

# Curso Profesional de Scrum

Las metodologías ágiles son una filosofía de trabajo que facilita el proceso de creación y entrega de valor continuo en proyectos de desarrollo de software.

## ¿Qué es Agile?

- Agile es la habilidad de crear productos y responder al cambio.
- Agile es muchas cosas:
  - Es un conjunto de marcos de trabajo y metodologías.
  - Agile se centra en las personas y como éstas trabajan y colaboran.
- El principio de Agile se basa en la habilidad para **crear productos** que responden a las necesidades específicas de un cliente, tomando en cuenta un escenario particular: si las necesidades o los requerimientos del cliente, el negocio o el mercado cambian, **el producto se adapta**.
- Esta capacidad de adaptación al cambio y la entrega de valor constante forman parte de los principios de las metodologías ágiles.

## Manifiesto por el Desarrollo Ágil de Software

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

- **Individuos e interacciones** sobre procesos y herramientas
- **Software funcionando** sobre documentación extensiva
- **Colaboración con el cliente** sobre negociación contractual
- **Respuesta ante el cambio** sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

## Diferencias entre el desarrollo ágil y el desarrollo tradicional

Grupos de trabajo:

- Equipos de trabajo pequeños
- Tiempos de iteración cortos (dos semanas)

Requerimientos

- Definición de una lista de requerimientos
- División de los requerimientos en listas más pequeñas para trabajar durante cada iteración
- Objetivos de desarrollo claros

## Generación de valor

Al trabajar mediante iteraciones cortas es posible hacer entregas paulatinas y constantes, por lo cual es más factible realizar ajustes, en caso de que sean necesarios y el cliente percibe más fácilmente el valor generado.

## 12 principios ágiles

1. **Satisfacción al cliente.** La prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. **Cambios.** Los cambios son bienvenidos en cualquier etapa del proyecto. Los procesos ágiles aprovechan el cambio para adaptarse a las necesidades del negocio y para proporcionar ventaja competitiva al cliente.
3. **Software funcional.** Entrega de software funcional en un periodo corto (entre dos semanas y dos meses).
4. **Colaboración.** Los responsables del negocio y los desarrolladores trabajan juntos durante todo el proyecto. La colaboración facilita la comunicación del equipo de trabajo con el cliente y esto también beneficia al usuario final del producto.
5. **Individuos motivados.** El éxito de los proyectos está determinado, en gran medida, por individuos que se sienten motivados. Se debe garantizar un buen entorno de trabajo y dar al equipo independencia para tomar decisiones y ejecutar las actividades relacionadas con el proyecto.
6. **Comunicación cara a cara.** Es el método más eficiente y efectivo de comunicar información al equipo.
7. **Progreso.** El software funcionando es la principal medida de progreso.
8. **Desarrollo sostenible.** El ritmo de desarrollo debe ser constante a lo largo de las iteraciones.
9. **Mejora continua.** Está determinada por la excelencia técnica y el buen diseño a lo largo del proceso, en cada una de las iteraciones, el equipo puede identificar fácilmente qué hizo bien y qué puede mejorar.
10. **Simplicidad.** El arte de maximizar la cantidad de trabajo no realizado es esencial. Se deben contemplar requerimientos bien definidos y un alcance claro para aprovechar al máximo los resultados de las tareas ejecutadas. *Unicamente vamos a trabajar las cosas que tenemos bien definidas en ese instante.*
11. **Auto-organización.** Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. **Autoevaluación.** El equipo tiene la capacidad de ajustar y perfeccionar su comportamiento.

## Los 12 principios del manifiesto (simplificados)



### ¿Qué es Scrum?

**Scrum es un marco de trabajo ágil** por el cual las personas pueden abordar problemas complejos adaptativos, a la vez que entregan productos del máximo valor posible de forma productiva y creativa.

Scrum es un proceso iterativo y las fases deben estar delimitadas en el tiempo. Si las iteraciones son de dos semanas, siempre serán de dos semanas.

Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce.

### Características de Scrum

- Scrum está enfocado en generar valor de la forma más rápida posible. Este marco de trabajo permite resolver un problema que requiere una solución de software (sistema bancario, sistema de aprendizaje o sistema de reservaciones de un restaurante). Para ello:
  - Todo el equipo colabora con el propósito de lograr un objetivo.
  - El equipo debe ser pequeño (entre 3 y 10 personas).
  - Las personas del equipo deben ser altamente flexibles y con capacidad de adaptación.

### 3 pilares de Scrum

Estos son los pilares sobre los cuales se fundamenta la filosofía de Scrum:

1. **Transparencia.** Cualquier persona involucrada en el proyecto puede conocer el estado actual del proyecto.
2. **Inspección.** Cualquier persona del equipo puede observar los requerimientos (artefactos) y analizar si están bien definidos o no, en cualquier momento del proceso.
3. **Adaptación.** La capacidad de reconocer cambios e implementarlos de la forma más rápida posible.

### Valores de Scrum

- **Compromiso.** Define que el equipo se compromete al 100% en el objetivo. No solo en el proyecto, sino en cada iteración.
- **Coraje.** No importa la tarea, cualquier integrante del equipo está en capacidad de llevarla a cabo.
- **Enfoque.** Cada integrante está enfocado en cumplir el objetivo planteado.
- **Apertura.** Los involucrados en el proyecto están dispuestos a escuchar aportes u observaciones que se realicen durante el proceso.
- **Respeto.** Todas las personas hacen parte de un mismo equipo. Por ello, el trato y la buena comunicación son importantes.

(IMAGÉN EN LA SIGUIENTE PANTALLA)

# SCRUM

La mejor Metodología de Desarrollo Ágil

## ¿Qué es?

Es un marco de trabajo de desarrollo de Software, que se ha adaptado a otros negocios, tiene como característica el trabajo colaborativo y el alcance de objetivos de manera rápida.

La metodología SCRUM tiene 3 grandes pilares:

- Transparencia
- Inspección
- Adaptación

Valores de SCRUM:

Compromiso  
Coraje  
Enfoque  
Apertura  
Respeto

## ¿Quiénes conforman el equipo?

Product Owner



- Dueño del producto.
- Da claridad en los objetivos
- Prioriza actividades.

Scrum Master



- Responsable de promover y apoyar SCRUM en el equipo.
- Facilita los eventos SCRUM.
- Elimina impedimentos

Equipo de Desarrollo



- Conjunto de profesionales que realizan diversos trabajos
- Son autoorganizados.
- Grupo multidisciplinario

En la actualidad es considerada como una de las metodologías con mejores prácticas para las organizaciones.

## Los componentes de Scrum

La práctica de Scrum en proyectos de desarrollo de software involucra 3 componentes principales:

1. Equipo
2. Ceremonias
3. Artefactos (requerimientos)

### Equipo de Scrum

El equipo de Scrum es autoorganizado y multifuncional. Tiene la capacidad de decidir la forma como va a alcanzar los objetivos trazados y de cumplir los objetivos que se le asignen. Los roles del equipo de Scrum son:

- **Product Owner** → Responsable de maximizar el valor del producto. La cara del cliente dentro del equipo de scrum. Entiende los requerimientos del cliente y el alcance del producto.
- **Scrum Master** → Responsable de promover y apoyar Scrum. Es el gurú de Scrum y es el encargado de asegurar la implementación de esta metodología en el proyecto. Está presente en el día a día del proceso y se asegura de que sea fluido y que avance.
- **Development Team** (Equipo de desarrollo) → Profesionales que se encargan de la entrega continua de producto terminado. Entre ellos, se encuentran programadores, diseñadores, arquitectos de información, entre otros. Son los encargados de construir el producto

### Ceremonias o eventos de Scrum

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas.

En el proceso de ejecución de Scrum se establece un marco para limitar las interacciones en momentos claves y con duraciones puntuales. De esta forma, se optimizan los tiempos del equipo.

- **Sprint** → Es el corazón de Scrum donde se crea el incremento del producto. Se refiere al tiempo delimitado (iteración) durante el cual el equipo trabaja en el desarrollo del producto.
- **Sprint planning** (Planificación de Sprint) → Ceremonia para definir qué se hará durante el sprint.
- **Daily stand-up** → Reunión diaria en la que participa todo el equipo de desarrollo para hablar acerca de cómo va el proceso para alcanzar el objetivo del sprint.
- **Sprint review** (Revisión de Sprint) → Al finalizar el sprint, se realiza esta review en la cual se hace visible el avance real durante el sprint y se determina si se logró o no el objetivo.
- **Sprint retrospective** (Retrospectiva de Sprint) → Este es el proceso que permite al equipo evaluar y analizar qué hizo bien y qué puede mejorar. Así es como se aplican prácticas de mejora continua.

### Artefactos de Scrum (requerimientos)

Los artefactos de Scrum son aquellos elementos que van a definir en qué consiste el producto (en qué se está trabajando, qué es lo que quiere el cliente) y son visibles para todas las personas involucradas en el proyecto.

Representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidad para la inspección y adaptación.

- **Product Backlog** (Lista del producto) → Es una lista ordenada de todo lo que se conoce que es necesario en el producto. Todo lo que quiera el cliente para su producto debe estar reflejado en esta lista.
- **Sprint Backlog** (Lista de pendientes del sprint) → Elementos de la lista de producto que son seleccionados para trabajar durante el sprint.



## EQUIPO

- Product Owner
- Scrum Master
- Development Team



## EVENTOS O CEREMONIAS

- Sprint planning
- Daily Stand-up
- Sprint review
- Sprint retrospective



## ARTEFACTOS

- Product Backlog
- Sprint Backlog



## El equipo de Scrum y sus roles

El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad.

Los equipos Scrum entregan productos de forma **iterativa e incremental**, maximizando las oportunidades de obtener **retroalimentación**.

**Incremental** se refiere que no se entrega/muestra al product owner todo el producto completo en una sola iteración; Sino por partes (varias iteraciones), y después de dos o tres iteraciones ya se entrega toda la pantalla con todas las funcionalidades.



### Product Owner

- Dueño del producto.
- Gestor de requisitos.
- Atiende peticiones del cliente.



### SCRUM Master

- Responsable de promover y apoyar Scrum
- Implementa la metodología
- Elimina impedimentos



### Dev Team

- Realizan el trabajo a entregar
- Auto-organización y auto-gestión
- Generan el incremento del producto



Conformar tu equipo de Scrum es muy fácil. Ten en cuenta los siguientes tips y ejecuta con éxito tus proyectos de desarrollo de software.

- Arma un equipo con todos los perfiles profesionales que necesitas para desarrollar tu proyecto.
- Conformar un equipo de 3 a 9 personas, pequeño y flexible.
- Asegúrate de tener todos los roles de SCRUM.
- Mantén sesiones diarias de máximo 15 minutos con un objetivo específico.



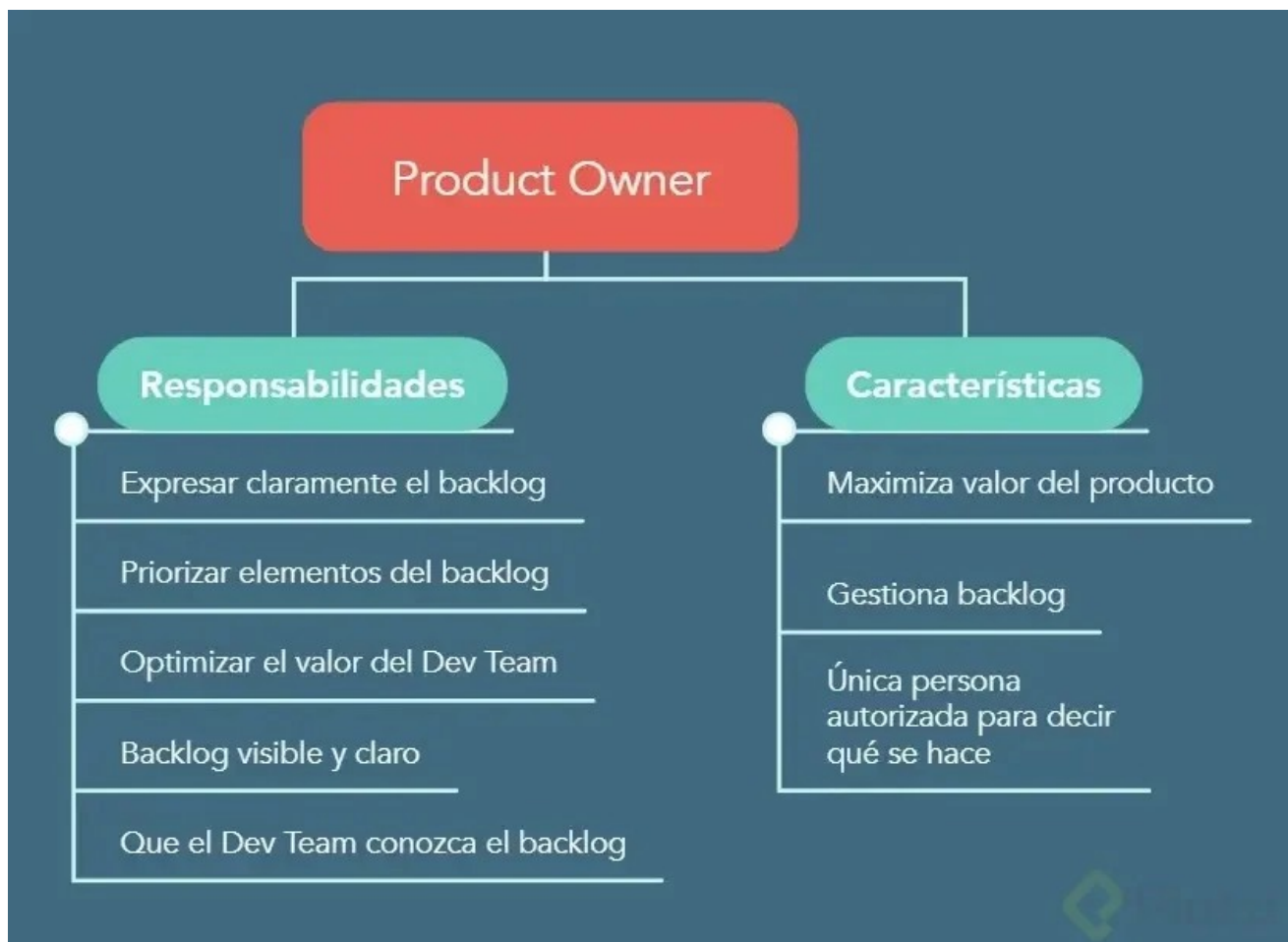
## ¿Qué hace el Product Owner o Dueño del Producto?

El Product Owner o Dueño de Producto en la metodología Scrum es el encargado de maximizar el valor del producto entregado por el equipo de desarrollo. Es la única persona del equipo responsable de gestionar el **Product Backlog** (lista del producto) y, por lo tanto, decide qué se va a trabajar y en qué orden.

### Responsabilidades del Product Owner

El Dueño del Producto en un equipo de Scrum es el encargado de despejar dudas al equipo de desarrollo y debe estar en capacidad de explicar en qué consiste cada funcionalidad del producto. Entre sus responsabilidades están:

- Expresar claramente los elementos de la lista del producto (product backlog) y los objetivos de desarrollo.
- Dar prioridad a los elementos de la lista del producto y decidir en qué orden se van a realizar los diferentes desarrollos o funcionalidades.
- Optimizar el valor del trabajo del equipo de desarrollo.
- Asegurar que la lista de producto sea visible y clara para todos los involucrados en el proyecto.
- Asegurar que el equipo de desarrollo conozca los elementos de la lista de producto.



Las decisiones de este rol van a ser reflejadas en el contenido y la priorización de la lista del producto. La decisión del **Product Owner** siempre se debe de respetar porque **es la única persona autorizada en modificar la lista del producto**. Esto no quiere decir que dicha autoridad será impuesta, se debe buscar consenso siempre para tener un equipo funcional. 🚫 Nadie puede forzar al equipo de desarrollo a que trabaje en otras actividades que no sean la prioridad establecida por el Product Owner o con base en un conjunto diferente de requisitos.

## **El rol del Scrum Master**

El Scrum Master es el responsable de promover y apoyar la ejecución de Scrum dentro del equipo, ayudando a todos a entender la teoría, la práctica, las reglas y los valores de este marco de trabajo. Si algo no se está haciendo bien, debe proponer mejoras.

Es un líder que está al servicio del equipo de Scrum y es el encargado de ayudar a entender a los externos al equipo Scrum cuando deben interactuar con el equipo.

### **¿Cómo ayuda el Scrum Master al Product Owner?**

- Asegura que los objetivos y el alcance sean entendidos por todo el equipo Scrum, que todos entiendan en el planning qué se va a trabajar.
- Transmite los conocimientos sobre Scrum y prácticas ágiles.
- Facilita los eventos de Scrum según se requiera: planea el lugar, la fecha, la hora, los recursos y se asegura de que todo esté listo cuando el equipo llegue a la sala.

### **¿Cómo ayuda el Scrum Master al equipo de desarrollo?**

- Guía al equipo de desarrollo para que sea autoorganizado y multifuncional. También identifica líderes que puedan resolver dudas o inconvenientes.
- Ayuda al equipo de desarrollo a crear productos de alto valor, asegurándose de que los objetivos durante el sprint sean entendidos. Recuerda al equipo los objetivos durante el sprint.
- Elimina impedimentos para el progreso del equipo de desarrollo.

### **¿Cómo ayuda el Scrum Master a la organización?**

- Lidera y guía a la organización en el proceso de adopción de Scrum.
- Trabaja en conjunto con otros Scrum Masters para incrementar la efectividad de la ejecución de Scrum en la organización.

(IMÁGENES ABAJO)



## El rol del Scrum Master



## Equipo de desarrollo en Scrum

El equipo de desarrollo en Scrum es el conjunto de profesionales que se encarga de crear y entregar producto "terminado" (El primer/segundo incremento terminado durante el sprint, que ese vendría siendo el objetivo (sprint backlog)), el cual se pueda poner en producción al final de cada sprint. De esta forma, el desarrollo es incremental.

La organización es la encargada de estructurar y empoderar a los equipos de desarrollo para que estos organicen y gestionen su propio trabajo.

## Características del equipo de desarrollo

- **Autoorganizado.** Sabe cómo y qué va a desarrollar durante el sprint. Se autogestiona en términos de roles y actividades internas.
- **Multifuncional.** Es un grupo multidisciplinario capaz de realizar cualquier actividad, si cuenta con las personas indicadas para la ejecución de las diferentes tareas.
- **No tiene títulos.** Es decir, no hay jerarquía y todos se deben tratar por igual, con respeto y buscando el consenso para sacar lo mejor de las diferentes habilidades individuales.
- **No hay subequipos.** No existen grupos dentro del equipo de desarrollo. Las actividades y responsabilidades se asignan por igual.
- **No se modifica el equipo de desarrollo hasta terminar el sprint.** Esto con el fin de no afectar el desarrollo de las actividades planeadas.

El tamaño óptimo del equipo de desarrollo debe ser lo suficientemente pequeño como para garantizar la agilidad, pero también lo suficientemente grande para completar el trabajo encomendado.

Lo ideal es que el equipo de desarrollo esté conformado por 3 a 9 personas, sin considerar al Product Owner o Scrum Master. Si estas dos personas también se dedican a desarrollar, se deben contar como parte del equipo de desarrollo. Así es posible asegurar que no existan dependencias externas y que no se vuelva inmanejable la gestión del equipo.



## ¿Qué es el Backlog, las Épicas y las Historias de Usuario?

En Scrum, la Lista de Producto (product backlog) es una lista ordenada de todo lo que se identifica y es necesario para el proyecto. Es la única fuente de requisitos para cualquier cambio que se realice en el producto.

Si durante la revisión del producto el cliente sugiere un cambio, ese cambio debe agregarse a la lista del producto, si una historia de usuario no está en la lista del producto no existe y nadie debe trabajar en ello.

- El Backlog es dinámico. Por lo tanto, va evolucionando a medida que avanza el proyecto, según los nuevos requerimientos.
- Los elementos en la lista más prioritarios tienen descripciones más completas.

- La Lista de Producto es dinámica, cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil.
- La Lista del Producto es un artefacto vivo, está cambiando constantemente y en ella residen todos los elementos necesarios para completar el producto.

### ¿Qué son las Historias de Usuario y las Épicas?

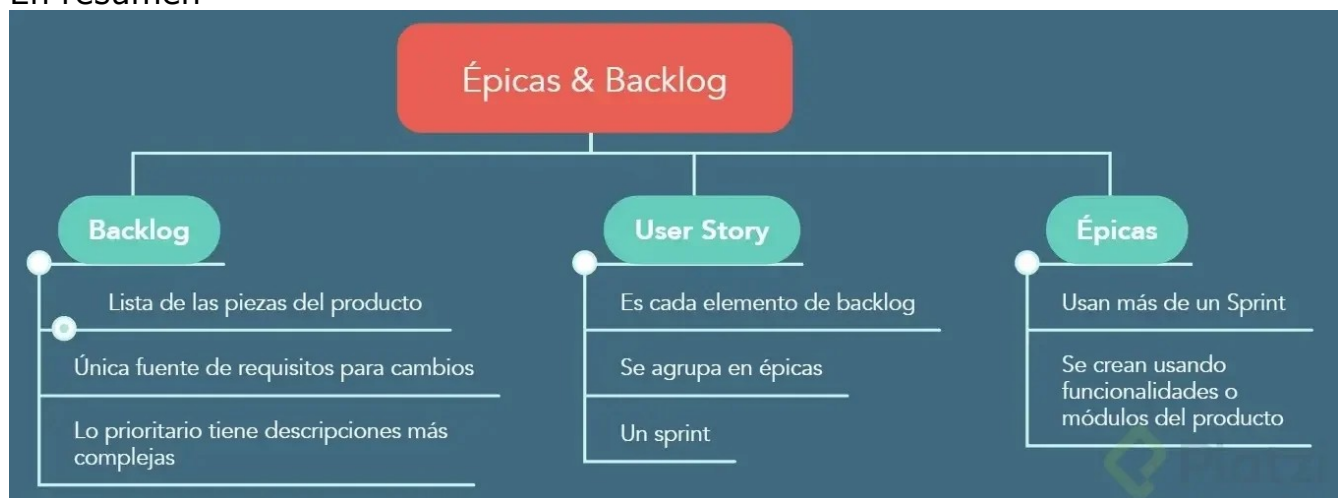
Los elementos del Backlog (Lista del Producto) se denominan Historias de Usuario. Todas las historias de usuario se pueden agrupar en elementos más grandes denominados **Épicas**.

Las Épicas se pueden crear con base en funcionalidades o módulos del producto que se está desarrollando y normalmente van a tomar más de un sprint para ser completadas.

Por ejemplo, si se desarrolla un sistema LMS (gestión de aprendizaje), las épicas pueden ser:

- Módulo de autenticación.
- Módulo de evaluación.
- Módulo para creación de cursos.
- Módulo para interacción de estudiantes.

En resumen



### ¿Cómo crear Historias de Usuario?

Las Historias de Usuario son los elementos más específicos de la lista de producto y contienen la visión del usuario sobre la funcionalidad esperada del producto. **No se deben confundir las Historias de Usuario con requerimientos**, en ellas se referencia lo que el usuario quiere o espera del producto terminado.

La historia de usuario es la funcionalidad que el usuario quiere realizar; Es la historia que el usuario nos está contando, ¿qué quiere el usuario de nuestro producto? ¿Qué quiere hacer el usuario con nuestro producto?

## Componentes de una Historia de Usuario

Estos son los componentes que debe tener una buena Historia de Usuario, para que sea fácil de entender por parte del equipo, genere valor y simplifique el proceso de desarrollo:

- **Título.** Permite conocer rápidamente de qué se trata la historia y por ello debe ser concreta y clara.
- **Descripción.** Contiene información sobre cómo se deben ejecutar las tareas, puede incluir componentes técnicos, especificar el tipo de diseño, el lugar donde se van a almacenar las tareas o el flujo de arquitectura que se debe seguir. En resumen, contempla una definición más específica para completar la historia.

Para la descripción es útil emplear una plantilla como la siguiente:

Como **<rol>** quiero **<acción>** para qué **<beneficio>**

Ejemplo:

Como estudiante <rol>

Quiero poder completar evaluaciones en la plataforma <acción>

Para poder ser evaluado y tener una calificación <beneficio>

- **Puntos.** Los puntos de una Historia de Usuario representan el esfuerzo que le va a tomar al equipo de desarrollo completar las actividades de la historia.
- **Criterio de aceptación.** A través de él se definen los requisitos que la historia debe cumplir para que esté completa.

## Historias de usuario

**Como** [Actor] (deseo/quiero) [*Verbo*<sup>1</sup>] [Concepto 1]...[Concepto n]

**Para** [*Verbo*<sup>2</sup>] [Concepto 1]...[Concepto n]

- Los conceptos son nombres (sustantivos)

- *Verbo*<sup>1</sup>: Acción

- *Verbo*<sup>2</sup>: Objetivo/Necesidad





## ¿Cómo definir que una Historia de Usuario está terminada?

La definición de "completado" en una Historia de Usuario debe incluir la lista de elementos requeridos. Este es un ejemplo:

- ✓ Funcionalidad (que se cumplan los criterios de aceptación)
- ✓ Sistema de gestión del código o de versiones (código actualizado en git)
- ✓ Pruebas creadas (unitarias, funcionales, de rendimiento)
- ✓ Documentación (a través de manuales o tutoriales)

Antes de comenzar la planeación del sprint se debe de invertir tiempo en la planeación de las Historias de Usuario. Existe una técnica llamada las tres C (por sus nombres en inglés):

1. **Cards.** Son las tarjetas creadas con los componentes o información de cada historia.
2. **Conversation.** Es la conversación del equipo acerca de la historia, para asegurar que no haya dudas en el proceso de desarrollo.
3. **Confirmation.** Todas las personas que trabajan en la historia llegan a un acuerdo y confirman que entienden su contenido.

## ¿Cómo debe ser una buena Historia de Usuario?

**I - Independiente.** No debe depender de otra Historia de Usuario. Si es así, debe marcarse con tiempo y no se debe empezar a trabajar hasta que se complete su dependencia.

**N - Negociable.** Si el equipo encuentra que la historia es muy grande para completarla durante el sprint, se puede negociar con el Product Owner para dividirla en historias más pequeñas.

**V - Valiosa.** Debe entregar valor al cliente, debe hacer algo por la funcionalidad o el producto.

**E - Estimable.** Se debe poder estimar el esfuerzo para completar la historia.

**S - Small** (pequeña). Debe ser lo suficientemente pequeña para cumplir una funcionalidad.

**T - Testeable** (comprobable). Se debe verificar que la historia se completó a través de los criterios de aceptación y de la definición de completado.

Plantilla de  
Cards (tarjetas)

### Componentes de la Historia de Usuario

Diagrama de una tarjeta de Historia de Usuario. La tarjeta es un rectángulo azul claro. En la parte superior izquierda, hay un campo "Título:". En la parte superior derecha, hay un campo "Puntos" con un icono de un cuadro. En la parte inferior izquierda, hay tres campos verticales: "Como", "Quiero" y "Para". En la parte inferior derecha, hay dos campos: "Criterios de aceptación:" y "Información adicional:".

## ¿Cómo estimar Historias de Usuario?

Para estimar Historias de Usuario en un proyecto Scrum se deben tener en cuenta 4 elementos:

- **Complejidad de la historia.** Junto con el equipo se debe definir qué tan difícil es implementar la historia.
- **Cantidad de trabajo requerido.** Representa el esfuerzo que debe invertir el equipo para llevar a cabo la historia.
- **Conocimientos necesarios.** Habilidades técnicas, de diseño y de negocio, entre otras, para completar la historia.
- **Incertidumbre.** Acciones que se requieren para ejecutar la Historia de Usuario, aunque no se tenga claro el tiempo o esfuerzo real.

El proceso de estimación de las Historias de Usuario se hace a través de **puntos**. Estos puntos no están relacionadas con una escala de medición, no representan horas o días de trabajo, sino un estimado empírico (basado en la propia experiencia y observación) con base en la experiencia del equipo.

## ¿Qué es el póker de planeación o planning poker?

El planning poker es una herramienta que sirve para que el equipo de un proyecto Scrum participe en la estimación de las Historias de Usuario.

Existen cartas físicas de planning poker pero, en caso de no tenerlas, también se puede hacer uso de aplicaciones como Scrum Poker Cards, disponible en PlayStore.

Los puntos para hacer la estimación se pueden establecer tomando como referencia diferentes escalas:

- Una de las más usadas es Fibonacci modificada (1, 2, 3, 5, 8, 13, 20, 40, 100,  $\infty$  (infinito, estimado muy grande) y ?  $\rightarrow$  (representa incertidumbre)
- 2X (1, 2, 4, 8, 16, 32)

La idea de usar estas escalas en lugar de números consecutivos es minimizar el tiempo que se puede perder en discusiones triviales o que no permitan dar continuidad al proyecto. (Que si es un 9 o un 10 más 15 mins de discusión por ello. En este caso simplemente se pone en 13)

Al final del proceso de estimación, se obtendrá el valor total de puntos de todas las Historias de Usuario y eso va a reflejar:

- **Velocidad.** Es el total de puntos de las historias de usuario completadas por el equipo durante un sprint.
- **Capacidad.** Total de historias de usuario que se pueden completar en un sprint futuro.

Durante el primer sprint es posible identificar la velocidad del equipo y con base en los resultados, será más fácil estimar los próximos sprints.

## ¿Cómo empezar? Prioridades y Backlog del Sprint

El Backlog del Sprint (lista de pendientes del Sprint) es un subconjunto de la lista del producto y contiene todos los elementos que serán desarrollados durante el Sprint.

De estos elementos dependerá el incremento a desarrollar y los objetivos del Sprint.

### Representación de la lista de pendientes del sprint

El Backlog del Sprint dependerá de las diferentes etapas de desarrollo. En él se incluye el flujo de proceso para cada Historia de Usuario, según el estatus en el que se encuentre:

Este sistema se puede gestionar en un espacio físico como una pizarra, pero también a través de herramientas digitales como Trello (gratuito) o Jira (de pago).

## Lista de Pendientes del Sprint



### Características del Backlog del Sprint

Estas son algunas características que se deben tener en cuenta para la lista de pendientes del sprint:

- Contemplar un plan lo suficientemente detallado para que todo el equipo esté en capacidad de comprenderlo en los daily stand-ups (Scrum diario).
- Asegurar que todas las personas que participan en el Scrum diario tengan conocimiento del Spring Backlog.
- El dueño del Backlog del Sprint es el equipo de desarrollo, por lo tanto, tiene potestad sobre esta lista para aceptar o no que se agreguen elementos al Sprint Backlog.
- Si un elemento se vuelve innecesario a mitad de un sprint se puede sacar de la lista de pendientes.

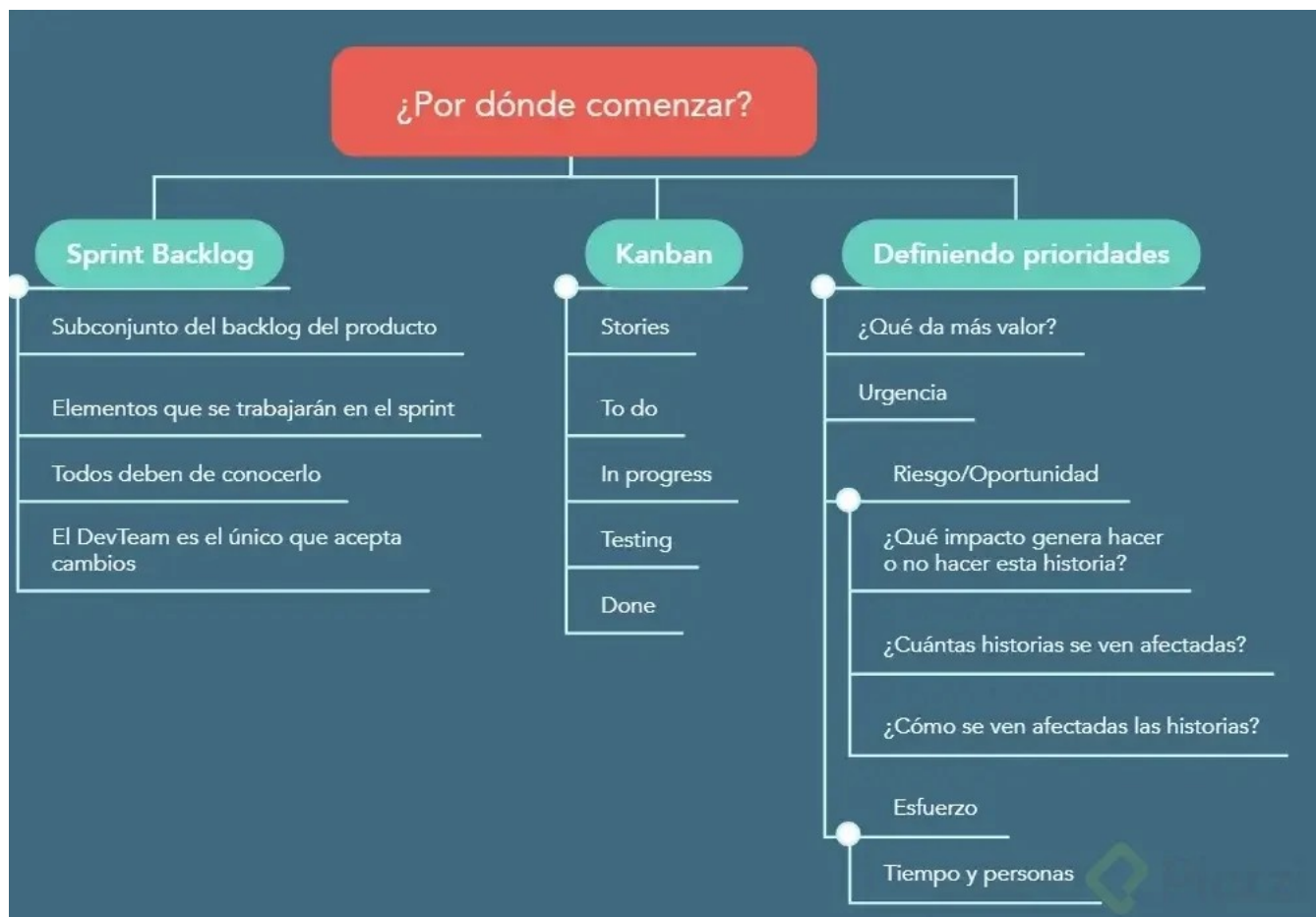
- El Product Owner podrá dialogar con el equipo de desarrollo para bajar la prioridad de una historia o incluso eliminarla.

### ¿Cómo se definen las prioridades del Backlog del Sprint?

Se sugiere tomar en cuenta los siguientes criterios para dar prioridad a las Historias de Usuario que forman parte de la lista de pendientes del Sprint:

- **Valor para el cliente.** Enfocarse en historias que generen más valor al producto.
- **Urgencia.** Por ejemplo, cuando una historia tiene una fecha para que se pueda utilizar o integrar al sistema.
- **Riesgo / Oportunidad.** Definir el impacto que la realización de una historia podría tener en el avance del proyecto o en la ejecución de nuevas historias.
- **Esfuerzo.** Qué tanto esfuerzo se requiere por parte del equipo para ejecutar la historia.

En resumen



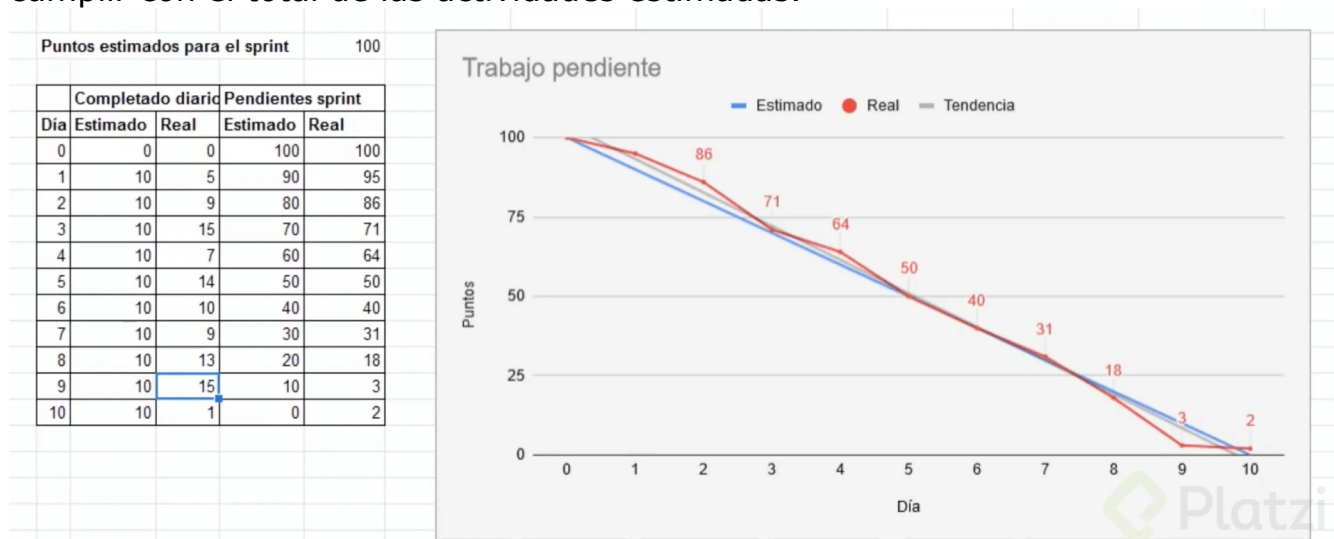
## ¿Cómo medir el avance de un proyecto Scrum?

El propósito de medir el avance de un proyecto Scrum es que el equipo pueda analizar el progreso y revisar si aún se puede cumplir el objetivo planeado al inicio de la iteración. ¿Cómo se logra esto? → Con base en los puntos de las Historias de Usuario.

Las siguientes 3 gráficas facilitan el proceso de medición del avance de un proyecto Scrum, tomando como base los puntos de las Historias de Usuario:

### Gráfica de trabajo pendiente o Bourn Down Chart 🚀

Este tipo de gráfica está enfocada en el sprint y hace visible el trabajo pendiente durante del desarrollo de un proyecto. Además, permite identificar la velocidad con la cual se están alcanzando los objetivos propuestos y si el equipo puede cumplir con el total de las actividades estimadas.



👉 En este ejemplo es posible ver:

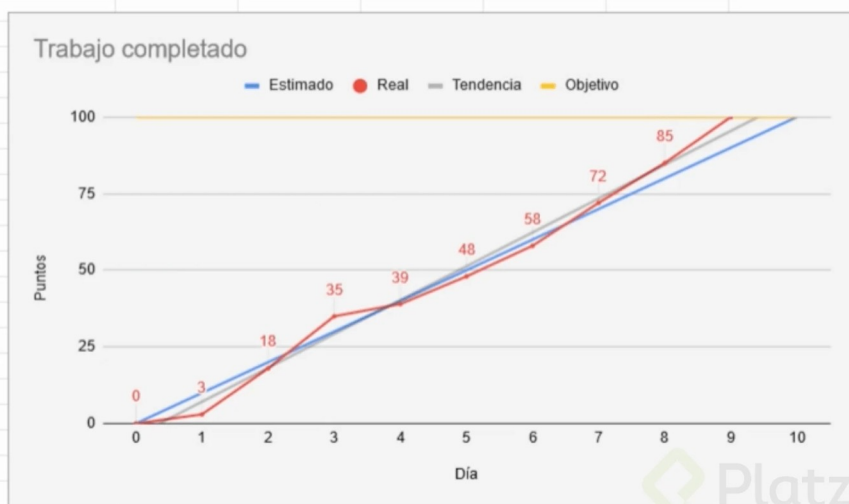
- Los puntos estimados para el Sprint: 100.
- El Sprint cuenta con 10 días.
- Se marcan dos líneas: una, la estimada por el equipo y dos, la real (trabajo efectivo del equipo).
- En la columna "completado" se muestra el avance del equipo.
- En la columna "pendientes" se puede ver lo que todavía no se ha realizado.
- A medida que se registran los avances, la línea gris permite conocer la tendencia.
- Al finalizar el Sprint y completar los datos, se pueden ver las diferencias entre el trabajo estimado versus el ejecutado.
- En tal caso que el progreso real no llegue a cero es posible definir o estimar las tareas que faltan para el siguiente Sprint.

## Bourn Up Chart ♥

Esta gráfica permite ver el trabajo que se ha completado durante el Sprint. Se marca una línea adicional (de color amarillo) la cual representa el ideal de puntos que se deben completar o el objetivo al que se quiere llegar.

En el momento en que la línea roja alcance a la línea amarilla, se ha cumplido el objetivo del Sprint. Si esto sucede antes de tiempo estimado, es una oportunidad para que el equipo revise oportunidades de mejora o prepararse para la Revisión del Sprint (Sprint Review).

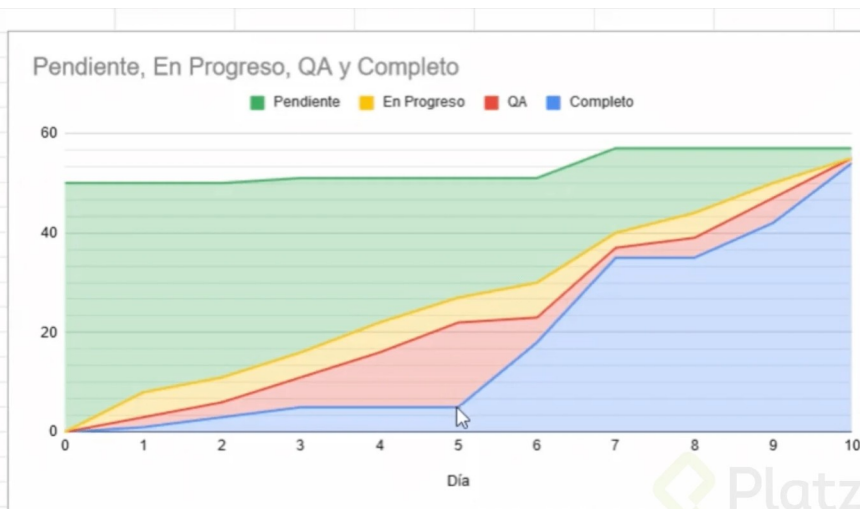
Puntos estimados para el sprint					
100					
	Completado diario		Completados sprint		
Día	Estimado	Real	Estimado	Real	Objetivo
0	0		0	0	100
1	10	3	10	3	100
2	10	15	20	18	100
3	10	17	30	35	100
4	10	4	40	39	100
5	10	9	50	48	100
6	10	10	60	58	100
7	10	14	70	72	100
8	10	13	80	85	100
9	10	15	90	100	100
10	10		100		100



## Gráfica de flujo acumulado 😊

En esta gráfica no se refleja el día a día del Sprint, sino que incluye el histórico de los Sprints anteriores.

Sprint	Pendiente	En Progreso	QA	Completo	Total
0	50	0	0	0	50
1	42	5	2	1	50
2	39	5	3	3	50
3	35	5	6	5	51
4	29	6	11	5	51
5	24	5	17	5	51
6	21	7	5	18	51
7	17	3	2	35	57
8	13	5	4	35	57
9	7	3	5	42	57
10	2	0	1	54	57



👉 En este ejemplo es posible observar:



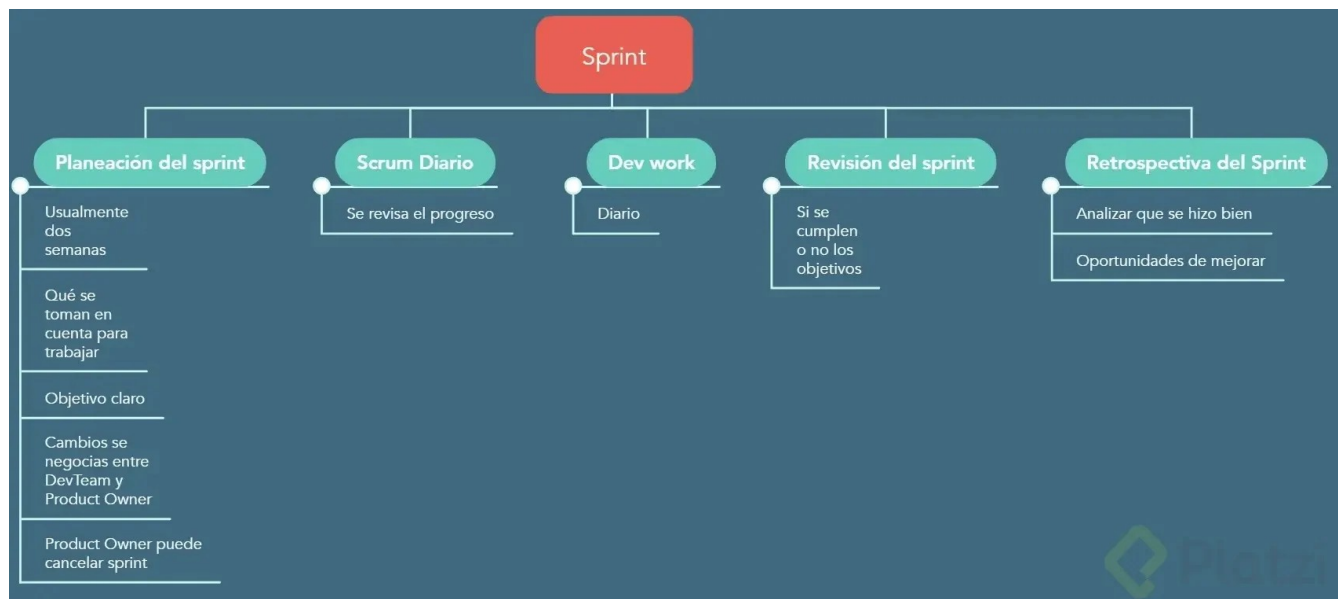
- Los estados del progreso de desarrollo (pendiente, progreso, QA (Quality Assurance) y completado).
- La línea de pendientes (verde) se va acumulando en el tiempo, ya que mientras el proyecto esté activo, se agregarán más historias en cada Sprint.
- Es posible analizar en cuánto tiempo se completan actividades que inician en un Sprint determinado.
- La gráfica permite identificar cuellos de botella y analizar las razones por las que se generan.

## ¿Qué es el Sprint en un proyecto Scrum?

El *Sprint* es el corazón de Scrum. Es un periodo determinado en el que se crea un incremento del producto. Puede durar entre 1 y 4 semanas (lo más común son 2 semanas), sin embargo, una vez se establezca la frecuencia del Sprint, esta siempre será la misma.

### Características del Sprint

- Cada sprint debe tener un objetivo claro.
- Cualquier cambio dentro del Sprint debe ser negociado entre el equipo de desarrollo y el Product Owner.
- Si los objetivos del Sprint quedan obsoletos, este podría ser cancelado. Aunque esto es poco común, solo el **Product Owner** tiene la autoridad para cancelar un Sprint.



### ¿Cuál es el ritmo del sprint?

El ritmo del Sprint refleja lo que va ocurriendo durante el proceso:

- **Planeación del sprint.** Allí se analiza qué se va a trabajar y qué se va a tomar en cuenta para ello.
- **Scrum diario.** Ocurre todos los días del Sprint para discutir el progreso hacia los objetivos.

- **Trabajo de desarrollo.** Va ocurriendo todos los días.
- **Revisión del Sprint.** Es la sesión donde se evalúa si se cumplen o no los objetivos.
- **Retrospectiva del Sprint.** En esta sesión el equipo analiza qué tan bien lo ha hecho y qué puede mejorar. Las mejoras identificadas se aplicarán en el siguiente Sprint.

En resumen



## ¿Qué es el Sprint Planning?

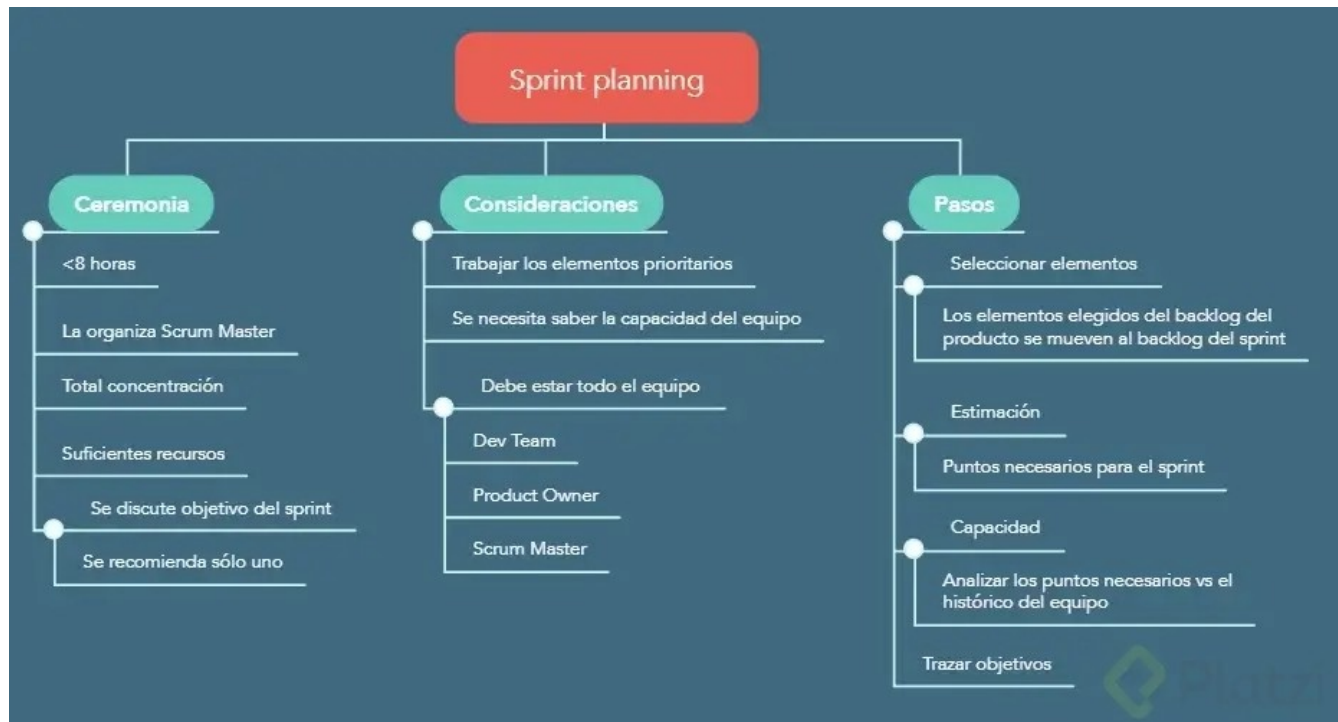
El Sprint Planning es la ceremonia de Scrum en la que se definen las Historias de Usuario que se van a desarrollar durante el Sprint. El Scrum Master es el encargado de organizar la sesión y en ella debe estar presente todo el equipo de Scrum: Scrum Master, Product Owner, Equipo de desarrollo.

## Características del Sprint Planning

- **Duración del Sprint Planning:** El Sprint Planning no debe durar más de 8 horas para sprints de 4 semanas. Normalmente, esta ceremonia puede tomar 1 hora para un sprint de 2 semanas.
- El Scrum Master es el encargado de organizar esta ceremonia. Se debe llevar a cabo en un lugar adecuado y que sea lo suficientemente espacioso para el equipo, puede incluir un pizarrón o la tecnología necesaria para llevar a cabo una videoconferencia con el cliente.
- El Scrum Master debe asegurarse de que todo lo necesario para la sesión esté disponible y funcione correctamente.

## Acuerdos del Sprint Planning

Nadie puede responder correos, llamadas o mensajes. Todo el equipo debe estar concentrado en el Sprint Planning y todos deben participar en el desarrollo de las actividades.



## Objetivos del Sprint Planning

Es importante que durante la sesión de planeación del Sprint se respondan las siguientes preguntas:

- ¿Qué puede entregarse al final del sprint?
  - Se trabajan los elementos más prioritarios de la Lista del Producto.
  - Se discute el objetivo a lograr en el Sprint. Se recomienda que sea un único objetivo.
  - Se necesita identificar la capacidad del equipo y conocer la velocidad de la última iteración.
- ¿Cómo se logrará hacer este trabajo?
  - Se analizan los elementos de la Lista de Producto seleccionados para el Sprint y se mueven a la Lista de Pendientes del Sprint.
  - Durante este proceso, se realiza la estimación de esfuerzo.
  - El Product Owner se encarga de aclarar cualquier duda con respecto a una Historia de Usuario o algún elemento del Backlog.
  - Puede haber invitados que aporten valor a la discusión de las funcionalidades (cliente, usuario final).

## Daily stand-up. Seguimiento de un proyecto Scrum

El Daily stand-up es una reunión diaria de no más de 15 minutos en la cual participa únicamente el equipo de desarrollo y donde se planifican las siguientes 24 horas de trabajo.

Esta sesión se lleva a cabo de pie con la finalidad de ser muy concretos y evitar que se extienda.

El Scrum Master es responsable de organizar la reunión, de preferencia a la misma hora y lugar para crear la costumbre. Otras personas pueden estar presentes, como el Product Owner, pero solo como observadores, sin participar.



### ¿Cómo hacer una Daily stand-up?

Esta sesión de seguimiento diario del proyecto permite optimizar la colaboración entre los integrantes del equipo. En ella se exponen los avances, la planeación y los posibles obstáculos de cada integrante del equipo de una forma concreta.

Para ello, se deben formular 3 preguntas:

- ¿Qué hice ayer?
- ¿Qué haré hoy?
- ¿Tengo algún impedimento? El Scrum Master ayuda a resolverlos.

La información que se comparte en una Daily stand-up debe estar enfocada exclusivamente a los objetivos del sprint.

El Scrum Master es responsable de asegurar que no haya impedimentos para que el equipo de desarrollo cumpla su objetivo.

Si se requieren más detalles sobre lo discutido, el equipo puede decidir que se lleve a cabo una reunión.

### **Beneficios del Daily Stand-Up**

- Mejora la comunicación del equipo de Scrum, gracias a las 3 preguntas que se responden (¿qué hice ayer?, ¿qué hice hoy?, ¿tengo algún impedimento?).
- Hace posible conocer y medir el progreso para cumplir el objetivo del Sprint.
- Ayuda a identificar rápidamente problemas o inconvenientes que pueden frenar el avance del equipo de desarrollo.

### **¿Cómo refinar Historias de Usuario?**

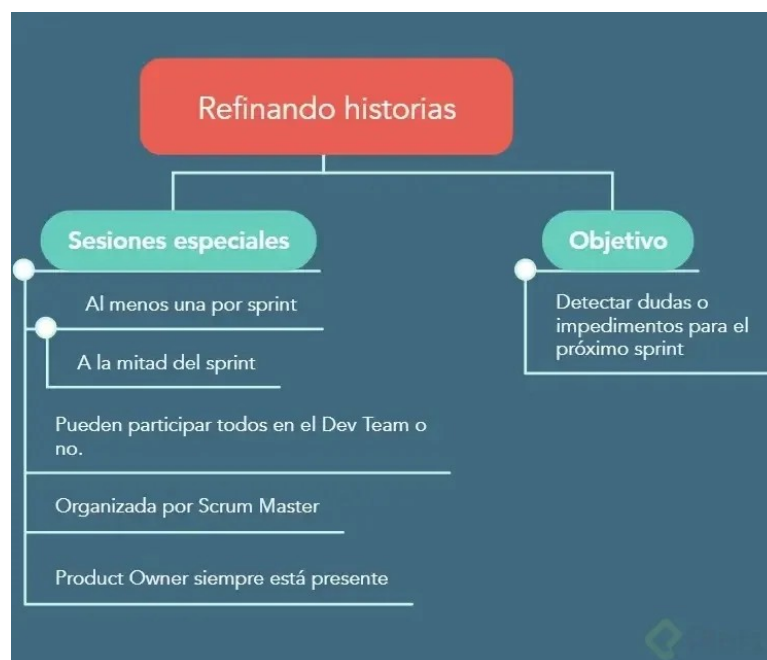
El proceso para refinar Historias de Usuario no está formalmente definido en Scrum. Sin embargo, es importante hacerlo.

### **¿Por qué es importante refinar Historias de Usuario?**

El objetivo de la sesión de refinamiento de Historias de Usuario es tomar las funcionalidades que se van a trabajar en el siguiente sprint y detectar dudas o posibles impedimentos, antes de comenzar el sprint.

En un proyecto de Scrum es necesario ver al futuro inmediato en el proceso de desarrollo. Dentro de los procesos de Scrum siempre debería haber en el Backlog funcionalidades listas para implementar.

La función del Product Owner es analizar y definir qué se va a trabajar en el próximo Sprint y para ello es importante refinar las historias y asegurarse de que tengan la información precisa para que se trabajen desde el primer día del Sprint.



## Características del proceso de refinamiento de Historias de Usuario

- Se recomienda llevar a cabo al menos una sesión para refinar las Historias de Usuario durante el Sprint.
- En esta sesión de refinamiento el Product Owner siempre debe estar presente.
- El Scrum Master es el encargado de organizar la ceremonia de refinamiento.
- Pueden participar los miembros del equipo de desarrollo. Sin embargo, no es necesario que todos ellos estén presentes.

## Sprint Review y Retrospectiva del Sprint

Al final del Sprint en un proyecto Scrum se llevan a cabo ceremonias como el Sprint Review y Retrospectiva que permiten identificar avances, logros y aprendizajes.

### ¿Qué es la Revisión del Sprint (Sprint Review)?

Esta ceremonia de Revisión del Sprint se lleva a cabo el último día del Sprint. Allí se muestran al cliente los avances de la iteración, el software que se ha desarrollado y su funcionamiento. No necesariamente tiene que ser una sesión, pueden ser varias reuniones.

### Características de la Revisión del Sprint

- La transparencia es importante. Esta reunión está enfocada en la comunicación con el cliente, es informal y abierta al diálogo, no es una sesión de seguimiento.
- Es facilitada por el Scrum Master y no debe durar más de cuatro horas para Sprints de un mes. Para sprints más cortos, se podría llevar a cabo en menos tiempo.
- Al final de la revisión se tendrá una lista de producto actualizada.
- La clave es mostrar el producto desarrollado al cliente, no hacer énfasis en presentaciones.

Después del Sprint Review se lleva a cabo una sesión de **Retrospectiva**, donde el equipo de Scrum analiza las oportunidades de mejora.

### ¿Qué es la Restrospectiva (Retro)?

La Retrospectiva es una reunión interna en la que solo participa el equipo de Scrum. Es facilitada por el Scrum Master y el cliente no está presente. Esta ceremonia se lleva a cabo en **un ambiente más relajado y el equipo puede expresar libremente lo que piensa** y dar feedback acerca de situaciones particulares.

### Características de la Retrospectiva

- Es una reunión positiva y productiva.
- No es una sesión para buscar culpas.



- No debe durar más de tres horas para un sprint de un mes. Para Sprints de 2 semanas, una Retrospectiva puede durar entre 15 min. a 30 min.

### **Elementos que se discuten en la Retrospectiva:**

1. Herramientas. Evaluar si se tiene todo lo necesario para cumplir el objetivo.
2. Relaciones. Identificar si la comunicación es fluida o hay problemas.
3. Personas. Analizar cómo es la interacción entre los integrantes del equipo.
4. Procesos. Determinar si los procesos se ajustan al desarrollo óptimo que se espera dentro del equipo.

### **Se sugiere que se respondan las siguientes preguntas:**

1. ¿Qué hicimos bien?
2. ¿Qué no hicimos tan bien?
3. ¿Qué podemos mejorar?

Con base en las respuestas a estas preguntas se identifican los elementos más importantes, se buscan soluciones y se crea un plan de acción para aplicar las mejoras.

## **Escalabilidad de equipos en Scrum**

Scrum hace posible la escalabilidad de equipos. Es capaz de funcionar con proyectos de 10 personas y de 100 personas. La forma en la que está ideado Scrum permite enfrentar proyectos de distintos niveles de complejidad.

Para lograr la escalabilidad de equipos existe un esquema conocido como **Scrum de Scrums**, el cual está compuesto por representantes encargados de coordinar los esfuerzos y elementos de trabajo de los diferentes equipos de Scrum.

### **¿Qué es Scrum de Scrums?**

Scrum de Scrums es una reunión donde frecuentemente se encuentran integrantes de todos los distintos equipos Scrum para coordinar esfuerzos.

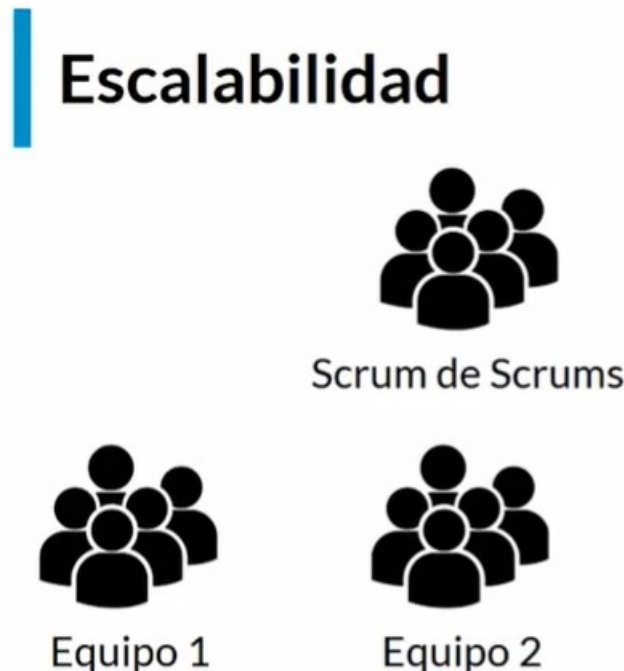
### **Características de la sesión Scrum de Scrums**

- No participa todo el equipo de Scrum, sino uno o dos representantes de cada equipo (Por ejemplo Product Owner, Scrum Master o el programador con más experiencia).
- Debe participar, al menos, un integrante de cada equipo.
- La reunión no debería superar 30 minutos.
- Es necesario que el representante de cada equipo tenga poder de decisión y que los compromisos adquiridos los adquiera en nombre de su equipo.

### **Preguntas sugeridas para la sesión Scrum de Scrums**

Estas son algunas de las preguntas que enriquecerán la sesión Scrum de Scrums y permitirán conocer los avances de cada equipo de trabajo:

1. ¿Qué hizo mi equipo de desarrollo desde la última reunión para lograr su objetivo del Sprint?
2. ¿Qué hará mi equipo de desarrollo antes de la siguiente reunión para lograr el objetivo del Sprint?
3. ¿Existe algún impedimento que esté afectando negativamente a mi equipo de desarrollo y que afecte el cumplimiento del objetivo del Sprint?
4. ¿Podría mi equipo de desarrollo afectar al objetivo del Sprint de los demás equipos?



### ¿Qué son las comunidades de práctica?

Las comunidades de práctica son grupos de personas que comparten un interés o una pasión por algo que hacen y aprenden a hacerlo mejor a medida que van interactuando.

En las comunidades de práctica se destacan 3 elementos:

- **Dominio**, qué nos gusta.
- **Práctica**, cómo lo compartimos y cómo lo practicamos
- **Comunidad**, a quién más le interesa.

### ¿Cómo organizar las comunidades de práctica?

Las comunidades se pueden organizar de dos formas:

- Basada en **roles**. Enfocada a la especialización.
- Basada en **tópicos**. Enfocada en crear una comunidad sobre un tema específico o particular, sin importar el rol.

# Comunidades de práctica

