**Trabajo Práctico Nro. 6**

**“ARMADO DE USER STORIES”**

**User Stories:**

**Las historias de usuario o User stories (US o stories) son la descripción de la funcionalidad de un software expresada en forma valiosa para las partes involucradas y útil para gestión de proyectos** de elevada incertidumbre.

Estas stories deben ser resultado de la colaboración y comunicación entre cliente y equipo. Evolucionan a lo largo de la vida del proyecto. Son como “trozos de funcionalidad que aportan valor”.

Las historias de usuario son la base del desarrollo de software incremental. Mike Cohn en su libro “User stories applied for agile software development” define una historia de usuario como un recordatorio para llevar una conversación futura con el cliente. Esta conversación futura se usará como base para confirmar la historia a través de su criterio de aceptación. Las historias de usuario tienen 3 aspectos fundamentales:

* **Descripción:** representa el requerimiento
* **Conversación:** detalle de la historia
* **Confirmación**: criterio de aceptación, lo que es aceptable para que el usuario considere la historia completa

Las historias deben estar escritas en el lenguaje del negocio de tal manera que el cliente pueda darles un valor y una prioridad. Las historias deben estar justificadas para cuantificar su beneficio desde el punto de vista del negocio...

La estructura básica de una historia de usuario es: “***As a… I want… so that…”***

* **Como/Siendo** (rol de usuario)
* **Quiero que** (una funcionalidad específica)
* **De modo que/Para que** (pueda generar un beneficio, de valor al negocio)

El user story (US o story) no entra en detalles “de cómo debe ser la implementación” ni engloba todos los posibles casos de uso. Simplemente define un concepto de manera que el programador que vaya a trabajar en ese US tenga una idea de por donde comenzar a trabajar. A medida que comienza la implementación del US y surgen las dudas, el programador colaborará con el **Product Owner** para aclarar esas dudas, y finalizar exitosamente el US, llegando a lograr el DONE.

**Consejos para escribir buenas US:**

* En general un story debe ser lo suficientemente **pequeño** como para que un programador pueda realizarlo sin problemas durante una iteración, y lo suficientemente **significativo** como para que añada valor de negocio al producto. Es ese equilibro el que diferencia un buen story de otro no tan bueno.
* Una vez escrito el story, el equipo lo lee, conversa sobre él para asegurarse de que todo el mundo entiende lo mismo y procede a estimarlo.
* Una vez ha sido estimado, está listo para ser introducido en la iteración y comenzar su implementación.
* Es importante que exista un test para cada story que se implementa. De manera que dichos test por un lado validan el correcto funcionamiento de la implementación, y por otro sirven de documentación para el mismo.
* Si al implementar un story nos vemos incapaces de realizar un test que lo cubra, probablemente estemos tratando de implementar un story demasiado grande. Estos stories reciben el nombre de **Épicos o Epics**, y deben ser subdivididos en otros stories más pequeños.
  + Una Epic, no es otra cosa que una gran User Story. Puede que una User Story, sea tan compleja y requiera tanto esfuerzo que no encaja dentro de una iteración, entonces para identificarlas se les llama así, “Epic”.
  + Un **Theme**, es un criterio de agrupación de User Stories. Se puede agrupar un conjunto de US acerca de la generación del reporte mensual de ventas y alli tendríamos un Theme. Como vemos un Theme es un termino de muy alto nivel

**NO son User stories:**

-Descripciones técnicas

-No representan un contrato de alcance cerrado y formal

-Una definición exhaustiva de los requerimientos

-Una descripción completa del sistema

**Como armar User Stories:**

* **Número:** debemos indicar un número único de US para poder identificarlo, permiten marcar precedencias de realización si usamos un Issue Tracker y debemos cargarlo para luego dividir las tareas y asignarlas
* **Título:** un breve texto para poder referenciar la US
* **Descripción:** describir y focalizarnos en el objetivo, en el por qué y no en el cómo. Usamos la estructura básica: Como… quiero…. Clarifica que se quiere conseguir, a quien aportara valor y porqué un US es útil. Este formato deja la posibilidad abierta de cómo debe implementarse una US, pueden aportar ideas de otras formas de obtener el objetivo de la US.
* **Prioridad:** valor numérico para priorización. Sin ninguna escala determinada, simplemente 100 vale más que 80. Otra opción es una escala de tres valores: 1-alta, 2-media y 3-baja
* **Estimación:** durante el Sprint planning el equipo asigna el esfuerzo en Story Points, necesarios para realizar el US.

**Número US:**

**TITULO:**

**DESCRIPCION:** estructura básica

**Prioridad:**

**Estimación:**

S

**Criterio de aceptación para el usuario**

* **Cuando …** (ejecuto una acción) **Espero …** (una reacción)

Repetir el criterio de aceptación hasta que el cliente considere que la historia esté completa.

Las pruebas de aceptación tienen una doble función, según se consideren como:

* Requerimientos: contribuyen a definir el alcance al expresar algún detalle relevante surgido de la conversación
* Test: demuestran la correcta implementación

Forman parte mínima de la definición de “done”:

* No son una descripción exhaustiva de todo requerimiento o test. No sustituyen la habilidad personal y el buen entendimiento del dominio/circunstancias
* Se pueden añadir/mejorar sobre la marcha, aunque mejor antes de codificar

***Ejemplo:***

*Usemos como ejemplo un* ***website de reclutamiento,*** *si identificamos candidatos a ofertas de trabajo como usuarios potenciales del sistema y hacer búsquedas de ofertas de trabajo como un requerimiento, una forma simple de definir una historia de usuario podría ser:*

User Story Nro: 18

Título: búsqueda de trabajo de candidato

Descripción: ***Como*** *candidato* ***Quiero*** *poder buscar ofertas de trabajo* ***De modo que*** *pueda encontrar puestos relevantes de acuerdo a mi experiencia profesional.*

Prioridad: alta

Estimación: a definir

Hasta allí hemos cumplido con la descripción y conversación con lo cual tenemos representado el requerimiento con los detalles de la historia. La historia también ha sido justificada desde el punto de vista del negocio pero falta un elemento muy importante para hacer que esta historia esté lista para llevarla al desarrollo, la confirmación, y esto lo definimos mediante el criterio de aceptación:

Criterio de aceptación para el usuario

|  |  |
| --- | --- |
| **Cuando** hago una búsqueda simple de ofertas de trabajo usando una categoría | **Espero**  - Ver todas las ofertas disponibles bajo la categoría y por cada oferta ver el puesto, nombre de la empresa y una breve descripción del puesto  - Ver la cantidad de postulantes por puesto  - Mostrar 100 puestos por página  - Poder navegar por el resto de páginas si se encontraron mas de 100 puestos |
| **Cuando** hago una búsqueda avanzada de ofertas de trabajo usando filtros | **Espero**  - Ver las ofertas disponibles de acuerdo a los filtros seleccionados y por cada oferta ver el puesto, nombre de la empresa y una breve descripción del puesto  - Ver la cantidad de postulantes por puesto  - Mostrar 100 puestos por página  - Poder navegar por el resto de páginas si se encontraron mas de 100 puestos |

El criterio de aceptación se repite hasta que el usuario cumpla con su definición de “terminado” o “done”.

Bill Wake autor del libro “eXtreme Programming Explored” sugirió el acrónimo **I.N.V.E.S.T.** para asegurar que una historia de usuario este definida correctamente. El acrónimo lo llevaremos al idioma castellano para concluir que debemos **I.N.V.E.R.T.ir** en historias de usuario, a continuación veamos el acrónimo:

|  |  |
| --- | --- |
| **Independientes** | Idealmente, no debe existir dependencia con otras historias, a veces esto es difícil pero deben hacerse los mejores esfuerzos |
| **Negociable** | Las historias no son contratos o documentos extensos de requerimientos, son descripciones específicas de funcionalidad |
| **Valorable** | Para los usuarios o compradores del producto. El valor es definido solamente por el cliente. |
| **Estimable** | Debe poder estimarse un grado de complejidad que determine el tamaño de la historia con respecto a otras. Esto depende en las conversaciones, que tan completa esté la historia, la experiencia del equipo y a veces de la tecnología a usarse para la implementación. Algunas veces para poder estimar se necesita un elemento de investigación como parte de la historia, en ese caso se divide la historia en dos partes, una dedicada a la investigación (*spike*) y la otra que es la funcionalidad en si que se va a post poner para una iteración futura |
| **Reducida** | Debe tener un tamaño reducido en la medida de lo posible pero no debe ser ni tan grande ni tan pequeña. Si es muy grande se consideran *epics*, por lo cual se tienen que partir en historias más pequeñas, si es muy pequeña agrupar con otras historias es lo mas indicado |
| **Testeable** | El criterio de aceptación debe ser capturado para facilitar las pruebas de aceptación |

**CASO A DESARROLLAR**

### Consignas

### *Roles: uno de los alumnos asumirá el rol de Product Owner (PO) y el otro el de Scrum Master. El resto del grupo son team-members. En el caso de ser un equipo de dos alumnos, deberán unirse con otro equipo para el desarrollo del TP. Ver roles en Nota al Final del TP [[1]](#endnote-1).*

### Business Analyst (BA): el JTP (profesor) actuará como Business Analyst, analista de negocios, en el caso que el PO necesite clarificar requerimientos

### Reunión de PO: Todos los PO del curso deben tener una reunión inicial para identificar los requerimientos y consignas, definiendo lineamientos generales. Pueden consultar al BA. Luego cada PO debe trabajar con su equipo.

### Revisión de requerimientos: Mientras los PO están reunidos los equipos con su SM deben leer los requerimientos generales y analizarlos para poder realizar preguntas.

### Desarrollar User Stories: el Product Owner deberá explicar/conversar los requerimientos para poder identificar todos los user stories a desarrollar. El PO siempre está disponible para resolver dudas, validar y para decidir qué y qué no se hace en cada momento, en función de los intereses del negocio. Debido a su inmersión dentro del equipo, es él quien decide qué y qué no se hace. El PO + el equipo deberán escribirlos según el formato especificado arriba, coordinando el Scrum Master. Indicar el titulo, descripción y prioridad de cada US. La estimación no es alcance del TP, a menos que queden tiempos para su desarrollo, usando Planning Poker.

### Desarrollar los criterios de aceptación de cada uno de los User Stories desarrollados, siguiendo lo explicado

### Entrega del TP completo

### Proyecto:

El presente proyecto plantea el desarrollo de un sistema de gestión empresas dedicadas a la venta al por mayor, utilizando para ello las últimas y más innovadoras soluciones desarrolladas dentro de la comunidad *open source*, minimizando el tiempo de desarrollo a la vez que maximizando la calidad del producto.

Para el desarrollo del sistema se contará con el apoyo de una empresa del sector que facilitará los datos y la experiencia necesarios para realizar un sistema ajustado al mundo real con el objetivo de implantarlo en empresas del medio, iniciando con una empresa de venta de telas al mayoreo, para el desarrollo de prendas, es decir fábricas de ropa casual.

El resultado de este proyecto es a su vez devuelto a la comunidad como software *open source* certificado por la OSI (*Open Source Initiative*) bajo Apache License Version 2.0. La vía escogida ha sido la creación de un proyecto con toda la información en el sito web de Sourceforge <http://oness.sourceforge.net>, comunidad dedicada a toda clase de proyectos *open source.*

### Funcionalidad general

Como requisitos generales del sistema se consideran principalmente auditoría, seguridad y la accesibilidad desde dispositivos móviles.

#### Auditoría

Los cambios realizados al sistema por los usuarios deben ser auditados. Para todas las altas, modificaciones y eliminaciones de datos se debe conocer el qué, quién y cuándo se han realizado.

#### Autenticación y autorización

Existirán varios roles de usuarios, y cada uno de ellos sólo podrá acceder a un subconjunto de los datos y las operaciones. Existirá un usuario administrador que tendrá acceso completo a la aplicación.

#### Los niveles de usuario

* Serán de consulta/actualización de datos (cliente-proveedor), operador, operador de nivel superior y administrador

#### Accesibilidad desde dispositivos móviles

### El sistema debe ser accesible desde cualquier localización utilizando tanto ordenadores portátiles como dispositivos móviles, principalmente PDAs. La localización puede ser tanto el propio local de la empresa como un lugar exterior.

### *Seguridad*

En cuanto a seguridad principalmente se debe considerar las características de gestión de autenticación y autorización transparente, incluyendo, pero no limitado a, protección de urls, métodos u objetos, cifrado de contraseñas, redirección automática a canales seguros como SSL o características *single sign on* (autenticación para todas las aplicaciones de la empresa a través de un punto único). También relacionadas con la seguridad están las características de auditoría que permiten conocer en todo momento los cambios realizados por los usuarios.

En cuanto la accesibilidad, desde el núcleo del sistema se deberá dotar a los distintos módulos de un mecanismo automático para mostrar un interfaz basado en el navegador del usuario, centrado en proporcionar un interfaz adaptado a dispositivos móviles como PDAs. Ha de tenerse en cuenta que aunque el número de sitios web que ofrece contenido para estos dispositivos crece rápidamente, la práctica totalidad de ellos requieren que el usuario introduzca otra dirección web distinta a la habitual, por ejemplo Google ofrece su interfaz para PDAs a través de <http://www.google.com/palm>, mientras que la solución utilizada en este proyecto evita la necesidad de que el usuario necesite conocer otra dirección ya que la elección se haría automáticamente.

### Funcionalidad del módulo de Gestión de contactos: “clientes y proveedores”

### *Objetivo General:* gestionar la información necesaria para realizar las operaciones comerciales.

Se tratará el concepto de “contacto”, es decir la persona u organización, que es una parte en una relación comercial en una empresa.

Cada uno de estos contactos tiene como atributo un nombre interno, que es el utilizado comúnmente para referirse a él, y atributos según sea:

* Persona: nombre, apellidos, cargo/rol en empresa, indicando el área donde trabaja
* Organización: razón social, nombre de fantasía
* Vinculo (relación de persona-persona, persona-empresa)
* Tipo de relación (cliente-proveedor).
  + En el caso de que un proveedor nos realice compras o un cliente se transforme en proveedor, deberán de tenerse una tercera opción de cliente-proveedor y agregar los datos faltantes según la relación.

Además de esto pueden tener información para contactar con ellos, como pueden ser:

* direcciones postales (destinatario, dirección, ciudad, código postal, provincia, país)
* números de teléfono (país, código de área, teléfono, extensión)
* número de celular (país, código de área, número)
* direcciones de correo electrónico (e-mails)
* direcciones web (url)

Cada uno de ellos con el propósito de esa información de contacto, por ejemplo

* correo electrónico del departamento de ventas-producción de la empresa/persona cliente/proveeedor
* dirección de la empresa
* dirección del depósito de entrega de productos
* dirección del depósito de retiro de productos
* dirección sucursal
* vinculaciones entre clientes-proveedores
* etc.

#### Creación de contactos

Para la creación de un contacto el usuario debe seleccionar si es una persona o una organización, y según su elección introducir los atributos correspondientes.

#### Visualización de contactos

Al usuario se le muestran los atributos correspondientes al tipo de contacto y la lista con las informaciones de contacto.

#### Borrado de contactos y su información

Cuando se considere que ya no son de relevancia los contactos podrán ser borrados por un operador de nivel superior cuando no se hayan realizado ningún tipo de transacción, pero si existen transacciones vinculadas en el sistema solo los inhabilitará, dejando el historial de transacciones. En esta caso deberá ingresarse la razón: quiebra, cierre de empresa, cambio de dueños o razón social, inhabilitación por falta de pagos, etc. En el caso de cambio de dueño o razón social, deberá indicarse como es el nuevo nombre para poder acceder a los datos anteriores.

#### Búsqueda de contactos

Se podrán realizar búsquedas y ordenamientos por cualquiera de los atributos, tanto de personas como de organizaciones, por ejemplo para agrupar pedidos a enviar, siendo accedido por usuarios de otras áreas.

#### Consulta de contactos propios

Los clientes-proveedores pueden tener acceso a consultar sus datos personales para poder comunicar cambios de los mismos. El cliente-proveedor seleccionará cualquier dato (como por ejemplo la dirección de destino (entrega/retiro), haciéndolo via web o por dispositivos móviles. Enviará el dato con la nueva información, pero debe ser aceptado por el operador en el sistema antes de actualizar la base de datos.

#### Añadir información de contacto a un contacto

Desde la visualización de un contacto se podrá añadir información de contacto, seleccionando el tipo de información (dirección postal, número de teléfono, correo electrónico o dirección web) se permitirá que el usuario introduzca los atributos correspondientes y vínculos entre contactos.

#### Visualización de información de contacto

Se mostrará el tipo de información, sus atributos y los propósitos de ella.

#### Modificación de información de contacto

Se podrán cambiar los atributos de cualquier tipo de información de contacto, excepto el nombre. En este caso, por ejemplo un error de tipeo, deberá ser un nivel de usuario operador superior.

1. **ROLES:**

   **SCRUM MASTER:**

   * Velar por que todos los participantes del proyecto sigan las **reglas y proceso de Scrum**, encajándolas en la cultura de la organización, y **guiar la colaboración** **en y entre equipos y con el cliente** de manera que las **sinergias** sean máximas. Esto implica:
     + Asegurar que la [lista de requisitos priorizada](http://www.proyectosagiles.org/lista-requisitos-priorizada-product-backlog) esté [preparada](http://www.proyectosagiles.org/replanificacion-proyecto) antes de la siguiente iteración.
     + **Facilitar las reuniones** de Scrum ([planificación de la iteración](http://www.proyectosagiles.org/planificacion-iteracion-sprint-planning), [reuniones diarias de sincronización del equipo](http://www.proyectosagiles.org/reunion-diaria-de-sincronizacion-scrum-daily-meeting), [demostración](http://www.proyectosagiles.org/demostracion-requisitos-sprint-review), [retrospectiva](http://www.proyectosagiles.org/retrospectiva-sprint-retrospective)), de manera que sean productivas y consigan sus objetivos.
     + Enseñar al equipo a **autogestionarse**. No da respuestas, si no que **guía al equipo con preguntas** para que descubra por sí mismo una solución.
   * **Quitar los impedimentos** que el equipo tiene en su camino para conseguir el objetivo de cada iteración (proporcionar un resultado útil al [cliente](http://www.proyectosagiles.org/cliente-product-owner) de la manera más efectiva) y poder finalizar el proyecto con éxito. Estos obstáculos se identifican de manera sistemática en las [reuniones diarias de sincronización del equipo](http://www.proyectosagiles.org/reunion-diaria-de-sincronizacion-scrum-daily-meeting) y en las reuniones de [retrospectiva](http://www.proyectosagiles.org/retrospectiva-sprint-retrospective).
   * **Proteger y aislar al equipo de interrupcione**s externas durante la [ejecución de la iteración](http://www.proyectosagiles.org/ejecucion-iteracion-sprint) (introducción de nuevos requisitos, "secuestro" no previsto de un miembro del equipo, etc.). De esta manera, el equipo puede mantener su **productividad** y el compromiso que adquirió sobre los requisitos que completaría en la iteración.

   El Scrum master debe ser capaz de:

   * Observar, escuchar, preguntar mucho y re-parafrasear para entender las necesidades, motivaciones y sentimientos de los otros, ponerse en su lugar antes de dar la propia opinión (si es realmente necesario que la dé). Es decir, evitar juzgar inmediatamente al otro y tener empatía.
   * Negociar, comunicar adecuadamente la información correcta en el momento correcto, adaptándola a las necesidades de la audiencia.
   * Enfocar al equipo, orientarlo para avanzar y cumplir con las expectativas del cliente, a la vez que cuidar la calidad del producto, sin dictar cómo hacerlo.
   * Motivar al equipo.
   * “Asume que [el equipo es el experto](http://www.proyectosagiles.org/equipo-team)” y así se lo hace saber, ellos son quienes conocen la mejor manera de realizar su trabajo y quienes tienen la responsabilidad de llevar a buen término el proyecto.
   * Ayuda al equipo a auto-organizarse para conseguir los objetivos del proyecto.
   * Ayuda a que su equipo avance, no se quede bloqueado, se mantenga focalizado en su trabajo, elimine ineficiencias, protege a su equipo de interrupciones externas y maximice su productividad.
   * Promueve la confianza entre el cliente y el equipo, y entre los miembros del equipo

   ## PRODUCT OWNER:

   Las responsabilidades del Cliente (que puede ser interno o externo a la organización) son:

   * Ser el **representante de todas las personas interesadas** en los resultados del proyecto (internas o externas a la organización, promotores del proyecto y usuarios finales [idealmente también debería ser un usuario clave] o consumidores finales del producto) y actuar como interlocutor único ante el equipo, con autoridad para tomar decisiones.
   * Definir los **objetivos del producto o proyecto**.
   * **Colaborar con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración**:
     + Participar en la [reunión de planificación de iteración](http://www.proyectosagiles.org/planificacion-iteracion-sprint-planning), proponiendo los requisitos más prioritarios a desarrollar, respondiendo a las dudas del equipo y detallando los requisitos que el equipo se comprometer a hacer.
     + Estar disponible durante el curso de la iteración para responder a las preguntas que puedan aparecer.
     + [No cambiar los requisitos que se están desarrollando en una iteración](http://www.proyectosagiles.org/ejecucion-iteracion-sprint), una vez está iniciada.
     + Participar en la [reunión de demostración de la iteración](http://www.proyectosagiles.org/demostracion-requisitos-sprint-review), revisando los requisitos completados.

   ## EQUIPO (Team):

   * Grupo de personas que de manera conjunta desarrollan el producto del proyecto. Tienen un objetivo común, comparten la responsabilidad del trabajo que realizan (así como de su [calidad](http://www.proyectosagiles.org/calidad-agilidad-cuarto-encuentro-agil-barcelona)) en cada [iteración](http://www.proyectosagiles.org/desarrollo-iterativo-incremental) y en el proyecto.

   Es un equipo **autoorganizado**, que comparte información y cuyos miembros confían entre ellos. Realiza de manera conjunta las siguientes actividades:

   * Seleccionar los requisitos que se compromete a completar en una iteración, de forma que estén preparados para ser entregados al cliente.
   * Estimar la complejidad de cada requisito en la [lista de requisitos priorizada](http://www.proyectosagiles.org/lista-requisitos-priorizada-product-backlog) del producto o proyecto.
   * En la reunión de planificación de la iteración decide cómo va a realizar su trabajo:
     + Seleccionar los requisitos que pueden completar en cada iteración, realizando al cliente las preguntas necesarias.
     + Identificar todas las tareas necesarias para completar cada requisito.
     + Estimar el esfuerzo necesario para realizar cada tarea.
     + Cada miembro del equipo se auto-asigna a las tareas.
   * Durante la iteración, trabajar de manera conjunta para conseguir los objetivos de la iteración. Cada especialista lidera el trabajo en su área y el resto colaboran si es necesario para poder completar un requisito.Pero debe siembre tenerse Humildad, evitar la prepotencia (que no es necesaria, la valía se demuestra realizando un trabajo excelente, el reconocimiento es una consecuencia que debe llegar por sí solo), tener una mente abierta a escuchar ideas diferentes de otros y flexibilidad para probar nuevas cosas.
   * Al finalizar la iteración:
     + [Demostrar](http://www.proyectosagiles.org/demostracion-requisitos-sprint-demonstration) al cliente los requisitos completados en cada iteración.
     + Hacer una [retrospectiva](http://www.proyectosagiles.org/retrospectiva-sprint-retrospective) la final de cada iteración para mejorar de forma continua su manera de trabajar.

   El equipo es **multidisciplinario:**

   * Los miembros del equipo tienen las habilidades necesarias para poder identificar y ejecutar todas las tareas que permiten proporcionar al cliente los requisitos comprometidos en la iteración.
   * Tienen que depender lo mínimo de personas externas al equipo, de manera que el compromiso que adquieren en cada iteración no se ponga en peligro.
   * Se crea una sinergia que permite que el resultado sea más rico al nutrirse de las diferentes experiencias, conocimientos y habilidades de todos. Colaboración creativa.

   Los miembros del equipo dedicarse al proyecto a **tiempo completo** para evitar dañar su productividad por cambios de tareas en diferentes proyectos, para evitar interrupciones externas y así poder mantener el compromiso que adquieren en cada iteración.

   El miembro del equipo ágil tiene que favorecer la **COMUNICACIÓN** y para ello tener las siguientes aptitudes:

   * Transparencia en las tareas que realiza y su estado, para que el resto del equipo tenga la información necesaria, que todos puedan colaborar y ayudarse a conseguir los objetivos de la iteración, evitando también que se realicen esfuerzos innecesarios.
   * Franqueza con el cliente sobre la situación del proyecto, para que pueda tomar decisiones basadas en lo que realmente está [hecho](http://www.proyectosagiles.org/lista-requisitos-priorizada-product-backlog#definicion-completado) y en la [velocidad del equipo](http://www.proyectosagiles.org/introduccion-estimacion-planificacion-agil#velocidad).
   * No hacerse dueño de conocimiento, si no compartirlo y ser capaz para enseñar.
   * Escucha activa, entender lo que le están explicando
   * Observar, escuchar, preguntar mucho y re-parafrasear para entender las necesidades, motivaciones y sentimientos de los otros, ponerse en su lugar antes de dar la propia opinión (si realmente es necesario ofrecerla). Es decir, evitar juzgar inmediatamente al otro y tener empatía.

   [↑](#endnote-ref-1)