



# Ingeniería de Software I – Clase 2 – 2025



# Temas

---

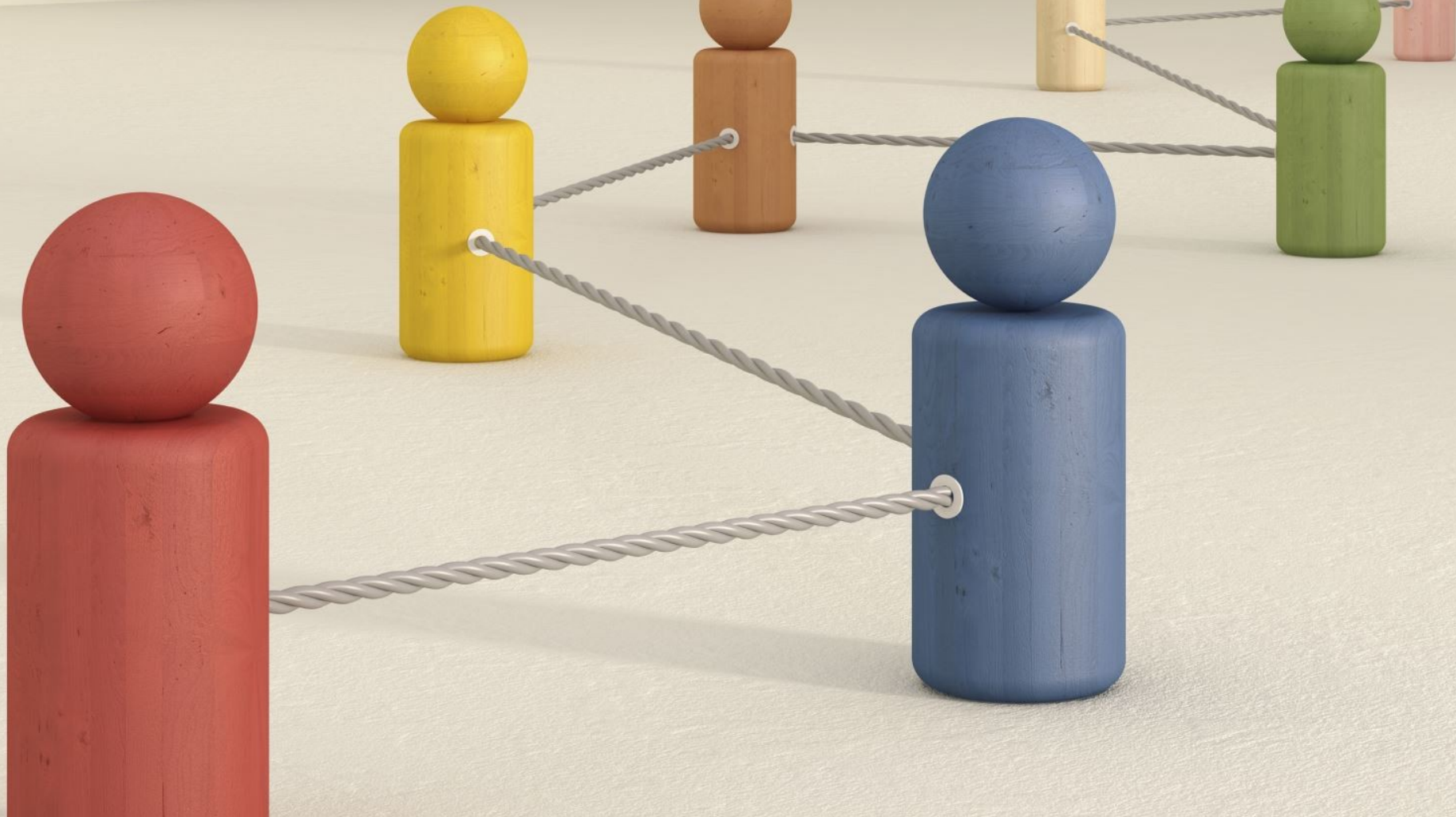
- Técnicas de elicitación dinámicas (continuación)
- Técnicas de elicitación estáticas
- Especificación de requerimientos de software

# Técnicas de elicitación (Continuación de la clase 1)

---

Habíamos visto que hay dos métodos los interactivos y los discretos

Continuamos con métodos interactivos de elicitación y luego veremos los discretos



# Métodos interactivos de elicitación de requerimientos Continuación

# Planeación Conjunta de Requerimientos (JRP)

---

Proceso mediante el cual se conducen reuniones de grupo altamente estructurados con el propósito de analizar problemas y definir requerimientos

- » Requiere de extenso entrenamiento
- » Reduce el tiempo de exploración de requisitos
- » Amplia participación de los integrantes
- » Se trabaja sobre lo que se va generando
- » Algunos bibliografía la menciona como JAD (Joint Application Design)



# Planeación Conjunta de Requerimientos (JRP)

---

## »Ventajas

Ahorro de tiempo

Usuarios involucrados

Desarrollos creativos

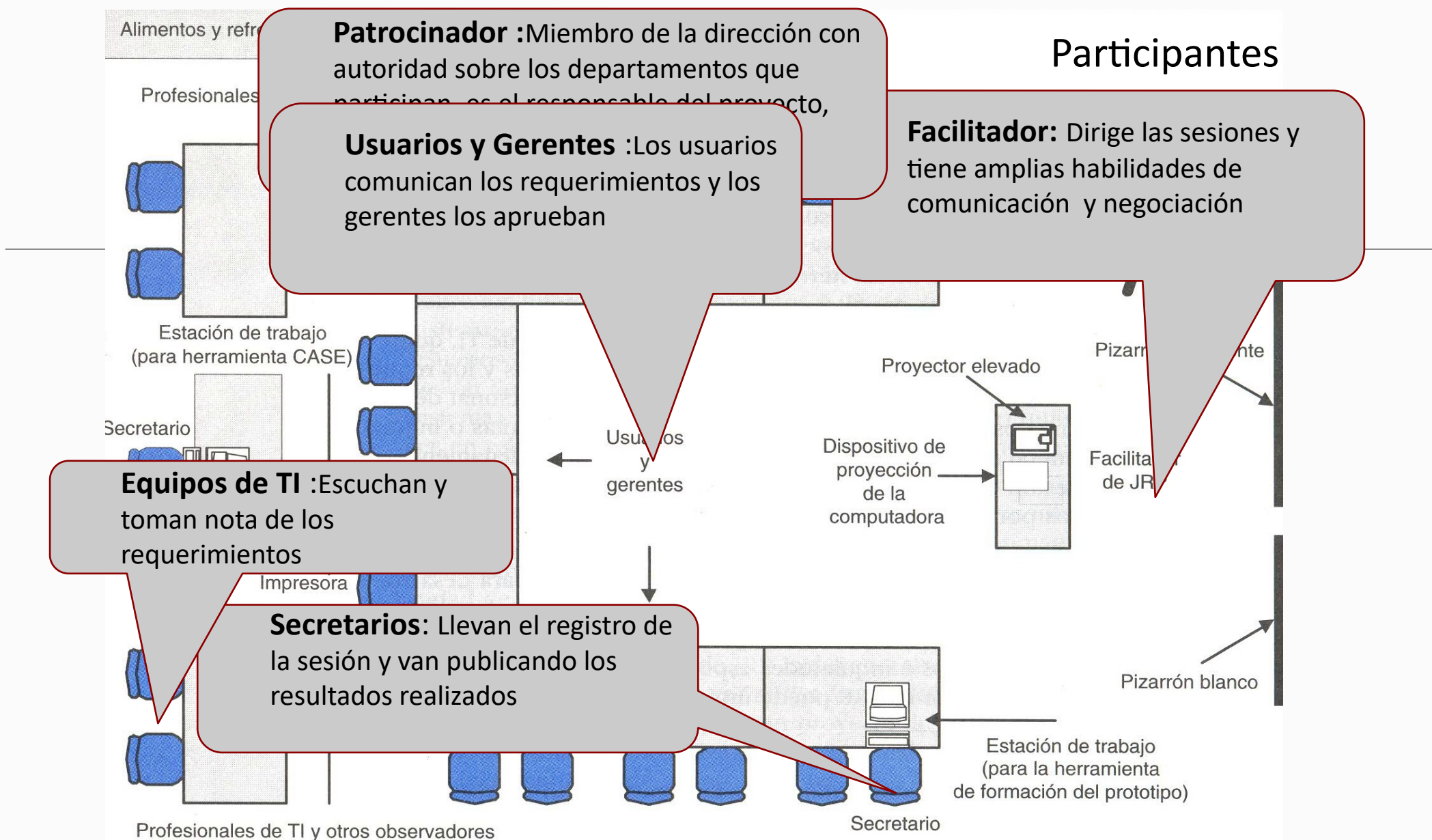
## »Desventajas

Es difícil organizar los horarios de los involucrados

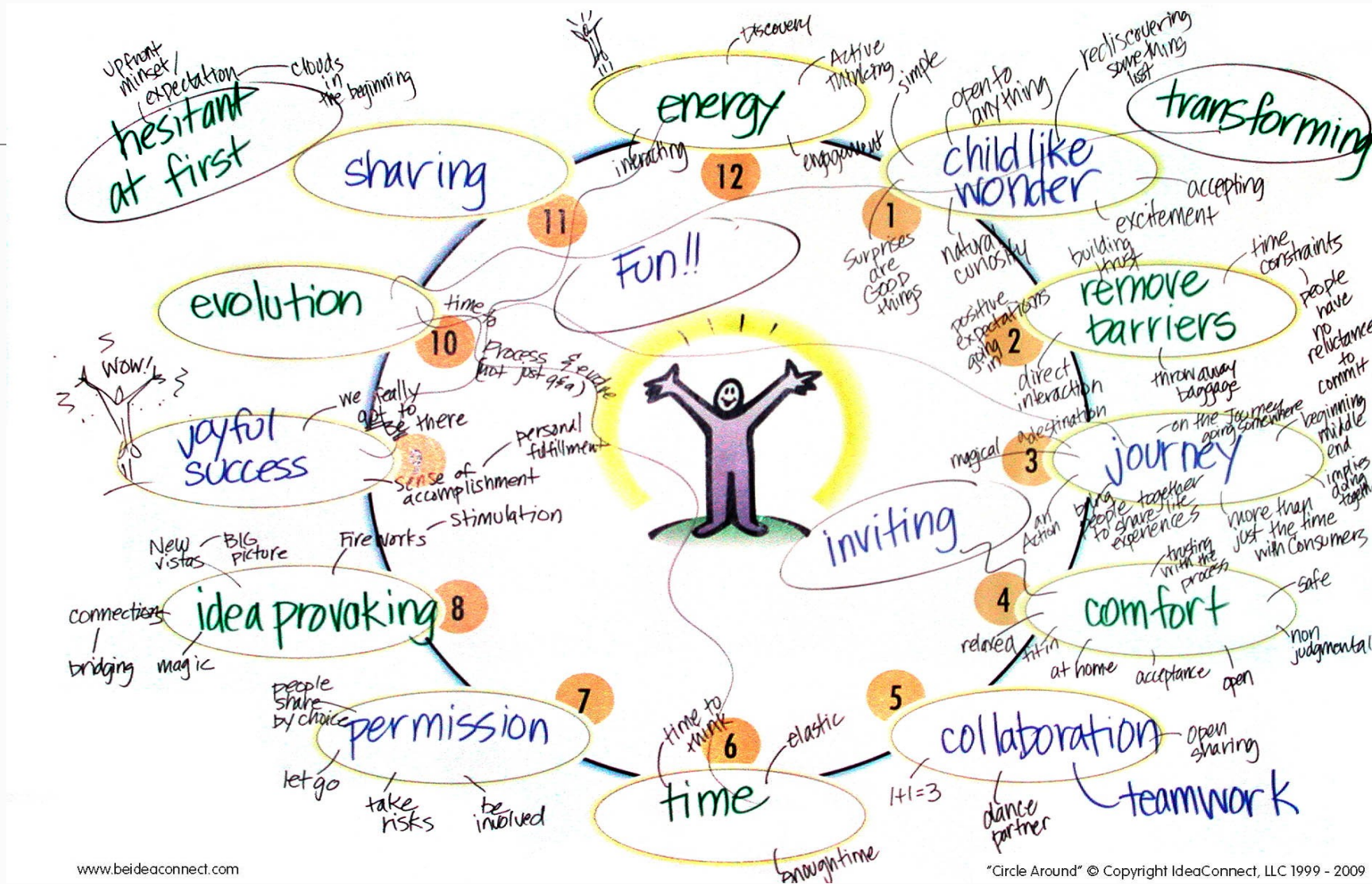
Es complejo encontrar un grupo de participantes integrados y organizados

6





# Lluvia De Ideas (Brainstorming)





# Lluvia De Ideas (Brainstorming)

---

- » Técnica para generar ideas al alentar a los participantes para que ofrezcan tantas ideas como sea posible en un corto tiempo sin ningún análisis hasta que se hayan agotado las ideas.
- » Se promueve el desarrollo de ideas creativas para obtener soluciones.
- » Se realizan reuniones del equipo involucrado en la resolución del problema, conducidas por un director.

# Lluvia De Ideas (Brainstorming)

---

» Los principios en que se basa esta técnica son:

*Cuantas más ideas se sugieren, mejores resultados se conseguirán.*

*La producción de ideas en grupos puede ser más efectiva que la individual.*

*Las ideas de una persona pueden hacer que aparezcan otras por “contagio”.*

*A veces las mejores ideas aparecen tarde.*

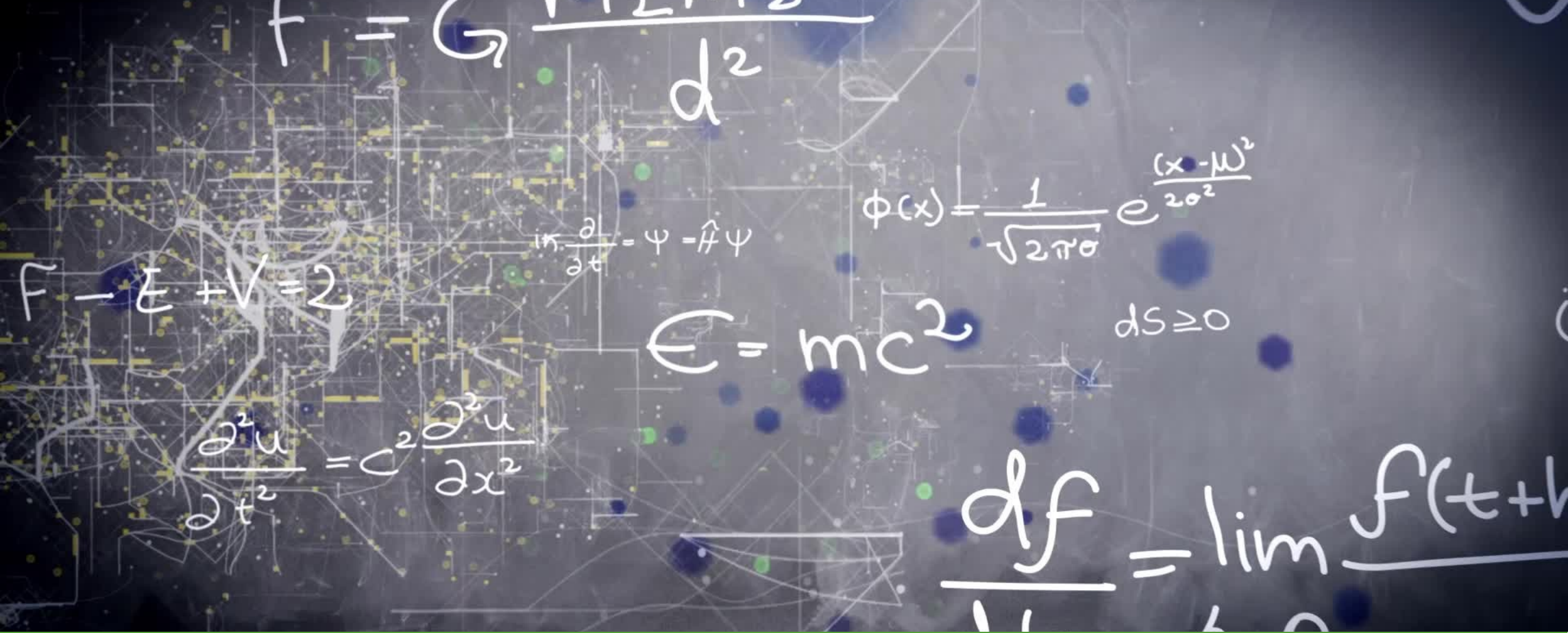
*Es mejor elegir sobre una variedad de soluciones.*

# Lluvia De Ideas (Brainstorming)

---

- » Incluye una serie de fases de aplicación:  
Descubrir hechos, Producir ideas, Descubrir soluciones
- » Clave para resolver la falta de consenso entre usuarios
- » Es útil combinarlo con la toma de decisiones
- » Ayuda a entender el dominio del problema
- » Encara la dificultad del usuario para transmitir
- » Ayuda a entender: al usuario y al analista





# Métodos discretos de elicitación de requerimientos

# Métodos discretos

---

Los métodos discretos son menos perturbadores que otras formas de averiguar los requerimientos

Se consideran insuficientes para recopilar información cuando se utilizan por sí solos, por lo que deben utilizarse junto con uno o varios de los métodos.

Utilizar diferentes métodos para acercarse a la organización es una práctica inteligente mediante la cual podrá formarse un panorama más completo de los requerimientos

- » Muestreo de la documentación, los formularios y los datos existentes
- » Investigación y visitas al sitio
- » Observación del ambiente de trabajo





# Muestreo de la documentación, los formularios y los datos existentes

---

» Recolección de hechos a partir de la documentación existente.

¿Qué tipo de documentos pueden enseñar algo acerca del sistema?

Organigrama (identificar el propietario, usuarios claves).

Memos, notas internas, minutas, registros contables.

Solicitudes de proyectos de sistemas de información anteriores.

Permiten conocer el historial que origina el proyecto.



# Muestreo de la documentación, los formularios y los datos existentes

---

»Recolección de hechos a partir de la documentación existente.

Documentos que describen la funcionalidad del negocio que está siendo analizada.

- Declaración de la misión y plan estratégico de la organización.

- Objetivos formales del departamento en cuestión.

- Políticas, restricciones, procedimientos operativos.

- Bases de Datos.

- Sistemas en funcionamiento.

Documentación de sistemas anteriores.

- Diagramas.

- Diccionario o Repositorios de proyecto.

- Documentos de diseño.

- Manuales de operación y/o entrenamiento.

# Investigación y visitas al sitio

---

- ✓ Investigar el dominio.
- ✓ Patrones de soluciones (mismo problema en otra organización).
- ✓ Revistas especializadas.
- ✓ Buscar problemas similares en internet.
- ✓ Consultar otras organizaciones.



# Observación del ambiente de trabajo

- » El analista se convierte en observador de las personas y actividades con el objeto de aprender acerca del sistema.
- » Lineamientos de la observación:
  - Determinar quién y cuándo será observado.
  - Obtener el permiso de la persona y explicar el porqué será observado.
  - Mantener bajo perfil.
  - Tomar nota de lo observado.
  - Revisar las notas con la persona apropiada.
  - No interrumpir a la persona en su trabajo.



# Observación del ambiente de trabajo

---

## » Ventajas

Datos confiables

El analista puede ver exactamente lo que se hace (tareas difíciles de explicar con palabras).

Análisis de disposiciones físicas, tránsito, iluminación, ruido.

Económica en comparación con otras técnicas.

## » Desventajas

La gente se siente incómoda siendo observada.

Algunas actividades del sistema pueden ser realizadas en horarios incómodos.

Las tareas están sujetas a interrupciones.

Tener en cuenta que la persona observada puede estar realizando las tareas de la forma “correcta” y no como lo hace habitualmente.





# Técnicas de Especificación de Requerimientos de software

# Especificación de Requerimientos

---

## Objetivos

- Permitir que *los desarrolladores expliquen cómo han entendido lo que el cliente pretende del sistema*
- Indicar a *los diseñadores qué funcionalidad y características va a tener el sistema resultante*
- Indicar al *equipo de pruebas qué demostraciones llevar a cabo para convencer al cliente de que el sistema que se le entrega es lo que había pedido.*

# Especificación de Requerimientos

## Propiedades de los Requerimientos

- **Necesario:** Su omisión provoca una deficiencia.
- **Conciso:** Fácil de leer y entender
- **Completo:** No necesita ampliarse
- **Consistente:** No contradictorio con otro
- **No ambiguo:** Tiene una sola implementación
- **Verificable:** Puede testearse a través de inspecciones, pruebas, etc.



# Especificación de Requerimientos

---

- Correcta
- No ambigua
- Completa
- Verificable
- Consistente
- Comprensible por los consumidores
- Modificable
- Rastreable
- Independiente del diseño
- Anotada
- Concisa
- Organizada
- Utilizable en operación y mantenimiento

# Especificación de Requerimientos

---

- Documento de definición de requerimientos  
Listado completo de todas las cosas que el cliente espera que haga el sistema propuesto
- Documento de especificación de requerimientos  
Definición en términos técnicos
- Documento de especificación de requerimientos de Software IEEE Std. 830-1998 (SRS)  
Objetivo:  
Brindar una colección de buenas prácticas para escribir especificaciones de requerimientos de software (SRS).  
Se describen los contenidos y las cualidades de una buena especificación de requerimientos.



# Especificación de Requerimientos

---

- Aspectos básicos de una especificación de requerimientos

## Funcionalidad

¿Qué debe hacer el software?

## Interfaces Externas

¿Cómo interactuará el software con el medio externo (gente, hardware, otro software)?

## Rendimiento

Velocidad, disponibilidad, tiempo de respuesta, etc.

## Atributos

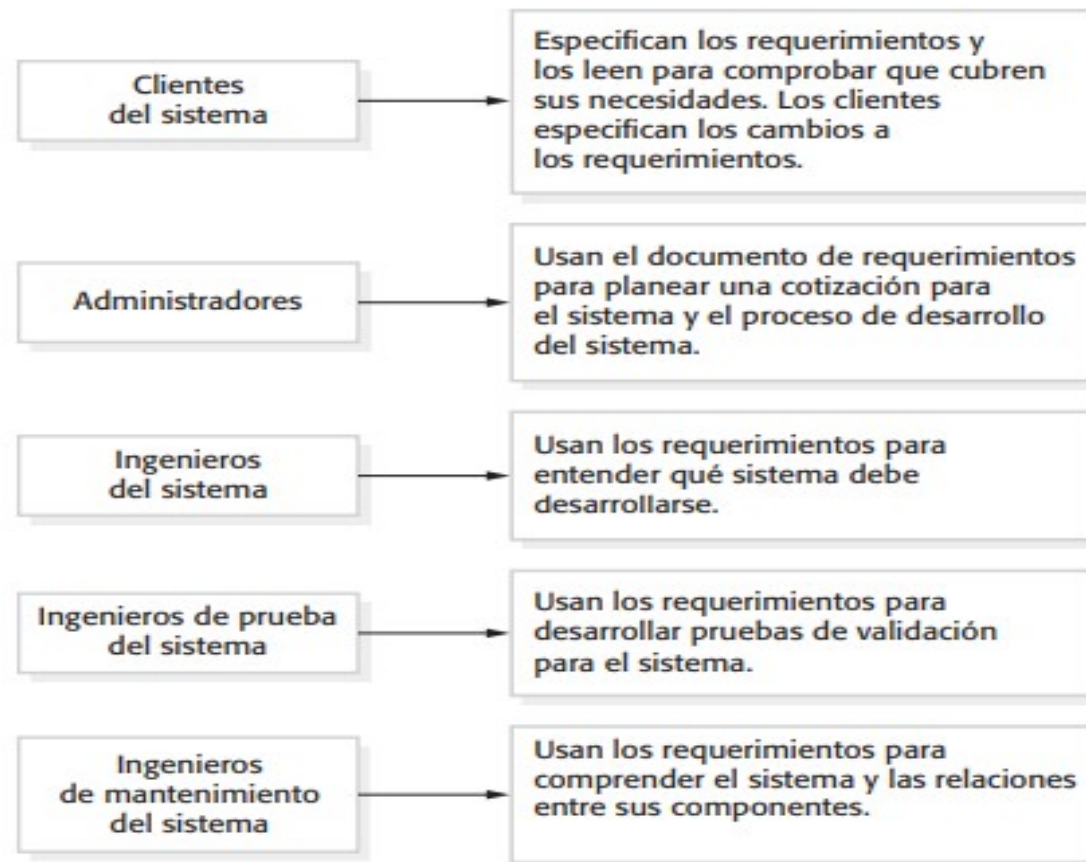
Portabilidad, seguridad, mantenibilidad, eficiencia

## Restricciones de Diseño

Estándares requeridos, lenguaje, límite de recursos, etc.

# Especificación de Requerimientos

- Usuarios de un documento de requerimientos



# Técnicas de Especificación de Requerimientos

---

## ■ Estáticas

Se describe el sistema a través de las *entidades u objetos, sus atributos y sus relaciones con otros. No describe cómo las relaciones cambian con el tiempo.*

Cuando el tiempo no es un factor mayor en la operación del sistema, es una descripción útil y adecuada.

Ejemplos: Referencia indirecta, Relaciones de recurrencia, Definición axiomática, Expresiones regulares, Abstracciones de datos, entre otras.

# Técnicas de Especificación de Requerimientos

---

## ■ Dinámicas

Se considera un sistema en función de los cambios que ocurren a lo largo del tiempo.

Se considera que el sistema está en un estado particular hasta que un estímulo lo obliga a cambiar su estado.

Ejemplos: Tablas de decisión, Diagramas de transición de estados, Tablas de transición de estados, Diagramas de persianas, Diagramas de transición extendidos, Redes de Petri, entre otras.

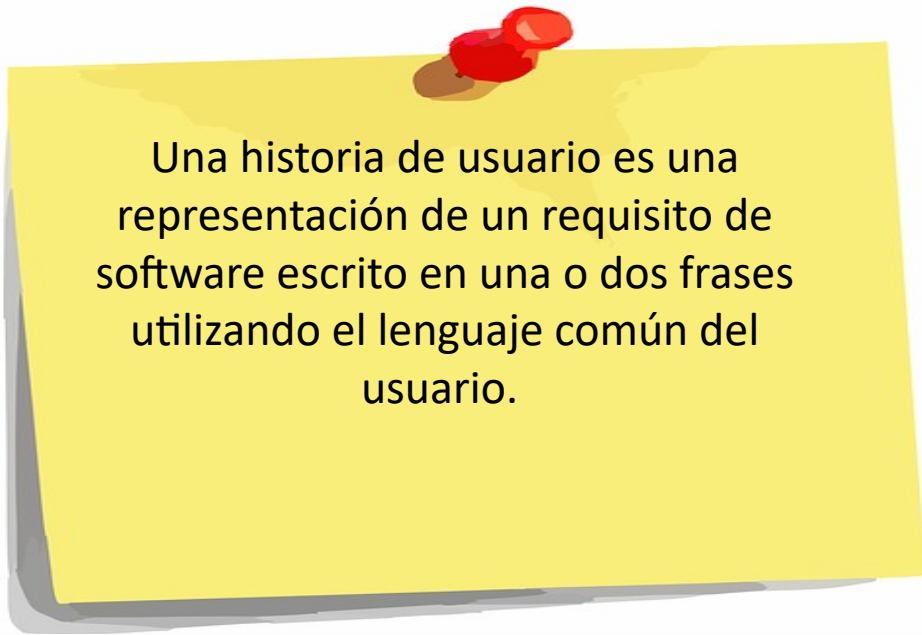
# Técnicas de Especificación de Requerimientos

## HISTORIAS DE USUARIO



# Historias de usuario - Definición

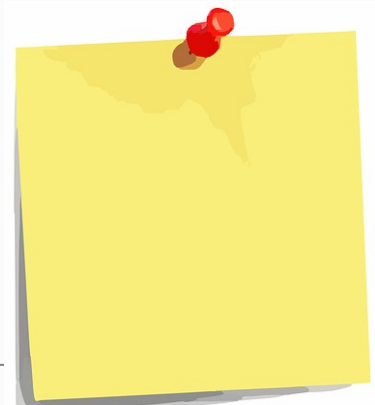
---



Una historia de usuario es una representación de un requisito de software escrito en una o dos frases utilizando el lenguaje común del usuario.

- Una historia de usuario es una descripción corta y simple de un requerimiento de un sistema, que se escribe en lenguaje común del usuario y desde su perspectiva.
- Son utilizadas en las metodologías de desarrollo ágiles (Ejemplo: XP, SCRUM) para la especificación de requerimientos  
Acompañadas de las discusiones con los usuarios y las pruebas de validación

# Historias de usuario – Conceptos



**Debe ser limitada**, ésta debería poder escribirse sobre una nota adhesiva pequeña.

Son **una forma rápida de administrar los requisitos** de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos.

Permiten **responder rápidamente a los requisitos cambiantes**.

Al momento de implementar las historias, los desarrolladores deben tener la posibilidad de **discutirlas con los clientes**.

Generalmente se espera que la **estimación de tiempo** de cada historia de usuario se sitúe entre unas **10 horas y un par de semanas**.

Estimaciones **mayores a dos semanas** son indicativo de que la historia es muy compleja y debe ser **dividida en varias historias**.

# Historias de usuario – Forma de redactarlas

---

Si bien el estilo puede ser libre, la historia de usuario debe responder a tres preguntas:

¿Quién se beneficia?

¿Qué se quiere?

¿Cuál es el beneficio?

Esquema:

Como (**rol**) quiero (**algo**) para (**beneficio**).

Ejemplos:

- Como **usuario registrado** quiero loguearme para **empezar a utilizar la aplicación**.
- Como **secretaria** quiero imprimir el listado de turnos asignados en una fecha determinada para **guardar la información de los mismos**.

# Historias de usuario – Forma de redactarlas

---

Como **Cliente** quiero **suscribirme por medio del sitio web** para **obtener un nuevo plan de T.V. por cable**

Como **Vendedor** quiero **registrar los productos y cantidades que me solicita un cliente** para **crear un pedido de venta**.

Como **Analista de compras** quiero **que el sistema notifique vía correo electrónico a los proveedores** para **avisar que se ha enviado una cotización de licitación**

# Historias de usuario – Características

---

**Independientes unas de otras:** De ser necesario, combinar las historias dependientes o buscar otra forma de dividir las historias de manera que resulten independientes.

**Negociables:** La historia en sí misma no es lo suficientemente explícita como para considerarse un contrato, la discusión con los usuarios debe permitir esclarecer su alcance y éste debe dejarse explícito bajo la forma de pruebas de validación.

**Valoradas por los clientes o usuarios:** Los intereses de los clientes y de los usuarios no siempre coinciden, pero en todo caso, cada historia debe ser importante para alguno de ellos más que para el desarrollador.

# Historias de usuario – Características

---

**Estimables:** Un resultado de la discusión de una historia de usuario es la estimación del tiempo que tomará completarla. Esto permite estimar el tiempo total del proyecto.

**Pequeñas:** Las historias muy largas son difíciles de estimar e imponen restricciones sobre la planificación de un desarrollo iterativo. Generalmente se recomienda la consolidación de historias muy cortas en una sola historia.

**Verificables:** Las historias de usuario cubren requerimientos funcionales, por lo que generalmente son verificables. Cuando sea posible, la verificación debe automatizarse, de manera que pueda ser verificada en cada entrega del proyecto.



# Historias de usuario - Criterios de aceptación

---

- Un criterio de aceptación es el criterio por el cual se define si una historia de usuario fue desarrollada según la expectativa del Product Manager/Owner (como representante de los criterios del cliente) y se si puede dar como hecha.
- Deben ser definidos durante la etapa inicial antes de la codificación, acompañan a la historia de usuario, porque complementan la historia de usuario y ayudan al equipo de desarrollo a entender mejor cómo se espera que el producto se comporte.

# Historias de usuario - Criterios de aceptación

---

- Los criterios de aceptación son utilizados para expresar el resultado de las conversaciones del cliente con el desarrollador. El cliente debería ser quien las escriba más que el desarrollador.
- Representan el inicio de la definición del cómo. No están diseñados para ser tan detallados como una especificación de diseño tradicional.
- Si una historia de usuario tiene más de 4 criterios de aceptación, debe evaluarse subdividir la historia.
- Puede añadirse un número de escenario para identificar al criterio, asociado a la historia de usuario en cuestión.

# Historias de usuario – Plantilla

Reverso

- **CRITERIOS DE ACEPTACIÓN:**
  - **Escenario 1:** título del criterio.  
Dado <un contexto inicial>,  
Cuando <ocurre un evento>,  
Entonces <garantiza uno o más resultados>
  - **Escenario 2:** título del criterio.  
Dado <un contexto inicial>,  
Cuando <ocurre un evento>,  
Entonces <garantiza uno o más resultados>
  - ....
  - **Escenario N:** título del criterio.  
Dado <un contexto inicial>,  
Cuando <ocurre un evento>,  
Entonces <garantiza uno o más resultados>

# Historias de Usuario

**Id:** Matricular persona

**Título:** Como *persona* quiero matricularme al instituto para poder hacer los cursos

**Reglas de Negocio:**

- Un DNI no puede estar registrado dos veces con diferentes matrículas
- El pago debe realizarse con tarjeta de crédito

# Historias de Usuario

Reverso

Criterios de Aceptación (Matricular persona):

## Escenario 1: Matriculación exitosa

**Dado** que el DNI 22.222.222 no se encuentra matriculado y las condiciones son las adecuadas para un pago exitoso.

**Cuando** la persona ingresa: "Juan Perez", DNI 22.222.222, dirección 7 #123 y presiona "Matricularse"

**Entonces** el sistema redirige al usuario al pago de matrícula con tarjeta de crédito, espera respuesta, matricula a la persona y genera su número de matrícula.

## Escenario 2: Matriculación fallida por matriculado existente

**Dado** que el DNI 12.123.123 se encuentra matriculado

**Cuando** la persona ingresa: "Ana Díaz", DNI 12.123.123 y dirección 51 #1321 y presiona "Matricularse"

**Entonces** el sistema informa que la persona ya se encuentra matriculada

## Escenario 3: Matriculación fallida por error en pago

**Dado** que el DNI 22.222.222 no se encuentra matriculado y las condiciones no son las adecuadas para un pago exitoso.

**Cuando** la persona ingresa: "Juan Perez", DNI 22.222.222, dirección 7 #123 y presiona "Matricularse"

**Entonces** el sistema redirige al usuario al pago de matrícula con tarjeta de crédito, espera respuesta e informa que el pago no ha sido correcto por lo que no se pudo matricular a la persona.

# Historias de usuario - Beneficios

---

- Al ser muy corta, ésta representa requisitos del modelo de negocio que pueden implementarse rápidamente (días o semanas).
- Necesitan poco mantenimiento.
- Mantienen una relación cercana con el cliente.
- Permite dividir los proyectos en pequeñas entregas.
- Permite estimar fácilmente el esfuerzo de desarrollo.
- Es ideal para proyectos con requisitos volátiles o no muy claros.





# Historias de usuario - Limitaciones

---

- Sin criterios de aceptación pueden quedar abiertas a distintas interpretaciones haciendo difícil utilizarlas como base para un contrato.
- Se requiere un contacto permanente con el cliente durante el proyecto lo cual puede ser difícil o costoso.
- Podría resultar difícil escalar a proyectos grandes.
- Requiere desarrolladores muy competentes.



# Historias de usuario - Épicas

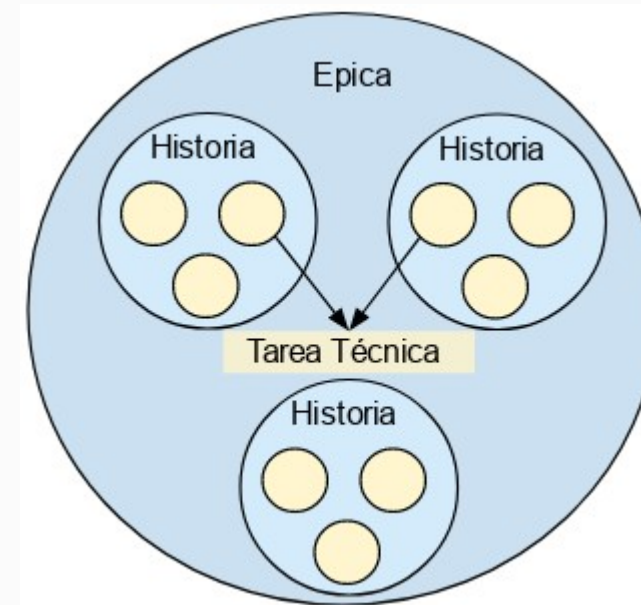
---

- Se denomina Épica a un conjunto de Historias de usuario que se agrupan por algún denominador común

La épica representa un objetivo alcanzable que nace de la necesidad del cliente.

Es un objetivo al que nos aproximamos y que esperamos alcanzar algún día.

La épica no es la funcionalidad.



# Ejemplo de Épicas

Épica: \_\_\_\_\_  
*Como Vicepresidente de mercadeo y ventas quiero revisar el desempeño histórico de las ventas para identificar las regiones geográficas y productos de mejor desempeño*

Esta épica se puede subdividir en:

Como VP de Mercadeo quiero seleccionar el período de tiempo en el cual realizaré la revisión de las ventas para analizar el desempeño

Como VP de Mercadeo quiero clasificar la información de ventas por región geográfica y productos para obtener la mejor zona de ventas

Este ejemplo fue obtenido de Mountain Goat Software. Two Examples of Splitting Epics

# Ejemplo de Épicas

Épica:

*Como un operador hotelero quiero establecer las tarifas óptimas para las habitaciones de mi hotel.*

Esta épica contiene las siguientes HU:

Como un operador hotelero quiero establecer la tarifa óptima para cotizar las habitaciones en base a los precios del año anterior.

Como un operador hotelero quiero establecer la tarifa óptima para cotizar las habitaciones en base a las tarifas de otros hoteles comparables con el mío.

44

Este ejemplo fue obtenido de **Mountain Goat Software. Two Examples of Splitting Epics**

# Características de las épicas

---

- » Las épicas suelen abarcar varios equipos de desarrollo
- » Recogen normalmente muchas historias de usuario
- » Los clientes determinan si eliminan o añaden historias dentro de cada épica
- » Una épica sirve para estructurar los temas e iniciativas (objetivos)
- » Las épicas también sirven para dar flexibilidad y agilidad al proyecto

45

# Bibliografía

---

- Libros consultados para técnicas de especificación de requerimientos

Pfleeger, Capítulo 4 , Ingeniería de Software, Pearson Prentice Hall 2002

Sommerville Ian, Capítulo 4, Ingeniería de software, Addison Wesley 2011

Whitten y Bentley, Análisis de Sistemas Diseño y Métodos, Capítulo 6, Mc Graw Hill 2008.

**Mike Cohn, User Stories Applied, Addison Wesley 2004.**