# *Product Manager*

## **Clase 7 - Producto: Backlog y Tableros**

Objetivo de la clase:

* Definir brevemente el Product Backlog.
* Realizar Estimaciones.
* Identificar un Tablero.
* Comparar Scrum y Kanban.

## **Product Backlog & Artefactos**

La palabra backlog significa “acumulación de algo, especialmente trabajo incompleto o cosas de las que debemos ocuparnos”. Backlog aparece siempre unida a Producto. “Product backlog” es el artefacto que sirve como contenedor de todo el trabajo que hay que hacer para desarrollar un producto. El Product Backlog se traduce en diversas user stories. Veamos cómo:

* En Scrum los requisitos se expresan como elementos del Product Backlog.
* El Product Backlog es una lista viva de requisitos funcionales y no funcionales priorizados por el/la PM tomando como parámetro su valor para el cliente. Es decir, los requerimientos y el orden de los mismos es cambiante a lo largo del proyecto.
* En Scrum, los requerimientos se van abordando en Sprints en el orden en que aparecen en el Product Backlog.

Dijimos que el backlog es un artefacto pero… ¿Qué significa eso? En Scrum, denominamos artefactos a información que el squad y los stakeholders utilizan para conocer a detalle el producto desarrollado y las acciones que se necesitan para producirlo.

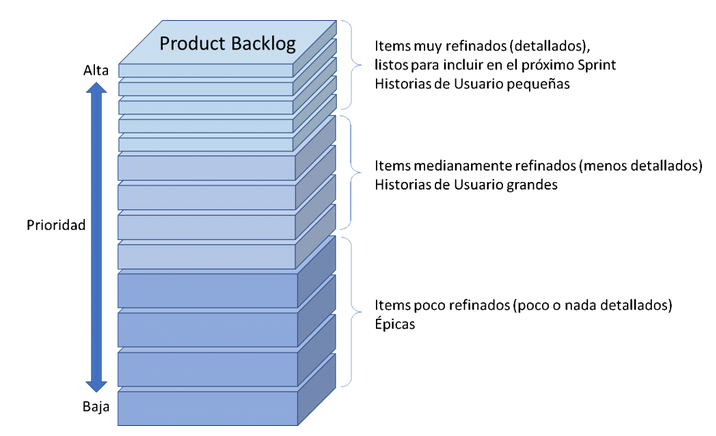
El Product Backlog nace de los requerimientos previamente establecidos. Como PM, nuestro deber es mantener el backlog saludable, revisarlo en cada iteración para asegurarnos de que la priorización es correcta y todo feedback ha sido incorporado.

Un Backlog de Producto se visualiza como una lista de tareas cuyos ítems son: Funcionalidades o features con historias de usuario y casos de uso; Tareas técnicas con piezas de arquitectura y configuración de entorno; Corrección de errores o bugs; y adquisición de conocimiento junto al análisis de opciones técnicas.


El Product Backlog no es una simple lista de tareas. Tanto éste como sus elementos (entradas o ítems), tienen una serie de características:

* **Documento vivo**: Está constantemente sujeto a cambios necesarios para el desarrollo del producto.
* **Valor agregado**: Está pensado en función de agregar valor al cliente tras diversas iteraciones.
* **Criterios de prioridad**: Los elementos se priorizan según su valor para el cliente.

La cuestión de **prioridad** es sumamente importante en la filosofía de las metodologías ágiles. El nivel de detalle de cada ítem del Backlog depende de su posición. Los que están ubicados más arriba estarán más detallados (refinados), para poder abordarse antes. Los que están ubicados más abajo contendrán menor detalle, pues se abordarán más tarde y es posible que antes se modifiquen o incluso se descarten.



## **Requerimientos, User stories & Épicas**

La especificación más conocida de requerimientos en metodologías ágiles es la de las Historias de Usuario o User Story. Las historias de usuario surgen del marco ágil eXtreme Programming (XP) creado por Kent Beck en el año 1999. Una historia de usuario describe la funcionalidad que aportará valor para el usuario y por ende, se incorporará al producto.

Una user story está compuesta por los siguientes componentes:

* 🙋‍♀️ Como **<Usuario de la funcionalidad>** - Para ayudarnos a identificar para quién estamos haciendo la funcionalidad y poder centrarnos en los usuarios y no en los solicitantes del requerimiento.
* 🎯 Quiero **<necesidad / requerimiento>** - Para dar claridad sobre lo que se espera, de manera simple y concreta.
* ✅ Para qué **<beneficio>** - Para entender cuál es el valor esperado al incorporar esta funcionalidad y así asegurarnos que cada Historia de Usuario tiene relevancia.

Como consumidora de la app, requiero que mi comida sea vegetariana para evitar comer carne.
Como consumidor del servicio, requiero validad de legumbres y vegetales para evitar almorzar siempre lo mismo.
Como usuario de la app de recetas, requiero variedad de recetas rápidas y sencillas para poder cocinar en el poco tiempo que tengo.

### **USER STORY & ÉPICAS**

Una Épica es una tarea de alto nivel no estimable basada en las necesidades/pedidos de los usuarios o clientes, la misma debe ser dividida en pequeñas tareas (llamadas User Story).

🏋️‍♀️ Ejemplo: Desarrollar una app de fitness (Épica)

Luego, al dividirla en funcionalidades, podremos crear nuestras User Story:

🤸‍♀️ Chat con un trainer o Registro de actividad diaria.

Cuando los proyectos están en sus inicios, es difícil que se vean todos sus detalles. Por ello, se comienza con la visión a alto nivel. De esta manera, las primeras User Story suelen ser bastante abarcativas y a estas historias se les llama Épicas.

Los aspectos claves para la definición de las User Stories en tableros son:

* Identificador: El identificador es el número que refiere a cada una de las User Stories. Facilita su búsqueda.
* Título: El título estará compuesto por el nombre de la User Story.
* Descripción: El detalle de lo esperado, lo más detallado y con menos incertidumbres posibles, nos brinda el contexto en el que se ubica la US.
* Criterios de discusión: Los criterios de aceptación permiten validar lo realizado. Se basan en una serie de requisitos que se deben cumplir para poder dar por terminada la US.
* Responsables: Aquí se indican las personas del squad que estarán a cargo de cada una de las User Stories.
* Discusiones posteriores: Aclaraciones de funcionalidad, cambios de scope al debatir el funcionamiento con el equipo, y toda la alteración que reciba de su definición original.

## **Sprints**

Son períodos de tiempo cortos y limitados en el que nuestro Squad trabajará para completar ciertas tareas y cumplir objetivos específicos. Son fundamentales en Scrum y Agile para entregar mejores productos. Cuando los sprints son muy largos, la complejidad y el riesgo pueden crecer mucho. La brevedad de tiempo permite generar más ciclos de aprendizaje y reducir riesgos.

*“Los Sprints permiten la previsibilidad al garantizar la inspección y la adaptación del progreso hacia las metas de producto”.* Scrum Guide

### **CEREMONIAS SCRUM**

Las ceremonias scrum son reuniones que tienen por objetivo mantener los controles mínimos y necesarios para hacer que los procesos funcionen. Existen 5 tipos: Sprint Planning, Daily standup o Daily, Sprint Review, Sprint Retrospective y Product Refinement. Veamos cada una de ellas:

* **Sprint Planning**: En esta ceremonia organizamos al equipo para el próximo Sprint, se inspecciona el Backlog y se eligen las user story a trabajar. Nuestro trabajo es presentar el backlog al equipo y asistir en las estimaciones de desarrollo, ayudando a clarificar, ordenar, re-negociar y llegar a acuerdos.
* **Daily**: Se realiza regularmente, preferiblemente a la misma hora y en el mismo lugar. En ella, cada integrante del Squad cuenta las actividades que realiza y qué avances y problemas tiene. La idea principal es detectar inconvenientes de manera temprana.
* **Sprint Review:** Aquí, buscamos el feedback de los stakeholders sobre lo desarrollado en el Sprint terminado. El objetivo es identificar oportunidades de mejora y decidir qué hacer en la siguiente iteración. Se debe vislumbrar el backlog con las tareas finalizadas y en proceso.
* **Sprint Retrospective**: Es una oportunidad del Squad para hacer mejoras. Se busca capitalizar el aprendizaje del Sprint que finalizó. Se debe hacer luego de la Review y antes del siguiente Planning: Inspeccionamos el último Sprint, planteamos posibles mejoras y cómo llevarlas a cabo y pensamos formas de mejorar la calidad.
* **Product Refinement:** Es opcional. Aquí presentamos las tareas que entrarían en los próximos Sprints. La idea es que las User stories sean entendidas por todo el Squad para obtener estimaciones acertadas y reducir la incertidumbre sobre las siguientes fases. El propósito fundamental: asegurarnos que nuestro Backlog está bien priorizado.

Elegir los items correctos para trabajar en cada Sprint es una decisión colaborativa entre Product Manager, Scrum Master y el equipo de desarrollo. ¿Cómo lo hacemos? 🤔 Para ello, utilizaremos herramientas de estimación.

## **Estimaciones**

Hay muchas formas de estimar el trabajo a realizar . Para ello, se deben tomar en cuenta ciertos factores que ayudarán a la toma de decisiones relacionadas con el desarrollo y el negocio. Aunque no es una ciencia exacta, al hacerla bien podemos tener una guía del nivel de esfuerzo que cada tarea tendrá en el equipo.

Algunas técnicas de estimación son:

* Tradicional: La estimación tradicional es la que implementa como unidad de medida el tiempo basado en horas reloj.
* T-Shirt Size: Es una técnica de estimación basada en asignar tallas de ropa (XS, S, M L, XL) al trabajo a realizar.
* Poker Planning: Utiliza como medida de estimación los story points, basados en la complejidad y esfuerzo del trabajo.

### **T-SHIRT SIZE**

Esta técnica ayuda al equipo a adoptar una estimación relativa disociando el valor de la estimación de un elemento con el número de horas equivalente necesarias para su implementación. Nota: No hay una convención escrita con respecto al tiempo que representa cada talla, la definición es relativa al proyecto y equipo.

Ventajas:

* Fácil de ejecutar.
* Fácil de entender. Especialmente, para grupos con menos experiencia

Desventajas:

* Más difícil de mapear los incrementos de forma cuantitativa.
* Cuanto más experiencia en los equipos, mejor adaptabilidad a otros tipos de estimación ágil.

### **POKER PLANNING**

Los Story Points indican el tamaño y la complejidad dado un User Story con relación a otro story que son parte del proyecto. Se puede buscar una User Story de “1” Story Point para definir la “base”. Los Story points estiman el esfuerzo.

lgunos equipos optan por tomarlos como medida de complejidad; pero también se pueden orientar para estimar el esfuerzo requerido de un elemento, incluyendo los riesgos e incertidumbre. Definimos un Story Point a partir de la siguiente ecuación:

**Story Points = Esfuerzo + Complejidad + Riesgo**

Los story points deben ser:

* Comprensibles para los clientes y desarrolladores.
* Comprobables con resultados y datos concretos.
* Del tamaño justo: Se recomienda que sean pequeños como para que el programador pueda crear varios en una iteración.

¿Cómo se calculan? Formalmente, se calculan por la sucesión de Fibonacci 🐌.

0, ½, 1, 2, 3, 5, 8, 13 – Siendo 13 la duración de un Sprint.

Por regla, todas las tarjetas deberían tener como mucho un 13 de estimación. Si algo requiere más tiempo, conviene partir el requerimiento en piezas más pequeñas y facilitar el seguimiento a las tareas.

Para empezar a hablar de horas, debemos considerar que el esfuerzo no es exactamente una unidad de tiempo. Sabiendo eso, podemos afirmar que:

* No existe una relación “lineal” de Story Points a Horas.
* Un Story Point es una herramienta de colaboración (Un Story Point es una medida que engloba: complejidad (inherente y accidental), tamaño, incertidumbre y habilidad/conocimiento sobre el problema planteado.

Se trata de una técnica de estimación que parte del consenso. ¿Cómo se realiza? Cada integrante del squad estima individualmente, pero no lo muestra al equipo. Una vez que ya hayan estimado todos, se “muestran las cartas” (estimaciones). Allí, se justifican las estimaciones y se vuelven a realizar hasta alcanzar el consenso.

**Nota**: No se debe estimar una tarea que no se entiende. Si quedan dudas sobre lo que debemos hacer, es el momento de preguntar y evacuar inquietudes. Algunas herramientas para realizar una estimación Poker Planning pueden ser Scrum Poker Online o Planning Poker Online.

### **HERRAMIENTAS DE GESTIÓN**

En ellas podrás encontrar el Product Backlog, Sprints y todo lo necesario para desarrollar y hacer seguimiento del producto. Algunas son Jira o Asana. **Recuerda**: La herramienta no es la metodología. No es imprescindible usar una de éstas para trabajar con metodologías ágiles.

**Resumen de lo visto en la clase de hoy:**

* Estimaciones
* Poker planning
* Herramientas de Gestión y Tableros.

#DemocratizandoLaEducación