

Estimación de tiempos y costos. Técnicas y herramientas

Estimación por Puntos de Historia: Guía para Equipos Scrum

Antes de continuar con la estimación por puntos de historia, primero aclararemos los siguientes perfiles :

- Product Manager
- Project Manager
- Product Owner

Diferencias entre Product Manager, Project Manager y Product Owner

Aunque sus nombres suenan similares, Product Manager (PM), Project Manager y Product Owner son roles distintos con responsabilidades diferentes. Confundirlos puede generar problemas de liderazgo, comunicación y ejecución en un proyecto.

A continuación, explicamos cada uno, sus funciones clave y cómo se diferencian, usando un ejemplo práctico.

Product Manager (Gerente de Producto)

El estratega: "¿Qué debemos construir?"*

El Product Manager es el máximo responsable del producto durante todo su ciclo de vida. Su misión es asegurarse de que el producto resuelva un problema real del cliente y entregue valor al negocio.

La primera pregunta que debe hacerse siempre es: ¿qué necesitan mis clientes?

Responsabilidades principales:

- Definir la visión y estrategia del producto.
- Investigar las necesidades del mercado y los usuarios.
- Priorizar funcionalidades según el impacto en el negocio.
- Supervisar desde el diseño hasta el lanzamiento y más allá.
- Coordinar con desarrollo, marketing, ventas y servicio al cliente.

En productos digitales (apps, plataformas, software), este rol se llama Digital Product Manager, ya que requiere conocimientos específicos del entorno digital.

Es el máximo responsable del producto y, por tanto, quien lo conoce mejor.

Project Manager (Administrador de Proyectos)

El ejecutor: "¿Cómo y cuándo se hará?"

El Project Manager es responsable de gestionar el proyecto: asegurar que se cumpla en tiempo, costo y alcance. No es dueño del producto, sino del proyecto temporal que lo crea.

Su función es gestionar recursos, equipos y cronogramas para entregar el proyecto en óptimas condiciones.

Responsabilidades principales:

- Planificar el alcance, presupuesto y cronograma.
- Asignar tareas y coordinar al equipo técnico.
- Gestionar riesgos operativos y tecnológicos.
- Comunicar avances a stakeholders.
- Aplicar procesos de control de cambios y calidad.

El Project Manager trabaja con metodologías tradicionales (como Waterfall) o ágiles, pero su enfoque es la ejecución eficiente del proyecto.

Se sabe de memoria los requisitos y deadlines de cada proyecto.

Product Owner

El especialista ágil: "¿Qué va primero en cada sprint?"

El Product Owner es un rol específico del marco Scrum, una metodología ágil. Es la persona que define qué se desarrolla y en qué orden, basado en el valor para el usuario.

Si existe un Product Owner, es porque se está trabajando con Scrum.

Responsabilidades principales:

- Mantener y priorizar el Product Backlog (lista de tareas).
- Decidir qué funcionalidades se incluyen en cada sprint.
- Ser el enlace entre el negocio y el equipo de desarrollo.
- Validar los resultados al final de cada iteración.
- Tener la última palabra sobre las características del producto.

A diferencia del Project Manager, el Product Owner no gestiona tiempos ni recursos, sino el contenido del trabajo.

Es una sola persona, según la Guía de Scrum.

¿Cómo funcionan juntos? Ejemplo práctico

Imagina que una empresa de e-commerce quiere lanzar una app móvil.

Rol	Función en este proyecto
Digital Product Manager	Define que la app debe permitir compras rápidas, tener una experiencia de usuario placentera y resolver el problema de navegar en móviles. Está presente antes, durante y después del lanzamiento.
Project Manager	Gestiona el presupuesto, coordina al equipo de desarrollo, asegura que el cronograma se cumpla y entrega la app en 6 meses.
Product Owner	Crea el backlog, decide que en el primer sprint se haga el login, en el segundo el carrito, y valida cada entrega con los usuarios.

Comparativa de roles

Característica	Product Manager	Project Manager	Product Owner
Enfoque principal	Estrategia del producto	Ejecución del proyecto	Valor del producto en sprints
Pregunta clave	¿Qué necesita el cliente?	¿Cómo y cuándo se hace?	¿Qué va primero?
Duración del rol	Todo el ciclo de vida del producto	Durante el proyecto (temporal)	Mientras dure el uso de Scrum
Metodología típica	Estrategia de producto	PMBOK, Waterfall, Ágil	Scrum (metodología ágil)
Responsable de	Visión, mercado, valor	Alcance, tiempo, costo	Product Backlog y priorización
Última palabra en	Qué funcionalidades tendrá el producto	Presupuesto y cronograma	Qué se desarrolla en cada sprint

¿Se solapan los roles?

Sí, dependiendo del tamaño de la organización:

- En empresas pequeñas o startups, una sola persona puede asumir los tres roles (por ejemplo, un Product Manager que también actúa como Project Manager y define el backlog).
- En empresas grandes, cada rol lo desempeña una persona o equipo diferente, lo que permite mayor especialización.

¿Quién tiene la última palabra?

Depende del tema:

- Funcionalidades del producto: Product Manager
- Presupuesto o plazo de entrega: Project Manager
- Prioridad en el sprint: Product Owner

Lo ideal es que los tres perfiles colaboren estrechamente para que el trabajo fluya y el resultado sea excelente.

Aunque comparten objetivos comunes, estos tres roles cumplen funciones distintas:

- El Product Manager piensa en el valor del producto.
- El Project Manager cuida del éxito del proyecto.
- El Product Owner garantiza que el equipo ágil trabaje en lo más importante en cada iteración.

¿Qué es la estimación por puntos de historia?

La estimación por puntos de historia es una técnica utilizada por los equipos Scrum para medir el esfuerzo relativo necesario para completar una user story (una funcionalidad desde la perspectiva del usuario).

"No se trata de adivinar cuántas horas tomará, sino de comparar qué tan grande es una tarea frente a otra."

A diferencia de las estimaciones en tiempo (horas o días), los puntos de historia consideran:

- La complejidad técnica
- El volumen de trabajo
- El nivel de incertidumbre o riesgo

Este enfoque permite al equipo planificar sprints con mayor realismo y comprometerse solo con lo que puede entregar.

¿Por qué no estimar en horas?

En proyectos de TI, especialmente en desarrollo de software, el tiempo exacto es difícil de predecir. Factores como errores inesperados, cambios de requisitos o integraciones complejas hacen que las estimaciones en horas sean poco confiables.

Además:

- Las personas trabajan a diferentes ritmos.
- Ocurren situaciones como sale a vacaciones la persona que mas sabe de ese tema.

- Personas que no pueden participar porque calamidades
- Un desarrollador puede tardar 4 horas en algo que a otro le toma 8.
- La presión por cumplir con "horas asignadas" reduce la calidad.

Por eso, Scrum prefiere una medida relativa y colaborativa: los puntos de historia.

¿Qué incluye un punto de historia?

Factor	Explicación
1. Complejidad	¿Qué tan difícil es técnicamente? ¿Requiere conocimientos nuevos?
2. Volumen de trabajo	¿Cuánto hay que hacer? (ej: múltiples pantallas, bases de datos, validaciones)
3. Incertidumbre	¿Hay partes desconocidas? ¿Depende de un servicio externo mal documentado?

Ejemplo:

Una función simple como "iniciar sesión" puede valer 3 puntos, mientras que "sincronizar con Google Calendar" podría valer 8 puntos, aunque ambas tomen tiempo similar.

Escala de estimación: Secuencia de Fibonacci

Para reflejar mejor la incertidumbre, los equipos usan una escala no lineal basada en la secuencia de Fibonacci:

1 – 2 – 3 – 5 – 8 – 13 – 20 – 40 – Muy Grande – Imposible - Descanso

¿Por qué esta secuencia?

Porque entre más grande es la tarea, mayor es la incertidumbre. No tiene sentido discutir si algo vale 9 u 11 horas; mejor decir: "es un 8 o un 13".

Relación con el Sprint: Velocidad del Equipo

El propósito principal de los puntos de historia es ayudar al equipo a saber cuánto puede comprometerse en cada sprint.

¿Qué es la velocidad?

Es el promedio de puntos completados en sprints anteriores.

Ejemplo:

Sprint	Puntos completados
1	20

2	22
3	18

Velocidad promedio 20 puntos/sprint

Durante la planificación del siguiente sprint, el equipo seleccionará historias cuya suma de puntos no exceda su velocidad.

Ejemplo práctico: App de Tareas

User Story	Descripción	Puntos acordados
US-01	El usuario puede registrarse con correo y contraseña	5 pts
US-02	El usuario puede iniciar sesión y recuperar contraseña	3 pts
US-03	El usuario puede crear, editar y eliminar tareas	8 pts
US-04	La app envía notificaciones push antes de vencer una tarea	3 pts

Total 29 pts

Si la velocidad del equipo es de 20–25 puntos, US-04 debe posponerse o dividirse.

Beneficios de usar puntos de historia

Beneficio	Explicación
Estimación más precisa	Basada en experiencia y consenso, no en suposiciones individuales.
Fomenta la colaboración	Todo el equipo participa en la estimación.
Reduce la presión por tiempo	Se enfoca en el valor entregado, no en horas trabajadas.
Mejora la predicción de entregas	Con la velocidad, se puede proyectar cuándo se terminará el backlog.

Errores comunes

Error	Solución
Comparar velocidades entre equipos	Cada equipo tiene su propia escala. No son comparables.
Dejar que solo uno decida	La estimación debe ser grupal.
Cambiar de escala constantemente	Usa una escala estable (como Fibonacci).
Forzar al equipo a aumentar su velocidad	La mejora viene con madurez, no con presión.

Frase clave

Un punto de historia no mide tiempo, mide esfuerzo. Y el esfuerzo se comparte, no se impone.

Técnica recomendada: Planning Poker

El Planning Poker es la forma más efectiva y divertida de asignar puntos de historia en equipo.

Cómo funciona:

1. El Product Owner presenta una user story.
2. El equipo discute brevemente su alcance.
3. Cada miembro elige una carta con su estimación (sin mostrarla aún).
4. Todos revelan sus cartas al mismo tiempo.
5. Si hay desacuerdo, se discute hasta llegar a un ****consenso****.
6. Se anota el punto acordado.

El consenso no significa unanimidad, sino que todos aceptan la decisión.

La estimación por puntos de historia es una herramienta poderosa que permite a los equipos Scrum comprometerse con realismo durante cada sprint. Al centrarse en el esfuerzo más que en el tiempo, fomenta la colaboración, reduce la incertidumbre y mejora la calidad de las entregas.

Para un proyecto ágil exitoso, no basta con trabajar mucho: hay que trabajar inteligentemente, y los puntos de historia son una guía esencial para lograrlo.

¿Qué es un criterio de aceptación?

Un criterio de aceptación es una condición específica y medible que debe cumplir un entregable o funcionalidad del proyecto para que sea considerado válido y aprobado por el cliente o usuario final.

En otras palabras:

¿Cómo sabemos que este trabajo está bien hecho y cumple con lo esperado?

Es una herramienta clave para evitar malentendidos entre el equipo de desarrollo y el cliente. Define cuándo algo está "terminado" desde el punto de vista del negocio, no solo técnico.

Características de un buen criterio de aceptación

Característica	Explicación
Claro y específico	No puede ser vago. Debe decir exactamente qué se espera.

Medible u observable	Se debe poder probar o verificar (sí/no).
Enfocado en el valor para el usuario	Responde a la pregunta: ¿Esto resuelve su necesidad?
Acordado con el cliente	No se define solo por el equipo técnico.

Ejemplo práctico

User Story:

Como usuario, quiero recuperar mi contraseña para acceder a mi cuenta si la olvido.

Criterio de aceptación vago:

El sistema debe permitir recuperar la contraseña."

¿Cómo? ¿Por correo? ¿Por SMS? ¿Cuánto tiempo tarda? No se puede verificar.

Criterios de aceptación claros:

1. El usuario puede hacer clic en "Olvidé mi contraseña".
2. El sistema envía un enlace de restablecimiento al correo registrado.
3. El enlace expira después de 1 hora.
4. El usuario puede crear una nueva contraseña de al menos 8 caracteres.
5. El sistema confirma al usuario que la contraseña fue cambiada con éxito.

Ahora sí se puede probar cada punto.

Relación con el documento "Los primeros pasos en el desarrollo de un proyecto"

En el contexto de planificación:

Alcance. En este punto deberán conocer qué es lo que se incluirá y lo que se excluirá... recolectar los requerimientos... criterios de aceptación...

Esto muestra que definir cómo se validará el resultado es parte fundamental de gestionar correctamente el alcance.

¿Para qué sirve un criterio de aceptación?

- Evita el re-trabajo: Todos saben desde el inicio qué significa "listo".
- Guía las pruebas de usuario (UAT): Son la base para diseñar casos de prueba.
- Protege el alcance del proyecto: Si algo no estaba en los criterios, no debe entregarse sin un cambio formal.
- Facilita la aprobación del cliente: No hay ambigüedad sobre si el producto cumple o no.

¿Dónde se usa?

Herramienta / Etapa	Uso del criterio de aceptación
User Stories (Ágil)	Cada historia tiene sus propios criterios.
Especificación del proyecto	Para definir cómo se validará cada entregable.
Pruebas de aceptación (UAT)	Se verifican uno por uno
Control de cambios	Si cambia el criterio, es un cambio de alcance.

Un criterio de aceptación es una condición clara y acordada que define cuándo un entregable del proyecto es considerado válido y completo por el cliente o usuario.

No basta con que el sistema funcione. Debe funcionar como el cliente espera.

Definir estos criterios desde el inicio es clave para evitar malentendidos, reducir riesgos y asegurar el éxito del proyecto.

¿Que es un pivote en estimacion scrum?

En la estimación de Scrum, un pivote es una historia de usuario o elemento del backlog que el equipo selecciona como referencia para estimar el resto de las tareas de forma relativa. A diferencia de una estimación basada en horas, que puede ser inexacta, el pivote permite estimar la complejidad y el esfuerzo de las tareas en comparación con un estándar acordado por todo el equipo.

Características clave del pivote

- **Punto de referencia:** El equipo elige un elemento del backlog que considera representativo de un tamaño pequeño y sencillo de completar, y le asigna un valor bajo de Puntos de Historia (por ejemplo, 1 o 3).
- **Estimación relativa:** Las demás historias de usuario se estiman comparándolas con el pivote. Por ejemplo, una tarea que se percibe como el doble de compleja que el pivote podría recibir una puntuación de 6.
- **Acuerdo del equipo:** La selección y el valor del pivote son acordados por todo el equipo de desarrollo, lo que asegura una comprensión común y facilita el proceso de estimación.

Ejemplo de cómo se utiliza un pivote

- **Selección del pivote:** En la planificación del sprint, el equipo revisa el backlog y elige una historia de usuario como pivote. Por ejemplo: "Como usuario, quiero poder iniciar sesión en la aplicación con mi correo electrónico". El equipo decide que esta tarea representa una complejidad de 3 puntos.
- **Estimación comparativa:**
 - Una tarea más sencilla, como "cambiar el texto de un botón", podría ser de 1 punto.
 - Una tarea de complejidad similar al pivote, como "recuperar contraseña", podría ser de 3 puntos.

- Una tarea mucho más compleja, como "implementar la autenticación de terceros con Google", podría ser de 8 puntos.
- **Flexibilidad:** El pivote puede cambiar en cada sprint si el equipo lo considera necesario, dependiendo de la naturaleza de las tareas a estimar.

Ventajas de usar un pivote

- **Mayor precisión:** Al enfocarse en la estimación relativa en lugar del tiempo exacto, el equipo evita discusiones innecesarias y sesgos individuales sobre cuánto tiempo tomará una tarea.
- **Consenso del equipo:** Promueve una comprensión compartida de la carga de trabajo y fomenta el compromiso y la colaboración dentro del equipo.
- **Adaptación:** Permite al equipo ajustarse a lo largo del tiempo, ya que la estimación se basa en la capacidad actual del equipo y en la naturaleza de las tareas.