

Historias de usuario

Tabla de contenido

Historias de usuario	2
Definición	2
Formato	2
Ejemplo.....	2
Historias de usuario y requisitos	2
Historias de usuario	2
3 Cs: Card, Conversation, Confirmation	2
Criterios de aceptación	2
Formato	2
Ejemplo.....	2
Epopeyas	3
Nombres	3
Épicas e historias de usuario.....	4
Formatos	4
Calidad de las historias de usuario	4
Regla INVEST	4
QUS Framework	4
Ejemplo 1.....	5
Ejemplo 2.....	5
Ejemplo 3.....	6
Historias de usuario	6
¿Cómo encontrarlas?.....	6
User Story Mapping	6
Impact Mapping	7
En el proceso de desarrollo	8
Product & Sprint Backlogs	8
Kanban.....	9

Historias de usuario

Definición

Una **historia de usuario** formalmente es una descripción de la funcionalidad esperada de parte de un sistema. Esta descripción está siempre expresada desde el punto de vista del usuario del sistema.

Formato

Si bien el trabajo original con el formato propuesto era más libre, simplemente un título, en la actualidad, el formato más aceptado es el que llamamos: “¿Quién, qué, por qué?”.

Como <actor>, quiero poder <función> de modo que/para así <razón>

Ejemplo

En un sistema de ventas, una posible historia de usuario podría ser:

Como **cliente**, quiero poder consultar los **productos disponibles** para así poder tomar una **decisión de compra**.

Como si fuera a ver las historias de usuarios simplemente describen qué es lo que los usuarios desean del sistema. **No profundizan en el detalle ni dan información adicional respecto a la funcionalidad esperada.** En este ejemplo no nos indica qué datos de los productos se van a informar. Tampoco se aclara qué pasa si no hay productos en el sistema.

Historias de usuario y requisitos

Es importante entonces destacar que las historias de usuario no son requisitos en el sentido estricto o tradicional del término. Algunos autores dicen que las historias de usuario son más bien punteros, o recordatorios de que se debe mantener una conversación con el usuario acerca de las funcionalidad deseada descrita en la historia.

[V / F] Las historias de usuario con útiles para definir el detalle de la funcionalidad esperada.

Historias de usuario

3 Cs: Card, Conversation, Confirmation

Esto nos lleva a un concepto llamado de las **3 Cs**, por las siglas en ingles de **Card**, **Conversation** y **Confirmation**. Al desarrollar las historias de usuario, se debe tener presente que hay 3 elementos que las componen:

- **Card:** La descripción de la historia propiamente dicha, que pone de manifiesto cuál es la intención del usuario con respecto a la utilización del sistema, que normalmente se registra en una ficha o en algún tipo de herramienta.
- **Conversation:** La conversación que debe mantenerse con el usuario para profundizar los detalles del comportamiento esperado.
- **Confirmation:** La confirmación que describe los criterios de aceptación que permitirán garantizar que se han cubierto todos los aspectos de la historia.

Criterios de aceptación

Formato

Los criterios de aceptación se especifican siguiendo el formato:

Dado que <contexto>, cuando <acción> entonces <consecuencias>

Ejemplo

Un posible criterio de aceptación para nuestra historia de usuario de consulta de productos disponibles podría ser:

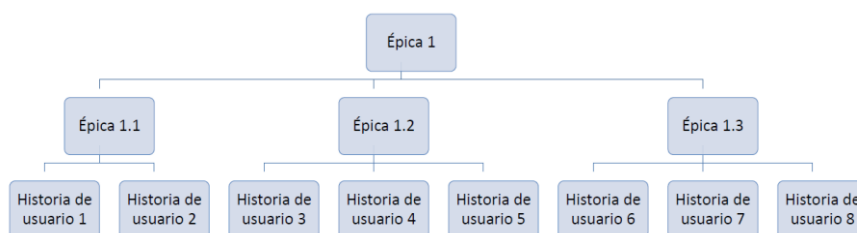
Dado que soy cliente
Cuando consulto productos disponibles
Entonces se me informa descripción, precio e inventario de los productos disponibles

Claramente se puede apreciar que los criterios de aceptación crearán posteriormente origen al caso de prueba que se podrían especificar mediante, por ejemplo, el lenguaje Gherkin. Sirve para completar la especificación del comportamiento esperado del sistema. Desde ese punto de vista, forman parte de la especificación de requisitos de software o son equivalentes en especificación de requisitos de software en un esquema más tradicional por supuesto.

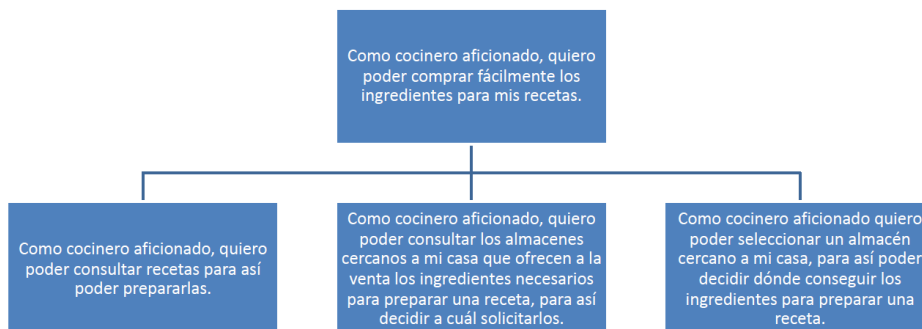
[V / F] Los criterios de aceptación ayudan a definir la funcionalidad.

Epopeyas

Ahora viene un problema que suele aparecer en cualquier proyecto de un tamaño no trivial, que es el manejo de numerosas historias de usuario. Para resolverlo, aparece el concepto de **épica o epopeya**. La idea es agrupar historias en relatos de mayor nivel, llamadas justamente épicas.

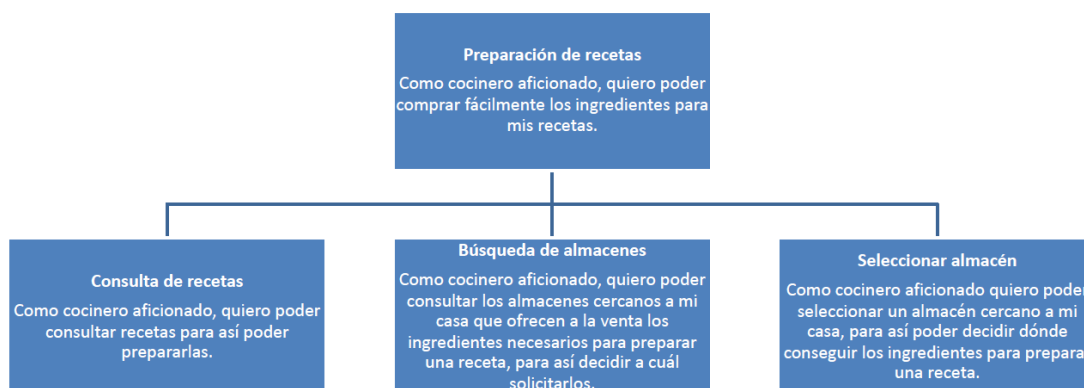


Por ejemplo, en este caso, una historia de usuario ha sido desagregada en 3 historias con mayor nivel de detalle:



Nombres

Lamentablemente, no hay convenciones acerca de cómo debe estar estructurada una épica, algunos pueden sugerir utiliza el mismo formato que se utiliza para las historias o simplemente describirlas con un título. También se le puede poner un título a las historias para simplificar la forma en la que nos referimos a ellas durante el desarrollo.



No es lo mismo hablar de la historia “consulta de recetas” que referirse a ella como la historia “Como cocinero aficionado, quiero poder consultar recetas para así prepararlas”.

Épicas e historias de usuario

Formatos

El gráfico anterior era simplemente con fines ilustrativos. Nadie construye un gráfico de ese tipo. Usualmente nos vamos a encontrar con una estructura como esta:

- **Gestión de compras**
 - **Consulta de productos**
 - Como cliente, quiero poder consultar los productos disponibles ordenándolos por categoría, para así decidir cuáles agregar a mi pedido.
 - **Selección de productos**
 - Como cliente, quiero poder agregar a mi pedido una o varias unidades de un producto mientras los estoy consultando, para así satisfacer mis necesidades de compra.
 - **Confirmación del pedido**
 - Como cliente, quiero poder confirmar mi pedido para así recibir los productos seleccionados en mi domicilio.

Inclusive hay herramientas que proponen agrupar historias de usuario y mostrarlas de manera similar a este formato que estamos viendo acá.

[V / F] Para solucionar el problema en la organización de una gran cantidad de historias de usuario se utilizan épicas.

Calidad de las historias de usuario

Regla INVEST

I	independiente	(de otras historias)
N	negociable	(una declaración flexible, no un contrato)
V	valiosa	(aporta valor al cliente/usuario)
E	estimable	(se puede estimar; negociable)
S	pequeña	(se puede desarrollar dentro de una iteración)
T	verificable	(se entiende lo suficiente como para ser probada)

A lo largo del tiempo han surgido criterios para determinar si una historia de usuario es correcta o no. Una de ellas es **INVEST**, que propone que cada historia debe ser **Independiente** de otras historias; **Negociable** ya que debe ser una declaración flexible, no un contrato solamente por como debe tener lugar en una conversación, hay cosas que por ahí tienen que ajustarse; **Valiosa** o valorable, tiene que aportar valor al cliente o al usuario; **Estimable**, es decir, que tiene un tamaño suficiente como para poder ser estimada; **Pequeña** (Small) que se puede desarrollar dentro de una iteración, esto tiene que ver mucho con el desarrollo iterativo e incremental o tiene que ver mejor dicho, con el desarrollo iterativo incremental; y **Verificable** o testeable, se entiende lo suficiente como para poder ser probado.

QUS Framework

Otro marco de referencia es el **Quality User Story (QUS)** que propone 13 atributos de calidad, agrupados en 3 categorías:

Calidad sintáctica <i>Estructura, sin considerar contenido</i>	Bien formada Atómica Mínima
Calidad semántica <i>Relación y significado de los elementos</i>	Conceptualmente acertada Orientada al problema No ambigua Libre de conflictos
Calidad pragmática <i>Interpretación subjetiva de la audiencia</i>	Oración completa Estimable Única Uniforme Independiente Completa

Algunos de estos criterios se aplican a historias individuales, y otros se aplican a un conjunto de historias de usuario.

Criterio	Descripción	Individual/grupo
Bien formada	Debe incluir al menos un rol y una acción	Individual
Atómica	Debe expresar una única funcionalidad	Individual
Mínima	Debe contener solamente un rol, una acción y un fin	Individual
Conceptualmente acertada	La acción expresa una funcionalidad y el fin una razón	Individual
Orientada al problema	No debe especificar la solución	Individual
No ambigua	Debe evitar el uso de términos que puedan llevar a múltiples interpretaciones	Individual
Libre de conflictos	Debe ser consistente con otras historias de usuario	Grupo
Oración completa	Debe ser una oración bien formada	Individual
Estimable	Debe poder ser estimada y planificada	Individual
Única	No debe haber duplicados	Grupo
Uniforme	Todas las historias de usuario emplean el mismo formato	Grupo
Independiente	No debe depender de otras historias; debe ser autocontenida	Grupo
Completa	No faltan pasos	Grupo

[V / F] Según el criterio INVEST que una historia de usuario sea pequeña facilita su estimación.

Veamos a continuación algunos ejemplo de aplicación:

Ejemplo 1

Quiero consultar los **productos disponibles**

Bueno, esta historia NO CUMPLE con el criterio de bien formada: Faltan el rol y el fin. Lo correcto habría sido expresarla siguiendo el formato: “quién, qué, para qué” que vimos anteriormente.

Una aclaración al respecto, este framework considera como opcional la indicación del por qué. Nosotros consideramos que, sin embargo, que el uso es obligatorio de el por qué, ya que nos permite justificar para qué es necesaria la funcionalidad deseada, para qué la necesitamos, cuál es el objetivo detrás de esa historia de usuario.

Ejemplo 2

Como **cliente**, quiero poder clicar en el botón “Consulta” para ver los **productos disponibles**

En este otro ejemplo se incumple el criterio de orientada al problema, ya que se dan detalles de implementación que son completamente irrelevantes en este contexto. Es más, nos está forzando quizás de manera demasiado anticipada, un tipo de solución posible. Yo en lugar de un botón podría decidir tener un link. Esa es una discusión que no debería tener lugar en este contexto.

Ejemplo 3

Como **cliente**, quiero poder consultar los **productos disponibles** para así luego **ingresar mi pedido**.

En este otro caso, se incurre en un error bastante común en la práctica. Se hace referencia incorrectamente a otra historia de usuario, en este caso la correspondiente al ingreso de pedidos. No hace falta hacer eso. Lo importante en la razón por la cual queremos consultar los productos en este caso probablemente para tomar una decisión de compra. **Cuando decimos para qué tenemos que especificar cuál es el valor que esperamos conseguir de esa historia de usuario.**

Historias de usuario

¿Cómo encontrarlas?

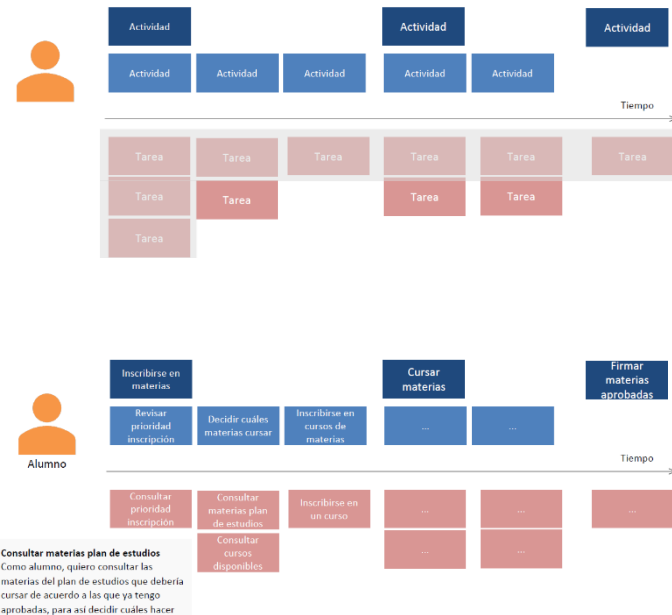
Uno de los problemas habituales que enfrentamos es cómo identificar historias de usuario. No siempre es fácil hacerlo, aún conociendo los conceptos teóricos. Lo que siempre hay que tener presente es que primero debemos entender cuáles son los límites del sistema que estamos analizando; y entendido este punto vamos a poder determinar quiénes son los que utilizarán el sistema. Luego, para cada uno de estos usuarios debemos identificar cuáles son sus objetivos, qué es lo que quieren lograr utilizar del sistema. Usualmente, los objetivos pueden incluir una jerarquía de subobjetivos. Al llegar al final de la jerarquía deberíamos estar en condiciones de encontrar una historia de usuario para cada subobjetivo.

[V / F] El *para qué* especifica cuál es el valor que se espera de la historia de usuario.

User Story Mapping

Un posible enfoque para descubrir historias de usuarios es el **User Story Mapping**. Un User Story Map es un modelo que describe las actividades que realizan a lo largo del tiempo los usuarios de una aplicación y su descomposición en subactividades y tareas. A partir de estas tareas podemos identificar las historias de usuario.

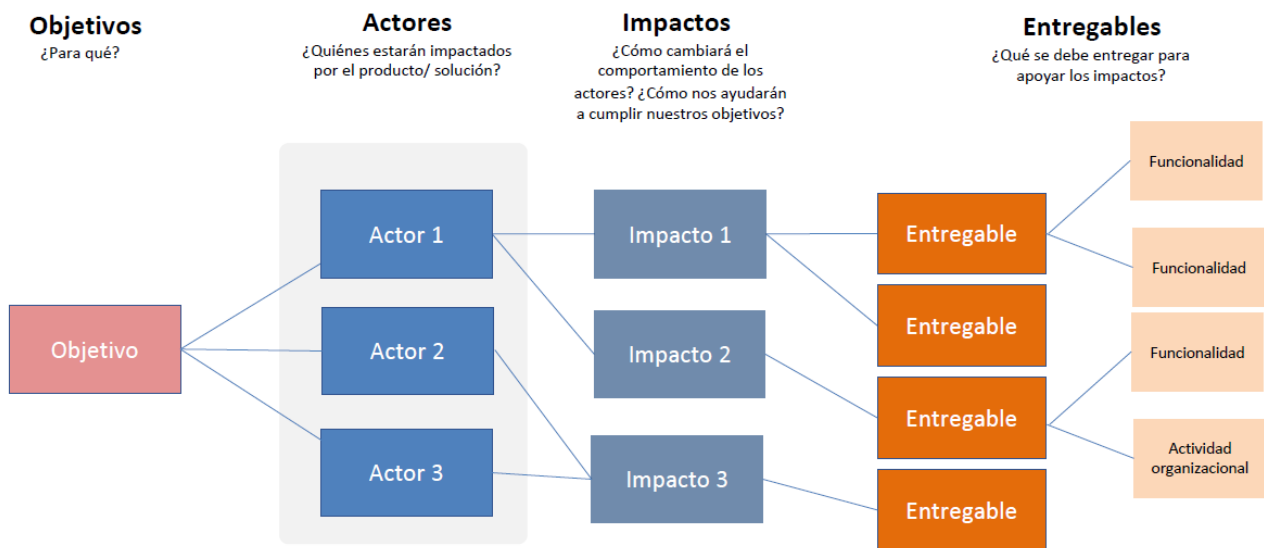
Por ejemplo, un alumno de una universidad podrá inscribirse en una materia, cursarlas, firmarlas, etc. Para inscribirse una materia revisará su prioridad de inscripción, decidirá cuáles materias cursar y luego se inscribirá en los cursos de las materias que haya seleccionado. **Para cada una de estas actividades habrá una tarea, como mínimo, que el sistema le ayudará a llevar a cabo.** En el ejemplo, consultar el plan de estudios, es una historia que contribuye a la actividad de decidir cuáles materias cursar.



[V / F] En la técnica *User Story Mapping* para cada una de las actividades hay al menos una tarea.

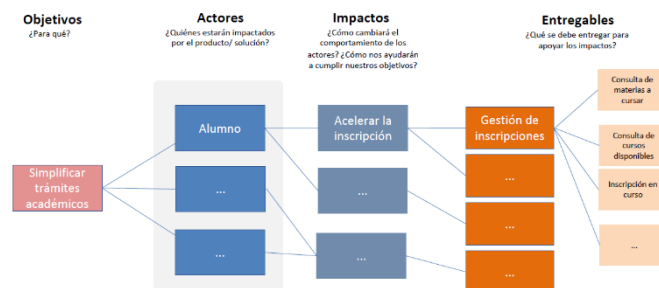
Impact Mapping

Otra alternativa, es emplear una técnica de planificación llamada **Impact Mapping**. Impact Mapping tiene como propósito alinear los equipos de trabajo con los objetivos de la organización.



Como se puede observar en el gráfico, a partir de la identificación de los objetivos (y estamos leyendo el gráfico de izquierda a derecha) que se definen para la solución a desarrollar, se identifican los actores afectados, cómo se pretende afectar el comportamiento de cada uno de esos actores, y finalmente, qué es lo que se deberá hacer para producir esos impactos (en el gráfico los entregables), que a su vez se pueden descomponer en productos, servicios funcionalidades.

En este ejemplo, el objetivo identificado para el sistema de gestión universitaria es simplificar los trámites académicos. Entre los actores impactados vemos al alumno. Uno de los impactos que se desea tener en los alumnos es acelerar la inscripción. Para ello, se entregará una funcionalidad de gestión de inscripciones que incluirá varias historias de usuario, entre ellas la consulta de materias a cursar, la consulta de materias disponibles y la inscripción en el curso.

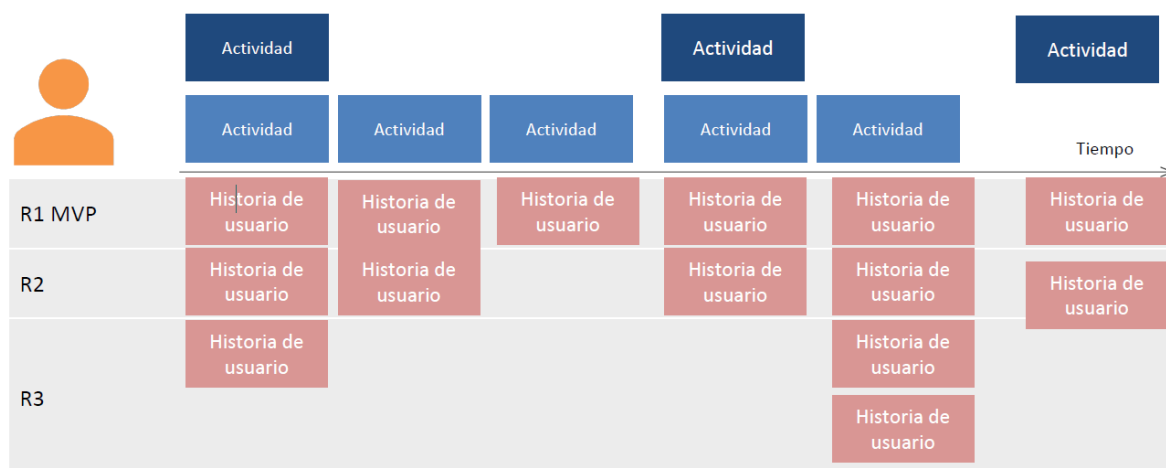


Este enfoque anterior **nos permite ir de lo general a lo particular, pero también recorren el camino inverso**. Muchas veces nos vamos a encontrarnos en esa situación. Esta técnica en particular tiene la ventaja de alinear funcionalidades con objetivos de la organización o con objetivos del proyecto, algo que muchas veces se pierde de vista.

[V / F] La técnica de *Impact Mapping* permite ir de lo general a lo particular y viceversa.

En el proceso de desarrollo

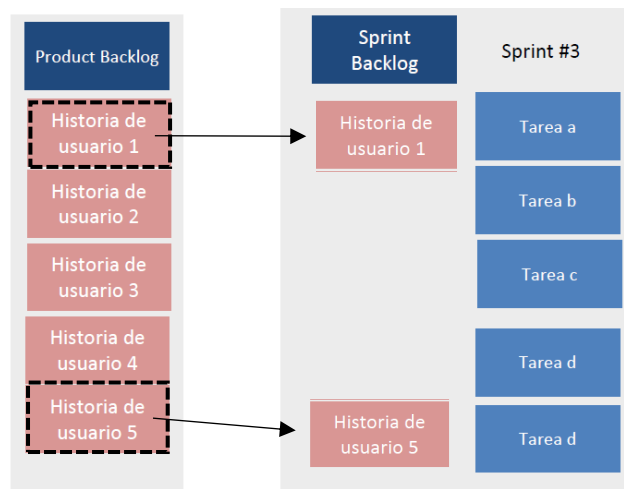
Volviendo a User Story Mapping, es importante destacar que, además de lo que ya hemos comentado, nos ayuda a realizar la planificación, nos ayuda a determinar qué historias de usuario vamos a asignar a cada uno de los releases.



Lo que vemos en el ejemplo es cómo se han analizado las historias en torno a cada uno de los releases que se han identificado, que en este caso son 3. Son esas cajas grises que están en forma horizontal. El primer release (R1), también llamado MVP o Producto Mínimo Viable, el release 2 (R2) y el release 3 (R3).

Product & Sprint Backlogs

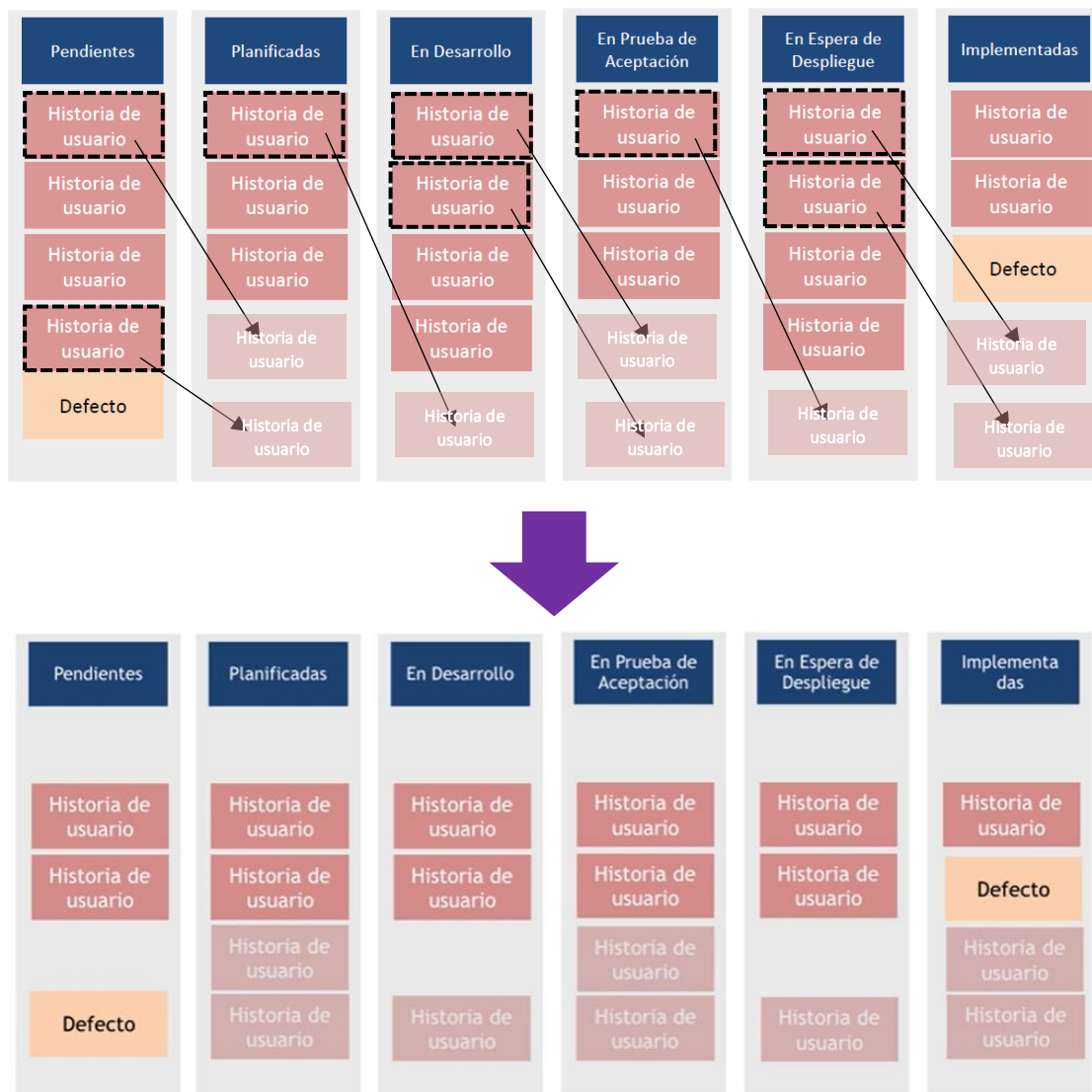
En Scrum, una metodología ágil que ya hemos mencionado, las historias de usuario se pueden incluir en un **Product Backlog**. Cada historia se estima empleando una técnica llamada user story points, que nos da una medida relativa de la complejidad que puede llegar a implicar, desarrollar y probar la funcionalidad asociada a cada una de las historias de usuario. En función de la capacidad disponible, y también con respecto al plan de entregas, el equipo asignado va a decidir qué historias de usuario desarrollar en cada iteración, que en Scrum recordemos que se llaman **Sprints**. A partir de ellas, se determinan las tareas que se deben ejecutar para implementar. Esta lista de tareas es el **Sprint Backlog**.



[V / F] La técnica de *Impact Mapping* facilita la planificación.

Kanban

Otro mecanismo muy interesante para organizar el trabajo que está basado en historias de usuario, es Kanban. Kanban es un sistema de planificación y organización del trabajo utilizado originalmente en las plantas de fabricación de Toyota y adaptado desde ya un tiempo al desarrollo de software. Su objetivo es ayudar a darle visibilidad al trabajo, a limitar el working progress o trabajo en simultáneo y mejorar el flujo de trabajo.



En el ejemplo vemos una tarjeta por cada historia de usuario y por cada defecto encontrado en el software. Esto es utilizado obviamente en el contexto de un esfuerzo de desarrollo de software, un proyecto de desarrollo de software. **Cada columna representa el estado en el cual se encuentran las historias y los defectos.** La idea es que la cantidad de tarjetas que puedan estar en cada estado esté limitada por las restricciones propias del equipo y la infraestructura disponible. A medida que cambiamos el estado de las historias, las vamos moviendo por el tablero.

[V / F] En el sistema *Kanban* cada columna representa un estado.