

# Ingeniería de software I





# ¿Qué veremos hoy?

## ✘ Otros ciclos de vida:

- RUP

## ✘ Como enfrentar el cambio

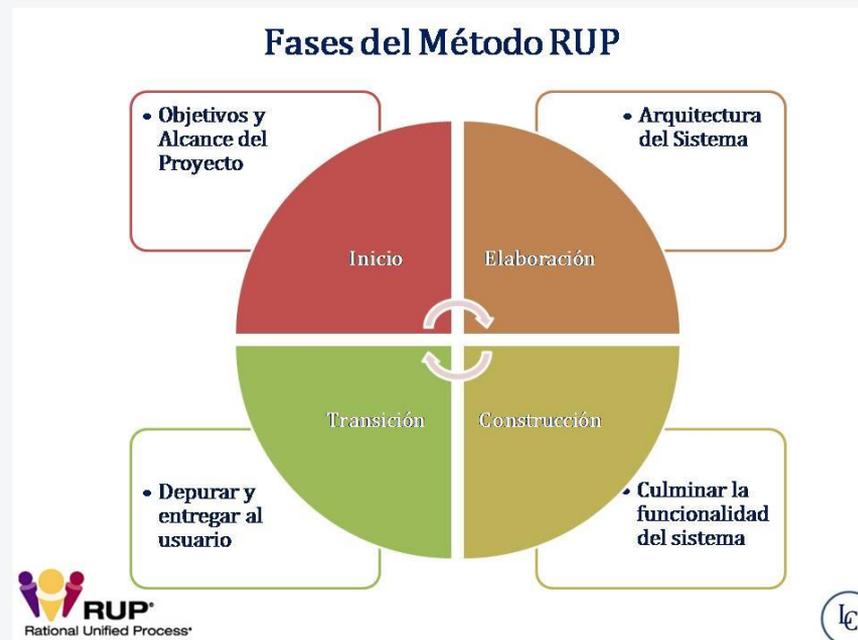
- Desarrollo incremental
- Prototipo

## ✘ Desarrollos ágiles

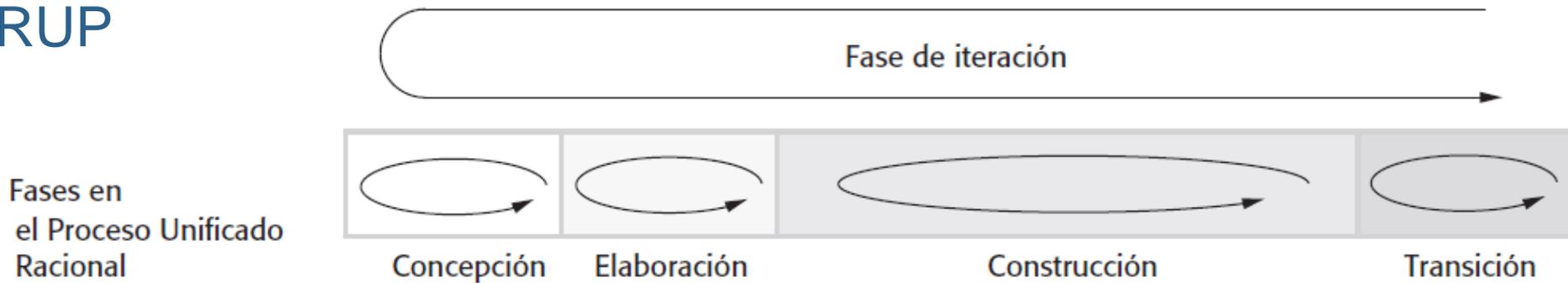


# *Otro ciclo de vida*

# RUP

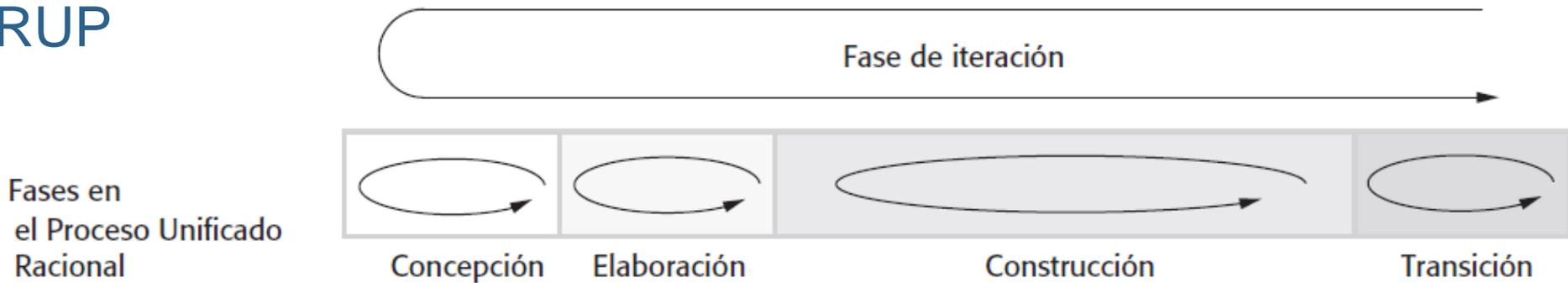


# RUP



**Concepción:** establece un **caso empresarial** para el sistema. Deben identificarse todas las **entidades externas** (personas y sistemas) que interactuarán con el sistema y definirán dichas interacciones. Luego se usa esta información para **valorar** el aporte del sistema hacia la empresa. Si este aporte es menor, entonces el proyecto puede cancelarse después de esta fase

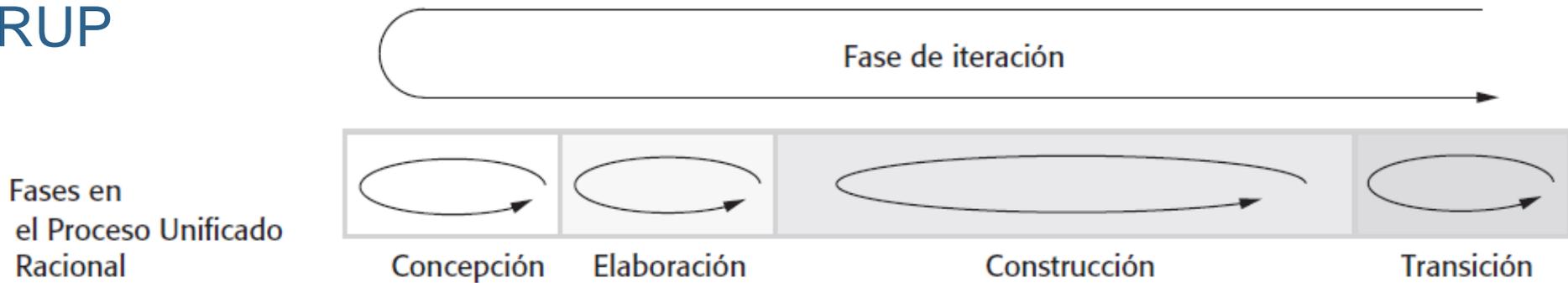
# RUP



**Elaboración:** desarrollar la **comprensión** del problema de dominio, establecer un **marco conceptual arquitectónico** para el sistema, diseñar el **plan del proyecto** e **identificar los riesgos clave** del proyecto.

Al completar esta fase, debe tenerse un **modelo de requerimientos** para el sistema, que podría ser una serie de **casos de uso del UML**, una descripción arquitectónica y un plan de desarrollo para el software.

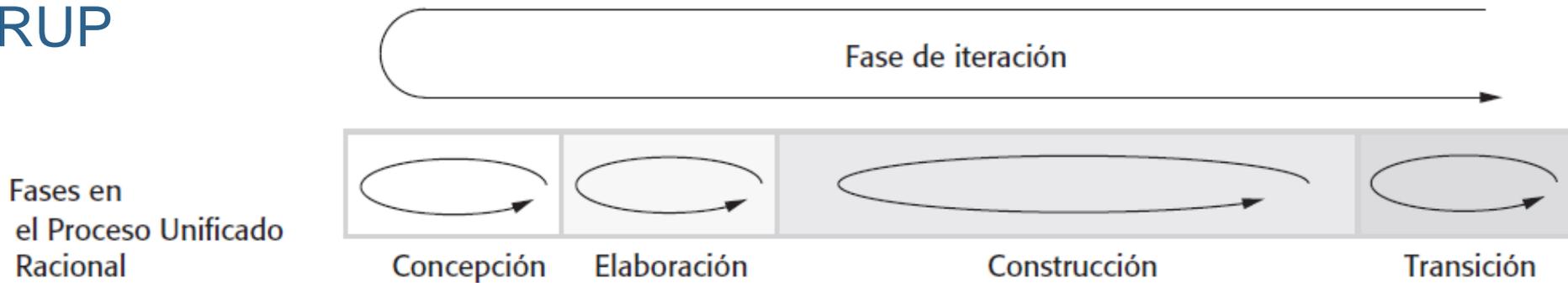
# RUP



**Construcción:** incluye diseño, programación y pruebas del sistema. Partes del sistema se desarrollan en paralelo y se integran durante esta fase.

Termina con un sistema de software funcionando y la documentación relacionada y lista para entregarse al usuario.

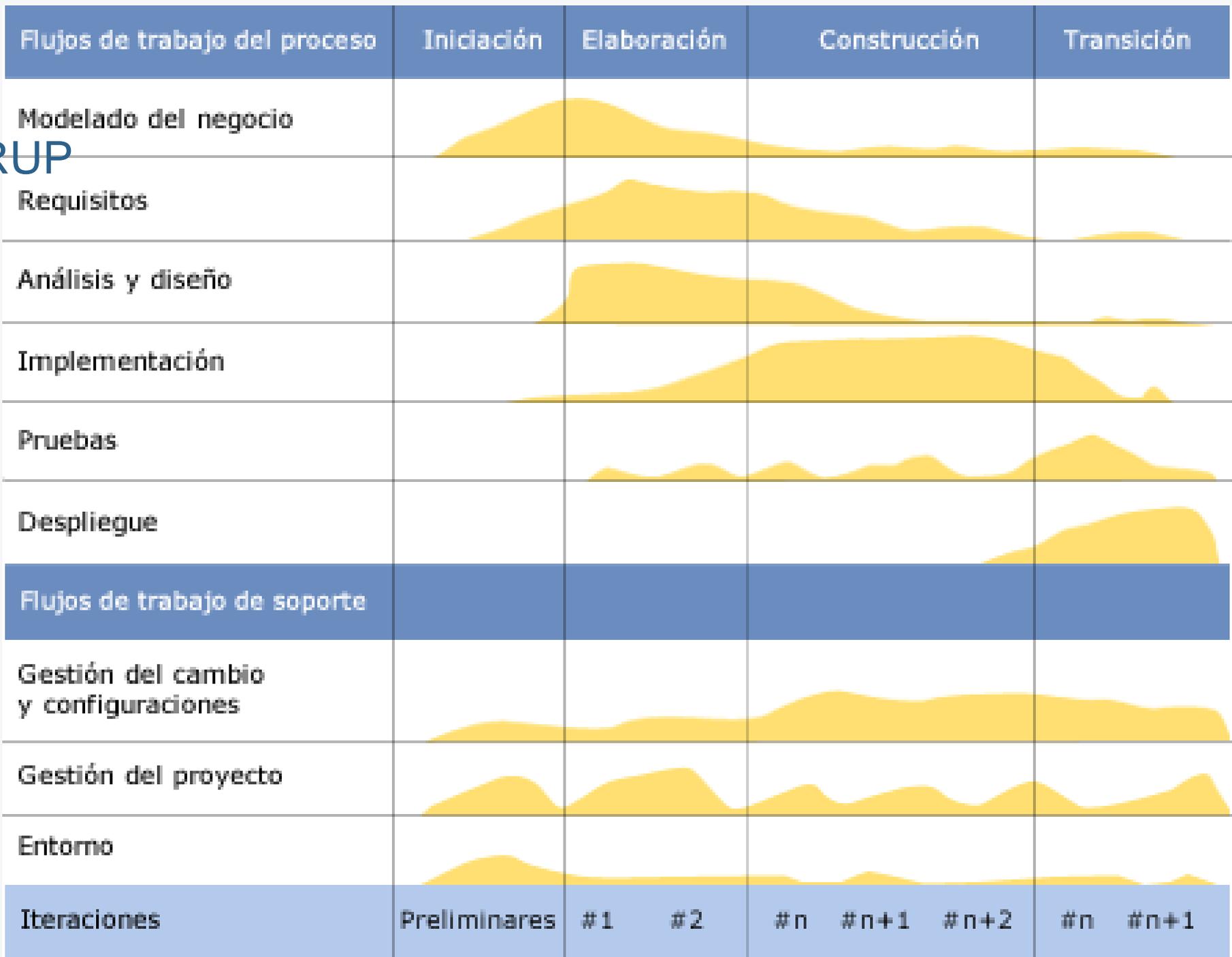
# RUP



**Transición** La fase final del RUP se interesa por el cambio del sistema desde la comunidad de desarrollo hacia la comunidad de usuarios, y por ponerlo a ***funcionar en un ambiente real***. Esto es algo ignorado en la mayoría de los modelos de proceso de software aunque, en efecto, es una actividad costosa y en ocasiones problemática.

En el complemento de esta fase se debe tener un sistema de software documentado que funcione correctamente en su entorno operacional.

# RUP



# Cambios

***Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia.***

***El equipo cambia. Los miembros del equipo cambian.***

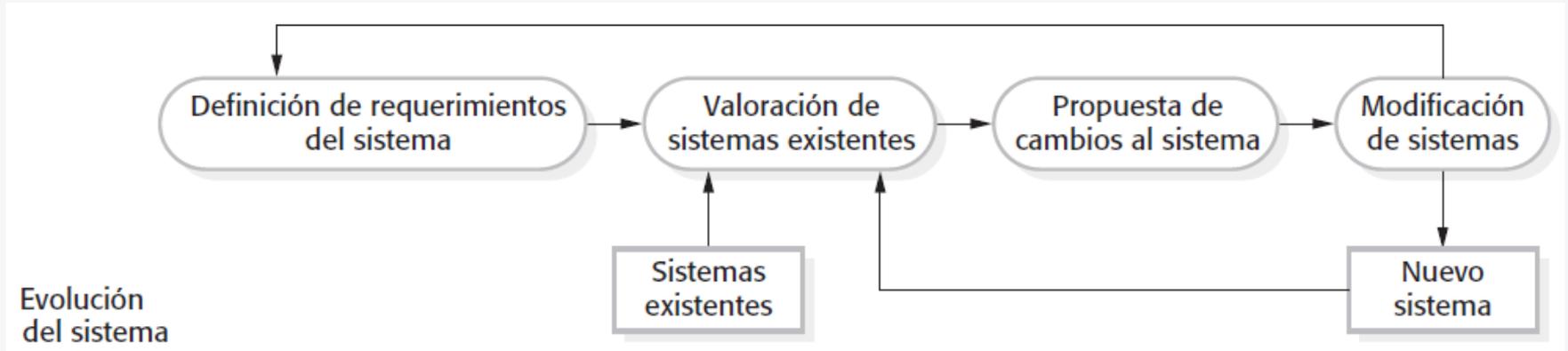
***El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar***

# Mantenimiento

*Mejor o peor que el desarrollo?*

Tiene mucho más sentido ver el desarrollo y el mantenimiento como un continuo.

# Cómo enfrentar el cambio



Dos enfoques relacionados que se usan para reducir los costos del rehacer

**Evitar el cambio:** incluye actividades que anticipan cambios posibles

**Tolerancia al cambio:** Los cambios propuestos pueden implementarse en incrementos que aún no se desarrollan

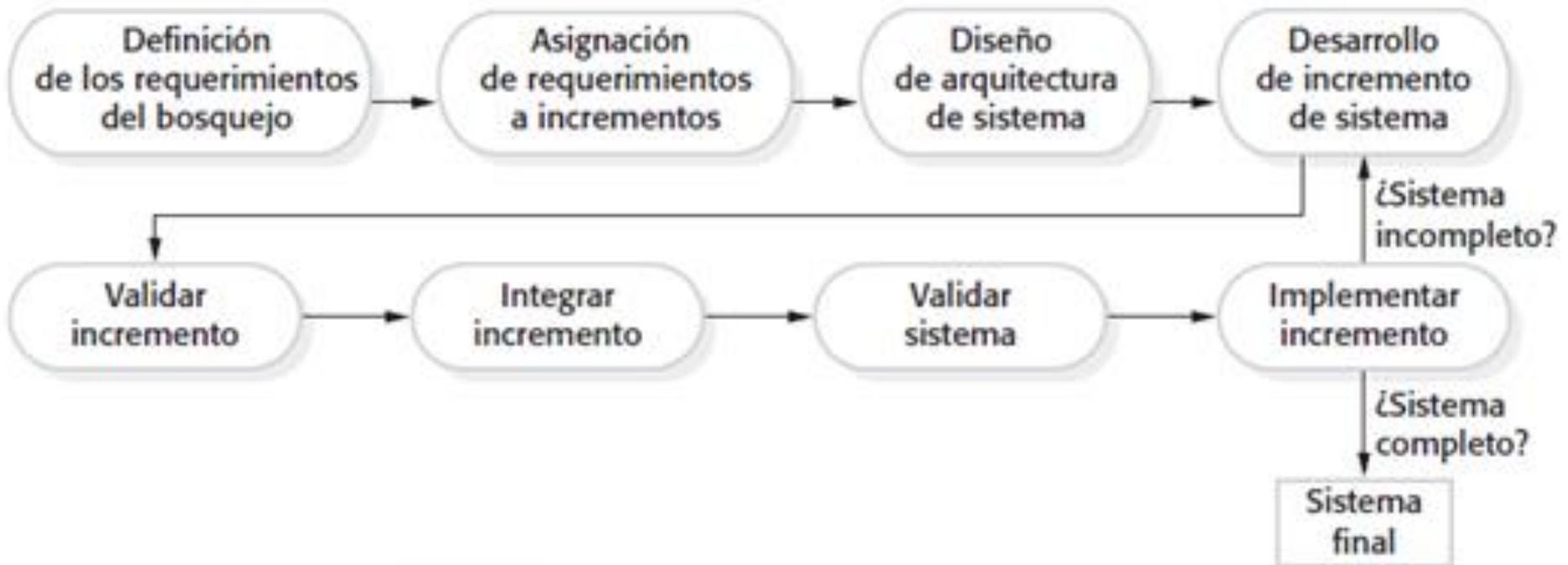
# **Cómo enfrentar el cambio**

Entrega incremental

Prototipo de sistema

# Entrega incremental

La entrega incremental es un enfoque donde **algunos** de los **incrementos** diseñados se **entregan** al cliente y se implementan para usarse en un entorno operacional



## Incremental

Cuando se utiliza un modelo incremental, el primer incremento es a menudo un producto esencial, sólo con los requisitos básicos. Este modelo se centra en la entrega de un producto operativo con cada incremento. Los primeros incrementos son versiones incompletas del producto final, pero proporcionan al usuario la funcionalidad que precisa y también una plataforma para la evaluación

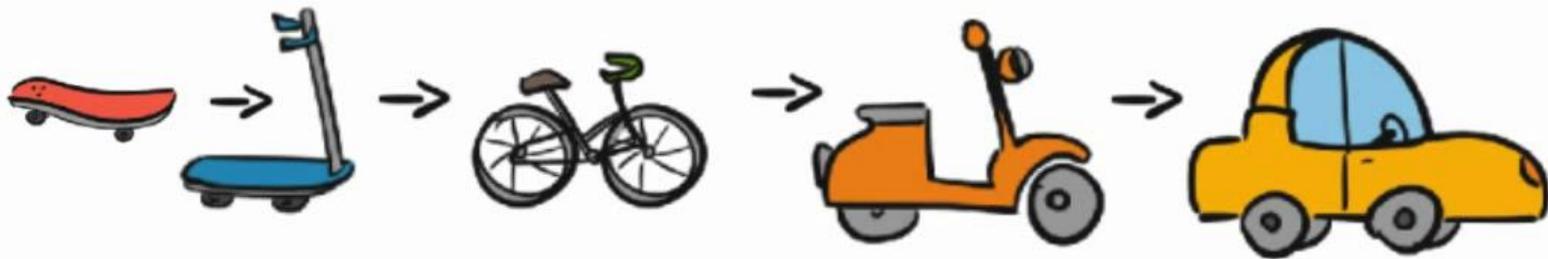
## **Ventajas de la entrega incremental:**

1. A diferencia de los prototipos, la **entrega incremental** es parte del **sistema real**.
2. El **primer** incremento cubre sus **requerimientos** más **críticos**, de modo que es posible usar inmediatamente el software.
3. El proceso mantiene los beneficios del desarrollo incremental en cuanto a que debe ser relativamente sencillo incorporar cambios al sistema.

## **Desventajas de la entrega incremental:**

1. Dificultad en identificar recursos comunes
2. Cuando se diseña un sistema de reemplazo
3. No hay especificación completa del sistema, hasta que se define el incremento final.

## MÓDELO ITERATIVO E INCREMENTAL





MODEL ITERATIVE

---

MODEL ITERATIVO E INCREMENTAL



# Creación del prototipo

Un prototipo es una versión inicial de un sistema de software que se usa para demostrar conceptos, tratar opciones de diseño y encontrar más sobre el problema y sus posibles soluciones

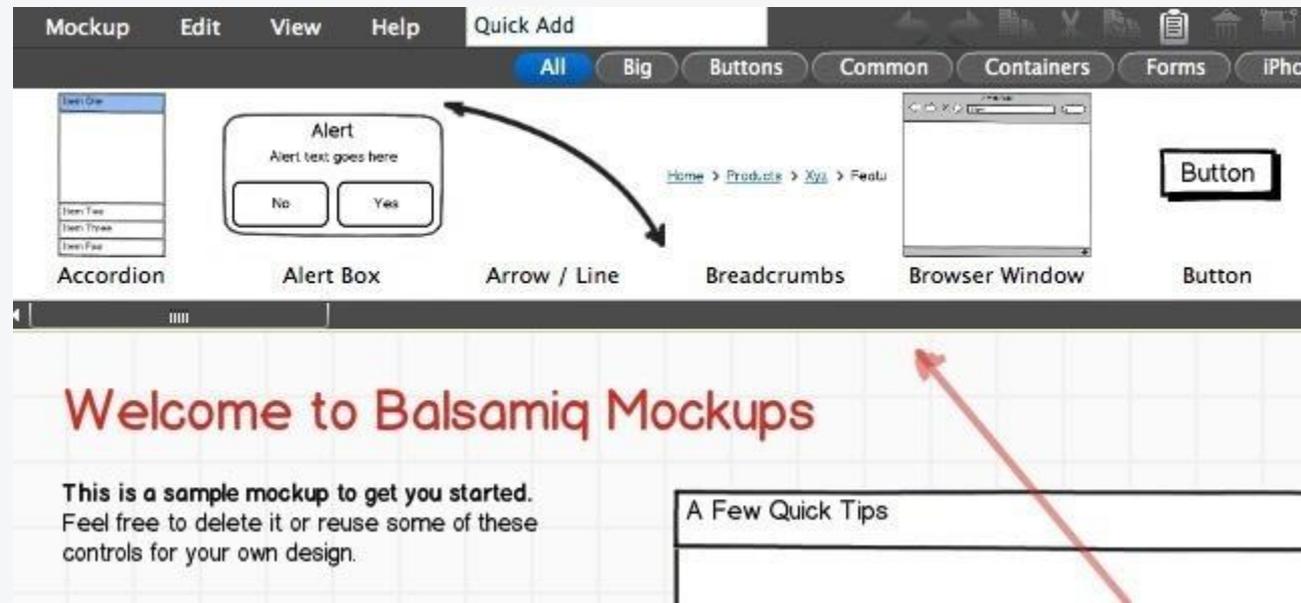


# Creación del prototipo

El rápido desarrollo iterativo del prototipo es esencial, para controlar costos y los usuarios experimenten por anticipado con el prototipo durante el proceso de software.

# prototipo

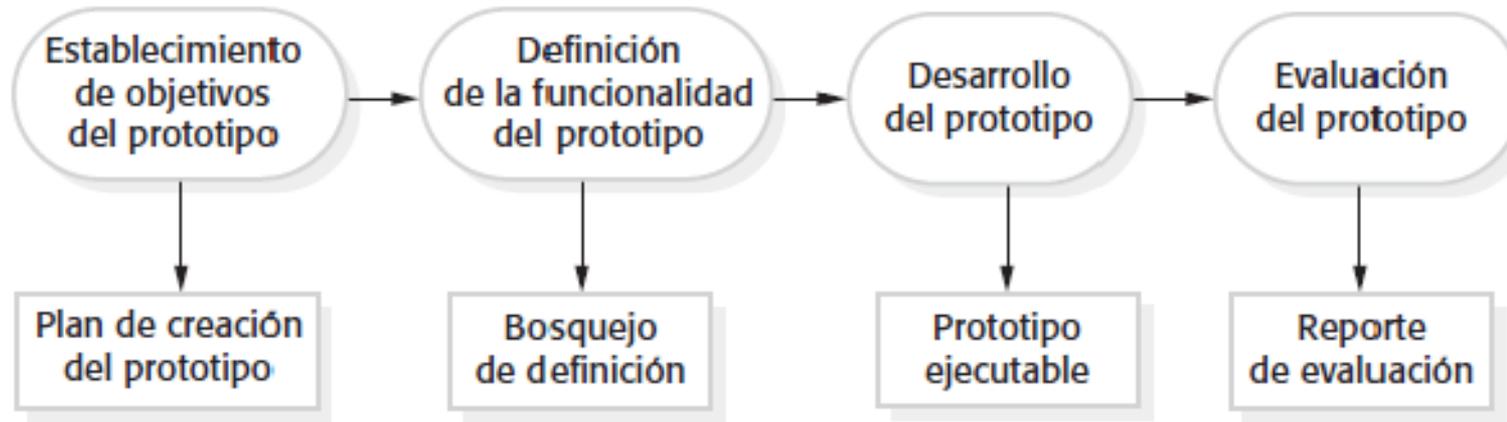
Obtener nuevas ideas para requerimientos y descubrir áreas de fortalezas y debilidades en el software.



# Creación del prototipo

La creación de prototipos es una parte esencial del proceso de diseño de interfaz del usuario.

# Creación del prototipo



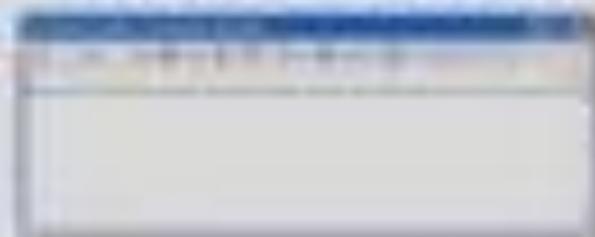
Proceso de desarrollo del prototipo

# Desventajas



UICAM

## Tipos de prototipos



**PROTOTIPO DE PANTALLA**  
 Representación de la interfaz de usuario que muestra la estructura general de la pantalla y la disposición de los elementos de la interfaz.

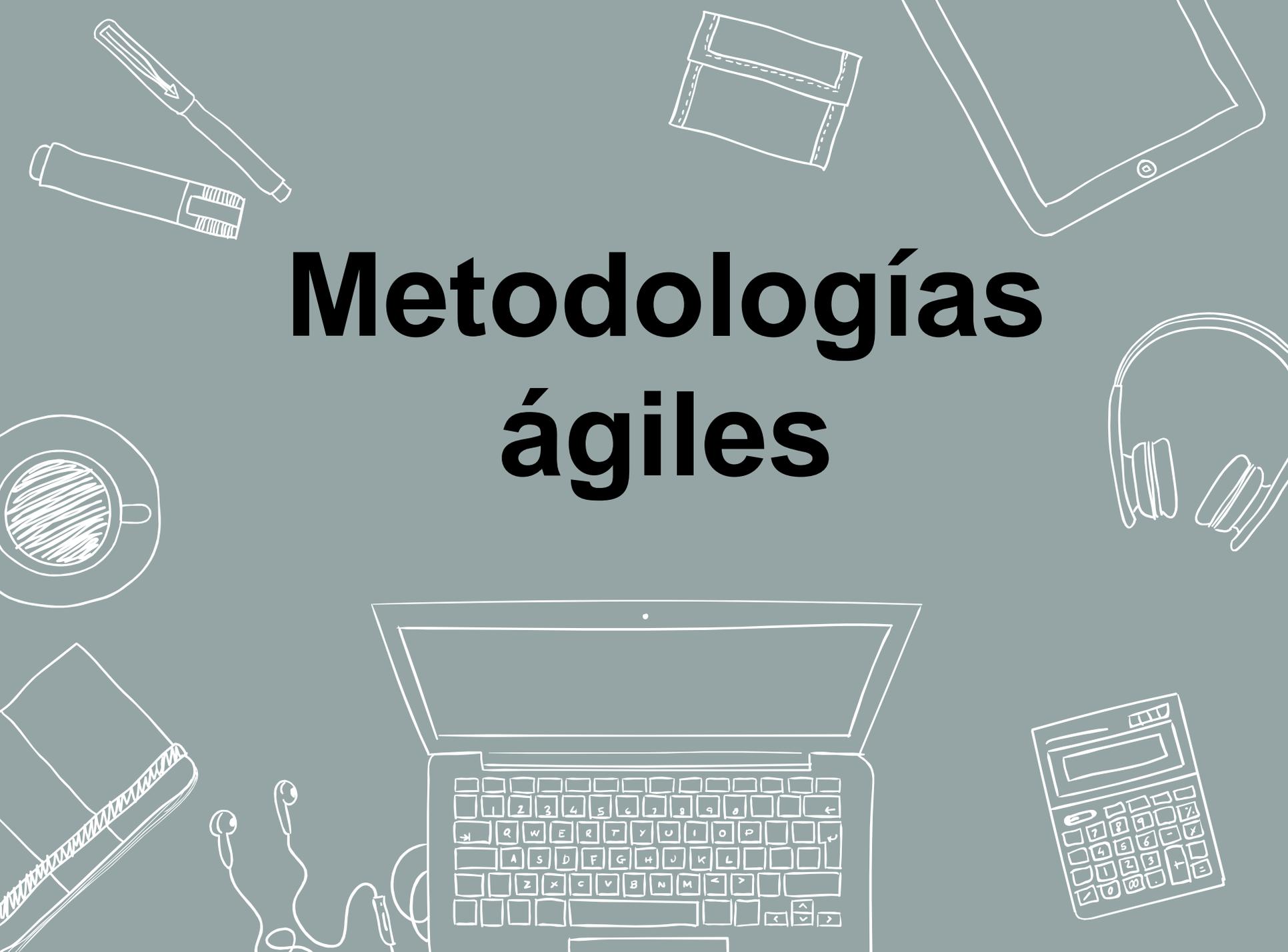
**PROTOTIPO DE FLUJO DE TRABAJO**  
 Representación de la secuencia de pantallas que el usuario verá al interactuar con el sistema.

**PROTOTIPO DE INTERFAZ DE USUARIO**  
 Representación de la interfaz de usuario que muestra la estructura general de la pantalla y la disposición de los elementos de la interfaz.

**PROTOTIPO DE INTERFAZ DE USUARIO**  
 Representación de la interfaz de usuario que muestra la estructura general de la pantalla y la disposición de los elementos de la interfaz.



# Metodologías ágiles





**Metodología ágil**

# Características

Procesos entrelazados

Sistemas desarrollados en diferentes versiones

Sistema de elaboración interactivo para interfaces

**Metodología ágil**



# Los principios de los métodos ágiles

Principio	Descripción
Participación del cliente	Los clientes deben intervenir estrechamente durante el proceso de desarrollo. Su función consiste en ofrecer y priorizar nuevos requerimientos del sistema y evaluar las iteraciones del mismo.
Entrega incremental	El software se desarrolla en incrementos y el cliente especifica los requerimientos que se van a incluir en cada incremento.
Personas, no procesos	Tienen que reconocerse y aprovecharse las habilidades del equipo de desarrollo. Debe permitirse a los miembros del equipo desarrollar sus propias formas de trabajar sin procesos establecidos.
Adoptar el cambio	Esperar a que cambien los requerimientos del sistema y, de este modo, diseñar el sistema para adaptar dichos cambios.
Mantener simplicidad	Enfocarse en la simplicidad tanto en el software a desarrollar como en el proceso de desarrollo. Siempre que sea posible, trabajar de manera activa para eliminar la complejidad del sistema.

**Metodología ágil**

Los métodos ágiles son:

Métodos de desarrollo incremental  
(incrementos mínimos)

Se crean las nuevas liberaciones del sistema, cada dos o tres semanas se ponen a disposición de los clientes.

Involucran a los clientes en el proceso de desarrollo para conseguir una rápida retroalimentación sobre los requerimientos cambiantes.

Minimizan la cantidad de documentación con el uso de comunicaciones informales, en vez de reuniones formales con documentos escritos.

Los métodos ágiles para

**1.** Desarrollo del producto, software pequeño o mediano para su venta.

**2.** Diseño de sistemas a la medida dentro de una organización, donde hay un claro compromiso del cliente por intervenir en el proceso de desarrollo, y donde no existen muchas reglas ni regulaciones externas que afecten el software.

# ¿Documentamos?





# Principios del Manifiesto Ágil

*Seguimos estos principios:*

Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

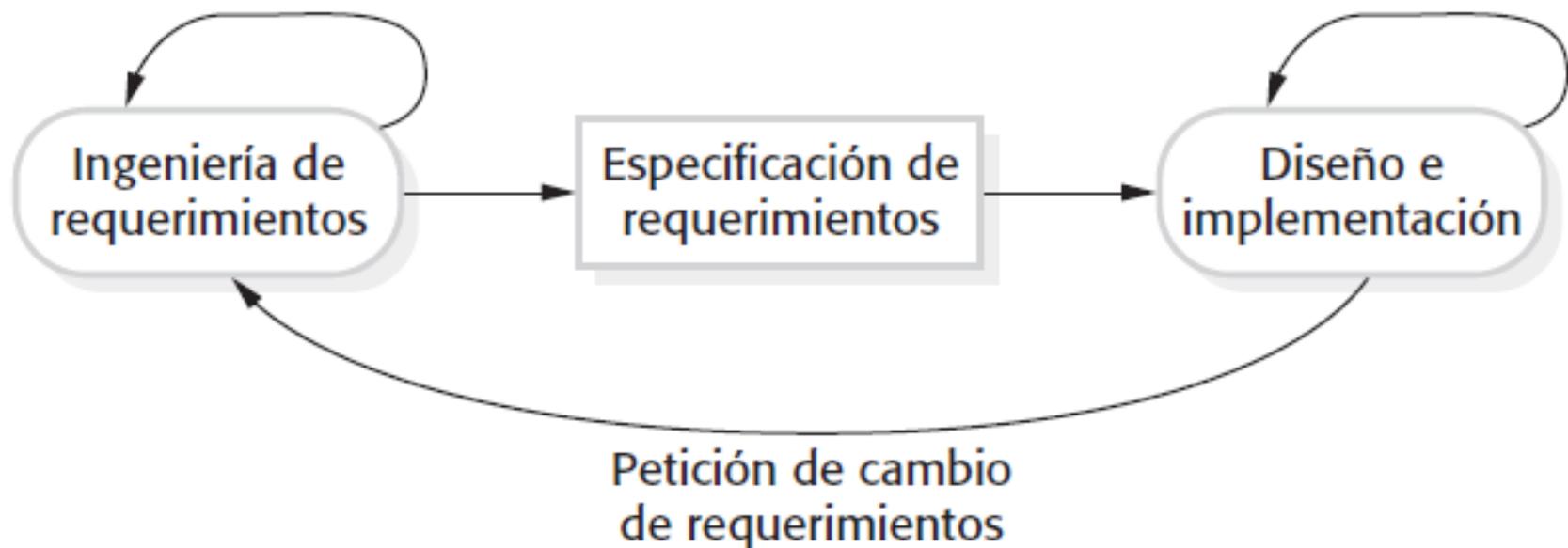
Los proyectos se desarrollan en torno a individuos

# Desarrollo dirigido por un plan y desarrollo ágil.

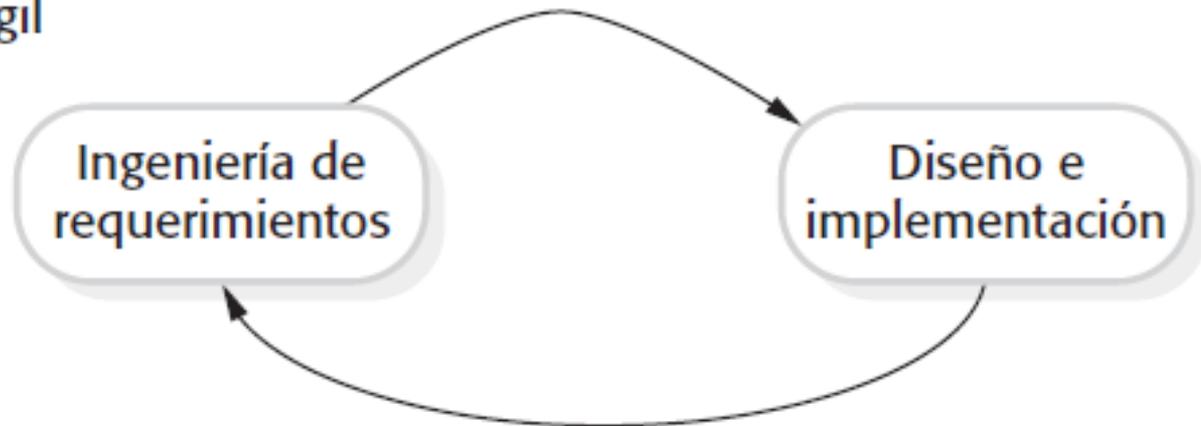
Trabajamos con las diferencias



## Desarrollo basado en un plan



## Desarrollo ágil

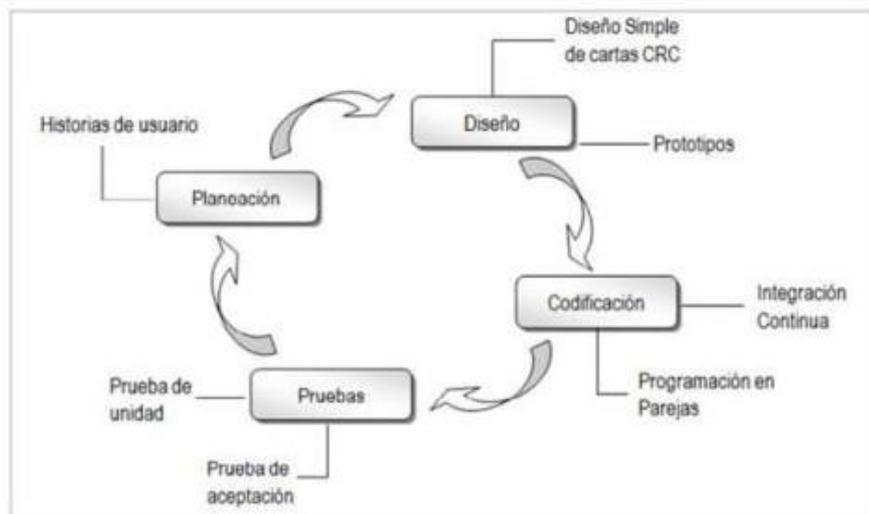
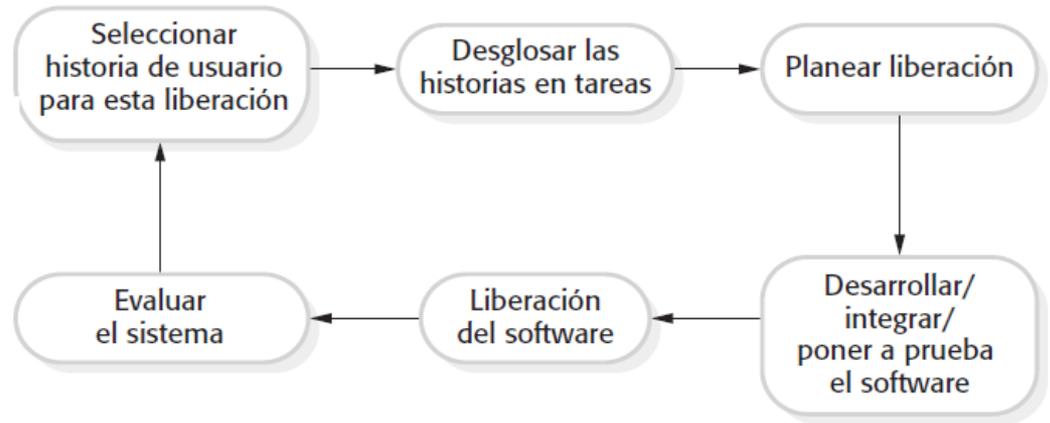


# Métodos ágeiles o por plan



# Programación extrema

El ciclo de liberación de la programación extrema



# WHAT IS XP?



Lightweight



Discipline

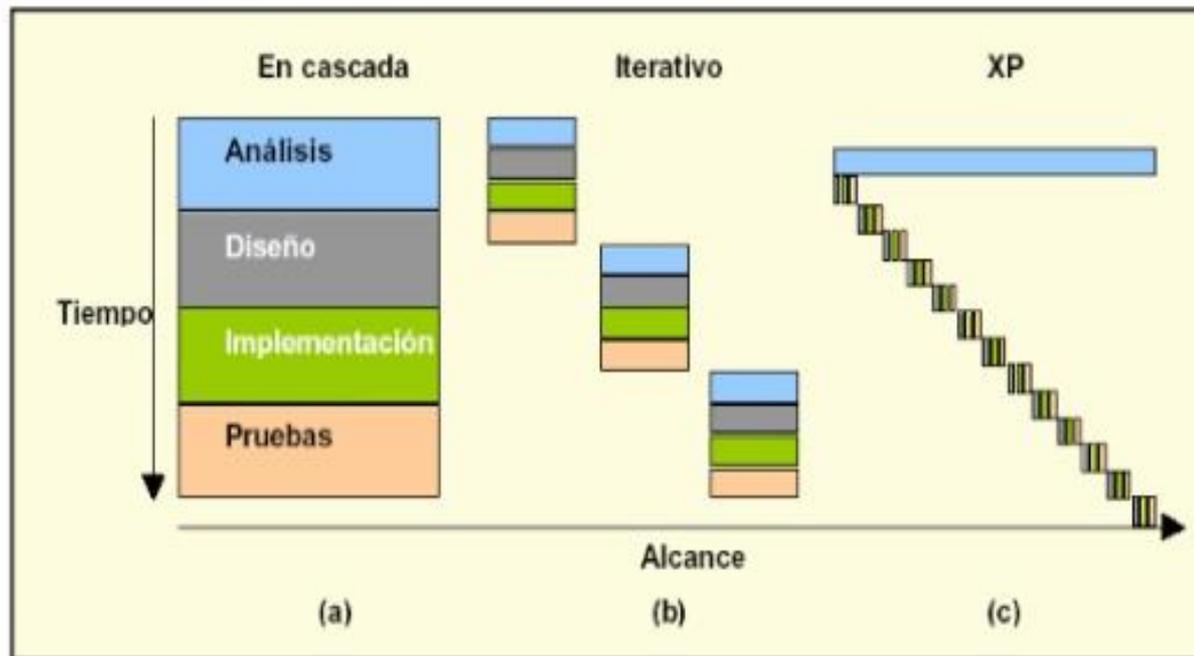


Humanistic

software  
development

# Programación extrema

## Ciclo de vida



un proyecto con XP lleva 10 a 15 ciclos o iteraciones

# Programación extrema

## Ventajas:

- Se consiguen productos usables con mayor rapidez.
- El proceso de integración es continuo, por lo que el esfuerzo final para la integración es nulo. Se consigue integrar todo el trabajo con mucha mayor facilidad.
- Se atienden las necesidades del usuario con mayor exactitud. Esto se consigue gracias a las continuas versiones que se ofrecen al usuario.
- Se consiguen productos más fiables y robustos contra los fallos gracias al diseño de los test de forma previa a la codificación.
- Obtenemos código más simple y más fácil de entender, reduciendo el número de errores.
- Gracias a la filosofía del “pair programming” (programación en parejas), se consigue que los desarrolladores apliquen las buenas prácticas que se les ofrecen con la XP.
- Gracias al “refactoring” es más fácil el modificar los requerimientos del usuario.
- Conseguimos tener un equipo de desarrollo más contento y motivado. Las razones son, por un lado el que la XP no permite excesos de trabajo (se debe trabajar 40 horas a la semana), y por otro la comunicación entre los miembros del equipo que consigue una mayor integración entre ellos

# Programación extrema

## **Desventajas**

- Resulta muy complicado planear el proyecto y establecer el costo y la duración del mismo.
- No se puede aplicar a proyectos de gran escala, que requieran mucho personal, a menos que se las subdivide en proyectos más pequeños.
- Es más complicado medir los avances del proyecto, pues es muy complicado el uso de una medida estándar.
- Altas comisiones en caso de fallar.

Principio o práctica	Descripción
Planeación incremental	Los requerimientos se registran en tarjetas de historia ( <i>story cards</i> ) y las historias que se van a incluir en una liberación se determinan por el tiempo disponible y la prioridad relativa. Los desarrolladores desglosan dichas historias en "tareas" de desarrollo.
Liberaciones pequeñas	Al principio se desarrolla el conjunto mínimo de funcionalidad útil, que ofrece valor para el negocio. Las liberaciones del sistema son frecuentes y agregan incrementalmente funcionalidad a la primera liberación.
Diseño simple	Se realiza un diseño suficiente para cubrir sólo aquellos requerimientos actuales.
Desarrollo de la primera prueba	Se usa un marco de referencia de prueba de unidad automatizada al escribir las pruebas para una nueva pieza de funcionalidad, antes de que esta última se implemente.
Refactorización	Se espera que todos los desarrolladores refactoricen de manera continua el código y, tan pronto como sea posible, se encuentren mejoras de éste. Lo anterior conserva el código simple y mantenible.
Programación en pares	Los desarrolladores trabajan en pares, y cada uno comprueba el trabajo del otro; además, ofrecen apoyo para que se realice siempre un buen trabajo.
Propiedad colectiva	Los desarrolladores en pares laboran en todas las áreas del sistema, de manera que no se desarrollan islas de experiencia, ya que todos los desarrolladores se responsabilizan por todo el código. Cualquiera puede cambiar cualquier función.
Integración continua	Tan pronto como esté completa una tarea, se integra en todo el sistema. Después de tal integración, deben aprobarse todas las pruebas de unidad en el sistema.
Ritmo sustentable	Grandes cantidades de tiempo extra no se consideran aceptables, pues el efecto neto de este tiempo libre con frecuencia es reducir la calidad del código y la productividad de término medio.
Cliente en sitio	Un representante del usuario final del sistema (el cliente) tiene que disponer de tiempo completo para formar parte del equipo XP. En un proceso de programación extrema, el cliente es miembro del equipo de desarrollo y responsable de llevar los requerimientos del sistema al grupo para su implementación.

## Roles de la Metodología XP



## Historia de Usuario

Número: 1      Nombre: Enviar artículo

Usuario: Autor

Modificación de Historia Número:      Iteración Asignada: 2

Prioridad en Negocio: Alta  
(Alta / Media / Baja)

Puntos (Estimados):

Riesgo en Desarrollo:  
(Alto / Medio / Bajo)

Puntos Reales:

Descripción:

Se introducen los datos del artículo (título, fichero adjunto, resumen, tópicos) y de los autores (nombre, e-mail, afiliación). Uno de los autores debe indicarse como autor de contacto. El sistema confirma la correcta recepción del artículo enviando un e-mail al autor de contacto con un userid y password para que el autor pueda posteriormente acceder al artículo.

Observaciones:

# Programación en pares



# Administración de un proyecto ágil



# Scrum



Metodología ágil

# Características

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizarlas una tras otra en un ciclo secuencial o de cascada.

# REGLAS



## Marco de reglas para desarrollo de software

Autores: Ken Schwaber y Jeff Sutherland  
"Scrum Development Process OOPSLA'95" 1995

### Aplicación de reglas definidas

#### Roles

- Dueño de producto
- Equipo de desarrollo
- Scrum Master

#### Eventos

- El Sprint
- Reunión de planificación
- Scrum diario
- Revisión de sprint
- Retrospectiva de sprint

#### Artefactos

- Pila de product
- Pila de sprint
- Incremento

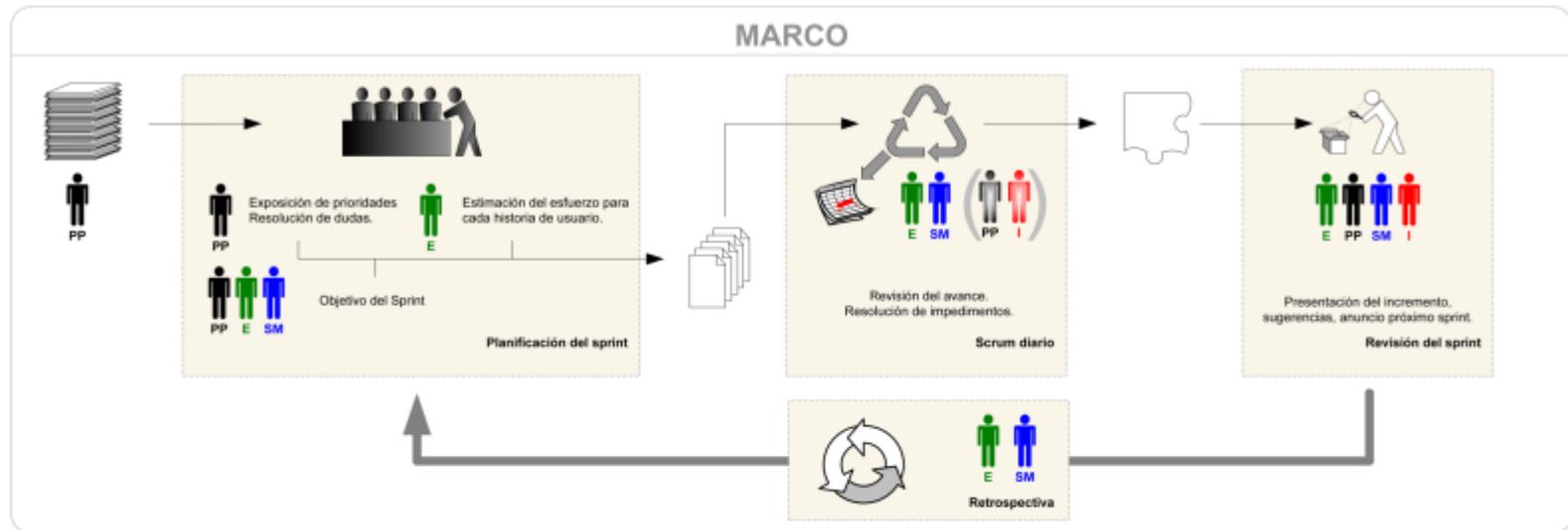
# MARCO DE TRABAJO

Scrum



## LAS REGLAS DE SCRUM

Rev.1.1



### ROLES

- PROPIETARIO DEL PRODUCTO**  
Determina las prioridades. Una sola persona.
- EQUIPO DE DESARROLLO**  
Construye el producto.
- SCRUM MASTER**  
Gestiona y facilita la ejecución de las reglas de Scrum.
- INTERESADOS**  
Resto de implicados. Asesoran y observan.

### ARTEFACTOS

- PILA DEL PRODUCTO**  
Relación de requisitos del producto, no es necesario excesivo detalle. Priorizados. Lista en evolución y abierta a todos los roles. El propietario del producto es su responsable y quien decide.
- PILA DEL SPRINT**  
Requisitos comprometidos por el equipo para el sprint con nivel de detalle suficiente para su ejecución.
- INCREMENTO**  
Parte del producto desarrollado en un sprint, en condiciones de ser usado (pruebas, codificación limpia y documentada).

### EVENTOS

- PLANIFICACIÓN DEL SPRINT**  
1 jornada de trabajo (máx.). El propietario del producto explica las prioridades. El equipo estima el esfuerzo de los requisitos prioritarios y se elabora la pila del sprint. El equipo define en una frase el objetivo del sprint.
- SPRINT**  
Ciclo de desarrollo básico en el marco estándar de scrum, de duración recomendada inferior a un mes y nunca mayor de 6 semanas.
- SCRUM DIARIO**  
15 minutos máximo. Responsabilidad del equipo. Cada miembro expone: Lo que hizo ayer. Lo que va a hacer hoy, si tiene o prevé problemas. Se actualiza la pila del sprint.
- REVISIÓN DEL SPRINT**  
Informativa, máx. 4 horas, presentación del incremento, planteamiento de sugerencias y anuncio del próximo sprint.
- RETROSPECTIVA**  
El equipo autoanaliza la forma de trabajo. Identificación de fortalezas y debilidades. Refuerzo de las primeras, plan de mejora de las segundas.

Ilustración 5: Marco scrum técnico

# ROLES

- Propietario del producto (Product Owner): es la persona responsable de lograr el mayor valor de producto para los clientes, usuarios y resto de implicados.
- Equipo de desarrollo: grupo o grupos de trabajo que desarrollan el producto.
- Scrum Master: Es el responsable del cumplimiento de las reglas de un marco de scrum técnico, asegurando que se entienden en la organización, y se trabaja conforme a ellas. Proporciona la asesoría y formación necesaria al propietario del producto y al equipo. Realiza su trabajo con un modelo de liderazgo servil: al servicio y en ayuda del equipo y del propietario del producto.

# EVENTOS

- ❖ Sprint: nombre que recibe cada iteración de desarrollo. Es el núcleo central que genera el pulso de avance a ritmo de “tiempos prefijados” (time boxing).
- ❖ Reunión de Planificación del sprint: reunión de trabajo que marca el inicio de cada sprint en la que se determina cuál es el objetivo del sprint y las tareas necesarias para conseguirlo.
- ❖ Scrum diario: breve reunión diaria del equipo, en la que cada miembro responde a tres cuestiones: 1.- El trabajo realizado el día anterior. 2.- El que tiene previsto realizar. 3.- Cosas que puede necesitar, o impedimentos que deben eliminarse para poder realizar el trabajo.
- ❖ Revisión del sprint: análisis e inspección del incremento generado, y adaptación de la pila del producto si resulta necesario. Una cuarta reunión se incorporó al marco estándar de scrum en la primera década de 2.000:
- ❖ Retrospectiva del sprint: revisión de lo sucedido durante el Sprint. Reunión en la que el equipo analiza aspectos operativos de la forma de trabajo y crea un plan de mejoras para aplicar en el próximo sprint.

# ARTEFACTOS

- ✓ Pila del producto: (product backlog) lista de requisitos de usuario, que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo.
- ✓ Pila del sprint: (sprint backlog) lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
- ✓ Incremento: resultado de cada sprint.

# PROCESO DE SCRUM



# HISTORIAS DE USUARIO

<p><i>Historia: Responder a comentarios</i></p> <p><i>Como: Lector del Blog</i></p> <p><i>Quiero: responder a comentarios de otros lectores</i></p> <p><i>Para: mantenerme en contacto con los demás usuarios del blog</i></p> <p><b>3</b></p>	<p><i>Historia: Agregar comentarios</i></p> <p><i>Como: Lector del Blog</i></p> <p><i>Quiero: adicionar comentarios a las entradas</i></p> <p><i>Para: mantenerme en contacto con el autor del blog</i></p> <p><b>3</b></p>	<p><i>Historia: Recibir alertas</i></p> <p><i>Como: Lector del Blog</i></p> <p><i>Quiero: recibir alertas cuando otros hagan comentarios a las entradas de mi elección</i></p> <p><i>Para: enterarme de lo que otros piensan sobre los temas de interés</i></p> <p><b>3</b></p>
--	---	---

Backlog Item #4  
Proyecto: LDAP

**Separar vista para Administradores / No Administradores.**

<p>Notas</p> <p>Permisos según perfiles:</p> <ul style="list-style-type: none"><li>- Administrador: Tendrá acceso a los informes de todos los distribuidores</li><li>- No administrador: Solo acceso a sus propios informes</li></ul>	<p>Prioridad</p> <p><b>4</b></p>
<p>Como probarlo</p> <p>Comprobar que un usuario Administrador puede adoptar el rol de un distribuidor y consultar sus informes.</p> <p>Comprobar que un usuario no administrador solo tiene acceso a sus propios informes.</p>	<p>Estimación</p> <p><b>5</b></p>

Tareas asociadas a esta historia de usuario:

Perfil administrador: opción para simular distribuidor.

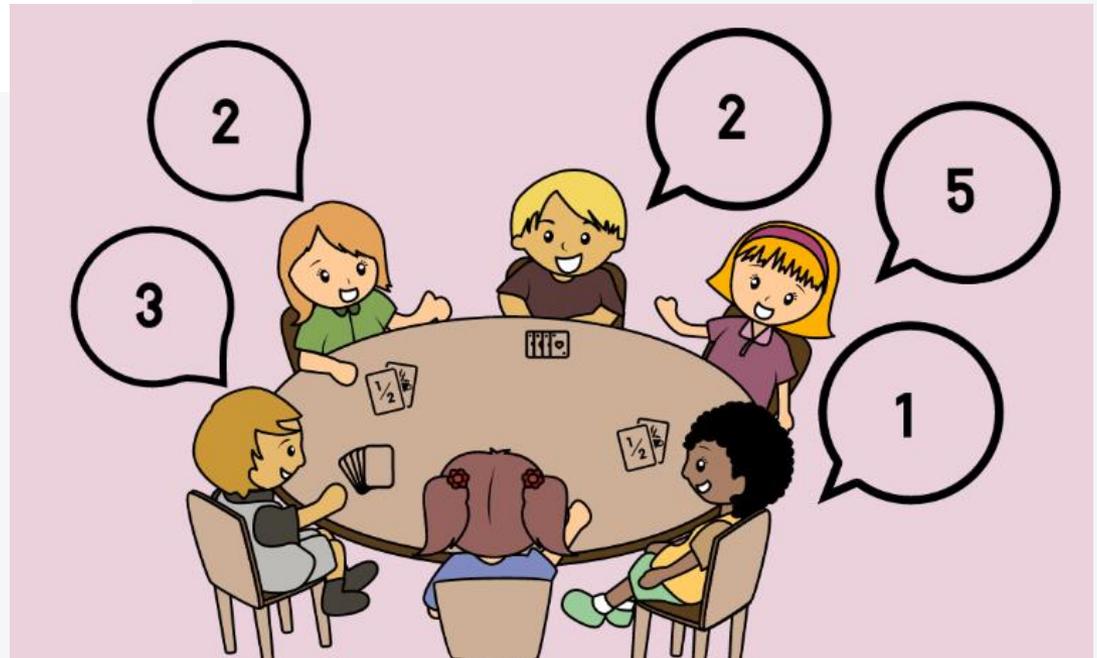
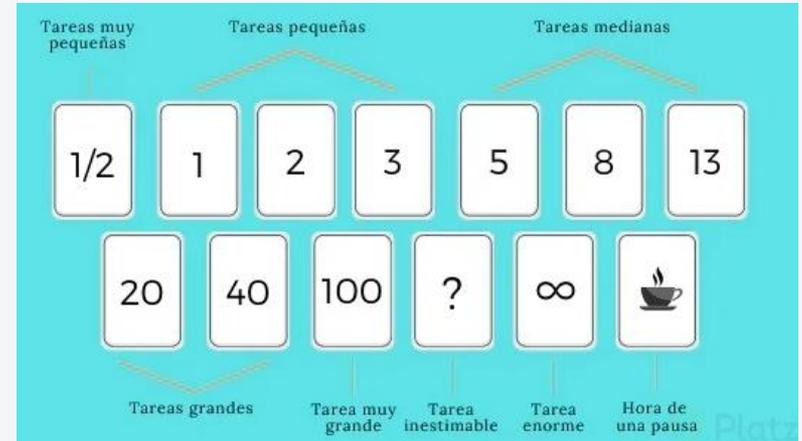
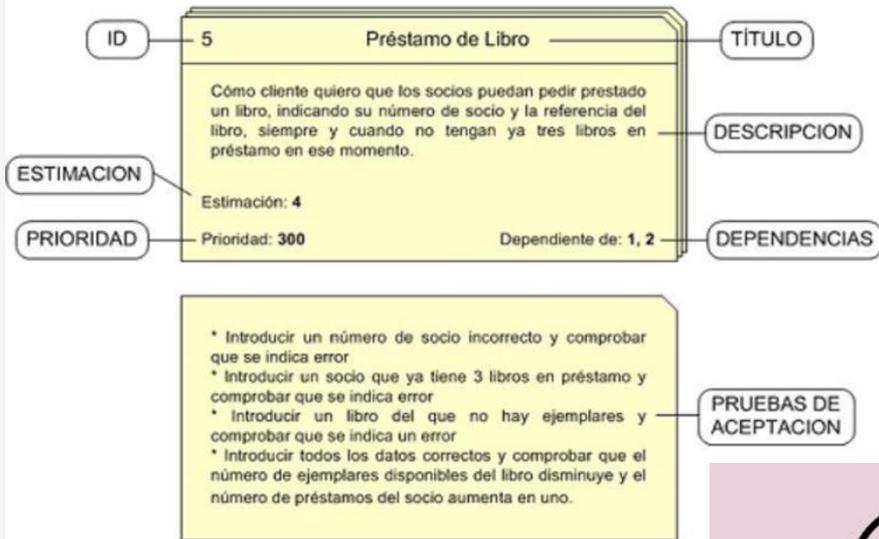
**3**

Perfil no administrador: Que no se vea la opción de simular distribuidor.

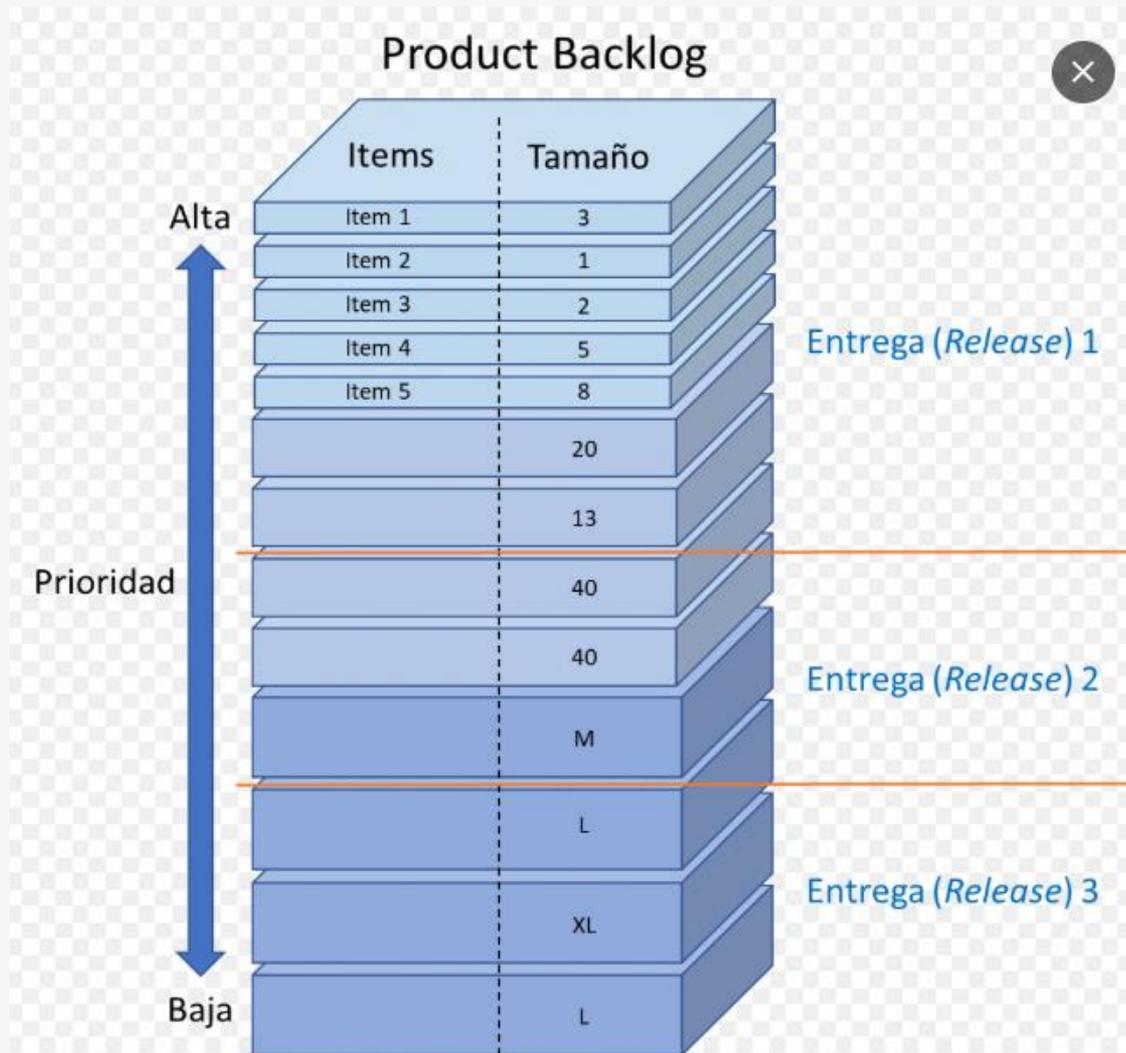
**2**

# ESTIMACIÓN HIST. USUARIOS

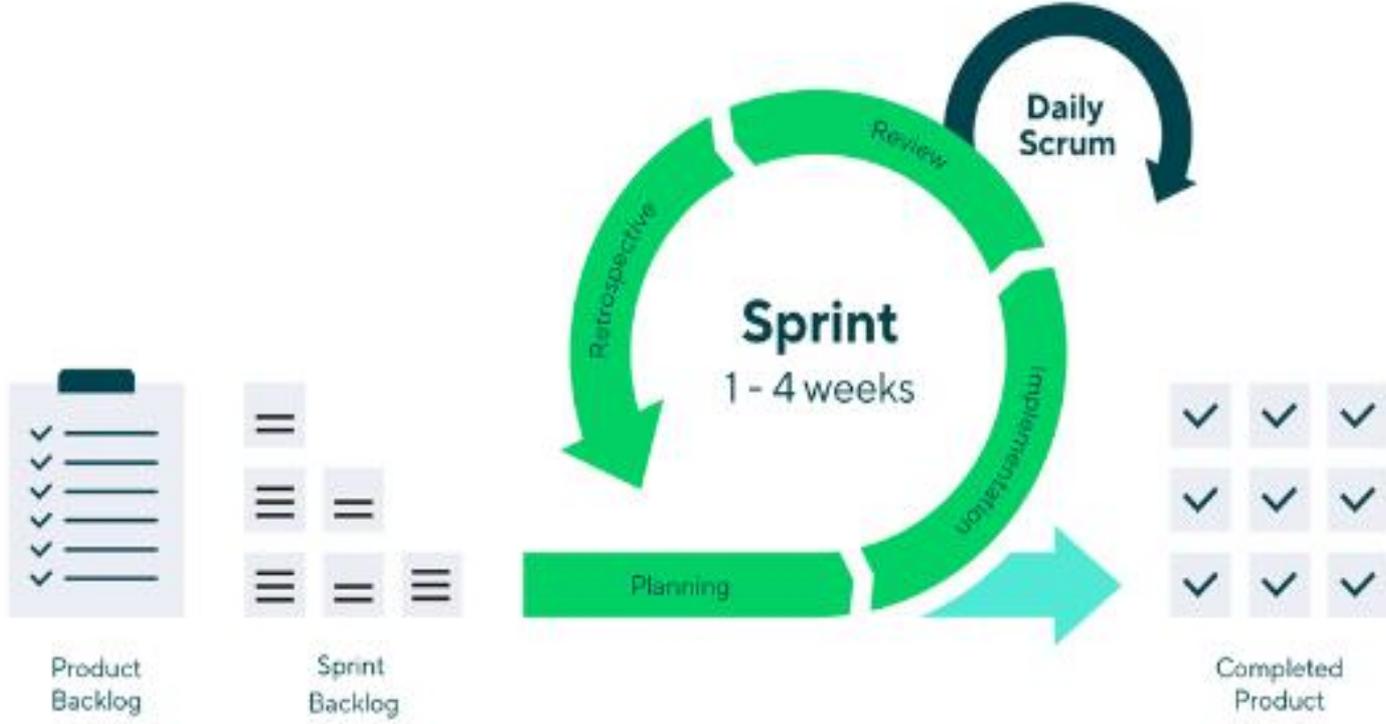
## ANVERSO



# Pila (Lista) de Producto



# SPRINT



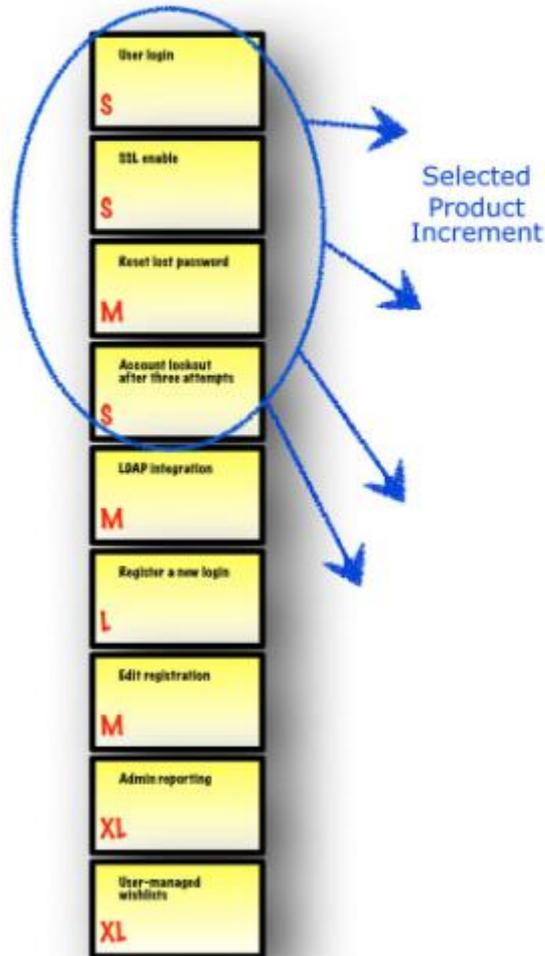
# Reunión Planificación Sprint



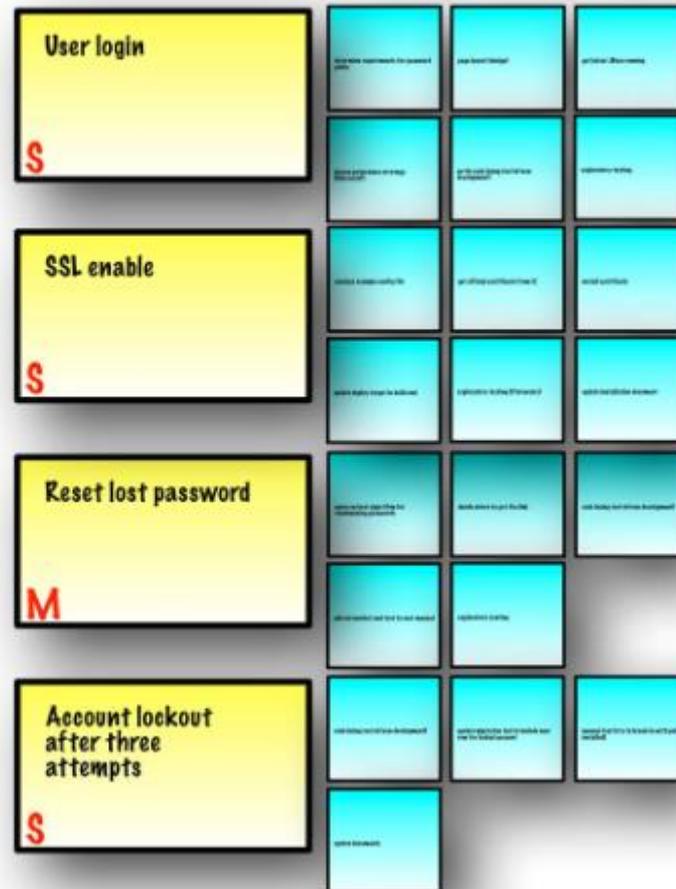
- Se determina el objetivo del Sprint.
- Se determina la Pila de Sprint con las historias de usuario de la Pila de Producto que se seleccionan por prioridad y que respeten el plazo del Sprint

# Pila (Lista) de Sprint

Product Backlog



Sprint Backlog



# Reunión Diaria

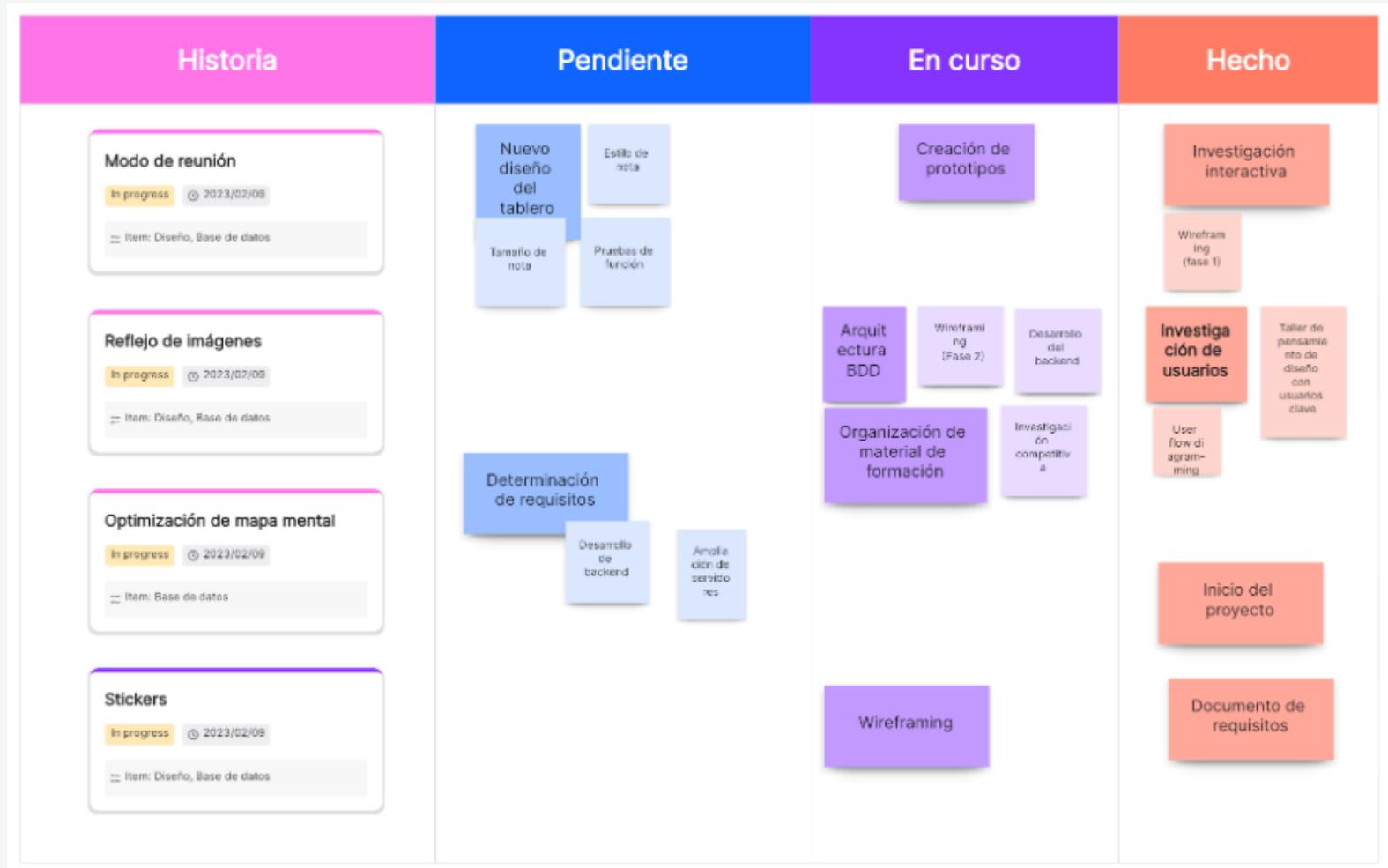


¿Qué hicimos ayer?

¿Qué hacemos hoy?

¿Qué problema encontramos para avanzar en la tarea por realizar?

# SEGUIMIENTO SPRINT



← → ↻ trello.com/b/K1yK6q6a/ti-ospelsym ☆ 🔍 📄 📱 🌐 LS

**Trello** Espacios de trabajo ▾ Reciente ▾ Marcado ▾ Plantillas ▾ Crear 🔍 Buscar 🔒 ⚙️ LS

A partir del 8 de abr., los Espacios de trabajo gratuitos se limitarán a 10 colaboradores. [Más información sobre los límites de colaboradores](#) ✕

**TI OSPELSYM** ☆ 🔒 🗨️ Tablero ▾ ⚡ ⚙️ Filtros LS FC JC M 👤 Compartir ⋮

- PENDIENTE SGI** ...
  - (0.1) Enviar correo cuando se realiza un descuento o una diferencia de coseguro para que veamos por que se realiza 📎 LS
  - cuando se crea un usuario se asocia con todas las delegaciones, cree un box por delegación y asocie el box a la delegación 📎 LS
  - Validar que los afiliados que están activos tengan fecha de alta y los que están de baja tengan fecha de...
- PENDIENTE SOTELSYM** ...
  - Ver el error que da con titulares y familiar cuando se quieren cambiar de grupo familiar
  - + Añada una tarjeta 📄
- PENDIENTE SITIO INTERNO** ...
  - + Añada una tarjeta 📄
- En proceso** ...
  - Relizar el módulo de ticket para sistemas 📎 LS
  - + Añada una tarjeta 📄
- En Espera** ...
  - (0.7) Cambio de Plan autom... 📎
  - + Añada una tarjeta 📄

STORIES	TODO	IN PROGRESS	TESTING	SPRINT ENDS 12/22
	2			
	2			
	3			
	2			
	2			
	4			
	2			
	2			

# Reunión Revisión Sprint - Visualizar Incremento

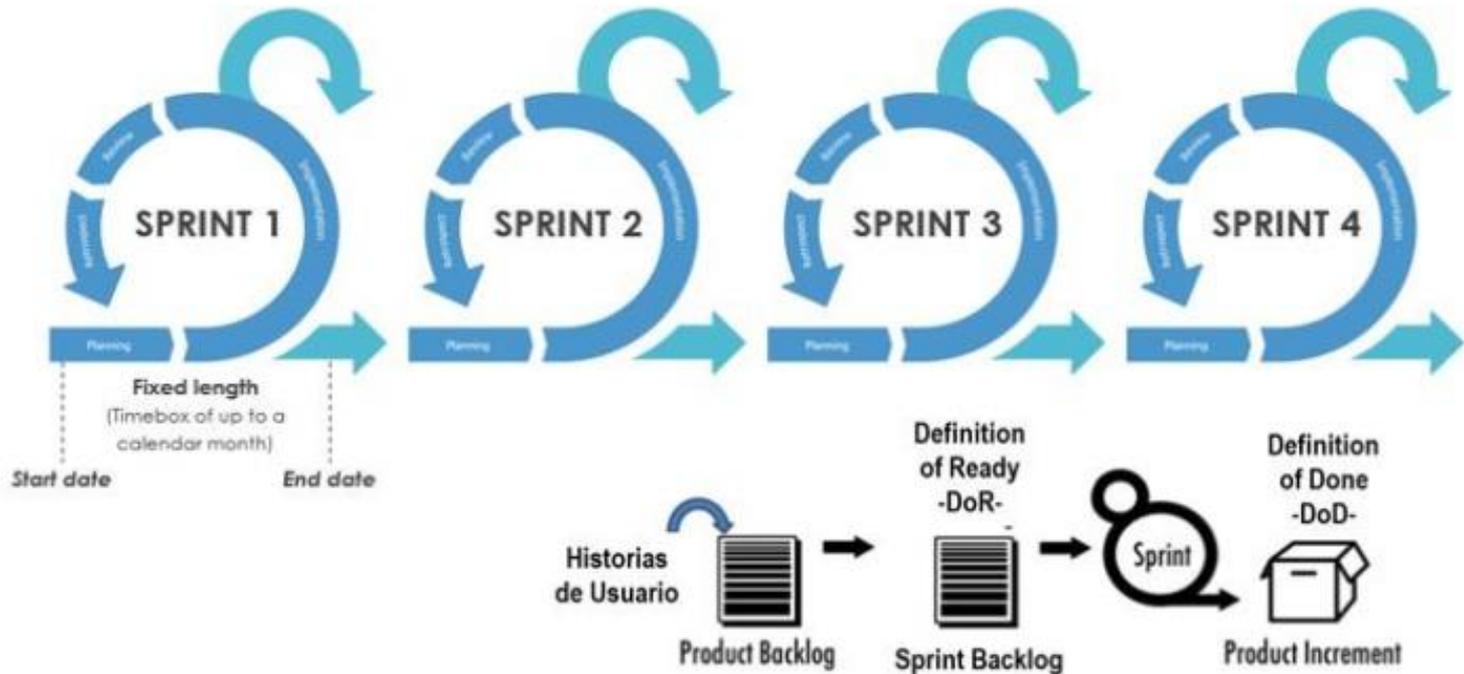


-Reunión para mostrar el incremento al ProductOwner

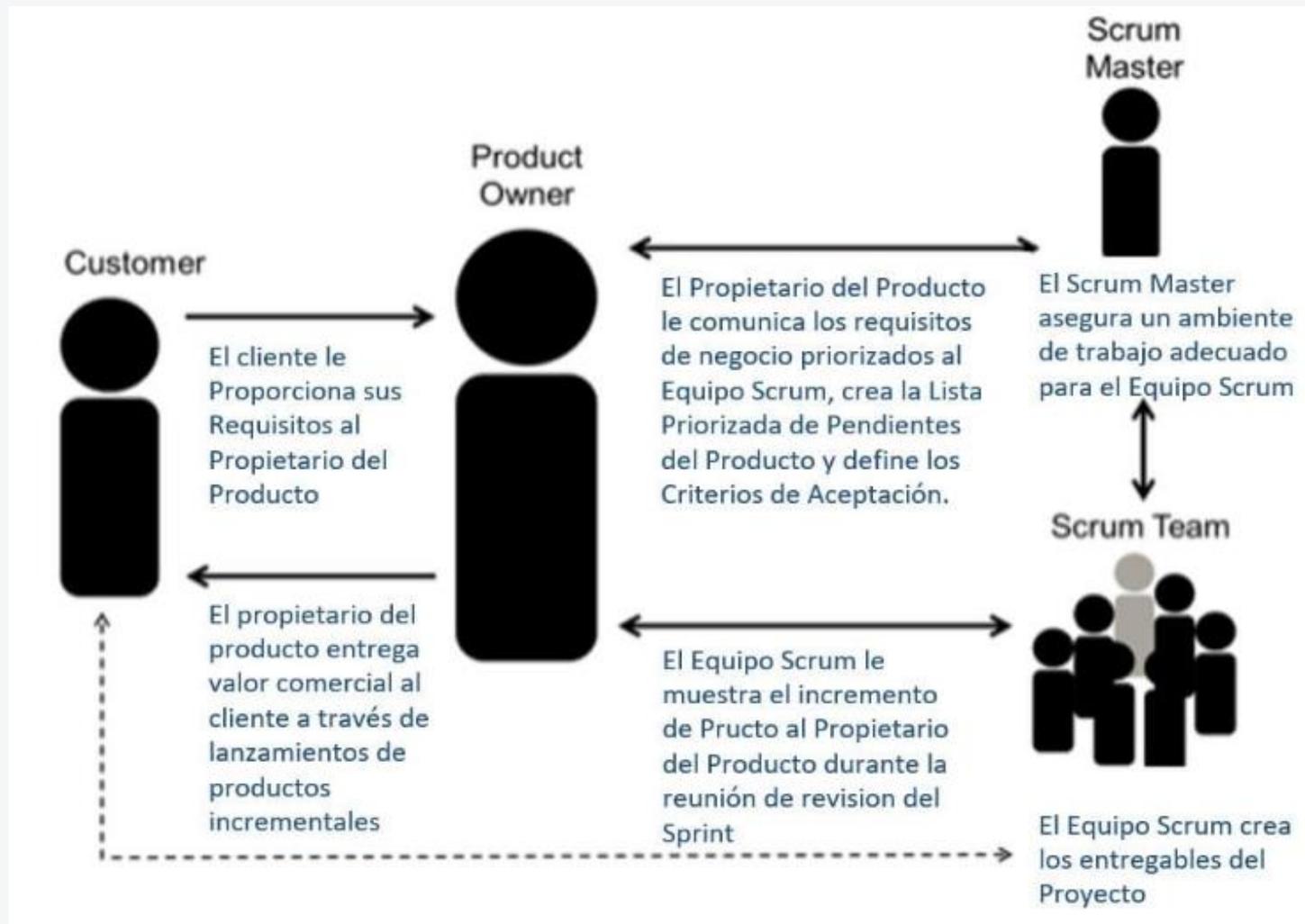
- Reajuste del la Pila de Producto en caso de ser necesario por nuevas características detectadas



# CADENA DE SPRINT



# INTERACCIÓN ENTRE ROLES DE SCRUM



# Recursos Audio Visuales

The image shows a screenshot of a YouTube channel page. At the top, there is a navigation bar with the YouTube logo, a search bar containing the word 'Buscar', and icons for notifications and a profile. Below the navigation bar is a blue banner for the channel. The banner features the text 'CONTÁCTAME PARA' in a light blue box, followed by two bullet points: '• Capacitaciones empresariales.' and '• Entrenarte para ser un Scrum Master.' To the right of the text is a photo of a woman with long dark hair, wearing a blue blazer. Further right, there are social media links for 'www.agilesporcrisruea.com', a WhatsApp number '+57 324 647 6262', and icons for Facebook, Instagram, and LinkedIn with the handle '@Agillesporcrisruea'. Below the banner is the channel's profile information, including a circular profile picture of a cartoon character, the channel name 'Ágil Es - Por Cris Rúa', the handle '@AgilEsPorCrisRuea', and statistics: '59.3 k suscriptores · 69 videos'. Below this is a greeting '¡Hola! >' and a link to 'agilesporcrisruea.com y 1 vínculo más'. At the bottom of the profile section is a 'Suscrito' button with a bell icon and a dropdown arrow.

<https://www.youtube.com/@AgilEsPorCrisRuea>

# Recursos Audio Visuales

## ¿Qué es Scrum?

<https://www.youtube.com/watch?v=nOlwF3HRrAY>

## Product Owner

<https://www.youtube.com/watch?v=X7gh07V6aKo&t=7s>

<https://www.youtube.com/watch?v=iAlZOaTphtM>

## ScrumMaster

<https://www.youtube.com/watch?v=cXinM8juOoc>

<https://www.youtube.com/watch?v=r2hU7MVIzxs>

## Historias de usuario

<https://www.youtube.com/watch?v=ky6wFiF5vMk>

<https://www.youtube.com/watch?v=Ueq786iZ30I>

## Sprint

[https://www.youtube.com/watch?v=02\\_WIm5UubU](https://www.youtube.com/watch?v=02_WIm5UubU)

<https://www.youtube.com/watch?v=FhiOeECwsS8&t=49s>

## Daily Scrum

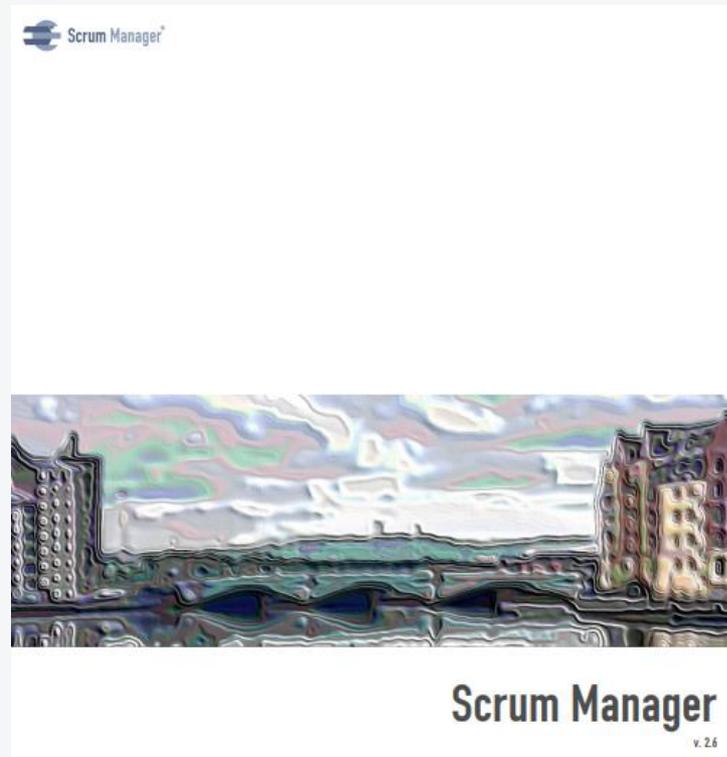
<https://www.youtube.com/watch?v=16loziutCZs>

## Retrospectiva

<https://www.youtube.com/watch?v=W6jZptSd3eM&t=160s>

# Recursos Gratuitos de Lectura

# Scrummanager



[https://www.scrummanager.com/files/sm\\_proyecto.pdf](https://www.scrummanager.com/files/sm_proyecto.pdf)

# Lecturas, videos y podcast

## Scrummanager

### Libros y audiovisuales

Los recursos de Scrum Manager son de uso libre para consulta, autoformación y actividades sin ánimo de lucro.  
[Creative Commons Atribución-NoComercial](#)  
Se pueden usar en actividades profesionales con ánimo de lucro adquiriendo una [licencia comercial](#).

#### Libros

- Scrum Master**  
Agilidad para equipos  
[Descargar](#)
- Scrum Level**  
Agilidad para empresas  
[Descargar](#)
- Historias de usuario**  
Gestión de requisitos ágil  
[Descargar](#)
- Contratos ágiles**  
Guía de referencia y negociación  
[Descargar](#)

### Podcast

Primera temporada completa:

- ¿Qué es Scrum? y estimación ágil | Scrum Manager Podcast #01**  
Diciembre 2023 - Scrum Manager Podcast  
[Descargar episodio](#)
- Cómo crear un backlog**  
Enero 2023 - Scrum Manager Podcast  
[Descargar episodio](#)
- Cómo se crean las historias de usuario**  
Febrero 2023 - Scrum Manager Podcast  
[Descargar episodio](#)
- Cómo se crean los contratos ágiles**  
Marzo 2023 - Scrum Manager Podcast  
[Descargar episodio](#)
- Eventos de Scrum**  
Abril 2023 - Scrum Manager Podcast  
[Descargar episodio](#)
- Agile Incentivos**  
Mayo 2023 - Scrum Manager Podcast  
[Descargar episodio](#)
- Definición de Done**  
Junio 2023 - Scrum Manager Podcast  
[Descargar episodio](#)
- SMART Goals**  
Julio 2023 - Scrum Manager Podcast  
[Descargar episodio](#)
- Cómo crear un backlog**  
Agosto 2023 - Scrum Manager Podcast  
[Descargar episodio](#)
- Contratos ágiles**  
Septiembre 2023 - Scrum Manager Podcast  
[Descargar episodio](#)
- Estimación ágil**  
Octubre 2023 - Scrum Manager Podcast  
[Descargar episodio](#)
- Ver los canchales**  
Noviembre 2023 - Scrum Manager Podcast  
[Descargar episodio](#)

### Herramientas para la agilidad organizacional

- Dinámica para análisis en equipo**  
[Descargar](#)  
[Información](#)
- Kit para evaluaciones presenciales**  
[Descargar kit](#)  
[Guía de uso](#)  
[Información](#)

### YouTube

Principal  
Shorts  
Suscripciones

Tú >

- Tu canal
- Historial
- Tus videos
- Ver más tarde

Scrum Manager  
Uncovering better ways

**Scrum Manager**  
@scrummanager · 1.58 K suscriptores · 37 videos  
Más información sobre este canal >

[Suscribirse](#)

<https://www.scrummanager.com/website/c/info/docs-media.php>

# Capacitación y certificación en SCRUM KLEER

The screenshot shows the KLEER website homepage. At the top left is the KLEER logo. The navigation menu includes 'Servicios', 'Cursos', 'Clientes', 'Publicamos', 'Somos', and 'ENGLISH'. Social media icons for Twitter, Facebook, Instagram, YouTube, and LinkedIn are on the right. The main content area features the heading 'Agilidad Organizacional' and a sub-heading 'Acompañamos a las organizaciones en la mejora sostenible de sus formas de trabajo y a adaptarse oportunamente a los cambios del entorno.' Below this is a blue button that says '¡Conoce más!'. To the right is a circular image of a meeting with sticky notes on a whiteboard. The page is decorated with colorful abstract shapes: a yellow bean-like shape at the top right, a red triangle on the left, and a blue circle at the bottom right containing an envelope icon.

**kleer** Servicios ▾ Cursos ▾ Clientes Publicamos ▾ Somos ENGLISH

Twitter Facebook Instagram YouTube LinkedIn

## Agilidad Organizacional

Acompañamos a las organizaciones en la mejora sostenible de sus formas de trabajo y a adaptarse oportunamente a los cambios del entorno.

[¡Conoce más!](#)

✉

<https://www.kleer.la/es/>

# CURSO GRATUITO CERTIFICADO KLEER



**kleer**  
ACADEMIA

## Academia Kleer Online\_

¡Aprende Agilidad con cursos  
online a tu propio ritmo!

[Súmate](#)

Login Registrarme

<https://academia.kleer.la/p/agilidad-y-scrum>



<https://www.youtube.com/watch?v=PILHc60egiQ>

# Kanban



# Diferencias con Scrum

- Menos restrictivo. No se utilizan iteraciones de duración fija. Puedes elegir el momento de hacer la planificación, la mejora de procesos, y la entrega. Puedes elegir hacer estas actividades de manera regular ( "la entrega todos los lunes") o bajo demanda ( "la entrega cada vez que tengamos algo útil para entregar").
- No se establecen roles.
- Limitación del WIP (Work In Progress, Trabajo en curso). Es decir se las tareas en curso no pueden ser mayor a una determinada cantidad.

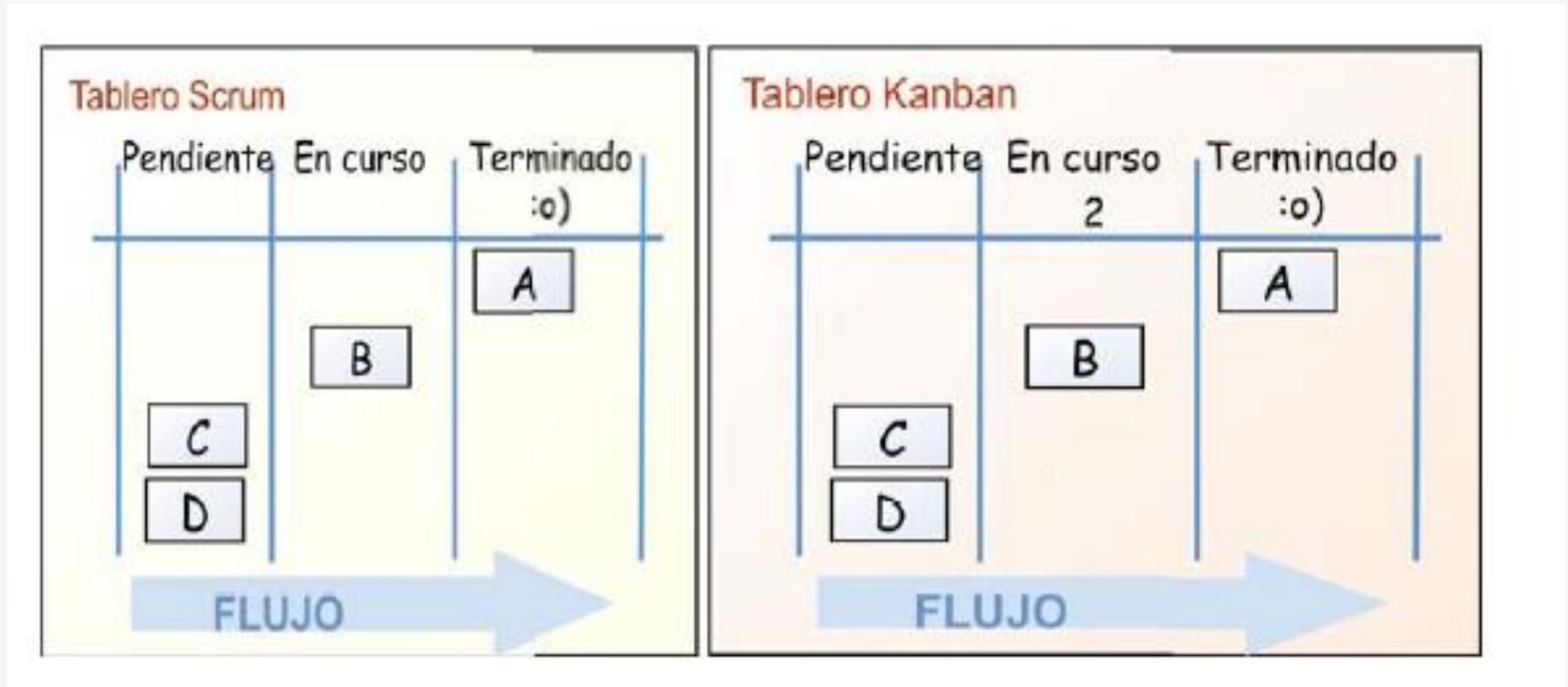
# Diferencias con Scrum

- Permite modificar las tareas de la pila en desarrollo, lo únicos que se debe tener en cuenta es el límite del WIP. En caso de que este se alcanzó se deberá cambiar una de las tareas desganada con anterioridad por la que se desea realizar actualmente.
- El tablero en Scrum se limpia al finalizar un Sprint, en Kanban puede ser en cualquier momento.
- El tablero con las tareas no se vinculan a un equipo específico de trabajo como es el caso de Scrum donde el tablero es del equipo que desarrolla el Sprint.

# Diferencias con Scrum

- No existen las reuniones diarias definidas en el marco kanban, pero muchos equipos las utilizan.

# WIP



¿cuál es la diferencia entre estas dos pizarras de ejemplo?  
Sí, el pequeño 2 en la columna central en la pizarra Kanban. Eso es todo. Ese 2 significa que "no puede haber más de 2 elementos en esta columna en un momento dado".

# Recursos Audio Visuales

The image shows a screenshot of a YouTube channel page. At the top, there is a navigation bar with the YouTube logo, a search bar containing the word 'Buscar', and several icons including a plus sign, a notification bell with '9+', and a profile icon. On the left side, there is a sidebar menu with options: 'Principal', 'Shorts', 'Suscripciones', 'Tú >', 'Tu canal', 'Historial', 'Tus videos', 'Ver más tarde', and 'Mostrar más'. The main content area features a blue banner with the text 'CONTÁCTAME PARA' and a list of services: '• Capacitaciones empresariales.' and '• Entrenarte para ser un Scrum Master.' The banner also includes the stylized text 'Ágil es Por Cris Rúa' with arrows pointing towards it, a photo of a woman, and contact information: 'www.agilesporcrisrúa.com', '+57 324 647 6262', and social media icons for Facebook, Instagram, and LinkedIn with the handle '@Agillesporcrisrúa'. Below the banner, the channel name 'Ágil Es - Por Cris Rúa' is displayed, along with the handle '@AgilEsPorCrisRúa', '59.3 k suscriptores', and '69 videos'. A greeting '¡Hola!' is followed by a link to 'agilesporcrisrúa.com y 1 vínculo más'. At the bottom, there is a 'Suscrito' button with a dropdown arrow.

<https://www.youtube.com/@AgilEsPorCrisRua>

# Recursos Audio Visuales

**¿Qué es Kanban?**

<https://www.youtube.com/watch?v=DkeZYBRVfUQ>

**Diferencias entre Scrum y Kanban**

[https://www.youtube.com/watch?v=Kxz0\\_rDmRho](https://www.youtube.com/watch?v=Kxz0_rDmRho)

# Scrum Caso Práctico



Fuente:

<https://uvadoc.uva.es/bitstream/handle/10324/1495/TFGB.117.pdf?sequence=1>

# Scrum Caso Práctico

## BackLog de Historias de Usuario

Prior	Proyecto	Nombre	Notas	Cómo probarlo
1	Gestión Incidencias	Crear tarea para que procese el excel y lo transforme en incidencias en BBDD.	Script que procese los excel subidos por los usuarios y los transforme en incidencias.	Comprobar que una vez que el usuario sube un Excel con incidencias, se procesan y se crean registros de incidencias en la web.
2	Gestión Incidencias	Crear back-end para subir Excel con incidencias	Almacenar el fichero Excel en BBDD.	
3	Gestión Incidencias	Crear front-end para subir Excel con incidencias.	Nueva opción en el menú y nueva ventana para que el usuario descargue la plantilla Excel y una vez cubierta la envíe al sistema.	Comprobar guía de estilo del cliente. Test de accesibilidad.
4	LDAP	Separar vista para Administradores / No Administradores.	Permisos según perfiles:  - Administrador: Tendrá acceso a los informes de todos los distribuidores - No administrador: Solo acceso a sus propios informes	Comprobar que un usuario Administrador puede adoptar el rol de un distribuidor y consultar sus informes. Comprobar que un usuario no administrador solo tiene acceso a sus propios informes.
5	LDAP	Configuración de perfiles	Es necesario tener dos perfiles diferentes a la hora de acceder a la Web:  Los perfiles a configurar son: - Administrador. - No administrador.	
6	LDAP	Instalar Servidor LDAP	Se requiere implantar LDAP en el acceso a la Web de Comisiones.	
7	Informes	Añadir un nuevo concepto de comisión en los informes	Se necesita el que concepto por el que se comisionan las portabilidades aparezca desglosado en una nueva columna de los informes de la Web de Comisiones. El nombre de la columna será "Comisión Portabilidad" y se acumulará la cantidad comisionada al distribuidor en concepto de portabilidad durante el ciclo de comisiones.	Una vez calculada la comisión por portabilidad, comprobar que se acumula en la columna de "Comisión Portabilidad".

# Scrum Caso Práctico

## Historia de Usuario

Backlog Item #4 Proyecto: LDAP	
<b>Separar vista para Administradores / No Administradores.</b>	
Notas	Prioridad
Permisos según perfiles: - Administrador: Tendrá acceso a los informes de todos los distribuidores - No administrador: Solo acceso a sus propios informes	4
Como probarlo	Estimación
Comprobar que un usuario Administrador puede adoptar el rol de un distribuidor y consultar sus informes. Comprobar que un usuario no administrador solo tiene acceso a sus propios informes.	5

Tareas asociadas a esta historia de usuario:

Perfil administrador: opción para simular distribuidor.  
3

Perfil no administrador: Que no se vea la opción de simular distribuidor.  
2

# Scrum Caso Práctico

## Sprint

### Reuniones diarias

#### Inicio del Sprint

Los miembros del equipo se reúnen en frente al tablero de Scrum. En la primera reunión nadie tiene una tarea asignada, por tanto, no es necesario responder a las preguntas: ¿Qué has hecho ayer? ¿Qué voy a hacer hoy? ¿Algún impedimento?. Cada desarrollador coge una tarea del tablero y comienza con la implementación.

Tareas que ha cogido cada desarrollador:

Story	Pending	Started	Done
11 pts started 0. Añadir un nuevo concepto de comisión en los in...		5.0 Desarrollador 1 Crear modelo para el concepto de "PORTABILIDAD". 3.0 Desarrollador 2 Procedimiento SQL que genere el informe. 1.0 Desarrollador 3 FRONT-END: Nueva columna en los informes: "Comisión Portabilidad" 2.0 Desarrollador 4 BACK-END: Recuperar comisión de BBDD.	
3 pts started 0. Instalar Servidor LDAP		3.0 Desarrollador 5 Instalar LDAP en: Desarrollo, Pruebas y Producción.	
2 pts pending	1.0		

#### Día 1:

Preguntas que tiene que responder cada uno de los desarrolladores: ¿Qué has hecho ayer? ¿Qué voy a hacer hoy? ¿Algún impedimento?

**Desarrollador 1:** Ha estado trabajando en la tarea: "Crear modelo para el concepto de PORTABILIDAD". Durante el día de hoy continuará con esta tarea. No ha encontrado ningún impedimento y estima que aún le quedan 4 puntos de trabajo, por tanto, ha avanzado un punto de trabajo.

**Desarrollador 2:** Ha estado trabajando en la tarea: "Procedimiento SQL que genere informe". Durante el día de hoy continuará con esta tarea. No hay encontrado ningún impedimento. Estima que le quedan 2 puntos de trabajo en esta tarea.

**Desarrollador 3:** Ha estado trabajando en la tarea: "FRONT-END: Nueva columna en los informes. Comisión Portabilidad" y ya está terminada, por tanto, actualiza el ticket de la tarea a la columna "¡Hecho!". Coge la siguiente tarea disponible en el tablero: "Configurar perfil: administrador".

**Desarrollador 4:** Ha estado trabajando en la tarea: "BACK-END: Recuperar comisión de BBDD". Durante el día de hoy continuará con esta tarea. No hay encontrado ningún impedimento. Estima que le quedan 1 puntos de trabajo en esta tarea.

**Desarrollador 5:** Ha estado trabajando en la tarea: "Instalar LDAP en: Desarrollo, Pruebas y Producción", continuará trabajando en esta tarea y la reestima en 2.5 puntos de trabajo.

# Scrum Caso Práctico

## Sprint

### Demostración

Una vez preparado el guión para la reunión, se convoca al cliente a una reunión en la que ha visto las nuevas funcionalidades que se han añadido al aplicativo.

### Reunión retrospectiva

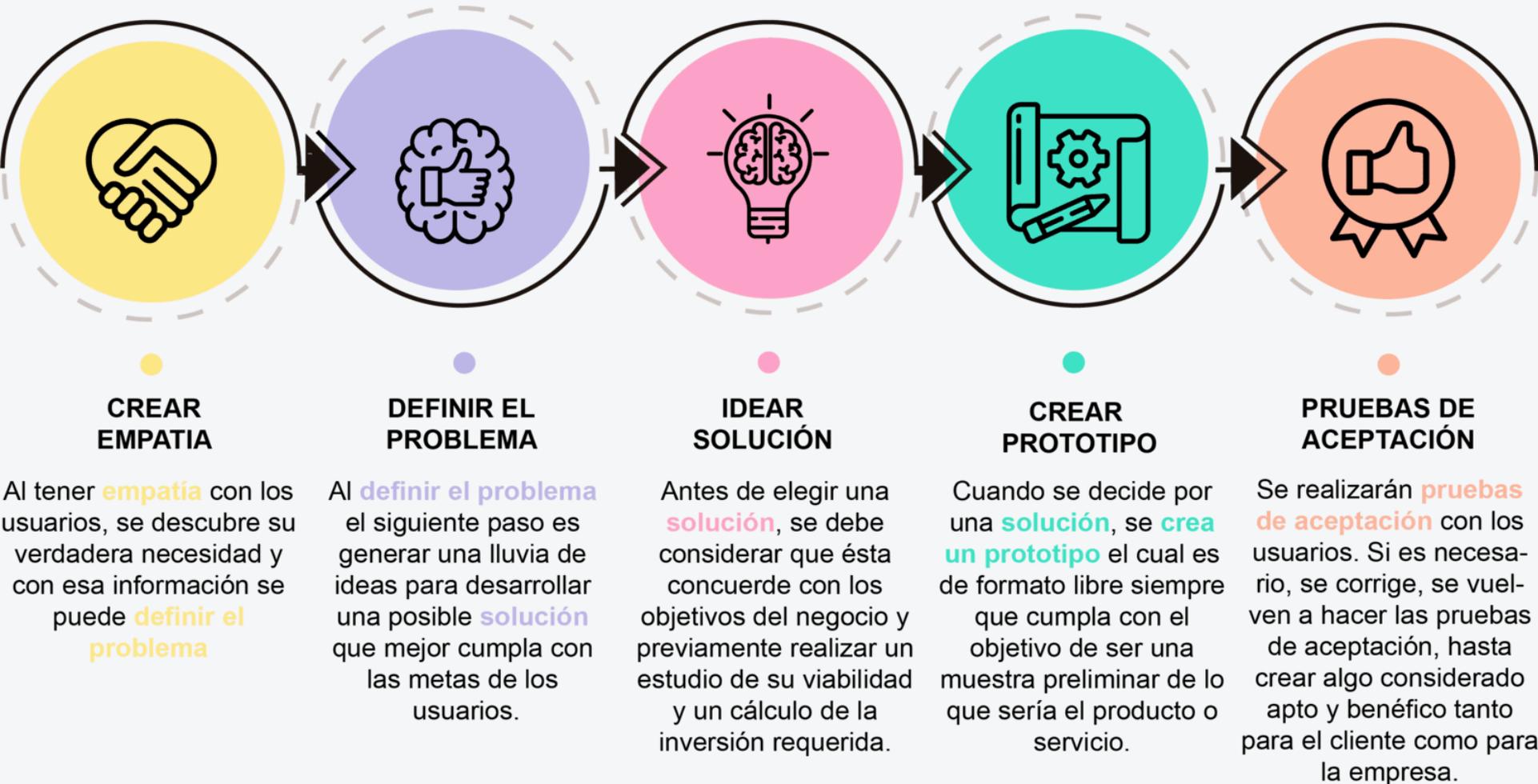
Una vez finalizado el Sprint se realiza la reunión retrospectiva, el objetivo de la reunión es reflexionar para mejorar el proceso: como está trabajando el equipo junto. prácticas o herramientas que se están utilizando.

Esta reunión está restringida a los miembros del equipo.

Las conclusiones se encuentran resumidas a continuación:

Positivo	Mejorable	Soluciones	Ideas
Mejora tareas QA Estimaciones cumplidas			Crear el guión de la demo como otro documento del sprint
Requisitos claros para el equipo Preparación Demo Guía de Tests para los usuarios Mejorar comunicación (QA-Equipo de desarrollo)	Mejorar la visión de los requisitos para el "cliente"  Intentar que los usuarios no se ciñan a la guía	Dar más visibilidad de los requerimientos incluyendo a los usuarios finales de los mismos (no solo al owner como jefe)	Conocer infraestructura Aportar/plantear posibles desarrollos que puedan interesar a negocio
	Mejorar la toma de requisitos durante el Sprint para el siguiente Sprint	Tener en cuenta las fechas de SMT con las planificaciones de sprint  Planificar la demo para 2 días después del fin de desarrollo de sprint para permitir tiempo de preparación, resolución de posibles problemas de última hora	
	No se ha llegado a la fecha del Sprint Entomos / servicios preparados Retraso de la demo para mostrar toda la funcionalidad	Invertir tiempo en repasar y montar si es necesario los entomos de desarrollo, QA, etc.  Se debería haber hecho la demo con las funcionalidades desarrolladas Resolver impedimentos en cuanto aparecen	

# DESIGN THINKING





[https://www.youtube.com/watch?v=\\_DkP38ebKIM](https://www.youtube.com/watch?v=_DkP38ebKIM)

[https://www.youtube.com/watch?v=\\_DkP38ebKIM](https://www.youtube.com/watch?v=_DkP38ebKIM)