

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería en Electricidad y Computación**



**“IMPLEMENTACIÓN DE UNA METODOLOGÍA PARA ATENDER  
REQUERIMIENTOS DE CAMBIOS O MEJORAS EN LOS APLICATIVOS DE  
UNA EMPRESA DE LOGÍSTICA DE COMERCIO EXTERIOR”**

**EXAMEN DE GRADO (COMPLEXIVO)**

PREVIO A LA OBTENCIÓN DEL TÍTULO DE

**MAGÍSTER EN SISTEMAS DE INFORMACIÓN  
GERENCIAL**

AUTOR

EDUARDO XAVIER CORONEL VARGAS

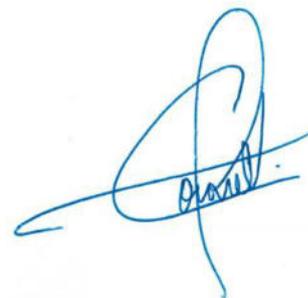
GUAYAQUIL, FEBRERO 2021

## **AGRADECIMIENTO**

A nuestro Padre Celestial que me da la vida, sabiduría y las herramientas necesarias para mi crecimiento personal y profesional y a mis padres quienes me han transmitido los valores y bases suficientes para ser cada día un mejor ser humano.

## DEDICATORIA

Dedico este trabajo a mi familia, a mi esposa Nancy quien es mi ayuda idónea y mi fuente de inspiración y a mis hijos Thiago y Valeria quienes son el motor que me impulsa día a día a ser la mejor versión de mí mismo.

A handwritten signature in blue ink, consisting of a large, stylized initial 'A' followed by a horizontal line and a vertical stroke.

## TRIBUNAL DE SUSTENTACIÓN



---

MSIG. Lenín Freire Cobo

COORDINADOR DE MSIG



---

MSIG. Juan Carlos García

PROFESOR MSIG

## RESUMEN

El presente trabajo tiene como objetivo principal brindarle a la Empresa de Logística de Comercio Exterior una metodología que muestre los pasos a seguir en el desarrollo de los proyectos de software del departamento de sistemas con la finalidad minimizar pérdidas de tiempo y garantizar un entregable de calidad.

Diariamente nacen proyectos de mejoras en los sistemas que surgen por el crecimiento natural de la empresa, las reuniones con usuarios, correos con solicitudes de cambios, incorporación de nuevas áreas de negocios, adquisición de nuevo software y demás requerimientos que muchas veces no pueden ser priorizados correctamente. Tampoco se realiza un control o seguimiento diario que muestre los avances o indique la existencia de bloqueantes de cualquier índole que dificulten la continuidad del proyecto. Además, en reiteradas ocasiones el programador es quien, directamente, debe agendar múltiples reuniones con los usuarios o interesados finales para aclarar requerimientos o definir alcances, lo cual genera mayores retrasos y pérdidas de tiempo.

En este documento se muestra un marco de trabajo o procedimiento que deberá seguirse cuando se requiera realizar un proyecto por mejoras o cambios en los procesos o sistemas que actualmente existen en la empresa,

así como también en la incorporación de nuevos productos de software. Se definirán los roles del departamento de desarrollo con las tareas asignadas a cada uno de ellos, se capacitará al personal en el uso de esta metodología y finalmente medirán los resultados para verificar que la metodología esté funcionando adecuadamente.

## ÍNDICE GENERAL

AGRADECIMIENTO .....	i
DEDICATORIA .....	ii
TRIBUNAL DE SUSTENTACIÓN .....	iii
RESUMEN .....	iv
ÍNDICE GENERAL .....	vi
ÍNDICE DE TABLAS .....	ix
ÍNDICE DE FIGURAS .....	x
INTRODUCCIÓN .....	xii
CAPÍTULO 1 .....	1
GENERALIDADES .....	1
1.1 DESCRIPCIÓN DEL PROBLEMA .....	1
1.2 SOLUCIÓN PROPUESTA .....	4
CAPÍTULO 2 .....	6
METODOLOGÍA PARA LA IMPLEMENTACIÓN DE LA SOLUCIÓN .....	6
2.1 LEVANTAMIENTO DE LA INFORMACIÓN .....	6
2.2 DESARROLLO ÁGIL Y SCRUM .....	8
2.3 TIPOS DE REQUERIMIENTOS .....	9
2.3.1 REQUERIMIENTOS DE MANTENIMIENTO .....	9
2.3.2 REQUERIMIENTOS DE PROYECTO .....	11
2.4 USUARIOS Y ROLES .....	13
2.4.1 ROL DE PROGRAMADOR .....	14

2.4.2 ROL DE FACILITADOR .....	14
2.4.3 ROL DUEÑO DEL PRODUCTO.....	15
2.5 EL SPRINT .....	16
2.5.1 EL LISTADO DE PENDIENTES .....	17
2.5.2 LA PLANIFICACIÓN.....	18
2.5.3 ESTIMACIÓN DE PUNTOS DE HISTORIA .....	19
2.5.4 EL DESARROLLO Y PRUEBAS .....	20
2.5.5 LA REUNIÓN DIARIA.....	22
2.5.6 LA RETROSPECTIVA.....	22
2.5.7 PASE A PRODUCCION .....	24
2.6 DOCUMENTACIÓN.....	24
2.6.1 MATRIZ DE PRUEBAS .....	25
2.6.2 FORMULARIO DE PASE A PRODUCCIÓN .....	26
2.6.3 BITÁCORA DE RETROSPECTIVA.....	27
2.6.4 ACTA DEL SPRINT.....	28
2.7 SOFTWARE PARA LA GESTIÓN DE LA METODOLOGÍA .....	29
2.7.1 CONFIGURACIÓN DE JIRA SOFTWARE .....	30
2.7.2 REGISTRO DE EMPRESA .....	31
2.7.3 REGISTRO DE USUARIOS.....	35
2.7.4 AGREGAR SPRINTS Y TAREAS .....	36
2.7.5 GESTIÓN DEL TABLERO DE PROYECTOS .....	41
2.7.6 GESTIÓN DEL TABLERO DE MANTENIMIENTO.....	43

2.7.7 GESTIÓN DE LA DOCUMENTACIÓN EN CONFLUENCE .....	45
2.8 PLAN DE IMPLEMENTACIÓN .....	50
CAPÍTULO 3.....	52
EVALUACIÓN DE RESULTADOS.....	52
3.1 MONITOREO Y MEDICIONES DE LA IMPLEMENTACIÓN .....	52
3.2 ANÁLISIS COMPARATIVO .....	54
3.3 BENEFICIOS DE LA METODOLOGÍA .....	58
CONCLUSIONES Y RECOMENDACIONES .....	60
BIBLIOGRAFÍA.....	63
ANEXOS .....	64
ANEXO 1. MATRIZ DE PRUEBAS.....	64
ANEXO 2. FORMULARIO DE PASE A PRODUCCIÓN .....	64
ANEXO 3. BITÁCORA DE RETROSPECTIVA.....	64
ANEXO 4. ACTA DEL SPRINT .....	65
GLOSARIO .....	66

## ÍNDICE DE TABLAS

Tabla 1 – Roles asignados a usuarios del departamento .....	13
Tabla 2 - Herramientas para la gestión de la metodología .....	30

## ÍNDICE DE FIGURAS

Figura 2.1 – Definición de la Tarea .....	10
Figura 2.2 – Definición de la Historia de Usuario .....	11
Figura 2.3 – Épicas e Historias de Usuario .....	12
Figura 2.4 – Acceso al portal de configuración de Jira .....	31
Figura 2.5 – Registro de cuenta de correo en Jira .....	32
Figura 2.6 – Registro del sitio en Jira.....	32
Figura 2.7 – Registro de correos del equipo de desarrollo .....	33
Figura 2.8 – Configuración general de Jira .....	33
Figura 2.9 – Crear proyecto (tablero) en Jira .....	34
Figura 2.10 – Tablero de Proyectos Jira .....	34
Figura 2.11 – Invitar a las personas del equipo de desarrollo.....	35
Figura 2.12 – Personas del equipo de desarrollo accediendo a Jira.....	36
Figura 2.13 – Crear sprint .....	37
Figura 2.14 – Crear historia de usuario o tarea .....	37
Figura 2.15 – Ingreso del detalle de la historia de usuario.....	38
Figura 2.16 – Listado de Historias Pendientes .....	40
Figura 2.17 – Carga de trabajo por usuario asignado.....	40
Figura 2.18 – Iniciar Sprint.....	41
Figura 2.19 – Tablero de Sprint Actual .....	42

Figura 2.20 – Tablero de Mantenimiento .....	43
Figura 2.21 – Tarea de Mantenimiento .....	44
Figura 2.22 – Acceso a Confluence .....	45
Figura 2.23 – Espacios de Trabajo .....	46
Figura 2.24 – Documentación del Tablero de Proyectos .....	47
Figura 2.25 – Agregar link de Historia Jira .....	48
Figura 2.26 – Documentación del Sprint y sus Historias.....	49
Figura 2.27 – Documentación del Tablero de Mantenimiento.....	50
Figura 3.1 – Efectividad en la Planificación .....	54
Figura 3.2 – Productividad .....	55
Figura 3.3 – Mantenimiento .....	57
Figura 3.4 – Calidad.....	58

## INTRODUCCIÓN

En este documento se muestra la problemática de una Empresa de Logística de Comercio Exterior, la metodología que se implementó para solventarla y la evaluación de los resultados obtenidos.

A continuación se describen los apartados del documento. En el capítulo 1 – Generalidades, se reconoce el problema y se propone una solución para solventarlo. En el capítulo 2 – Metodología para la Implementación de la solución, se detallan los tipos de requerimientos que se presentan en el día a día, los usuarios y roles de la metodología, el sprint con sus dinámicas, la documentación que se utiliza y finalmente el software para la gestión de la metodología. En el capítulo 3 – Evaluación de Resultados, se evalúa la metodología y se listan los beneficios de la solución.

Finalmente, el documento presenta las conclusiones, recomendaciones y la bibliografía.

# **CAPÍTULO 1**

## **GENERALIDADES**

### **1.1 DESCRIPCIÓN DEL PROBLEMA**

La Empresa de Logística de Comercio Exterior ofrece, de forma general, servicios de asesoría en trámites aduaneros, consolidación de carga, transporte nacional e internacional y estiba de mercadería. Contaba físicamente con un pequeño departamento de desarrollo de sistemas y soporte tecnológico, en el cual trabajaban el jefe departamental, cuatro programadores y tres personas para redes y soporte técnico.

Con el tiempo, la empresa creció, se abrieron más áreas de negocios y por tanto ingresó más personal que requería de atención a nivel técnico y nuevas implementaciones en los sistemas. Por lo cual, la empresa adquirió un nuevo edificio con amplias instalaciones y se dividió el departamento en dos áreas: soporte técnico y desarrollo de software.

El área de desarrollo está compuesta de un coordinador de desarrollo, cuatro programadores y una persona encargada de la mesa de ayuda que recepta los tickets creados por los usuarios y ofrece atención de primer nivel. En caso de que el ticket no pueda ser resuelto por mesa de ayuda, el coordinador asigna el caso, de acuerdo a la prioridad, a uno de los programadores y realiza el seguimiento del mismo hasta el cierre.

Sin embargo, el área de desarrollo también tiene otros proyectos de mejoras en los sistemas que surgen por el crecimiento natural de la empresa, las reuniones con usuarios, correos con solicitudes de cambios, incorporación de nuevas áreas de negocios, adquisición de nuevo software, etc. En la actualidad, para atender estos proyectos, el coordinador envía un correo electrónico con el requerimiento a uno o varios programadores quienes se encargan de reunirse con el usuario o con el coordinador para despejar dudas, realizar el desarrollo y las pruebas integrales hasta la entrega del mismo.

En los proyectos pequeños, que toman poco tiempo y solo requieren de la atención de un programador, no se presentan mayores inconvenientes. Sin embargo, para proyectos más grandes, que requieren más tiempo de desarrollo, en donde trabajan en conjunto varios programadores y el desarrollo debe orquestarse para hacer las pruebas y entregar el producto final al usuario, se presentan a menudo problemas tales como:

- Falta de definición de requerimientos.
- No están claros los alcances del proyecto.
- Se agendan varias reuniones consecutivas con usuarios para despejar dudas.
- No se establecen tiempos objetivos en la entrega de los proyectos.
- No existen niveles de priorización en los entregables.
- No se realiza un correcto seguimiento del proyecto.
- Carga de trabajo.
- Cambios en producción.

Además, a estos problemas podría sumarse la pérdida de tiempo y en muchas ocasiones, inconformidad de parte del usuario, que nuevamente solicitará cambios en los sistemas.

## 1.2 SOLUCIÓN PROPUESTA

La solución que se propone para resolver el problema detallado en la sección anterior es la implementación de una metodología que muestre los lineamientos necesarios para que los programadores cuenten con toda la información necesaria para desarrollar los proyectos, minimizando las pérdidas de tiempo por falta de definiciones, alcances, reuniones indefinidas por aclaración de requerimientos, entregas tardías de los proyectos, cambios post-producción, etc.

Para su implementación nos basaremos en la metodología de desarrollo ágil Scrum y un conjunto de mejores prácticas para el desarrollo de sistemas.

La finalidad del presente documento es el de mostrar un marco de trabajo o procedimiento que deberá seguirse cuando se requiera realizar un proyecto por mejoras o cambios en los procesos o sistemas que actualmente existen en la empresa, así como también en la incorporación de nuevos productos de software. También se definirán los roles del departamento de desarrollo con las tareas asignadas a cada uno de ellos.

Además, se realizarán mediciones periódicas y análisis de las mismas con la finalidad de realizar ajustes que permitan mejorar la gestión de la metodología implementada.

## **CAPÍTULO 2**

### **METODOLOGÍA PARA LA IMPLEMENTACIÓN DE LA SOLUCIÓN**

#### **2.1 LEVANTAMIENTO DE LA INFORMACIÓN**

En el departamento de Desarrollo de Sistemas de la Empresa de Logística de Comercio Exterior se implementan soluciones de software de acuerdo a las exigencias del negocio y entidades regulatorias.

Cada día llegan solicitudes de ajustes a los sistemas existentes, tales como cambios en las parametrizaciones, generación de incidentes o errores, implementación de nuevo software, creación de nuevos reportes, etc., que son captados a través de la mesa de ayuda, correos electrónicos y reuniones de trabajo.

El coordinador de desarrollo se encarga de distribuir los proyectos o tareas entre el equipo de programadores, a los cuales les indica la prioridad de cada una y, en caso de que una tarea sea muy extensa, se asigna a dos o más personas. De acuerdo a la complejidad de la tarea pueden agendarse varias reuniones de trabajo entre los usuarios funcionales (que presentaron el requerimiento) y el coordinador de desarrollo (se puede incluir también a los programadores) para explicar los alcances del proyecto. No se definen cronogramas de actividades, sólo se plantea una fecha estimada de culminación ya que un programador puede estar atendiendo más de un proyecto a la vez.

El programador realiza el desarrollo del proyecto y una vez culminado se agenda una revisión con los usuarios, los cuales prueban el aplicativo, hacen sus observaciones y aprueban el trabajo realizado. El programador realiza los ajustes de acuerdo a las observaciones del usuario y envía el pase a producción. Cuando ya está en producción el

coordinador le indica al usuario final para que verifique las funcionalidades implementadas y finaliza la atención del requerimiento o proyecto.

La solicitud formal del requerimiento se realiza por correo electrónico interno, de igual manera las bitácoras de reuniones con usuarios, las pruebas, aprobaciones y todo lo relacionado al proyecto hasta la culminación del mismo.

## **2.2 DESARROLLO ÁGIL Y SCRUM**

Para mitigar los problemas presentados en el capítulo anterior, en el departamento de sistemas decidió adoptar una metodología de desarrollo basada en el marco de trabajo Scrum, combinado con el uso de buenas prácticas como documentación de requerimientos y pruebas, consenso en estimaciones de tiempo de desarrollo de tareas, pruebas cruzadas, etc. que permitan minimizar pérdidas de tiempo y mejorar la calidad del software a entregarse al usuario final.

Scrum es un marco de trabajo que permite a los programadores desarrollar las tareas en forma colaborativa, de tal manera que se pueda

dividir el trabajo y generar pequeños entregables que aporten valor y que luego se integren hasta culminar la épica o el proyecto final.

## **2.3 TIPOS DE REQUERIMIENTOS**

De acuerdo al levantamiento de información realizado se puede determinar que existen dos tipos de requerimientos:

- Requerimientos de Mantenimiento
- Requerimientos de Proyectos

### **2.3.1 REQUERIMIENTOS DE MANTENIMIENTO**

Definimos que un requerimiento de mantenimiento es una tarea, normalmente pequeña y de poco esfuerzo. Surgen por una solicitud que llega a la mesa de ayuda o correo del usuario que lo requiere.

Ejemplos de este tipo de tareas serían:

- Agregar valores a un parámetro de la base de datos
- Modificar el mensaje de alguna alerta
- Modificar algún archivo de propiedades de un aplicativo

- Cambiar la etiqueta de alguna opción del aplicativo
- Cambiar el tamaño de un campo de una tabla de la base de datos

Estas tareas se agregarán a un tablero de mantenimiento a la columna “Por Hacer” y se van asignando y realizando de acuerdo a la prioridad determinada por el coordinador de sistemas.

La tareas pueden ser asignadas a los programadores o, en caso de ser una tarea sencilla puede ser resuelta directamente por la persona que atiende la mesa de ayuda, para lo cual recibirá la capacitación y respaldo de cualquier miembro del equipo. La descripción de la tarea se planteará bajo el estilo “Problema/Solución” (Ver figura 2.1).

**Figura 2.1 – Definición de la Tarea**

The image shows a screenshot of a task definition interface. At the top, the title is "Agregar opción en el combo Formularios de la Pantalla de Registro de Comprobantes". Below the title, there are three buttons: "Adjuntar", "Crear subtarea", and "Vincular incidencia", followed by a dropdown arrow and a three-dot menu icon. Below the buttons, there is a "Descripción" section. Under "Descripción", there are two sub-sections: "Problema:" and "Solución:". The "Problema:" section contains the text: "Se necesita que el usuario pueda elegir la opción de "Liquidación de Compras" en el combo Formularios de la Pantalla de Registro de Comprobantes". The "Solución:" section contains the text: "Agregar la opción Liquidación de Compras en el parámetro FOR01-RC de la tabla FR\_PARAMETROS en la BD." To the right of the title and description sections, there are two red brackets. The top bracket is labeled "TÍTULO DE LA TAREA" and the bottom bracket is labeled "DESCRIPCIÓN DE LA TAREA".

**Agregar opción en el combo Formularios de la Pantalla de Registro de Comprobantes**

Adjuntar   Crear subtarea   Vincular incidencia   ▾   ⋮

Descripción

**Problema:**  
Se necesita que el usuario pueda elegir la opción de "Liquidación de Compras" en el combo Formularios de la Pantalla de Registro de Comprobantes

**Solución:**  
Agregar la opción Liquidación de Compras en el parámetro FOR01-RC de la tabla FR\_PARAMETROS en la BD.

TÍTULO DE LA TAREA

DESCRIPCIÓN DE LA TAREA

Fuente: Creado por el autor

### 2.3.2 REQUERIMIENTOS DE PROYECTO

Definimos que un requerimiento es de proyecto cuando se trata de solicitudes de desarrollo de mediana o alta complejidad. Normalmente surgen de las necesidades de los usuarios, por nuevos alcances o definiciones, cambios en las políticas de gobierno, nuevos controles, mejoras en seguridad, etc.

Para estos requerimientos se crean “historias de usuario”, a diferencia de los de mantenimiento, donde sólo existen “tareas”. Las historias de usuario son una descripción corta de la funcionalidad que se desea obtener una vez que el producto esté en ambiente de producción. Van acompañadas de los criterios de aceptación, que constituyen los requisitos básicos para que la historia sea aprobada.

**Figura 2.2 – Definición de la Historia de Usuario**



Fuente: Creado por el autor

Un ejemplo de cómo se detallaría una historia de usuario se muestra en la figura 2.2. Nótese que la descripción se compone de tres partes (palabras en negrita). Primero, se señala la persona que lo solicita. Luego, se pone la necesidad y finalmente el objetivo al que se quiere llegar.

Un proyecto puede estar compuesto de varias historias de usuario relacionadas, a esto también se le conoce con el nombre de épica. El objetivo es que cada historia de usuario sea una parte funcional del proyecto asignada a una persona específica encargada de su desarrollo.

*Figura 2.3 – Épicas e Historias de Usuario*



Fuente: Creado por el autor

Las historias de usuario deben asignarse a un único programador (Ver figura 2.3). En caso de que la historia sea muy extensa deberá dividirse

de tal manera que sólo exista un solo responsable por tarea o historia ser asignadas a los programadores.

## 2.4 USUARIOS Y ROLES

De acuerdo al levantamiento de información realizado se estableció que en el departamento de desarrollo de sistemas existen:

- Cuatro ingenieros de desarrollo de software
- Una persona en la gestión de tickets de la mesa de ayuda
- Un coordinador de desarrollo
- Un coordinador de proyectos

**Tabla 1 – Roles asignados a usuarios del departamento**

Usuarios	RoI Scrum
Programadores, Ingenieros de desarrollo y Mesa de Ayuda	Programador
Coordinador de Desarrollo	Facilitador
Coordinador de Proyectos	Dueño del Producto

Fuente: Elaborado por el autor

De acuerdo con estas definiciones, en la tabla 2.1 se establecen los roles de la metodología Scrum asignados a cada uno de los usuarios del departamento de desarrollo de sistemas.

#### **2.4.1 ROL DE PROGRAMADOR**

El rol de Programador se asigna al personal de desarrollo de software. Se incluye además a la persona encargada de la mesa de ayuda, ya que a parte de realizar el soporte de primer nivel, se encarga de resolver determinados tareas sencillas de mantenimiento, como creación/modificación de parámetros, elaboración de documentación, limpieza de archivos de logs, etc.

Su función principal es la de desarrollar nuevas funcionalidades o modificar las existentes en las diversas aplicaciones de la empresa, en función de los requerimientos indicados en las historias de usuario o en las tareas.

#### **2.4.2 ROL DE FACILITADOR**

El rol de Facilitador se asigna al coordinador de desarrollo y se encarga principalmente de revisar que los programadores no tengan ningún

bloqueante que les impida continuar con el desarrollo. Dispone de amplia experiencia técnica y funcional de las aplicaciones, por lo cual capacita a los programadores, de ser necesario, para que realicen sus tareas.

Se encarga de dar seguimiento a las tareas y verificar el estado de las mismas. Se reúne con el Dueño del Producto para aclarar los requerimientos funcionales y propone opciones para los cambios a realizarse en las aplicaciones.

Además, revisa que se estén cumpliendo las metas del equipo de desarrollo, los estándares de buenas prácticas en la programación, que se suba el código a los repositorios, se eliminen los bloqueantes que se pudiesen presentar y se realice la documentación de las pruebas previo al pase a producción.

### **2.4.3 ROL DUEÑO DEL PRODUCTO**

El rol de Dueño del Producto se asigna al Coordinador de Proyectos y su función principal es la de elaborar las historias de usuario en base a los requerimientos solicitados y les otorga la prioridad de atención.

Tiene bajo su tutela las aplicaciones de la empresa y por tanto, como dueño de las mismas solicita los cambios, mejoras o incorporación de

nuevos productos y servicios en coordinación con el Jefe del Departamento de Sistemas.

Es el encargado de explicar las historias de usuario a los programadores y actualiza el detalle de cada historia para que el programador pueda guiarse al realizar la programación.

Además, coordina reuniones con los usuarios finales para probar las nuevas funcionalidades y de aclaración de requerimientos en caso de ser necesario.

## **2.5 EL SPRINT**

Se define como el intervalo de tiempo del que disponen los programadores para entregar las historias o las tareas que se especifiquen. Para el caso de esta metodología será de diez días laborables.

Durante este intervalo de tiempo o ciclo se realizarán varias actividades que permitirán trabajar en conjunto al equipo para cumplir con los objetivos propuestos.

### 2.5.1 EL LISTADO DE PENDIENTES

Antes de iniciar el sprint, el Dueño del Producto se encargará de realizar el levantamiento de información de cada uno de los proyectos a desarrollarse, lo cual puede incluir análisis de factibilidad, definiciones de alcances, reuniones con usuarios finales, etc. Todo depende de la complejidad de cada proyecto.

Finalmente, creará todas las historias necesarias para el desarrollo, con el formato indicado anteriormente (*ver subcapítulo 2.3.2*). A esto se le denomina “crear el listado de pendientes o crear el backlog” y se puede usar una aplicación tipo procesador de palabras (como Word) o una herramienta especializada como Jira (la cual veremos más adelante).

Una vez creadas las historias de usuario, el Dueño del Producto se reunirá con el Facilitador para revisar el contenido de cada una, agregar o quitar criterios de aceptación, dividir las en caso de que estén muy extensas, etc. A esto se le conoce como “afinar el listado de pendientes o afinar el backlog”.

## 2.5.2 LA PLANIFICACIÓN

Una vez establecidas las tareas pendientes el Dueño del Producto convoca una reunión llamada Planificación, donde se muestra lo que va a realizarse durante el periodo que dura el Sprint y participa todo el equipo. Se revisan las tareas del listado de pendientes y cada programador elige las historias o tareas que desea tomar. El Sprint siempre iniciará con la Planificación.

El Dueño del Producto actúa como moderador de la reunión y explica cada una de las historias y sus criterios de aceptación. Los miembros del equipo pueden hacer preguntas para despejar dudas y se pueden realizar correcciones a la descripción de cada historia en caso de requerirse.

No es necesario que los programadores tengan que tomar tareas similares a otras que hayan realizado antes, sino que pueden tomar historias con temas totalmente nuevos, puesto que la idea es que todos tengan conocimientos similares, de tal manera, que ante la ausencia de algún programador, otro pueda tomar sus historias fácilmente. Adicionalmente, el Facilitador puede convocar una reunión de capacitación interna o externa (en caso de que llegue un capacitador ajeno a la empresa) y deberá ser considerada como una tarea del Sprint.

Se le asignará un programador a cada historia. En caso de que existan varias historias relacionadas se podrá crear una épica con la descripción general del proyecto y se hace referencia a la misma en cada historia.

### **2.5.3 ESTIMACIÓN DE PUNTOS DE HISTORIA**

Las historias o tareas del sprint deberán tener asignado un puntaje que se denomina “Puntos de Historia”. Este puntaje lo asignan los programadores en consenso, en función de la complejidad de la historia. Los valores que puede tomar están dados en función de la escala Fibonacci: 1, 2, 3, 5 y 8.

Cabe destacar que la estimación de este puntaje depende mucho del criterio y la experiencia del programador (al haber realizado historias similares). Por ejemplo, una tarea de validar la obligatoriedad de un campo de una pantalla, puede ser fácil para el programador que haya realizado una tarea similar o conozca el módulo donde deben hacerse los ajustes y le otorgue el puntaje de 3. Pero para otro programador que no conozca el aplicativo puede ser de 8. Para este caso, el punto de historia más adecuado será el promedio de los puntos, o sea el valor Fibonacci de 5.

Si los programadores consideran que una historia es más compleja que 8 puntos, esa tarea debe dividirse y formar otra historia.

#### **2.5.4 EL DESARROLLO Y PRUEBAS**

Una vez que se hayan asignado todas las historias a los programadores con su respectiva puntuación inicia el desarrollo del Sprint. Cada programador tomará sus historias en el orden que desee y empezará a trabajar en ellas. Estas tareas pasarán del estado “Por Hacer” al estado “En Curso”.

El programador deberá realizar el desarrollo de acuerdo con los estándares y buenas prácticas de programación, con la finalidad de generar código de calidad.

Cuando termine el desarrollo y sus pruebas, deberá elaborar el documento de pruebas y se la asignará a cualquiera de los otros programadores (en calidad de Probador de Software) para que realicen pruebas adicionales, con la finalidad de garantizar la calidad del entregable. La tarea pasará al estado “En Pruebas”. En este punto, el Probador revisa la historia, realiza sus escenarios de pruebas y las documenta en el mismo archivo de pruebas que fue creado por el programador de la historia.

Cabe destacar que el documento de pruebas deberá tener todos los escenarios de pruebas de acuerdo con los criterios de aceptación de la historia y podrá ser revisado por el Facilitador y/o Dueño del Producto.

En caso de que el Probador de Software detecte algún escenario de error o inconsistente, se lo comunicará al programador para que realice los ajustes necesarios y la historia volverá al estado “En Curso” y una vez que esté lista el Probador volverá a probar y finalmente moverá la historia al estado “Listo”.

La definición de “Listo” siempre será la culminación de la tarea con aprobación de un programador en calidad de Probador. En caso de que la historia requiera realizar pruebas con usuarios (lo cual se define al inicio del Sprint) el Dueño del Producto se encargará de hacer una reunión de pruebas con usuarios, con la debida anticipación, de tal manera que se realice antes de que termine el Sprint.

Puede darse el caso de que se requiera agregar una historia adicional al Sprint en curso. En este caso se debe procurar que no se vean afectadas las demás historias.

### **2.5.5 LA REUNIÓN DIARIA**

En esta reunión diaria, de duración no mayor a quince minutos, participarán todos los miembros del equipo. Cada uno explicará las actividades que realizó el día anterior y las que están desarrollando en ese momento.

El objetivo de la reunión es verificar si algún programador tiene bloqueantes que le estén impidiendo continuar con el desarrollo de sus historias. De ser el caso, el Facilitador deberá hacer la gestión necesaria para ayudarlo a resolver el problema para que se continúe con el desarrollo. La idea es que el programador se dedique a hacer su trabajo específico de análisis, programación y pruebas, mas no a estar resolviendo problemas de gestión, como dudas de alcances, problemas con el ambiente de desarrollo, etc. La comunicación temprana de los problemas que se presenten permitirá que se encuentren soluciones a tiempo, para evitar que no se concluyan las historias.

### **2.5.6 LA RETROSPECTIVA**

El último día del sprint, se realizará una reunión en la que participarán todos los integrantes del equipo, llamada Retrospectiva. Surge con la

idea de mejorar continuamente el desarrollo de los proyectos, la productividad, la velocidad y calidad de los entregables.

En esta reunión se procura hacer una revisión de los eventos presentados en el Sprint con la finalidad de mantener lo que se hizo bien, mejorar las cosas que no funcionaron bien y planear una estrategia para obtener puntos de mejoras. La Retrospectiva se basará en tres preguntas principales: ¿Qué hicimos bien en el Sprint?, ¿Qué nos está afectando? y ¿Qué mejoras vamos a implementar?. El Facilitador actúa como moderador de la reunión y será el encargado de anotar las respuestas a estas preguntas, hacer dinámicas para que todo el equipo participe y aporte con ideas relevantes de mejora y compromisos del equipo.

Con esta reunión se da por finalizado el Sprint y al día siguiente inicia uno nuevo y se repite el ciclo. En caso de que no se hayan concluido historias del Sprint actual pasarán al siguiente y en el Acta del Sprint se ubicarán los motivos por los cuales no se pudieron terminar y lo que faltaría.

### **2.5.7 PASE A PRODUCCIÓN**

El pase a producción de las historias terminadas se realizará en un máximo de tres días posteriores a la culminación del Sprint. Sin embargo, la documentación del pase (los fuentes que deben pasar a producción) se debe realizar en el transcurso del Sprint.

El pase se hará por Sprint, es decir, de todas las historias (exceptuando las que tengan alguna dependencia con otras que aún no finalizan). El Facilitador se encargará de revisar y realizar el pase a producción, pidiendo la presencia de los programadores en caso de ser necesario por cualquier eventualidad.

Las tareas por requerimientos de mantenimiento, se podrán pasar a producción en cualquier momento, una vez que se encuentren en estado "Listo".

## **2.6 DOCUMENTACIÓN**

Es necesario que cada sprint quede documentado con la finalidad de que fácilmente se pueda acceder a la información de lo que sucedió en el mismo y en caso de ser necesario podamos responder en alguna auditoria a interrogantes como: ¿Cuántas historias se entregaron?,

Cuántos puntos se hicieron?, ¿Quién hizo las pruebas?, Qué objetos se enviaron al pase a producción?, etc.

### **2.6.1 MATRIZ DE PRUEBAS**

En este documento se registran todos los escenarios de pruebas de cada historia y consta de cuatro páginas. El formato del documento se muestra en el Anexo 1.

El nombre físico del archivo se compone de la siguiente manera:

MP_	Iniciales que significan Matriz de Pruebas
SP-X_	Código de la historia
DescHistoria	Descripción de la Historia

En la primera página de documento se registran los datos generales de la historia y los componentes que se van a probar. En la segunda se ingresan los datos del cambio realizado y los resultados de las pruebas. En la tercera se ingresan los requisitos necesarios para poder realizar las pruebas y en la cuarta página se ubican las capturas de pantalla del resultado de las pruebas.

## 2.6.2 FORMULARIO DE PASE A PRODUCCIÓN

En este documento se registran los objetos que se van a enviar a producción para que el Coordinador de Desarrollo o alguien designado por él pueda realiza el pase. Por esta razón, es importante el orden en que se registren, pues ese será el orden de compilación. El formato del documento se muestra en el Anexo 2.

El nombre físico del archivo se compone de la siguiente manera:

Pase_	Documento
SprintX_	Número del Sprint
Fecha	Fecha en formato (ddmmaaaa)

El detalle del documento consta de los siguientes campos:

**Sprint o código Jira:** Se pone el número del Sprint. Se registra el código Jira cuando necesite realizarse el pase a producción de una historia específica antes de que finalice el Sprint.

**Tipo de archivo:** Se debe escoger el tipo del listado.

**Ruta de origen:** Existe un servidor Windows donde se almacenan los archivos para pases a producción. Los archivos deben estar en una carpeta con el nombre del Sprint.

Nombre Archivo: Se ubica el nombre del archivo

Destino: La ruta donde se podrán los archivos. En caso de que sea un script de base de datos se colocará el nombre de la base.

Tipo fuente: Se indica si el fuente es nuevo, modificado o se deberá eliminar.

Acción: La acción que se tomará con respecto a ese archivo. Se debe elegir un elemento de la lista.

Cuando se trate del pase del Sprint este documento debe ser unificado, es decir, en un solo documento deberán estar todos los objetos de todas las historias del Sprint.

### **2.6.3 BITÁCORA DE RETROSPECTIVA**

En este documento se registra el detalle de la reunión al final del Sprint llamada Retrospectiva, es decir, las opiniones de cada programador con respecto a lo que se hizo bien durante el Sprint, lo que se debe mejorar o lo que le está afectando al equipo y los compromisos de mejora.

El Facilitador se encarga de ingresar la información durante la reunión.

El formato del documento se muestra en el Anexo 3.

El nombre físico del archivo se compone de la siguiente manera:

Retrospectiva_	Documento
SprintX_	Número del Sprint
Fecha	Fecha en formato (ddmmaaaa)

En la primera página del documento se registra la información general de la reunión, la fecha, el número del Sprint y los participantes. En la segunda se ingresan las opiniones de los programadores, propias de la reunión y en la tercera página se indica el seguimiento que se le está dando a los compromisos de los Sprints pasados.

#### **2.6.4 ACTA DEL SPRINT**

En este documento se registra el detalle del Sprint y sus historias de usuario. Se realiza para tener a la mano un resumen de lo que sucedió en el Sprint, las historias que se entregaron, los puntos generados, fechas de pases, los motivos por los que no se entregaron las historias (si se da el caso), etc.

El nombre físico del archivo se compone de la siguiente manera:

Acta_	Documento
SprintX_	Número del Sprint

Fecha                      Fecha en formato (ddmmaaaa)

## **2.7 SOFTWARE PARA LA GESTIÓN DE LA METODOLOGÍA**

Para la implementación de la metodología surgió la necesidad de instalar algunas herramientas de software. Principalmente, se requería de un tablero virtual donde se gestionen las actividades, el estado y avances de las mismas con la finalidad de hacer el respectivo seguimiento hasta el cierre de cada una. Todo esto, de acuerdo a los roles y responsabilidades establecidos en la metodología.

Por otro lado, también se requería una herramienta que permita documentar los requerimientos o proyectos, con la finalidad de que tanto el programador como el coordinador tengan a la mano las definiciones de los alcances acordados con el usuario, las pruebas realizadas de las nuevas funcionalidades, etc. y que se se vaya actualizando de acuerdo a la evolución del proyecto.

Después de haber evaluado varias opciones se decidió incorporar las herramientas de Atlassian (Ver tabla 2.2).

**Tabla 2 - Herramientas para la gestión de la metodología**

No. Usuarios	Descripción	Costo anual
10	Licencia de Atlassian Jira Software Cloud, para la gestión del tablero de actividades	\$ 0
10	Licencia de Atlassian Confluence Cloud, para la gestión de la documentación de requerimientos y entregables	\$ 0

Fuente: Elaborado por el autor

Para la gestión del tablero de actividades y verificar el estado de las historias de usuario y tareas se incorporó Jira Software.

Se utilizó Confluence para la gestión documental de cada proyecto, es decir, todo lo relacionado al levantamiento de información, correos de respaldo, documentación de pruebas del desarrollo, aprobaciones de usuarios y archivos para el pase a producción.

### 2.7.1 CONFIGURACIÓN DE JIRA SOFTWARE

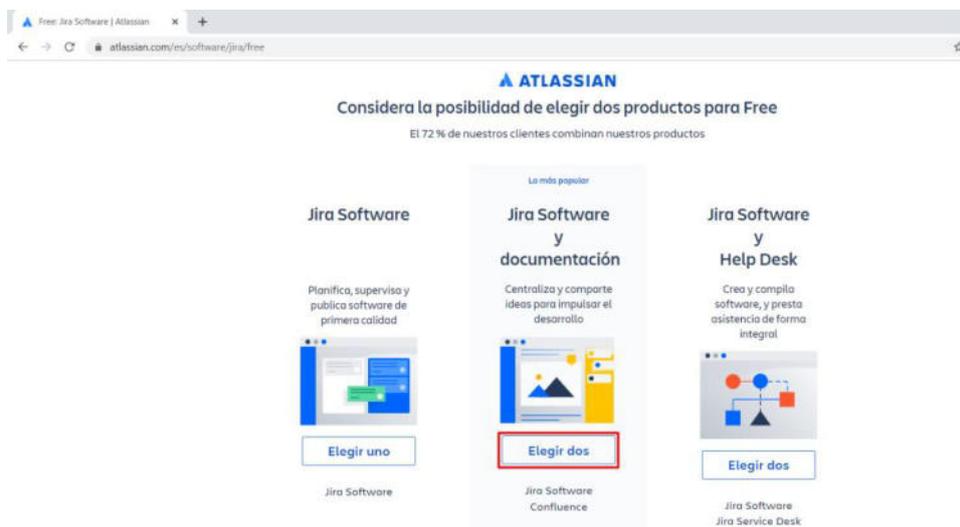
En esta sección se mostrará la configuración básica de Jira y el uso de esta herramienta. No se intenta describir el manejo de la herramienta como un manual de usuario, sino mas bien, de poner en práctica los conceptos y técnicas aplicadas en la metodología.

Si se desea profundizar en algún tema de la herramienta que no se describa en este trabajo, se deberá acceder a la documentación que se encuentra en el portal de Atlassian: <https://support.atlassian.com/jira-cloud-administration/resources>.

## 2.7.2 REGISTRO DE EMPRESA

Para empezar a trabajar con el tablero de actividades fue necesario registrar una cuenta de correo en Jira (Ver figura 2.4). Accedemos al portal <https://www.atlassian.com/es/software/jira/free> y elegimos Jira Software y documentación.

**Figura 2.4 – Acceso al portal de configuración de Jira**



Fuente: Jira Software Web

Luego nos aparecerá la opción para registrar nuestra cuenta de correo empresarial en Jira (Ver figura 2.5).

**Figura 2.5 – Registro de cuenta de correo en Jira**

Fuente: Jira Software Web

Una vez que se haya registrado la cuenta se deberá acceder al buzón del correo para aceptar la verificación y continuar (Ver figura 2.6).

**Figura 2.6 – Registro del sitio en Jira**

Fuente: Jira Software Web

En las pantallas siguientes se deberá elegir las opciones relacionadas con el Desarrollo de Software. Luego, en la pantalla de “Invita a tu Equipo”, daremos clic en Omitir, pues agregaremos las cuentas de correo del equipo de desarrollo más adelante (Ver figura 2.7).

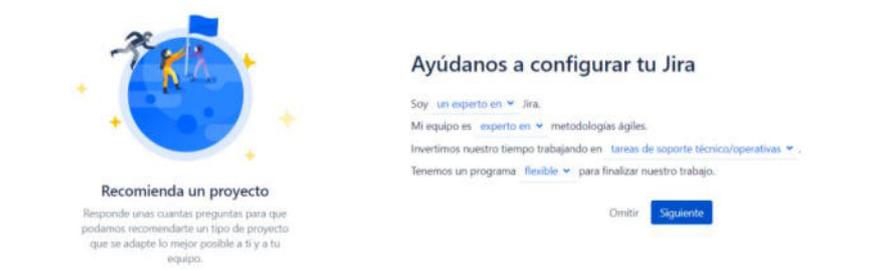
**Figura 2.7 – Registro de correos del equipo de desarrollo**



Fuente: Jira Software Web

En la siguiente pantalla (figura 2.8) se configura el tablero. Se seleccionan las opciones de experto para que no se muestre la guía de configuración.

**Figura 2.8 – Configuración general de Jira**

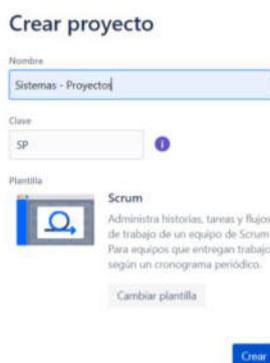


Fuente: Jira Software Web

A continuación se deberá seleccionar la plantilla Scrum y luego, se creará un “Proyecto” que hace referencia al tablero donde se colocarán

todas las historias y tareas relacionadas a los proyectos de desarrollo del área de sistemas (Ver figura 2.9).

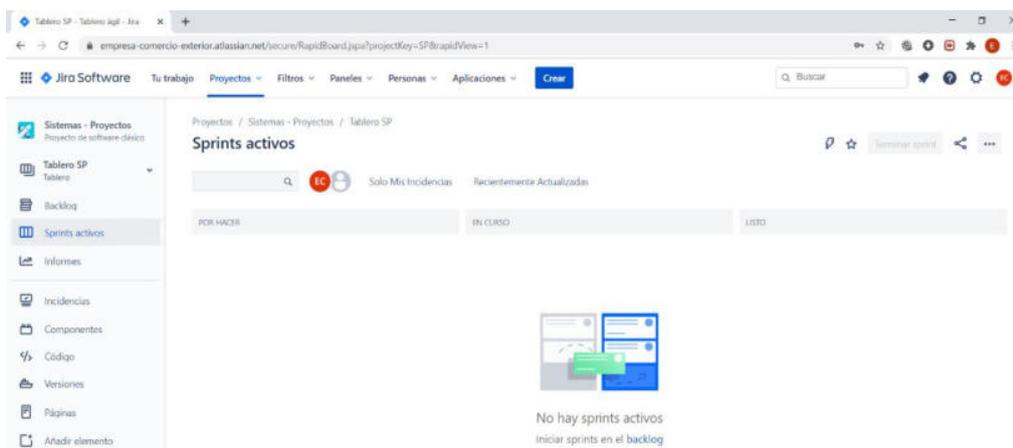
**Figura 2.9 – Crear proyecto (tablero) en Jira**



Fuente: Jira Software Web

Una vez culminada la configuración nos aparecerá una pantalla con el tablero vacío para empezar a ingresar las tareas o historias de usuario (Ver figura 2.10).

**Figura 2.10 – Tablero de Proyectos Jira**



Fuente: Jira Software Web

### 2.7.3 REGISTRO DE USUARIOS

Es momento de registrar los usuarios que van a interactuar con la herramienta Jira Software. Se deberán registrar todos los participantes del equipo, el Dueño del Producto, Facilitador y los Programadores.

Para registrar usuarios en la aplicación ir al menú Personas > Invitar a un compañero de equipo. Se deberá ingresar en el formulario la dirección de correo de cada uno de los miembros del equipo de desarrollo para que Jira les envíe una invitación. Dar clic en “Invitar” (Ver figura 2.11)

**Figura 2.11 – Invitar a las personas del equipo de desarrollo**

Invitar personas

Cada persona a la que invites tendrá acceso predeterminado a los siguientes productos. También puedes [enviar invitaciones a medida](#).

Product

Jira Software x Confluence x

Jira Core viene incluido en otros productos de Jira.

Dirección de correo electrónico

bdiazlope@comercio-externior.com.ec

Dirección de correo electrónico

cquimissol@comercio-externior.com.ec

Dirección de correo electrónico

acastillove@comercio-externior.com.ec

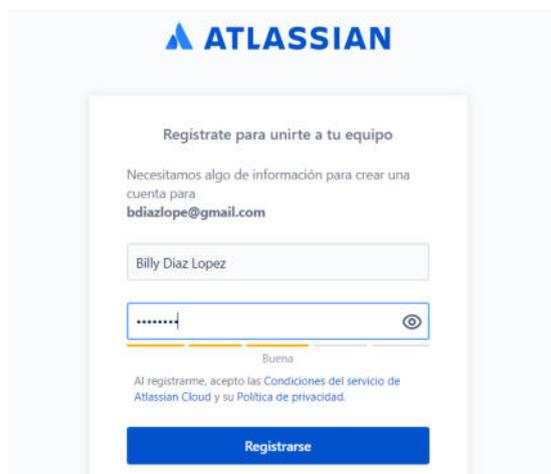
[+ Añadir más](#)

Cancelar **Invitar**

Fuente: Jira Software Web

Las personas deben acceder al correo para ver la invitación y registrarse en Jira. Esto les permitirá acceder al tablero del equipo para ver sus historias o tareas, cambiar el estado de las mismas, etc. (Ver figura 2.12).

**Figura 2.12 – Personas del equipo de desarrollo accediendo a Jira**



The image shows a registration form for Atlassian. At the top is the Atlassian logo. Below it, the text reads "Regístrate para unirte a tu equipo". Underneath, it says "Necesitamos algo de información para crear una cuenta para" followed by the email address "bdiazlope@gmail.com". There are two input fields: the first contains the name "Billy Diaz Lopez", and the second is a password field with a strength indicator below it showing "Buena" (Good) with a yellow bar. Below the password field, there is a line of text: "Al registrarme, acepto las Condiciones del servicio de Atlassian Cloud y su Política de privacidad." At the bottom of the form is a blue button labeled "Registrarse".

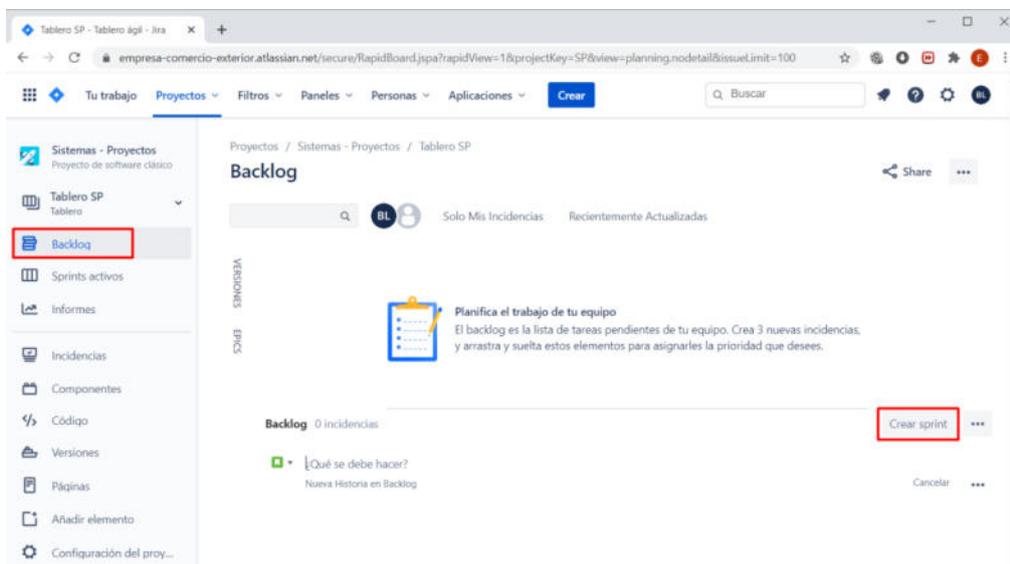
Fuente: Jira Software Web

Una vez ingresados los usuarios, es momento de ingresar las historias de usuario o tareas, según el caso.

#### **2.7.4 AGREGAR SPRINTS Y TAREAS**

A continuación se deberá crear el sprint, que como lo indicamos anteriormente, será el intervalo de tiempo que tendremos para entregar las historias o las tareas que se definan. Para agregar un sprint el Dueño del Producto deberá acceder a Jira con su usuario y clave, ubicarse en el listado de pendientes (Backlog) y dar clic en Crear Sprint (Ver figura 2.13).

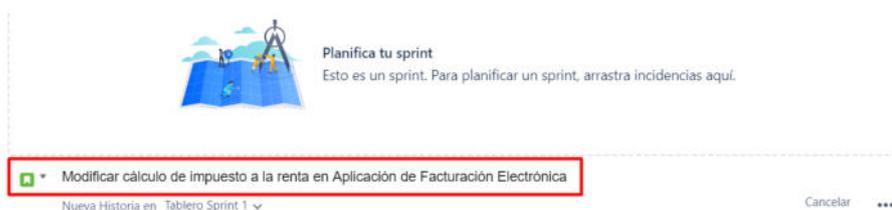
**Figura 2.13 – Crear sprint**



Fuente: Jira Software Web

Luego de haber creado el sprint se deberán crear las historias de usuario o tareas, para lo cual nos ubicamos en “Crear Incidencia”. Se debe ingresar el título y presionar la tecla enter (Ver figura 2.14).

**Figura 2.14 – Crear historia de usuario o tarea**



Fuente: Jira Software Web

La tarea ya está creada, para agregar el detalle, se da clic en el código SP que Jira genera automáticamente.

**Figura 2.15 – Ingreso del detalle de la historia de usuario**

Proyectos / Sistemas - Proyectos / SP-2

**Modificar cálculo de impuesto a la renta en Aplicación de Facturación Electrónica**

Adjuntar Crear subtarea Vincular incidencia

Descripción

**Historia de Usuario**

Como dueño del aplicativo de facturación electrónica **necesito** que se modifique el cálculo del impuesto a la renta **para** que se visualice de acuerdo a lo establecido por el SRI

**Criterios de Aceptación**

Para que la presente historia se de por aceptada deben aceptarse los siguientes criterios:

- Aplica a todos los clientes sin distinción.
- Aplica únicamente al comprobante de retención.
- Calcular el impuesto a partir del parámetro PORC\_IMPPTO\_RENTA.
- Validar que el comprobante se grabe correctamente con valor correcto

Actividad

Mostrar: Comentarios Historial Registro de trabajo

BL Añadir un comentario...

Consejo de expertos: pulsa para comentar

Tareas por hacer

Responsable: Eduardo Coronel

Informador: Billy Diaz Lopez

Etiquetas: Ninguno

Sprint: Tablero Sprint 1

Prioridad: Medium

Story Points: 3

Estimación original: 0m

Seguimiento de tiempo: Sin tiempo registrado

Epic Link: Ninguno

Componentes: Ninguno

Versiones corregidas: Ninguno

Mostrar menos

Fuente: Jira Software Web

Si se trata de una historia de usuario, en la descripción se deberá usar el formato “de historia” con sus criterios de aceptación (Ver figura 2.15). Si es una tarea, se usará el formato “Problema / Solución” indicado en las secciones anteriores.

En el campo Responsable se ubicará la persona a la cual se le asignará la historia o tarea. En caso de que no se conozca todavía el programador que va a tomarla se puede quedar con el valor Sin Asignación, por defecto.

En el campo Story Points se deberá colocar el valor de los puntos de historia de acuerdo con la complejidad de la tarea. Este valor se otorga en consenso con los programadores (Ver sección 2.4.3).

En el campo “Estimación original” se ingresará el tiempo en que se estima se culminará la historia, mientras que en el campo “Seguimiento de Tiempo” se deberá ingresar el tiempo real que tomó la historia. Estos valores los utilizaremos más tarde en las estadísticas de las estimaciones, por lo cual es importante que se ingresen datos reales.

En el campo Epic Link (opcional) se ingresará el nombre de la épica. Esto solo aplica cuando se trate de un proyecto que agrupe varias historias de usuario (Ver figura 2.3).

El Dueño del Producto en coordinación con el Facilitador ingresará las historias de usuario (Ver figura 2.16) para que, cuando se realice la Planificación, se revisen y expliquen al equipo de desarrollo y se proceda a iniciar el Sprint (Ver sección 2.4.2).

Las historias pueden ser ingresadas en cualquier momento, inclusive durante el Sprint, todo depende de su prioridad. Sin embargo, se recomienda que las historias que van al Sprint hayan sido ingresadas y planificadas con anticipación para evitar que continuamente se modifiquen o se agreguen alcances.

**Figura 2.16 – Listado de Historias Pendientes**

Proyectos / Sistemas - Proyectos / Tablero SP

Backlog

Solo Mis Incidencias Recientemente Actualizadas

Tablero Sprint 1 7 incidencias

Iniciar sprint Páginas vinculadas 0

Asignado a	Historia	Story Points
LB	Modificar cálculo de impuesto a la renta en Aplicación de Facturación Electrónica	3
LB	Notificación de creación de programaciones	8
LB	Enviar correo a cliente por pago realizado	8
LB	Pantalla de aprobación de supervisor en SAP B1	13
EC	Enviar notificación a supervisor por monto de facturación	8
EC	Modificar pantalla de ingreso de embarque internacional en App Magaya	5
LB	Agregar campos de pantalla de compras en SAPB1	5

7 incidencias Restante 0 Estimar 50

Backlog 1 incidencia

Depuración de archivos en servidor de BD SP-7

Fuente: Jira Software Web

La carga de trabajo de cada programador (figura 2.17) se podrá visualizar de manera consolidada al dar clic en el vínculo “Ver resumen de trabajo”.

**Figura 2.17 – Carga de trabajo por usuario asignado**

Backlog

Solo Mis Incidencias Recientemente Actualizadas

Tablero Sprint 1

Iniciar sprint Páginas vinculadas 0

Carga de trabajo por usuario asignado, - Tablero Sprint 1

Responsable	Incidencias	Story Points	Tiempo Restante Estimado
Sin asignar	0	0	0
LB Laura Bermula	2	21	0
EC Eduardo Coronel	3	16	0
AG Andrés Gavilanes	2	13	0
<b>Total</b>	<b>7</b>	<b>50</b>	<b>0</b>

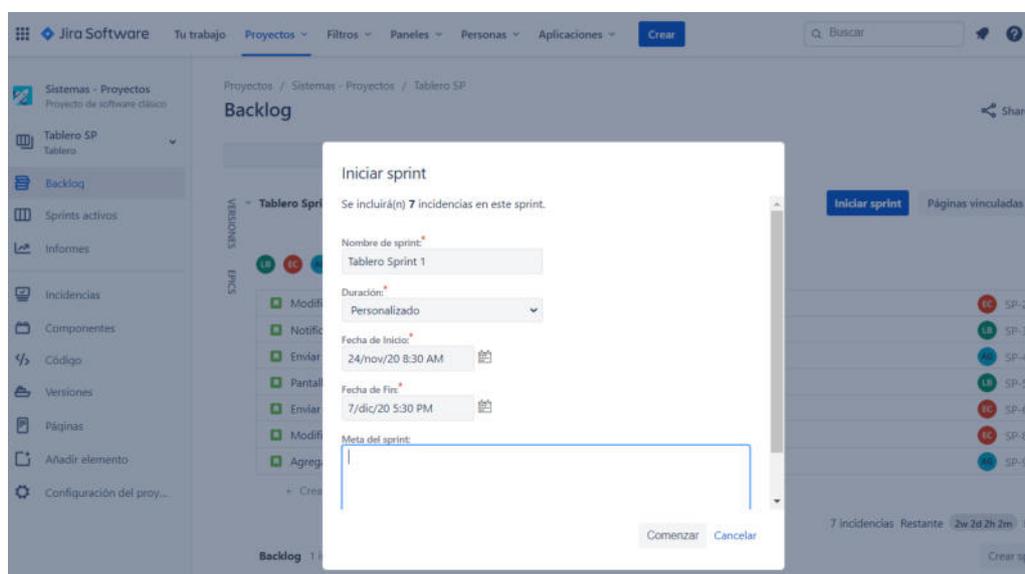
Cerrar

7 incidencias Restante 0 Estimar 50

Fuente: Jira Software Web

Para iniciar el sprint se debe dar clic en el botón “Iniciar Sprint”. Aparecerá una pantalla para configurar el tiempo de duración del sprint (Ver figura 2.18) que, de acuerdo a la metodología, será de dos semanas calendario. Finalmente dar clic en comenzar.

**Figura 2.18 – Iniciar Sprint**

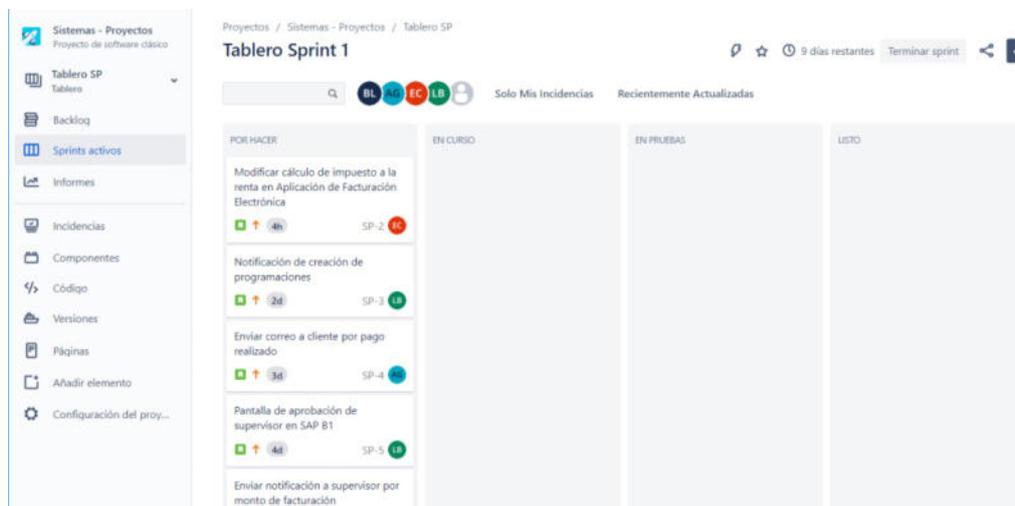


Fuente: Jira Software Web

## 2.7.5 GESTIÓN DEL TABLERO DE PROYECTOS

Ahora, todos los integrantes del equipo podrán visualizar el tablero en la opción “Sprints Activos”. En este momento todas las tareas se encuentran en la columna “Por Hacer” (Ver figura 2.19).

**Figura 2.19 – Tablero de Sprint Actual**



Fuente: Jira Software Web

Cuando el programador empiece a trabajar en una historia, ésta debe moverse a la columna “En curso”. Cuando la culmine y elabore el documento de pruebas la moverá a la columna “En Pruebas”, asignándole la tarea a un compañero para que, de acuerdo al documento pueda realizar las pruebas como un Probador. Finalmente, éste último moverá la historia a Listo. (Ver sección 2.4.4).

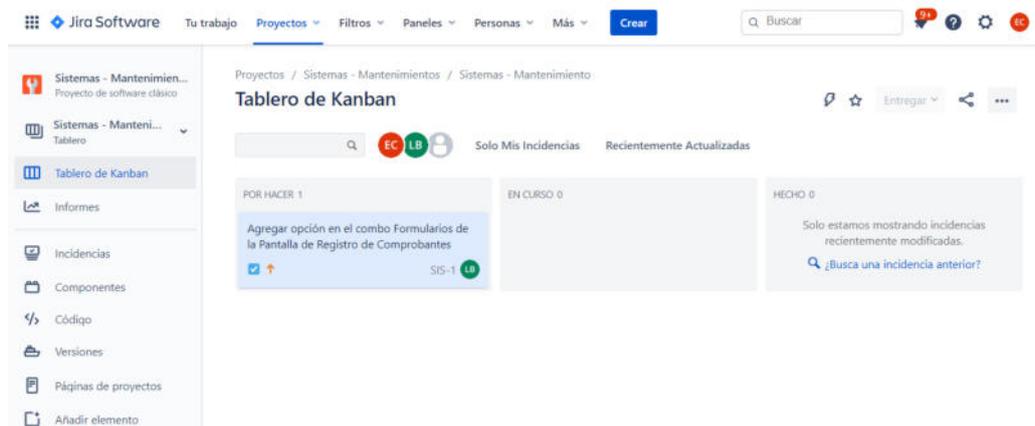
El último día del Sprint, en la reunión de la Retrospectiva, se revisará que se hayan movido las tareas a Listo y que la documentación de las pruebas esté completa. En ese momento se procederá a realizar el cierre del Sprint presionando el botón “Terminar Sprint” del tablero. Las historias que no se hayan culminado pasarán al siguiente Sprint y se llenará el acta respectiva (Ver sección 2.5.5).

## 2.7.6 GESTIÓN DEL TABLERO DE MANTENIMIENTO

Para los requerimientos de mantenimiento se creará otro tablero en la herramienta Jira Software. Para esto, nos ubicamos en el tablero actual y damos clic en “Crear Tablero”. El nuevo tablero será de Tipo Kanban (Ver figura 2.20), no es Scrum porque no va a tener Sprints, sino que a medida que vayan surgiendo las tareas de mantenimiento se van agregando y asignando a los programadores de acuerdo a la prioridad establecida por el Facilitador/Dueño del Producto.

Se deberán otorgar los permisos para que los demás usuarios puedan ver este tablero y gestionar el estado de las tareas.

**Figura 2.20 – Tablero de Mantenimiento**



Fuente: Jira Software Web

Se presiona el botón “Crear” para registrar una nueva tarea. Al igual que las historias se le agrega el título, la descripción en formato

Problema/Solución, el responsable, los puntos de historia (a pesar de no ser una historia se requiere medir la complejidad de la tarea) y la prioridad (Ver figura 2.21).

**Figura 2.21 – Tarea de Mantenimiento**

The screenshot shows a Jira task card with the following details:

- Issue ID:** SIS-1
- Actions:** Adjuntar, Crear subtarea, Vincular incidencia
- Descripción:**
  - Problema:** Se necesita que el usuario pueda elegir la opción de "Liquidación de Compras" en el combo Formularios de la Pantalla de Registro de Comprobantes
  - Solución:** Agregar la opción Liquidación de Compras en el parámetro FOR01-RC de la tabla FR\_PARAMETROS en la BD.
- Metadata:**
  - Tareas por hacer:** 1
  - Responsable:** Laura Bermula (LB)
  - Informador:** Eduardo Coronel (EC)
  - Story Points:** Ninguno
  - Etiquetas:** Ninguno
  - Prioridad:** Medium
  - Estimación original:** 0m

Fuente: Jira Software Web

El programador / persona de mesa de ayuda toma la tarea asignada y la mueve a "En Curso" para empezar a trabajar en ella. Una vez culminada se la asigna al Facilitador para que la revise y la mueva a la columna "Hecho". Por tratarse de tareas pequeñas y de baja complejidad no se requieren realizar documentos ni pruebas cruzadas.

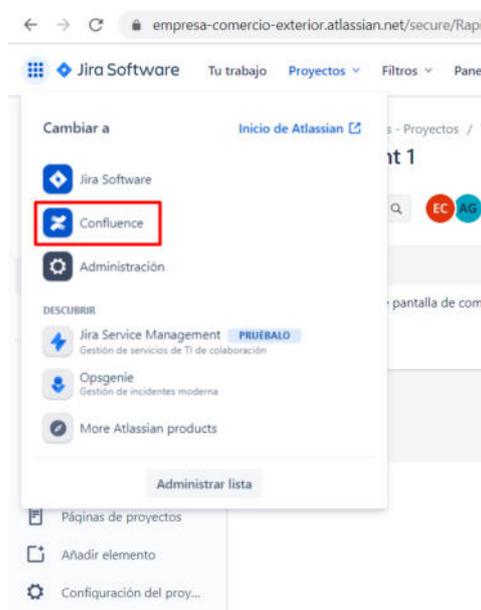
El pase a producción se puede realizar en cualquier momento, después de elaborar el documento de pase a producción (Ver sección 2.6.2).

## 2.7.7 GESTIÓN DE LA DOCUMENTACIÓN EN CONFLUENCE

El aplicativo Confluence es un gestor documental que funciona de forma similar a una wiki, el cual permite a los usuarios crear y editar contenido de manera colaborativa. Aquí se subirán todos los formularios, correos y demás documentación referente a cada historia del Sprint y de cada tarea del tablero de mantenimiento.

Para acceder desde Jira se puede dar clic a la izquierda de la palabra Jira Software y seleccionar Confluence (Ver figura 2.22).

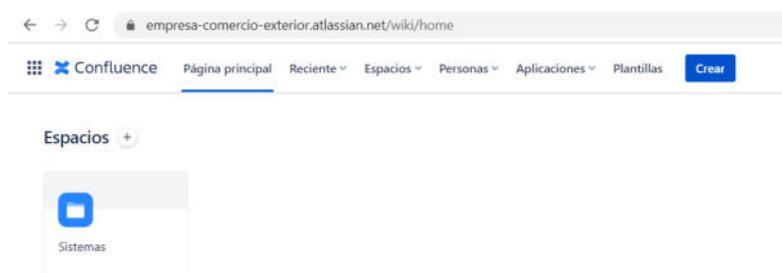
**Figura 2.22 – Acceso a Confluence**



Fuente: Confluence Web

Lo primero que se visualiza es el espacio de trabajo “Sistemas” (Ver figura 2.23). Al dar clic en el mismo aparecerán las páginas llamadas Tablero Proyectos y Tablero Mantenimiento.

**Figura 2.23 – Espacios de Trabajo**



Fuente: Confluence Web

La página Tablero Proyectos (figura 2.24) contiene toda la documentación referente al Sprint. Estará formada por una tabla con filas y columnas. La primera fila corresponde a la cabecera de la tabla e indica lo que debe colocarse en cada columna. En la columna “Historia o tarea” se ingresarán los Sprints y cada una de las historias creadas en el mismo.

En la columna “Correos de soporte” se ingresarán todos los correos con los requerimientos, alcances, bitácoras de reunión, etc de cada historia según corresponda.

**Figura 2.24 – Documentación del Tablero de Proyectos**

Historia o Tarea	Correos de soporte	Matriz de Pruebas / Acta Sprint	Aprobación usuario funcional / Bitácora Retrospectiva	Pase a producción
SPRINT 1				
<a href="#">SP-2: Modificar cálculo de impuesto a la renta en Aplicación de Facturación Electrónica INCURSO</a>				

Fuente: Confluence Web

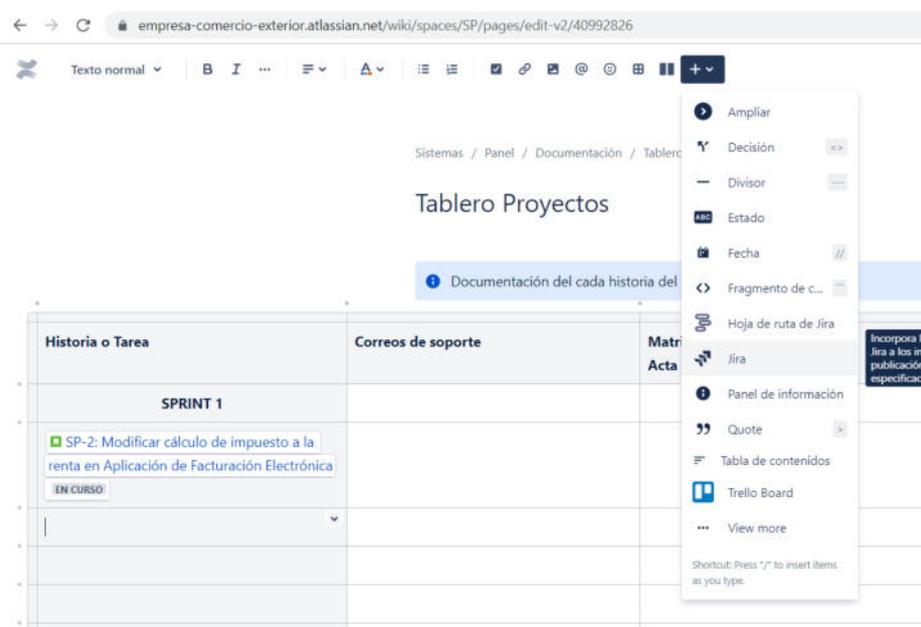
En la columna “Matriz de pruebas / Acta Sprint” se ingresará la Matriz de Pruebas (si es una fila de historia) o el Acta del Sprint (si es la fila del Sprint).

En la columna “Aprobación de usuario funcional / Bitácora Retrospectiva” se ingresará el correo de aprobación de las pruebas del usuario (cuando aplique) o el documento Bitácora de la Retrospectiva (si es la fila del Sprint).

En la columna “Pase a Producción” se ingresará el documento de pase a producción. Por lo general sólo se coloca en la fila de la historia, a menos que se haya hecho un pase de alguna historia en específico.

Para agregar los documentos se deberá editar el tablero dando clic en el ícono de edición. Las historias se deberá agregar como links de Historia Jira, buscando por el código de la historia (Ver figura 2.25).

**Figura 2.25 – Agregar link de Historia Jira**



Fuente: Confluence Web

Los documentos serán agregados presionando el ícono de “agregar imagen o archivo”. Aparecerá una pantalla para buscar el archivo en una ubicación local para seleccionar el mismo.

Es importante que el nombre de cada archivo se encuentre con el formato mostrado en la sección 2.6. No pueden existir nombres de archivos repetidos, porque Confluence los reemplaza.

**Figura 2.26 – Documentación del Sprint y sus Historias**

Historia o Tarea	Correos de soporte	Matriz de Pruebas / Acta Sprint	Aprobación usuario funcional / Bitácora Retrospectiva	Pase a producción
<p><b>SPRINT 1</b></p> <p>SP-2: Modificar cálculo de impuesto a la renta en Aplicación de Facturación Electrónica <b>RECORRIDO</b></p> <p>SP-3: Notificación de creación de programaciones <b>INPROGRESO</b></p>	<p>Notificaciones de programaciones.mcg 53 KB</p>	<p>Acta_Sprint_30112020.xlsx 10 KB</p>		<p>Pase_Sprint_30112020.xlsx 12 KB</p>

Fuente: Confluence Web

Una vez que se ha terminado de subir la documentación se debe dar clic en Publicar para que los cambios puedan ser visualizados por los demás miembros del equipo (Ver figura 2.26).

Cada programador deberá ingresar la documentación que corresponda a sus historias. Subir el Acta del Sprint y la bitácora de la Retrospectiva será tarea del Facilitador. El pase a producción unificado lo puede agregar cualquier programador con la ayuda y supervisión de los demás.

El tablero de mantenimiento tendrá solo tres columnas. En la columna “Tarea” se coloca el link de acceso a la Tarea Jira, igual que en el tablero anterior. En la columna “Correos de soporte” se ingresarán todos los correos con los requerimientos, alcances, bitácoras de reunión, etc.

correspondientes a la tarea y en la columna “Pase a Producción” se ingresará el documento de pase a producción respectivo.

Al igual que el anterior, para agregar los documentos se deberá editar el tablero dando clic en el ícono de edición y una vez que se hayan subido todos los archivos se debe dar clic en el botón Publicar (Ver figura 2.27).

**Figura 2.27 – Documentación del Tablero de Mantenimiento**

Sistemas / Panel / Documentación / Tablero Mantenimiento

Tablero Mantenimiento

Documentación del cada historia del tablero de mantenimiento.

Historia o Tarea	Correos de soporte	Pase a Producción
<input checked="" type="checkbox"/> SIS-1: Agregar opción en el combo Formularios de la Pantalla de Registro de Comprobantes <b>TAREAS POR HACER</b>		

Fuente: Confluence Web

En caso de que se requiera algún documento de cualquier tablero solo se debe dar clic en el archivo que se requiera y automáticamente se descargará en el equipo local.

## 2.8 PLAN DE IMPLEMENTACIÓN

Para poder ejecutar este proyecto de Implementación de la metodología se requiere capacitar al equipo de desarrollo en dos ámbitos:

1. Explicar los conceptos de la metodología indicados en este trabajo.

Tiempo aproximado: 10 horas

2. Capacitar en el uso de Jira y Confluence

Tiempo aproximado: 20 horas

Luego de esta capacitación será conveniente realizar el primer Sprint con la finalidad de poner en práctica los conocimientos y familiarizarse con la metodología.

## **CAPÍTULO 3**

### **EVALUACIÓN DE RESULTADOS**

#### **3.1 MONITOREO Y MEDICIONES DE LA IMPLEMENTACIÓN**

En los primeros Sprints el equipo manifestó sus opiniones, acuerdos y desacuerdos con respecto a la manera en que se estaba llevando a cabo la metodología. Surgieron muchos cuestionamientos por cómo se realizaban las actividades, la documentación, las reuniones, etc.

Además, se realizaron algunos cambios en las definiciones, que llevó por ejemplo a modificar la definición de “Listo” de cada historia.

Es importante destacar que este comportamiento es muy normal y ventajosamente la metodología es bastante flexible en este aspecto, lo principal es mantener siempre la comunicación en el equipo y establecer acuerdos en consenso. Aquí es donde se destaca la labor del Facilitador, para captar las ideas del equipo, coordinar las mejoras y monitorear que se estén cumpliendo los objetivos.

Otro aspecto importante es que constantemente se deben realizar mediciones para verificar que los cambios que se realizan surtan el efecto esperado o, caso contrario, hacer los ajustes necesarios. Existen una gran cantidad de indicadores y mediciones que se pueden obtener con la metodología, de las cuales destacamos la productividad, efectividad en la planificación, porcentaje de mantenimiento por sprint y calidad.

Cuando culmina el Sprint, el Dueño del Producto elabora un resumen de los resultados basados en los puntos de historia, es decir, la cantidad de puntos que se planificaron, se comprometieron y se realizaron en los tableros de proyectos y mantenimiento. Además, en caso de que se

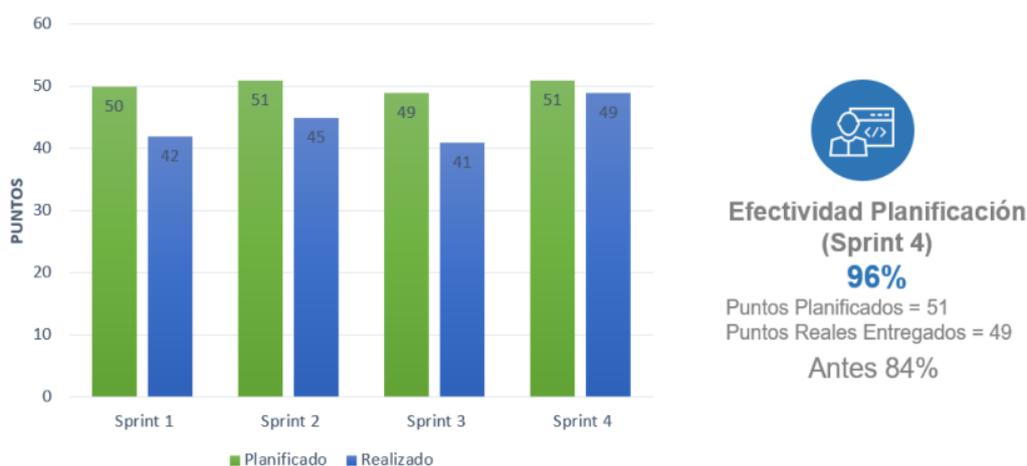
reporten incidentes en producción referentes a alguna historia del Sprint, deberán registrarse para realizar las mediciones de calidad.

La productividad puede variar de un Sprint a otro por varios factores, siendo el ausentismo el principal causante, ya sea por vacaciones, descanso médico, etc. Es importante dejar establecido en el Acta de Sprint las novedades que se presentaron y justificar de alguna manera el rendimiento del equipo.

### 3.2 ANÁLISIS COMPARATIVO

Para realizar un análisis comparativo evolutivo se tomaron en cuenta las mediciones por puntos de historia de los últimos cuatro Sprints.

**Figura 3.1 – Efectividad en la Planificación**

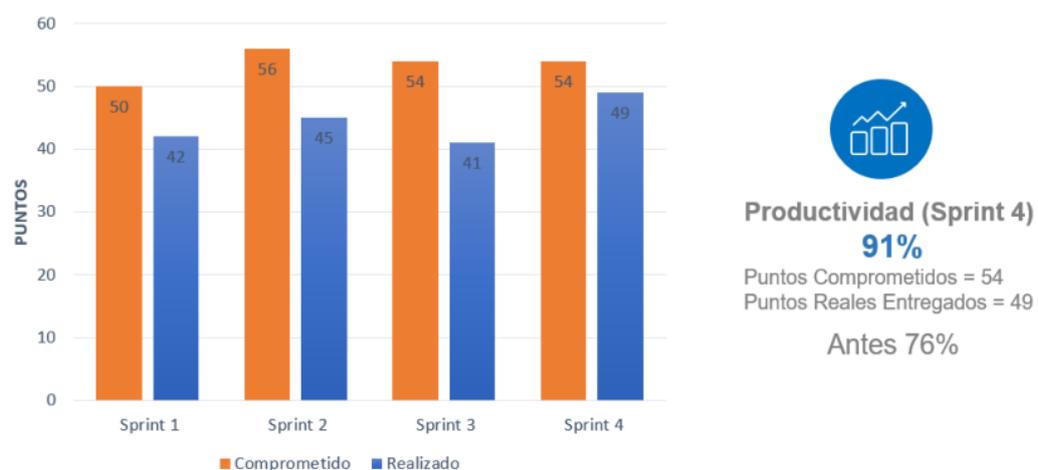


Fuente: Elaborado por el autor

La figura 3.1 muestra la efectividad de la planificación, es decir, los puntos planificados vs. los puntos que se entregaron al final del Sprint, de las historias terminadas. Lo importante de esta gráfica es que nos permite determinar qué tan efectivo está siendo el equipo de desarrollo con sus estimaciones medidas en puntos de historia (complejidad de la tarea).

Una gráfica similar a esta sería la de Productividad en Horas, donde se utilizarían las horas de estimación original vs las horas de seguimiento de tiempo que se registran en cada historia de Jira (Ver Figura 2.15 de la sección 2.7.4). Por esta razón es importante que se registre información real en estos campos, ya que de esta manera se podrán realizar mediciones más precisas.

**Figura 3.2 – Productividad**

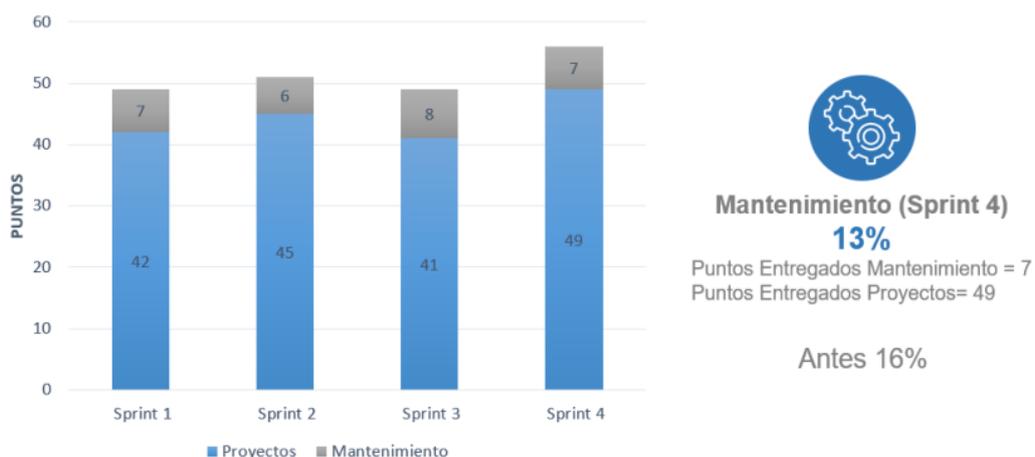


Fuente: Elaborado por el autor

En la figura 3.2 se muestra la productividad del sprint, es decir, la cantidad de puntos de las historias comprometidas vs. la cantidad de puntos de las historias realizadas. Esta gráfica es importante para verificar en qué porcentaje el equipo cumple con sus compromisos. En esta gráfica también se puede medir la velocidad del equipo, la cual sería el promedio de la cantidad de puntos realizados en los cuatro últimos sprints, para este caso sería de 44 puntos.

A diferencia de la figura 3.1 donde se mostraba lo planificado vs. lo realizado, en esta gráfica se encuentran las historias que fueron agregadas durante el Sprint, lo cual es un indicador que determina qué tan eficiente es el equipo en la planificación. En el Sprint 4 observamos que se agregaron tres puntos, puesto que se planificaron 51 puntos y se comprometieron 54, lo cual indica que posiblemente se crearon historias nuevas durante el Sprint o se cambiaron historias planificadas por historias de la lista de pendientes.

En la figura 3.3 se muestra el mantenimiento durante el Sprint, es decir, la cantidad de puntos realizados en el tablero de mantenimiento durante el tiempo que dura el Sprint. Esta gráfica es importante cuando se observa una disminución en la velocidad promedio del equipo. Por ejemplo, en el Sprint 3 solo se entregaron 41 puntos pero fue el Sprint que más entregó puntos de mantenimiento.

**Figura 3.3 – Mantenimiento**

Fuente: Elaborado por el autor

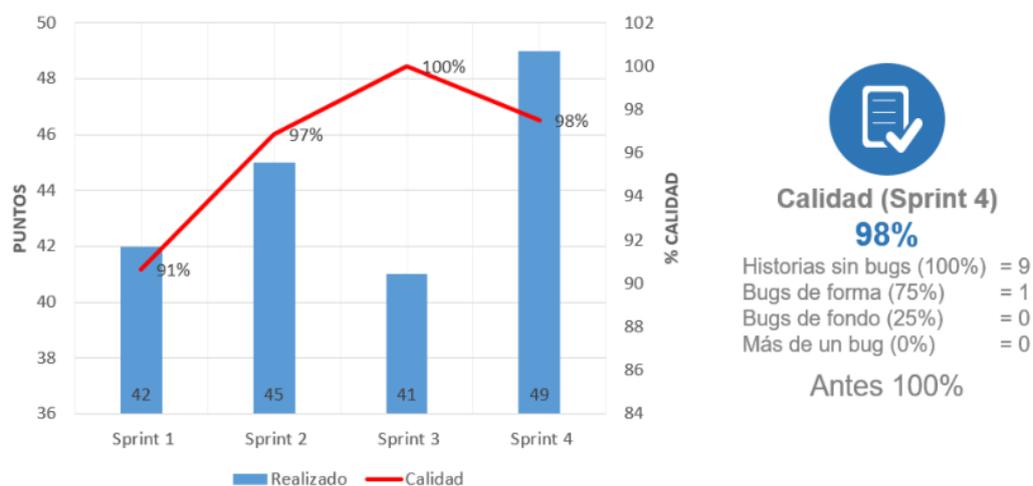
En la figura 3.4 se muestra la calidad del Sprint una vez que está en producción. Esta gráfica es importante para verificar que se entregue al usuario final un producto de calidad.

Cabe recordar que el pase se realiza de dos a tres días después de que finaliza el Sprint, por tanto, esta gráfica puede actualizarse hasta antes de culminar el Sprint en curso y validar que no se presenten errores en producción.

Las mediciones de esta gráfica se calculan en función de la cantidad de errores (bugs) reportados por el usuario. Inicialmente todas las historias que se pasan a producción van a tener una calidad del 100%. Si se presentan errores de forma, la historia baja su calidad al 75%. Si se presentan errores de fondo baja al 25%. Y si ocurre más de un error en

la misma historia su calidad baja al 0%. Finalmente, se suman y se promedian los porcentajes de todas las historias para obtener la calidad del Sprint.

**Figura 3.4 – Calidad**



Fuente: Elaborado por el autor

### 3.3 BENEFICIOS DE LA METODOLOGÍA

La implementación de la metodología otorgó al equipo de desarrollo de sistemas los siguiente beneficios:

- Generar entregables quincenales que aporten valor al usuario.
- Mejorar la calidad de las aplicaciones entregadas al usuario final.
- Organizar la documentación y disponer de una aplicación donde se tenga siempre a la mano.

- Mejorar las definiciones, alcances y priorización de cada historia o tarea.
- Conocer el estado real de cada tarea y el tiempo invertido en el desarrollo de la misma.
- Estimar de mejor manera las tareas y en función de la velocidad del equipo proyectar tiempos de entrega de proyectos grandes.
- Tener mayor comunicación, poder exponer los problemas a tiempo y trabajar en equipo para eliminar los bloqueantes.
- Realizar mediciones periódicas y poder tomar decisiones en función de los resultados.

## **CONCLUSIONES Y RECOMENDACIONES**

Una vez culminada la implementación de la metodología se concluye que:

1. Se detallan y especifican con mejor claridad las tareas y proyectos, lo cual permite al programador entender de mejor manera los alcances y disponer de una reunión "Planificación" para preguntar y despejar dudas.
2. La incorporación de puntos de historia permitió que el equipo mejore sus estimaciones, ya que se evalúa en función de la complejidad y no del tiempo.
3. Se cuenta con mayor seguimiento de las tareas y se conoce el estado real de cada una.

4. Se puede dividir el trabajo y priorizar las tareas de acuerdo a la necesidad.
5. Con la incorporación de las pruebas cruzadas se mejoró la calidad de los entregables y se minimizan errores en producción.
6. La documentación del Sprint está siempre a la mano y actualizada para la revisión y conocimiento del equipo.

Adicionalmente, se recomienda:

1. Realizar el seguimiento diario a las actividades con la finalidad de eliminar los bloqueantes a tiempo
2. Coordinar las pruebas con antelación. Si son pruebas cruzadas indicarle anticipadamente al programador para que reserve el tiempo para esta actividad. Si son usuarios enviar una convocatoria de reunión desde los primeros días del Sprint.
3. Realizar la Retrospectiva y procurar cumplir los compromisos adquiridos en la reunión, con la finalidad de mejorar continuamente y evitar bloqueantes.
4. Reunirse el Facilitador y el Dueño del Producto antes de que inicie el Sprint para afinar el listado de pendientes y evitar perder tiempo redactando o corrigiendo el texto de las historias en la Planificación.

5. Realizar las mediciones después de cada Sprint y mostrarle los indicadores al equipo en las Retrospectivas a fin de que se puedan analizar el cumplimiento de los objetivos o establecer correctivos según el caso.

## BIBLIOGRAFÍA

- [1] Schwaber, K., & Sutherland, J. (2020). *La Guía de Scrum*. Recuperado diciembre de 2020, de <https://www.scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>
- [2] Atlassian (2020). *Licencias Jira Software*. Recuperado diciembre de 2020, de <https://www.atlassian.com/es/software/jira/pricing>
- [3] Atlassian (2020). *Guía de Jira Software*. Recuperado diciembre de 2020, de <https://www.atlassian.com/es/software/jira/guides>
- [4] Atlassian (2020). *Guía de Confluence Software*. Recuperado diciembre de 2020, de <https://www.atlassian.com/es/software/confluence/guides>
- [5] Martel, A. (2015). *Gestión práctica de proyectos con Scrum: Desarrollo de software ágil para el Facilitador*.
- [6] Salvat, J. (2009). *Métricas de iteración*. Recuperado diciembre de 2020, de <https://proyectosagiles.org/2009/11/08/metricas-iteracion-scrum-sprint-metrics/>

## ANEXOS

### ANEXO 1. MATRIZ DE PRUEBAS



MP\_SP-2\_ModCalcu  
loImptoRenta.xlsx

### ANEXO 2. FORMULARIO DE PASE A PRODUCCIÓN



Pase\_Sprint1\_30112  
020.xlsx

### ANEXO 3. BITÁCORA DE RETROSPECTIVA



Retrospective\_Sprin  
t1\_30112020.xlsx

## ANEXO 4. ACTA DEL SPRINT



Acta\_Sprint1\_30112  
020.xlsx

## GLOSARIO

**Scrum** es un marco de trabajo que se utiliza para trabajar colaborativamente en el desarrollo de un proyecto para lo cual se divide el trabajo en partes que al integrarse formen la solución completa.

**Sprint** son los ciclos o iteraciones que se crean en la metodología Scrum. Tienen un tiempo relativamente corto (de una semana hasta un mes) durante el cual se genera un producto o entregable.

**Historias de usuario** se definen como los requerimientos del cliente expresados como una necesidad que debe satisfacer el equipo de desarrollo de software.

**Backlog** es un listado de tareas o historias pendientes de realizar y que se van desarrollando según su prioridad.