**Institución Educativa Francisco Miranda**

**Trabajo hecho por:** Ely Milena Tobar Montalvo

Y

Sebastián Moreno Castaño

**Interpretación de Requisitos**

**Semana 5**

**Especificación de requisitos en entornos ágiles**



Media Técnica en Programación de Sistemas de Información

Politécnico Jaime Isaza Cadavid - 2020

# Metodologías ágiles de desarrollo

A medida que se ha avanzado en técnicas y herramientas para el desarrollo de productos de software se ha evidenciado que en muchos casos cuando se hacía entrega de los productos de software finales y se analizaba el uso de las funcionalidades típicas en el sistema del total de funcionalidades se utilizaban el 7% y habían funcionalidades con un 45% que nunca se usaban.

Esto da a entender que cuando se tiene un desarrollo incremental se pierde la capacidad de evidenciar con el cliente y/o usuario que tan a gusto está con el producto de software, ahora si se opta por un proceso iterativo e incremental donde se realizan entregas que evidencian pequeñas funcionalidades de todo lo que haría el software al final, se logran mejores resultados a nivel de las funcionalidades claves del producto a entregar.

En este punto las metodologías ágiles permiten adaptar la forma del trabajo a las condiciones del proyecto, logrando flexibilidad e inmediatez en la respuesta para acomodar el proyecto y su desarrollo a las circunstancias específicas del entorno. Así se logra mejorar la satisfacción del cliente dado que estará involucrado y comprometido a lo largo de todo el proyecto.

En cada etapa se informará al cliente de los logros y progresos del producto, con la visión de involucrarse directamente para sumar su experiencia y conocimiento, logrando optimizar las características del producto final con una visión detallada y completa de su estado.

Teniendo en cuenta estos aspectos también se logra evidenciar con las metodologías ágiles que hay una mejora en la motivación e implicación del equipo de desarrollo, ya que estas metodologías permiten a todos los miembros del equipo conocer el estado del proyecto en cualquier momento, así, los compromisos son negociados y aceptados por todos los miembros del equipo.

Existen varias metodologías ágiles que se adaptan para el desarrollo de productos de software, entre ellas:

1. SCRUM
2. Extreme Programming (XP)
3. Kanban
4. Design Sprint

Para nuestro caso estudio vamos a abordar los conceptos bajo la metodología ágil SCRUM

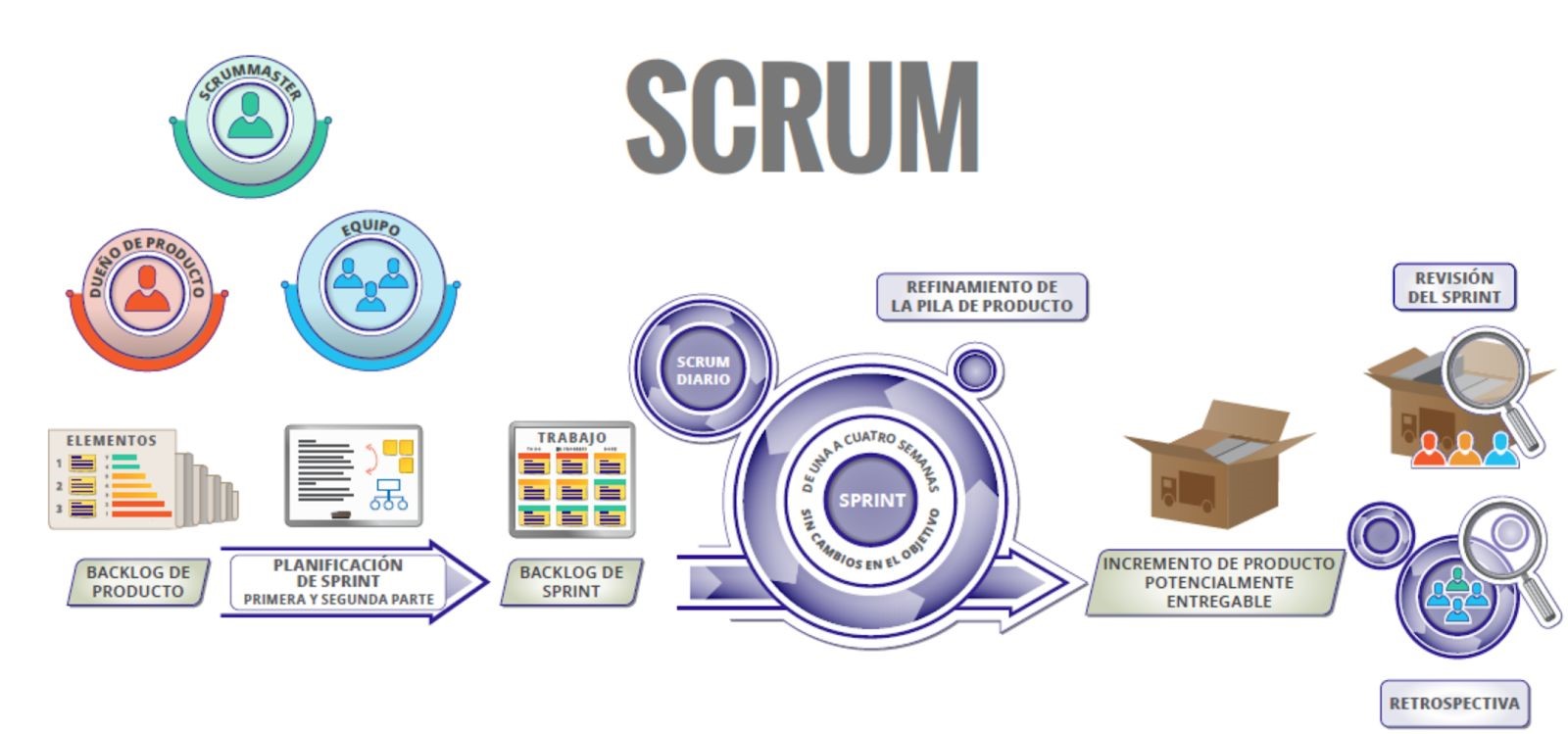
## Conceptos claves de SCRUM

Lo primero a tener en cuenta, es identificar los integrantes de un equipo básico de SCRUM, que se evidencia en estos tres roles básicos:

1. **Product Owner:** es el que tiene la visión del producto. Su rol principal, es transmitir lo que se ve a realizar por medio de documentos, “historias de usuario” también denominadas ​**User Stories.** ​El conjunto de estas historias de usuario constituye la lista de necesidades que forman lo que se denomina un ​**Product Backlog.**
2. **Scrum Master:** ​su rol se puede definir como el de un “facilitador”, tiene la misión de eliminar los obstáculos que puedan surgir entre los miembros del equipo, garantizando que la metodología de SCRUM se cumpla correctamente.
3. **Equipo de desarrollo Scrum:** ​son los encargados con distintos roles ​*(arquitecto, desarrollador frontend, desarrollador backend, arquitecto de infraestructura en al nube)* ​de implementar y ejecutar las historias de usuario definidas.

## Ciclo de vida - SCRUM

En la siguiente imagen vamos a evidenciar los diferentes elementos que están involucrados en la metodología Scrum, que nos ayuda a simplificar su ciclo de vida:



El ciclo de vida es el siguiente:

1. El ​**Product Owner**​ redacta las​ **User Stories**​ y las sitúa en el ​**Product Backlog**​.
2. A continuación, el ​**Product Owner** prioriza estas ​**User Stories** y ordena el ​**Product Backlog** ​en consecuencia.
3. El equipo ​**Scrum** se junta en la reunión de planificación del ​**Sprint**​, con el objetivo de establecer la lista de las ​**User Stories** ​que se tratarán durante el ​**Sprint**​. Esto forma el ​**Sprint Backlog** y a continuación se descomponen en tareas por el equipo de desarrollo.
4. Entonces el ​**Sprint**​ puede comenzar con una iteración de 2, 3 o 4 semanas.
5. El equipo se reúne diariamente para realizar el Scrum Diario.
6. Como consecuencia del **Sprint,** obtenemos un producto potencialmente entregable que forma parte de una demostración durante la revisión del ​**Sprint.** 7. El ciclo termina con la retrospectiva del ​**Sprint**​.

Dentro de nuestro product backlog (pila del producto) debemos identificar tres aspectos claves que van desde la una prioridad bajo hasta una más alta descritas a continuación.

## Épicas

Las épicas son funciones o actividades de alto nivel que abarcan Sprints o incluso Release. por ejemplo:

* Procesar pago de productos
* Administrar las cuentas del cliente

El Product Owner trabaja con los interesados y el equipo para crear épicas que aborden los objetivos deseados, logrando así definirse antes del plan de release.

## Características

Las características son expresiones tangibles de funcionalidad, pero aún demasiado grandes para ser construidas, por ejemplo:

* Como comprador, quiero configurar una billetera móvil para que pueda pagar las compras a través de NFC.
* Como cliente del banco, quiero buscar una sucursal para poder depositar cheques.

Estas se deben definir antes del plan de releases y se descomponen en historias más pequeñas.

## Historias de usuario

Son funcionalidades que están listas para ser construidas por el equipo de desarrollo, por ejemplo:

* Como cliente del banco, quiero buscar una sucursal cerca de mi ubicación actual para poder minimizar el trayecto.
* Como comprador, quiero agregar una cuenta a mi billetera móvil para agregar fondos.

Estas historias de usuario se definen entre el Product Owner y el equipo de desarrollo y deben ser refinadas antes de la planeación de cada sprint.

Cuando vamos a crear y diseñar nuestras historias de usuario se parte del concepto de las 3 C, donde debemos tener:

* Card (Tarjeta)
* Conversación
* Confirmación

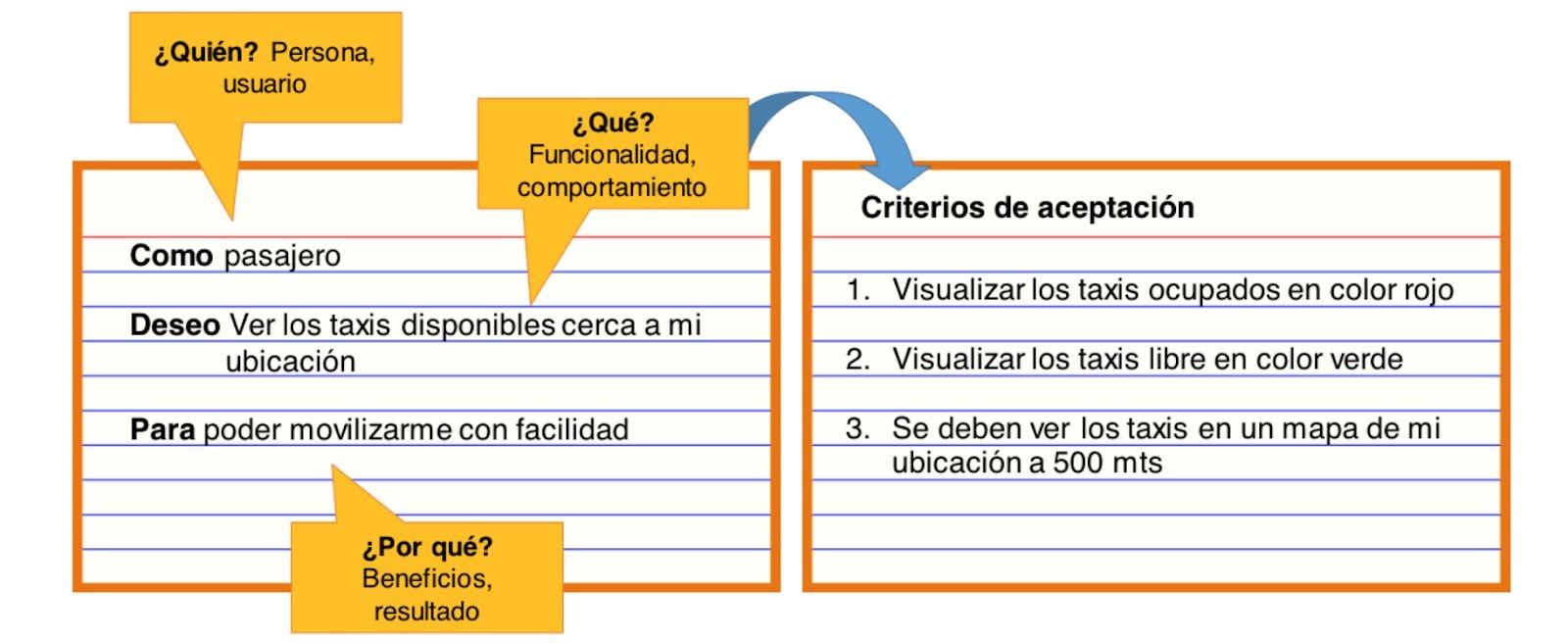
De esta forma una historia de usuario debe cumplir con la siguiente estructura:

**Como un** ​ **<**​ **Quién>**

**Quiero <**​ **Característica>**

**Para <**​ **Beneficio>**

Observemos el siguiente ejemplo:

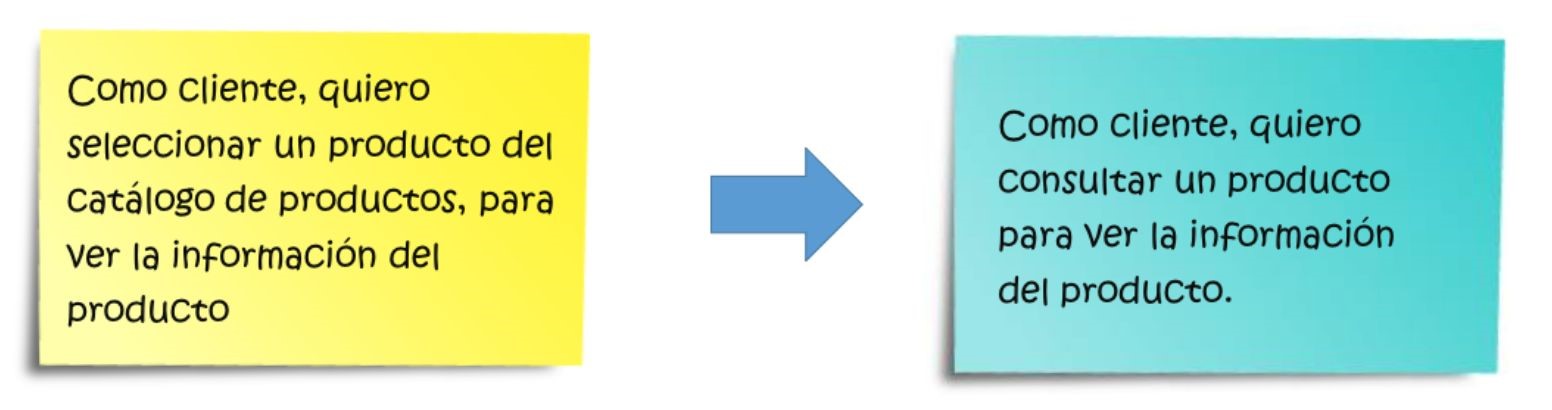


## Criterio INVEST

Las historias de usuario deben cumplir ciertas normas que nos ayudan a que queden bien identificadas y aporten valor a la funcionalidad específica que va a abordar.

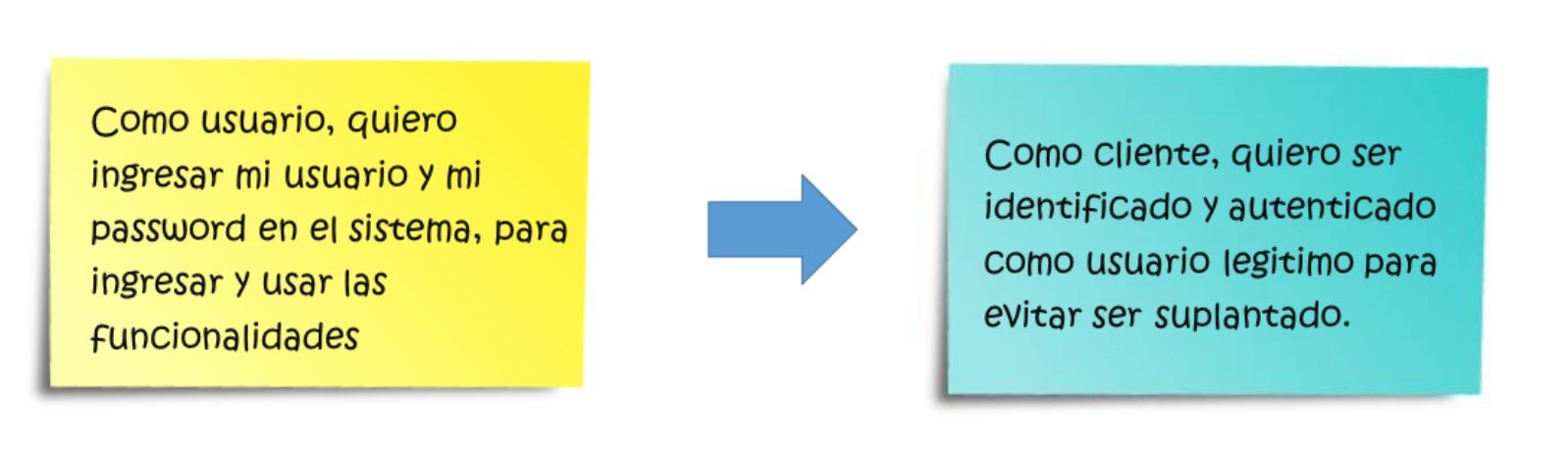
### Independent (Independiente)

Cada historia de usuario debe ser independiente una de otra, esto nos permite trabajar en las historias de alto valor de manera inmediata y no tiene que implementar otras historias (con menos valor) debido a la interdependencia.



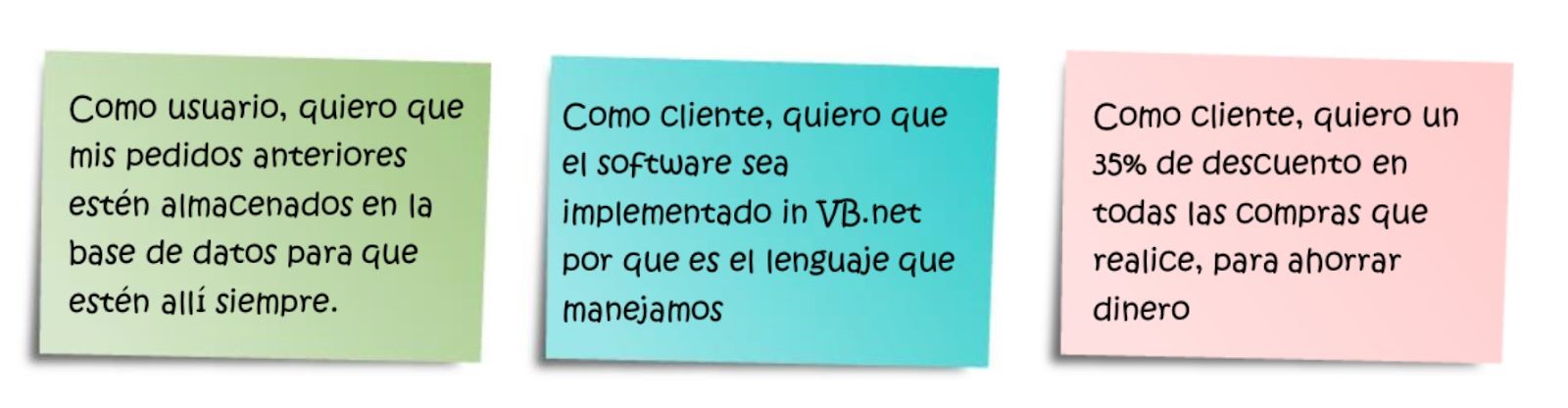
### Negotiable (Negociables)

La historia de usuario debe transmitir el problema, no la solución. De esta forma aseguramos que la solución se vuelva negociable.



### Valuable (Valiosa)

Cada historia de usuario debe proporcionar algún valor para un usuario y para el negocio.



### Estimable (Estimables)

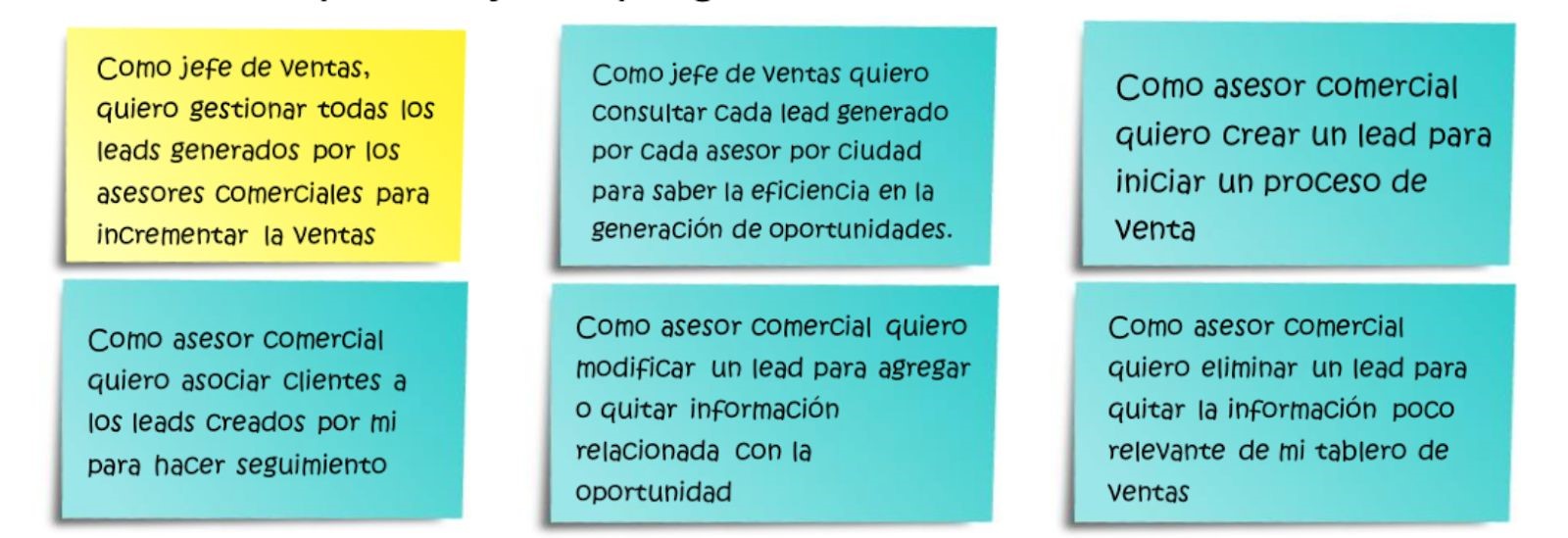
Cada historia de usuario, debe estar descrita de tal manera que pueda ser estimada por el equipo de desarrollo.

Si no puede ser estimada, es demasiado grande, demasiado vaga, arriesgada o alguna combinación de estas.

Las soluciones incluyen agregar criterios de aceptación, dividir la historia o definirla mejor.

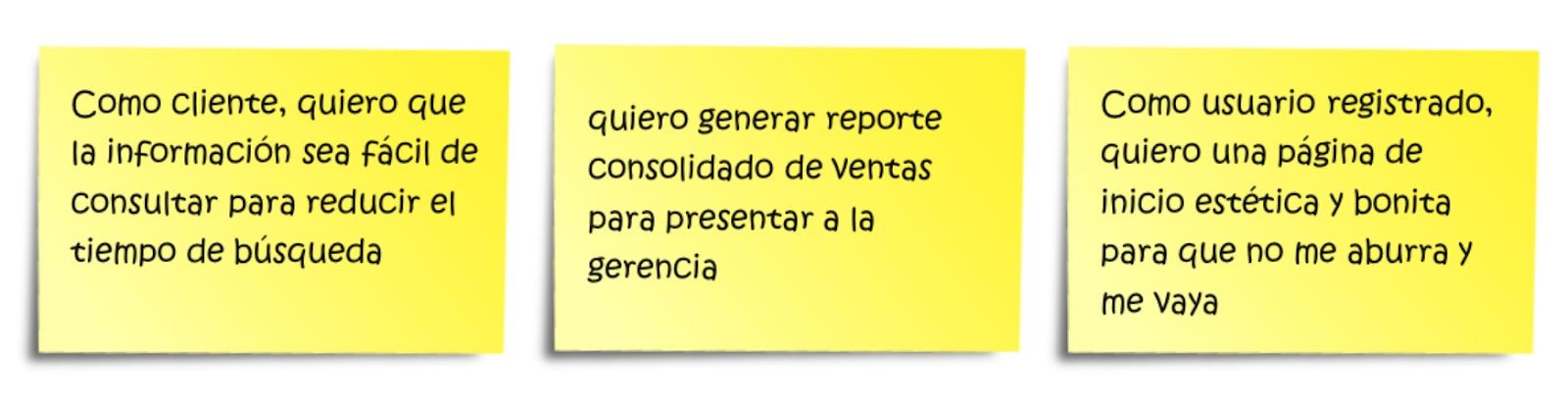
### Small (Pequeñas)

Cada una de las historias de usuario debe ser lo más pequeña posible. De esta manera, se puede implementar más rápido y los componentes de software son más fáciles de probar y desplegar.



### Testable (Verificable)

Cada historia de usuario debe estar descrita de forma medible o finita y su comportamiento se puede comprobar por alguna persona o máquina.

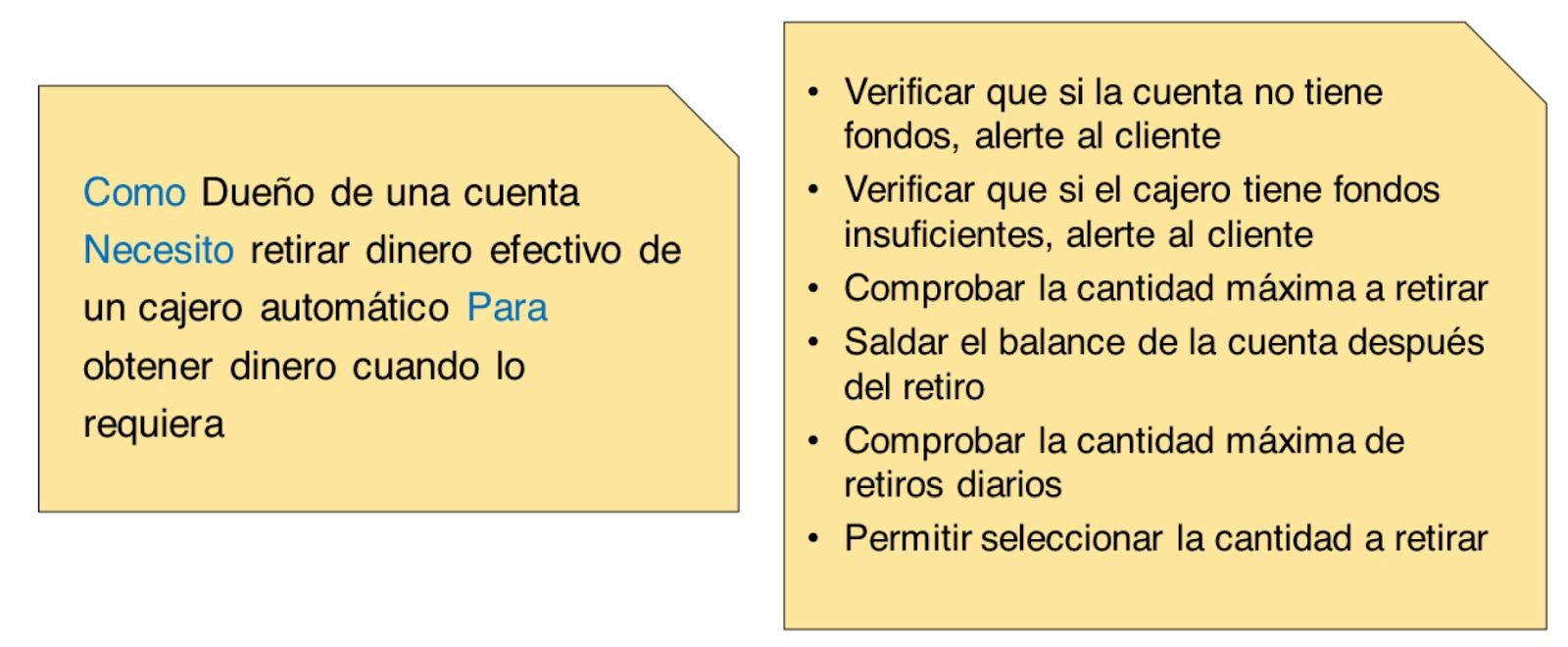


## Criterios de aceptación

Los criterios de aceptación nos permiten definir límites de las historias de usuario, además de ayudar a responder lo que se necesita para que la Historia de Usuario proporcione valor.

En pocas palabras, los criterios de aceptación nos van a ayudar a:

* Obtener un entendimiento común de la historia de usuario.
* Otorgar a los desarrolladores y probadores (testers) a obtener pruebas.
* Evidenciar a los desarrolladores cuando dejar de agregar más funcionalidades a un producto.



## Documentar prueba de aceptación

* **Condiciones iniciales** 
  1. **Dado** ​ **[**​ **Contexto ]**

**○ Y [ Algo más de contexto, si es el caso ]**​  **● Eventos desencadenantes**

**○ Cuándo [**​ **Evento ]**

* **Resultados y postcondiciones** 
  1. **Entonces [**​ **Resultado ]**

**○ Y [**​ **otro resultado, si es el caso ]**

## Ejemplo de criterios de aceptación

**Como** Dueño de una cuenta **Necesito**​ retirar dinero efectivo de un cajero automático

**Para** obtener dinero cuando quiera.​

### ● Escenario 1: La cuenta tiene fondos suficientes

○ **Dado** que el balance de la cuenta es de un millón de pesos​

○ **Y** mi tarjeta es válida​

○ **Y** el cajero contiene suficiente dinero​

○ **Cuando** el dueño de la cuenta solicite 20 mil pesos​

○ **Entonces** el cajero debe entregar 20 mil pesos​

○ **Y** el balance de la cuenta debe quedar en 980 mil pesos​

○ **Y** la tarjeta debe ser de vuelta​

### ● Escenario 2: La cuenta tiene fondos insuficientes

○ **Dado** que el balance de la cuenta es de 10 mil pesos​

○ **Y** mi tarjeta es válida​

○ **Y** el cajero contiene suficiente dinero​

○ **Cuando** el dueño de la cuenta solicite 20 mil pesos​

○ **Entonces** el cajero no debe entregar dinero​

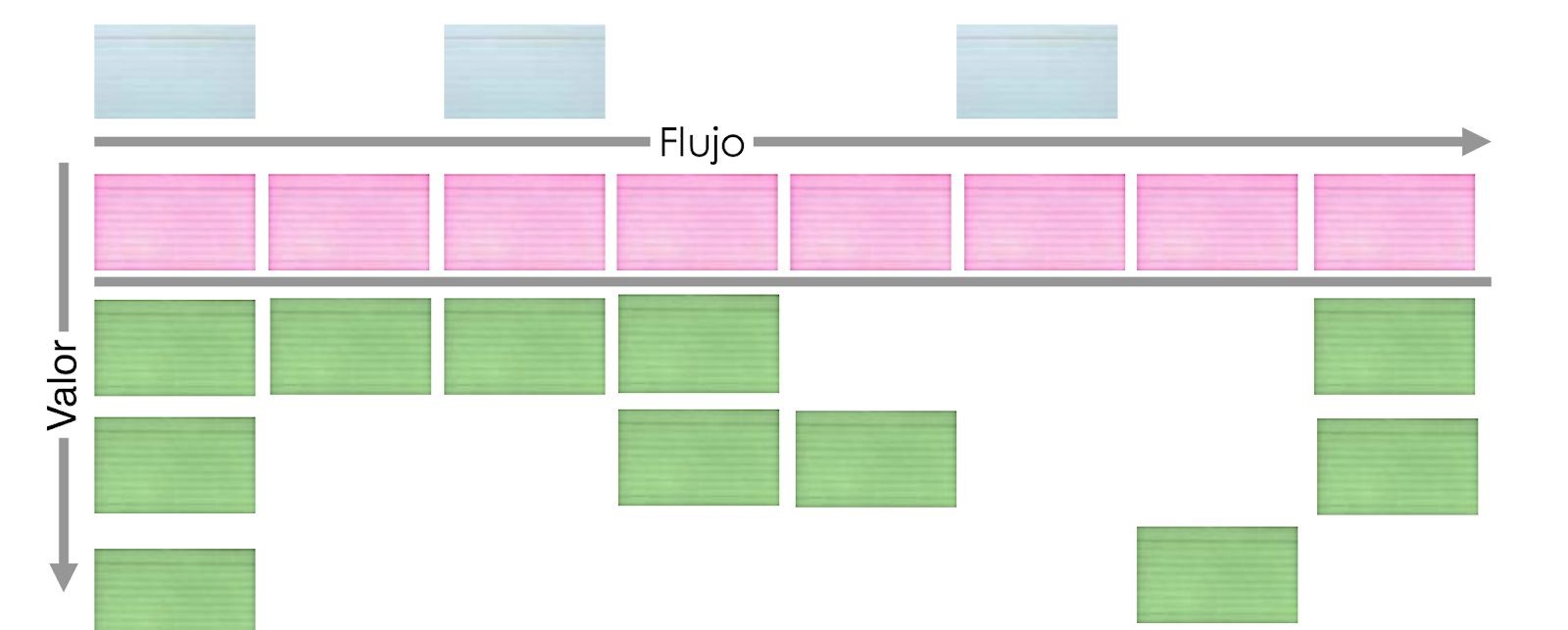
○ **Y** el cajero debe decir que tiene fondos insuficientes​

○ **Y** la tarjeta debe ser de vuelta​

## Mapa de historias de usuario

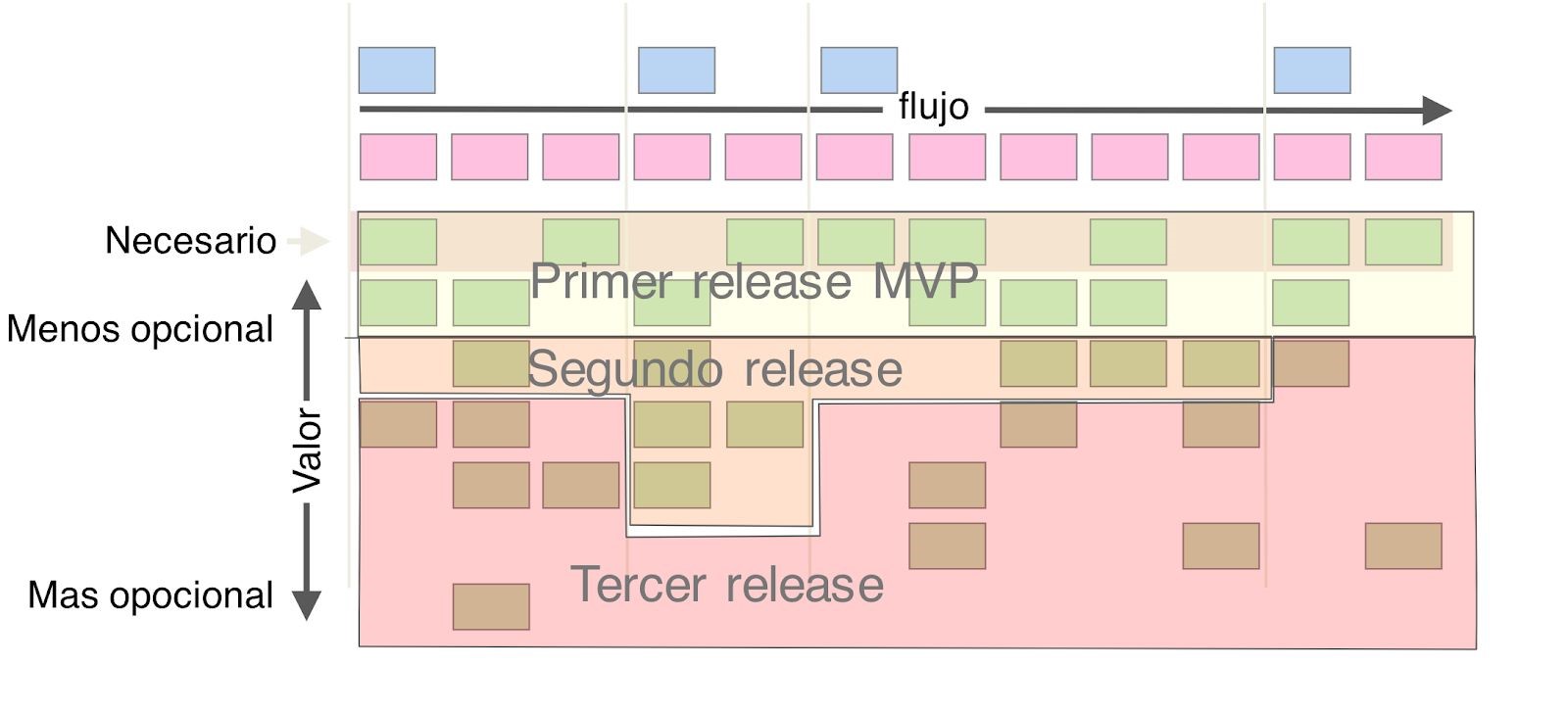
Los mapas de historias de usuario, ayudan a priorizar las historias por relevancia o valor del negocio. De esta forma podemos mover las tareas hacia arriba y hacia abajo para indicar que tan necesarias son de acuerdo al valor del negocio de cada una de las historias de usuario.





## Ejemplo de Mapa de historias de usuario





Los mapas de historias de usuario son útiles ya que:

* Permiten tener un panorama visual
* Se crean desde el punto de vista del usuario / persona
* Permite centrarnos en lo verdaderamente importante
* Permite dar una estimación de esfuerzo y tamaño
* Facilita la planeación del proyecto

## Taller

Cada uno de los siguientes puntos están orientados a motivar a los estudiantes a evidenciar el trabajo en equipo, por lo cual se debe hacer de acuerdo al proyecto pedagógico integrador.

1. Especificar 5 historias de usuario.
2. Describir cada uno de los criterios de aceptación para las historias de usuario especificadas en el punto 1.
3. Documentar las pruebas de aceptación para 5 criterios de aceptación.
4. Realizar el mapa de historias de usuario.

**Solución**

1. **Especificar 5 historias de usuario.**

**Esta en el archivo de Excel**

1. **Documentar las pruebas de aceptación para 5 criterios de aceptación**

Como usuario registrado de una aplicación de transporte. Necesito pedir un transporte que me lleve a mi destino. Para poder llegar al lugar que deseo ir

Como dueño de una cuenta bancaria. Necesito retirar dinero. Para poder pagar la comida. Dado que no tengo dinero en efectivo para pagar la comida

Como usuario registrado. Deseo ver los automóviles cercanos a mí posición . Para poder transportarme de un lugar a otro mediante la aplicación móvil

Como vendedor comercial. Necesito saber la cantidad de recursos con los que cuenta la empresa. Para poder vender los recursos implementados por la empresa

Como dueña de un restaurante. Necesito saber las opiniones de las personas mediante una encuesta. Para poder mejorar lo que haga falta

1. **Realizar el mapa de historias de usuario.**

