



**UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES**

CENTRO DE CIENCIAS BÁSICAS

DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN

TRABAJO PRÁCTICO

**SEGUIMIENTO Y EVALUACIÓN DE LA METODOLOGÍA DE
PROCESOS EN LA GESTIÓN DE REQUERIMIENTOS CON
HERRAMIENTA JIRA**

QUE PRESENTA

L.I CINTHYA ELIZABETH PEDROZA MERCADO

**PARA OPTAR POR EL GRADO DE MAESTRÍA EN INFORMÁTICA Y
TECNOLOGÍAS COMPUTACIONALES**

TUTOR

MTRO. JORGE EDUARDO MACÍAS LUÉVANO

INTEGRANTES DEL COMITÉ TUTORAL

ASESORA: DRA. LIZETH ITZIGUERY SOLANO ROMO

ASESORA: DRA. LAURA ARMINDA GARZA GONZÁLEZ

Aguascalientes, Ags, 13 de junio de 2022

M. EN C. JORGE MARTÍN ALFÉREZ CHÁVEZ
DECANO DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE

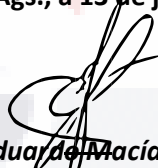
Por medio del presente como **TUTOR** designado del estudiante **CINTHYA ELIZABETH PEDROZA MERCADO** con ID **92194** quien realizó el *trabajo práctico* titulado: **SEGUIMIENTO Y EVALUACIÓN DE LA METODOLOGÍA DE PROCESOS EN LA GESTIÓN DE REQUERIMIENTOS CON HERRAMIENTA JIRA**, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que *ella* pueda proceder a imprimirlo así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE

“Se Lumen Proferre”

Aguascalientes, Ags., a 13 de junio de 2022.



Mtro. Jorge Eduardo Macías Luévano
Tutor de *trabajo práctico*

c.c.p.- Interesado

c.c.p.- Secretaría Técnica del Programa de Posgrado

M. EN C. JORGE MARTÍN ALFÉREZ CHÁVEZ
DECANO DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE


Por medio del presente como **Miembros del Comité Tutorial** designado del estudiante **CINTHYA ELIZABETH PEDROZA MERCADO** ID 92194 quien realizó el *trabajo práctico* titulado: **SEGUIMIENTO Y EVALUACIÓN DE LA METODOLOGÍA DE PROCESOS EN LA GESTIÓN DE REQUERIMIENTOS CON HERRAMIENTA JIRA**, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia damos nuestro consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que nos permitimos emitir el **VOTO APROBATORIO**, para que *ella* pueda proceder a imprimirlo así como continuar con el procedimiento administrativo para la obtención del grado.


Ponemos lo anterior a su digna consideración y sin otro particular por el momento, le enviamos un cordial saludo.

ATENTAMENTE

"Se Lumen Proferre"

Aguascalientes, Ags., a 13 de junio de 2022.


Mtro. Jorge Eduardo Macías Luévano
Tutor de trabajo práctico


Dra. Lizeth Itziguery Solano Romo
Asesora de trabajo práctico


Dra. Laura Arminda Garza González
Asesora de trabajo práctico

c.c.p.- Interesado

c.c.p.- Secretaria Técnica del Programa de Posgrado



DICTAMEN DE LIBERACION ACADEMICA PARA INICIAR LOS TRAMITES DEL EXAMEN DE GRADO



Fecha de dictaminación dd/mm/aaaa: 13/06/2022

NOMBRE: Cinthya Elizabeth Pedraza Mercado **ID** 92194

PROGRAMA: Maestría en Informática y Tecnologías Computacionales **LGAC (del posgrado):** Gestión de Sistemas y de Tecnologías de Información para Mejorar Competitividad, Innovación y Cambio Organizacional

TIPO DE TRABAJO: () Tesis () Trabajo Práctico

TITULO: SEGUIMIENTO Y EVALUACIÓN DE LA METODOLOGÍA DE PROCESOS EN LA GESTIÓN DE REQUERIMIENTOS CON HERRAMIENTA JIRA

IMPACTO SOCIAL (señalar el impacto logrado): Apoyar a las áreas de sistemas, a través de la herramienta JIRA, para la gestión y seguimiento del mantenimiento de sistemas de información, generando un proceso o dando orden al ya existente y generando información en tiempo real para el cliente.

INDICAR	SI	NO	N.A. (NO APLICA)	SEGÚN CORRESPONDA:
<i>Elementos para la revisión académica del trabajo de tesis o trabajo práctico:</i>				
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	El trabajo es congruente con los LGAC del programa de posgrado
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	La problemática fue abordada desde un enfoque multidisciplinario
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Existe coherencia, continuidad y orden lógico del tema central con cada apartado
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Los resultados del trabajo dan respuesta a las preguntas de investigación o a la problemática que aborda
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Los resultados presentados en el trabajo son de gran relevancia científica, tecnológica o profesional según el área
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	El trabajo demuestra más de una aportación original al conocimiento de su área
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Las aportaciones responden a los problemas prioritarios del país
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Generó transferencia del conocimiento o tecnológica
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cumple con la ética para la investigación (reporte de la herramienta antiplagio)
<i>El egresado cumple con lo siguiente:</i>				
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cumple con lo señalado por el Reglamento General de Docencia
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cumple con los requisitos señalados en el plan de estudios (créditos curriculares, optativos, actividades complementarias, estancia, profesora, etc)
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cuenta con los votos aprobatorios del comité tutorial, en caso de los posgrados profesionales si tiene solo tutor podrá liberar solo el tutor
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cuenta con la carta de satisfacción del Usuario
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Coincide con el título y objetivo registrado
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tiene congruencia con cuerpos académicos
SI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tiene el CVU del Conacyt actualizado
N.A.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Tiene el artículo aceptado o publicado y cumple con los requisitos institucionales (en caso que proceda)
<i>En caso de Tesis por artículos científicos publicados</i>				
N.A.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Aceptación o Publicación de los artículos según el nivel del programa
N.A.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	El estudiante es el primer autor
N.A.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	El autor de correspondencia es el Tutor del Núcleo Académico Básico
N.A.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	En los artículos se ven reflejados los objetivos de la tesis, ya que son producto de este trabajo de investigación.
N.A.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Los artículos integran los capítulos de la tesis y se presentan en el idioma en que fueron publicados
N.A.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	La aceptación o publicación de los artículos en revistas indexadas de alto impacto

Con base a estos criterios, se autoriza se continúen con los trámites de titulación y programación del examen de grado:

SI
 No

FIRMAS

Elaboró:

* NOMBRE Y FIRMA DEL CONSEJERO SEGÚN LA LGAC DE ADSCRIPCIÓN

Dr. Cesar Eduardo Velázquez Amador

NOMBRE Y FIRMA DEL SECRETARIO TÉCNICO:

MITC. Jorge Eduardo Macías Luévano

* En caso de conflicto de intereses, firmará un reporto miembro del NAR de la LGAC correspondiente distinto al tutor o miembro del comité tutorial, según sea el caso.

Revisó:

NOMBRE Y FIRMA DEL SECRETARIO DE INVESTIGACIÓN Y POSGRADO:

Dr. Hayden Martínez Rivalcaba

Autorizó:

NOMBRE Y FIRMA DEL DECANO:

M. en C. Jorge Martín Alfaro Chávez

Nota: procede el trámite para el Depto. de Apoyo al Posgrado

En cumplimiento con el Art. 101C del Reglamento General de Docencia que a la letra señala entre las funciones del Consejo Académico: ... Conde la eficiencia terminal del programa de posgrado y el Art. 101F las funciones del Secretario Técnico, Revise el seguimiento de los alumnos.

Agradecimientos

Agradezco al Instituto para el Desarrollo de la Sociedad del Conocimiento del Estado de Aguascalientes (IDSCEA) por brindar apoyos económicos que nos ayudan en ese camino de superación profesional.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por mi beca, que me permitió poder continuar con mis estudios.

A la Universidad Autónoma de Aguascalientes por brindarme la oportunidad ampliar mis conocimientos profesionales en sus aulas con maestros excelentes y todas las facilidades para que este camino me llevara al objetivo de concluir la maestría.

Al Maestro Jorge Eduardo Macías Luévano por todo su apoyo, confianza y su conocimiento para guiarme en este trabajo práctico.

A la Doctora Lizeth Itziguery Solano Romo por su apoyo, comprensión y por formar parte de este trabajo.

A la Doctora Laura Arminda Garza González por su apoyo, seguimiento y por formar parte de este trabajo.

A la Gerencia de TI del Corporativo Ortus, por permitirme implementar en el área de TI mis conocimientos y mi trabajo práctico.

Dedicatorias

Para mi esposo Luis Eduardo Galván por su apoyo, comprensión y motivación que siempre me acompañaron en este camino para cumplir mis objetivos y superarme profesionalmente.

Para mis padres Fernando Pedroza y Paty Mercado, por su apoyo incondicional, su ejemplo de lucha y perseverancia, a mis hermanos Karla, Fernanda y Carlos por su apoyo y motivación.

Índice General

Contenido

Resumen	4
Abstract	5
Introducción	6
Antecedentes	7
Diagnóstico	8
Justificación	8
Sector, población o grupo afectado por la problemática	8
Objetivo General	9
Objetivos específicos	9
Fundamentación teórica	10
Sistema de Información	10
Proceso	12
JIRA	15
Mantenimiento de Sistemas	22
Agilidad	27
Metodología ágil	28
El manifiesto ágil	28
Diseño de la intervención o aplicación de la metodología de la intervención	36
Descripción de la intervención	36
Descripción de la intervención	43
Resultados obtenidos.....	57
Evaluación de la intervención	58
Alcances y limitaciones de la intervención	59
Aportes a la organización o los beneficios	61
Recomendaciones para desarrollos futuros.....	61
Conclusiones	62
Bibliografía	64

Índice de Gráficas o Figuras

Ilustración 1 Ejemplo de Proceso (Elementos de Un Sistema de Gestion de Calidad Basado En Las Normas ISO 9001:2008 - Monografias.Com, n.d.).....	12
Ilustración 2 Flujogramas Matricial (Pardo Alvarez, 2012).....	13
Ilustración 3 Flujograma lineal (Pardo Alvarez, 2012).....	14
Ilustración 4 Simbología (Pardo Alvarez, 2012).....	15
Ilustración 5 Tablero (Jira Software de Seguimiento de Proyectos e Incidencias, n.d.)...	17
Ilustración 6 Tablero Scrum (Jira Software de Seguimiento de Proyectos e Incidencias, n.d.).....	17
<i>Ilustración 7 Tablero Kanban (Jira Software de Seguimiento de Proyectos e Incidencias, n.d.).....</i>	<i>18</i>
Ilustración 8 Hojas de ruta (Jira Software de Seguimiento de Proyectos e Incidencias, n.d.).....	19
Ilustración 9 Creación de informes ágiles (Jira Software de Seguimiento de Proyectos e Incidencias, n.d.).....	20
Ilustración 10 Flujo de Trabajo (Jira Software de Seguimiento de Proyectos e Incidencias, n.d.).....	21
Ilustración 11 Hoja de ruta de producto (Jira Software de Seguimiento de Proyectos e Incidencias, n.d.).....	21
Ilustración 12 Las fuentes del mantenimiento del software (Uaa - Mantenimiento y Evolución de Sistemas de Información, n.d.).....	23
Ilustración 13 Origen de los defectos del software (Uaa - Mantenimiento y Evolución de Sistemas de Información, n.d.).....	24
Ilustración 14 Valores del manifiesto ágil (Hernández Bejarno & Baquero Rey, n.d.).....	29
Ilustración 15 Principios del Manifiesto ágil (Hernández Bejarno & Baquero Rey, n.d.)....	30
Ilustración 16 Características generales en las metodologías ágiles (Hernández Bejarno & Baquero Rey, n.d.).....	32
Ilustración 17 Características comunes en metodologías ágiles (Hernández Bejarno & Baquero Rey, n.d.).....	35
Ilustración 18 Lista de Requerimientos (Pedroza C).....	37
Ilustración 19 Sprint 2 iniciado (Pedroza C).....	38
Ilustración 20 Tableros activos del Sprint 2 (Pedroza C).....	38
Ilustración 21 Tableros activos del Sprint 2 (Pedroza C).....	39
Ilustración 22 Sprint 2 por finalizar (Pedroza C).....	40
Ilustración 23 Finalización Sprint 2 (Pedroza C).....	40
Ilustración 24 Avance de Sprint 3 (Pedroza C).....	41
Ilustración 25 Finalización del Sprint 3 (Pedroza C).....	41
Ilustración 26 Finalización del Sprint 4 (Pedroza C).....	42
Ilustración 27 Lista de Requerimientos (Pedroza C).....	44
Ilustración 28 Correos de notificación (Pedroza C).....	44
Ilustración 29 Hoja de ruta Inicial (Pedroza C).....	45

Ilustración 30 Sprint 1 Iniciado (Pedroza C) 46

Ilustración 31 Tablero activo del Sprint 1 (Pedroza C) 47

Ilustración 32 Sprint 1 con actividades nuevas (Pedroza C)..... 47

Ilustración 33 Sprint 1 con las actividades en curso (Pedroza C) 48

Ilustración 34 Finalización Sprint 1 (Pedroza C) 49

Ilustración 35 Tablero de Sprint 2 (Pedroza C) 49

Ilustración 36 Tablero Sprint 2 (Pedroza C) 50

Ilustración 37 Finalización del Sprint 2 (Pedroza C) 50

Ilustración 38 Tablero Sprint 3 (Pedroza C) 51

Ilustración 39 Finalización del Sprint 3 (Pedroza C) 52

Ilustración 40 Tablero Sprint 4 (Pedroza C) 52

Ilustración 41 Informe de Velocidad (Pedroza C)..... 53

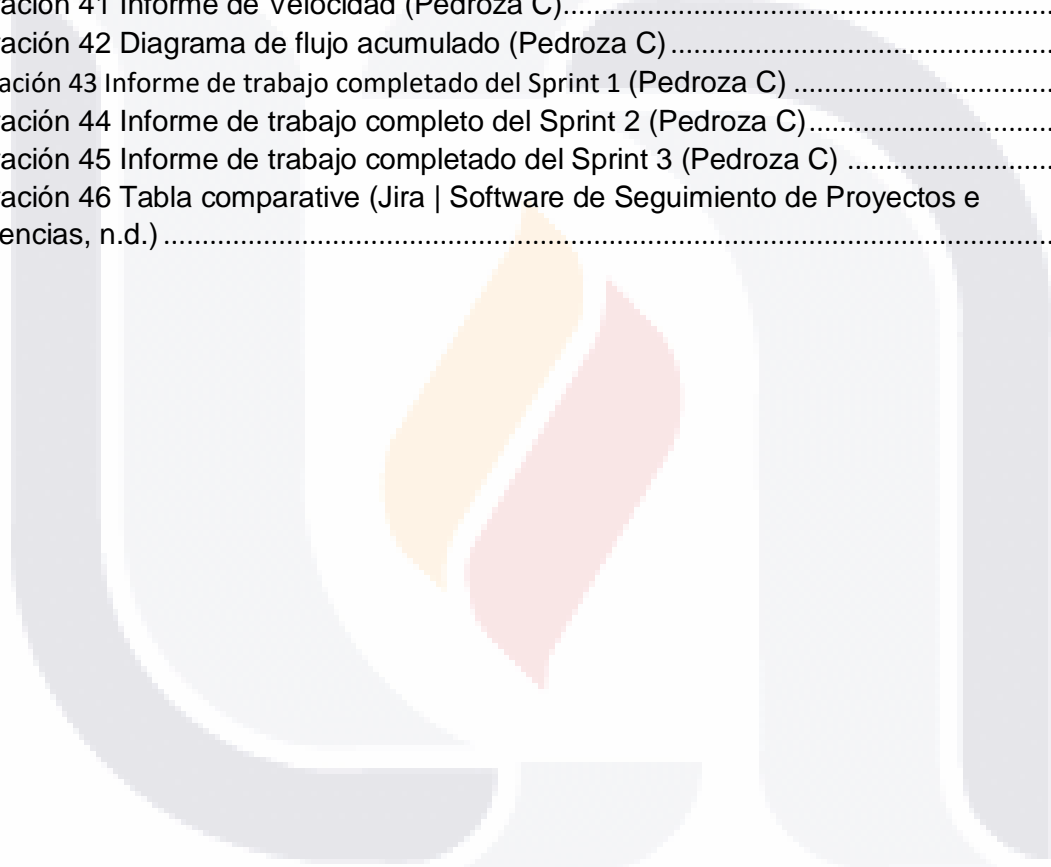
Ilustración 42 Diagrama de flujo acumulado (Pedroza C) 54

Ilustración 43 Informe de trabajo completado del Sprint 1 (Pedroza C) 55

Ilustración 44 Informe de trabajo completo del Sprint 2 (Pedroza C)..... 55

Ilustración 45 Informe de trabajo completado del Sprint 3 (Pedroza C) 56

Ilustración 46 Tabla comparative (Jira | Software de Seguimiento de Proyectos e Incidencias, n.d.) 60



Resumen

El presente trabajo práctico presenta el uso de la herramienta JIRA para dar seguimiento a todos los requerimientos que son parte del mantenimiento de un sistema, se busca demostrar que la metodología que se sigue con la herramienta permite dar atención y solución a las necesidades de la empresa que son resueltas por un sistema de software. Durante la investigación se identificó que las empresas buscan dar solución a sus necesidades en el menor tiempo posible y con el mejor seguimiento, asegurando la calidad del producto, dando a conocer el estatus real de cada solicitud, así como conocer de manera gráfica o a través de informes para la toma de decisiones.



Abstract

This practical work presents the use of the JIRA tool to monitor all the requirements that are part of the maintenance of a system, it seeks to demonstrate that the methodology that is followed with the tool allows giving attention and solution to the needs of the company which are resolved by a software system. During the investigation, it was identified that companies seek to provide a solution to their needs in the shortest possible time and with the best follow-up, ensuring the quality of the product, making known the real status of each request, as well as knowing graphically through reports for decision-making.



Introducción

En la actualidad las empresas enfrentan diversas situaciones en la atención y solución de requerimientos para el mantenimiento de software, pueden contar con procesos, pero no tener una gestión precisa o pueden conocer metodologías, pero no aplicarlas e incluso contar con herramientas y un proceso, pero no utilizarlo.

Existen metodologías para el mantenimiento de software como es MANTEMA la cual es una metodología que fue desarrollada entre el grupo Alarcos del Departamento de Informática de la Universidad de Castilla La Mancha y la empresa Atos ODS, ambos de España.

El objetivo de esta metodología es hacer que el mantenimiento sea un proceso controlable y medible, mediante la identificación y definición clara de todos los elementos que intervienen en el proceso. El proceso está compuesto por varias actividades y cada actividad está compuesta por varias tareas.

En el estándar ISO/IEC 12207, de Procesos del Ciclo de Vida del Software, se establece que el propósito del proceso de mantenimiento es conservar la capacidad del sistema de proporcionar un servicio, lo que incluyen aspectos como documentación, paso en el que se definen las actividades; gestión de la configuración; gestión de aseguramiento de la calidad, que garantiza que se cumplan los requerimientos especificados y los planes establecidos; revisión conjunta, auditoría, verificación y validación.

Las necesidades de mantenimiento se dan para garantizar que el software cumpla satisfactoriamente con los requerimientos solicitados y se aplica a cualquier desarrollo independiente del modelo de ciclo de vida utilizado.

El mantenimiento debe ser realizado con el objetivo de:

- Corregir fallas.
- Mejorar el diseño.
- Implementar mejoras.
- Definir interfaces con otros sistemas.
- Adaptar programas a diferentes tipos de hardware, software, características del sistema y capacidad de las redes de telecomunicaciones.
- Migrar software heredado
- Retirar el software

En la actualidad el software se está convirtiendo en el corazón de las empresas, por lo que es de suma importancia que los cambios y la evolución de dicho software

sean controlados permitiendo que estos sistemas crezcan siendo más útiles y con más funcionalidad.

Jira Software es una herramienta ágil de gestión de proyectos compatible con cualquier **metodología** ágil, ya sea scrum, Kanban o la propia. Se pueden tener desde tableros hasta informes ágiles, se puede planificar, supervisar y gestionar todos los proyectos de desarrollo de software ágil con una sola herramienta.

Jira Software se lanzó en 2002 como una herramienta de seguimiento de incidencias y gestión de proyectos para los equipos. Desde entonces, más de 65 000 empresas en todo el mundo han adoptado Jira por su flexibilidad para admitir cualquier tipo de proyecto y la posibilidad de trabajar con miles de aplicaciones e integraciones.

Equipos de 2 y 2000 personas de empresas de todos los tamaños



Antecedentes

Hoy en día, dentro de las empresas, las áreas de Tecnologías de Información (TI) tienen sistemas de información que requieren ser actualizados y mejorados constantemente, debido a las necesidades y actualizaciones que se van generando en el día a día, se ha identificado que no siempre se aplican metodologías y que no siempre se puede dar soluciones de manera inmediata, así como tampoco se lleva un seguimiento, control y gestión de los requerimientos del sistema, por lo anterior se busca contar con herramientas que permitan ese seguimiento al mantenimiento de software de calidad y que pueda aplicarse a través del uso de una metodología de mantenimiento de software.

Diagnóstico

Se ha identificado en algunas empresas que dentro de su área de TI, se tienen sistemas de software que requieren modificaciones constantes y frecuentes, lo que implica un seguimiento muy específico y una atención detallada y precisa, de manera que no se asuman requerimientos que no están considerados y que se dedique esfuerzo en actividades no necesarias, por lo que se identifica la necesidad de utilizar una metodología para atender los requerimientos con características limitadas y que no abarcan la magnitud de un proyecto en dimensión, tecnología esfuerzo y desarrollo. Asimismo, se considera el seguimiento de estas solicitudes en una herramienta de código abierto que facilita la gestión y administración ofreciendo seguimiento oportuno.

Justificación

El Trabajo práctico busca conocer si la herramienta y la metodología que se han utilizado cumple con el proceso de mantenimiento en el área de TI de cualquier empresa, lo cual beneficiará a los usuarios solicitantes de cada requerimiento, al líder que Gestiona todo el proceso, a los interesados en el seguimiento y ejecución de los cambios en el sistema, también beneficiará a los clientes externos, ya que las modificaciones del sistema tienen por objetivo brindar mejores servicios y productos.

Los motivos que me llevaron a buscar un seguimiento y una evaluación del uso de la herramienta para el mantenimiento de las aplicaciones, fue buscar mejorar los tiempos de atención, así como la mejora en la calidad de los desarrollos y un control de aquellos requerimientos que sean regulatorios y que las Empresas o áreas de TI requieran de manera urgente con la finalidad de brindar un beneficio a los usuarios o clientes, ya sea por peticiones o situaciones que no están dentro del proceso de mantenimiento.

Sector, población o grupo afectado por la problemática

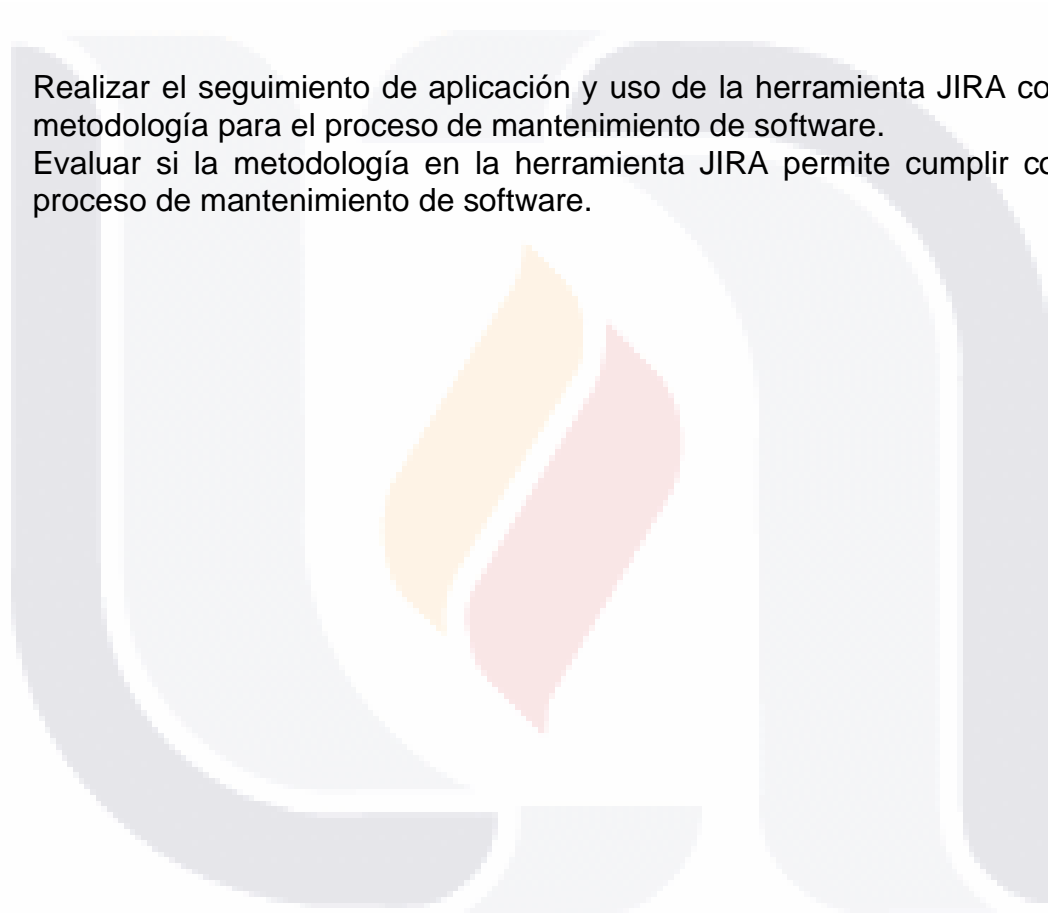
Las empresas o áreas de TI que cuentan o requieren contar con un seguimiento y una mejor atención a las necesidades de mejoras y mantenimiento de software.

Objetivo General

Realizar el seguimiento de la metodología del proceso de mantenimiento a través de la herramienta JIRA y la evaluación de la herramienta para el seguimiento del proceso de mantenimiento de software en el área de TI de las Empresas.

Objetivos específicos

- Realizar el seguimiento de aplicación y uso de la herramienta JIRA con su metodología para el proceso de mantenimiento de software.
- Evaluar si la metodología en la herramienta JIRA permite cumplir con el proceso de mantenimiento de software.



Fundamentación teórica

Sistema de Información

El sistema de información: Es el conjunto formal de procesos que operando sobre una colección de datos estructurada de acuerdo con las necesidades de una empresa, recopila, elabora y distribuye la información necesaria para la operación de dicha empresa y para las actividades de dirección y control correspondientes, apoyando, al menos en parte, los procesos de toma de decisiones necesarios para desempeñar las funciones de negocio de la empresa de acuerdo con su estrategia (Lapiedra Alcamí et al., 2011)

La definición del párrafo anterior está enfocada a los sistemas de información formal, que es la parte del sistema de información que toda la empresa conoce y sabe cómo utilizar. También existen sistemas con información informal, Los sistemas de información informales no son resultado de un proceso diseñado, sino que brindan información de casualidad, esto no significa que se consideren importantes los sistemas de información informales, se mencionan para que se pueda identificar que existen características que los identifican, como son la naturaleza, pueden ser menos estudiables, menos planificables, y seguramente menos dirigibles, al menos desde un punto de vista con beneficios para una Organización. Con esto, se busca transmitir la idea que el sistema de información de una empresa debe estar al servicio de su enfoque de negocio.

Buscando complementar la definición de sistema ya mencionada en el párrafo anterior, se explicará el concepto de sistema informático para aclarar los conceptos y evitar confusiones. El sistema informático consiste en la compleja interconexión de una gran cantidad de componentes de hardware y software, los cuales son básicamente sistemas deterministas y formales, de tal forma que con una entrada específica siempre se obtiene una misma salida. Los sistemas de información son sistemas que se ven influenciados por los objetivos, valores y creencias de individuos y grupos, así como por el desempeño de la tecnología. Así pues, el comportamiento del sistema de información no es determinista y no se ajusta a la representación de ningún modelo algorítmico formal.

Componentes de los sistemas de información

Los sistemas de información engloban: equipos y programas informáticos, telecomunicaciones, bases de datos, recursos humanos y procedimientos (García Bravo, 2000).

Equipos informáticos: Actualmente todas las empresas utilizan Computadoras. Las organizaciones grandes utilizan diversos sistemas computarizados. También hay dispositivos que sirven como entradas para el equipo

de cómputo como son teclado y ratón y dispositivos para producir el output del sistema (impresoras).

Programas informáticos: Hay dos tipos de programas informáticos: programas del sistema y aplicaciones. Los programas del sistema administran los recursos del sistema computarizado y simplifican la programación. Las aplicaciones ayudan directamente al usuario final a hacer su trabajo. Ejemplos de aplicaciones: programas de hoja de cálculo o procesadores de texto.

Bases de datos: Una base de datos es una colección de datos interrelacionados. Una base de datos debe estar organizada para que se pueda acceder a ellos por sus atributos. Las bases de datos son administradas por programas de sistemas conocidos como sistemas de administración de bases de datos (DBMS).

Telecomunicaciones: Las telecomunicaciones son el medio de transmisión electrónica de información a largas distancias. Actualmente los sistemas de información pueden estar conectados en redes locales (LAN) en redes amplias (WAN) y con internet (WEB), gracias a estas conexiones con los sistemas los usuarios tienen acceso a los recursos informáticos de la empresa, como por ejemplo Bases de Datos.

Recursos humanos: se deben diferenciar entre personas especialistas en sistemas de información y usuarios finales. El personal especializado de sistemas de información incluye analistas de sistemas, programadores y operadores. Los usuarios finales, suelen ser la mayoría de las personas en una Organización, son las personas que utilizan los sistemas de información o el resultado que estos generan.

Procedimientos: Los procedimientos están conformados por las políticas y métodos que deben ser considerados al utilizar, operar y mantener un sistema de información.

Funciones del sistema de información: Los sistemas de información son desarrollados en las empresas para ayudar en el desempeño de las actividades que se realizan en cada área donde se tiene un sistema de información. Todo sistema de información lleva a cabo una serie de funciones que pueden ser agrupadas en:

Funciones de captación y recolección de datos.

Funciones de almacenamiento.

Tratamiento de la información.

Distribución o diseminación de la información.

Proceso

Las organizaciones desarrollan procesos para generar los productos y servicios que entregan a sus clientes. Estos procesos constituyen los métodos de trabajo empleados por las organizaciones para dar valor a sus clientes, ya sean internos o externos. (Pardo Alvarez, 2012)

Proceso: conjunto de recursos y actividades interrelacionadas, repetitivas y sistemáticas, mediante las cuales una entrada se convierte en un resultado (salida).

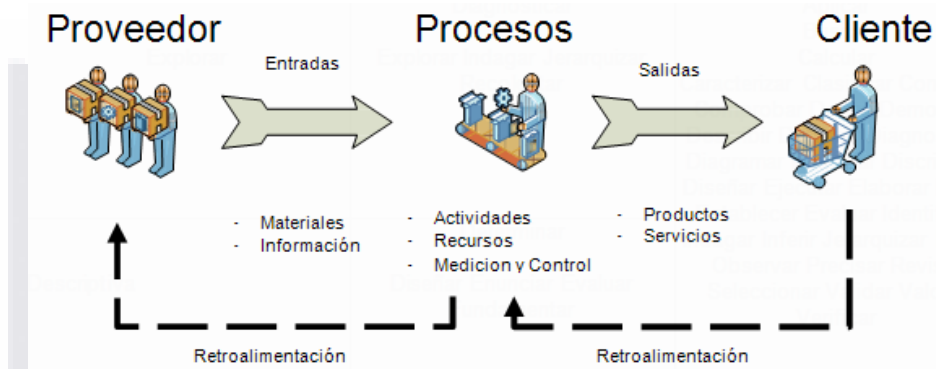


Ilustración 1 Ejemplo de Proceso (Elementos de Un Sistema de Gestion de Calidad Basado En Las Normas ISO 9001:2008 - Monografias.Com, n.d.)

En la ilustración 1 se puede ver un ejemplo de proceso donde se tienen entradas, salidas y la retroalimentación.

Todos los procesos necesitan para su funcionamiento una serie de recursos: personas encargadas de las actividades a realizar, equipos de apoyo, infraestructura, etc. Los recursos son inherentes al desarrollo de las distintas actividades del proceso, ya que sin ellos su ejecución no es posible. Dimensionar adecuadamente estos recursos es una de las principales cuestiones a considerar cuando se pone en marcha un proceso, y debería revisarse de manera periódica. Los procesos están formados por actividades, también denominadas tareas, pasos, acciones, operaciones, etc.

Las actividades de los procesos están interrelacionadas, no existen actividades aisladas. Siempre existirá una actividad precedente y una posterior, excepto en el caso de la primera y última actividad.

El proceso como unidad de gestión

En cualquier organización deben ser conscientes los trabajadores de los procesos desarrollados en ella y deben convertirlos en la referencia de cualquiera de sus actuaciones. Este simple acto puede favorecer a:

- Crear un sentimiento de pertenencia que brinda mayor sentido a las actividades desarrolladas y mitigará la posible sensación de trabajo aislado.
- Mejora la comunicación y el nivel de entendimiento entre los empleados de la organización.
- Incrementar el grado de eficacia con que desarrollan sus actividades.
- Disminuir el estrés asociado al desempeño del trabajo.

El proceso debería ser una referencia en la gestión, desarrollarlo según lo previsto, verificar periódicamente si están alcanzando lo pretendido y mejorar todo aquello que pudiera causar una afectación a la eficacia o eficiencia del resultado final.

Sería muy recomendable que la organización tuviera asimilado el concepto de proceso y hablase en clave de procesos no como algo exclusivo de los gerentes, sino como algo común en el lenguaje de todos los miembros de la organización. Sería altamente positivo considerar los procesos como la unidad de gestión, tratando de manejar procesos tipo en todos los niveles.

Representación gráfica de un proceso

La representación gráfica de un proceso, como un ente individual, se puede realizar mediante una herramienta denominada diagrama de flujo o flujograma.

Tipos de diagramas de flujo

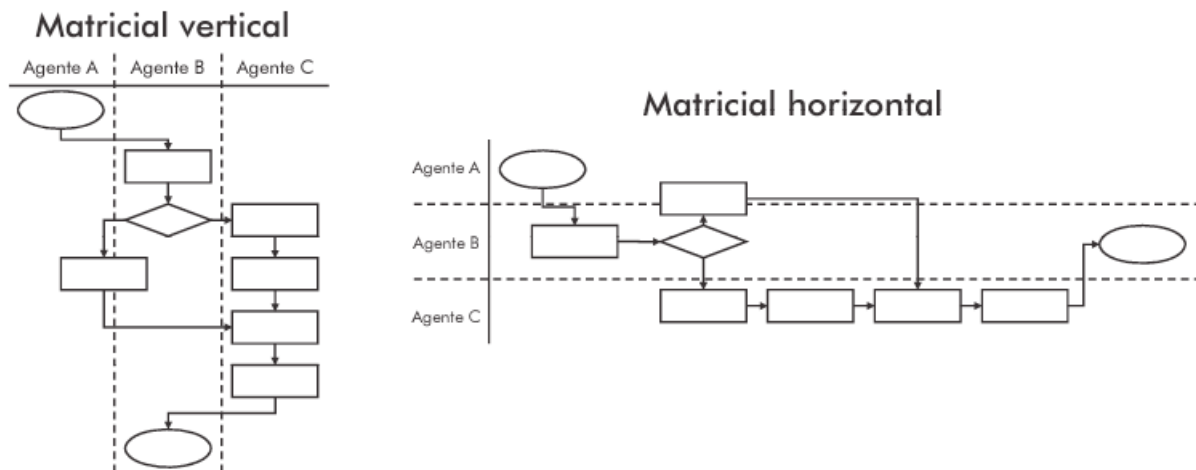


Ilustración 2 Flujogramas Matricial (Pardo Alvarez, 2012)

- Flujogramas de tipo matricial. Esta clase de flujogramas se caracteriza porque los agentes intervinientes en el proceso aparecen en la cabecera del dibujo, y subordinadas a ellos se sitúan las actividades desempeñadas por cada uno. (Pardo Álvarez, 2012)

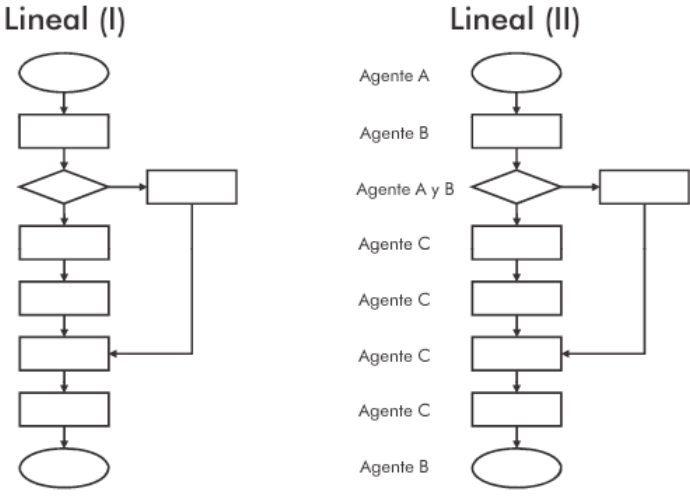


Ilustración 3 Flujograma lineal (Pardo Alvarez, 2012)

- Flujogramas de tipo lineal. En este caso, todas las actividades del proceso aparecen secuenciadas una debajo de la otra, es de muy fácil construcción, pero aporta menos información sobre el proceso.

Simbología y su significado





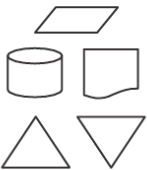
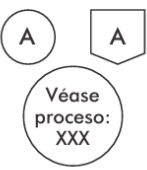
Símbolo	Nombre	Descripción
	Elipse u óvalo	Indica el inicio y el final del diagrama de flujo. Está reservado a la primera y a la última actividad. Un proceso puede tener varios inicios y varios finales
	Rectángulo o caja	Se utiliza para definir cada actividad o tarea. Debe incluir siempre un verbo de acción. Las cajas se pueden numerar
	Rombo	Aparece cuando es necesario tomar una decisión. Incluye siempre una pregunta
	Flecha	Utilizada para unir el resto de símbolos entre sí, indicando la dirección secuencial de las actividades
	Símbolos de entrada y salida	Se utilizan para representar entradas necesarias para ejecutar actividades del proceso, o para recoger salidas generadas durante el desarrollo del mismo El símbolo elegido de entrada se conectará con una flecha hacia la actividad que lo emplea El símbolo elegido de salida se conectará con una flecha desde la actividad de la que surge hacia el símbolo de salida
	Conectores	Usados para representar conexiones con otras partes del flujograma o con otros procesos. Si el proceso es largo y el diagrama de flujo no cabe en una hoja, se suele utilizar algún símbolo para conectar una hoja con otra. Una letra o un número en el interior del símbolo indican que la secuencia enlaza con un símbolo equivalente. También se pueden utilizar para vincular el proceso que estamos dibujando con otro proceso relacionado

Ilustración 4 Simbología (Pardo Alvarez, 2012)

JIRA

Atlassian ha creado la herramientas Jira y Trello, que emplean miles de equipos en todo el mundo. El objetivo de Atlassian es desarrollar increíbles productos, prácticas y entornos de trabajo abiertos para todos ellos. (Jira Software, 2018). Actualmente **Atlassian** Cuenta con una variedad de productos clasificados de la siguiente manera:

- PLANIFICA, SUPERVISA Y OFRECE SOPORTE:
 - Jira Software: Seguimiento de proyectos e incidencias
 - Jira Align: Planificación empresarial ágil
 - Jira Core: Gestión empresarial básica
 - Jira Service Management: ITSM de alta velocidad
 - OpsGenie: Respuesta ante incidentes moderna
 - Statuspage: Comunicación de incidencias
- COLABORA
 - Confluence: Colaboración en documentos
 - Trello: Colabora visualmente en cualquier proyecto

- CODIFICA, COMPILA Y LANZA
 - Bitbucket: Gestión de código Git
 - Sourcetree: Cliente de escritorio de Git y Mercurial
 - Bamboo: Gestión de integración y publicación
- SEGURIDAD E IDENTIDAD
 - Atlassian Access: Seguridad y control para la nube
 - Crowd: Gestión de usuarios para entornos autogestionados

Para este caso práctico se utilizará la herramienta **Jira Software**, conocida como la herramienta de desarrollo de software líder de los equipos ágiles. Jira Software está diseñado para que todos los miembros de tu equipo de software puedan planificar, supervisar y publicar software de gran calidad.

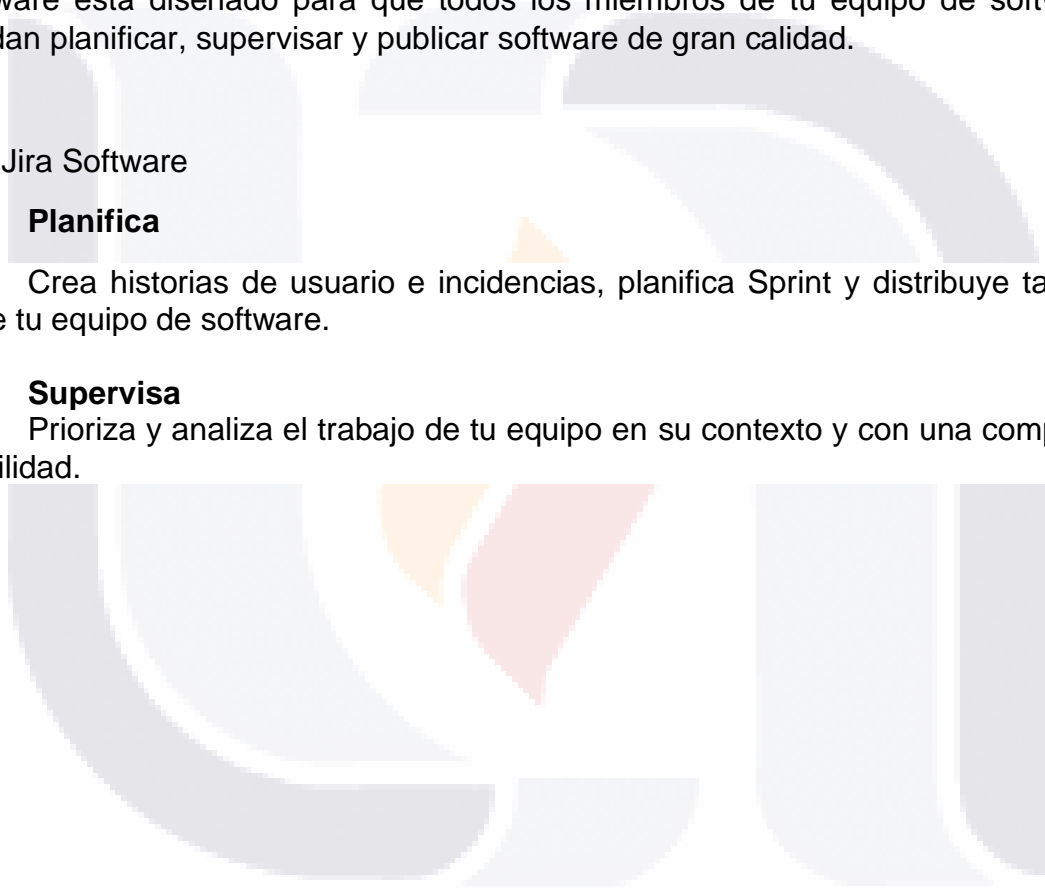
Con Jira Software

Planifica

Crea historias de usuario e incidencias, planifica Sprint y distribuye tareas entre tu equipo de software.

Supervisa

Prioriza y analiza el trabajo de tu equipo en su contexto y con una completa visibilidad.



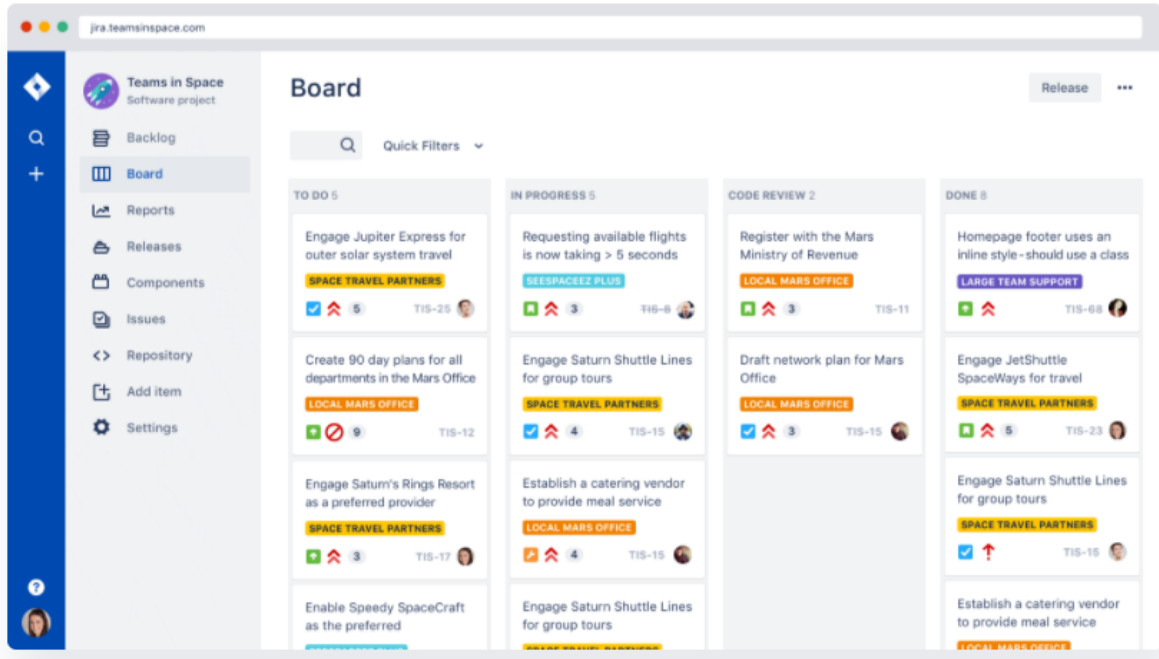


Ilustración 5 Tablero (Jira | Software de Seguimiento de Proyectos e Incidencias, n.d.)

Para la planificación y supervisión se tienen las opciones de tablero scrum o tableros Kanban

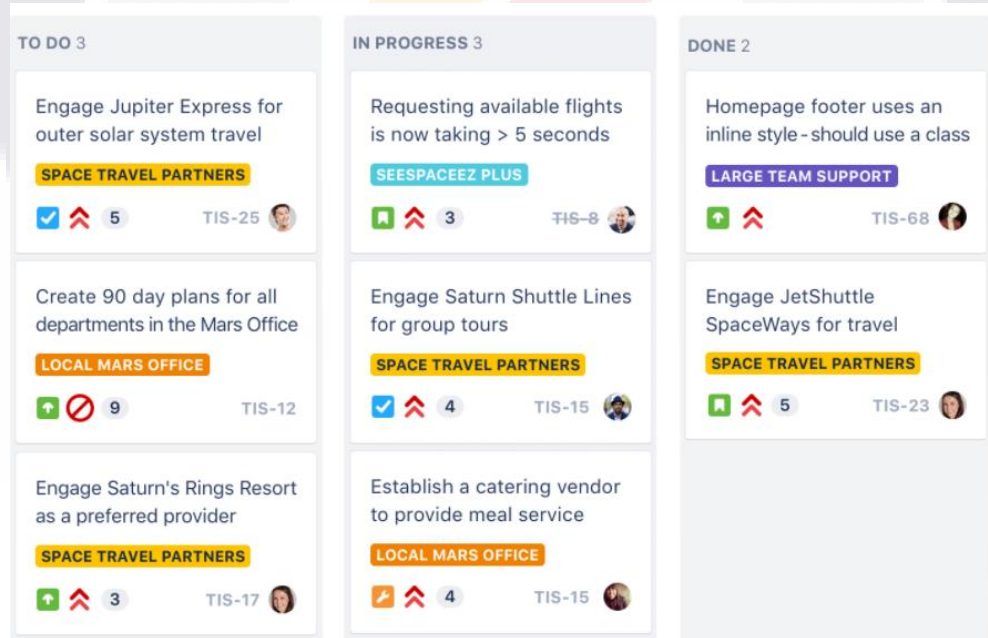


Ilustración 6 Tablero Scrum (Jira | Software de Seguimiento de Proyectos e Incidencias, n.d.)

En la Ilustración 6 se muestra un ejemplo de Tableros Scrum donde Los equipos ágiles pueden mantenerse enfocados en ofrecer valor iterativo e incremental, lo más rápido posible, con tableros de scrum personalizables.

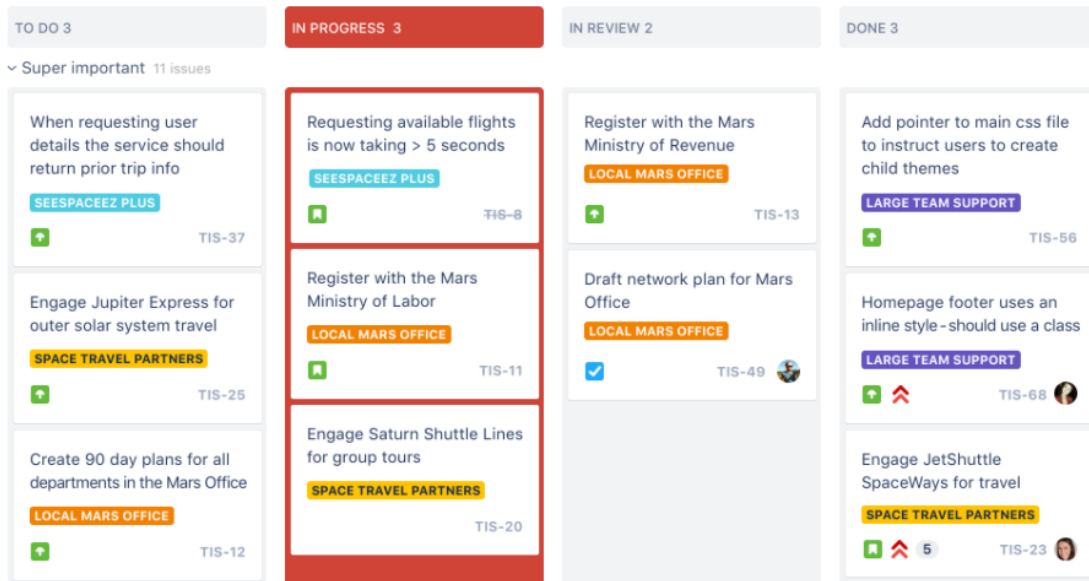


Ilustración 7 Tablero Kanban (Jira | Software de Seguimiento de Proyectos e Incidencias, n.d.)

En la ilustración 7 se muestra un Tablero Kanban, donde gracias a los paneles flexibles, tu equipo dispondrá de total visibilidad respecto al futuro para lograr constantemente la máxima producción en ciclos de mínima duración.

Lanza

Realiza lanzamientos con confianza y seguridad, sabiendo que la información que tienes es siempre la más actualizada.

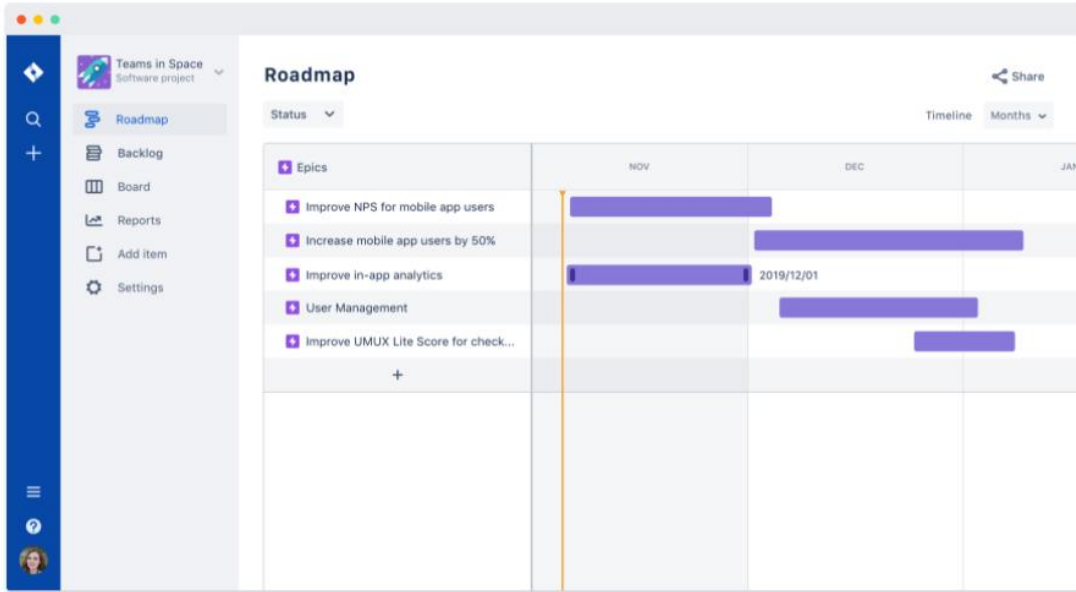


Ilustración 8 Hojas de ruta (Jira | Software de Seguimiento de Proyectos e Incidencias, n.d.)

En la Ilustración 8 se tiene una Hojas de ruta donde se puede plasmar un esbozo de tu idea completa, comunícales tus planes a las partes interesadas y asegúrate de que tu hoja de ruta se conecta con el trabajo de tu equipo: todo ello en unos pocos clics en Jira Software Cloud.

Informe

Mejora el rendimiento del equipo con datos visuales en tiempo real que tu equipo puede emplear.



Ilustración 9 Creación de informes ágiles (Jira | Software de Seguimiento de Proyectos e Incidencias, n.d.)

En la ilustración 9 se observan los tipos de informes ágiles que tiene la herramienta, donde los equipos tienen acceso a montones de informes incorporados de serie que recopilan datos prácticos y funcionales en tiempo real sobre el rendimiento del equipo sprint a sprint.

Elige un flujo de trabajo o crea uno

Cada equipo cuenta un proceso único para lanzar software. Utiliza un workflow predefinido o crea uno adaptado a la forma de trabajar de tu equipo.

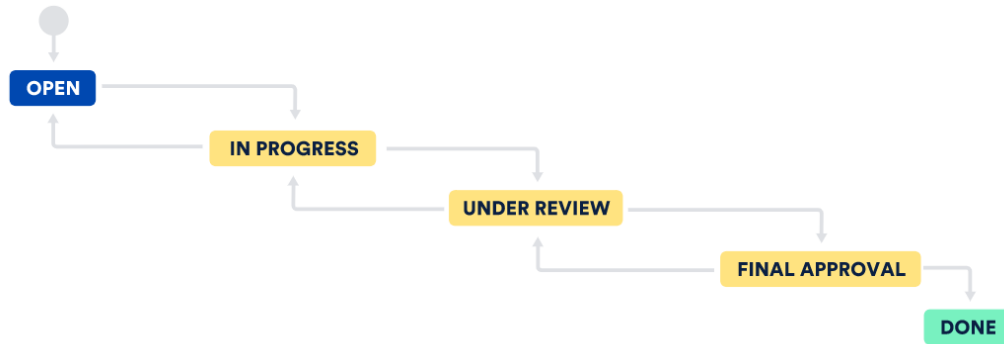


Ilustración 10 Flujo de Trabajo (Jira | Software de Seguimiento de Proyectos e Incidencias, n.d.)

Se integra con las herramientas que ya usas

Mejora tu flujo de trabajo con Confluence, Bitbucket y otros cientos de herramientas para desarrolladores.

Conecta el trabajo de tu equipo con la hoja de ruta de tu producto

Lanza productos de forma más rápida y fiable creando planes más inteligentes para tu equipo y tu organización.

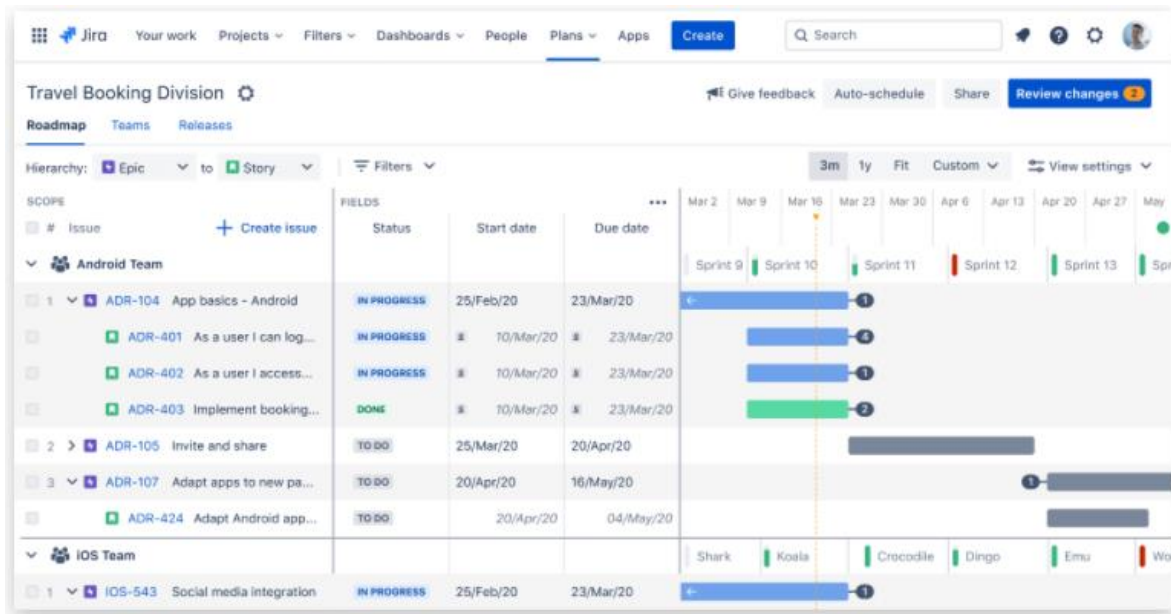


Ilustración 11 Hoja de ruta de producto (Jira | Software de Seguimiento de Proyectos e Incidencias, n.d.)

Mantenimiento de Sistemas

Identificando la considerable velocidad con que se ha desarrollado el hardware, el desarrollo del software ha sufrido un retraso histórico en cuanto a la elaboración y disposición de metodologías y herramientas, así como de modelos o teorías. Una de las principales causas de esta situación ha sido la poca importancia que se le ha dado al mantenimiento y a la evolución del software desde la Gestión de empresas, y de los responsables de procesos informáticos y usuarios. (Uaa - *Mantenimiento y Evolución de Sistemas de Información*, n.d.)

(Uaa - *Mantenimiento y Evolución de Sistemas de Información*, n.d.)

Conceptos Generarles

Se introduce una definición de mantenimiento y los cuatro tipos de mantenimiento que se puede identificar: correctivo, adaptativo, perfectivo y preventivo.

Definición de mantenimiento: El estándar IEEE 1219 [IEEE, 1998] define el Mantenimiento del Software como «la modificación de un producto software después de haber sido entregado (a los usuarios o clientes) con el fin de corregir defectos, mejorar el rendimiento u otros atributos, o adaptarlo a un cambio en el entorno».

En el estándar ISO/IEC 12207, de Procesos del Ciclo de Vida del Software [ISO/ IEC, 2017] se establece que el propósito del proceso de mantenimiento es conservar la capacidad del sistema de proporcionar un servicio. En el libro de la Ingeniería de software (Pressman R.S, 2013)el autor menciona que “la fase de mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software, y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente”. En las anteriores definiciones de mantenimiento aparecen indicados, directa o indirectamente, cuatro tipos de mantenimiento: correctivo, adaptativo, perfectivo y preventivo, que suelen ser los referenciados (1219-1998 - 1219-1998 - *Estándar IEEE Para Mantenimiento de Software - Estándar IEEE*, n.d.)

Necesidades de Mantenimiento

Las necesidades de mantenimiento se dan para garantizar que el software cumpla satisfactoriamente con los requerimientos solicitados y se aplica a cualquier desarrollo independiente del modelo de ciclo de vida utilizado. El mantenimiento deber ser realizado con el fin de Corregir fallas, Mejorar el diseño, Implementar mejoras, Definir interfaces con otros sistemas. Adaptar programas a diferentes tipos

de hardware, software, características del sistema y capacidad de las redes de telecomunicaciones. Migrar software heredado, incluso quitar el propio sistema implica una situación de mantenimiento.

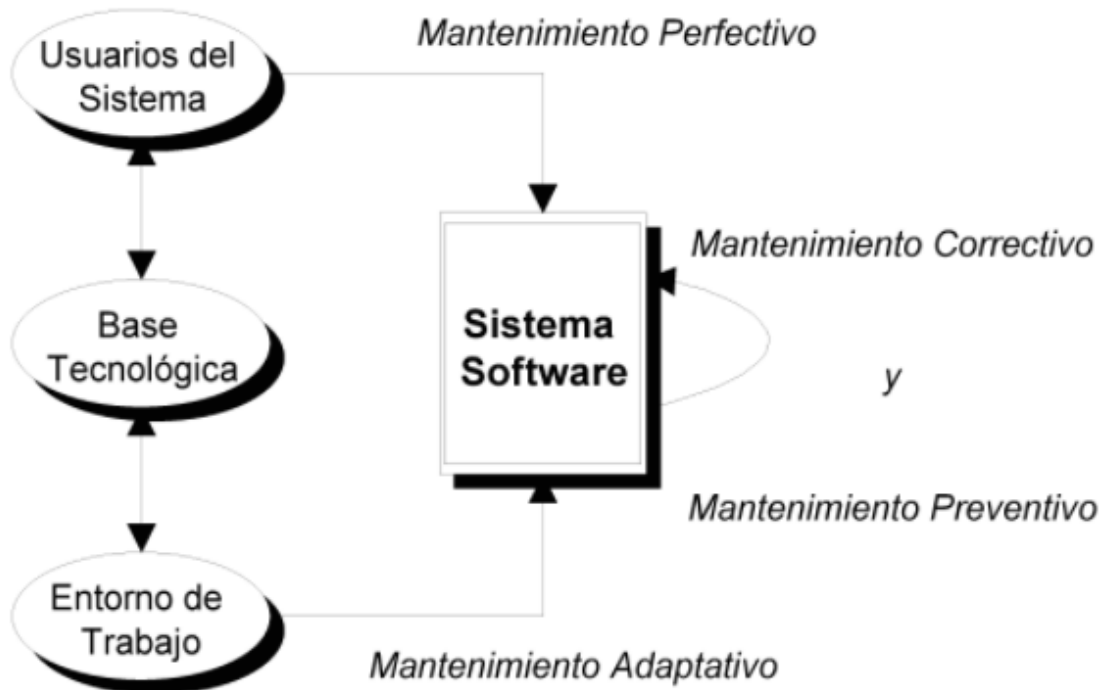


Ilustración 12 Las fuentes del mantenimiento del software (Uaa - Mantenimiento y Evolución de Sistemas de Información, n.d.)

En la ilustración 12 puede observarse un resumen del papel que representa cada tipo de mantenimiento, también se muestra que mientras que el cambio tecnológico afecta indirectamente a los sistemas software, el entorno de trabajo y los usuarios lo hacen directamente, generando peticiones de mantenimiento adaptativo y perfecto respectivamente. (Uaa - Mantenimiento y Evolución de Sistemas de Información, n.d.)

Mantenimiento correctivo: A pesar de las pruebas y verificaciones que aparecen en etapas anteriores del ciclo de vida del software, los programas pueden tener defectos. El mantenimiento correctivo tiene por objetivo localizar y eliminar los posibles defectos de los programas. Un defecto es una característica del sistema con el potencial de causar un fallo. Un fallo ocurre cuando el comportamiento de un sistema es diferente del establecido en la especificación. Entre otros, los fallos en el software pueden ser de: Procesamiento, por ejemplo, salidas incorrectas de un programa. Rendimiento, por ejemplo, tiempo de respuesta demasiado alto en una

búsqueda de información. Programación, por ejemplo, inconsistencias en el diseño de un programa. Documentación, por ejemplo, inconsistencias entre la funcionalidad de un programa y el manual de usuario.

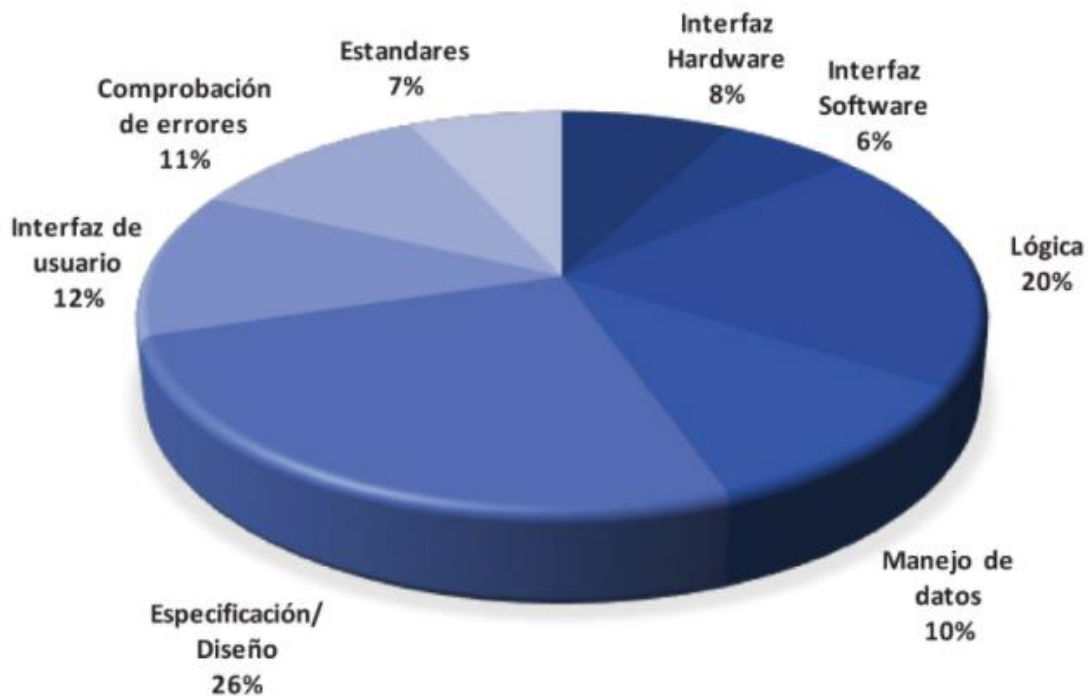


Ilustración 13 Origen de los defectos del software (Uaa - Mantenimiento y Evolución de Sistemas de Información, n.d.)

En la Ilustración 13 se muestra una distribución de las causas de los defectos en un estudio clásico, según el cual el 37,4% de los defectos se origina en la fase de especificación de requisitos, el 25,5% en la fase de diseño y el 36,3% en la fase de codificación.

Mantenimiento adaptativo: Este tipo de mantenimiento consiste en la modificación de un programa debido a cambios en el entorno (hardware o software) en el cual se ejecuta. Estos cambios pueden afectar al sistema operativo (cambio a uno más moderno), a la arquitectura física del sistema informático (paso de una arquitectura de red de área local a Internet/Intranet o a la nube) o al entorno de desarrollo del software (incorporación de nuevos elementos o herramientas). La envergadura del cambio necesario puede ser muy diferente: desde un pequeño retoque en la estructura de un módulo hasta tener que reescribir prácticamente todo el programa para su ejecución en un ambiente distribuido en una red. Los cambios en el entorno software pueden ser de dos clases: En el entorno de los datos, por ejemplo, al dejar de trabajar con un sistema de gestión de bases de datos relacionales y sustituirlo por una base de datos NoSQL. En el entorno de los procesos, por ejemplo, migrando a una nueva plataforma de desarrollo con

componentes distribuidos, Java, ActiveX, etc. El mantenimiento adaptativo es cada vez más usual debido principalmente al cambio, cada vez más rápido, en los diversos aspectos de la informática: nuevas generaciones de hardware, nuevos sistemas operativos o versiones de los antiguos, y mejoras en los periféricos o en otros elementos del sistema. Frente a esto, la vida útil de un sistema software puede superar fácilmente los veinte años.

Mantenimiento perfectivo: Cambios en la especificación, normalmente debidos a cambios en los requisitos de un producto software, implican un nuevo tipo de mantenimiento llamado perfectivo. La casuística es muy variada. Desde algo tan simple como cambiar el formato de impresión de un informe, hasta la incorporación de un nuevo módulo aplicativo. Podemos definir el mantenimiento perfectivo como el conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el usuario.

Este tipo de mantenimiento aumenta cuando un producto software tiene éxito comercial y es utilizado por muchos usuarios, ya que cuanto más se utiliza un software, más peticiones de los usuarios se reciben demandando nuevas funcionalidades o mejoras en las existentes.

Mantenimiento preventivo: Este último tipo de mantenimiento consiste en la modificación del software para mejorar sus propiedades (por ejemplo, aumentando su calidad y/o su mantenibilidad) sin alterar sus especificaciones funcionales. Por ejemplo, se puede reestructurar los programas para mejorar su legibilidad, o bien incluir nuevos comentarios que faciliten la posterior comprensión del programa. En algunos casos se ha planteado el Mantenimiento para la Reutilización, consistente en modificar el software (buscando y modificando componentes para incluirlos en bibliotecas) para que sea más fácilmente reutilizable. En realidad, este tipo de mantenimiento es preventivo, especializado en mejorar la propiedad de reusabilidad del software.

Se identifican algunas actividades que se realizan cuando se realizan cambios al software:

Análisis de impacto costo/beneficio: se busca analizar diferentes alternativas de implementación y/o a comprobar su impacto en la planificación, costo y facilidad de operación.

Comprensión del cambio: identificar en donde se encuentra el error e identificar la causa raíz o en el entendimiento de la mejora solicitada.

Diseño del cambio: se refiere al diseño propuesto para el cambio, pudiéndose incluir un rediseño del sistema.

Codificación y pruebas unitarias: se codifica y prueba el funcionamiento de cada componente modificado.

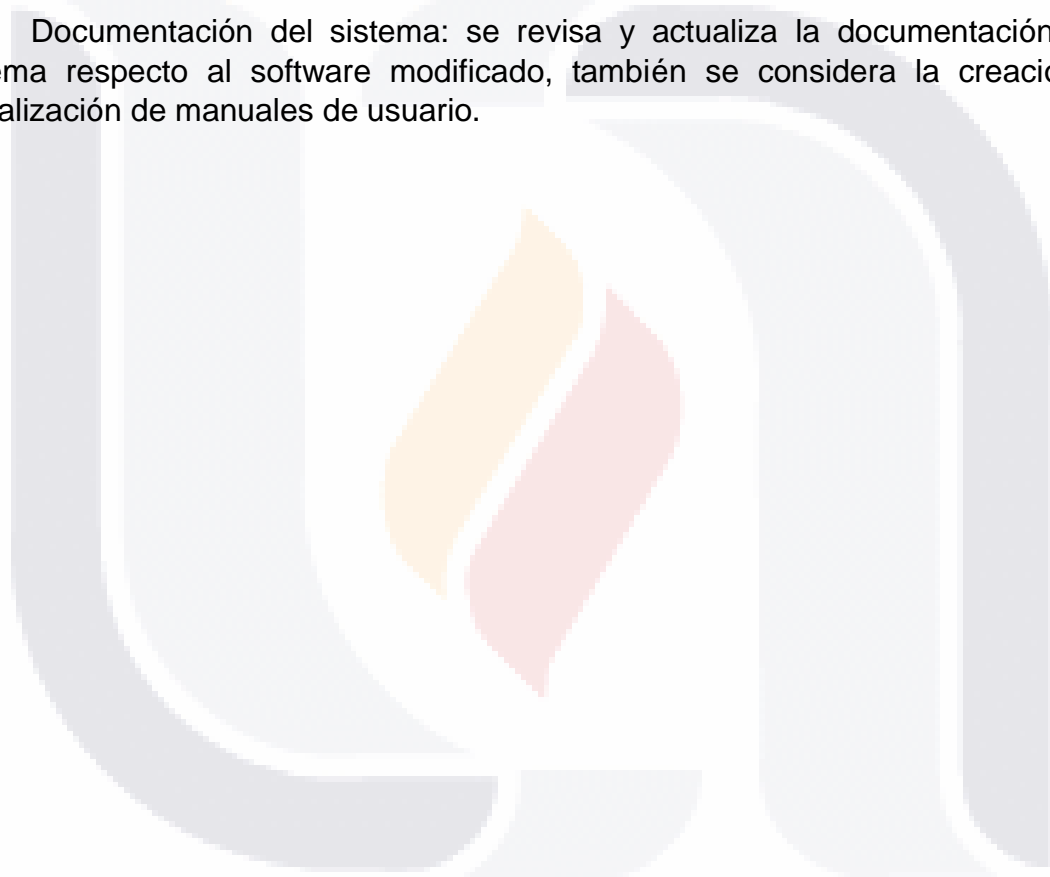
Inspección, certificación y consultoría: esta actividad se dedica a inspeccionar el cambio, comprobar otros diseños, reuniones de inspección, etc.

Pruebas de integración: se refiere a comprobar la integración de los componentes modificados con el resto del sistema.

Pruebas de aceptación: en esta actividad, el usuario comprueba, junto al personal encargado del mantenimiento, la adecuación del cambio a sus necesidades.

Pruebas de regresión: en esta actividad se somete el software modificado a casos de pruebas previamente almacenados y por los que ya pasó.

Documentación del sistema: se revisa y actualiza la documentación del sistema respecto al software modificado, también se considera la creación o actualización de manuales de usuario.



Agilidad

La gestión ágil surgió como contraste a un cierto modelo, llamado la gestión de proyectos predictiva. Ambas tienen sus virtudes y resultan más factibles en ciertas industrias. La predictiva se centra en planificar, en calcular un presupuesto y marcar plazos de entrega. Si el proyecto final se termina en la fecha acordada, sin exceder el costo y con todas las funcionalidades del plan inicial, se considera un éxito. La gestión predictiva es fruto de la Revolución Industrial: viene del mundo de la construcción, de la automovilística, de las fábricas. Por ejemplo: si lo que el cliente busca es una casa, ésta tendrá que construirse de forma que sea sólida, segura, cumpla con las necesidades de sus habitantes; y, en un escenario ideal, dentro del plazo previsto y sin exceder el costo.

Procesos y gestión predictiva

La crisis del software es un término informático acuñado en 1968, en la primera conferencia organizada por la OTAN (Organización del Tratado del Atlántico Norte), sobre desarrollo de software, en la que se analizaron los problemas de la programación. Se puso de manifiesto la necesidad de crear una disciplina científica que permitiera aplicar un enfoque sistemático y cuantificable al desarrollo, operación y mantenimiento de sistemas informáticos. Esto se llevó al intento de aplicar ingeniería de procesos al software, surgiendo así la Ingeniería del Software. Esta primera estrategia se basó en dos cimientos:

- Ingeniería de procesos: en los entornos de producción industrial existía un principio básico de calidad, contrastado con éxito: “la calidad del resultado depende de la calidad de los procesos empleados”. Dicho de otra forma: no se necesita a gente muy preparada; mientras los procesos utilizados sean de calidad, el resultado final será de calidad.
- Gestión predictiva: un tipo de gestión que se centra en garantizar el cumplimiento de agendas y presupuestos. Mientras la disciplina evolucionaba y se perfeccionaba a través de diferentes modelos de procesos y cuerpos de conocimiento para gestión de proyectos (MILQ9858, ISO9000, ISO9000-3, ISO 12207, SPICE, SW-CMM...) en la industria del software surgían dudas y se cuestionaba esta estrategia. Desde mediados de los 90 hasta 2010 han sido habituales las posturas radicales entre los defensores de los modelos de procesos y de los marcos ágiles. En la práctica puede verse que a veces los criterios del éxito no son siempre el cumplimiento de funcionalidad, costos y fechas. Por otro lado, también se cuestiona si en el desarrollo de software, como en otros trabajos basados en el conocimiento, se puede producir con patrones de procesos industriales. Se empieza a aceptar que el conocimiento tácito de la persona que realiza el trabajo puede

aportar más al valor del resultado que la tecnología y los procesos empleados. (Marta, n.d.)

Metodología ágil

Las metodologías de desarrollo ágil se basan principalmente en la colaboración con los usuarios de software durante todo el proceso de desarrollo, la facilidad para adaptar el producto a cambios en requerimientos y en la entrega del producto. Basadas en el Manifiesto Ágil, las metodologías de desarrollo ágil han sido aceptadas y son utilizadas con éxito en proyectos donde los requerimientos son inicialmente desconocidos y se van construyendo durante el proceso de desarrollo a partir de interacciones con los usuarios y de la retroalimentación obtenida a partir de las mismas

El manifiesto ágil

De acuerdo con lo que se ha mencionado anteriormente, sobre la gestión predictiva, se puede analizar que se está poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan. Con esta actividad se ha llegado a valorar:

- A los individuos y su interacción, por encima de los procesos y las herramientas.
- La colaboración con el cliente, por encima de la negociación contractual.
- El software que funciona, por encima de la documentación absoluta.
- La respuesta al cambio, por encima del seguimiento de un plan. Aunque hay valor en los elementos de la derecha, valoramos más los de la izquierda.

En febrero de 2001, 17 profesionales del software fueron convocados por Kent Beck, que había publicado un par de años antes el libro en el que explicaba la nueva metodología Extreme Programming (*Extreme Programming Explained: Embrace Change - Kent Beck - Google Libros*, n.d.). Todos ellos tenían algo en común: eran críticos de los modelos de producción basados en procesos. Se reunieron en Salt Lake City para discutir sobre los procesos empleados por los equipos de programación. En la reunión se acuñó el término “métodos ágiles” para definir a aquellos que estaban surgiendo como alternativa a las metodologías formales, tales como CMMSW (precursor de CMMI), PMI, SPICE (proyecto inicial de ISO 15504, a su vez precursor de ISO 33000), a los que consideraban excesivamente pesados y rígidos por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo. Los integrantes de la reunión resumieron en cuatro postulados lo que ha quedado denominado como Manifiesto Ágil, que son los valores sobre los que se asientan estos métodos. Son los que

abren y se desarrollan en este apartado. También establecieron 12 principios. Este conjunto de valores y principios permiten el desarrollo de software de calidad, de manera rápida, adaptable a los requisitos cambiantes del usuario

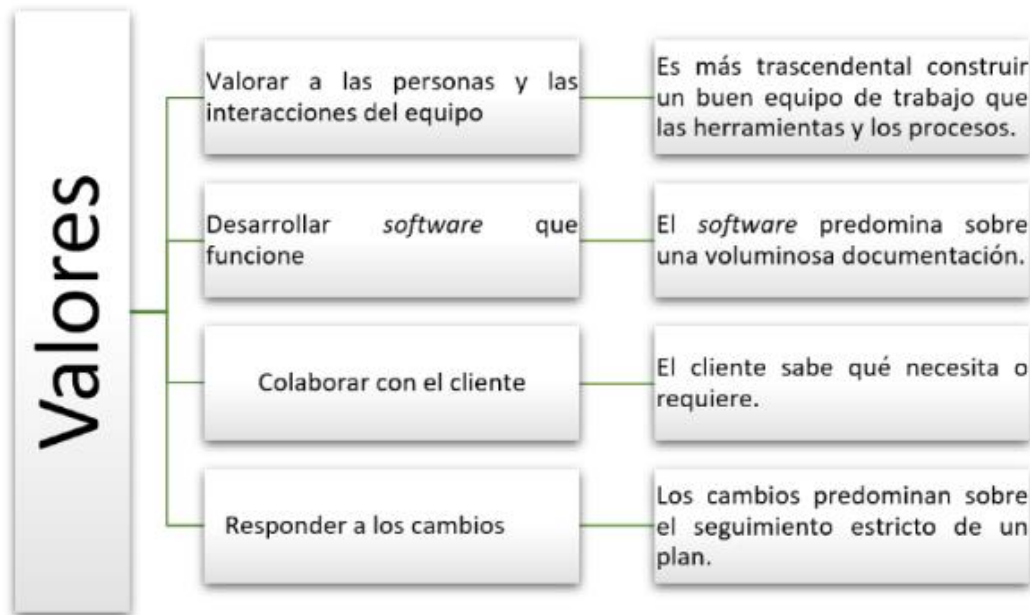


Ilustración 14 Valores del manifiesto ágil (Hernández Bejarno & Baquero Rey, n.d.)

En la ilustración 14 se mencionan los 4 valores el manifiesto ágil, donde el primero es el más importante, ya que las tareas requieren el talento y la motivación de las personas. El segundo valor menciona que la documentación se debe reducir al mínimo indispensable, es decir, eliminar toda aquella que consume trabajo sin aportar un valor directo al producto. El tercer punto habla de que es más eficiente tomar feedback directo a la vez que se desarrolla el producto, y en consecuencia redefinir y mejorar los requisitos de las partes que quedan, para que el cliente sea consciente de los cambios conforme van surgiendo, debe acompañar al equipo durante el proceso. Y el último valor, significa que tiene más importancia la repuesta al cambio, que el seguimiento a un plan, esto quiere decir, que, para desarrollar productos de requisitos inestables, en los que es inherente el cambio y la evolución rápida y continua, resulta mucho más valiosa la capacidad de respuesta que la de seguimiento y aseguramiento de planes.



Ilustración 15 Principios del Manifiesto ágil (Hernández Bejarno & Baquero Rey, n.d.)

En la ilustración 15 se tiene los 12 principios del manifiesto ágil, donde en el número uno se menciona que la principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor. En el número 2, son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles consideran el cambio como ventaja competitiva para el cliente. En el principio 3, tener entregas frecuentes de software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves. En el punto 4, las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto. En el principio 5, la construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea. El punto 6, donde la forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara. En el

principio 7, el software que funciona es la principal medida del progreso. En el principio 8, los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida. En el punto 9, se menciona que la atención continua a la excelencia técnica enaltece la agilidad. En el principio 10, la simplicidad como arte de maximizar la cantidad de trabajo que no se hace, es esencial. En el punto 11, las mejores arquitecturas, requisitos y diseños emergen de equipos que se autoorganizan. Y en el último principio, el número 12, se menciona que, en intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia. (Marta, n.d.)

Las técnicas ágiles y la madurez de la industria del software

Las metodologías ágiles están planteadas para afrontar el problema de los requerimientos inciertos o variables de las entregas, que tienen como objetivo satisfacer los requerimientos esenciales del cliente. Al ser utilizadas las primeras versiones de la aplicación de software, se comprende mejor el problema a solucionar y surgen nuevos ajustes a los requerimientos, realizados en entregas posteriores. La entrega continua de nuevas versiones incluye requerimientos determinados del momento, lo cual permite hacer frente a los cambios de última hora, pues se tiene la necesidad de una comunicación continua con el cliente, como parte interesada. La metodología ágil que se seleccione depende de cada proyecto, según sus necesidades particulares. Hay que tener en cuenta la experiencia y experticia de la casa desarrolladora de software, la capacidad y experiencia del equipo y la clase de proyecto a desarrollar para elegir un método específico. La cantidad de personal y otros recursos necesarios también puede llegar a ser un aspecto decisivo. Para los métodos ágiles, la base del éxito está en el equipo de trabajo y no en la planificación. Lo fundamental en las metodologías ágiles es el código fuente, adaptable en lugar de predictivo y orientado a los usuarios y no al proceso.

Área	Metodologías ágiles
Desarrolladores	Ágiles, capacitados, colaborativos y situados en el mismo lugar.
Clientes	El cliente es parte del equipo de desarrollo. No existe un contrato tradicional o es más flexible.
Requerimientos	Generalmente en vías de desarrollo; cambian rápidamente. Pequeños cambios en cada versión, fácilmente adaptable a los cambios.
Arquitectura	Diseñada para los requerimientos actuales. Menos énfasis en la arquitectura.
Diseño	Diseño y construcción integrados
Tamaño	Equipos y productos pequeños.
Objetivo Rapidez.	Rapidez.
Artefactos	Pocos artefactos.
Roles	Pocos roles.

Ilustración 16 Características generales en las metodologías ágiles (Hernández Bejarno & Baquero Rey, n.d.)

En la ilustración 16 se muestra una tabla que sugiere las características generales que debe tener un proyecto basado en la filosofía de metodologías ágiles.

Scrum

El modelo de desarrollo Scrum se caracteriza por definir roles y prácticas que aportan a la ejecución del proyecto un alto nivel de dinamismo, a través de la creación de equipos autoorganizados y una comunicación fluida, lo cual permite un ágil desarrollo de software (*Una Guía Para El CUERPO DE CONOCIMIENTO DE SCRUM (Guía SBOK™) Una Guía Completa Para La Entrega de Proyectos Utilizando Scrum*, 2016). Uno de los fundamentos de este framework es el ciclo de vida iterativo e incremental en el que se va liberando el producto por partes, donde cada entrega es un incremento a la funcionalidad respecto a la anterior.

La base principal de Scrum es el sprint, que corresponde a un lapso, durante el cual se realiza un avance del producto. En los sprints se realizan una serie de reuniones para la planificación del sprint, Scrum diarios, revisión del sprint, retrospectiva del sprint.

La implementación en el sprint corresponde a la funcionalidad del prototipo que proviene de la lista de producto o “product backlog”, la cual contiene un conjunto de requisitos funcionales o historias de usuario. (Hernández Bejarno & Baquero Rey, n.d.)

Los roles fundamentales de Scrum son:

- Scrum master: es el líder del equipo y su rol fundamental es velar por la integración del equipo, mediante la creación de un ambiente adecuado para completar satisfactoriamente el desarrollo del producto. El Scrum master guía, facilita el espacio, elimina los obstáculos que enfrenta el equipo y verifica que se estén siguiendo los procesos de Scrum.
- ProductOwner: es la persona responsable de integrar los requisitos del cliente o interesados, ya sean internos o externos, estableciendo una comunicación clara con el equipo Scrum sobre el producto, los requisitos funcionales y los requerimientos no funcionales, para que este, a su vez, realice entregas de valor.
- Equipo Scrum: son todos aquellos que entienden la especificación de los requerimientos del negocio y participan en el desarrollo del software. Está integrado por programadores, personal de calidad, entre otros.

En cuanto a las herramientas de Scrum, en primer lugar, se encuentra la lista de requisitos priorizada (product backlog), que constituye las perspectivas del cliente en razón a los objetivos y entregas del producto o proyecto. El cliente es el responsable de establecer y gestionar esta lista, con el apoyo del Scrum máster y del equipo, quien brinda el costo estimado para cumplir cada requisito. Al inicio de la primera iteración, el cliente debe haber definido el objetivo del producto o proyecto y la lista de requisitos. Otra herramienta es la lista de tareas de la iteración (sprint backlog), para lo cual el equipo elabora un plan que permita completar los requisitos establecidos para la iteración y se obliga a demostrar al cliente que, al finalizar la iteración, se hará entrega del producto con un incremento al proyecto acordado. Esta lista se puede establecer cuando el equipo está teniendo problemas y no avanza, con el fin de tomar decisiones al respecto. En la gráfica llamada "burndown chart" se puede visualizar el trabajo pendiente a lo largo del tiempo, la velocidad con que se están llevando a cabo los requisitos y el cumplimiento de las tareas.

Metodología XP

La metodología de Programación Extrema (XP) se fundamenta en valores como la sencillez, la comunicación, la retroalimentación y el coraje. Sus principios se basan en la simplicidad, los cambios incrementales, la calidad y el feedback. Sus objetivos están centrados en responder a las necesidades de los clientes y comprometer a todo el equipo, mediante una serie de roles y fases.

Los roles existentes en esta metodología son programador, cliente, encargado de pruebas (tester), encargado de seguimiento (tracker), entrenador (coach), consultor y gestor (the big bos). La metodología incluye las siguientes fases:

- Fase de exploración: especifica el alcance general del proyecto, el cliente define lo que necesita mediante la redacción de historias de usuario y los

programadores evalúan los tiempos de desarrollo con base en esta información. Las estimaciones pueden variar cuando se analicen con más detalle en cada iteración. Esta fase comprende un par de semanas y el resultado es una visión general del sistema y un plazo total estimado.

- Fase de planificación: es un escenario de corta duración, donde el cliente, los directivos y el grupo de desarrolladores establecen el orden en que deben implementarse las historias de usuario, agrupándolas por entregas. Se realizan una serie de reuniones de planificación.
- Fase de iteraciones: es la de mayor importancia en el ciclo de desarrollo de XP, pues es el momento en el que se elaboran las funcionalidades. Al final de cada una se cuenta con un entregable funcional que implementa las historias de usuario asignadas a la iteración. Las historias de usuario requieren que el cliente participe activamente durante esta fase del ciclo, con el fin de realizar un análisis y su correspondiente desarrollo en esta iteración.
- Fase de puesta en producción: al finalizar cada iteración se realiza la entrega de módulos funcionales y sin errores. No se hacen desarrollos funcionales adicionales, pero es posible hacer ajustes.

Kanban

Esta técnica se originó en Toyota para controlar el desarrollo del trabajo en el contexto de una línea de producción. Actualmente se utiliza en la gestión de proyectos de software, al igual que Scrum, gestiona el desarrollo incremental, las historias de usuario, las tareas, etc. Esta metodología emplea un tablero denominado Kanban, que le permite a cada integrante del equipo obtener una visualización completa del estado de implementación del software. Que tiene como beneficios:

- Ser estímulo al rendimiento en pro de optimizar los procesos para obtener mejores resultados
- Mejora la organización y colaboración, al permitir el uso compartido del mismo tablero por parte de todo el equipo, lo que facilita participar en todas las tareas, distribuir el trabajo y comunicarse con los colegas.

Metodologías ágiles				
Características		XP	Scrum	Kanban
Uso	Cumplimiento de los requisitos.	Sí.	Sí.	Sí.
	Aumento de la productividad.	Sí.	Sí.	Sí.
	Iteraciones cortas.	Sí.	Sí.	Sí.
	Centrado en los usuarios.	Sí.	Sí.	Sí.
	Integración de cambios.	Sí.	Sí.	Sí.
	Los requerimientos funcionales pueden cambiar.	Sí.	Sí.	Sí.
	El plan de trabajo puede variar.	Sí.	NO.	Sí.
	Intercambio de conocimiento.	Alto.	Bajo.	Bajo.
	Tamaño de los proyectos.	Pequeño.	Grande/Pequeño.	Pequeño.
	Tamaño del equipo.	Pequeño.	Pequeño.	Pequeño.
	Grado de interacción en el equipo de trabajo.	Alto.	Alto.	Alto.
	Organización del equipo.	Autoorganizado.	Autoorganizado.	Autoorganizado.
	Definición de requisitos.	Sí.	Sí.	Sí.
	Modelado.	Sí.	Sí.	Sí.
	Codificación.	Sí.	Sí.	Sí.
	Pruebas unitarias.	Sí.	Sí.	Sí.
	Pruebas de integración.	Sí.	Sí.	Sí.
	Control de calidad.	No.	No.	

Ilustración 17 Características comunes en metodologías ágiles (Hernández Bejarno & Baquero Rey, n.d.)

En la ilustración 17 se tiene una tabla de las metodologías ágiles y su uso, de manera que sea más visual la comparación entre ellas, con la finalidad de que en cada proyecto se pueda utilizar la más conveniente de acuerdo a sus características.

Diseño de la intervención o aplicación de la metodología de la intervención

Descripción de la intervención

Qué se hizo.

Se realizó el registro de todos los requerimientos, se crearon los **sprints**, la planeación y asignación de los integrantes del equipo a cada requerimiento, se crearon flujos de trabajo. La supervisión de los requerimientos de manera que fueran priorizados y analizados. Se hicieron lanzamientos con la información actualizada y se generaron informes con el rendimiento del equipo.

Cómo se llevó a cabo

Se comenzó con la creación de un Proyecto que contiene los requerimientos de software. Se continuó con el registro de todos los requerimientos que inicialmente se encontraban en un archivo en Excel, se registró de cada requerimiento como una historia, de manera que se fue creando el Backlog.

Una vez que se tuvieron todos los requerimientos registrados, se creó el primer sprint donde se fueron asignando ciertos requerimientos, esto se consideraron de acuerdo con la priorización y estimación de tiempo y esfuerzo identificado para cada incidencia. Cuando se tenía la configuración completa del Sprint se daba inicio. Durante el curso del sprint, se fueron asignando los requerimientos en el tablero de “Por hacer”, “En curso” y “Listo”, al finalizar todos los requerimientos del sprint, se finalizaba y se generaba el siguiente sprint. Al contar con información de uno o varios sprints se pudo tener información para generar gráficas y presentar reportes de análisis.

Cuando se realizó

Del 9 de marzo al 28 de mayo del 2021

Que etapas comprendió

Registro: llenado de requerimientos en herramienta

Atención: asignación y planeación de cómo serían atendidos.

Seguimiento: gestionar el cumplimiento de los compromisos en tiempo y forma.

Cuáles fueron las actividades

1. Se comenzó con el registro de los requerimientos en la herramienta JIRA.

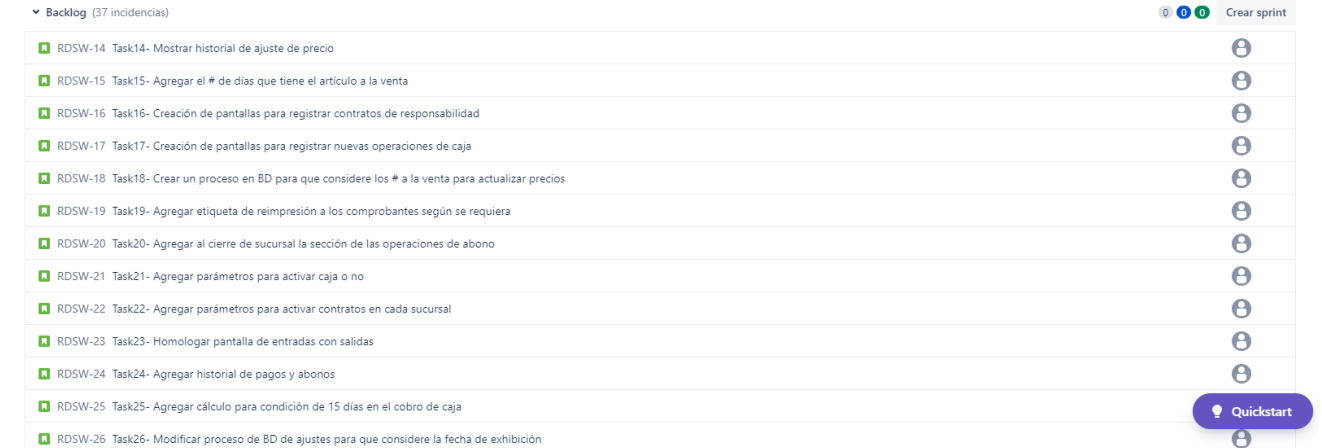


Ilustración 18 Lista de Requerimientos (Pedroza C)

En la Ilustración 18 se observa un listado de los requerimientos que fueron documentados en la herramienta JIRA para proceder a su atención.

2. Se crearon los sprints con fecha semanal, se asignaron los requerimientos en cada sprint, con las fechas definidas. La cantidad de requerimientos dependía de la estimación de tiempo que consideró el equipo

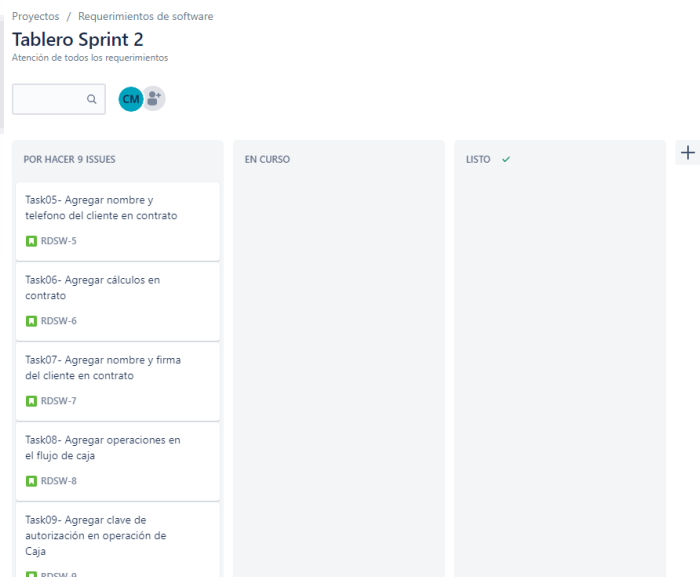


Ilustración 19 Sprint 2 iniciado (Pedroza C)

En la Ilustración 19 se observa cuando se ha iniciado el sprint 2 y cómo se muestran en el tablero “Por hacer” todos los requerimientos que se han considerado ser atendidos en el periodo de tiempo definido para el sprint 2.

3. Todos los días se generaron sesiones de 15 minutos para revisar el estatus de los requerimientos de cada sprint y donde se daba seguimiento a situaciones particulares que pudieran generar algún atraso. Se resolvían las situaciones y se continuaba hasta la finalización, durante este periodo del sprint cada requerimiento se fue moviendo de lugar según correspondiera, es decir, al inicio todos los requerimientos se encontraban en el tablero de “Por hacer”, una vez asignados a cada integrante del equipo, y listos para iniciarse, se cambiaban al tablero “En curso”, en esta sección se mantienen hasta que se concluyen las tareas para resolver la incidencia. Al finalizar los requerimientos se asignan a la sección “Listo”.



Ilustración 20 Tableros activos del Sprint 2 (Pedroza C)

En la ilustración 20 se observa cómo se han cambiado de posición los requerimientos de acuerdo con su estatus, es decir, para ese momento del sprint 2

se tienen 2 **issues** en la fase de “Listo” o terminados, 3 en la fase de “En curso” y 4 “Por hacer”.

Tablero Sprint 2

Atención de todos los requerimientos

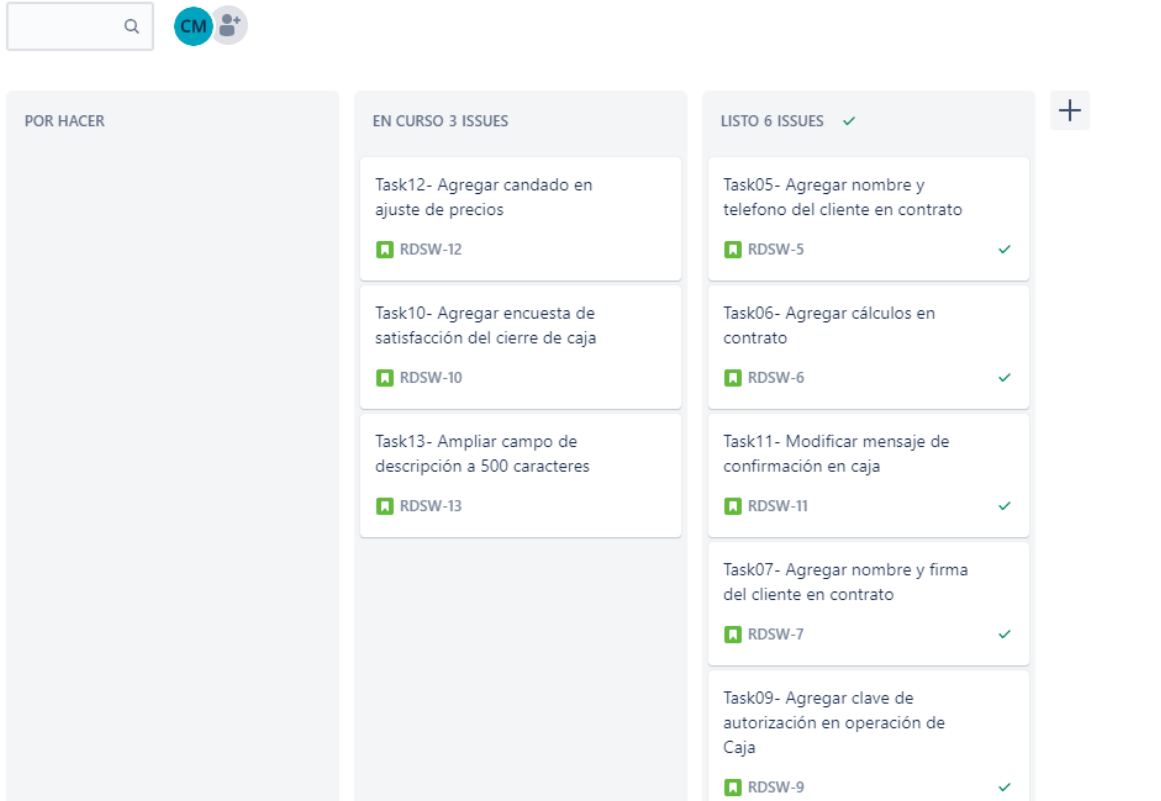


Ilustración 21 Tableros activos del Sprint 2 (Pedroza C)

En la ilustración 21 se puede observar que se continua con el sprint 2 y se han Avanzado con los requerimientos, de manera que se tiene 3 “En curso” y 6 “Listos”.

Tablero Sprint 2

Atención de todos los requerimientos

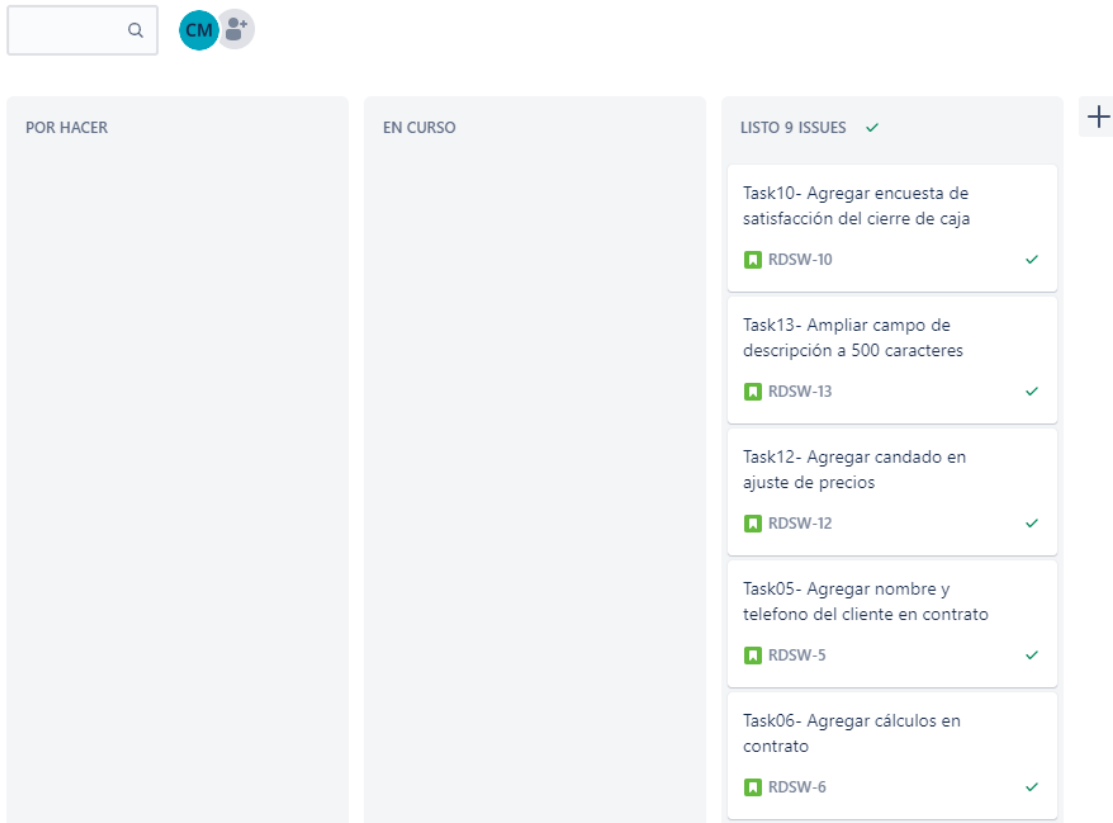


Ilustración 22 Sprint 2 por finalizar (Pedroza C)

En la ilustración 22 se observa que todos los requerimientos del sprint han sido concluidos, por lo que se encuentran en el tablero de “Listo”.

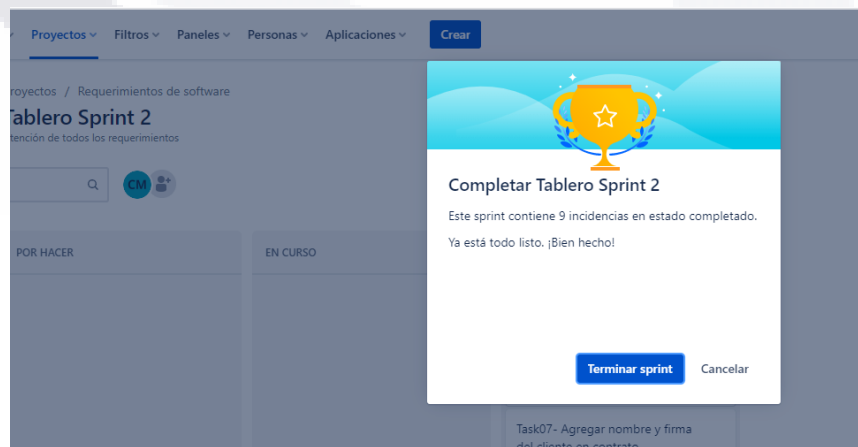


Ilustración 23 Finalización Sprint 2 (Pedroza C)

En la ilustración 23 se muestra la finalización del sprint2 donde se menciona que todos los **issues** fueron concluidos.

- Se presentó una situación en el 3er sprint ya que no fueron concluidos todos los requerimientos, por lo que se terminó el sprint y los tickets pendientes pasaron de Vuelta al backlog.

Tablero Sprint 3



Ilustración 24 Avance de Sprint 3 (Pedroza C)

En la ilustración 24 se observa el avance del sprint 3 donde se tienen 3 issues en “En curso” y 3 en “Listo”, lo que puede demostrar un buen avance en la atención de los requerimientos

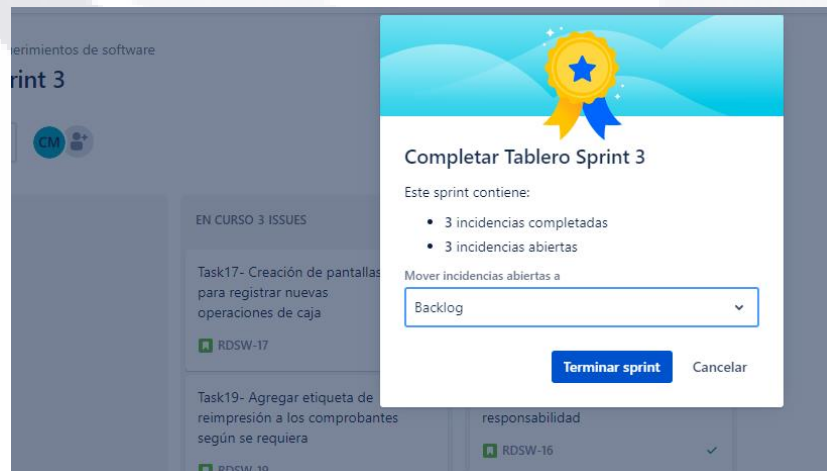


Ilustración 25 Finalización del Sprint 3 (Pedroza C)

En la ilustración 25 se muestra la pantalla de finalización del sprint 3, donde se menciona que quedaron 3 requerimientos “En curso”, es decir, que no fueron concluidos, por lo que se pregunta a dónde deben ser dirigidos para su seguimiento y atención, por lo que en esta situación se regresó a la lista del Backlog.

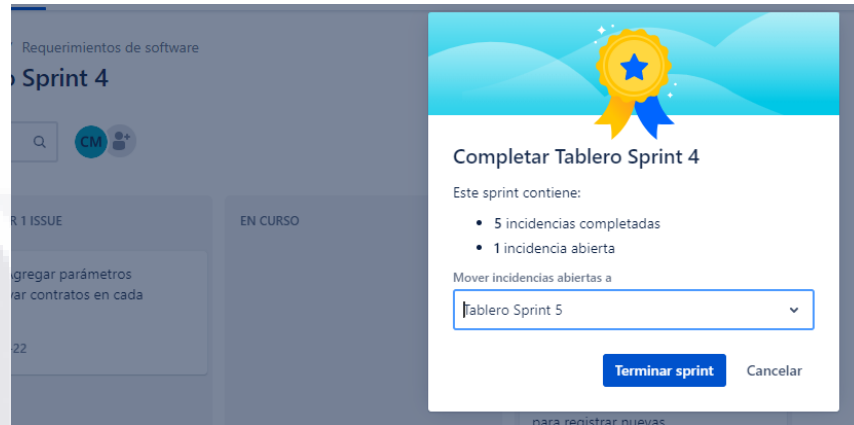


Ilustración 26 Finalización del Sprint 4 (Pedroza C)

En la ilustración 26 se muestra un caso similar al sprint 3, donde se quedó un issue pendiente, pero en este caso se asigna directamente al Sprint 5, que es el siguiente sprint en espera de iniciar.

Con qué recursos se contó

- Se contó con un equipo con los conocimientos en Scrum y JIRA
- Se utilizó Jira software gratuito
- Se contó con un líder preparado en JIRA y Scrum
- Se tuvo el apoyo del cliente con la definición de sus requerimientos

Cuáles son los beneficios esperados

Uno de los beneficios esperados es que la metodología de la herramienta JIRA sea de utilidad para la gestión de los requerimientos que se reciben y se atienden en la Gerencia de proyectos de una empresa.

Otro beneficio que se espera es que, al tener un proceso definido con el uso de la herramienta, se pueda dar una atención más detallada y controlada a los requerimientos de manera que se cumplan con las expectativas de los usuarios.

Se considera un beneficio esperado el conocer la herramienta y aprovechar todas las opciones que brinda para la gestión de requerimientos de software.

Resultado de la intervención

Descripción de la intervención

Qué se hizo.

Dentro del Corporativo Ortus, se llevó a Cabo la implementación del Caso práctico, en dicha empresa se presenta la situación de crear un Sistema donde se tenga el registro y asignación de operadores en las unidades de cada empresa y el seguimiento a las licencias de conducir. Por ello se realizó el registro de todos los requerimientos, se crearon los sprint's, se gestionó la planeación y asignación de requerimientos a los integrantes del equipo. La supervisión en la atención a los requerimientos se facilitó gracias a la lista de actividades y la transición de cada una en las etapas de "Por hacer", "En curso" y "Listo". Se hicieron lanzamientos con la información actualizada y se generaron informes con el rendimiento del equipo.

Cómo se llevó a cabo

Se comenzó con la creación de un Proyecto llamado "**Sistema y Dashboard para Ortus**" que contiene los requerimientos de software. Se continuó con el registro de todos los requerimientos que se identificaron inicialmente, como resultado de la primera sesión de entendimiento con el usuario. Con el listado de requerimientos se fue creando el Backlog.

Una vez que se tuvieron todos los requerimientos registrados, se creó el primer sprint donde se fueron asignando ciertos requerimientos, esto se consideraron de acuerdo con la priorización y estimación de tiempo y esfuerzo identificado para cada incidencia. Cuando se tenía la configuración completa del Sprint se daba inicio. Durante el curso del sprint, se fueron asignando los requerimientos en el tablero de "Por hacer", "En curso" y "Listo", al finalizar todos los requerimientos del sprint, se finalizaba y se generaba el siguiente sprint. Al contar con información de uno o varios sprints se pudo tener información para generar gráficas y presentar reportes de análisis.

Cuándo se realizó

Del 4 de octubre al 30 de diciembre de 2021

Que etapas comprendió

Registro: llenado de requerimientos en herramienta

Atención: asignación y planeación de cómo serían atendidos.

Seguimiento: gestionar el cumplimiento de los compromisos en tiempo y forma

Cuáles fueron las actividades

1. Se comenzó con el registro de los primeros requerimientos en la herramienta JIRA.

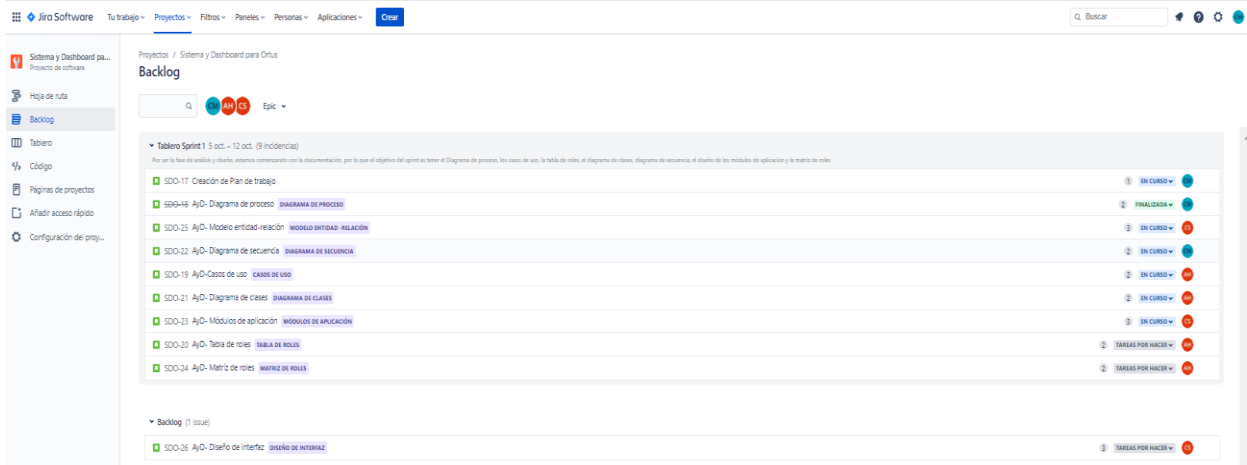


Ilustración 27 Lista de Requerimientos (Pedroza C)

En la Ilustración 27 se observa un listado de los requerimientos que fueron documentados en la herramienta JIRA para proceder a su atención.

Como parte de las actividades del llenado de requerimientos en la herramienta está la asignación a los integrantes del equipo, cuando se fue asignando cada actividad, se muestra en la herramienta y adicional se notifica por correo electrónico, lo que permite una mayor atención por parte del equipo para atender los requerimientos.

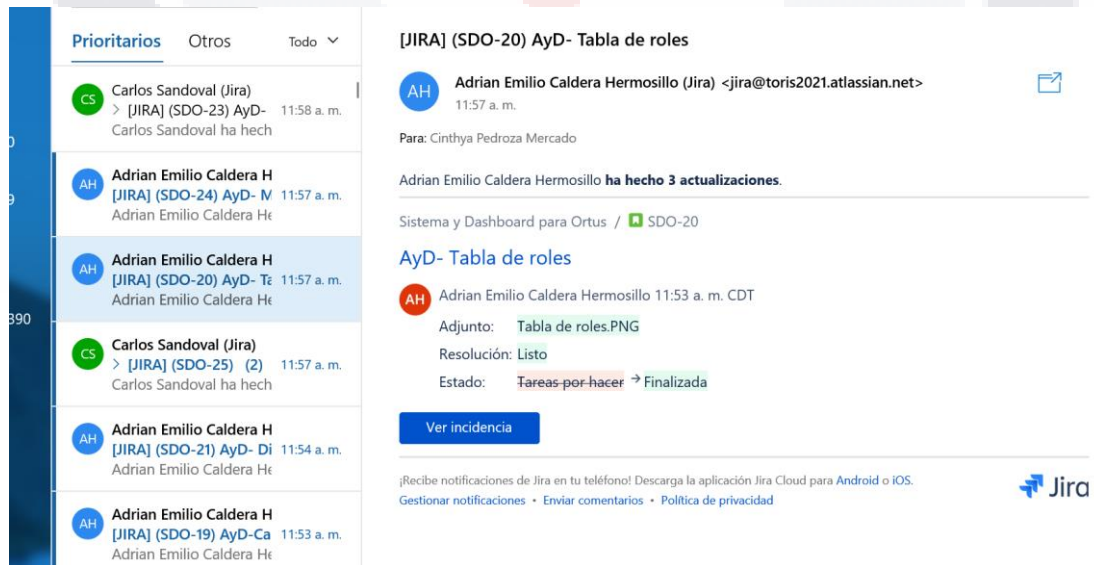


Ilustración 28 Correos de notificación (Pedroza C)

En la ilustración 28 se observa la serie de correos donde se notifica a cada usuario e integrante del equipo a quien se le asignan actividades, de manera que el seguimiento sea más preciso.

Se creó la hoja de ruta con todas las actividades y las fechas iniciales de entrega.

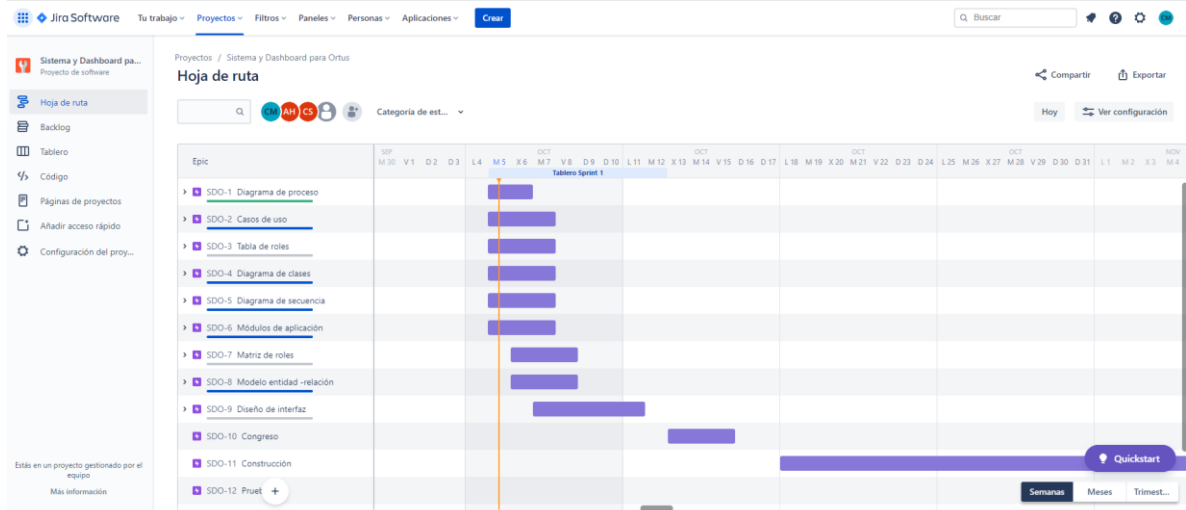


Ilustración 29 Hoja de ruta Inicial (Pedroza C)

En la Ilustración 29 se pueden apreciar las actividades y la fechas que se consideraron inicialmente para la entrega para cada requerimiento.

2. Se crearon los sprints con fecha semanal, se asignaron los requerimientos en cada sprint, con las fechas definidas. La cantidad de requerimientos dependía de la estimación de tiempo que consideró el equipo

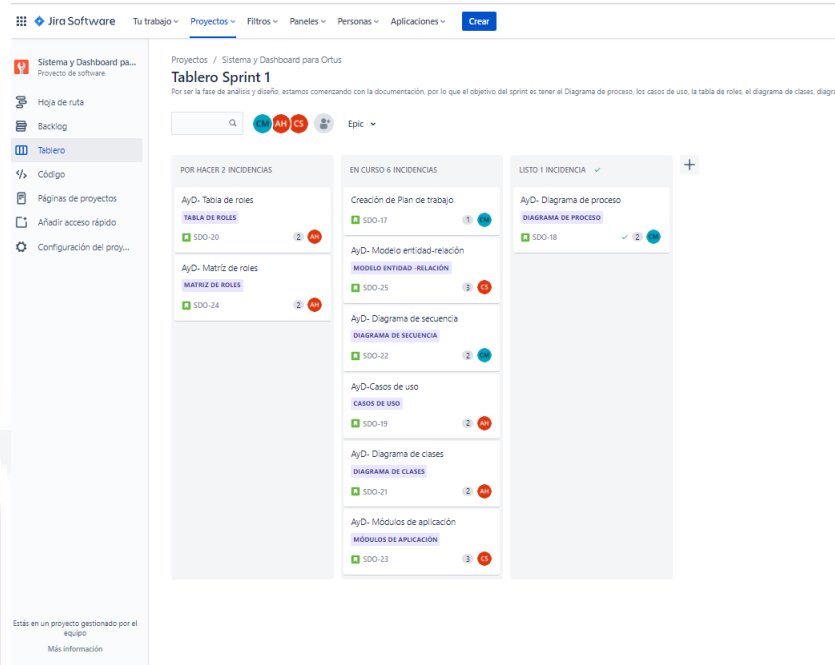


Ilustración 30 Sprint 1 Iniciado (Pedroza C)

En la Ilustración 30 se observa que ha iniciado el Sprint 1 y las fases en la que se encuentran los requerimientos.

3. Todos los días se generaron sesiones de 15 minutos con el usuario para revisar el estatus de los requerimientos de cada sprint y donde se daba seguimiento a situaciones particulares que pudieran generar algún atraso. Se resolvían las situaciones y se continuaba hasta la finalización, durante este periodo del sprint cada requerimiento se fue moviendo de lugar según correspondiera, es decir, al inicio todos los requerimientos se encontraban en el tablero de “Por hacer”, una vez asignados a cada integrante del equipo, y listos para iniciarse, se cambiaban al tablero “En curso”, en esta sección se mantienen hasta que se concluyen las tareas para resolver la incidencia. Al finalizar los requerimientos se asignan a la sección “Listo”.

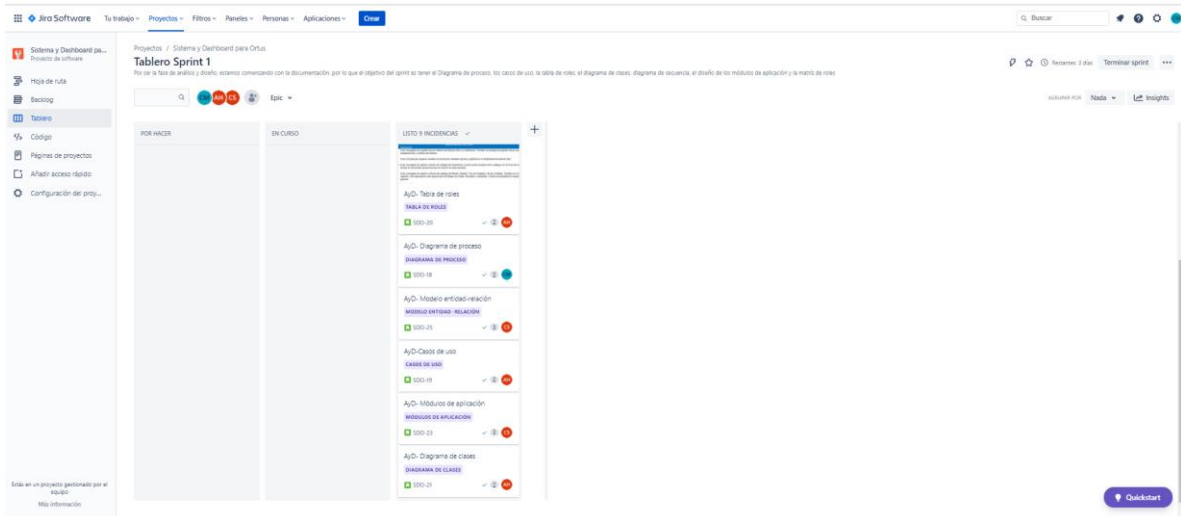


Ilustración 31 Tablero activo del Sprint 1 (Pedroza C)

En la ilustración 31 se observa cómo se han cambiado de posición los requerimientos de acuerdo con su estatus, es decir, para ese momento del sprint 1 se tienen todas las actividades en la fase de “Listo” o terminados. Debido a que las actividades del 1er sprint se habían terminado y todavía quedaba un periodo de tiempo disponible, se agregaron actividades nuevas.

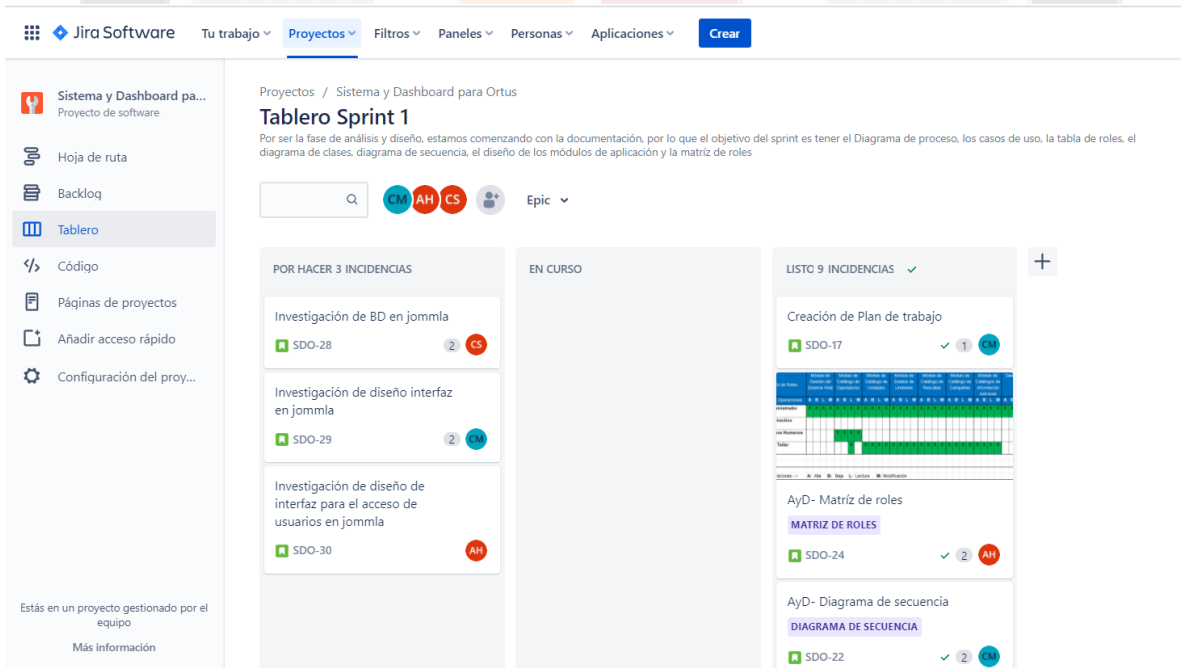


Ilustración 32 Sprint 1 con actividades nuevas (Pedroza C)

En la Ilustración 32 se puede apreciar que se concluyeron todas las actividades antes de que finalizara el sprint, por lo que se decidió agregar actividades que pudieran ser resueltas en el tiempo que quedaba del Sprint 1.

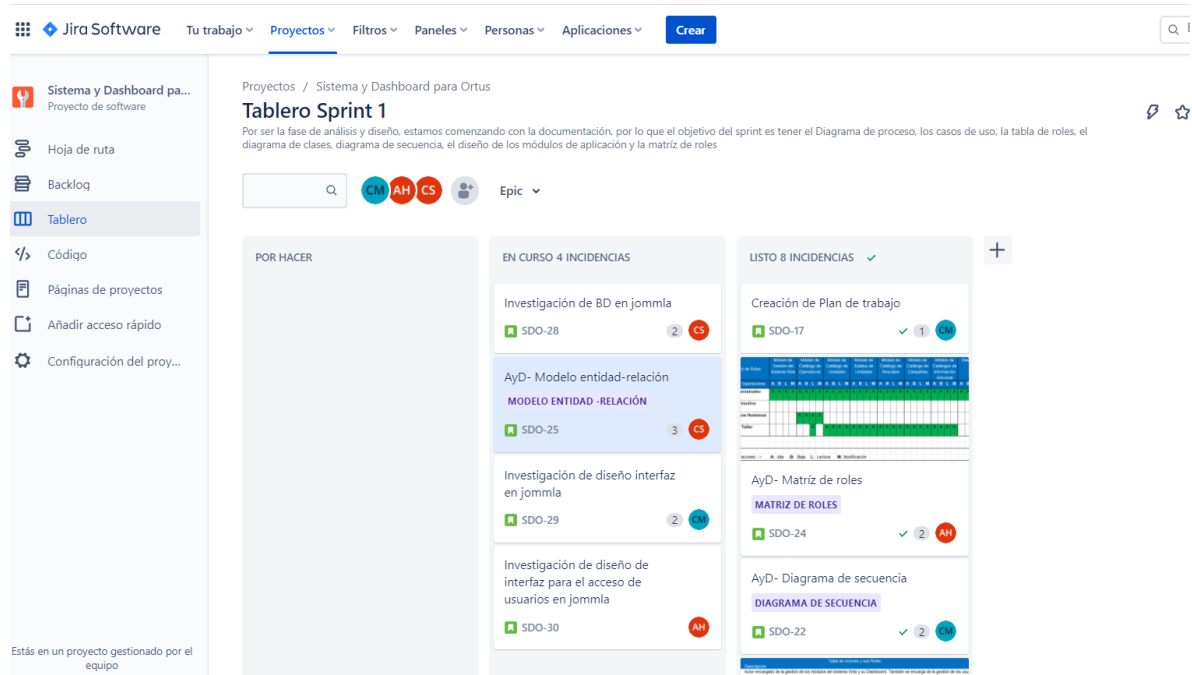


Ilustración 33 Sprint 1 con las actividades en curso (Pedroza C)

En la Ilustración 33 se observa que no todos los requerimientos del sprint han sido concluidos, por lo que se encuentran “En curso” y el resto en “Listo”.

El tiempo de primer sprint llegó a su fin y aún se tenían requerimientos en curso, por lo que se asignaron al siguiente Sprint.

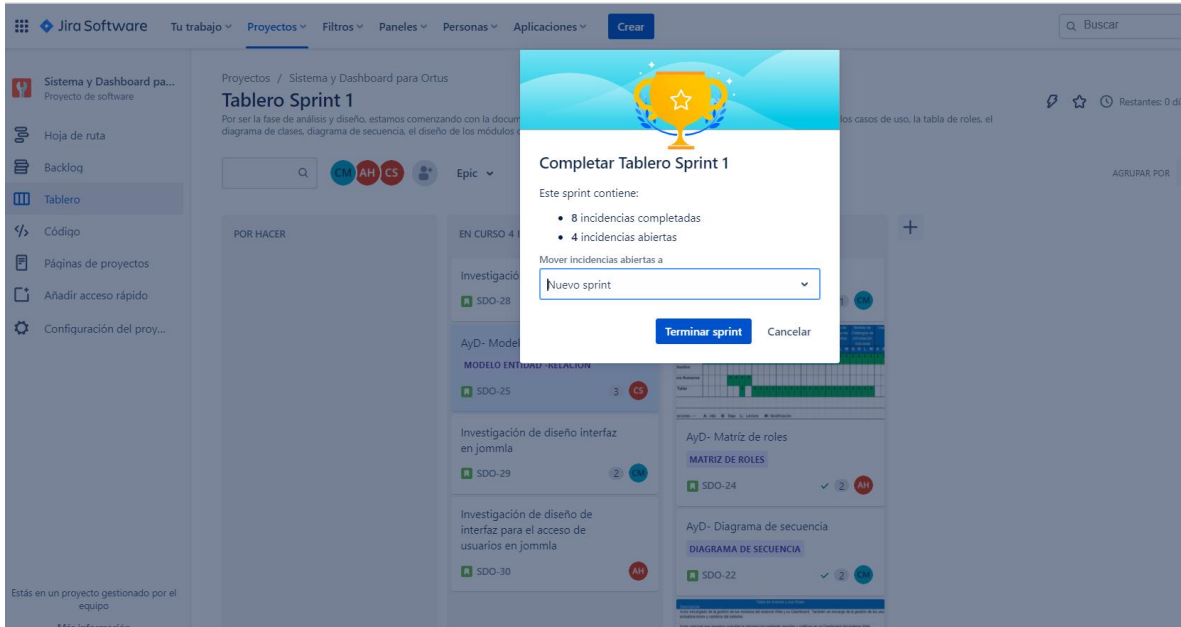


Ilustración 34 Finalización Sprint 1 (Pedroza C)

En la ilustración 34 se muestra la finalización del sprint1, se consideran las actividades finalizadas y las que quedan pendientes, mismas que serán asignadas al Sprint 2.

4. Se continuaron con las actividades del Sprint 2, donde se dejaron las actividades pendientes de sprint 1 y las que se consideraron adecuadas para este nuevo período de tiempo.

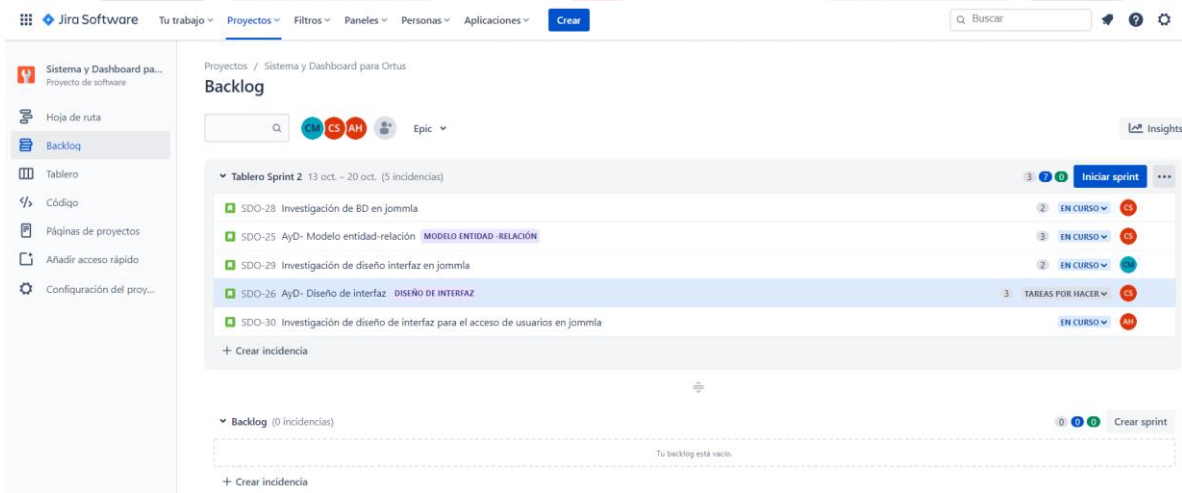


Ilustración 35 Tablero de Sprint 2 (Pedroza C)

En la ilustración 35 se observan los requerimientos del Sprint 2.

Durante los días que se asignaron al Sprint 2, se fue avanzando con los requerimientos hasta que quedaron 2 pendientes con estado en “En curso”.

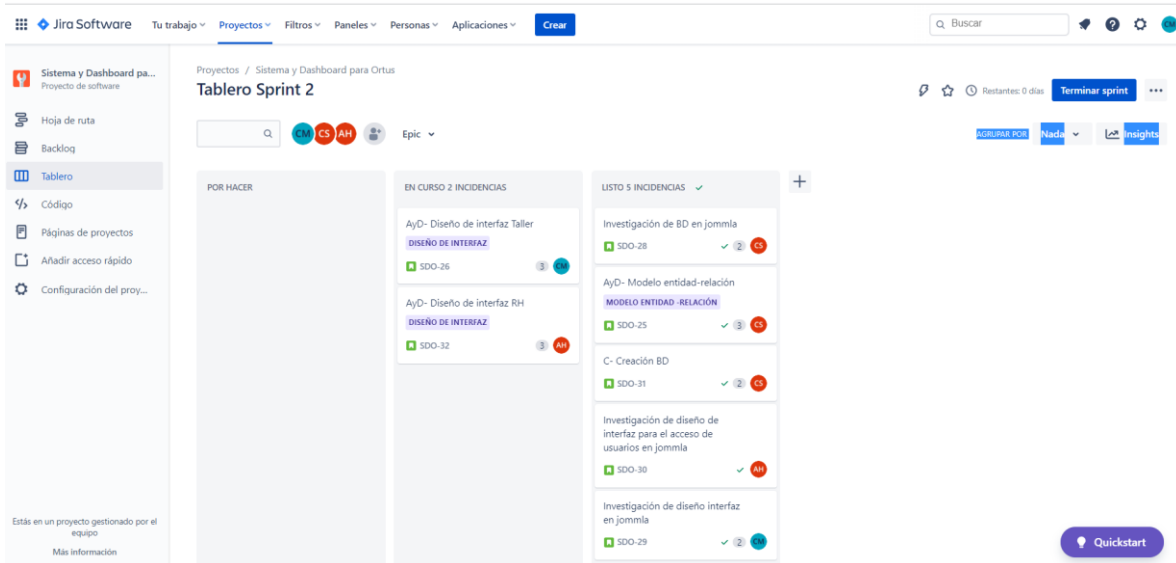


Ilustración 36 Tablero Sprint 2 (Pedroza C)

En la ilustración 36 se observa que la mayoría de los requerimientos se lograron atender, solo restan 2 que se quedan abiertas.

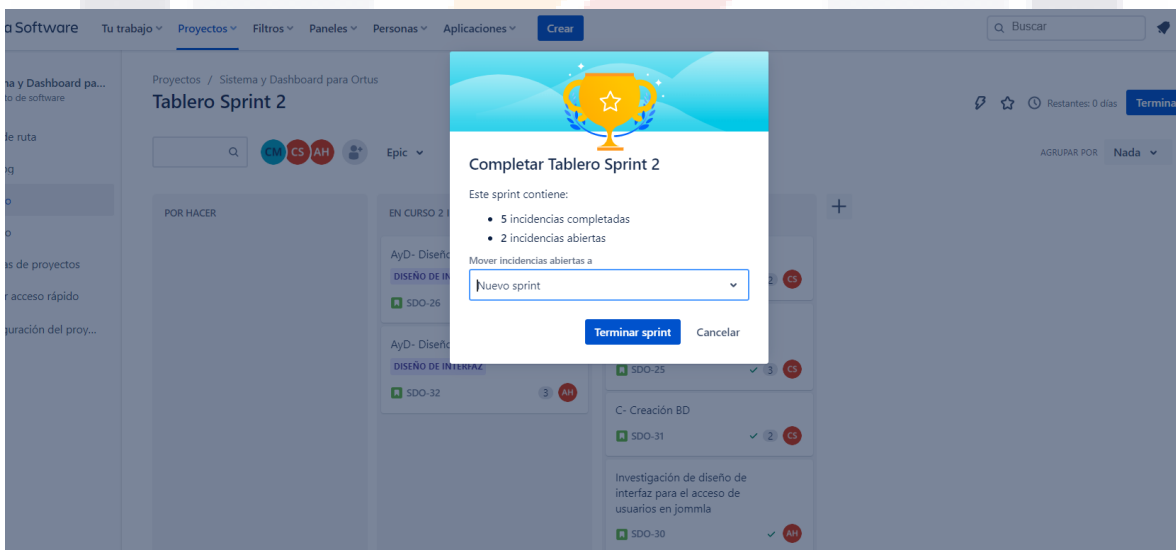


Ilustración 37 Finalización del Sprint 2 (Pedroza C)

En la ilustración 37 se muestra la pantalla de finalización del sprint 2, donde se menciona que quedaron 2 requerimientos “En curso”, es decir, que no fueron concluidos, se continua con la misma dinámica de agregarlos al siguiente Sprint, en este caso será al Sprint 3.

Se sigue trabajando con la misma dinámica de avanzar las actividades de un estatus a otro

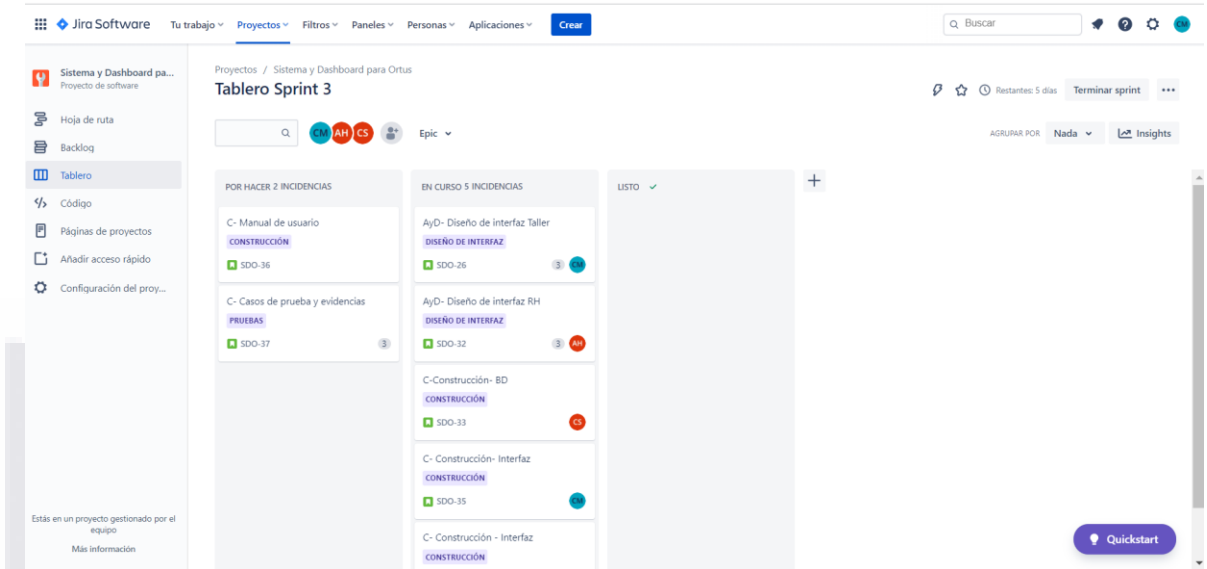


Ilustración 38 Tablero Sprint 3 (Pedroza C)

En la Ilustración 38 se observa el Sprint 3 con varios requerimientos en estatus de “En curso”.

Se trabajó en todas las actividades que se tenían, pero no fue posible concluir ninguna. Lo que lleva a identificar que las actividades están llevando más tiempo del estimado, esta situación es muy importante tenerla en cuenta, ya que son situación que se deben considerar para la próxima definición de un Sprint, es decir, que se evalúen las estimaciones anteriores identificando los tiempos más adecuados para cada requerimiento, de manera que puedan ser atendidos dentro de un Sprint.

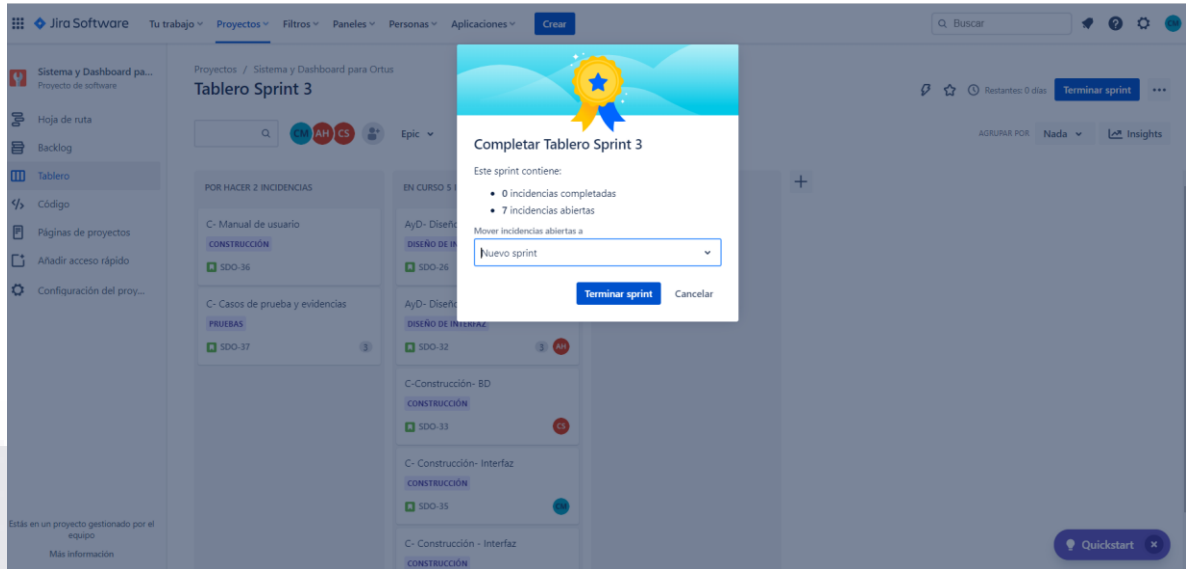


Ilustración 39 Finalización del Sprint 3 (Pedroza C)

En la Ilustración 39 se observa la gran cantidad de incidencias que se quedaron pendientes lo complicado que fue este Sprint, ya que no se finalizó ninguna actividad.

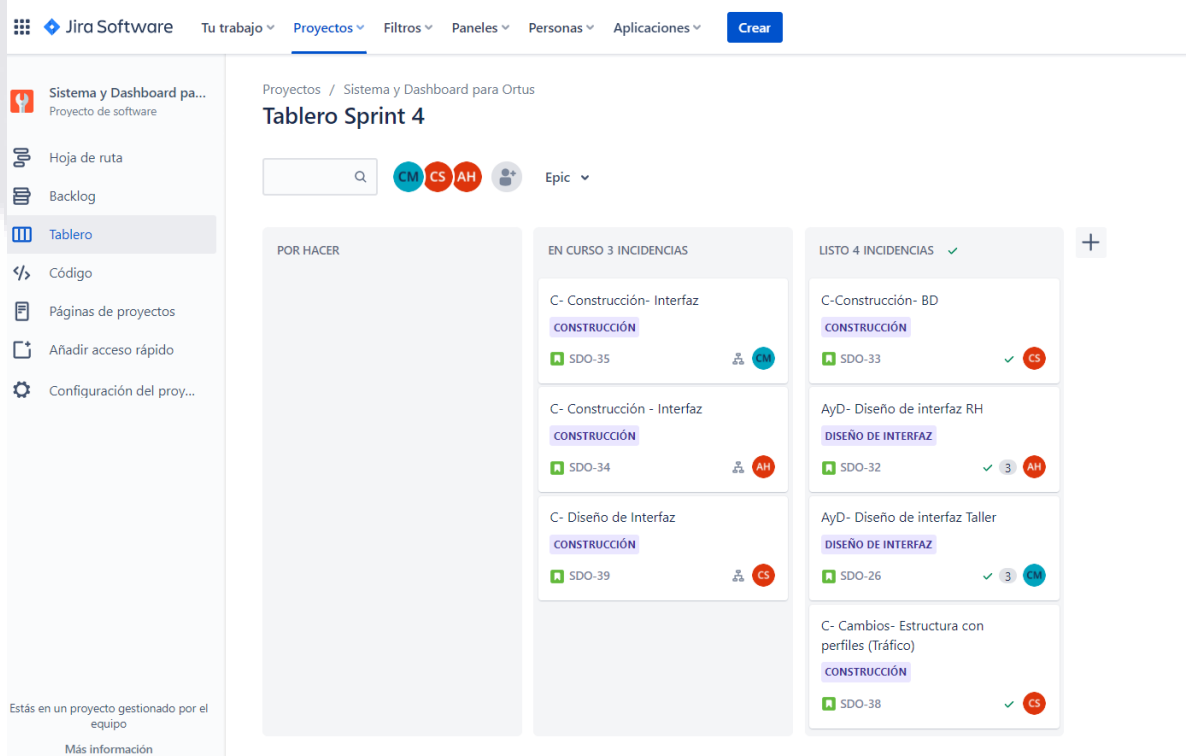


Ilustración 40 Tablero Sprint 4 (Pedroza C)

En la Ilustración 40 se puede observar que el Sprint 4 tuvo un mejor avance en los requerimientos y se han podido concluir varias actividades.

Dentro de la herramienta JIRA es posible generar informes, los cuales permiten realizar un análisis sobre los tiempos y requerimientos de cada Sprint, con el objetivo de evaluar si la cantidad actividades es la adecuada en el periodo de tiempo de cada Sprint, o poder identificar que actividades llevan más o menor tiempo y considerar una estimación más aproximada en futuros Proyectos. Estos informes cuentan con una sección donde se explica cómo leer e interpretar cada uno de dichos gráficos.

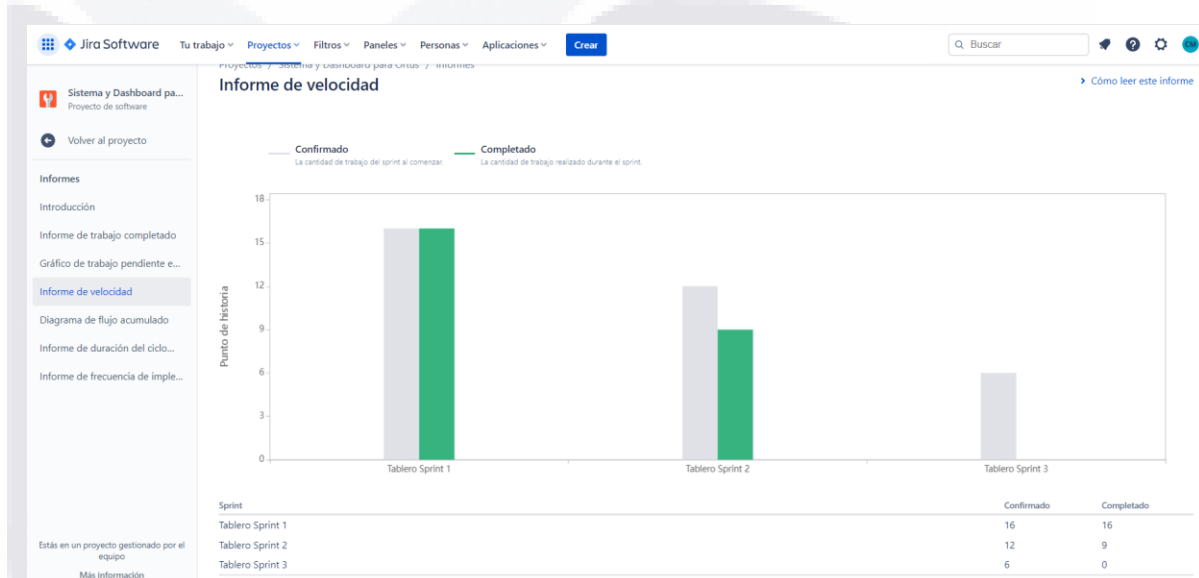


Ilustración 41 Informe de Velocidad (Pedroza C)

En la Ilustración 41 se observa un tipo de informe, el cual representa la velocidad en la que se fue atendiendo cada Sprint, es decir, la cantidad de actividades esperadas contra las que realmente fueron finalizadas. Para el Sprint 1 se logró completar las 16 que fueron definidas. En el Sprint 2 solo se completaron 9 de las 12 que se esperaba terminar y en el Sprint 3 no se complete ninguno de los requerimientos.

Otro de los informes que brinda la herramienta es un diagrama de flujo acumulado, donde se tiene un recuento de incidencias que fueron atendidos por fechas. De igual manera que el otro informe, cuenta con la sección de “Cómo leer este informe”.

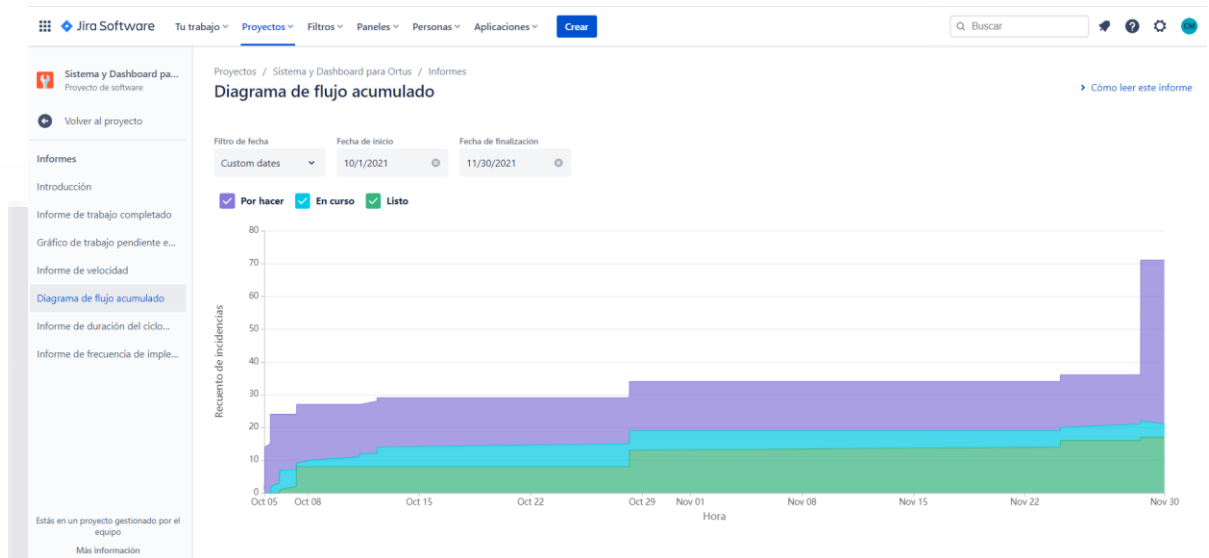


Ilustración 42 Diagrama de flujo acumulado (Pedroza C)

En la Ilustración 42 se observa un Diagrama donde se fue acumulando por cada fecha cierta cantidad de requerimientos, mismo que se puede observar que los últimos días del mes de noviembre la cantidad de actividades se incrementó exponencialmente.

La herramienta proporciona otro tipo de gráfica llamado “Informe de trabajo completado”

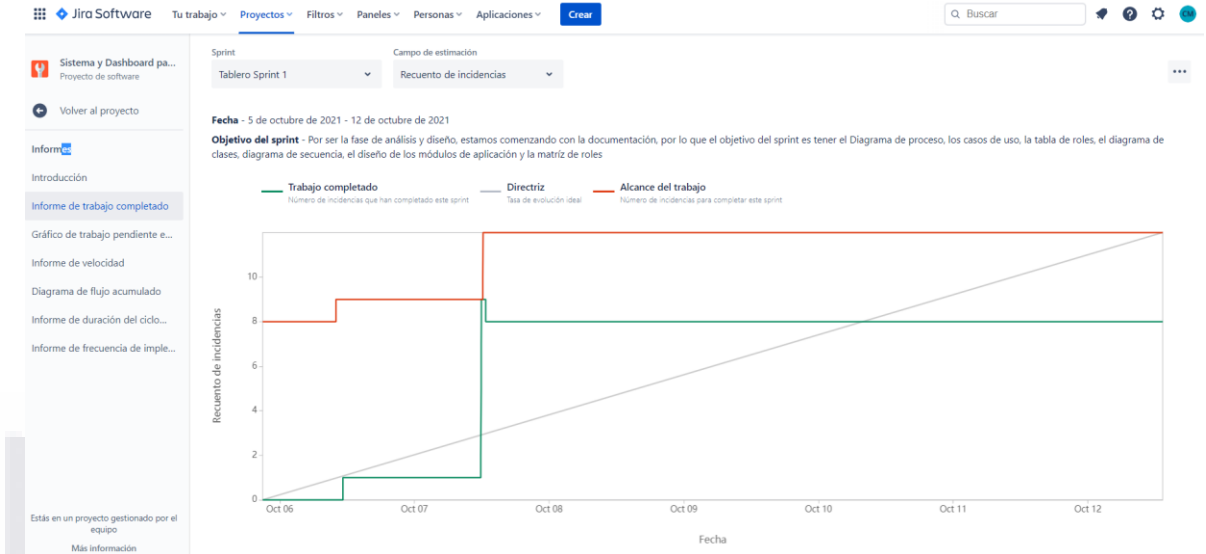


Ilustración 43 Informe de trabajo completado del Sprint 1 (Pedroza C)

En la Ilustración 43 se tiene la representación gráfica del trabajo completado, el alcance de trabajo y la Directriz, esta última línea representa el comportamiento ideal de la evolución y avance de las actividades.

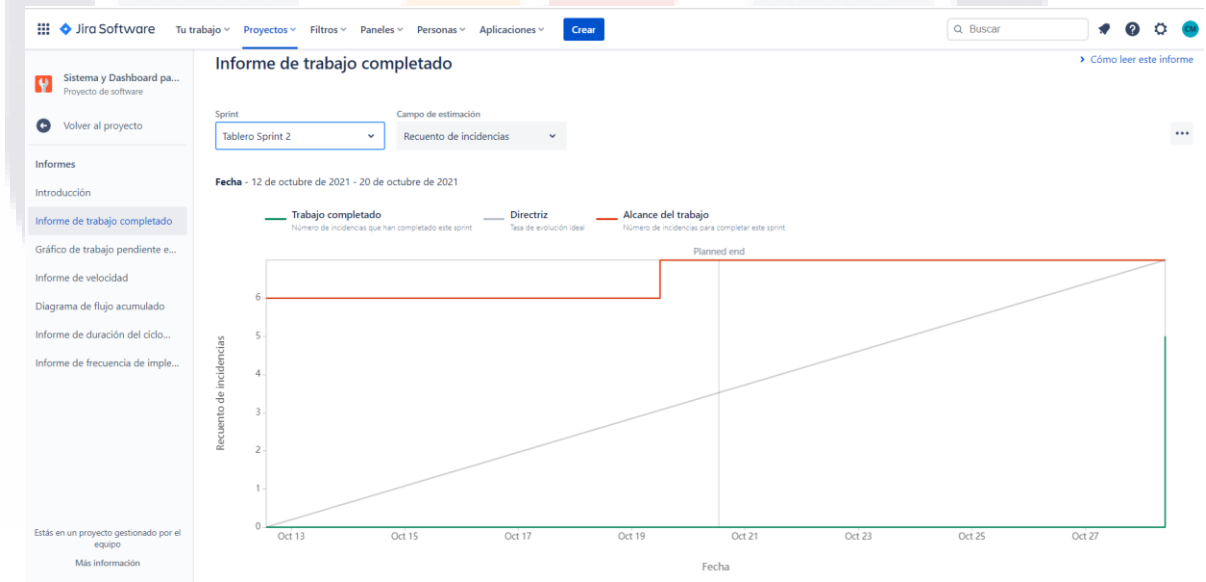


Ilustración 44 Informe de trabajo completo del Sprint 2 (Pedroza C)

En la Ilustración 44 se muestra el informe de trabajo completado del Sprint 2 y se puede apreciar que la línea está en el eje de las x en posición 0, lo que representa que no se logró completar ningún requerimiento, y la línea roja representa los 6 requerimientos que se asignaron al sprint y que se esperaba terminar, pero no se completaron en este periodo de tiempo.

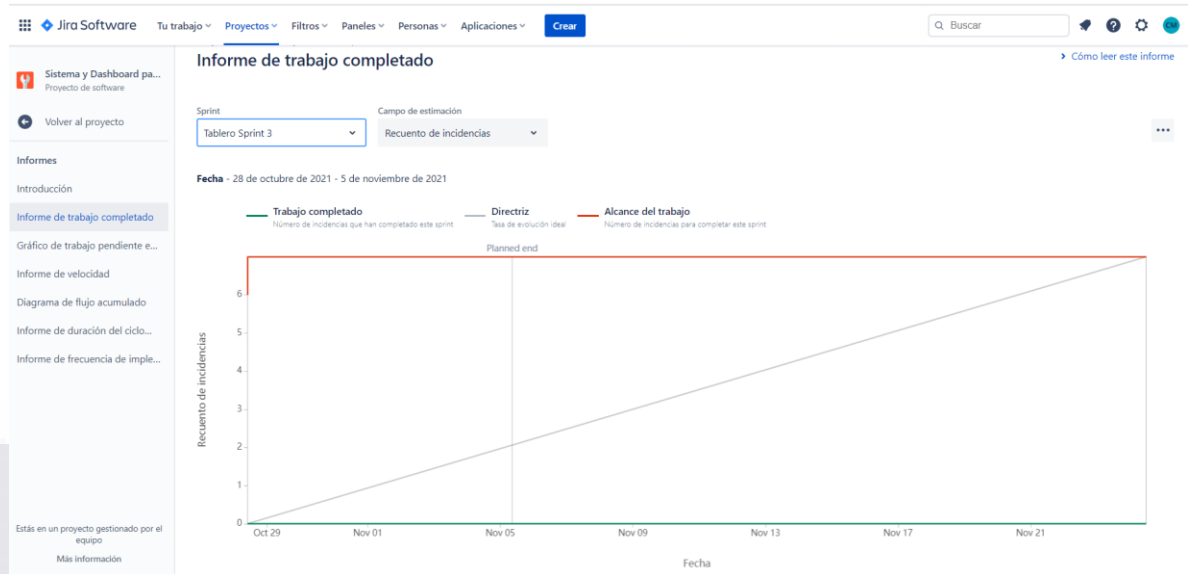


Ilustración 45 Informe de trabajo completado del Sprint 3 (Pedroza C)

En la Ilustración 45 se muestra el informe de trabajo completado del Sprint 3 y se puede apreciar que en el periodo de tiempo no se actualizó el estatus de los requerimientos, lo que muestra la línea verde que está en el eje de las x en posición 0, representa que no se logró completar ningún requerimiento, y la línea roja representa los 6 requerimientos que se asignaron al sprint y que se esperaba terminar, pero no se completaron en este periodo de tiempo.

Resultados obtenidos

Respecto a la problemática observada en algunas empresas sobre el desarrollo de software, se detectan varios inconvenientes en el seguimiento y gestión del mantenimiento de sistemas, por lo que, considerando esta situación en las organizaciones, se buscó una herramienta que facilitara las actividades de acompañamiento y gestión en la atención de requerimientos en un sistema de software como parte del mantenimiento.

Por lo anterior se propuso el uso de la metodología de la herramienta JIRA, ya que esta metodología se basa en Scrum, que permite la gestión ágil del desarrollo de software para este caso particular, se gestionó el proceso de mantenimiento. Como parte inicial del proceso se generó un cronograma, donde se pudo proyectar la fecha de inicio y fecha fin de dicho proyecto, se registraron las actividades, donde a cada actividad se le dio una ponderación de acuerdo con la magnitud de la solución y en una sesión donde el equipo se fue asignando cada actividad, se registró en la herramienta JIRA al responsable de cada actividad. Una vez definida la dimensión de las actividades y quiénes serían los responsables, cumpliendo con la priorización definida por el negocio, se asignaron las tareas a los tableros de acuerdo con la fase en la que encontraban las actividades, el primer tablero tiene por encabezado: “Por hacer”, aquí se encuentra la lista de requerimientos que se trabajaron en el sprint, una vez que comenzaron a ser atendidos por el equipo cambiaron al tablero “En curso” y cuando por fin fueron completados pasaban al tablero de “Listo”, posteriormente cuando concluía el tiempo del sprint y no se lograban terminar todas las actividades se fueron incrementando en la lista inicial, por lo que el siguiente sprint las incluía, de esta manera se fue trabajando hasta lograr que todos los requerimientos fueran concluidos, logrando como resultados: la creación de un proceso para atender las modificaciones y peticiones a un sistema que ya existe, para llevar un control y seguimiento, un tablero con detalles de las actividades, que podría leerse sin tener que preguntar al equipo, se lograron resolver los problemas que se fueron presentando y se dio un entendimiento de los requerimientos en el menor tiempo, el tiempo de atención de cada actividad fue visible y claro para el usuario, se pudieron generar gráficas para identificar los puntos de atraso y mejorar tiempos.

Con lo anterior se puede confirmar que se cumple el objetivo general propuesto sobre el seguimiento de la metodología del proceso de mantenimiento con la herramienta JIRA y al evaluar dicha herramienta en función de su metodología para cumplir con el proceso de mantenimiento.

Al finalizar los 3 meses en los que se estimó el tiempo de desarrollo del sistema web y el dashboard para el control de unidades y operadores, se logró la entrega de dicho sistema con su documentación.

Evaluación de la intervención

Al inicio del caso práctico se plantearon los siguientes objetivos:

- Realizar el seguimiento de aplicación y uso de la herramienta JIRA con su metodología para el proceso de mantenimiento de software.
- Evaluar si la metodología en la herramienta JIRA permite cumplir con el proceso de mantenimiento de software.

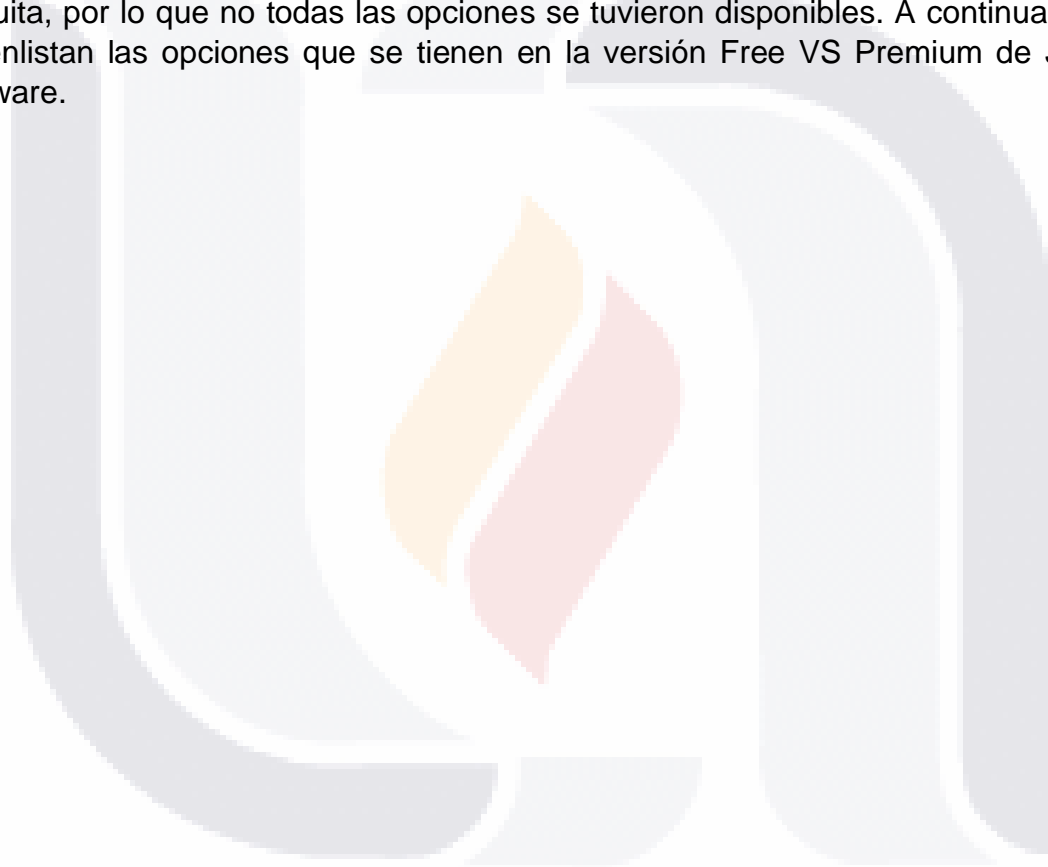
Para iniciar con el proceso de mantenimiento se generó un cronograma, donde se pudo proyectar la fecha de inicio y fecha fin de dicho proyecto, se registraron las actividades, donde a cada actividad se le dio una ponderación de acuerdo con la magnitud de la solución y en una sesión donde el equipo se fue asignando cada actividad, se registró en la herramienta JIRA al responsable de cada actividad. Una vez definida la dimensión de las actividades y quiénes serían los responsables, cumpliendo con la priorización definida por el negocio, se asignaron las tareas a los tableros de acuerdo con la fase en la que encontraban las actividades, el primer tablero tiene por encabezado: “Por hacer”, aquí se encuentra la lista de requerimientos que se trabajaron en el sprint, una vez que comenzaron a ser atendidos por el equipo cambiaron al tablero “En curso” y cuando por fin fueron completados pasaban al tablero de “Listo”. Al terminar cada sprint se dieron revisiones de las actividades pendientes y las nuevas que estaban consideradas para el siguiente sprint, por lo que se fueron definiendo el listado que sería atendido en la siguiente fase (Sprint 2) y como no se lograban terminar todas las actividades se fueron incrementando en la lista inicial, por lo que el siguiente sprint las incluía, de esta manera se fue trabajando hasta lograr que todos los requerimientos fueran concluidos. Durante los sprint, se tuvieron sesiones diarias de 15 minutos para revisar con el equipo si se tenía alguna situación particular que afectara la atención de los requerimientos. Dentro del proceso también se dio seguimiento con gráficas sobre cada sprint, estas gráficas se pueden generar en la herramienta, lo que apoya mucho en la gestión para analizar avances y aquellas situaciones que pueden atrasar, de manera que se puedan dar lecciones aprendidas para cada sprint.

Y al finalizar este proyecto, se puede concluir que se cumplieron los objetivos, ya que inicialmente se utilizó la herramienta para un proyecto de desarrollo de software, donde se siguió una metodología ágil, que proporciona JIRA Software en la versión gratuita. Adicional, se cumple el segundo objetivo, ya que la herramienta si permitió que se pudiera llevar a cabo una gestión formal y concreta de los requerimientos del negocio. El proyecto fue muy breve ya que solo abarcó 3 meses, pero la herramienta y la metodología de JIRA, permitieron mostrar avances al cliente y que el equipo tuviera la participación y compromiso necesarias para el proyecto.

Alcances y limitaciones de la intervención

Para el caso práctico se contó con un alcance limitado, ya que en El Corporativo Ortus no cuentan con el personal interno que desarrolle los sistemas que el Negocio puede necesitar, es decir, todo se lleva a través de proveedores que solo tienen un contacto mínimo, por lo que no tienen un alcance directo en la Gestión de equipos de Desarrollo que atienden requerimientos. Sin embargo, fue posible poner en práctica el objetivo del caso práctico, para evaluar la herramienta y la metodología de JIRA.

La versión que se utilizó de la herramienta JIRA Software fue una versión gratuita, por lo que no todas las opciones se tuvieron disponibles. A continuación, se enlistan las opciones que se tienen en la versión Free VS Premium de JIRA software.



	Free JIRA SOFTWARE	Premium JIRA SOFTWARE
Límite de usuarios (por sitio)	10 usuarios	20 000 usuarios
Límite de sitios	1	1
Tableros scrum y kanban	✓	✓
Backlog	✓	✓
Creación de informes ágiles	✓	✓
Flujos de trabajo personalizables	✓	✓
Aplicaciones e integraciones	✓	✓
Hojas de ruta	Basic*	Advanced**
Automatización	Un solo producto	Global y varios productos
Archivado de proyectos		✓
Verificación de dominios y captura de cuentas	✓	✓
Funciones del proyecto		✓
Permisos avanzados		✓
Información del administrador		✓
Listas de aceptación de IP		✓
Espacio aislado		✓
Canales de publicaciones		✓
Registros de auditoría		✓
Acceso anónimo		✓
Resistencia de datos		✓
SLA con tiempo de actividad garantizado →		99,9 %
Almacenamiento	2 GB de almacenamiento de archivos	Almacenamiento ilimitado
Soporte	Asistencia de la comunidad	Soporte Premium ininterrumpido (solo en inglés)

Ilustración 46 Tabla comparativa (Jira | Software de Seguimiento de Proyectos e Incidencias, n.d.)

En la ilustración 46 se presenta una tabla comparativa de la versión Free VS Premium de JIRA software.

Durante el uso de la herramienta es importante tener claro, que como en toda herramienta, se puede generar una limitación, es decir, si el equipo y los involucrados en participar en el proyecto no documentan en tiempo y forma, no es posible tener información real y con aporte a mejorar tiempos y proporcionar una gestión adecuada.

Aportes a la organización o los beneficios

Algunos usuarios del Negocio ya tenían conocimiento de una plataforma de control de código fuente, administración de proyectos y colaboración en equipo, esto dio un punto de curiosidad y expectativa para conocer JIRA y su metodología. Para este Proyecto se brindó al usuario responsable del Sistema un seguimiento muy de cerca con la atención a sus requerimientos.

La actividad de seguimiento del cliente en JIRA facilitó la revisión y solución de dudas para desarrollar módulos y atender errores que se dieron durante el periodo de desarrollo, así como en quién del equipo tomaba la responsabilidad de cada actividad.

Se podría decir que, para todo cliente, es fundamental conocer los tiempos que tardará el sistema que pide a un equipo de desarrollo, por lo que la herramienta tiene esa información de manera clara y específica, y esto le da un beneficio más y le genera tranquilidad al cliente, pues al involucrarlo con la información de estatus del proyecto, se le da una colaboración más cercana al proyecto y lo sensibiliza sobre sus requerimientos y necesidades, ya que puede ser más consciente de las afectaciones e impactos que pueden generar los cambios no considerados en el alcance original del sistema.

Recomendaciones para desarrollos futuros

Es importante tomar en cuenta que todos los integrantes del equipo siempre registren en la herramienta JIRA ya que esto permite obtener información real sobre la situación que se tiene con el proyecto, en cuanto al desarrollo, así como a las actividades pendientes y completas.

Conclusiones

JIRA software es una herramienta que muchas empresas están utilizando, y esto se debe a todas las facilidades que proporciona para conectarse con otras plataformas para el control de código fuente, la administración de proyectos y colaboración en equipo. En la actualidad estas actividades son muy importantes en la industria del software, ya que se busca que los tiempos que conllevan se reduzcan sin afectar los resultados. Para este caso práctico se utilizó la versión gratuita, el periodo de valoración de la herramienta fue de 3 meses, y con un equipo de 3 personas dedicado al proyecto.

Como se planteó en los objetivos del trabajo práctico, se logró evaluar el uso de la herramienta y su metodología para dar seguimiento a la gestión de los requerimientos, y se puede concluir de manera satisfactoria, ya que se cumplió con los tiempos estimados del proyecto y se tuvieron excelentes resultados como fue la creación de un proceso para el mantenimiento, control y seguimiento oportuno, Información real de las actividades, mejoras en los tiempos por las lecciones aprendidas y gráficas de estatus y avances.

Para un futuro en el que la empresa quiera implementar la herramienta JIRA Software en su organización, lo podrá hacer, ya que tendrá toda la confianza en que será de gran ayuda para planificar, realizar seguimientos y publicar el trabajo, así como para generar informes al respecto. Si en algún momento la empresa tiene la necesidad de adquirir la versión de Jira Software Premium, pueden tener la tranquilidad de saber que ya conocen la herramienta y han podido conocerla y la han utilizado para una colaboración entre desarrolladores de software, el líder y el usuario de negocio.

Glosario

Dashboard: es una herramienta de gestión de la información que monitoriza, analiza y muestra de manera visual los indicadores clave de desempeño (KPI), métricas y datos fundamentales para hacer un seguimiento del estado de una empresa, un departamento, una campaña o un proceso específico.

Feedback: Consiste en la manifestación de una opinión, en la demostración de un punto de vista o incluso, en el análisis de algo. El feedback es utilizado, por ejemplo, para evaluar a una persona, una empresa, un producto o un servicio.

Framework: es esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software.

IEEE: Instituto de Ingenieros eléctricos y electrónicos (Institute of Electrical and Electronics Engineers), es una asociación mundial de ingenieros dedicada a la normalización y el desarrollo en áreas técnicas.

Ingeniería del Software: es la disciplina que trata del establecimiento y uso de métodos y principios sólidos de ingeniería para obtener software fiable que se ejecute en máquinas reales. En definitiva, la ingeniería del software pretende aplicar en lo posible los procedimientos, bien conocidos y probados, de otras ingenierías considerando los aspectos tecnológicos y burocráticos relacionados con el diseño, producción y mantenimiento sistemáticos de programas de computadores.

Issues: La palabra issue sirve para describir una situación que puede ser difícil. O como su traducción lo dice, son problemas o errores.

JIRA: es una herramienta de Atlassian, su objetivo es desarrollar increíbles productos, prácticas y entornos de trabajo abiertos para todos ellos en materia de software.

Kanban: es un método para gestionar el trabajo que surgió en Toyota Production System (TPS). A finales de los años 40, Toyota implementó en su producción el sistema “just in time” (justo a tiempo) que en realidad representa un sistema de arrastre. Esto significa que la producción se basa en la demanda de los clientes y no en la práctica tradicional “pull” de fabricar productos e intentar venderlos en el mercado.

Metodología: Conjunto de métodos que se siguen en una investigación científica, un estudio o una exposición doctrinal.

Metodología XP: “Extreme Programming” o “Programación Extrema” es una de las llamadas metodologías Ágiles de desarrollo de software más exitosas. Es habitual relacionarla con scrum, y la combinación de ambas asegura un mayor control sobre el proyecto, y una implementación más efectiva y eficiente.

Scrum: es un marco que permite el trabajo colaborativo entre equipos. Al igual que un equipo de rugby (de donde proviene su nombre) cuando entrena para un gran partido, scrum anima a los equipos a aprender a través de las experiencias, a autoorganizarse mientras aborda un problema y a reflexionar sobre sus victorias y derrotas para mejorar continuamente.

Sistema de información: Es el conjunto formal de procesos que operando sobre una colección de datos estructurada de acuerdo con las necesidades de una empresa, recopila, elabora y distribuye la información necesaria para la operación de dicha empresa y para las actividades de dirección y control correspondientes.

Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

Sprint: es el nombre que se le da al ciclo o iteraciones que se trabajan en un proyecto de Scrum.

Bibliografía

1219-1998 - 1219-1998 - *Estándar IEEE para mantenimiento de software - Estándar IEEE*. (n.d.). Retrieved December 9, 2020, from <https://ieeexplore-ieee.org.dibpxy.uaa.mx/document/720567>

Elementos de un Sistema de Gestion de Calidad basado en las Normas ISO 9001:2008 - Monografias.com. (n.d.). Retrieved July 9, 2020, from <https://www.monografias.com/trabajos88/elementos-sistema-gestion-calidad-basado-normas-iso-9001-2008/elementos-sistema-gestion-calidad-basado-normas-iso-9001-2008.shtml>

Extreme Programming Explained: Embrace Change - Kent Beck - Google Libros. (n.d.). Retrieved May 29, 2022, from https://books.google.es/books?id=G8EL4H4vf7UC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Hernández Bejarno, M., & Baquero Rey, L. E. (n.d.). *Ciclo de vida de desarrollo ágil de software seguro*.

Jira | Software de seguimiento de proyectos e incidencias. (n.d.). Retrieved November 19, 2020, from <https://www.atlassian.com/es/software/jira>

Jira Software. (2018). *Jira Software: Funciones | Atlassian*. <https://www.atlassian.com/es/software/jira/features%0Ahttps://es.atlassian.com/software/jira/features>

Lapiedra Alcamí, R., Devecce Carñana, C., & Guiral Herrando, J. (2011). Introducción a la gestión de sistemas de información en la empresa. In *Universitat Jaume I. Servei de Comunicació i Publicacions* (Vol. 2010). <https://elibro.net/es/lc/uaa/titulos/51689>

Marta, P. (n.d.). *Scrum Master*. https://scrummanager.net/files/scrum_master.pdf

Pardo Alvarez, J. M. (2012). *Configuración y usos de un mapa de procesos*. Aenor - Asociación Espano.

Pressman R.S. (2013). Ingeniería de Software un enfoque práctico. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).

uaa - Mantenimiento y evolución de sistemas de información. (n.d.). Retrieved November 20, 2020, from <https://elibro.net/es/lc/uaa/titulos/127090>

Una guía para el CUERPO DE CONOCIMIENTO DE SCRUM (Guía SBOK™) Una guía completa para la entrega de proyectos utilizando Scrum. (2016). www.scrumstudy.com