



## **ANÁLISIS Y DESARROLLO DE SOFTWARE**

### **ENTREGABLES DEL PROYECTO BELLA Y VALE**

**Ficha: 2627092**

*Documento identificando la metodología para el proyecto de desarrollo de software.*

**GA1-220501093-AA1-EV04.**

**Proyecto de Software**



Aprendices:

**EQUIPO DE TRABAJO**

**Carlos Téllez, Jonnathan Reyes, Miguel Guevara, Santiago Muentes y Yuly Toro**

Profesor:

**NICOLAS SILDARRIAGA**

Servicio Nacional de Aprendizaje  
SENA

Junio de 2024

Jamundí 12 de junio de 2024

Respetado profesor

Nicolas Saldarriaga  
Instructor Vocero  
Ficha: 2627092

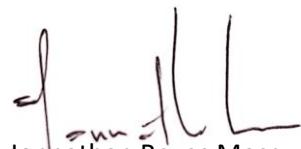
Cordial saludo,

Asunto remisión: GA1-220501093-AA1-EV04.

De manera atenta me permito presentar el entregable de la evidencia Documento identificando la metodología para el proyecto de desarrollo de software.

GA1-220501093-AA1-EV04.

Atentamente,



Jonnathan Reyes Mosquera

CC. 1.061.532.705

Cel. 3146140320

Email: [jonnathanreys@gmail.com](mailto:jonnathanreys@gmail.com)

# METODOLOGÍA PARA EL DESARROLLO DEL SOFTWARE DE BELLA Y VALE

## 1. INTRODUCCIÓN

Consideramos importante mencionar que el desarrollo de software es un proceso complejo en el que intervienen múltiples partes interesadas, cada una con sus propios objetivos, requisitos y desafíos. Los ingenieros, tecnólogos y técnicos de software deben equilibrar muchos factores que compiten entre sí para producir un código de alta calidad en una cantidad práctica de tiempo y presupuesto. Una buena metodología de desarrollo de software permite a los desarrolladores crear programas de forma más rápida, más económica y con menos riesgo de fracaso. También facilitará a otros desarrolladores de software la lectura de su código si está escrito de acuerdo con las mejores prácticas. Conocer las diferentes metodologías de desarrollo de software resulta indispensable, con los pros y los contras de cada una de ellas, para poder decidir qué enfoque metodológico es el más adecuado para cada proyecto.

La metodología de desarrollo de software en ingeniería de software, es un **framework**, o sea, un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Un framework para metodología de desarrollo de software consiste en ser:

- *Una filosofía de desarrollo de programas de computación con el enfoque del proceso de desarrollo de software*
- *Herramientas, modelos y métodos para asistir al proceso de desarrollo de software*

Estos frameworks se asocian habitualmente con organizaciones que desarrollan, apoyan el uso y promueven la metodología.

Las metodologías de desarrollo de software tienen como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelado de sistemas que permitan desarrollar software de calidad, incluyendo heurísticas o métodos para solucionar problemas de forma práctica de construcción y criterios de comparación de modelos de sistemas.

La Metodología de Desarrollo de Software a ser usada en nuestro proyecto BELLA y VALE es el marco de trabajo que nos permite estructurar, planear y mantener el control del proceso de desarrollo de este sistema de información. Consideramos que, al implementar la metodología de desarrollo de software de manera correcta, determinada, posibilita que el plan proporcione superiores estimaciones, entregue un sistema estable, mantenga al cliente del software reportado, el cual podrá observar una comprensión clara del trabajo elaborado e identifique los esfuerzos previamente, lo cual posibilitara el tiempo suficiente para hacer ajustes.

La metodología a emplear en el desarrollo del software será una manera de gestionar el plan de desarrollo de proyecto de nuestro interés. Un procedimiento de desarrollo de proyecto principalmente aborda preguntas como son la selección de propiedades para incluirlas en la versión actual, en qué momento se lanzará el software, quién labora en qué y qué pruebas se hacen.

La elección de la metodología de desarrollo de software puede determinar efectivamente el éxito o el fracaso de un proyecto. Hoy en día, las mejores metodologías de desarrollo se basan en principios y métodos ágiles. Cualquier metodología de desarrollo de software ágil sigue principios de desarrollo ajustados que están diseñados para maximizar la eficiencia, el rendimiento y eliminar el desperdicio.

Consideramos que el seguir una metodología bien determinada, posibilita que los equipos de desarrollo hagan sistemas estables, garanticen que los consumidores se encuentren informados, tengan una comprensión clara de la labor en cuestión, ofrezcan superiores estimaciones e identifiquen probables problemas en una fase temprana del plan.

## 2. METODOLOGÍA DE DESARROLLO EMPLEADA PARA BELLA Y VALE

Para administrar un proyecto de manera eficiente, nuestro equipo de desarrollo tuvo que elegir la metodología de desarrollo de software que funcionará mejor para el proyecto en mención. Tuvimos en cuenta que todas las metodologías tienen diferentes fortalezas y debilidades y existen por diferentes razones.

En ese momento nos preguntamos cual es la metodología más conveniente para nuestro proyecto, encontrando que no hay una respuesta única a esta pregunta, ya que la elección de la mejor metodología depende de diversos factores, como la naturaleza del proyecto, los requisitos del cliente, el tamaño del equipo y la cultura organizacional. Por lo que fue importante evaluar las necesidades y características de nuestro proyecto antes de seleccionar una metodología.

Nuestra investigación nos llevó a encontrar que los equipos de desarrollo utilizan la metodología de desarrollo ágil para minimizar el riesgo (como errores, sobrecostos y requisitos cambiantes) al agregar nuevas funciones. Es una de varias metodologías de desarrollo de software basadas en el desarrollo **iterativo e incremental**, en contraposición a las metodologías tradicionales de desarrollo de software lineal o cascada.

Aclaramos que, si estamos pensando en implementar **Scrum** en nuestro proceso de desarrollo ágil, hay algunas cosas que debemos saber. En primer lugar, Scrum y Agile no son lo mismo. Lo aclaramos así:

**Agile:** Se refiere a un conjunto de valores y metodologías que los desarrolladores pueden utilizar para crear un mejor software.

**Scrum:** Es una forma específica de implementar la agilidad. Además, la metodología Scrum proporciona un marco de trabajo sobre cómo debe trabajar un equipo. Está diseñado para eliminar los errores comunes del ciclo de vida del desarrollo de software, como que los ingenieros, tecnólogos y técnicos escriban demasiado código antes de comprobar si funciona.

*En resumen, Scrum es una metodología ágil que permite a los equipos colaborar y realizar un trabajo de alto impacto, entregando productos de manera eficiente y creativa con el máximo valor para el cliente.*

En todos los métodos ágiles, los equipos desarrollan el software en iteraciones que contienen mini incrementos de la nueva funcionalidad. Comenzamos adoptando marcos de trabajo ágiles con el **Scrum**, estableciendo ciclos de trabajo cortos, promoviendo la transparencia y fomentando la comunicación constante. También cada uno de nosotros investigó y se capacitó aportando cada uno al equipo de trabajo las mejores prácticas y herramientas ágiles.

La **comunicación** en esta metodología desempeña un papel crucial en la obtención de resultados relevantes. La interacción constante con los clientes y el equipo de desarrollo permite comprender mejor los requisitos y expectativas, promoviendo una mayor satisfacción del cliente. Además, la comunicación efectiva dentro del equipo de desarrollo garantiza la colaboración, la alineación y el logro de los objetivos comunes.

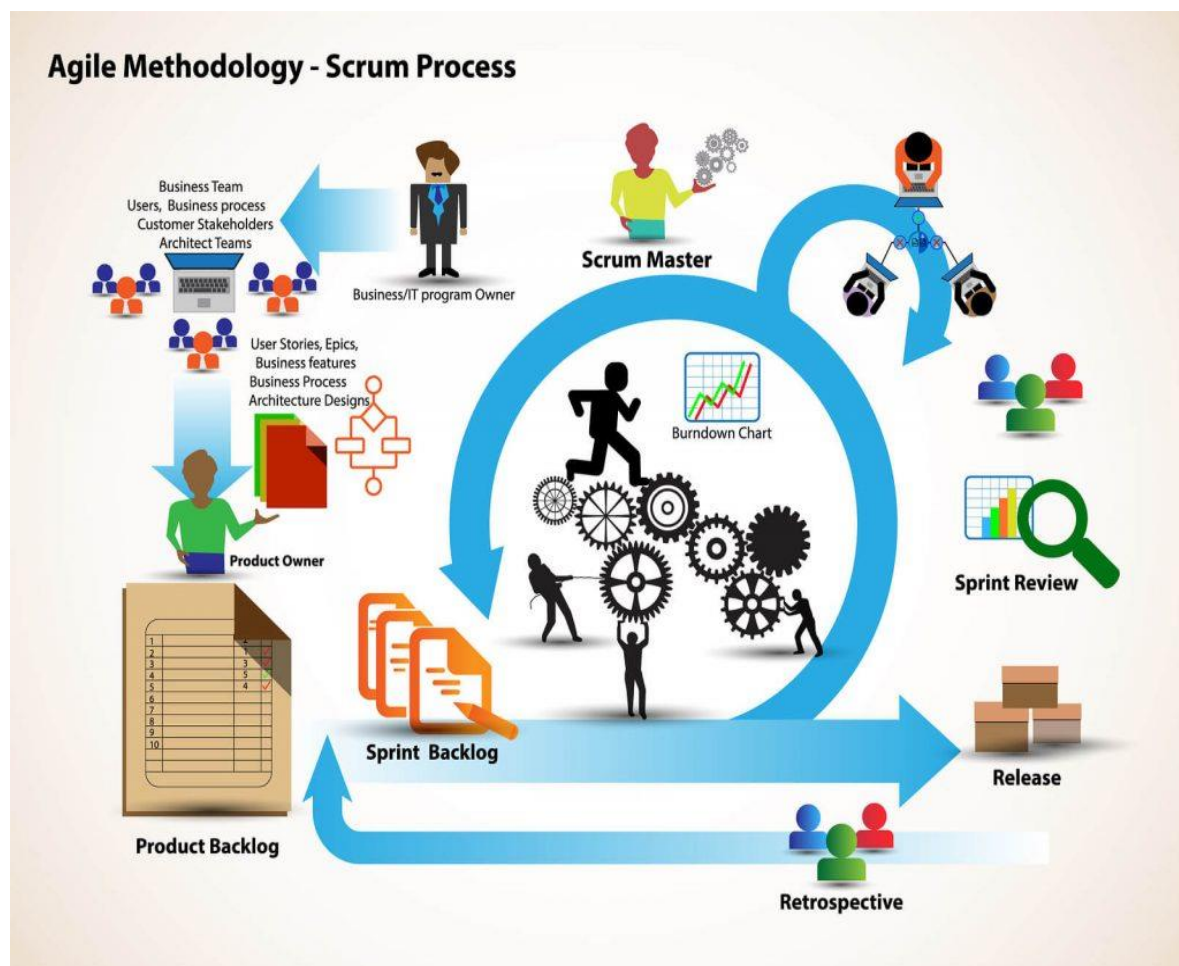
Hay muchas formas diferentes del método de desarrollo ágil, que incluyen scrum, crystal, programación extrema (XP) y desarrollo basado en características (FDD).

Nosotros hemos optado por la metodología de desarrollo de software **ágil** porque es uno de los mejores enfoques de desarrollo de software que se utiliza actualmente para diseñar un proceso de gestión de software disciplinado y porque también permite algunas alteraciones frecuentes en el proyecto de desarrollo. Decidimos por **Scrum** la cual es una metodología ágil que se enfoca en la entrega incremental y la adaptación continua. Es iterativo e incremental. Se basa en ciclos de trabajo cortos llamados "sprints" y promueve la transparencia, la responsabilidad y la autoorganización del equipo. Scrum es ampliamente utilizado en proyectos de desarrollo de software por empresas muy reconocidas a nivel mundial.

En cada sprint, el equipo de desarrollo se organizó en torno al objetivo específico y trabajó en colaboración para lograrlo. Se realizaron reuniones diarias de seguimiento, conocidas como "*stand-ups*" o sea reuniones cortas, en las que los miembros del equipo compartieron su progreso y planificaron las tareas para el día siguiente.

Nuestra aplicación del Scrum también se basó en la inspección y adaptación continua. Al final de cada sprint, llevamos a cabo una reunión de revisión en la que se muestra el trabajo realizado y se recopiló el *feedback*, la retroalimentación del cliente, para este caso pensamos en nuestros instructores evaluativos. Esto permitió ajustar y mejorar el producto en base a las necesidades y expectativas del cliente.

Una característica destacada de Scrum fue el uso de tableros visuales, como el famoso "*Scrum board*" o tablero virtual. En este tablero se representaron las tareas pendientes, en progreso y completadas, lo que permitió tener una visión clara del estado del proyecto en todo momento.



Nosotros desarrollamos el Scrum a través de unas herramientas, un flujo de trabajo y unos roles que se encargaron de diferentes partes del proyecto para que todo saliera bien.

El flujo de trabajo en el scrum practicado por el equipo de ByV se desarrolló en las siguientes fases:

## PRIMERA FASE DE SCRUM: *Inicio*

En la primera de las fases de la metodología Scrum es donde estudiamos y analizamos los requerimientos, se definen los roles que forman el equipo Scrum y se organiza la forma de cómo ejecutar el proyecto.

Para que esta fase de Scrum tenga un buen resultado nos fue necesario responder las siguientes preguntas en referencia al proyecto:

¿Cuál será el resultado final del proyecto?

¿Cómo se va a realizar el proyecto?

¿Quiénes son las personas que conformarán el equipo Scrum?

¿Cuánto tiempo se tiene para lograr el proyecto?

Para cumplir con el objetivo de la primera de las fases de la metodología Scrum, establecimos seis (6) procesos que fueron:

### *1- Visión del proyecto*

Este es el proceso de Scrum donde se estableció que es lo que desea el cliente y cuáles son sus necesidades iniciales, también explica por qué lo quiere y cuál es el objetivo del proyecto. El cliente lo necesita para impulsar su negocio, que más clientes de compra se enteren de sus excelentes productos, tener un mejoramiento de posición en el mercado, incrementar su rentabilidad y ser sostenible en el tiempo.

Su objetivo es proporcionar soluciones a problemas de publicidad, mercadeo, ventas y forma de pago, mejorar la eficiencia de estas áreas y satisfacer las necesidades y deseos de los usuarios finales que consumen productos en cuero.

Se definió al dueño del producto o Product Owner quién será la persona encargada de ser la voz del cliente frente al equipo Scrum, como dueño del producto dependerá la calidad del entregable, así como la aprobación o rechazo del trabajo del equipo de desarrollo en cada Sprint. Para ello y por consenso nombramos al compañero **Jonnathan Reyes**

### *2- Identificar al Scrum Master y los colaboradores*

Una vez definido cuál será el objetivo del proyecto y nombrado al Product Owner, fue necesario nombrar la persona que será el Scrum Master o líder del proyecto y las personas colaboradoras o los Stakeholders. En su orden se nombró a **Carlos Téllez, Miguel Guevara, Yuly Toro y Santiago Muentes.**

El Scrum Master es la persona encargada de instruir a todo el equipo Scrum sobre los principios y prácticas de Scrum, así como velar de que se cumplan a cabalidad.

Los Stakeholders en cambio son las personas colaboradoras en el desarrollo y en este caso son responsables del resultado del proyecto junto con los demás miembros del equipo

### *3- Formar al equipo Scrum*

Otra de los procesos fundamentales de la primera de las fases de la metodología Scrum, es la de establecer el equipo de desarrollo quien será el encargado de elaborar el entregable de cada uno de los Sprint y para este caso son los estudiantes que presentamos este trabajo para la Tecnología ADSO-SENA-2024

El equipo de desarrollo es multifuncional y auto organizado, además dispone de todas las capacidades necesarias para desarrollar todas las tareas dentro de un Sprint.

El Dueño del producto dentro de sus funciones de esta primera etapa de las fases de la metodología Scrum armó el equipo Scrum que le acompañará y para esto le apoyó el Scrum Master y así se llevó a cabo.

### *4- Desarrollar las Épicas*

Una épica en SCRUM es una *historia de usuario o un requerimiento* del proyecto, que está descrito a un nivel tan general que posteriormente es necesario descomponer en historias de usuario más pequeñas para que el equipo de desarrollo pueda trabajarlas durante un Sprint.

Entonces en este proceso de primera etapa de las fases de la metodología Scrum se busca tener una visión más real de todo lo que el cliente necesita o todo lo que debería tener el resultado final del proyecto, sin hacer un estudio previo de factibilidades.

El Product Owner dentro de este proyecto Scrum, para cumplir con este proceso le fue necesario elaborar mediante reuniones periódicas virtuales todas las épicas, es decir los requisitos funcionales y no funcionales del software que se consideraron necesarias para que el resultado pueda cumplir con las expectativas del cliente y en esto colaboró todo el equipo

### *5- Crear el Backlog priorizado del producto*

Luego de escribir todas las épicas fue necesario comenzar a ordenar todo en una lista de requerimientos que serán procesados a lo largo del proyecto, esto es el Product Backlog o *Pila del producto*.

El Product Backlog fue ordenado antes de comenzar el proyecto por el Product Owner, para que los objetivos cumplidos y el entregable cumplan con las necesidades del negocio o el cliente.



Además, para cumplir con el principio de adaptación de Scrum el orden de esta lista es revisada Sprint a Sprint y fue variando según las prioridades del negocio fueron cambiando o de nuevos requerimientos a medida que el proyecto avanzó.

#### *6- Realizar la planificación del lanzamiento*

Todo proyecto bien planificado debe tener un comienzo y un final, es por ello que en esta fase de la metodología Scrum se define la fecha de la presentación del entregable final o del producto terminado. Para este caso fue en **Junio de 2024**.

Así es más fácil identificar cualquier variación de lo planificado y definir el lapso de tiempo que tendrán cada uno de los Sprint a fin de que el equipo de desarrollo pueda llevar un ritmo de trabajo sostenible.

## **SEGUNDA FASE DE SCRUM: *Planeación y estimación***

Para poder aumentar las probabilidades de éxito de este proyecto Scrum fue necesario realizar un buen control de todos los recursos involucrados y esto comenzó con una buena planificación y estimación.

El Scrum Master o líder durante esta segunda fase de la metodología, su función consistió en asignar las distintas tareas a cada uno de los roles del equipo Scrum y velar porque se cumpliera según la metodología.

Además, que las historias de usuario y las tareas quedaran bien entendidas por todos los participantes, para que al finalizar el entregable pueda cumplir con los criterios de aceptación del Product Owner.

Para cumplir con objetivo de la segunda de las fases de la metodología Scrum, se establecieron y cumplieron seis (6) procesos, los cuales son:

#### *1- Crear las historias de usuario*

El Dueño del Producto construyo junto con el equipo conformado el Product Backlog para lo cual fue necesario comenzar a detallar las historias de usuario que el equipo de desarrollo estuvo trabajando durante el Sprint.

Esto facilitó la comprensión de cada uno de los requerimientos y delimitó hasta donde tiene que llegar el equipo durante el desarrollo, para evitar desperdicios de recursos.

#### *2- Estimar las historias de usuario*

Una vez identificadas cuáles son las historias de usuario que se desarrollaron durante el Sprint, se hizo necesario hacer una estimación inicial de cuánto tiempo, era necesario para finalizar el proyecto y asignarle un valor con el fin de poder construir el diseño de la arquitectura del software y sus componentes o módulos con

la creación de diagramas de flujo, diagramas de clase, diagramas de secuencia y de despliegue que permiten visualizar su estructura definitiva.

El Product Owner solicitó ayuda al equipo de desarrollo Scrum, para que gracias a su experiencia pudiera llegar a una estimación más cercana a lo que es la realidad.

### *3- Comprometerse con las historias de usuario*

Uno de los valores de Scrum es el compromiso, el cual es necesario mantener en todas las fases de la metodología Scrum, sin embargo, en esta etapa como fue crucial que el equipo Scrum se comprometiera a terminar todas las historias de usuario acordadas, así se hizo.

Para ello el equipo de desarrollo y el dueño de producto revisaron la magnitud de las historias de usuario y se analizó que era factible realizarlas en el tiempo que durara el sprint.

### *4- Identificar las tareas*

Ya acordadas cuales son las historias de usuario a desarrollar durante el Sprint fue necesario identificar las tareas necesarias para realizarlas, para ello fue necesario que todo el equipo de desarrollo participara. Las tareas identificadas fueron los prototipos de la interfaces para cada uno de los módulos propuestos, el establecimiento de los módulos de software de la página, seleccionar las imágenes adecuadas, establecer las herramientas tecnológicas para su desarrollo como fueron el uso del editor de código Visual Studio Code con sus diversas extensiones, el paquete de Xampp para la base de datos, los lenguajes de diseño html y css, los lenguajes de programación java script y php que sigan y cumplan un estilo de codificación y también cumplan con los estándares del proyecto, al igual que Phpunit o Katalon Studio para las pruebas respectivas practicadas mediante el plan y casos de pruebas de los módulos codificados, realización de las pruebas unitarias, pruebas de integración, pruebas de sistema y pruebas de aceptación del usuario, como la realización de los ajustes respectivos. Posteriormente el respectivo despliegue e instalación, es decir una vez que el software ha pasado las pruebas satisfactoriamente, se implementa en el entorno de producción. Esto incluyo la instalación del software en los servidores del cliente o en la nube mediante el hosting y dominio, configuración de bases de datos y otros componentes necesarios.

### *5- Estimar las tareas*

Una vez que se tuvieron todas las tareas necesarias para completar las historias de usuario y demás que se realizarán en los Sprint, fue necesario calcular la estimación en tiempo que se llevará cada una, para ver si es factible terminirlas dentro del timeboxing establecido, osea la técnica de gestión del tiempo sencilla que incluye repartir por adelantado una unidad de tiempo máxima y fija para una actividad, y luego completarla dentro de ese período. Por las circunstancias académicas establecimos seis semanas.

En el caso de que la cantidad de horas estimadas por el equipo de desarrollo para realizar todas las tareas supere la cantidad de horas del Sprint, habrá que replantear los objetivos del Sprint con el Dueño del Producto, pero para el caso no hubo necesidad de prolongar este tiempo.

#### *6- Crear el Sprint Backlog*

Una vez definidas, priorizadas y estimadas todas las tareas del Sprint fue necesario colocarlo en una lista que se conoce como el Sprint Backlog la cual estuvo al alcance y su gestión por todo el Equipo Scrum de desarrollo.

Para facilitar la gestión de las tareas del Sprint Backlog durante la ejecución del Sprint lo más recomendable fue armar un tablero Kanban, así el estado de lo desarrollado y lo planificado estuvo siempre visible, lo que permitió detectar desviaciones rápidamente.

### **TERCERA FASE DE SCRUM: Implementación**

En esta fase de Scrum es donde el Equipo de Desarrollo produce el entregable gracias a tareas recabadas en la fase anterior, muchos de estos se repetirán diariamente durante el Sprint a fin de mantener los 6 principios de Scrum.

Para cumplir con el objetivo de la tercera de las fases de la metodología Scrum, se establecieron tres (3) procesos los cuales son:

#### *1- Crear Entregables*

Se refiere al trabajo diario que realiza el Equipo de Desarrollo para cumplir cada una de las tareas contenidas dentro del Sprint Backlog en el estricto orden de prioridades.

Esto con la finalidad de garantizar que, de no llegar a cumplir todas las tareas programadas del Sprint, siempre quedaran ejecutadas las que son más relevantes según la importancia establecida por el Product Owner.

El Scrum Master del proyecto estuvo pendiente para que cada miembro del equipo sólo estuviera elaborando una tarea a la vez, así se evitaron los embudos en los procesos y que todo el equipo estuviera trabajando.

#### *2- Realizar Daily Standup*

Fue la revisión diaria para saber el avance del Equipo de Desarrollo y si existía algún inconveniente que pudiera generar algún impedimento, para así lograr seguir avanzando.

Para ello cada persona del equipo debió responder las siguientes preguntas:

¿Qué hice ayer?

¿Qué voy a hacer hoy?

¿Existe o existió algún impedimento en mis tareas?

Esto se realizó en reuniones virtuales periódicas con la herramienta de comunicación de Microsoft Teams que llevó acabo todo el Equipo de Desarrollo en la denominada “Daily Scrum Meeting” o Reunión Diaria del Sprint.

Debemos de anotar que para lograr una mayor efectividad a esta reunión asistieron todos los integrantes del Equipo Scrum, para no tener que ir dando informaciones por separado y que todos estuvieran al día de lo que pasó en el proyecto.

### *3- Refinamiento del Backlog priorizado del Producto*

En este proceso de la tercera de las 5 fases de la metodología Scrum se busca preparar los insumos necesarios para arrancar el siguiente Sprint asegurando que habrá suficientes historias de usuario y demás tareas para continuar.

El objetivo de este proceso de Scrum fue ordenar las historias de usuario y tareas según la prioridad que estableció el Dueño del Producto, y se definió y estableció el esfuerzo de las próximas historias y tareas a trabajar.

Así cuando se volvió a realizar el ciclo para generar el próximo entregable, el equipo Scrum solo necesitó identificar y estimar las tareas relacionadas a las historias seleccionadas para crear el Sprint Backlog.

## **CUARTA FASE DE SCRUM: Revisión y retrospectiva**

En la cuarta fase de la metodología Scrum se busca realizar una revisión del entregable, además de los procesos que se realizaron durante el Sprint, con la finalidad de medir y mejorar la eficiencia del equipo Scrum.

Para cumplir con el objetivo de la cuarta de las fases de la metodología Scrum, establecimos dos (2) procesos los cuales son:

### *1- Demostrar y validar el Sprint*

Una vez terminado cada Sprint el Equipo de Desarrollo le mostró al Dueño del producto el incremento de valor que han generado, para que el aceptará o no el resultado.

Para ello en este proceso de la cuarta de las fases de la metodología Scrum se llevó a cabo una ceremonia denominada “Sprint Review” o Revisión del Sprint en donde se evaluó el Entregable según los Criterios de Aceptación establecidos.

Según lo que dicta la Guía de Scrum debimos de planificar el timeboxing de esta ceremonia en 4 horas para los Sprint y así acortar tiempo.

## *2- Retrospectiva del Sprint*

Para cumplir con la mejora continua dentro de las fases de la metodología Scrum, fue necesario revisar los procesos e inconvenientes que fueron parte de un Sprint, con la finalidad de que el Equipo Scrum pudiera mejorar su eficiencia.

Esta revisión se hizo durante la ceremonia del “Sprint Retrospective” o Retrospectiva del Sprint, aquí es donde buscamos soluciones y oportunidades para mejorar los procesos en los próximos Sprint que se llevaron a cabo.

## **QUINTA FASE DE SCRUM: Lanzamiento**

Esta fase de la metodología Scrum busca entregar el resultado final al cliente y poder acumular experiencia que puedas utilizar en próximos proyectos para mejorar la eficiencia del equipo Scrum.

Para cumplir con el objetivo de la quinta de las fases de la metodología Scrum, establecimos dos (2) procesos los cuales son:

### *1- Enviar Entregables*

Este proceso de Scrum tuvo como objetivo entregar el resultado final al Cliente en este caso académico al Profesor Nicolas Saldarriaga y sus colegas Carlos Mora y Carlos Fuel, una vez que se cumplieron todas las historias de usuario y las diversas tareas definidas y realizadas por el dueño del producto y todo el equipo de desarrollo.

### *2- Retrospectiva del Proyecto*

La retrospectiva del proyecto es el proceso de la quinta de las fases de la metodología Scrum que permitió realizar una inspección de todo lo que sucedió durante el proyecto para buscar la mejora continua.

Cuando realizamos este proceso junto con el equipo Scrum, se analizó lo que fue óptimo y lo que se pudo mejorar, y así poder implementar este conocimiento en futuros proyectos.

Para terminar, se estableció el respectivo plan de mantenimiento que se llevara a cabo después del despliegue, recordemos que el software necesita ser mantenido y actualizado para corregir errores, agregar nuevas características o adaptarse a cambios en el entorno. Esto puede incluir la resolución de problemas, la aplicación de parches de seguridad y la optimización del rendimiento.



Durante la aplicación de esta metodología Scrum nos surgieron diversas preguntas y respuestas que damos a conocer:

**¿Es necesario que un proyecto cumpla a cabalidad con todos los procesos y fases de la metodología Scrum?**

Como el marco de trabajo Scrum se puede adaptar a cualquier tipo de organización, equipo de trabajo o proyecto, la Guía de Scrum no establece ninguna restricción en cuanto al uso de cada una de las 5 fases o los 19 procesos de Scrum.

Todo dependerá de la complejidad y magnitud del proyecto que se encuentre realizando, en mi experiencia como consultor de proyectos le puedo decir que muchas veces podrás prescindir de algún proceso y que muchas veces pasará por todos ellos sin darse cuenta.

**¿Se debe repetir las 5 fases de la metodología Scrum cada vez que comienza un Sprint?**

Basados en nuestra experiencia como estudiantes de la Tecnología ADSO, por lo general las fases de la metodología Scrum que se repiten durante el desarrollo de un proyecto con Scrum son la de Planificación y Estimación, Implementación y la de Revisión y Retrospectiva, porque es en donde están los eventos de Scrum que tienen que ver con el desarrollo del entregable.

Debido a que las otras dos fases de la metodología Scrum están más relacionadas al arranque y cierre del proyecto.

**Recomendaciones finales sobre las 5 fases de la metodología Scrum**

Respetado Profesor como podrá ver es factible implementar las 5 fases de la metodología Scrum en cualquier tipo de proyecto en donde quiera controlar el timeboxing y los recursos adecuadamente para lograr una mayor eficiencia.

Como tuvimos dudas sobre cómo implementar las 5 fases de la metodología Scrum, hicimos contacto para asesorías con los profesores Saldarriaga, Mora y Fuel.

## HERRAMIENTAS EMPLEADAS EN LA METODOLOGÍA AGIL-SCRUM

**1- El Backlog de Producto:** Este va a ser un tablero en el que esté toda la información del proyecto. En el encontraremos:

1°- *Las historias de Usuario:* Que son tarjetitas en las que se explica quien, como y para qué se va a realizar el proyecto y los criterios que tiene que cumplir. En nuestro ejemplo:

Un miembro del equipo: Asistente del área de Pago del producto de cuero

Quiero: Poder realizar mejor exhibición de productos para los clientes

Para: Para poder explicar la excelente calidad de sus diseños y materias primas

2°- *Criterios del proyecto:*

Deben ser exhibiciones que extraigan datos de diferentes plataformas de anuncios como Facebook o Google

Debe aparecer la información del cliente

Debe ser fácil de entender

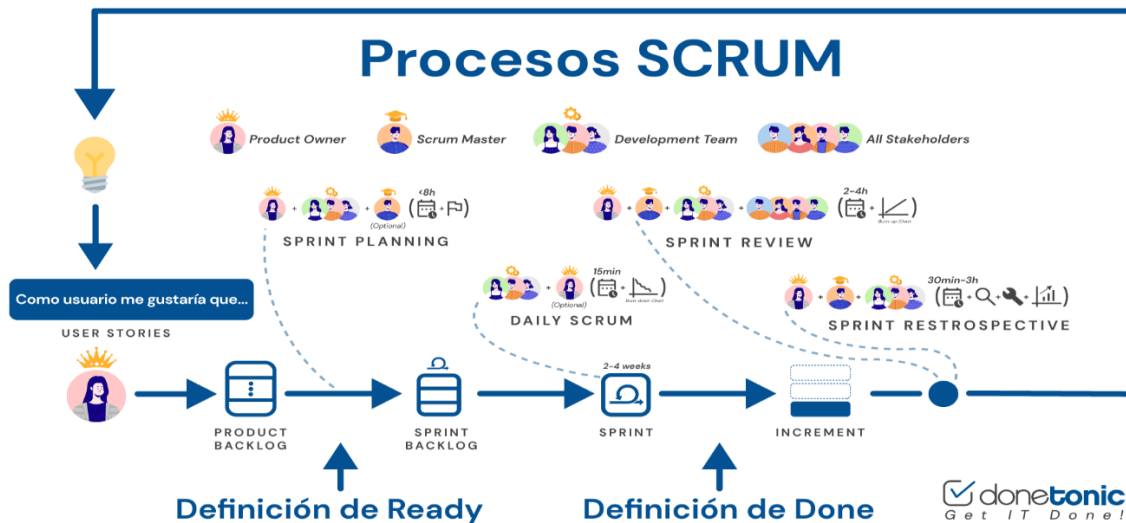
Debe incluir los siguientes pasos que vamos a tomar

3°- *El tablero Kanban:* En este tablero se juntan las primeras tareas y se dividen en fechas y personas responsables y se añaden columnas como "Pendientes" "En progreso" y "Completadas".

4°- *Sprint:* Son periodos de tiempo fijo en el que se desarrolla el proyecto, normalmente entre 1 y 3 semanas. Durante este periodo de tiempo se desarrollan 3 fases:

- a) *Planeación:* Que es donde se reúne el equipo con los clientes o usuarios que se van a beneficiar de la solución.

- b) Daily Scrum: Son reuniones diarias cortas en las que los miembros del equipo explican las tareas que han realizado, y las que van a hacer, igualmente comentan dudas y exponen obstáculos que no les permiten avanzar.
- c) Revisión y Retrospectiva de Sprint: Al final del proyecto hay una reunión para revisar con los beneficiarios del producto como está quedando y retocar y responder dudas para llegar al mejor resultado posible. Además, se evalúa el desempeño de los integrantes del equipo y qué cosas se pueden mejorar para el próximo Sprint.



En general existen 3 roles muy importantes dentro del Scrum y que aplicamos:

*El Product Owner.* Que ha preparado las historias de usuario, se ha reunido con los interesados del proyecto para entender al detalle los requerimientos del proyecto. Ha plasmado la visión en el Backlog de Producto y es el encargado de celebrar las reuniones diarias, priorizar tareas y asignarlas y facilitar la comunicación a todo el equipo para que esté todo claro.

*El Scrum Master.* Garantiza que todo sigue las normas de la metodología y facilita los obstáculos a los integrantes del equipo para que únicamente estén centrados en sacar las tareas adelante.

*Los miembros del equipo de desarrollo.* Diseñadores, programadores y otros profesionales encargados de sacarlo adelante.



## **FASES DEL CICLO DE VIDA DEL SOFTWARE DESARROLLADO BELLA Y VALE**

Los periodos o etapas del desarrollo de nuestro proyecto de software fueron las siguientes:

### **Planificación**

Antes de comenzar el plan de desarrollo de nuestro proyecto de información, se necesitó de hacer ciertas labores que influirán decisivamente en el triunfo del mismo.

Varias de las labores de esta etapa integran ocupaciones como la decisión del entorno del plan, la ejecución de un análisis de viabilidad, la investigación de los peligros asociados, la estimación del costo del plan, su planeación temporal y la asignación de recursos a las diversas fases del plan.

En esta fase de planeación, intentamos que el código no sea indescifrable siguiendo diversas pautas como las siguientes:

- Evitar bloques de control no estructurados.
- Identificar de manera correcta las cambiantes y su alcance.
- Elegir algoritmos y construcciones de datos idóneas para el problema.
- Mantener la lógica de la aplicación lo más simple viable.
- Documentar y comentar correctamente el código de los programas.
- Facilitar la interpretación visual del código usando normas de formato de código anteriormente consensuadas en los equipamientos de desarrollo.
- Además, se debe considerar la compra de recursos necesarios para que el programa funcione, además de desarrollar casos de prueba para verificar el desempeño del mismo según se vaya programando.

### **Análisis**

Obviamente, y de manera previa y por consenso del equipo de desarrolladores consultamos y determinamos qué es exactamente lo que tiene que hacer el programa informático. Por esa razón, la fase de estudio en el periodo de vida del software correspondió al proceso por medio del cual se aspira hallar qué es en verdad lo que es necesario y llegamos a una comprensión idónea de los requerimientos del software (las propiedades que el software debería poseer).

### **Diseño**

En esta etapa se estudiaron las probables posibilidades de utilización para el software que se queremos construir, así como dictaminar la composición general del mismo.

El diseño es una fase compleja y su proceso debería desarrollarse de forma iterativa.

Es probable que la solución inicial no sea la más correcta, por lo cual en tal caso se debe refinarla. Sin embargo, hay catálogos de patrones de diseño bastante útiles que recogen errores que otros han cometido para no caer en la misma trampa.

## **Implementación**

En esta etapa se debe escoger los instrumentos idóneos, un ámbito de desarrollo que facilite el trabajo y un lenguaje de programación apropiado para el tipo de software a construir. Esta elección depende tanto de las elecciones de diseño tomadas como del ámbito en el cual el software deba funcionar.

## **Pruebas**

Como cometer un error es humano, la etapa de pruebas del periodo de vida del software busca identificar los fallos realizados en los periodos anteriores para corregirlos. Desde luego, lo ideal es realizarlo previo a que el cliente final se los encuentre. Se plantea que una prueba es un triunfo si se detecta cualquier error.

## **Instalación o despliegue**

Esta etapa es colocar el software funcionando, por lo cual debimos planear el ámbito teniendo presente las dependencias existentes entre los diferentes elementos del mismo.

Es viable que haya elementos que funcionen de manera correcta por separado, sin embargo, que al combinarlos causen inconvenientes. Por esto, se debe utilizar combinaciones conocidas que no ocasionen inconvenientes de compatibilidad.

## **Uso y mantenimiento**

Esta es una de las etapas más relevantes del periodo de vida de desarrollo de nuestro proyecto de software. Pues el software ni se rompe ni se desgasta con la utilización, su mantenimiento incluye 3 aspectos diferenciados:

- Borrar las deficiencias detectados a lo largo de su historia eficaz (mantenimiento correctivo).
- Adaptarlo a novedosas necesidades (mantenimiento adaptativo).
- Añadirle novedosas funciones (mantenimiento perfectivo).

Aunque suene contradictorio, cuanto mejor es el software más tiempo se debe invertir en su mantenimiento.

La primordial razón es que se utilizará más (incluso de maneras que no se habían previsto) y, por ende, habrá más propuestas de mejoras.

El ciclo de desarrollo de software es necesario para entregar un producto de calidad que satisfaga las necesidades del usuario.

## CONCLUSIONES

Las acciones de investigar y estudiar por cada uno de los miembros de nuestro equipo de trabajo sobre el tema de metodologías de desarrollo de software, nos condujo a aprender y comprender las características y beneficios de estas metodologías, lo cual nos permitió seleccionar la más adecuada para nuestro proyecto de desarrollo de software y maximizar la eficiencia del equipo.

Podemos aseverar que no existe una única metodología “perfecta”, pero la clave está en adaptar y personalizar estas metodologías según las necesidades y características de los proyectos de software.

Una metodología está diseñada para reducir el riesgo de fracaso y crear un software de mayor calidad, en una cantidad asumible de tiempo y presupuesto. Cada metodología tiene sus ventajas e inconvenientes, por lo que es importante elegir la que mejor se adapte a cada situación.

El uso exitoso de Scrum depende de que las personas se vuelvan más competentes en vivir cinco valores: ***Compromiso, Foco, Franqueza, Respeto y Coraje***

El equipo Scrum se compromete a lograr sus objetivos y a apoyarse mutuamente. Su foco principal está en el trabajo del Sprint para lograr el mejor progreso posible hacia estos objetivos. El equipo Scrum y sus interesados son francos sobre el trabajo y los desafíos. Los miembros del equipo Scrum se respetan entre sí para ser personas capaces e independientes, y son respetados como tales por las personas con las que trabajan. Los miembros del equipo Scrum tienen el coraje de hacer lo correcto, para trabajar en problemas difíciles.

Los tres pilares del scrum son los componentes fundamentales de una gestión de proyectos exitosa. Con la *transparencia, la inspección y la adaptación*, los equipos gestionan las complejidades y las incertidumbres con agilidad y fomentan una cultura de innovación.

## BIBLIOGRAFÍA Y WEBGRAFIA

[https://informatecdigital.com/software/13-metodologias-de-desarrollo-de-software-guia-completa/#google\\_vignette](https://informatecdigital.com/software/13-metodologias-de-desarrollo-de-software-guia-completa/#google_vignette)

[https://es.wikipedia.org/wiki/Metodolog%C3%ADa\\_de\\_desarrollo\\_de\\_software](https://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software)

<https://vgst.net/blog/development/metodologias-de-desarrollo-de-software>

[https://todoingenierias.com/metodologias-de-desarrollo-de-software-mas-usadas-y-caracteristicas/#scrum\\_un\\_enfoque\\_colaborativo\\_y\\_adaptable](https://todoingenierias.com/metodologias-de-desarrollo-de-software-mas-usadas-y-caracteristicas/#scrum_un_enfoque_colaborativo_y_adaptable)

<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

<https://www.atlassian.com/es/agile/project-management/3-pillars-scrum>

<https://www.productiviza.com/scrum-ejemplos-practicos-y-reales-explicados/>

<https://alvaroarrarte.com/fases-de-la-metodologia-scrum/>