



Facultad de Ingeniería  
Escuela de Ingeniería de Sistemas y Computación

Mauricio Gaona  
mauricio.gaona@correounivalle.edu.co

Profesor

2025-I

## Agenda



01

### RESUMEN

Aspectos relevantes vistos en la clase anterior.

02

### METODOLOGÍAS ÁGILES

Requerimientos y proceso de las metodologías ágiles.

03

### ACTIVIDADES PROXIMA CLASE



## Conceptos





## Resumen

### **¿Qué es software ?**

El software es un conjunto estructurado de instrucciones codificadas en un lenguaje de programación, que, al ser ejecutadas por un equipo informático (hardware), le permiten realizar tareas específicas y procesar información.

### **Que es Ingeniería de Software?**

La ingeniería de software es una disciplina que aplica principios, metodologías y prácticas de la ingeniería al diseño, desarrollo, implementación, pruebas y mantenimiento de software. Se enfoca en la creación sistemática y eficiente de software de alta calidad que satisfaga las necesidades y expectativas de los usuarios.





## Conceptos

## Resumen

### El Ciclo de vida del desarrollo de software

Representa las etapas que describen el proceso de desarrollo de software desde su concepción hasta su puesta en producción.

**Análisis:** Es el proceso de entender el problema o necesidad de los usuarios y a partir de estas escribir los requerimientos de un sistema

**Diseño:** Es una descripción o especificación de procesos, componentes e interfaces que representan una solución para el sistema a construir.



## Conceptos

## Resumen

### **El Ciclo de vida del desarrollo de software**

Representa las etapas que describen el proceso de desarrollo de software desde su concepción hasta su puesta en producción.

La codificación es el proceso de transformar el diseño de un sistema en un formato de lenguaje de programación

Las pruebas de software es el proceso de evaluar y verificar que un sistema o aplicación de software hace lo que se supone que debe hacer

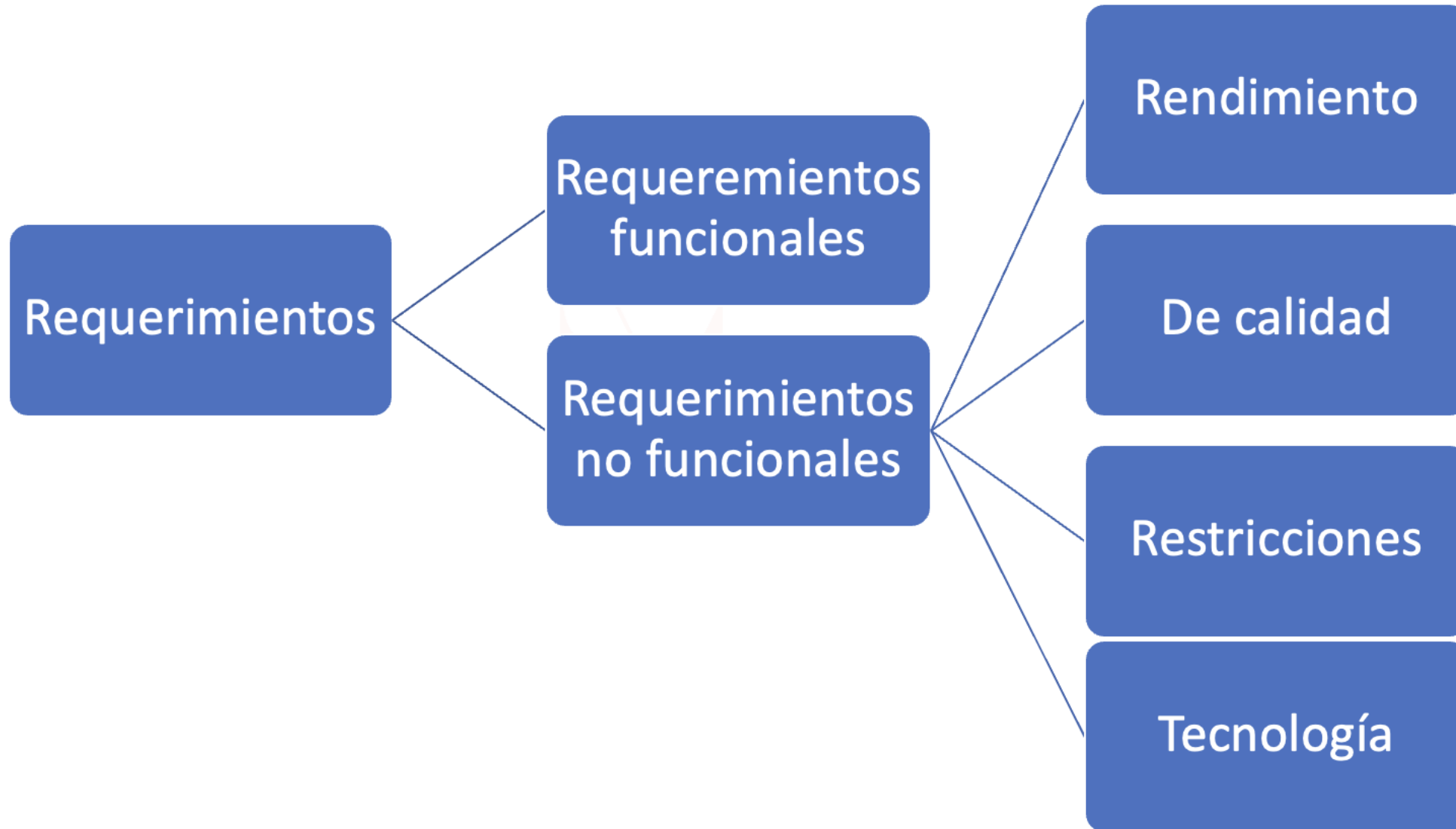
## REQUERIMIENTOS DEL SOFTWARE

Un requisito o requerimiento es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso, que es no ambigua, comprobable o medible, y necesaria para la aceptación del producto o proceso.

del Valle

Los requerimientos especifican lo que desea el cliente y definen las acciones o actividades que debe hacer o cumplir un sistema de software.







### Descripción de requerimientos

El sistema debe permitir crear un usuario del sistema.

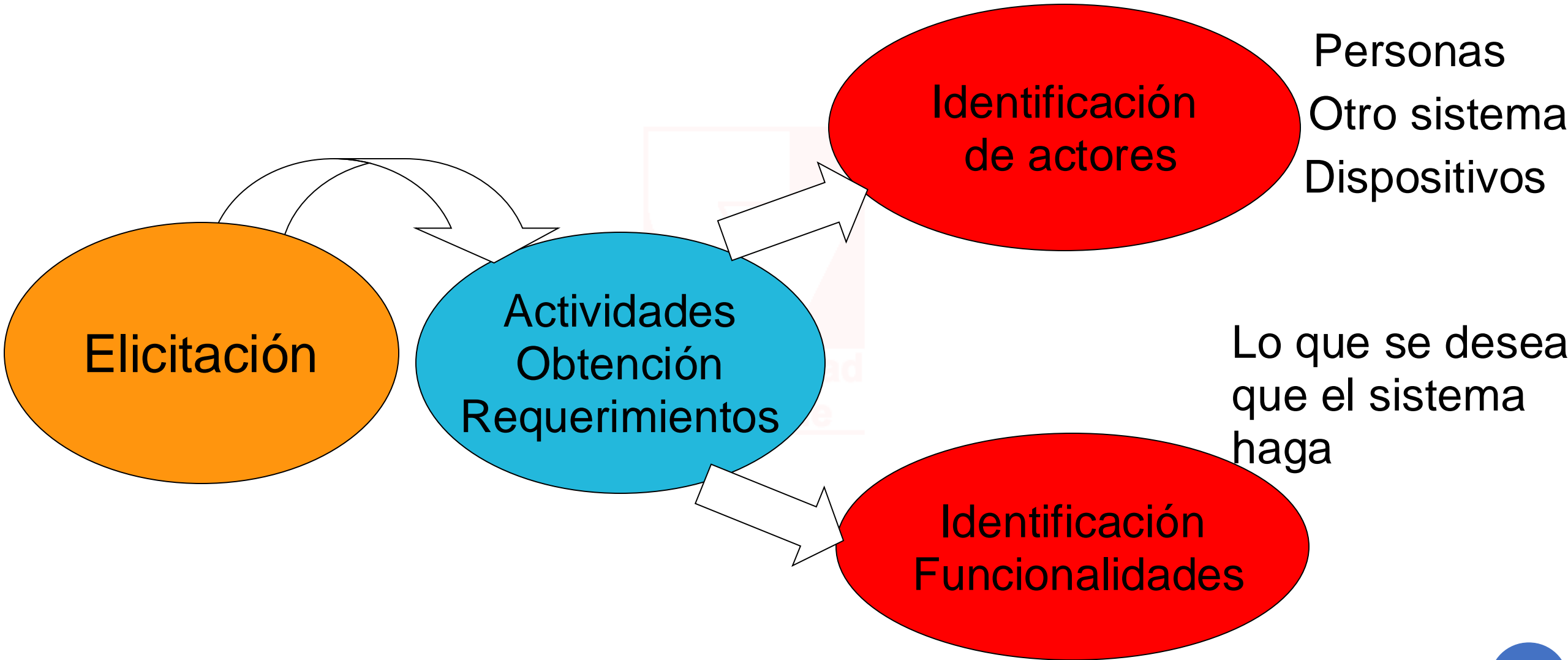
El sistema debe permitir registrar participantes a un evento.

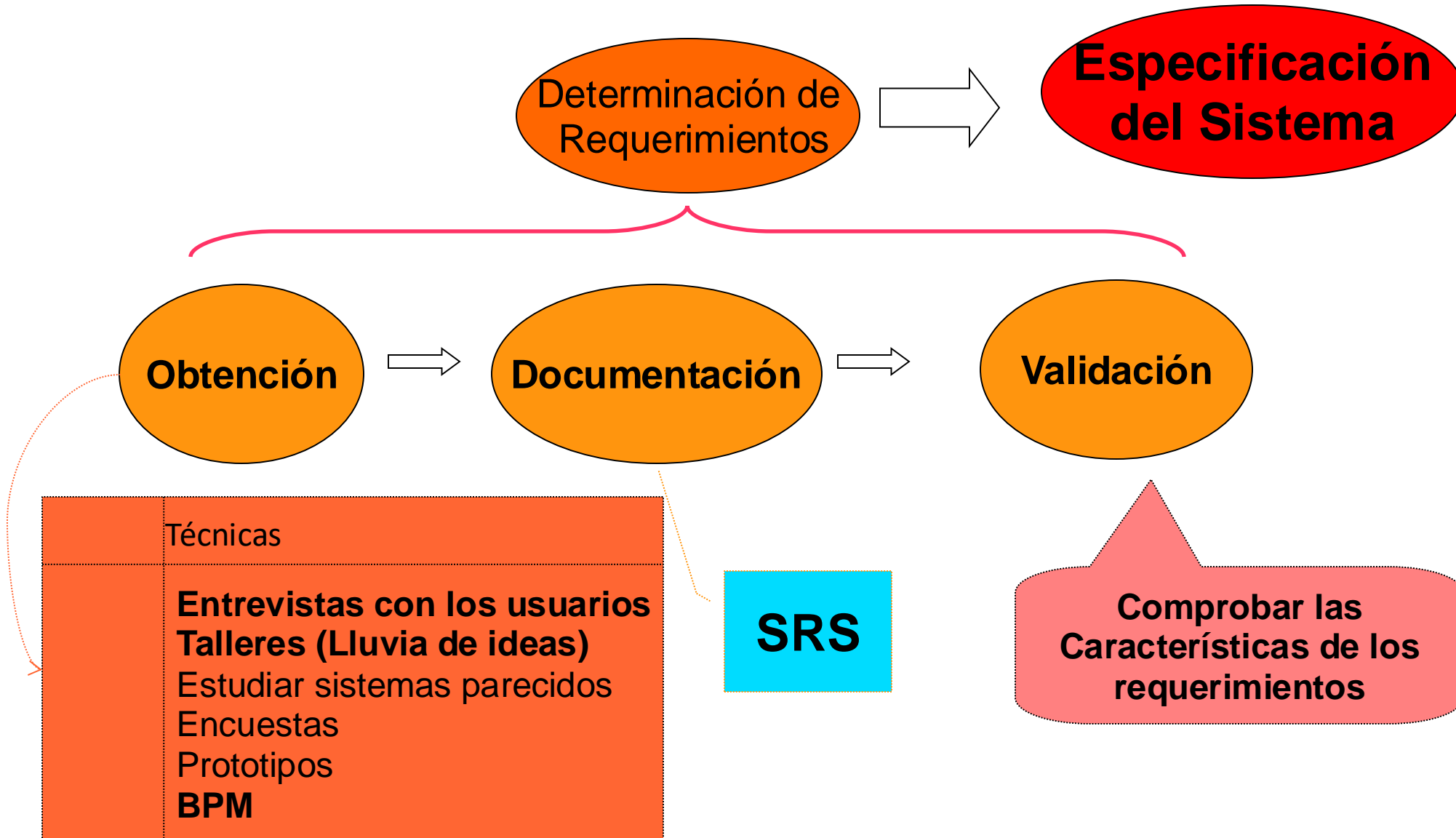
El sistema debe permitir registrar la venta de un artículo.

El sistema debe permitir anular una nota débito.

El sistema debe permitir a los usuarios contratistas poder registrar sus pagos de certificación al sistema de seguridad social junto al valor pagado por cada concepto como son: ARL, Pensión y EPS junto con el número de la planilla que lo soporta.









## Técnicas para validación de requerimientos

### 1. Validación de expertos

Personas con experiencia revisan los requerimientos y aprueban o rechazan el requerimiento

### 2. Prototipado de interfaz de usuario

El prototipado de interfaz de usuario es una técnica de representación aproximada de la interfaz de usuario

Los dos tipos principales de prototipos de interfaz de usuario son:

- **Desechables:** se utilizan sólo para la validación de los requisitos y posteriormente se desechan. Pueden ser prototipos en papel o en software.
- **Evolutivos:** una vez utilizados para la validación de los requisitos, se mejora su calidad y se convierten progresivamente en el producto final.

### 3. Recorrido de BPM: Hacer un BPM que muestre todo el proceso, algoritmo de alto nivel





## Preguntas ?



### Técnicas para obtener requerimientos

Tarea: control de lectura.

1. Ver video: <https://youtu.be/2KZKMY75cJM>
2. Leer documento: **Requirements Gathering Techniques**  
<https://www.linkedin.com/pulse/requirements-gathering-techniques-samgra-malik/>



## Metodologías ágiles



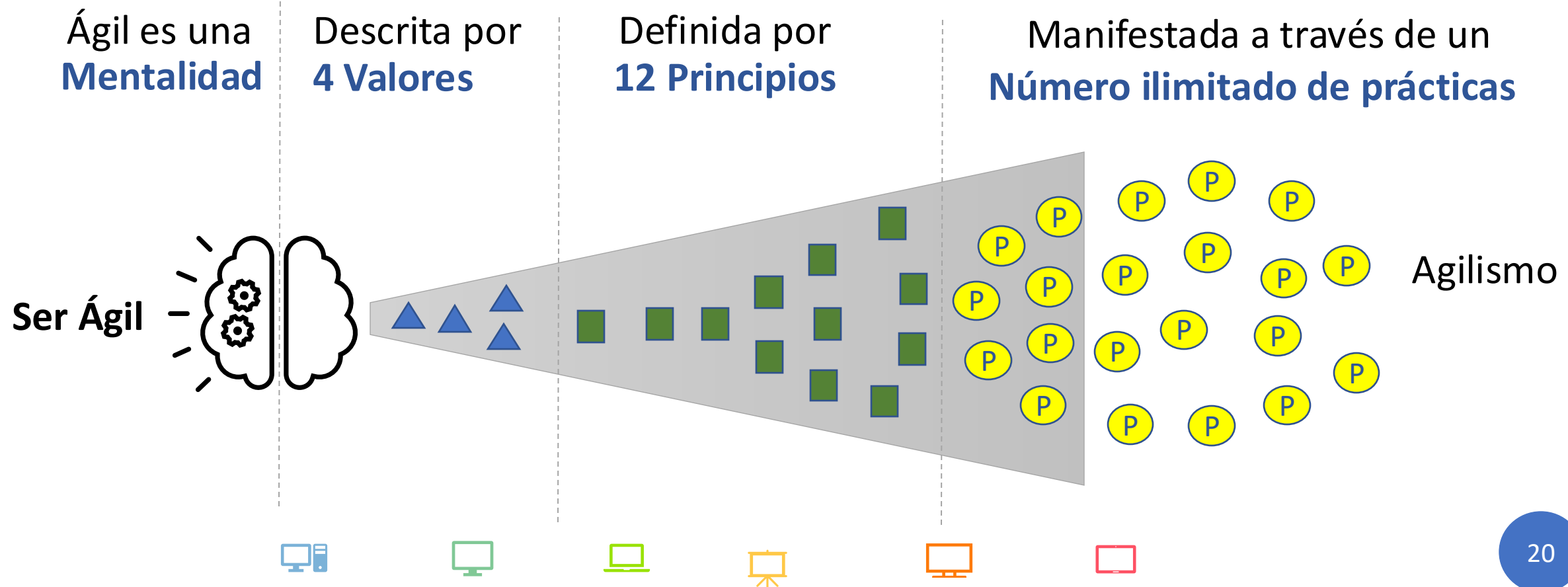
Ágil es un término que se usa para describir los enfoques iterativos del desarrollo de software que enfatizan en la entrega incremental, la colaboración en equipo, la planificación continua y el aprendizaje continuo, en lugar de intentar entregarlo todo de una vez cerca del final.





## Desarrollo de software ágil

El desarrollo de software ágil representa una forma de pensar y hacer las cosas.



## Desarrollo de software ágil

- El Desarrollo ágil de software es un paradigma usado en las metodologías de desarrollo de software basado en procesos ágiles.
- Se concibieron como una alternativa a las prácticas de desarrollo de software tradicional.
- Es una “sombra” para un conjunto de valores, principios y prácticas.
- El desarrollo ágil es una forma diferente de gestionar equipos y proyectos de desarrollo de Software.
- Una metodología ágil no es hacer lo mismo más rápido y tampoco es que nunca más vamos a planear y hacer contratos. (scrumcolombia)



## Manifiesto ágil

Manifiesto ágil  
"Biblia"





### Manifiesto Agil (*Agile Manifesto*)

---

(a.k.a. Manifiesto por el Desarrollo Ágil de Software)

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

<b>Individuos e interacciones</b>	sobre	procesos y herramientas
<b>Software funcionando</b>	sobre	documentación extensiva
<b>Colaboración con el cliente</b>	sobre	negociación contractual
<b>Respuesta ante el cambio</b>	sobre	seguir un plan

**Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.**



## Metodologías ágiles: 12 Principios definidos en el manifiesto ágil

- 01 Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- 02 Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo.
- 03 Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia por periodos de tiempo lo más corto posibles.
- 04 Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- 05 Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- 06 El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara (Presencial o virtual).



## Metodologías ágiles: 12 Principios definidos en el manifiesto ágil

07

El software funcionando es la medida principal de progreso.

08

Los procesos ágiles promueven el desarrollo sostenible.

09

La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

10

La simplicidad, o el arte de maximizar la cantidad de trabajo realizado, es esencial.

11

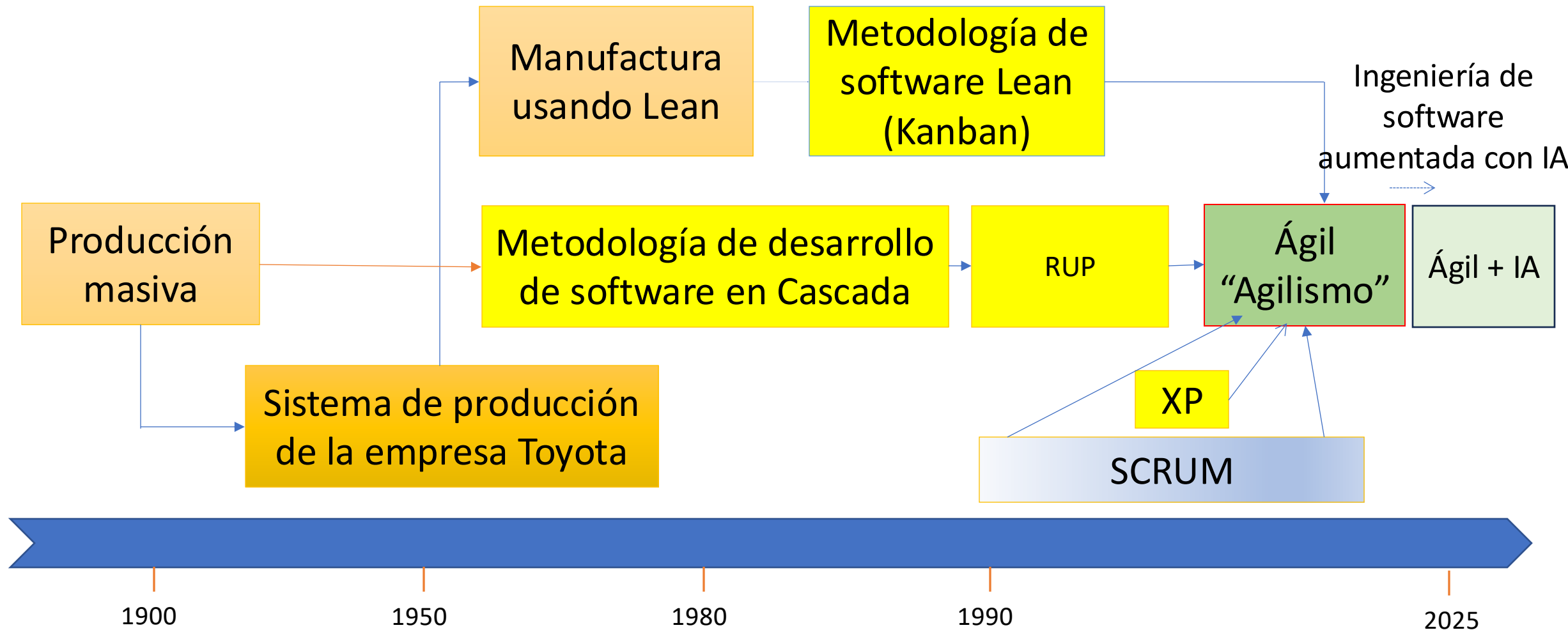
Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

12

A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.



## Historia de las metodologías ágiles









## Scrum

Metodología ágil de desarrollo de software.

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto de desarrollo de software.

Las fases en las que se divide y define un proceso de SCRUM son las siguientes:

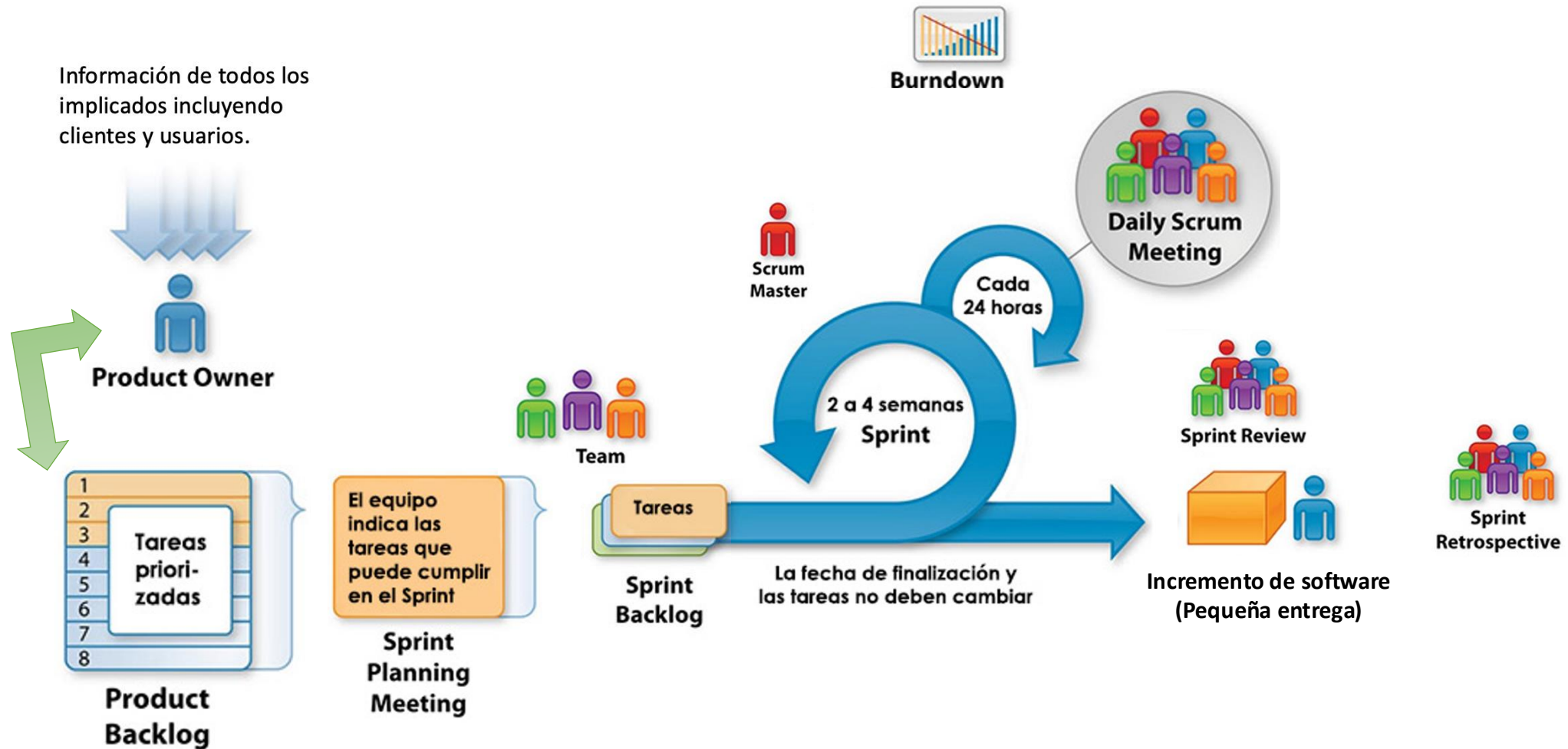
El ¿**Quién?** y el ¿**Qué?**: Identifica los roles de cada uno de los miembros del equipo y define su responsabilidad en el proyecto.

El ¿**Dónde?** y el ¿**Cuándo?**: que representan el Sprint (iteración)

El ¿**Por qué?** y el ¿**Cómo?**: representan las herramientas que utilizan los miembros de Scrum

## Conceptos

### Metodología ágil Scrum



### Historias de usuario (HU)

Las Historias de Usuario describen las necesidades de algo que es valioso para un usuario de un sistema o software.

Una historia debe permitir conocer la base de las necesidades del usuario, así como los detalles que después permitan fijar la batería de pruebas para poder determinar si un desarrollo es correcto o no.

Las historias de usuario luego se pueden dividir en tareas técnicas para su implementación.

Las historias de usuario son la base para aplicar las metodologías de desarrollo.





## Requerimientos en las metodologías ágiles

### Características deseables de las historias de usuario (HU)

#### El modelo INVEST

Una buena historia de usuario también sigue el modelo de INVEST: Independiente, Negociable, Estimable, Pequeña (Small), y Testeable. Veamos lo que significa.

- **Independiente** - una historia debería ser independiente de otras. Facilitan la planificación, priorizar y estimación.
- **Negociable** - La "tarjeta" de la historia es tan sólo una descripción corta que no incluye detalles. Los detalles se añaden mediante la conversación.
- **Valiosa** - cada historia tiene que tener valor para el cliente (para el usuario o para el comprador).
- **Estimable** - el equipo necesitan poder estimar una historia de usuario. Historias demasiado grandes o inconcretas, no se pueden estimar.
- **Pequeña** - una buena historia debe ser pequeña en esfuerzo, debería ser realizable en menos de una semana.
- **Testeable** - una historia necesita poder probarse y saber que la HU se ha completado con éxito.

## Requerimientos en las metodologías ágiles

### Ejemplos de historias de usuario

Para documentar una Historia de Usuario se utiliza el formato popularizado por la empresa **Connextra**

Como **estudiante** deseo **registrarme en un curso** para **matricularme en la universidad**

Actor

La tarea

Propósito

Plantilla de **Connextra**: Como **<Actor>** deseo **<tarea>** para **<propósito>**

[Quiero]



## Requerimientos en las metodologías ágiles

### Ejemplos de historias de usuario

1. El sistema debe permitir crear un usuario del sistema. (tradicional)
  1. Como **administrador** del sistema, deseo **crear un usuario del sistema** para **gestionar los accesos y roles asignados**.
  2. Como **Vendedor**, deseo **registrar los productos y cantidades que me solicita un cliente** para **crear un pedido** de venta.
  3. Como **Auxiliar**, deseo poder registrar los participantes a un evento para conocer cuantas personas están inscritas al evento.



## Ejercicio: Elaborar la descripción de una historia de usuario

Plantilla: Como <Actor> deseo <tarea> para <propósito>

Elaborar la descripción de una historia de usuario para crear una factura de venta de un producto a un cliente

Elaborar la descripción de una historia para el proceso de retiro en efectivo en un cajero automático







## Características de la historia de usuario

Ser lo suficientemente completa como para demostrar el valor para el usuario.

Centrarse en el usuario.

Si no se identifica necesidades precisas, se puede iniciar con una historia de usuario **Épica**.

Ser breve, sencilla y clara.

**Spyke** (investigar de cosas que el equipo de desarrollo no conoce)

[Incluir archivos y documentación de apoyo si es necesario].

Ser lo suficientemente completa como para demostrar su valor, pero lo suficientemente sencilla como para desarrollarlo en una sola iteración.

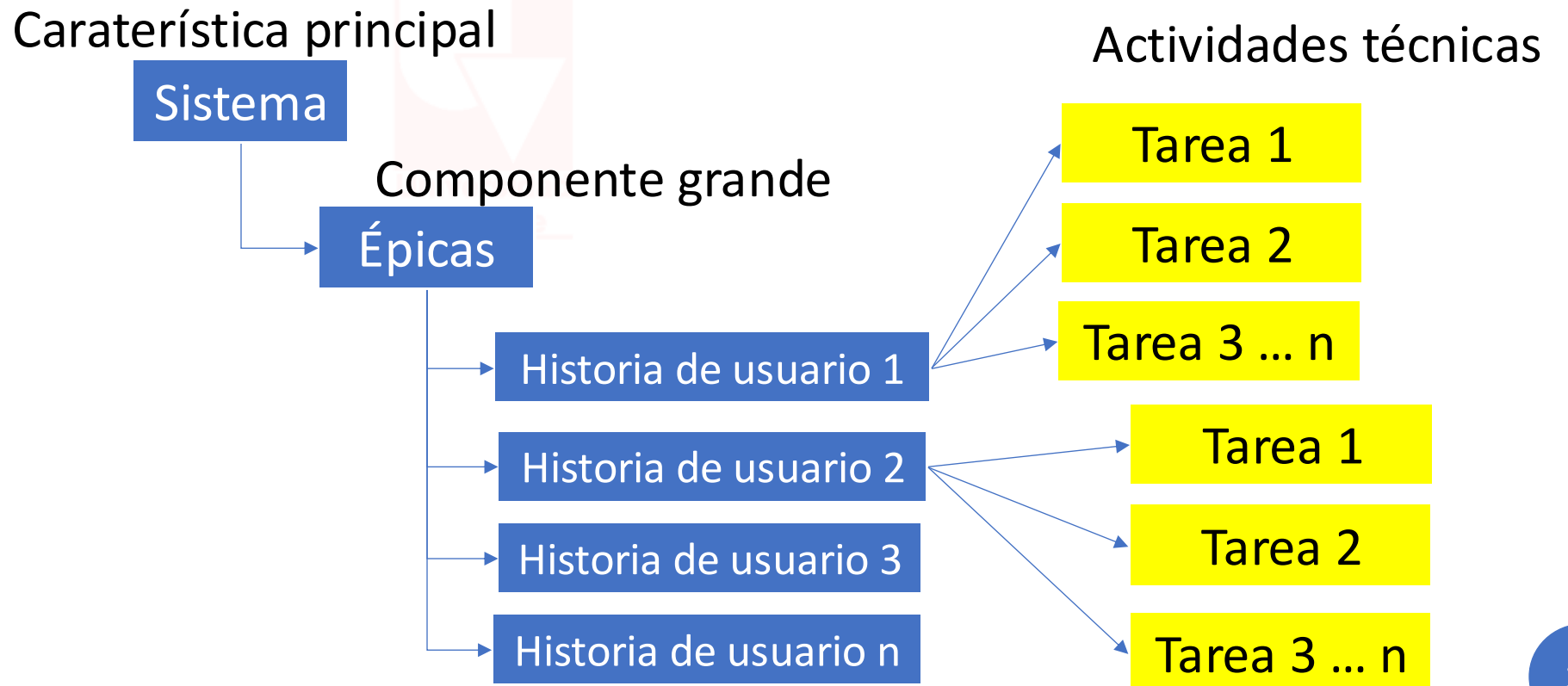
Se recomienda **dividir una historia de usuario en tareas** sencillas



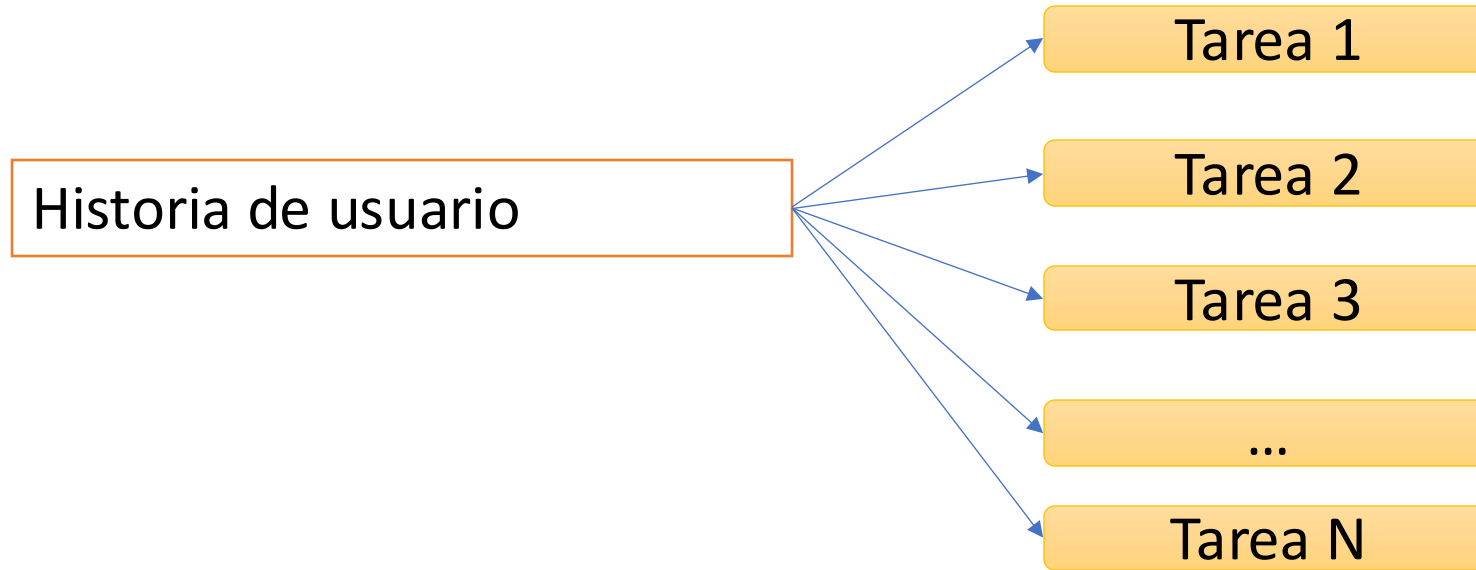


## Historia de usuario épicas

“Una historia épica es una gran historia de usuario que no se puede entregar como se define en una sola iteración o es lo suficientemente grande como para dividirse en historias de usuario más pequeñas.”



## Descomponer una Historia de usuario en tareas



¿Para qué dividir una historia de usuario en tareas ?

- Para expresar los pasos para implementar una historia de usuario.
- Para precisar la actividad en la que está trabajando.
- Puede ayudar para pedir colaboración más específica.
- Puede conducir a estimaciones más precisas.
- Se pueden usar como una medida de progreso.

### ¿Cómo descomponer historia de usuario en tareas?

No hay una fórmula para hacer esto.

Depende de la HU las tareas a realizar.

### Recomendaciones:

- Descomponga la HU en tareas técnicas.
- No programe tareas demasiado pequeñas.
- Trate de programar tareas con duración de horas y que puedan mostrar avances.
- **TIP:** Antes de implementar tareas debes de cero verifique si hay componentes listos que pueda usar.

Descomponer una Historia de usuario en tareas

Ejemplo: HU003. Como usuario operador deseo poder registrar en el sistema datos de un cliente para tener información del cliente.

¿Cómo podemos dividir esta HU en tareas?

## Dividir la HU en tareas técnicas

Tarea	Descripción de la tarea
1	Diseñar e Implementar formulario para capturar los datos del cliente.
2	Adicionar tablas y/o campos en la BDatos para registrar los datos del formulario.
3	Implementar lógica para leer y validar los campos del formulario.
4	Implementar mensaje fallo del proceso de lectura o validación si ocurre
5	Implementar el registro de datos del formulario en la BDatos.
6	Implementar mensaje de éxito o fallo del proceso en la bases de datos (si lo hay).
7	Ejecutar plan de pruebas (Realizar las pruebas de aceptación definidas).
8	Realizar actividades del Definition of Done ( <b>DoD</b> ).

## Historia de usuario épicas

HU: Como usuario administrador del sistema deseo poder gestionar usuarios para poder administrar los usuarios

Esta épica se puede subdividir en:

- Como usuario administrador deseo poder **crear** un usuario para registrarlo en el sistema.
- Como usuario administrador deseo poder **modificar** los datos de un usuario para actualizar información de un usuario.
- Como usuario administrador deseo poder **listar** los usuarios registrados en el sistema para conocer el tipo de usuarios se tienen el sistema.
- Como usuario administrador deseo inhabilitar (**borrar**) un usuario para desactivar un usuario del sistema.





## Historia de usuario épicas

Como Vicepresidente de mercadeo y ventas, quiero revisar el desempeño histórico de las ventas, para poder identificar las regiones geográficas y productos de mejor desempeño

Esta épica se puede subdividir en:

- Como VP de Mercadeo, deseo seleccionar el período de tiempo en el cual realizaré la revisión de las ventas para conocer el comportamiento de las ventas de diferentes periodos.
- Como VP de Mercadeo, deseo clasificar la información de ventas por región geográfica para tener la información clasificada por regiones.
- Como VP de Mercadeo, deseo clasificar la información de ventas por productos para tener la información clasificada según el tipo de producto.

Ejemplos de HU:

<http://www.pmoinformatica.com/2015/05/historias-de-usuario-ejemplos.html>



## Detalles de una historia de usuario

1

Historia de Usuario (HU)			
Código HU:	HU0023	Fecha:	21/02/2025
Sprint:	1	Prioridad:	Alta
Actor(es):	Gerente	Puntos:	3



2

**Descripción:** Como **usuario** deseo poder **loguearme en el sistema** para poder **ingresar al sistema**

3

**Detalles de la HU:**[Colocar aquí toda la información que se requiera para entender la HU: **Tareas**, Texto (entradas, proceso, salida), fotos videos, audio, BPM, diseño de interfaces de usuario, etc]



## Detalles de una historia de usuario

4

### **Restricciones:**

1. Solo los usuarios con privilegios de gerente pueden acceder al reporte.
2. El tiempo de respuesta para loguearse no debe exceder 4 segundos

5

### **Criterios de aceptación: (Detalle de las HU desde el punto de vista de calidad y se traducen en pruebas)**

Criterio 1, Criterio 2, ...

6

### **DoD (Definition of Done):**

Lista de actividades que se deben cumplir para que la historia esté en condiciones de ser entregado al cliente.





## Detalles de una historia de usuario

1

Historia de Usuario (HU)			
Código HU:	HU0023	Fecha:	21/02/2025
Sprint:	1	Prioridad:	Alta
Actor(es):	Gerente	Puntos:	3

2

**Descripción:** Como **usuario** deseo poder **loguearme en el sistema** para poder **ingresar al sistema**

3

**Detalles de la HU:**[Colocar aquí toda la información que se requiera para entender la HU: **Tareas**, Texto (entradas, proceso, salida), fotos videos, audio, BPM, diseño de interfaces de usuario, etc]

4

### Restricciones:

1. Solo los usuarios con privilegios de gerente pueden acceder al reporte.
2. El tiempo de respuesta para loguearse no debe exceder 4 segundos

5

### Criterios de aceptación: (Detalle de las HU desde el punto de vista de calidad y se traducen en pruebas)

Criterio 1, Criterio 2, ...

6

### DoD (Definition of Done):

Lista de actividades que se deben cumplir para que la historia esté en condiciones de ser entregado al cliente.



## Detalles de una historia de usuario : Criterios de aceptación

Como usuario deseo poder loguearme en el sistema para poder ingresar al sistema

Item	Criterio de aceptación : las condiciones que un producto de software debe satisfacer para ser <b>aceptado</b> por un usuario o cliente.
------	---

- |   |   |
|---|---|
| 1 | Un usuario no puede enviar un formulario sin completar todos los campos.                      |
| 2 | El login no puede ser igual al password.  |
| 3 | El password debe tener al menos 6 caracteres, una letra en mayúscula, un símbolo y un número. |
| 4 | Los campos de entrada de datos se deben validar antes de ejecutarlos o guardarlos.            |

Prueba	Login	Password	Resultado esperado
1			Login o password vacíos
2	123456789	123456789	Login o password incorrecto
3	usuario	Secret()22	Login exitoso
4	prueba	SELECT * FROM Users WHERE Username='1' OR '1' = '1' AND Password='1' OR '1' = '1'	Login o password incorrecto

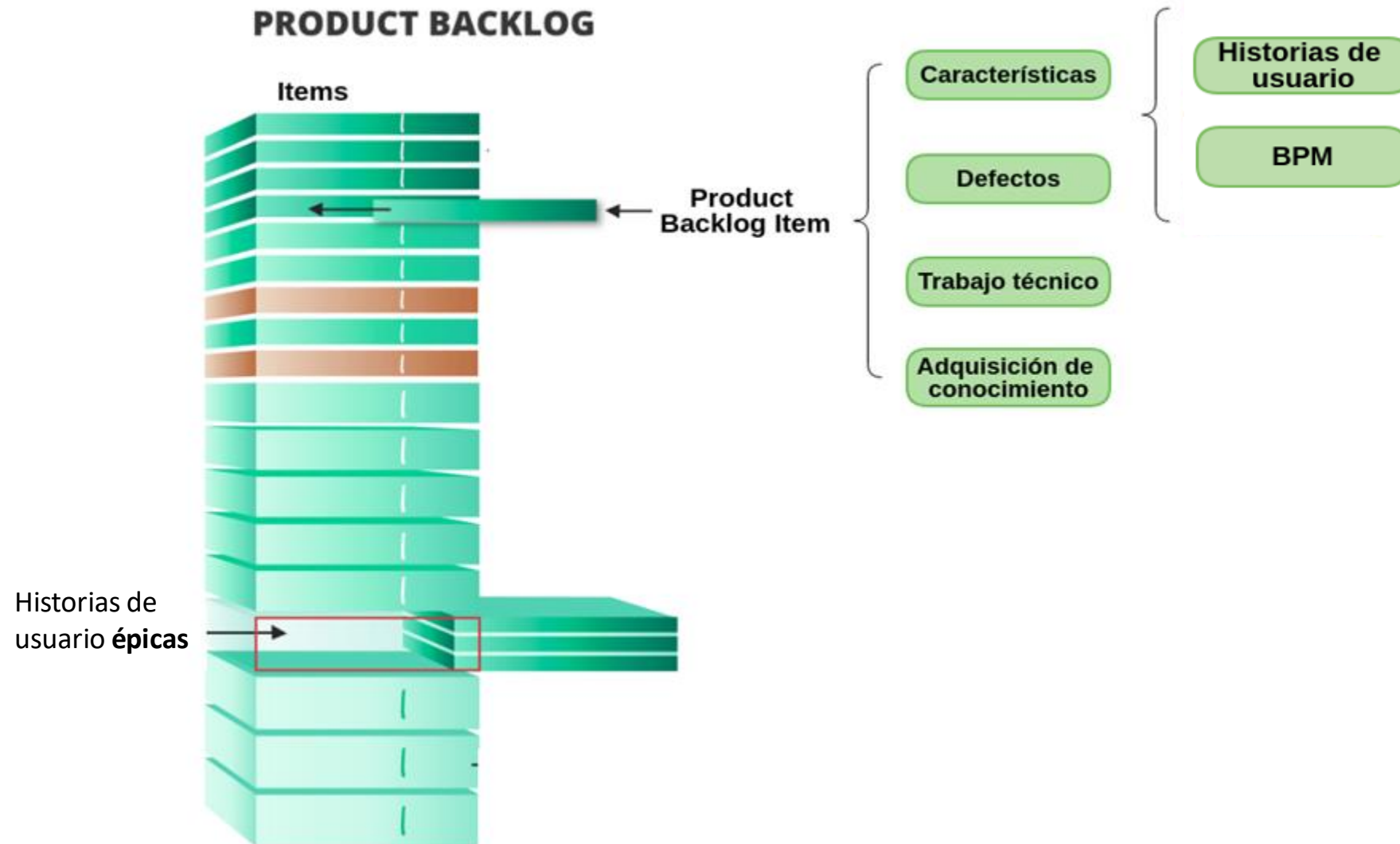
Pruebas  
de  
Aceptación



## PRÁCTICAS ÁGILES EN SCRUM

**Product Backlog** : Una lista **priorizada** y **estimada** de las historias de usuario, que representan las funcionalidades del sistema que se va a construir; en algunas ocasiones se incluyen otros aspectos requeridos durante las etapas de desarrollo del sistema.

## PRÁCTICAS ÁGILES EN SCRUM



## PRÁCTICAS ÁGILES EN SCRUM

### Priorización de las Historias de Usuario

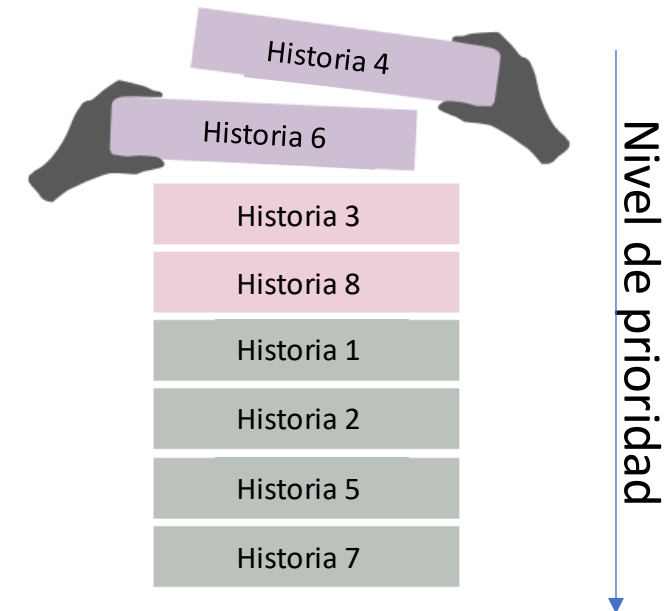
¿Qué es priorizar?

La priorización como principio significa "hacer lo primero". Como proceso, significa "evaluar un grupo de elementos y clasificarlos en orden de importancia o necesidad".

La priorización de requerimientos en todas las metodologías de desarrollo de software se considera una parte vital del proyecto, pero es especialmente importante en el desarrollo de software Agile.



Product Backlog

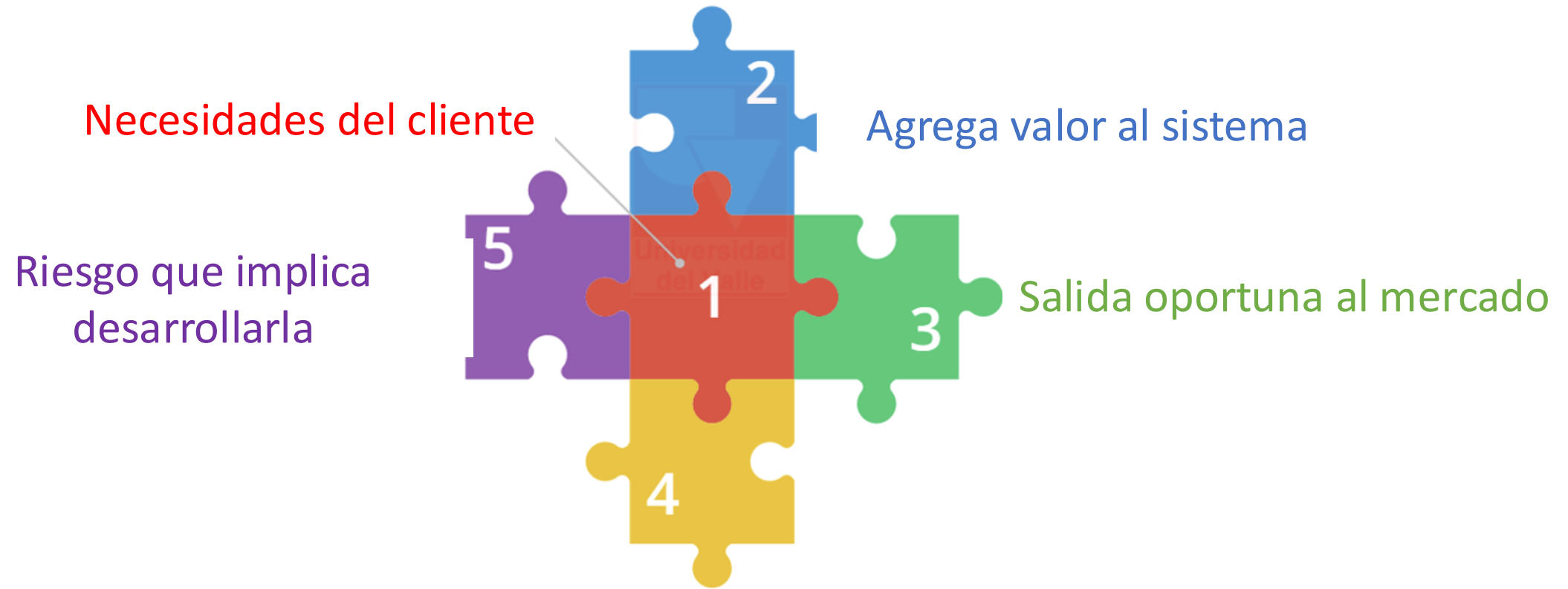


Product Backlog priorizado

## PRÁCTICAS ÁGILES EN SCRUM

### Priorización de HU

#### Criterios para priorizar



Son requeridas por otras historias



## PRÁCTICAS ÁGILES EN SCRUM

### Priorización de HU Técnica de clasificación de lista

Esta técnica es un ordenamiento simple y que no requieren ningún tipo de entrenamiento ni preparación.

En la gestión de Proyectos, se trata de indicar a cada HU un orden de prioridad, empezando por el 1, luego el 2, 3, y continuando hasta n, que es la cantidad total de HU.

Ventajas	Desventajas
<ul style="list-style-type: none"><li>• Solo puede haber un número uno</li><li>• Aporta precisión y evita la confusión</li></ul>	<ul style="list-style-type: none"><li>• Requiere un conocimiento profundo de todas las HU definidas y un esfuerzo grande por parte del equipo, para situar cada una de las HU en la posición correcta.</li></ul>





## PRÁCTICAS ÁGILES EN SCRUM

### Priorización de HU

### Técnica de clasificación por rangos

Para asignar una historia de usuario a un rango se aplican los criterios de prioridad.

Listas de tres rangos: Alta, Media y Baja

Ejemplo: Usando los criterios definidos y basados en la experiencia del equipo de desarrollo

#	Historia de Usuario	Prioridad
1	HU1	ALTA
2	HU2	MEDIA
3	HU3	ALTA
4	HU4	ALTA
5	HU5	BAJA
7	HU6	MEDIA
8	HU7	BAJA
9	HU8	ALTA

#	Historia de Usuario	Prioridad
1	HU1	ALTA
2	HU3	ALTA
3	HU4	ALTA
4	HU8	ALTA
5	HU2	MEDIA
7	HU6	MEDIA
8	HU5	BAJA
9	HU7	BAJA



## PRÁCTICAS ÁGILES EN SCRUM

### Priorización de HU usando rangos

#### Priorización de Historias de Usuario usando el método de Matriz de Urgencia

La matriz de urgencia es un método que emplea dos criterios clave —**Urgencia** y **Valor de Negocio**— asignando a cada uno una puntuación del 1 al 5 para establecer la prioridad de las historias de usuario. Para priorizar las historias de usuario mediante la matriz de urgencia, se sigue los siguiente pasos:

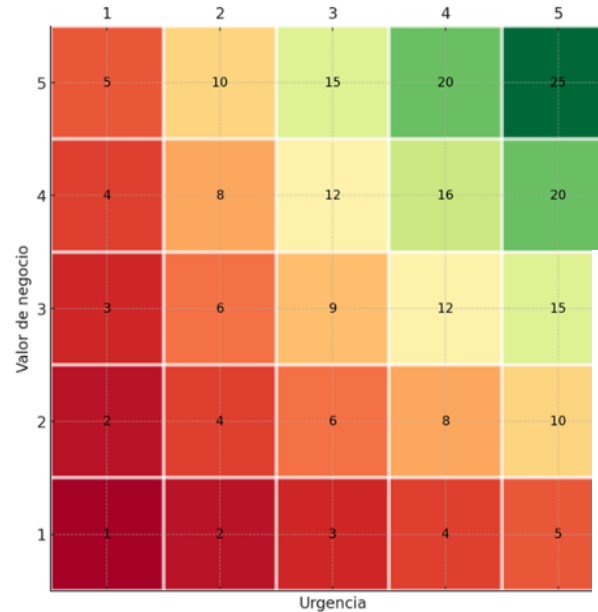
1. Asignar un valor de urgencia a cada historia de usuario, en una escala de 1 (menos urgente) a 5 (muy urgente). Este valor indica la necesidad de implementar la historia de usuario pronto para mantener su relevancia.
2. Asignar un valor de valor de negocio a cada historia de usuario, también en una escala de 1 (bajo beneficio) a 5 (alto beneficio). Este valor refleja la importancia o el beneficio que aportará al negocio o los usuarios al desarrollar esa funcionalidad.
3. Multiplicar el valor de urgencia por el valor de negocio para obtener un total que determinará la prioridad de la historia de usuario.
4. Ordenar las historias de usuario por su total, priorizando aquellas con puntuaciones más altas para su desarrollo e implementación.

## PRÁCTICAS ÁGILES EN SCRUM

### Priorización de HU usando rangos: Método de la Matriz de urgencia

Historias de usuario a priorizar

HU	Descripción
HU1	Como usuario deseo ingresar un comando de voz para poder realizar búsquedas
HU2	Como sistema deseo transformar un comando de voz a texto para que el sistema pueda procesarlo
HU3	Como sistema deseo asociar el texto a un comando del contexto para poder realizar acciones
HU4	Como usuario deseo recibir mensaje "comando no reconocido" para saber si el comando ingresado es válido
HU5	Como sistema deseo generar una respuesta de voz para enviarla al usuario
HU6	Como usuario deseo recibir un mensaje del proceso en ejecución para conocer el estado de la tarea
HU7	Como sistema deseo obtener posición actual del dispositivo GPS para enviarla al usuario
HU8	Como sistema deseo realizar búsqueda por contexto de ubicaciones cercanas a la ubicación actual según el GPS para asociar la búsqueda al comando y a la posición del GPS
HU9	Como usuario deseo recibir mensaje de ubicación no encontrada para informarle al usuario
HU10	Como sistema deseo generar un mapa con las ubicaciones cercanas para poder mostrarlas en el mapa al usuario los lugares buscados con la ruta óptima para llegar
HU11	Como usuario deseo poder ver mapa con las ubicaciones cercanas para saber cuál es el sitio más cercano que estoy buscando y la ruta óptima para llegar



Matriz de priorización de urgencia

HU	Urgencia (Puntos)	Valor de negocio (Puntos)	Prioridad (Puntos)
HU4	Alta (5)	Alta (5)	25
HU9	Alta (5)	Alta (5)	25
HU1	Alta (5)	Media (3)	15
HU6	Alta (5)	Media (3)	15
HU2	Media (3)	Alta (5)	15
HU7	Media (3)	Alta (5)	15
HU5	Media (3)	Media (3)	9
HU3	Media (3)	Media (3)	9
HU8	Media (3)	Baja (1)	4
HU10	Baja (1)	Media (3)	4
HU11	Baja (1)	Alta (5)	6

Resultado de la priorización de las historias de usuario usando la matriz de urgencia

## PRÁCTICAS ÁGILES EN SCRUM

### Priorización de HU usando rangos: Método MoSCoW

**MoSCoW** es una técnica que aporta un valor semántico de lo que realmente es importante.

En este método se plantea una categorización de las HU en función de palabras que tengan un significado concreto:

- ⬡ M: Esta funcionalidad debe estar (**MUST**).  
No negociable, la aplicación la requiere es vital.
- ⬡ S: Esta funcionalidad debería estar (**SHOULD**).  
Cosas importantes pero no vitales.
- ⬡ C: Esta funcionalidad podría estar (**COULD**).  
Cosas de bajo impacto en la aplicación
- ⬡ W: Esta funcionalidad no estará ahora, quizás en un futuro (**WON**).  
Cosas no prioritarias y aportan poco valor



### MosCoW





## PRÁCTICAS ÁGILES EN SCRUM

### Priorización de HU usando MosCow

Ejemplo: Funcionalidades requeridas para un sistema

- como usuario deseo registrarse en el sistema para poder usarlo
- Como usuario deseo iniciar sesión para poder ingresar al sistema
- Como usuario deseo restablecer la contraseña para tener control de acceso al sistema
- Como usuario deseo elegir la forma de pago para tener diferentes opciones de pago
- Como usuario deseo eliminar la cuenta para retirarme del sistema
- Como usuario deseo abrir una página de seguimiento de un producto para hacer seguimiento
- Como usuario deseo elegir opciones de seguimiento de un producto para saber donde esta en cada momento
- Como usuario deseo tener una versión de móvil de la aplicación para usarla desde el teléfono
- Como usuario deseo elegir el tema visual de la aplicación para tener diferentes formas de ver la aplicación.



## PRÁCTICAS ÁGILES EN SCRUM

### Priorización de HU usando MosCow

#	Historia de Usuario	Prioridad
1	Como usuario deseo registrarse en el sistema para poder usarlo	<b>MUST</b>
2	Como usuario deseo iniciar sesión para poder ingresar al sistema	<b>MUST</b>
3	Como usuario deseo restablecer la contraseña para tener control de acceso al sistema	<b>MUST</b>
4	Como usuario deseo abrir una página de seguimiento de un producto para hacer seguimiento	<b>MUST</b>
5	Como usuario deseo elegir la forma de pago para tener diferentes opciones de pago	<b>SHOULD</b>
6	Como usuario deseo eliminar la cuenta para retirarme del sistema	<b>SHOULD</b>
7	Como usuario deseo elegir opciones de seguimiento de un producto para saber dónde está en cada momento	<b>SHOULD</b>
8	Como usuario deseo tener una versión de móvil de la aplicación para usarla desde el teléfono	<b>COULD</b>
9	Como usuario deseo elegir el tema visual de la aplicación para tener diferentes formas de ver la aplicación.	<b>WON'T</b>

## PRÁCTICAS ÁGILES EN SCRUM

### Estimar Historias de Usuario

Determinar el esfuerzo, complejidad y riesgo que se requiere para desarrollar una HU.



**La estimación se mide en puntos** (Puntos de Historias de Usuario)

Un Punto de HU mide:

- a) La cantidad de esfuerzo que supone desarrollar la historia de usuario
- b) la complejidad de su desarrollo y
- c) el riesgo inherente.

[Nota: Un punto de HU = X horas

Ejemplo: 1 punto = 8 horas. (Esto depende de las características de cada equipo de desarrollo)]



## PRÁCTICAS ÁGILES EN SCRUM

### Estimar Historias de Usuario

#### ¿Qué es estimar?

Estimar es una forma de **cuntificar** la magnitud del proyecto y de las tareas que tenemos que asumir, a hacernos una idea aproximada del tiempo y recursos que vamos a consumir.

*Importante: las estimaciones son sólo valores aproximados, no son valores exactos.*

La estimación en la metodología ágil consiste en asignar puntos a una tarea o Historia de usuario. No hay una fórmula para realizarlo de una manera exacta.

La estimación ágil se basa en la estimación relativa, y una técnica colaborativa, sencilla, divertida y efectiva de poder estimar historias de usuario es la de **Planning Poker**.



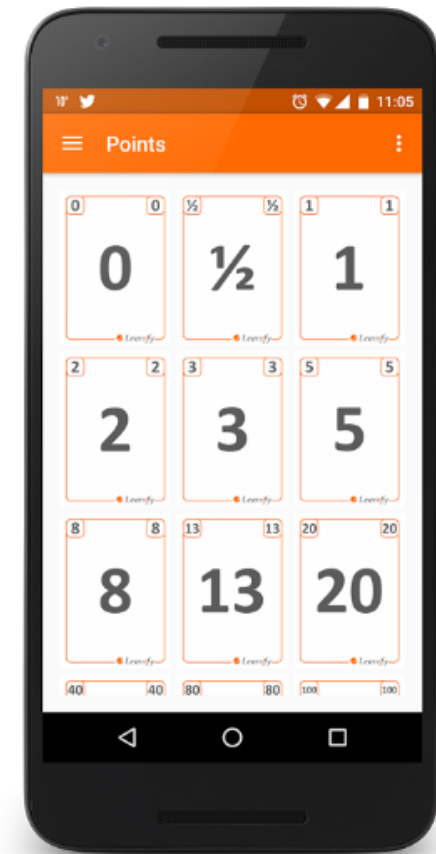
## PRÁCTICAS ÁGILES EN SCRUM

### Estimar Historias de Usuario

#### Planning Poker

El objetivo del *planning poker* es obtener una medida de **tamaño relativo** de todas las historias respecto una historia base.

Planning Poker es una técnica de estimación puesta en marcha por primera vez por James Grenning en un equipo Ágil utilizando XP en 2002, donde se utiliza una baraja de cartas con una distribución de números muy parecida a la secuencia de Fibonacci (0, 1/2, 1, 2, 3, 5, 8, 13, etc.).



## PRÁCTICAS ÁGILES EN SCRUM

### Estimar Historias de Usuario

#### Planning Poker

Para realizar una sesión de planning poker hay que seguir los siguientes pasos:

1. Reunir a todo el equipo y repartir las barajas a cada miembro o (iniciar la App).
2. El product owner, o moderador, lee una historia de usuario, y responde cualquier duda que tengan los miembros del equipo de desarrollo. Esto sirve para que todos tengan una comprensión en común de lo que hay que desarrollar.
3. Cada miembro del equipo de desarrollo selecciona una carta, equivalente a la estimación que el considera adecuada, y la pone boca abajo, cuando todos tengan seleccionada una carta, se ponen boca arriba todas a la vez.
4. Una vez mostradas las cartas, nos quedamos con la estimación media más elegida (moda), o se debate hasta conseguir la unanimidad.

Esto se repite con cada historia de usuario que haya que estimar.



## PRÁCTICAS ÁGILES EN SCRUM

### Estimar Historias de Usuario

#### Ventajas Planning Poker

- ⬡ Estimar colaborativamente ayuda a detectar riesgos, que se pueden ir solventando antes de comenzar la historia de usuario.
- ⬡ Promueve la participación y la colaboración.
- ⬡ Abstrae el concepto de tiempo.
- ⬡ Estimar entre varias personas nos acerca al máximo a la realidad, evitando márgenes de error muy grandes que afecten al desarrollo de producto.



#### Desventajas Planning Poker:

- ⬡ El llegar a un consenso, no garantiza el acierto
- ⬡ Requiere de un histórico de información
- ⬡ Cultura



## PRÁCTICAS ÁGILES EN SCRUM

### Estimación de HU usando Planning Poker

#	Historia de Usuario	Prioridad	Estimación
1	Como usuario deseo registrarse en el sistema para poder usarlo	MUST	1
2	Como usuario deseo iniciar sesión para poder ingresar al sistema	MUST	2
3	Como usuario deseo restablecer la contraseña para tener control de acceso al sistema	MUST	2
4	Como usuario deseo abrir una página de seguimiento de un producto para hacer seguimiento	MUST	2
5	Como usuario deseo elegir la forma de pago para tener diferentes opciones de pago	SHOULD	1
6	Como usuario deseo eliminar la cuenta para retirarme del sistema	SHOULD	1
7	Como usuario deseo elegir opciones de seguimiento de un producto para saber donde esta en cada momento	SHOULD	2
8	Como usuario deseo tener una versión de móvil de la aplicación para usarla desde el teléfono	COULD	24
9	Como usuario deseo elegir el tema visual de la aplicación para tener diferentes formas de ver la aplicación.	WON'T	2

**Recomendación:** Si una historia de usuario tiene más de 8 puntos divídala en varias historias.



## Preguntas ?





## Próxima clase

- Prácticas ágiles en detalle
- BPM
- Metodología Scrum





Desarrollo I

Economy of the  
European Union

# Gracias

