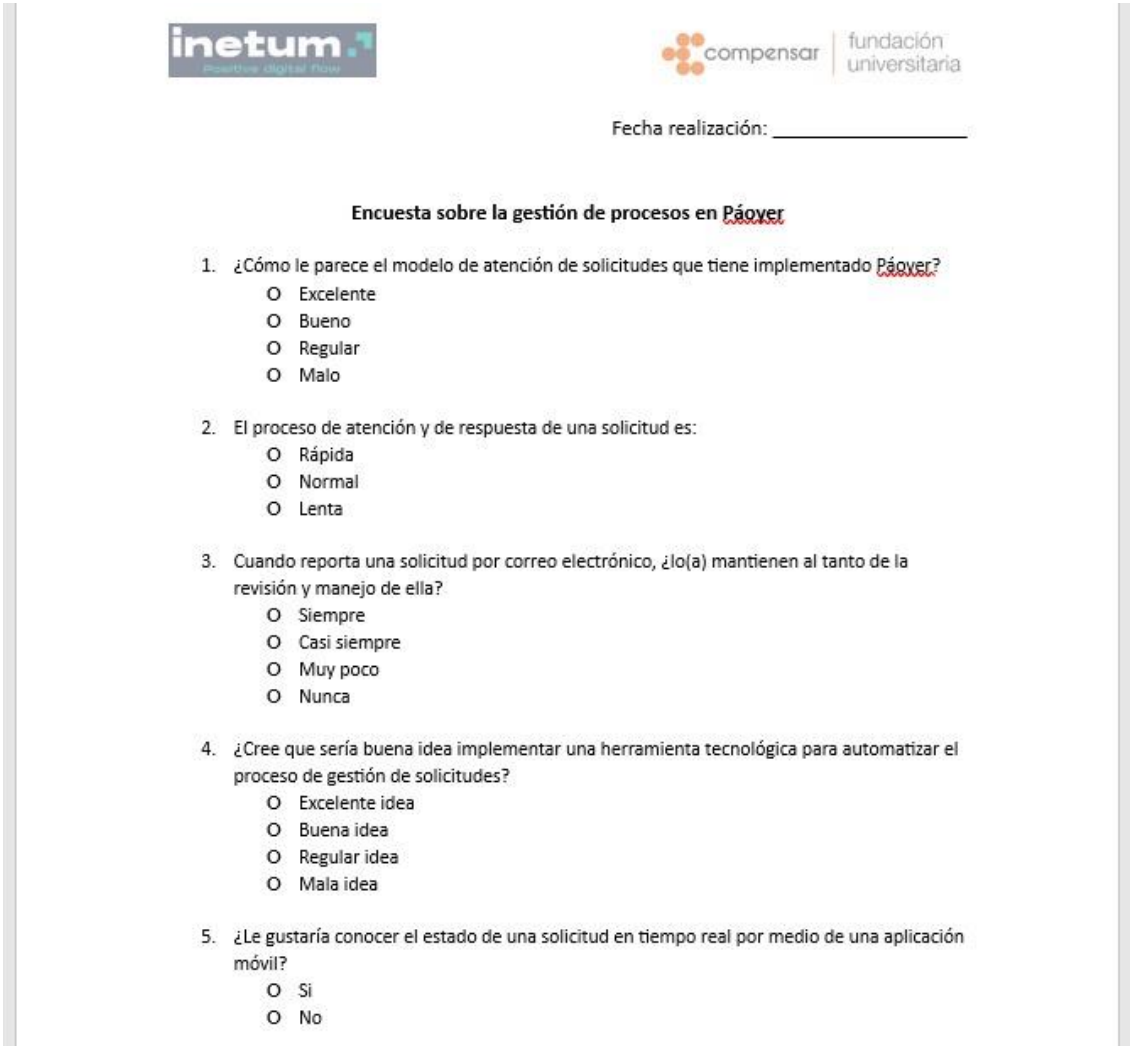

Usuario Final
Líder Técnico
Gestor(a) de Solicitud
CM
Desarrollador
Gestor(a) Calidad
Usuario final o líder nómina OAS
Líder Nómina OAS
Reporte de solicitud
Catalogación
Requerimiento
No
Si
Información Clara
Si
Crea doc. Levantamiento de información
Aprueba documento
No
Si
Estima desarrollo
Estima Pruebas
Estima Configuración
Consolida y estima
Aprueba estimaciones
No
Reenvía documento
Si
Envía documento
Asigna requerimiento
Requiere fuentes o consecutivo
Entrega fuentes o consecutivo
Entrega fuentes o consecutivo
Si
No
Solicita fuentes o consecutivo al CM
Envía la solicitud al administrador de configuración OAS
Se devuelve solicitud si es incorrecta
Crea documento de pruebas unitarias, formato de control de cambios y paquete de instalación
Envía solicitud de instalación
Solicitud exitosa
No
Si
Envía solicitud al administrador de configuración OAS
Instala en ambiente descrito y emite logs y observaciones
Instalación exitosa
No
Si
Informa instalación exitosa al desarrollador
Producción
No
Si
Envía solicitud de pruebas a QA
Solicitud exitosa
No
Si
Realiza pruebas del requerimiento
Pruebas exitosas
No
Si
Certifica pruebas del requerimiento
Envía solicitud de pruebas a gestor de solicitudes
Realiza devolución de pruebas
Remite solicitud de pruebas al UF
Se puede atacar la devolución
No
Si
Prueba solicitud
Pruebas exitosas
No
Si
En las pruebas del UF, el UF podría plantear una nueva vesrión de la solicitud
Informa al desarrollador las pruebas exitosas
Producción
No
Si
Cierra solicitud
Requiere Asignación
Soporte
Incidencia
No
Si
Responde solicitud
Respuesta Satisfactoria
Si
Cierra solicitud
Analiza solicitud
Asigna solicitud
Analiza y responde solicitud
Respuesta Satisfactoria
Si
No

- Técnica de Encuesta:

  Con la finalidad de poder recolectar información acerca de la necesidad y problema identificado a optimizar, se plantea realizar encuesta de tipo cuantitativa a un grupo de diez (10) usuarios funcionales acerca de la gestión de procesos, con la finalidad de abarcar a mayor profundidad sobre la necesidad y efectuar una posterior interpretación y toma de decisión con respecto a la ejecución e inicio del proyecto, siempre partiendo de los datos y resultados obtenidos.

  Figura 2

  Encuesta aplicada a usuarios funcionales de la herramienta analizada.



inetum. Positive digital flow

compensar | fundación universitaria

Fecha realización: _____

Encuesta sobre la gestión de procesos en Páover

1. ¿Cómo le parece el modelo de atención de solicitudes que tiene implementado Páover?
   - O Excelente
   - O Bueno
   - O Regular
   - O Malo

2. El proceso de atención y de respuesta de una solicitud es:
   - O Rápida
   - O Normal
   - O Lenta

3. Cuando reporta una solicitud por correo electrónico, ¿lo(a) mantienen al tanto de la revisión y manejo de ella?
   - O Siempre
   - O Casi siempre
   - O Muy poco
   - O Nunca

4. ¿Cree que sería buena idea implementar una herramienta tecnológica para automatizar el proceso de gestión de solicitudes?
   - O Excelente idea
   - O Buena idea
   - O Regular idea
   - O Mala idea

5. ¿Le gustaría conocer el estado de una solicitud en tiempo real por medio de una aplicación móvil?
   - O Si
   - O No

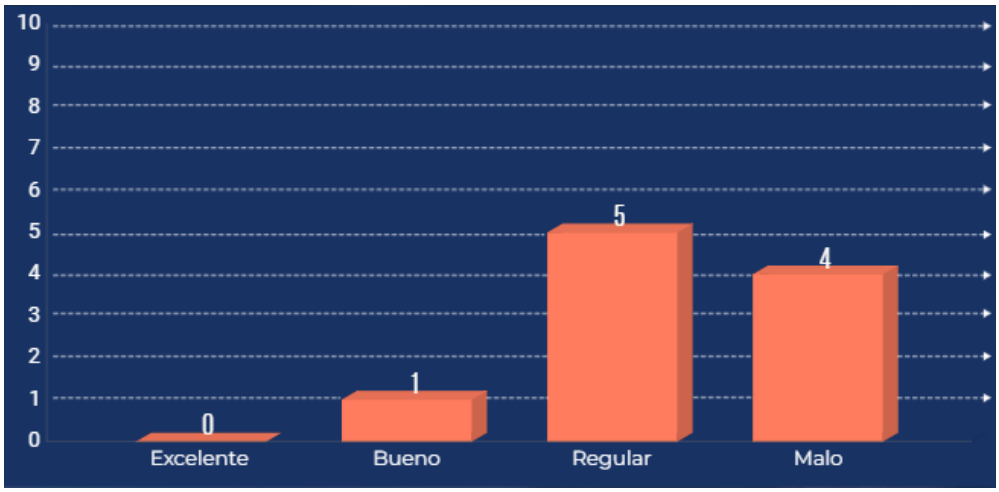
Resultados de la encuesta:

1. ¿Cómo le parece el modelo de atención de solicitudes que tiene implementado Páoyer?

   Para el 90% de los encuestados, creen que el modelo de gestión de solicitudes implementado no es bueno:

   Figura 3

   Análisis de resultados sobre la primera pregunta de la encuesta.

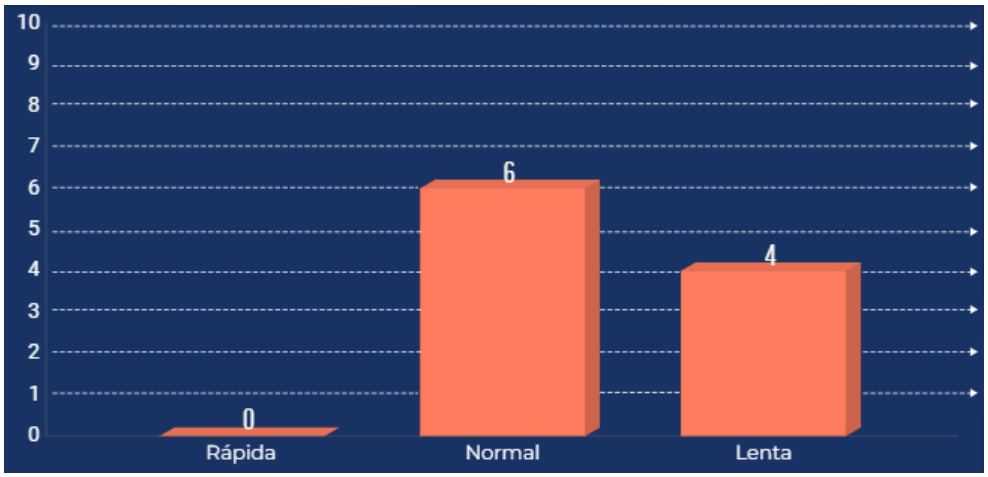


2. El proceso de atención y de respuesta de una solicitud es.

   El 100% de los encuestados hacen referencia a que la atención y respuesta no es rápida, solamente al 60% les parece normal y el 40% restante es lenta:

   Figura 4

   Análisis de resultados sobre la segunda pregunta de la encuesta.

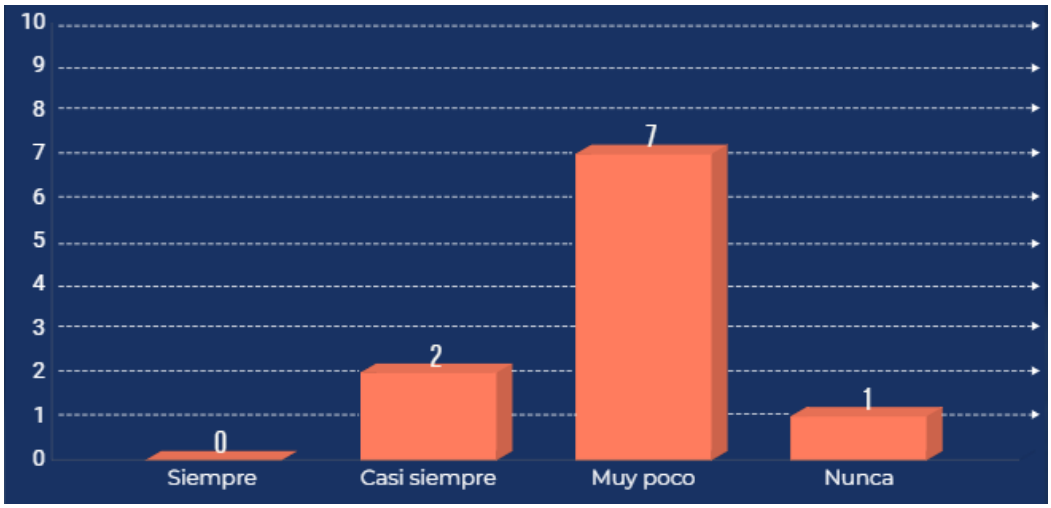


3. Cuando reporta una solicitud por correo electrónico, ¿lo(a) mantienen al tanto de la revisión y manejo de ella?

   El 80% de los encuestados sienten que son poco retroalimentados por parte del área encargada cuando reportan una novedad:

   Figura 5

   Análisis de resultados sobre la tercera pregunta de la encuesta.

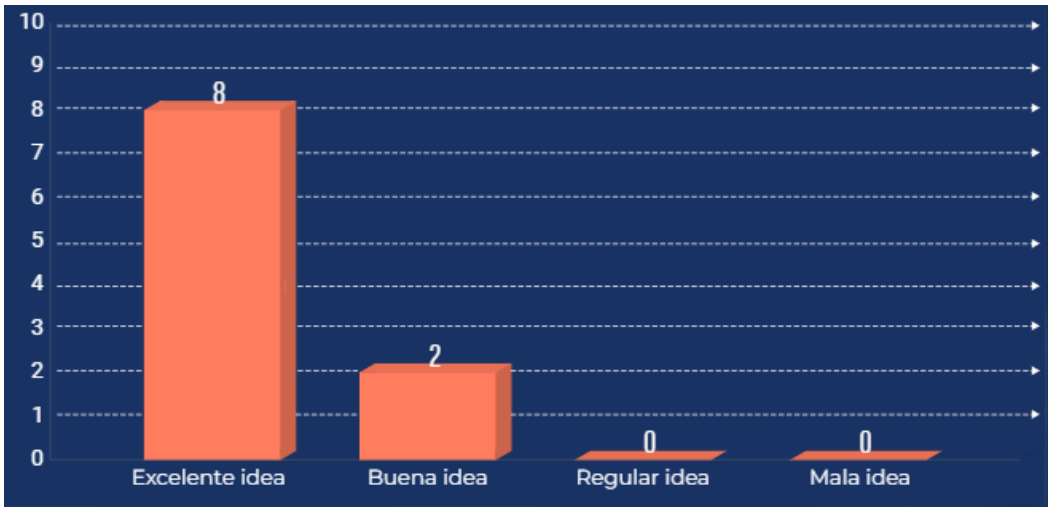


4. ¿Cree que sería buena idea implementar una herramienta tecnológica para automatizar el proceso de gestión de solicitudes?

   Para el 0% de los encuestados, creen que es una mala o regular idea implementar una herramienta tecnológica para optimizar los procesos, al contrario, el 80% les parece una excelente idea y tan solo el 20% creen que es buena idea.

   Figura 6

   Análisis de resultados sobre la cuarta pregunta de la encuesta.

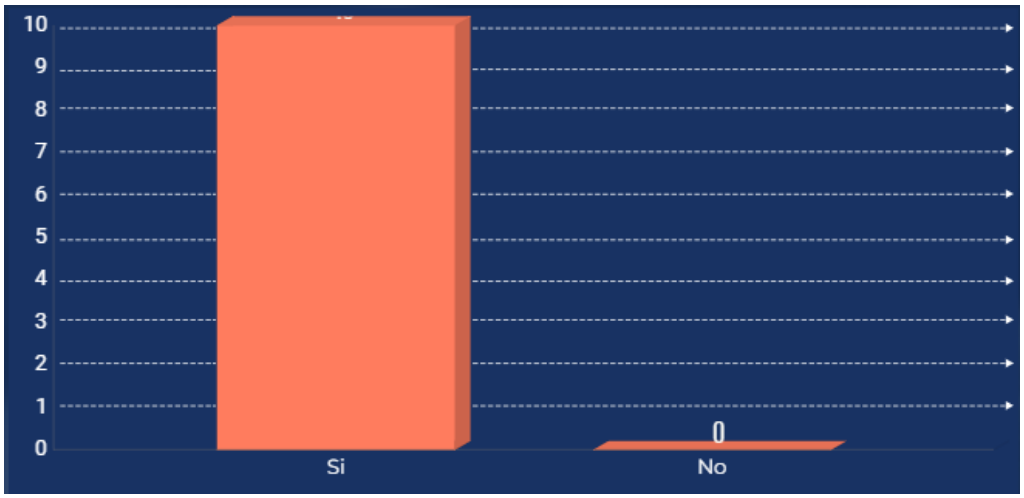


5. ¿Le gustaría conocer el estado de una solicitud en tiempo real por medio de una aplicación móvil?

   Se obtiene un 100% de aceptación con respecto a la creación de una aplicación móvil.

Figura 7

Análisis de resultados sobre la quinta pregunta de la encuesta.



Metodología de desarrollo

La metodología de desarrollo de software seleccionada para proceder con la elaboración del proyecto fue Extreme Programming o XP Programming, dado que esta metodología cuenta con la facilidad y permite construir el software final de manera progresiva bajo un marco de desarrollo ágil,

admitiendo así una entrega de una nueva funcionalidad según cada etapa de la metodología, y adicionalmente, permite usar e interactuar con el software paulatinamente antes de que se culmine el desarrollo en su totalidad, logrando obtener así un resultado de mayor calidad para optimizar la eficiencia de entrega y avance por parte del equipo de desarrollo. Adicionalmente, de manera paralela se utilizará el método ágil Kanban, debido a que permite optimizar el flujo y secuencia de trabajo y en alta calidad, concediendo evitar la sobrecarga de compromisos y actividades, y lo que garantiza mejorar la entrega de trabajo y de acciones planteadas en el backlog del proyecto, como fue descrito por el pionero David J. Anderson, el método proporciona una orientación de cambio, incremental y progresivo, de procesos y sistemas, para las diferentes organizaciones de trabajo de conocimiento que se puedan llevar a cabo.

La Tabla 1 presenta el listado de actividades en el orden correspondiente a trabajar, el cual formó un total de 21 tareas.

Tabla 1

Listado de actividades (cronograma).

Actividad	**Fecha inicio**	**Fecha fin**
Análisis y planteamiento del problema	Febrero 05 de 2023	Febrero 16 de 2023
Construcción ficha Anteproyecto	Febrero 17 de 2023	Marzo 02 de 2023
Elaboración MER	Marzo 03 de 2023	Marzo 09 de 2023
Construcción base de datos	Marzo 10 de 2023	Marzo 14 de 2023
Diseño y elaboración Front HomePage	Marzo 15 de 2023	Marzo 16 de 2023
Elaboración Front del Login	Marzo 17 de 2023	Marzo 30 de 2023
Análisis e implementación lógica para roles y permisos de usuarios	Marzo 31 de 2023	Abril 02 de 2023

Creación Front de HomePage usuario logueado	Abril 03 de 2023	Abril 04 de 2023
Maquetación Front reporte de solicitud	Abril 05 de 2023	Abril 10 de 2023
Construcción Front catalogación de solicitud	Abril 11 de 2023	Abril 13 de 2023
Elaboración Front asignación de solicitudes	Abril 14 de 2023	Abril 16 de 2023
Creación Front módulo seguimiento de solicitudes	Abril 17 de 2023	Abril 18 de 2023
Maquetación Front de diligenciamiento de tiempos de solicitudes	Abril 19 de 2023	Abril 20 de 2023
Construcción Front de control de versiones de recursos	Abril 21 de 2023	Abril 23 de 2023
Elaboración Front módulo de documentación (FCCS)	Abril 24 de 2023	Abril 25 de 2023
Creación Front módulo área QA	Abril 26 de 2023	Abril 27 de 2023
Maquetación Front sección de trazabilidad de documentación	Abril 28 de 2023	Mayo 09 de 2023
Desarrollo Backend del sistema de información	Mayo 10 de 2023	Mayo 14 de 2023
Pruebas unitarias del software	Mayo 15 de 2023	Mayo 18 de 2023
Documentación técnica	Mayo 19 de 2023	Mayo 21 de 2023
Documentación funcional	Mayo 22 de 2023	Mayo 24 de 2023

Nota: La tabla 1 se elaboró según lo estructurado en el diagrama de Gantt, el cual permite determinar los tiempos que se llevarán a cabo por actividades o tareas planteadas para dar gestión y solución al proyecto, la cual termina siendo una herramienta que garantiza efectuar una planificación y trazabilidad sobre dichas actividades.

Lo anterior, permite que las actividades a desarrollar se complementen con la metodología XP a raíz de que está diseñada para ofrecer un resultado de software óptimo, y brindar manejo a las necesidades que los usuarios requieren en el momento adecuado. Dado que, ayuda al equipo técnico a ajustarse a los requerimientos cambiantes y a la variación de actividades, y al ser un software y/o proyecto de investigación pequeño, está en constante cambio debido al planteamiento y estructuración con el cual se diseñó.

Para cumplir a cabalidad con la metodología de desarrollo, se efectuó seguimiento sobre ella y cada una de sus fases y variables, esto, para planificar, constituir y progresar con el desarrollo de software siguiendo la respectiva parametrización de la metodología.

Dicha metodología cuenta con las siguientes 5 reglas principales, que se gestionaron para la elaboración del proyecto:

- Planificación: Se dedujo la necesidad que presentaban los usuarios funcionales con respecto al reporte de novedades al usar un software que se encontraba en etapa de mantenimiento del ciclo de vida del software, y posteriormente se planeó atacar dicha necesidad para brindar acción de mejora sobre los diferentes procesos efectuados.

  Para efectuar lo anterior, se planteó creación de trazabilidad de solicitudes para obtener mayor comprensión y así abarcar la necesidad identificada. La imagen de ello se puede encontrar en el ítem *Técnicas de Investigación Documental* del agregado *Técnicas e instrumentos de recolección de información*.
- Gestión: Se identificó y se describió el planteamiento de problema con respecto a la necesidad deducida, para proceder con la trazabilidad de este, avanzar con el procedimiento general del ciclo de vida del software e indagar y plantear el estado del arte al respecto.

Para continuar con ello, se realizó encuesta a los usuarios funcionales con la finalidad de ampliar y comprender la necesidad del cliente. Las evidencias se pueden contemplan en el ítem *Técnica de Encuesta* del agregado *Técnicas e instrumentos de recolección de información*.

- Diseño: Se plantean objetivos, historias de usuario, maquetación de base de datos y mockups de la aplicación, con el fin de avanzar en la gestión correspondiente de esta y así cumplir a cabalidad con la solución del problema propuesto.

  La evidencia de la estructuración y maquetación de la base de datos se podrá encontrar en el agregado *Resultados*.

- Codificación: Se avanza en la construcción e implementación del desarrollo del software de acuerdo con el diseño propuesto, con la finalidad de solventar lo allí planteado y dar cumplimiento a cada uno de los objetivos y las historias de usuario propuestas.

  A continuación, se asocian algunas imágenes de la codificación elaborada en el proyecto:

Figura 8

Entidad solicitud.

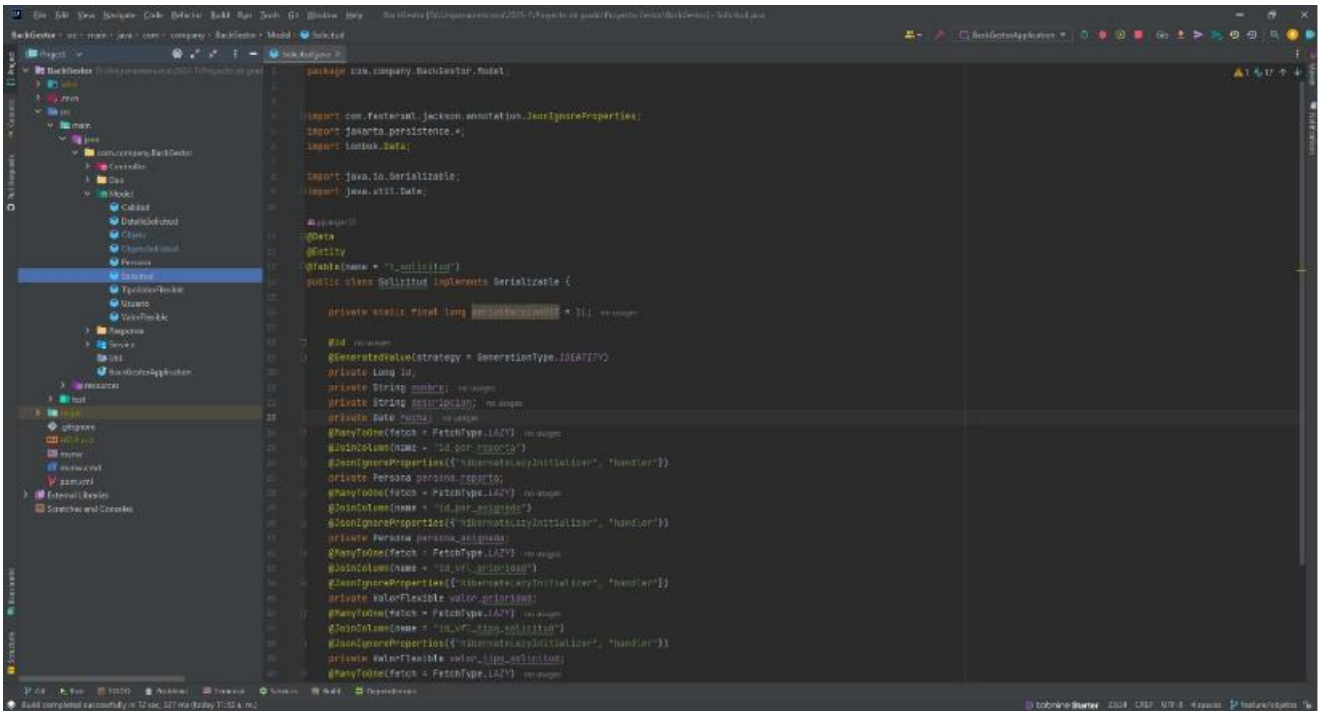


Figura 9

Interfaz de servicios de solicitud.

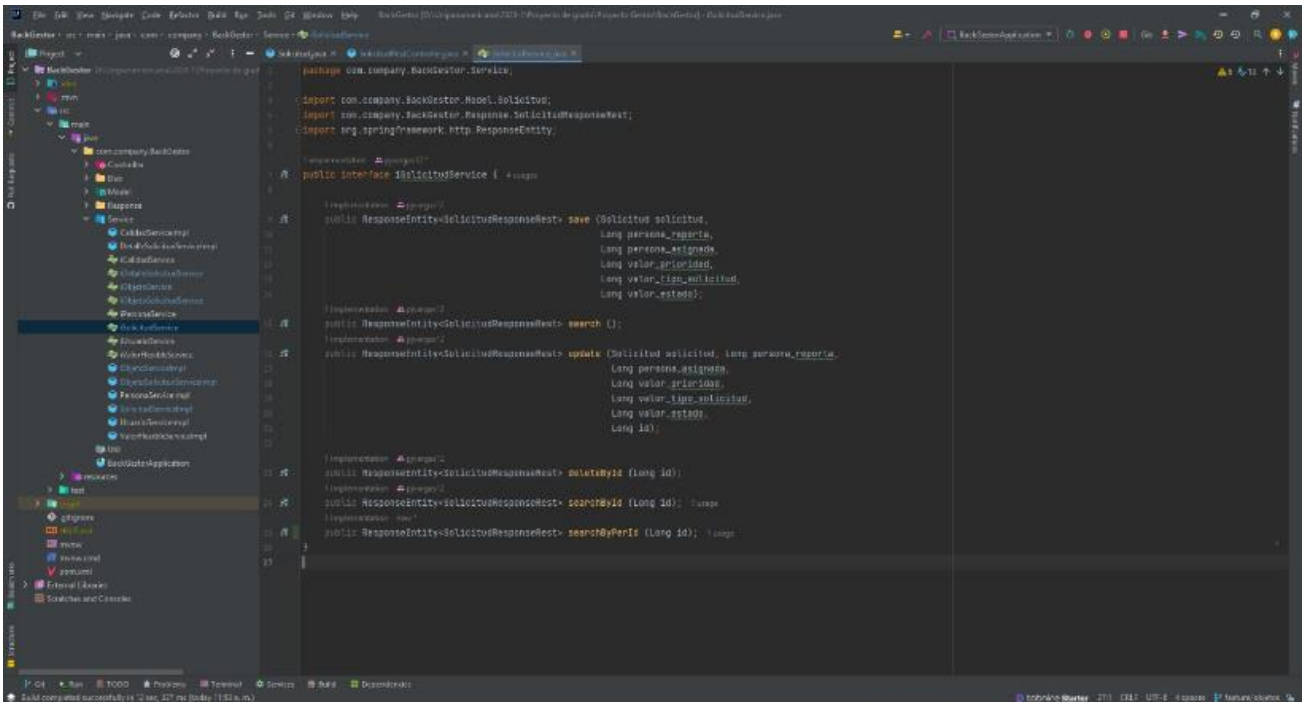


Figura 10

Implementación del servicio.

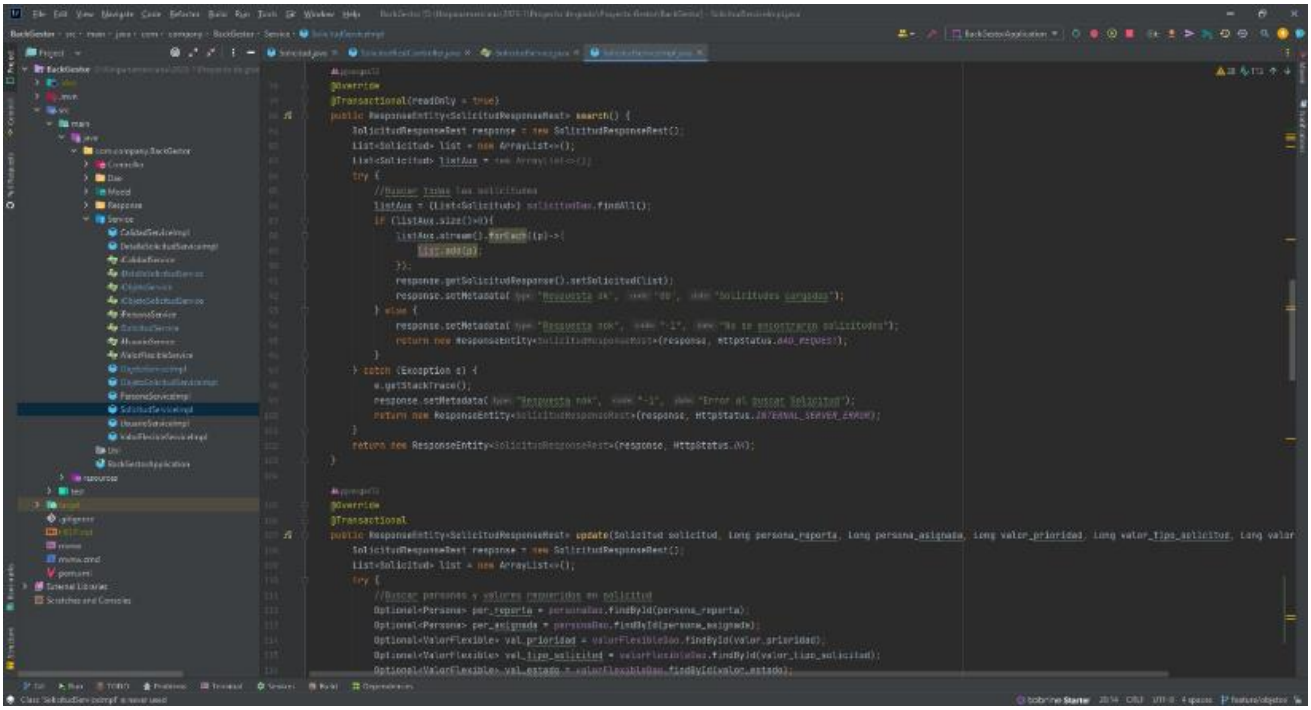
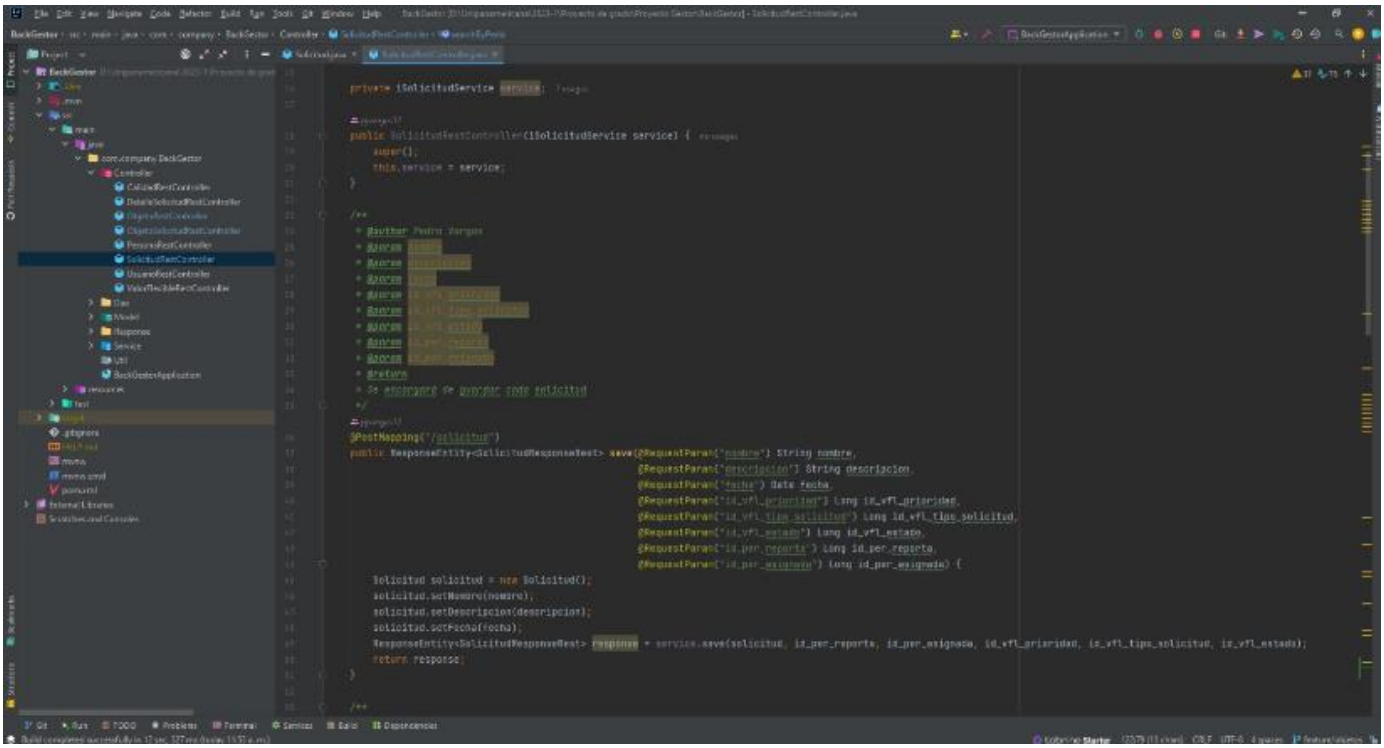


Figura 11

Controlador de solicitud.



- Pruebas: Se efectúan escenarios de calidad sobre el desarrollo del software elaborado e implementado, para certificar y cerciorarse del software creado, y así corroborar y realizar los requisitos y requerimientos tanto funcionales como no funcionales diseñados para dar solución a la necesidad propuesta, y así brindar un producto eficaz y óptimo según los diferentes planteamientos ofrecidos.

  Para mayor comprensión de los escenarios de pruebas, se anexa el documento *PU_Gestor de seguimiento y control de versión de procesos.pdf*.

Resultados

En el desarrollo del proyecto se lograron obtener diferentes tipos de resultados, tantos técnicos, lógicos como visuales. A continuación, se informará acerca de los resultados técnicos y visuales, dado que los lógicos se pueden contemplar en el ítem *Codificación* del agregado *Metodología de desarrollo*.

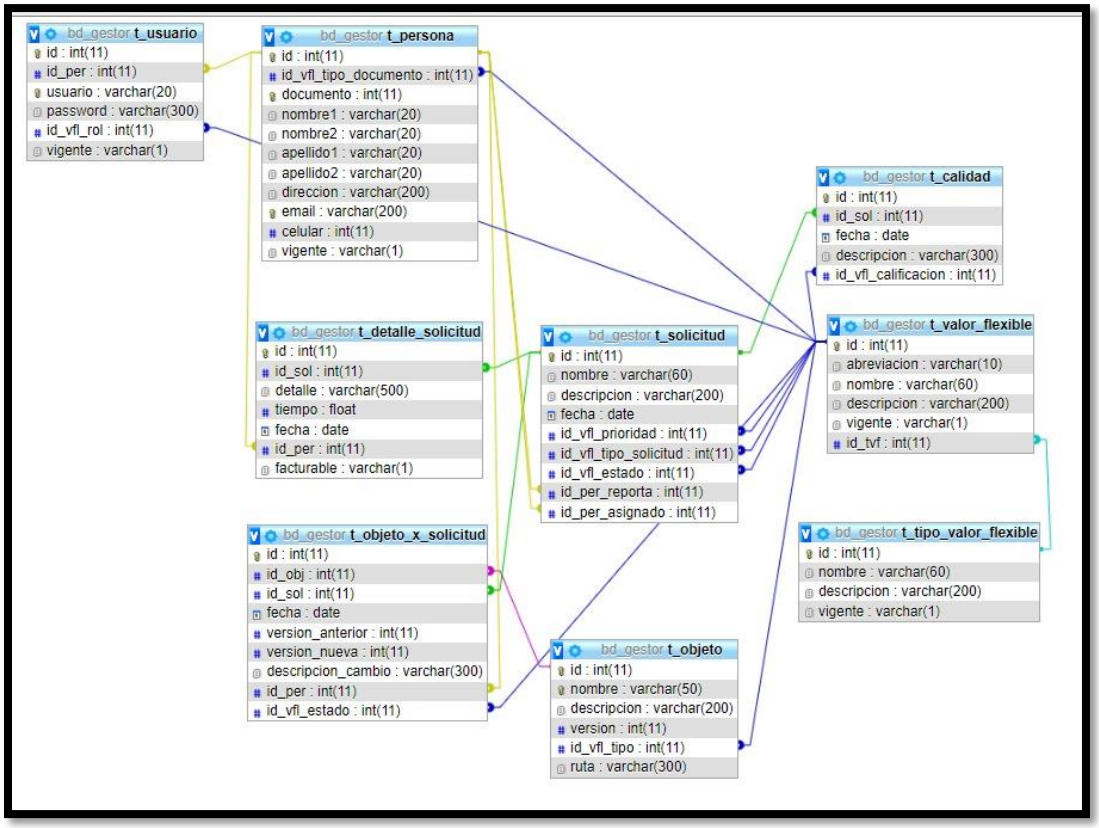
- Resultados técnicos:

  Se creó la base de datos BD_GESTOR en el lenguaje MySql, con el fin de poderla enlazar con el backend por medio de la herramienta SpringBoot, esto, para hacer los llamados correspondientes (consultas/querys), las respectivas agregaciones (insert/nuevos registros), modificaciones (updates/cambios) o suprimir información según sea el caso (delete), lo anterior, para tener el control correspondiente de toda la información que se procesará por medio del aplicativo creado.

  La base de datos cuenta con 8 tablas, por estándar de desarrollo cada tabla inicia nombrada ***T_*** esto, para tener una correcta estructura y manejo de ello, adicionalmente, cada tabla cuenta con una llave primaria que permite enlazar y relacionar las tablas de una manera eficaz para el procesamiento de la información:

  Figura 12

  Diseño y estructura de base de datos.



- Resultados visuales:

  Como efecto a la elaboración del backend, se crearon las respectivas interfaces del sistema de información en el framework Angular enlazado con el lenguaje TypeScript, este último, contiene la lógica correspondiente del frontend. A continuación, se contextualizará acerca de algunas interfaces con las que cuenta el software:

Figura 13

Home page del software.



Figura 14

Inicio de sesión.

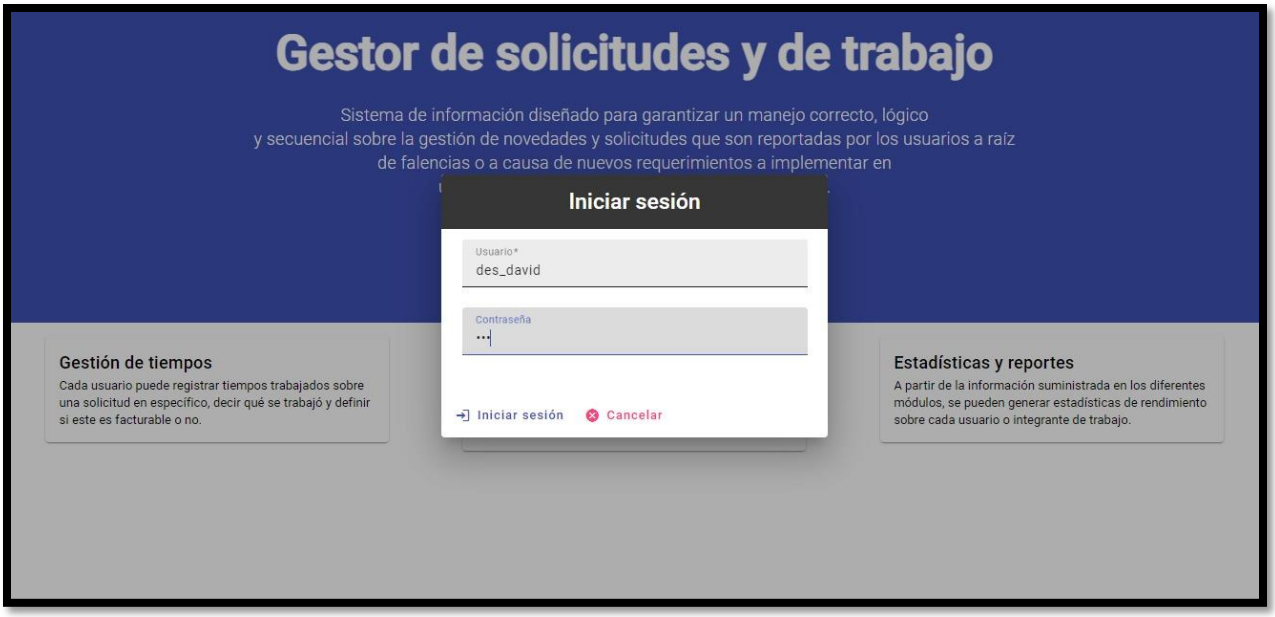


Figura 15

Página principal del desarrollador.

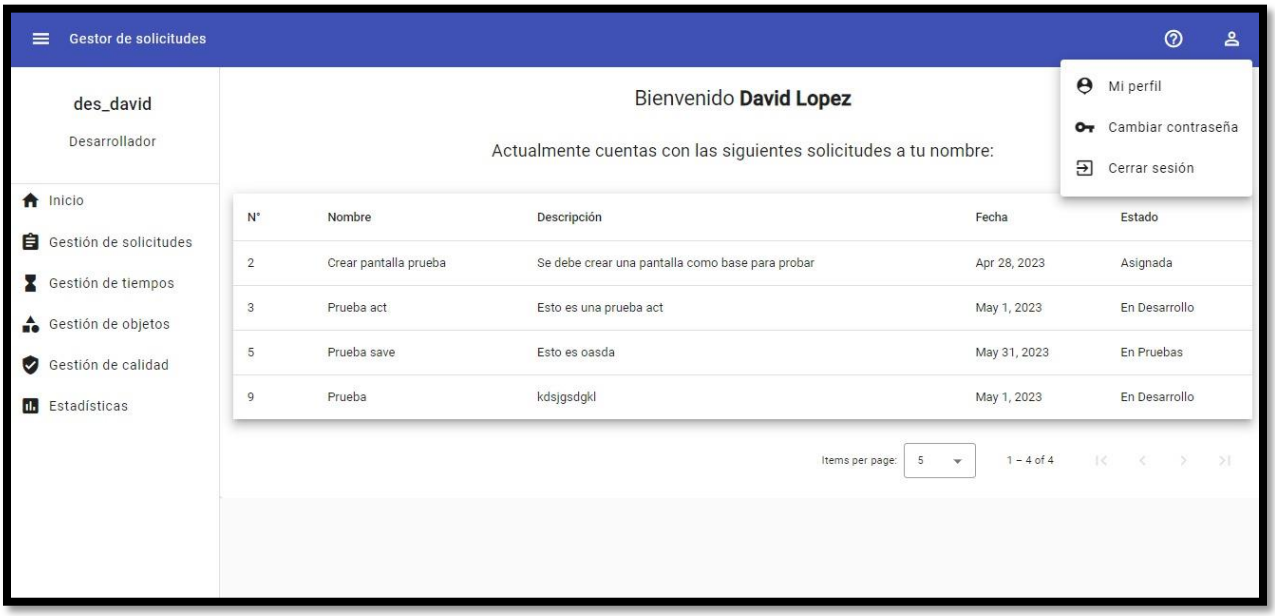


Figura 16

Listado de solicitudes.

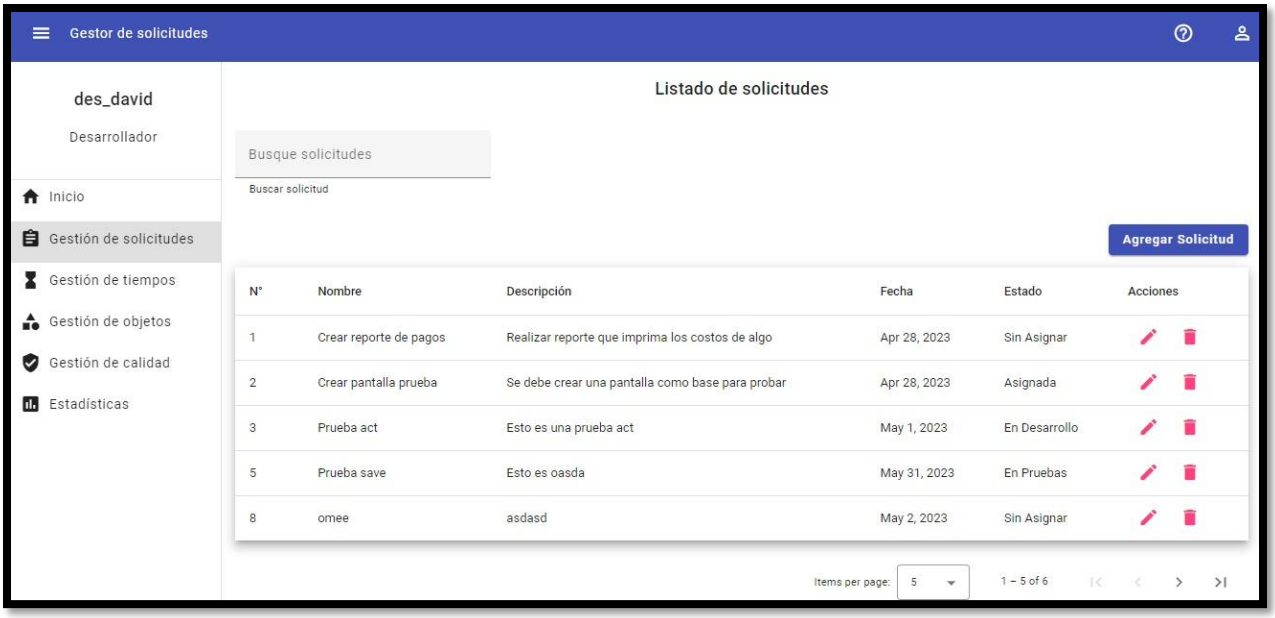


Figura 17

Añadir nuevas solicitudes.

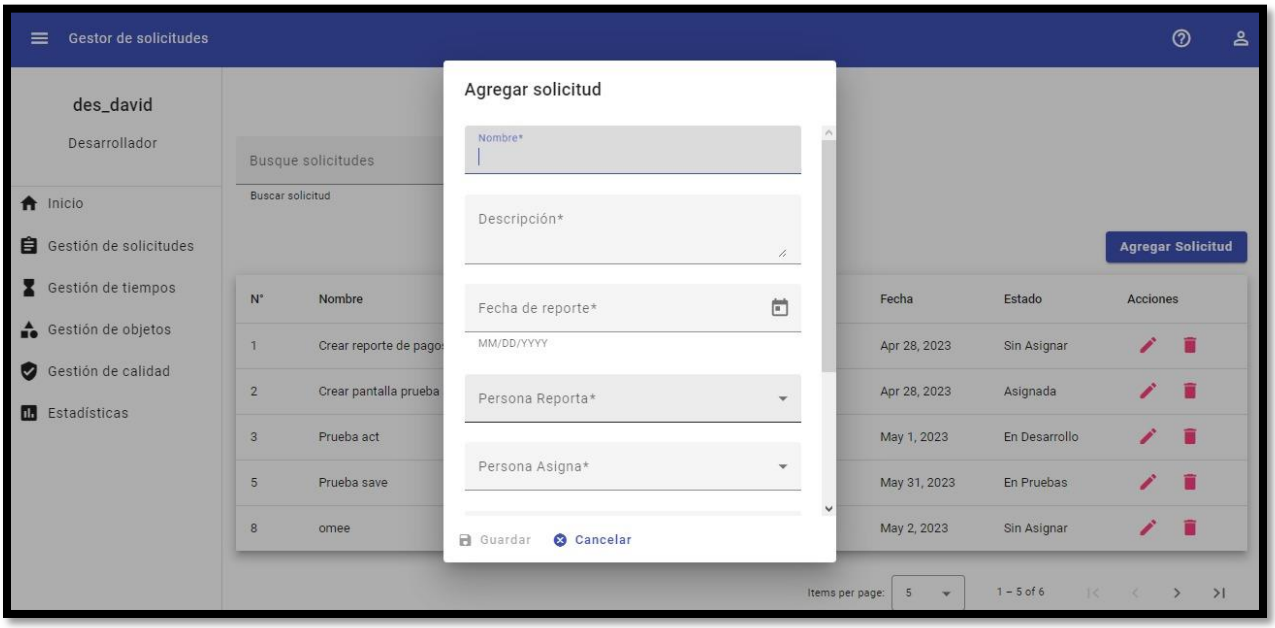


Figura 18

Módulo gestión de tiempos.

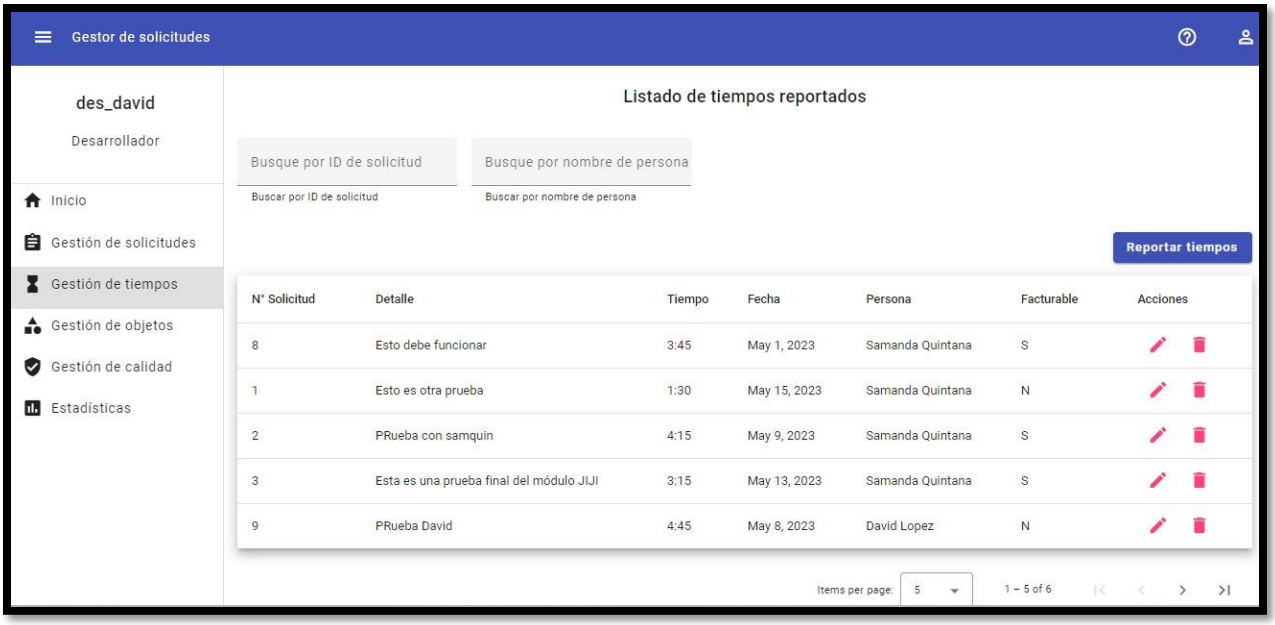


Figura 19

Gestión de objetos.

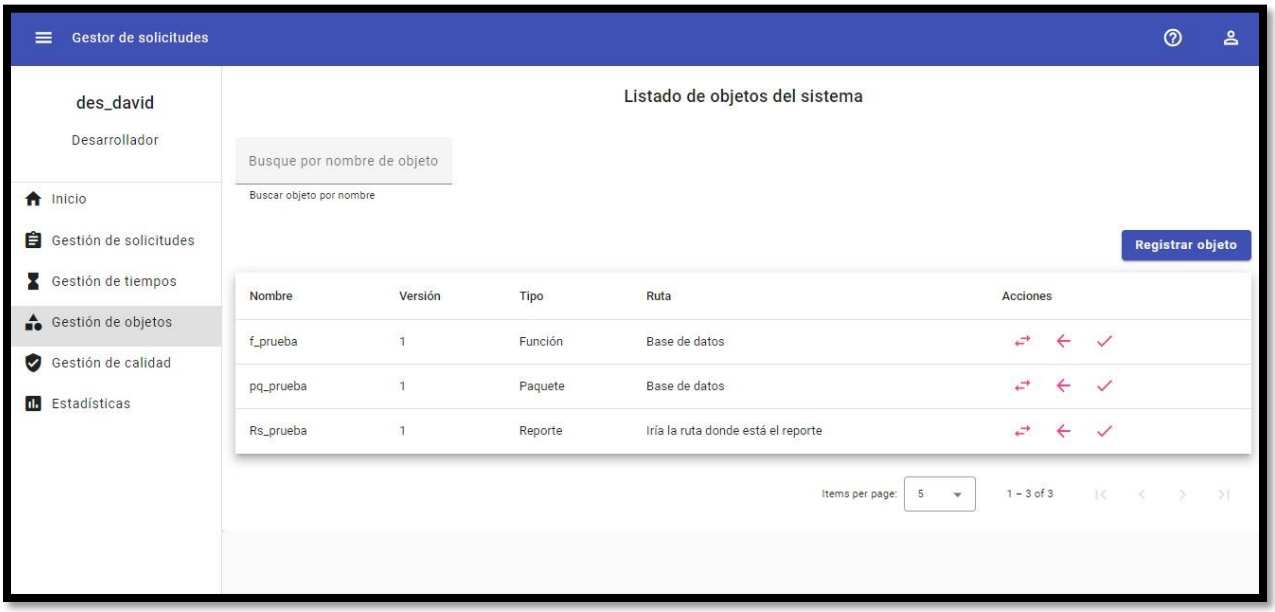


Figura 20

Añadir objetos solicitados.

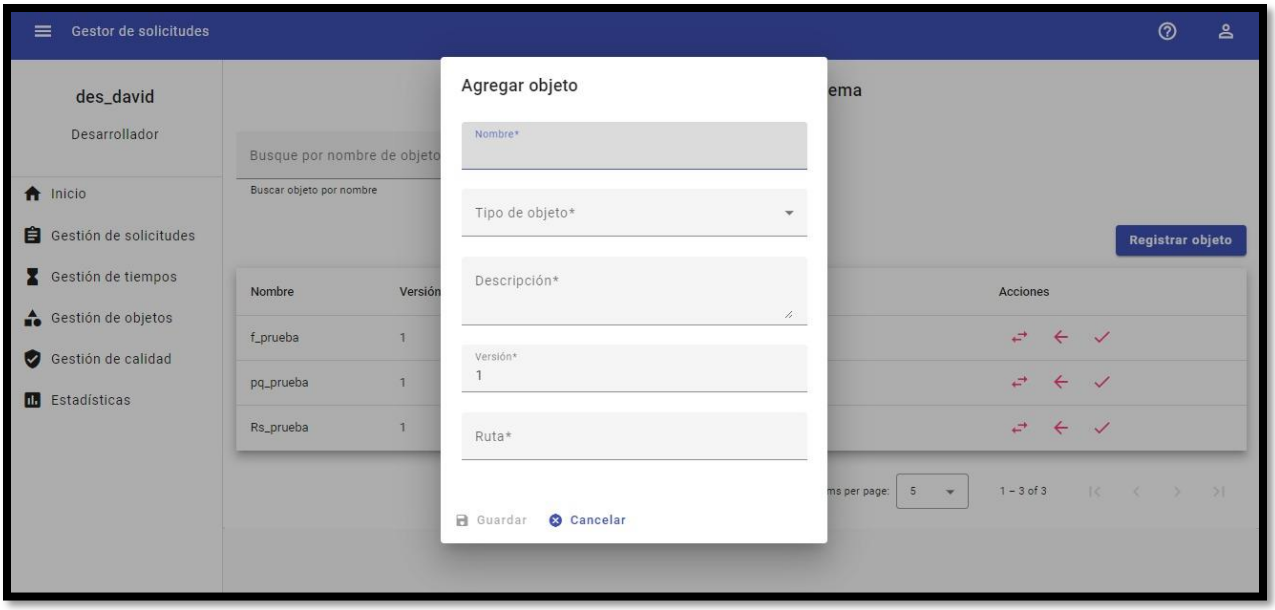


Figura 21

Acciones para realizar sobre los objetos.



Figura 22

Módulo de objetos para usuario administrador.

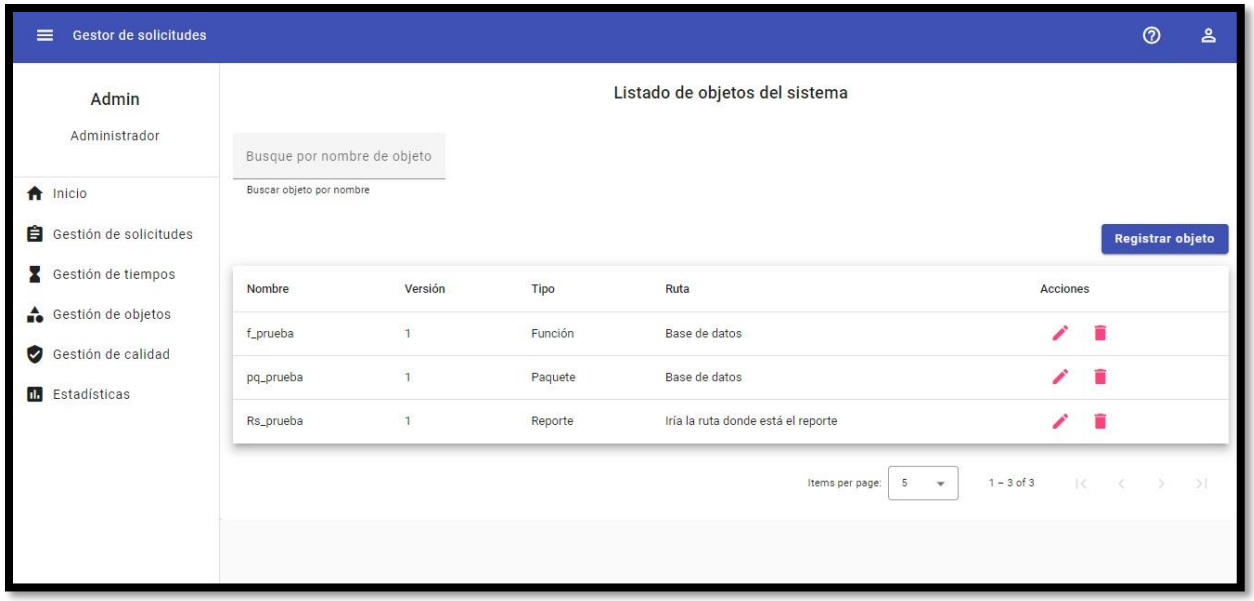
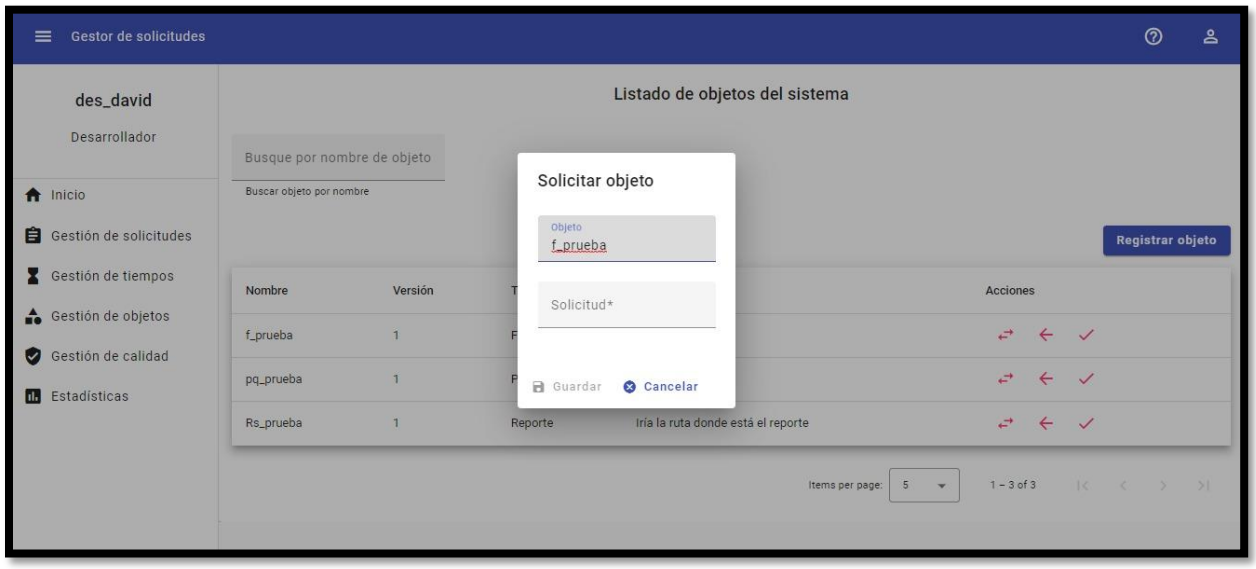


Figura 23

Solicitud de objetos.



Discusión

Una vez aplicado y desarrollado el software, se efectúan los análisis correspondientes sobre los resultados obtenidos, las novedades identificadas y aquellas variables que no fueron contempladas al inicio del proyecto, pero de las cuales se les brindó el respectivo manejo para atenderlas y brindar la

debida solución. A raíz de lo anterior, se realizan intervenciones en la codificación y documentación elaborado, con el fin de moldear la novedad identificada y solventarla de raíz.

Para gestionar y mitigar lo anterior, se manejaron las metodologías ágiles según el planteamiento y lineamientos que contiene cada una, en donde se corroboró que la metodología de desarrollo XP es efectiva para agilizar el desarrollo del software de una manera progresiva, obteniendo constantes resultados e implementaciones paralelas.

La validez de los resultados es congruente con respecto a la propuesta de desarrollo planteada, dado que el software cumple a cabalidad con respecto a la necesidad bosquejada en el actual proyecto de investigación.

Por ende, de acuerdo con todo el ciclo de vida del software, a la gestión documental y ciencia de la investigación, se obtienen las siguientes conclusiones:

- La aplicación del sistema de información para la gestión de novedades y solicitudes reportadas en la etapa de mantenimiento del ciclo de vida del software procesa y cumple con cada requerimiento funcional y no funcional planteado.
- Se deduce acción de mejora y optimización en los procesos operativos que conlleva el reporte y gestión de una novedad, dado que deja de ser un proceso manual y se convierte en automático.
- Se da cumplimiento a cada objetivo específico planteado en el actual proyecto de investigación, en el cual se contempló de manera inmersa todo lo que conlleva el ciclo de vida del software, cada una de sus etapas y sus fases.

Referencias

PRESSMAN, R. S. (2002). INGENIERIA DEL SOFTWARE: UN ENFOQUE PRACTICO (7a. ed.). MCGRAW-HILL, 2010.

PANTALEO, G. RINAUDO, L. (2015). INGENIERIA DE SOFTWARE (1a. ed.).

GÓMEZ, P. S. MORALEDA, G. E. (2020). APROXIMACIÓN A LA INGENIERÍA DEL SOFTWARE (2a. ed.).

NORIEGA, M. R. (2015). EL PROCESO DE DESARROLLO DE SOFTWARE (2a. ed.).

LAINEZ, F. J. (2015). DESARROLLO DE SOFTWARE ÁGIL. EXTREMME PROGRAMMING Y SCRUM (2a. ed.).

TRUJILLO, S. D. CHÁVEZ B. A. (2016). SISTEMA DE CONTROL DE VERSIONES PARA EL DESARROLLO DE SOFTWARE SEGURO.

MENESES, C. A. (2017). PROCEDIMIENTO PARA LA IMPLEMENTACION DE SOFTWARE PARA EL CONTROL DE VERSIONES EN AMBIENTES COLABORATIVOS Y DISTRIBUIDOS.

ABRUTSKY, A. (2016). IMPLANTACIÓN DE UN SISTEMA DE CONTROL DE VERSIONES.

NURLISA, Z. N. NGAH, A. DERAMAN, A. (2018). VERSION CONTROL SYSTEM: A REVIEW. 3rd International Conference on Computer Science and Computational Intelligence 2018.

GEISSHIRT, K. ZATTIN, E. OLSSON, A. VOSS, R. (2018). GIT VERSION CONTROL COOCKBOOK. Leverage version control to transform your development workflow and boost productivity (Second Edition.).

MARTELO, R., MADERA, J. BETÍN, A. (2015). SOFTWARE PARA GESTIÓN DOCUMENTAL, UN COMPONENTE MODULAR DEL SISTEMA DE GESTIÓN DE SEGURIDAD DE LA INFORMACIÓN (SGSI).

LÓPEZ, M. R. (2015). METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE APLICADAS A LA GESTIÓN DE PROYECTOS EMPRESARIALES.

ASTIGARRAGA, J., CRUZ-ALONSO, V. 2022. ¡SE PUEDE ENTENDER CÓMO FUNCIONAN GIT Y GITHUB!. ECOSISTEMAS 31(1): 2332.

RAM, K. 2013. GIT CAN FACILITATE GREATER REPRODUCIBILITY AND INCREASED TRANSPARENCY IN SCIENCE. SOURCE CODE FOR BIOLOGY AND MEDICINE 8: 7.

BLISCHAK, J.D., DAVENPORT, E.R., WILSON, G. 2016. A QUICK INTRODUCTION TO VERSION CONTROL WITH GIT AND GITHUB. PLOS COMPUTATIONAL BIOLOGY 12: E1004668.

FERNANDEZ, A. (2008), GIT, INSTALACIÓN Y CARACTERÍSTICAS, DIALNET (N° 97), PÁGS. 28-31

BERUBE, D. (2011) TRUCOS GIT, DIALNET (Nº. 73), PÁGS. 57-60