

Pivotal Tracker

Les presenté [Pivotal Tracker](#)! Es la herramienta de **agile management** que deberán usar para el TP. La idea es que les sirva para trackear el progreso del proceso de desarrollo iteración a iteración, cuales son las metas, cuánto y qué hace falta para alcanzarlas, quién es el responsable, etc... Para los que no la conozcan, les cuento un poco de que se trata.

Project Page

En la página de su proyecto en Pivotal (a.k.a **Project Page**) van a poder ver una serie de “listados” o “paneles” que son simplemente conjuntos de tareas o **stories**, como se llama a la unidad de trabajo en Pivotal. Estos paneles principales son:

- **Icebox:** contiene las stories que aún no se han priorizado. Típicamente las stories se crean allí y luego se mueven al Backlog o Current.
- **Backlog:** contiene las stories que se han priorizado. La prioridad es simplemente el orden en que se listan. Las stories en la cima del Backlog son las más importantes y se trabajarán primero.
- **Current:** contiene las stories que se trabajarán en la actual iteración o sprint.
- **Done:** contiene las iteraciones que han ocurrido en el pasado. Estas iteraciones contienen las stories que se han terminado y aceptado, en el orden en que fueron aceptadas.
- **My Work:** es una lista especial que muestra las stories que se te han asignado, que estás actualmente trabajando o que están listas para ser aceptadas.

Stories

Una **story** representa una tarea concreta. Existen varias formas de crear una story y lo único que necesitan definir para su creación es el título. Todo lo demás es opcional y pueden agregarlo o modificarlo luego. La idea es que el título sea un one-liner concreto acerca de un requerimiento, como puede ser: **Un usuario debe poder desloguearse**. Para especificar más detalles tienen el campo descripción.

Una story tiene asociado un **requester**: en general, alguien que representa los intereses del cliente y que tiene poder de decisión sobre los requerimientos y el criterio de aceptación; y un **owner** : el encargado de la implementación de la story (normalmente un developer).

Las stories pueden estimarse utilizando un sistema de puntos. El puntaje representa una medida de la complejidad y riesgo de la tarea. Si se ponen a investigar un poco verán que existen diferentes sistemas de puntaje, para unificar criterios usaremos todos la misma escala, llamada de Fibonacci. Los valores posibles

son 1,2,3,5 y 8. Como también verán una story de 8 puntos generalmente es un indicador de que esa story es muy grande y debería dividirse en stories más sencillas.

Las stories facilitan discusiones a través de comentarios. Los comentarios permiten dejar un rastro del “train-of-thought” de una story. Pueden ser editados por su autor (o un project owner), pueden incluir adjuntos. Todos los miembros del equipo pueden elegir recibir notificaciones via email o en sus dispositivos móviles. Para seguir una story, pueden usar el checkbox de **Follow this story** de la vista expandida de una story y recibirán notificaciones de todos los comentarios posteados a las stories que siguen, a menos que deshabiliten las notificaciones en la página de Notification Settings. Para hacer que alguien siga una story mencionenlos con **@name** en un comentario en la story.

Tipos de Stories

Existen 4 tipos de stories:

- **Features:** son stories que agregan valor al proyecto. Son las únicas que en general se estiman. Se puede cambiar la configuración para que los otros tipos de stories también se pueda estimar pero no suele recomendarse. Ejemplo: `Agregar botón de Contacto en la página principal`, `Permitir borrar items del carrito de compra`.
- **Chores:** son stories que son necesarias pero no tienen un valor directo u obvio para el proyecto (desde el punto de vista del cliente). Ejemplo: `Actualizar certificados SSL`, `Actualizar librería de networking`.
- **Bugs:** representan comportamiento no esperado.
`El botón de login tiene un tamaño incorrecto/está mal posicionado`,
`La fecha de expiración no puede ser en el pasado`.
- **Releases:** son marcadores o `milestones` que permiten trackear el progreso hacia las metas. Es posible especificarles fechas límites o “deadlines”. La idea es que todas las tareas para un release deben ir por encima del marcador del release. Pivotal Tracker mostrará este deadline como una línea gruesa negra, al final de cada iteración en donde la fecha caiga. Conforme vayan cambiando el scope del release (el número de puntos de las stories por encima del marcador de Release) y la velocidad del proyecto, el Release se irá moviendo de forma acorde, pero el la linea demarcadora del deadline permanecerá fija a la iteración particular. Si el marcador del Release se mueve más allá del deadline, se pondrá rojo. Esta es una excelente forma de administrar el scope de las iteraciones y experimentar con diferentes escenarios.

Labels y Epics

Existen otra formas de agrupar stories como lo son los `Labels` y las `Epics`.

Los **Labels** permiten agregar una categorización rápida a stories que estén relacionadas. Como por ejemplo las stories asociadas con el Login podrían llevar el Label `Login`, o las que se relacionan con la

funcionalidad de búsqueda, el Label `Search` . También son útiles para “marcar” stories indicando algo especial sobre ellas. Por ejemplo, las stories que tengan el Label `Necesita Diseño` permitirían rápidamente encontrar las stories que deben tener asignado un Diseñador (y asignar uno en caso de no tenerlo).

Las **Epics** se usan en general para agrupar funcionalidades de alto nivel, casos de uso o flujos que consistan de varias stories. Un ejemplo podría ser la Epic `Admin User` , que agrupa las funcionalidades de los tipos de usuario admins, o `New Users` , que agrupa las funcionalidades de los usuarios nuevos. Las Epics permiten al equipo planear, discutir y monitorear el progreso de features o flujos a un nivel de abstracción mayor que las stories. Permiten ver el estado de grandes funcionalidades con una sola mirada. Son similares a las stories y pueden ordenarse en su propio panel de forma independiente a las stories en el **Backlog** para visualizar de forma más obvia y sencilla las prioridades de alto nivel del equipo. Suelen utilizarse para colaborar en el diseño de features complejas, y facilitarle a los developers el acceso a recursos (como ser mock-ups) para una feature que abarca muchas stories. Para asociar una story a una Epic, hay que aplicarle a la story un tipo de label especial (`Linked Label`) que tiene color violeta, en contra-posición del resto de labels de color verde. Pueden arrastrar stories dentro de la lista de stories de una Epic para re-priorizarlas relativamente entre ellas. Si arrastran una story desde el **Icebox** o **Backlog** a la lista de stories de una Epic, además de priorizarla en relación a las stories de esa Epic, la asociarán automáticamente a esa Epic.

Workflow

De forma breve, el workflow más sencillo tiene la siguiente pinta:

1. Definir requerimientos y escribir las stories, en general en el **Icebox**.
2. Priorizar las stories en el **Backlog**
3. Discutir y estimar las stories
4. Empezar la siguiente story disponible en **Current** o **Backlog**.
5. Check-in después de la implementación para terminar (marcar como terminada) la story.
6. Hacer deploy al entorno de staging/demo/test, “hacer entrega/deliver” de la story
7. Aceptar o rechazar la story. (Feedback Loop)

Claramente, no todos los pasos son llevados a cabo por la misma persona (o al menos por el mismo rol, a veces una persona cumple varios roles). Como dijimos antes, es el **requester** el que tiene poder de aceptar o rechazar una story, en general es también quien la define y prioriza. Si una story se acepta, al final de la iteración pasará al panel **Done**. Si, por el contrario, se rechaza, pasa a un estado **Rechazado** y podrá ser re-iniciada (aparece un botón de **Restart**). Quien la rechazó deberá exponer una razón, que aparecerá en los comentarios de la story, y el/los owner(s) de la story recibirá una notificación.

Velocity, Team Strength and Volatility

La velocidad del proyecto (a.k.a **Project's Velocity**) es una medida del output del proyecto. Es el promedio

del número de puntos de stories (features) que fueron aceptados en las últimas iteraciones (entre 1 y 4, por default es 3). Pivotal Tracker calcula la velocidad automáticamente, y la usa para predecir cual es el número de iteraciones que se necesitarán para completar el **Backlog** de stories. Es posible modificar la velocidad calculada en su vista del proyecto y experimentar con diferentes escenarios. Pueden especificar un valor especulativo y ver como impacta en el **Backlog** o los releases futuros. Otros usuarios no ven este cambio de velocidad, y no se almacena, por lo que la velocidad vuelve a su valor calculado automáticamente cuando re-abren el proyecto.

En un principio, se utiliza una velocidad inicial que por default es 10. Se puede especificar en **Project Settings**.

La fortaleza del equipo (a.k.a. **Team's Strength**) es una medida de la capacidad del equipo de desarrollo para una determinada iteración. Pivotal Tracker lo usa para el cálculo automático de las iteraciones, en conjunto con la velocidad del proyecto. Puede usarse para representar que en una iteración el equipo de desarrolla se verá disminuido o fortalecido por la falta o la adición de miembros del equipo (ejemplo: vacaciones, redistribución de recursos para alcanzar un deadline).

La volatilidad (a.k.a. **Volability**) es una medida de qué tan consistente es la velocidad del proyecto. Mientras que la velocidad muestra lo que se hará, basado en el ritmo en el que se ha desarrollado el trabajo en el pasado, la volatilidad muestra qué tan confiable es. Una baja volatilidad dice que las estimaciones son precisas y que se puede estar seguro de lo que se hará en cada iteración. Es un porcentaje, calculado automáticamente usando las mismas iteraciones que para el cálculo de la velocidad del proyecto. Para más detalles acerca de la volatilidad ver [este enlace](#).

Observaciones:

- A veces las stories se mueven directamente a **Current** cuando las arrastran y tiran en el **Backlog**. Esto es porque el número de puntos que caben en la iteración actual está determinada por la velocidad del proyecto (**Project's velocity**).
- Las stories que ya se han comenzado permanecen en la cima de **Current**, pero pueden cambiarle las prioridades a las stories sin comenzar en cualquier momento arrastrándolas a cualquier lugar del panel **Backlog** o **Current**. También pueden poner stories sin comenzar de vuelta en el **Icebox**.
- Pueden cambiar el orden de los paneles como prefieran. Por default, el orden establecido de izquierda a derecha es el inverso al orden de desarrollo (**Done, Current, Backlog, Icebox**). Personalmente, yo prefiero que el orden de izquierda a derecha coincida con el de desarrollo (**Icebox, Backlog, Current, Done**), pero es algo totalmente subjetivo.

Les recomiendo vean los siguientes links, algunos videos cortos sobre los conceptos y el workflow de Pivotal Tracker, y se pongan a jugar un poco con la herramienta. Además, el segundo es un tutorial muy completo y con mucho material escrito que puede servirles de material de consulta.

- Introduction to the Concepts of Pivotal Tracker: <https://www.youtube.com/watch?v=bzCZysm5IG8>
- Getting Started: <https://www.pivotaltracker.com/help/gettingstarted>