

# Modelado y Especificación de Requisitos Funcionales

Pablo Sánchez

Dpto. Ingeniería Informática y Electrónica  
Universidad de Cantabria  
Santander (Cantabria, España)  
[p.sanchez@unican.es](mailto:p.sanchez@unican.es)



# Advertencia

Todo el material contenido en este documento no constituye en modo alguno una obra de referencia o apuntes oficiales mediante los cuales se puedan preparar correctamente las pruebas evaluables necesarias para superar la asignatura.

Este documento contiene exclusivamente una serie de diapositivas cuyo objetivo es servir de complemento visual a las actividades realizadas en el aula.

Dicho de forma más clara, **estas transparencias no son apuntes y su objetivo no es en modo alguno servir para que el alumno pueda preparar la asignatura.**

# Objetivos del Tema

- 1 Entender el papel de los modelos orientados a solución en Ingeniería de Requisitos.
- 2 Aprender a descomponer jerárquicamente requisitos, utilizando tanto objetivos como escenarios.
- 3 Aprender a modelar y especificar requisitos de alto nivel utilizando objetivos.
- 4 Aprender a modelar y especificar requisitos de nivel medio utilizando escenarios.
- 5 Aprender a modelar y especificar requisitos de nivel medio utilizando historias de usuario.

# Bibliografía I



Axel van Lamsweerde.

*Requirements Engineering: From System Goals to UML Models to Software Specifications.*

Wiley, January 2009.



Alistair Cockburn.

*Writing Effective Use Cases.*

Addison-Wesley, Octubre 2000.



Klaus Pohl.

*Requirements Engineering: Fundamentals, Principles and Techniques.*

Springer, June 2010.



International Telecommunication Union (ITU).

*User Requirements Notation (URN) Language Definition.*

Standard Z.151. Octubre 2012.

## Bibliografía II



Cohn, M. (2004).

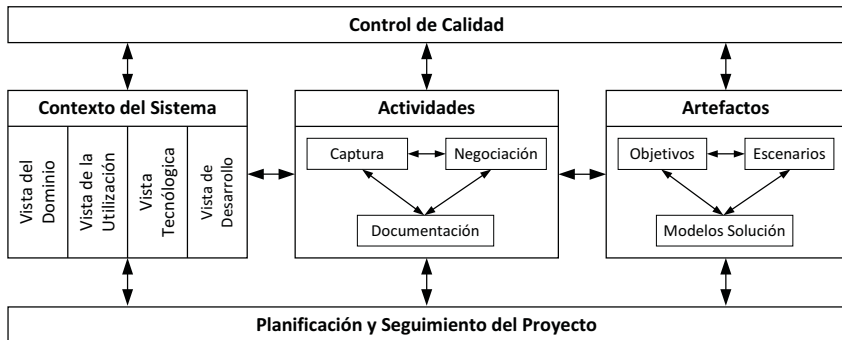
*User Stories Applied: For Agile Software Development.*

Addison-Wesley Professional.

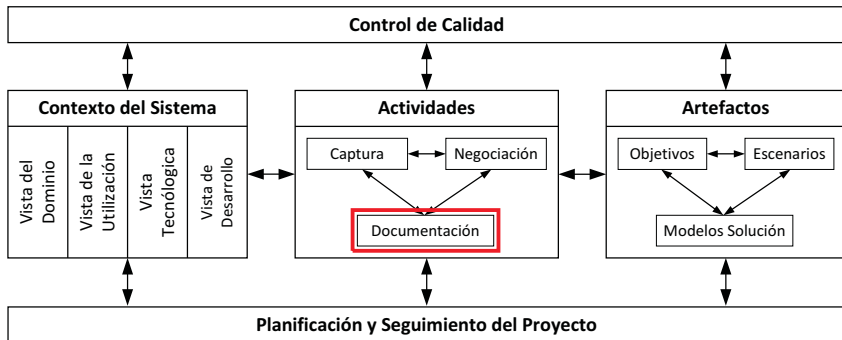
# Índice

- 1 **Introducción**
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Proceso de Ingeniería de Requisitos

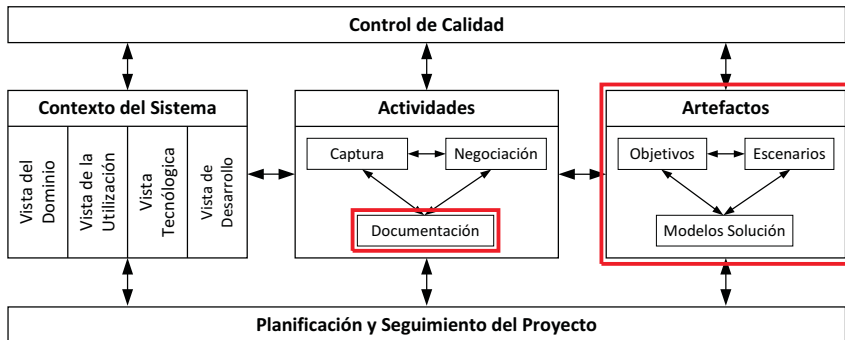


# Proceso de Ingeniería de Requisitos





# Proceso de Ingeniería de Requisitos



# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Proceso de Ingeniería de Requisitos



# Modelos Orientados a la Solución

## Requisitos Orientados a la Solución

Los *requisitos orientados a la solución* deben especificar, con un nivel suficiente de detalle, las propiedades y características del sistema a desarrollar.

# Perspectivas de los Requisitos Orientados a la Solución

**Datos** Especifica los datos que manipulará el sistema, así como las relaciones y restricciones entre ellos (EER, Clases UML).

**Funcional** Especifica las funcionalidades que implementará el sistema, las relaciones entre las entradas y salidas de cada función, así como las posibles dependencias entre funciones y restricciones (flujos de datos, actividades UML).

**Comportamiento** Especifica cómo reacciona un sistema a estímulos externos (máquinas de estado).

# Perspectivas de los Requisitos Orientados a la Solución

**Datos** Especifica los datos que manipulará el sistema, así como las relaciones y restricciones entre ellos (EER, Clases UML).

**Funcional** Especifica las funcionalidades que implementará el sistema, las relaciones entre las entradas y salidas de cada función, así como las posibles dependencias entre funciones y restricciones (flujos de datos, actividades UML).

**Comportamiento** Especifica cómo reacciona un sistema a estímulos externos (máquinas de estado).

# Perspectivas de los Requisitos Orientados a la Solución

**Datos** Especifica los datos que manipulará el sistema, así como las relaciones y restricciones entre ellos (EER, Clases UML).

**Funcional** Especifica las funcionalidades que implementará el sistema, las relaciones entre las entradas y salidas de cada función, así como las posibles dependencias entre funciones y restricciones (flujos de datos, actividades UML).

**Comportamiento** Especifica cómo reacciona un sistema a estímulos externos (máquinas de estado).

# Características de los Requisitos Orientados a la Solución

- 1 Consensuados y libres de conflictos.
- 2 Completos, suficientemente detallados y no ambiguos.
- 3 Enfocados a crear una solución sw (generación de código).



# Características de los Requisitos Orientados a la Solución

- 1 Consensuados y libres de conflictos.
- 2 Completos, suficientemente detallados y no ambiguos.
- 3 Enfocados a crear una solución sw (generación de código).

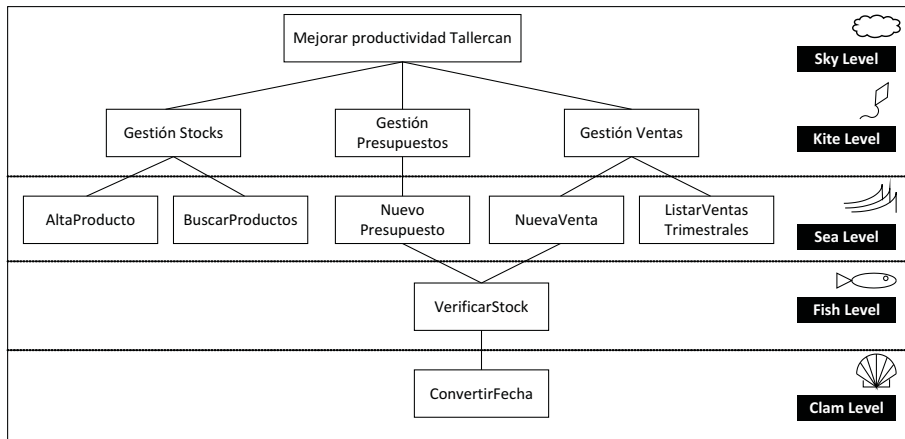
# Características de los Requisitos Orientados a la Solución

- 1 Consensuados y libres de conflictos.
- 2 Completos, suficientemente detallados y no ambiguos.
- 3 Enfocados a crear una solución sw (generación de código).

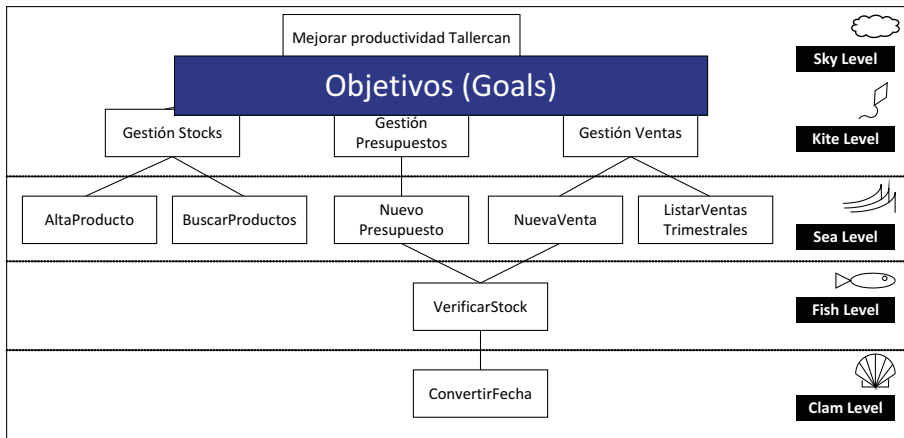
# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

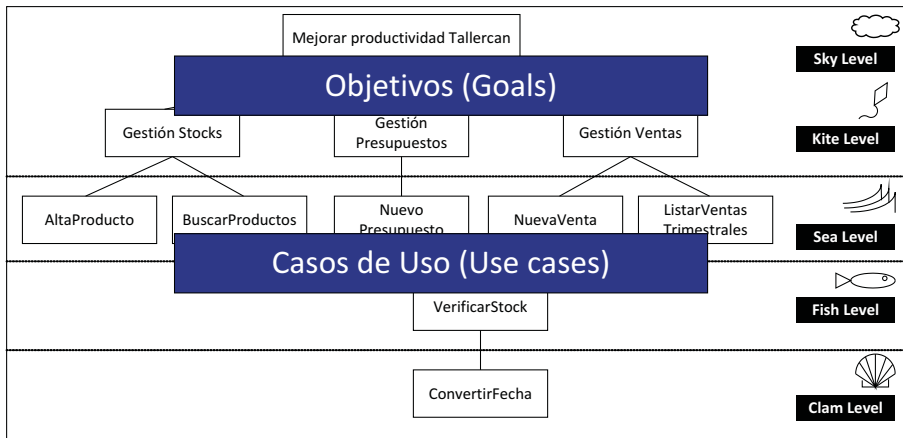
# Jerarquías de Requisitos



# Jerarquías de Requisitos



# Jerarquías de Requisitos



# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
  - **Introducción**
  - Sintaxis GRL/KAOS - Elementos Básicos
  - Sintaxis GRL/KAOS - Elementos Avanzados
  - Ventajas de los Objetivos
  - Proceso de Elaboración de Modelos Orientados a Objetivos
  - Especificación de Objetivos
  - Patrones Orientados a Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias



# Proceso de Ingeniería de Requisitos



# Definición de Objetivo

## Objetivos [van Lamsweerde, 2009]

Un *objetivo* de un sistema software es una descripción prescriptiva de un propósito que dicho sistema debe satisfacer mediante la cooperación de sus elementos constituyentes.

# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
  - Introducción
  - **Sintaxis GRL/KAOS - Elementos Básicos**
  - Sintaxis GRL/KAOS - Elementos Avanzados
  - Ventajas de los Objetivos
  - Proceso de Elaboración de Modelos Orientados a Objetivos
  - Especificación de Objetivos
  - Patrones Orientados a Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Modelado de Objetivos con GRL/KAOS

## Goal Requirements Language (GRL)

*Goal Requirements Language* es un lenguaje de modelado de objetivos, basado en  $i^*$ , integrado dentro del lenguaje *URN (User Requirements Notation)*, que es actualmente estándar ITU (*International Telecommunication Union*).

## Knowledge Acquisition in autOmated Specification of software (KAOS)

KAOS es una metodología con un lenguaje propio de modelado de objetivos, producida por el grupo de investigación del Prof. van Lamswerdee.

# Modelado de Objetivos con GRL/KAOS

## Goal Requirements Language (GRL)

*Goal Requirements Language* es un lenguaje de modelado de objetivos, basado en  $i^*$ , integrado dentro del lenguaje *URN (User Requirements Notation)*, que es actualmente estándar ITU (*International Telecommunication Union*).

## Knowledge Acquisition in autOmated Specification of software (KAOS)

KAOS es una metodología con un lenguaje propio de modelado de objetivos, producida por el grupo de investigación del Prof. van Lamswerdee.

# Sintaxis GRL/KAOS - Objetivo

## Objetivo (duro)

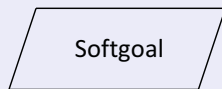
Un *objetivo (duro)* de un sistema sw es un objetivo cuya satisfacción debe alcanzarse de manera absoluta; y que, además, no puede satisfacerse parcialmente en la mayoría de los casos.



# Sintaxis GRL/KAOS - Objetivo Blando

## Objetivo Blando

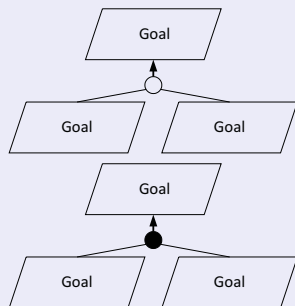
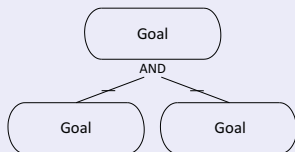
Un *objetivo blando* de un sistema sw es un objetivo cuya satisfacción no puede alcanzar de forma absoluta, de forma que es satisfecho hasta un cierto grado, el cual suele venir determinado por un *criterio de satisfacción*.



# Sintaxis GRL/KAOS - Relaciones AND

## Refinamiento AND

Un *refinamiento AND* descompone un objetivo padre en diversos subobjetivos hijos, los cuales han de satisfacerse todos para que el padre pueda ser satisfecho. Si la satisfacción de los objetivos es suficiente para la satisfacción del objetivo padre, se dice que el refinamiento es *completo*.

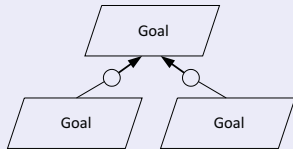
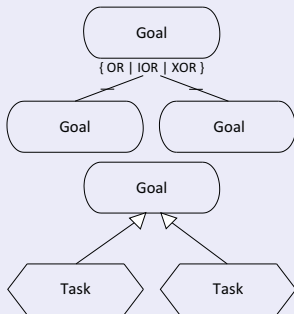




# Sintaxis GRL/KAOS - Relaciones OR

## Refinamiento OR

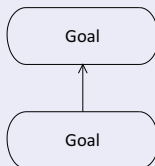
Un *refinamiento OR* descompone un objetivo padre en diversos subobjetivos hijos, de los cuales, al menos uno debe satisfacerse para que el padre pueda ser satisfecho. Las relaciones OR representan diferentes maneras de alcanzar un mismo objetivo.



# Sintaxis GRL/KAOS - Contribución

## Contribución (GRL)

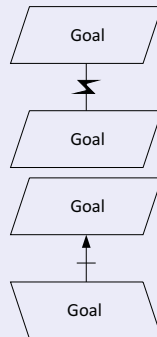
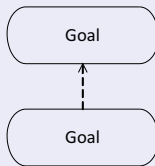
Una *contribución* describe el impacto intencionado que un elemento posee sobre un determinado objetivo.



# Sintaxis GRL/KAOS - Correlación

## Correlación

Una *correlación* describe el impacto no intencionado que un determinado elemento tiene sobre un determinado objetivo.



# Sintaxis GRL/KAOS - Escala de Contribución

## Tipos de Contribución/Correlación (GRL)

**Make** La contribución es positiva y suficiente.

**Help** La contribución es positiva, pero no suficiente.

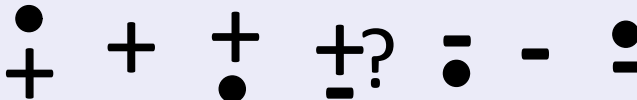
**Some Positive** La contribución es positiva, pero no significativa.

**Unknown** Un elemento afecta al otro, pero no se sabe cómo.

**Some Negative** La contribución es negativa, pero no significativa.

**Hurt** La contribución es negativa, pero no impide la satisfacción del elemento destino.

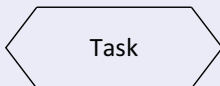
**Break** La contribución es negativa e impide la satisfacción del elemento destino.



# Sintaxis GRL/KAOS - Tarea

## Tarea/Operación

Un *tarea* u *operación* representa una forma concreta de llevar a cabo una acción que afecta al estado del sistema. Las tareas representan la materialización de un objetivo.



# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
  - Introducción
  - Sintaxis GRL/KAOS - Elementos Básicos
  - **Sintaxis GRL/KAOS - Elementos Avanzados**
  - Ventajas de los Objetivos
  - Proceso de Elaboración de Modelos Orientados a Objetivos
  - Especificación de Objetivos
  - Patrones Orientados a Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Sintaxis GRL/KAOS - Recurso

## Recurso/Entidad

Un *recurso* o *entidad* es un elemento físico o lógico que forma parte del sistema sw a construir o de su contexto.

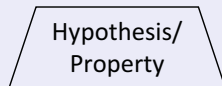
Resource

Concern

# Sintaxis GRL/KAOS - Hipótesis

## Creencia/Hipótesis o Propiedad

Una *creencia*, *hipótesis* o *propiedad* es un predicado sobre el dominio del sistema. Se utiliza normalmente para justificar una decisión. Las *propiedades* se diferencian de las *hipótesis* o *creencias* en que las primeras están probadas.

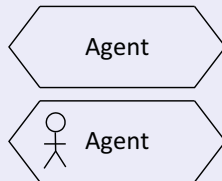




# Sintaxis GRL/KAOS - Agente

## Actor/Agente

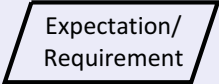
Un *actor* o *agente* es una entidad (persona o sistema) que tiene intereses o expectativas sobre el sistema y/o ejecuta acciones para satisfacer unos determinados objetivos.



# Sintaxis GRL/KAOS - Requisito

## Requisito o Expectativa (KAOS)

Un *requisito* es un objetivo que es responsabilidad de un único agente del sistema a construir. Una *expectativa* es un objetivo que es responsabilidad de un único agente del entorno del sistema a construir.

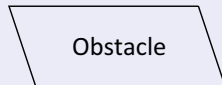


Expectation/  
Requirement

# Sintaxis GRL/KAOS - Obstáculo

## Obstáculo (KAOS)

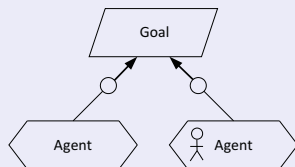
Un *obstáculo* a un objetivo es una situación factible dentro de un sistema software, que en caso de que ocurra, impide la satisfacción de dicho objetivo.



# Sintaxis GRL/KAOS - Responsabilidades

## Asignación Responsabilidad (KAOS)

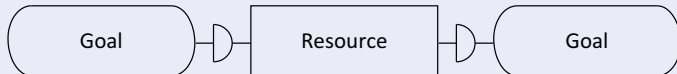
Una *asignación de responsabilidad* entre un objetivo y un agente indica que el agente es el responsable último de la satisfacción de dicho objetivo.



# Sintaxis GRL/KAOS - Dependencias GRL

## Dependencia

Una *dependencia* describe como un elemento de un actor depende de otro elemento de otro actor.



# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
  - Introducción
  - Sintaxis GRL/KAOS - Elementos Básicos
  - Sintaxis GRL/KAOS - Elementos Avanzados
  - **Ventajas de los Objetivos**
  - Proceso de Elaboración de Modelos Orientados a Objetivos
  - Especificación de Objetivos
  - Patrones Orientados a Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Ventajas de los Objetivos

- 1 Facilita el refinamiento de la *visión del sistema*.
- 2 Dirige y organiza el proceso de captura de requisitos.
- 3 Favorece la identificación y análisis de alternativas.
- 4 Permite detectar requisitos irrelevantes.
- 5 Permite justificar la existencia de los requisitos.
- 6 Permite evaluar la completitud de una especificación de requisitos.
- 7 Favorece la identificación y resolución de conflictos.
- 8 Los objetivos poseen una alta estabilidad con respecto a los cambios en el sistema.

# Ventajas de los Objetivos

- 1 Facilita el refinamiento de la *visión del sistema*.
- 2 Dirige y organiza el proceso de captura de requisitos.
- 3 Favorece la identificación y análisis de alternativas.
- 4 Permite detectar requisitos irrelevantes.
- 5 Permite justificar la existencia de los requisitos.
- 6 Permite evaluar la completitud de una especificación de requisitos.
- 7 Favorece la identificación y resolución de conflictos.
- 8 Los objetivos poseen una alta estabilidad con respecto a los cambios en el sistema.



# Ventajas de los Objetivos

- 1 Facilita el refinamiento de la *visión del sistema*.
- 2 Dirige y organiza el proceso de captura de requisitos.
- 3 Favorece la identificación y análisis de alternativas.
- 4 Permite detectar requisitos irrelevantes.
- 5 Permite justificar la existencia de los requisitos.
- 6 Permite evaluar la completitud de una especificación de requisitos.
- 7 Favorece la identificación y resolución de conflictos.
- 8 Los objetivos poseen una alta estabilidad con respecto a los cambios en el sistema.

# Ventajas de los Objetivos

- 1 Facilita el refinamiento de la *visión del sistema*.
- 2 Dirige y organiza el proceso de captura de requisitos.
- 3 Favorece la identificación y análisis de alternativas.
- 4 Permite detectar requisitos irrelevantes.
- 5 Permite justificar la existencia de los requisitos.
- 6 Permite evaluar la completitud de una especificación de requisitos.
- 7 Favorece la identificación y resolución de conflictos.
- 8 Los objetivos poseen una alta estabilidad con respecto a los cambios en el sistema.

# Ventajas de los Objetivos

- 1 Facilita el refinamiento de la *visión del sistema*.
- 2 Dirige y organiza el proceso de captura de requisitos.
- 3 Favorece la identificación y análisis de alternativas.
- 4 Permite detectar requisitos irrelevantes.
- 5 Permite justificar la existencia de los requisitos.
- 6 Permite evaluar la completitud de una especificación de requisitos.
- 7 Favorece la identificación y resolución de conflictos.
- 8 Los objetivos poseen una alta estabilidad con respecto a los cambios en el sistema.

# Ventajas de los Objetivos

- 1 Facilita el refinamiento de la *visión del sistema*.
- 2 Dirige y organiza el proceso de captura de requisitos.
- 3 Favorece la identificación y análisis de alternativas.
- 4 Permite detectar requisitos irrelevantes.
- 5 Permite justificar la existencia de los requisitos.
- 6 Permite evaluar la completitud de una especificación de requisitos.
- 7 Favorece la identificación y resolución de conflictos.
- 8 Los objetivos poseen una alta estabilidad con respecto a los cambios en el sistema.

# Ventajas de los Objetivos

- 1 Facilita el refinamiento de la *visión del sistema*.
- 2 Dirige y organiza el proceso de captura de requisitos.
- 3 Favorece la identificación y análisis de alternativas.
- 4 Permite detectar requisitos irrelevantes.
- 5 Permite justificar la existencia de los requisitos.
- 6 Permite evaluar la completitud de una especificación de requisitos.
- 7 Favorece la identificación y resolución de conflictos.
- 8 Los objetivos poseen una alta estabilidad con respecto a los cambios en el sistema.

# Ventajas de los Objetivos

- 1 Facilita el refinamiento de la *visión del sistema*.
- 2 Dirige y organiza el proceso de captura de requisitos.
- 3 Favorece la identificación y análisis de alternativas.
- 4 Permite detectar requisitos irrelevantes.
- 5 Permite justificar la existencia de los requisitos.
- 6 Permite evaluar la completitud de una especificación de requisitos.
- 7 Favorece la identificación y resolución de conflictos.
- 8 Los objetivos poseen una alta estabilidad con respecto a los cambios en el sistema.

# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
  - Introducción
  - Sintaxis GRL/KAOS - Elementos Básicos
  - Sintaxis GRL/KAOS - Elementos Avanzados
  - Ventajas de los Objetivos
  - Proceso de Elaboración de Modelos Orientados a Objetivos
  - Especificación de Objetivos
  - Patrones Orientados a Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Identificación de Objetivos

- 1 Identificar objetivos estratégicos y problemas del sistema actual.
- 2 Identificar deseos y esperanzas de cada *stakeholder*.
- 3 Buscar palabras claves en la documentación a analizar (oral o escrita).
- 4 Utilizar categorías predefinidas (no funcionales).
- 5 Análisis de ventajas e inconvenientes de cada alternativa.



# Identificación de Objetivos

- 1 Identificar objetivos estratégicos y problemas del sistema actual.
- 2 Identificar deseos y esperanzas de cada *stakeholder*.
- 3 Buscar palabras claves en la documentación a analizar (oral o escrita).
- 4 Utilizar categorías predefinidas (no funcionales).
- 5 Análisis de ventajas e inconvenientes de cada alternativa.

# Identificación de Objetivos

- 1 Identificar objetivos estratégicos y problemas del sistema actual.
- 2 Identificar deseos y esperanzas de cada *stakeholder*.
- 3 Buscar palabras claves en la documentación a analizar (oral o escrita).
- 4 Utilizar categorías predefinidas (no funcionales).
- 5 Análisis de ventajas e inconvenientes de cada alternativa.

# Identificación de Objetivos

- 1 Identificar objetivos estratégicos y problemas del sistema actual.
- 2 Identificar deseos y esperanzas de cada *stakeholder*.
- 3 Buscar palabras claves en la documentación a analizar (oral o escrita).
- 4 Utilizar categorías predefinidas (no funcionales).
- 5 Análisis de ventajas e inconvenientes de cada alternativa.

# Identificación de Objetivos

- 1 Identificar objetivos estratégicos y problemas del sistema actual.
- 2 Identificar deseos y esperanzas de cada *stakeholder*.
- 3 Buscar palabras claves en la documentación a analizar (oral o escrita).
- 4 Utilizar categorías predefinidas (no funcionales).
- 5 Análisis de ventajas e inconvenientes de cada alternativa.

# Refinado de Objetivos

- 1 Preguntar *cómo* y *por qué*.
- 2 Dividir responsabilidades.
- 3 Análisis de obstáculos, amenazas y conflictos.

# Refinado de Objetivos

- 1 Preguntar *cómo* y *por qué*.
- 2 Dividir responsabilidades.
- 3 Análisis de obstáculos, amenazas y conflictos.

# Refinado de Objetivos

- 1 Preguntar *cómo* y *por qué*.
- 2 Dividir responsabilidades.
- 3 Análisis de obstáculos, amenazas y conflictos.

# Criterio de Parada

- 1 Parar cuando un objetivo pueda ser asignado a un único agente o actor.
- 2 Parar cuando un objetivo esté fuera del contexto del sistema.



# Criterio de Parada

- 1 Parar cuando un objetivo pueda ser asignado a un único agente o actor.
- 2 Parar cuando un objetivo esté fuera del contexto del sistema.

# Fallos comunes

- ❶ Confundir objetivos y operaciones.
- ❷ Confundir descomposiciones AND con descomposiciones OR.
- ❸ Ambigüedades o sobreespecificaciones.
- ❹ Modelar secuencia temporal como dependencias.

# Fallos comunes

- ❶ Confundir objetivos y operaciones.
- ❷ Confundir descomposiciones AND con descomposiciones OR.
- ❸ Ambigüedades o sobreespecificaciones.
- ❹ Modelar secuencia temporal como dependencias.

# Fallos comunes

- ❶ Confundir objetivos y operaciones.
- ❷ Confundir descomposiciones AND con descomposiciones OR.
- ❸ Ambigüedades o sobreespecificaciones.
- ❹ Modelar secuencia temporal como dependencias.

# Fallos comunes

- ❶ Confundir objetivos y operaciones.
- ❷ Confundir descomposiciones AND con descomposiciones OR.
- ❸ Ambigüedades o sobreespecificaciones.
- ❹ Modelar secuencia temporal como dependencias.

# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
  - Introducción
  - Sintaxis GRL/KAOS - Elementos Básicos
  - Sintaxis GRL/KAOS - Elementos Avanzados
  - Ventajas de los Objetivos
  - Proceso de Elaboración de Modelos Orientados a Objetivos
  - **Especificación de Objetivos**
  - Patrones Orientados a Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Reglas para la Documentación de Objetivos

- 1 Concisión
- 2 Evitar usar oraciones pasivas (inglés) o impersonales.
- 3 Documentar la intención de los *stakeholders* de forma precisa y verificable.
- 4 Descomponer los objetivos de alto nivel en diversos objetivos de bajo nivel.
- 5 Redactar explícitamente el valor adicional de cada objetivo.
- 6 Explicar brevemente (si fuere necesario) la justificación de cada objetivo.
- 7 Evitar introducir restricciones innecesarias.

# Reglas para la Documentación de Objetivos

- 1 Concisión
- 2 Evitar usar oraciones pasivas (inglés) o impersonales.
- 3 Documentar la intención de los *stakeholders* de forma precisa y verificable.
- 4 Descomponer los objetivos de alto nivel en diversos objetivos de bajo nivel.
- 5 Redactar explícitamente el valor adicional de cada objetivo.
- 6 Explicar brevemente (si fuere necesario) la justificación de cada objetivo.
- 7 Evitar introducir restricciones innecesarias.



# Reglas para la Documentación de Objetivos

- 1 Concisión
- 2 Evitar usar oraciones pasivas (inglés) o impersonales.
- 3 Documentar la intención de los *stakeholders* de forma precisa y verificable.
- 4 Descomponer los objetivos de alto nivel en diversos objetivos de bajo nivel.
- 5 Redactar explícitamente el valor adicional de cada objetivo.
- 6 Explicar brevemente (si fuere necesario) la justificación de cada objetivo.
- 7 Evitar introducir restricciones innecesarias.

# Reglas para la Documentación de Objetivos

- 1 Concisión
- 2 Evitar usar oraciones pasivas (inglés) o impersonales.
- 3 Documentar la intención de los *stakeholders* de forma precisa y verificable.
- 4 Descomponer los objetivos de alto nivel en diversos objetivos de bajo nivel.
- 5 Redactar explícitamente el valor adicional de cada objetivo.
- 6 Explicar brevemente (si fuere necesario) la justificación de cada objetivo.
- 7 Evitar introducir restricciones innecesarias.

# Reglas para la Documentación de Objetivos

- 1 Concisión
- 2 Evitar usar oraciones pasivas (inglés) o impersonales.
- 3 Documentar la intención de los *stakeholders* de forma precisa y verificable.
- 4 Descomponer los objetivos de alto nivel en diversos objetivos de bajo nivel.
- 5 Redactar explícitamente el valor adicional de cada objetivo.
- 6 Explicar brevemente (si fuere necesario) la justificación de cada objetivo.
- 7 Evitar introducir restricciones innecesarias.

# Reglas para la Documentación de Objetivos

- 1 Concisión
- 2 Evitar usar oraciones pasivas (inglés) o impersonales.
- 3 Documentar la intención de los *stakeholders* de forma precisa y verificable.
- 4 Descomponer los objetivos de alto nivel en diversos objetivos de bajo nivel.
- 5 Redactar explícitamente el valor adicional de cada objetivo.
- 6 Explicar brevemente (si fuere necesario) la justificación de cada objetivo.
- 7 Evitar introducir restricciones innecesarias.

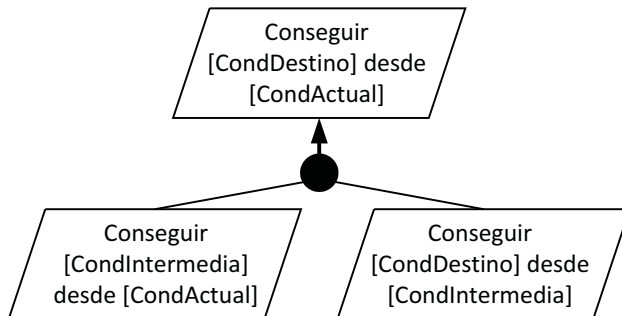
# Reglas para la Documentación de Objetivos

- 1 Concisión
- 2 Evitar usar oraciones pasivas (inglés) o impersonales.
- 3 Documentar la intención de los *stakeholders* de forma precisa y verificable.
- 4 Descomponer los objetivos de alto nivel en diversos objetivos de bajo nivel.
- 5 Redactar explícitamente el valor adicional de cada objetivo.
- 6 Explicar brevemente (si fuere necesario) la justificación de cada objetivo.
- 7 Evitar introducir restricciones innecesarias.

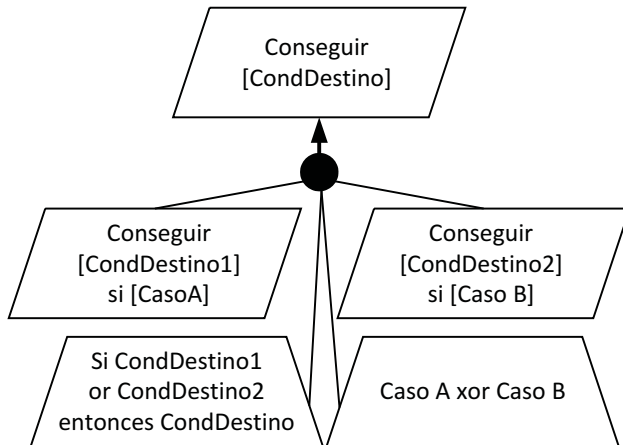
# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
  - Introducción
  - Sintaxis GRL/KAOS - Elementos Básicos
  - Sintaxis GRL/KAOS - Elementos Avanzados
  - Ventajas de los Objetivos
  - Proceso de Elaboración de Modelos Orientados a Objetivos
  - Especificación de Objetivos
  - **Patrones Orientados a Objetivos**
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Patrón Punto Intermedio

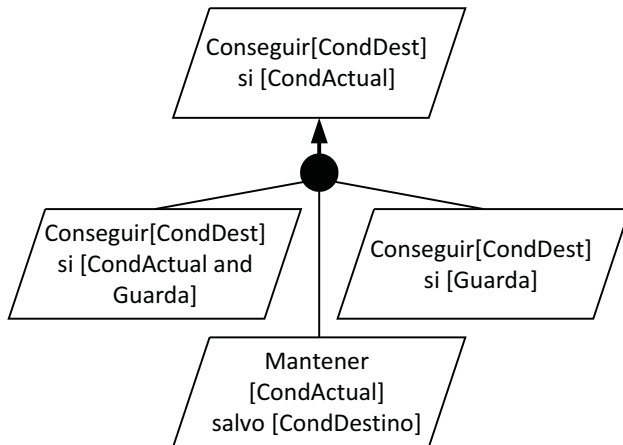


# Patrón Descomposición por Casos

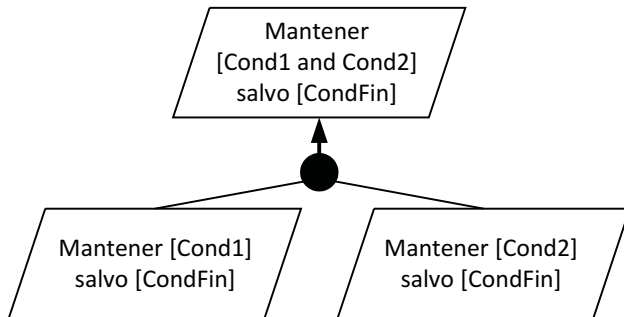




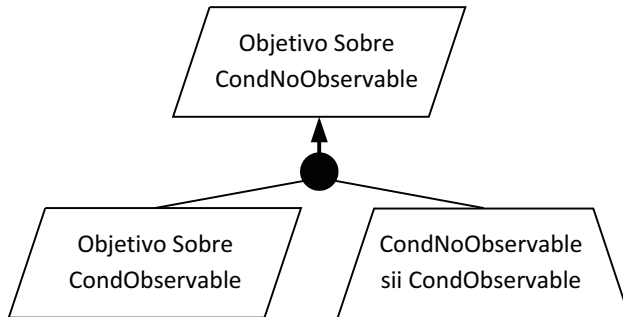
# Patrón Introducir Guarda



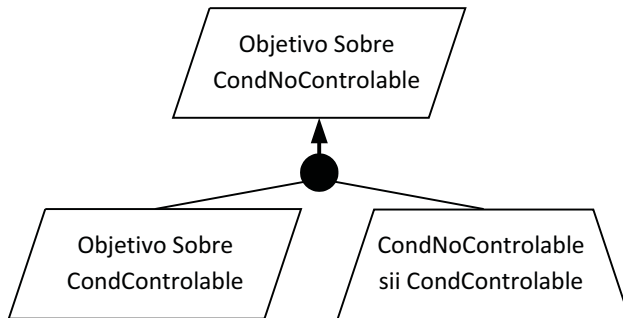
# Patrón Divide y Vencerás



# Patrón Condición No Observable



# Patrón Condición No Controlable



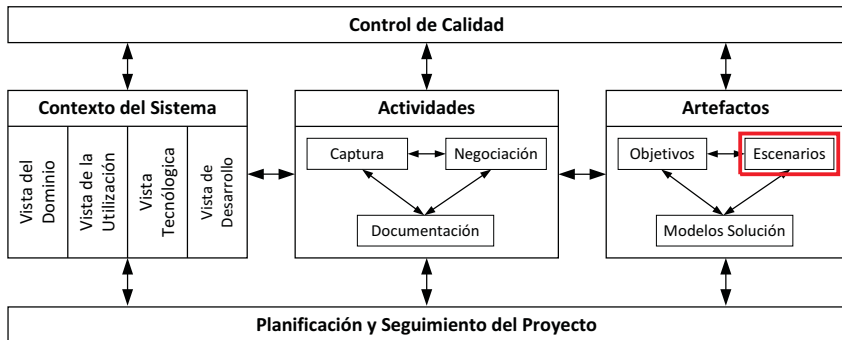
# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
- 5 **Modelado y Especificación de Escenarios**
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
- 5 Modelado y Especificación de Escenarios
  - Definición de Escenario
  - Elementos de un Escenario
  - Reglas de Escritura de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Proceso de Ingeniería de Requisitos



# Definición de Escenario

## Escenario [Pohl, 2010]

Un *escenario* describe un ejemplo concreto de satisfacción, o fallo en la satisfacción, de un determinado objetivo o conjunto de objetivos.



# Tipos de Escenarios

- 1 Sistema actual o sistema a construir.
- 2 Positivos o negativos (permitidos o prohibidos).
- 3 Descriptivos, exploratorios o explicativos.
- 4 De tipos, de instancia o mixtos.
- 5 Internos, de interacción con el sistema o de contexto.
- 6 Principal, alternativo o excepcional.

# Tipos de Escenarios

- 1 Sistema actual o sistema a construir.
- 2 Positivos o negativos (permitidos o prohibidos).
- 3 Descriptivos, exploratorios o explicativos.
- 4 De tipos, de instancia o mixtos.
- 5 Internos, de interacción con el sistema o de contexto.
- 6 Principal, alternativo o excepcional.

# Tipos de Escenarios

- 1 Sistema actual o sistema a construir.
- 2 Positivos o negativos (permitidos o prohibidos).
- 3 Descriptivos, exploratorios o explicativos.
- 4 De tipos, de instancia o mixtos.
- 5 Internos, de interacción con el sistema o de contexto.
- 6 Principal, alternativo o excepcional.

# Tipos de Escenarios

- ➊ Sistema actual o sistema a construir.
- ➋ Positivos o negativos (permitidos o prohibidos).
- ➌ Descriptivos, exploratorios o explicativos.
- ➍ De tipos, de instancia o mixtos.
- ➎ Internos, de interacción con el sistema o de contexto.
- ➏ Principal, alternativo o excepcional.

# Tipos de Escenarios

- ➊ Sistema actual o sistema a construir.
- ➋ Positivos o negativos (permitidos o prohibidos).
- ➌ Descriptivos, exploratorios o explicativos.
- ➍ De tipos, de instancia o mixtos.
- ➎ Internos, de interacción con el sistema o de contexto.
- ➏ Principal, alternativo o excepcional.

# Tipos de Escenarios

- 1 Sistema actual o sistema a construir.
- 2 Positivos o negativos (permitidos o prohibidos).
- 3 Descriptivos, exploratorios o explicativos.
- 4 De tipos, de instancia o mixtos.
- 5 Internos, de interacción con el sistema o de contexto.
- 6 Principal, alternativo o excepcional.

# Escenario de Éxito, Escenario Principal y Extensión

**Escenario de Éxito** Escenario que termina con la satisfacción del objetivo.

Escenario de No Éxito Escenario que no es de éxito

Escenario Principal Escenario de éxito que describe el modo habitual de utilización del sistema.

Extensión Fragmento de un escenario que describe una variación, de éxito o no, del mismo.

Condición de Extensión Razones o circunstancias por las cuales un cierto comportamiento se desvía de su escenario principal y entra en una extensión.

# Escenario de Éxito, Escenario Principal y Extensión

**Escenario de Éxito** Escenario que termina con la satisfacción del objetivo.

**Escenario de No Éxito** Escenario que no es de éxito

**Escenario Principal** Escenario de éxito que describe el modo habitual de utilización del sistema.

**Extensión** Fragmento de un escenario que describe una variación, de éxito o no, del mismo.

**Condición de Extensión** Razones o circunstancias por las cuales un cierto comportamiento se desvía de su escenario principal y entra en una extensión.



# Escenario de Éxito, Escenario Principal y Extensión

**Escenario de Éxito** Escenario que termina con la satisfacción del objetivo.

**Escenario de No Éxito** Escenario que no es de éxito

**Escenario Principal** Escenario de éxito que describe el modo habitual de utilización del sistema.

**Extensión** Fragmento de un escenario que describe una variación, de éxito o no, del mismo.

**Condición de Extensión** Razones o circunstancias por las cuales un cierto comportamiento se desvía de su escenario principal y entra en una extensión.

# Escenario de Éxito, Escenario Principal y Extensión

**Escenario de Éxito** Escenario que termina con la satisfacción del objetivo.

**Escenario de No Éxito** Escenario que no es de éxito

**Escenario Principal** Escenario de éxito que describe el modo habitual de utilización del sistema.

**Extensión** Fragmento de un escenario que describe una variación, de éxito o no, del mismo.

**Condición de Extensión** Razones o circunstancias por las cuales un cierto comportamiento se desvía de su escenario principal y entra en una extensión.

# Escenario de Éxito, Escenario Principal y Extensión

**Escenario de Éxito** Escenario que termina con la satisfacción del objetivo.

**Escenario de No Éxito** Escenario que no es de éxito

**Escenario Principal** Escenario de éxito que describe el modo habitual de utilización del sistema.

**Extensión** Fragmento de un escenario que describe una variación, de éxito o no, del mismo.

**Condición de Extensión** Razones o circunstancias por las cuales un cierto comportamiento se desvía de su escenario principal y entra en una extensión.

# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
- 5 Modelado y Especificación de Escenarios
  - Definición de Escenario
  - Elementos de un Escenario
  - Reglas de Escritura de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Elementos de un Escenario

**Evento de Disparo (Trigger)** Evento que da lugar al inicio del caso de uso.

**Precondición** Condición necesaria para que el escenario termine con éxito.

**Garantías Mínimas** Condición mínima que ha de satisfacerse cuando el objetivo del escenario no puede alcanzarse.

**Garantías de Éxito** Condición que se cumple siempre y cuando el escenario termina con éxito.

**Paso** Acción con o del sistema que puede ser:

**Escenario Principal** Conjunto de pasos que describen el escenario principal.

**Extensiones** Descripción de las posibles extensiones.

## Elementos de un Escenario

**Evento de Disparo (Trigger)** Evento que da lugar al inicio del caso de uso.

**Precondición** Condición necesaria para que el escenario termine con éxito.

**Garantías Mínimas** Condición mínima que ha de satisfacerse cuando el objetivo del escenario no puede alcanzarse.

**Garantías de Éxito** Condición que se cumple siempre y cuando el escenario termina con éxito.

**Paso** Acción con o del sistema que puede ser:

**Escenario Principal** Conjunto de pasos que describen el escenario principal.

**Extensiones** Descripción de las posibles extensiones.



# Elementos de un Escenario

**Evento de Disparo (Trigger)** Evento que da lugar al inicio del caso de uso.

**Precondición** Condición necesaria para que el escenario termine con éxito.

**Garantías Mínimas** Condición mínima que ha de satisfacerse cuando el objetivo del escenario no puede alcanzarse.

**Garantías de Éxito** Condición que se cumple siempre y cuando el escenario termina con éxito.

**Paso** Acción con o del sistema que puede ser:

- Una interacción entre un actor y el sistema (en cualquier dirección),

- Una interacción entre componentes del sistema.

- Una interacción entre el sistema y un sistema externo.

- Una condición que se establece para el escenario, como una precondición o una garantía.

**Escenario Principal** Conjunto de pasos que describen el escenario principal.

**Extensiones** Descripción de las posibles extensiones.



# Elementos de un Escenario

**Evento de Disparo (Trigger)** Evento que da lugar al inicio del caso de uso.

**Precondición** Condición necesaria para que el escenario termine con éxito.

**Garantías Mínimas** Condición mínima que ha de satisfacerse cuando el objetivo del escenario no puede alcanzarse.

**Garantías de Éxito** Condición que se cumple siempre y cuando el escenario termina con éxito.

**Paso** Acción con o del sistema que puede ser:

- 1 Una interacción entre un actor y el sistema (en cualquier dirección);
- 2 Una validación para proteger algún objetivo de un *stakeholder*;
- 3 Un cambio interno en el sistema para satisfacer algún objetivo de un *stakeholder*.

**Escenario Principal** Conjunto de pasos que describen el escenario principal.

**Extensiones** Descripción de las posibles extensiones.

# Elementos de un Escenario

**Evento de Disparo (Trigger)** Evento que da lugar al inicio del caso de uso.

**Precondición** Condición necesaria para que el escenario termine con éxito.

**Garantías Mínimas** Condición mínima que ha de satisfacerse cuando el objetivo del escenario no puede alcanzarse.

**Garantías de Éxito** Condición que se cumple siempre y cuando el escenario termina con éxito.

**Paso** Acción con o del sistema que puede ser:

- 1 Una interacción entre un actor y el sistema (en cualquier dirección);
- 2 Una validación para proteger algún objetivo de un *stakeholder*;
- 3 Un cambio interno en el sistema para satisfacer algún objetivo de un *stakeholder*.

**Escenario Principal** Conjunto de pasos que describen el escenario principal.

**Extensiones** Descripción de las posibles extensiones.

# Elementos de un Escenario

**Evento de Disparo (Trigger)** Evento que da lugar al inicio del caso de uso.

**Precondición** Condición necesaria para que el escenario termine con éxito.

**Garantías Mínimas** Condición mínima que ha de satisfacerse cuando el objetivo del escenario no puede alcanzarse.

**Garantías de Éxito** Condición que se cumple siempre y cuando el escenario termina con éxito.

**Paso** Acción con o del sistema que puede ser:

- 1 Una interacción entre un actor y el sistema (en cualquier dirección);
- 2 Una validación para proteger algún objetivo de un *stakeholder*;
- 3 Un cambio interno en el sistema para satisfacer algún objetivo de un *stakeholder*.

**Escenario Principal** Conjunto de pasos que describen el escenario principal.

**Extensiones** Descripción de las posibles extensiones.

# Elementos de un Escenario

**Evento de Disparo (Trigger)** Evento que da lugar al inicio del caso de uso.

**Precondición** Condición necesaria para que el escenario termine con éxito.

**Garantías Mínimas** Condición mínima que ha de satisfacerse cuando el objetivo del escenario no puede alcanzarse.

**Garantías de Éxito** Condición que se cumple siempre y cuando el escenario termina con éxito.

**Paso** Acción con o del sistema que puede ser:

- 1 Una interacción entre un actor y el sistema (en cualquier dirección);
- 2 Una validación para proteger algún objetivo de un *stakeholder*;
- 3 Un cambio interno en el sistema para satisfacer algún objetivo de un *stakeholder*.

**Escenario Principal** Conjunto de pasos que describen el escenario principal.

**Extensiones** Descripción de las posibles extensiones.

# Elementos de un Escenario

**Evento de Disparo (Trigger)** Evento que da lugar al inicio del caso de uso.

**Precondición** Condición necesaria para que el escenario termine con éxito.

**Garantías Mínimas** Condición mínima que ha de satisfacerse cuando el objetivo del escenario no puede alcanzarse.

**Garantías de Éxito** Condición que se cumple siempre y cuando el escenario termina con éxito.

**Paso** Acción con o del sistema que puede ser:

- 1 Una interacción entre un actor y el sistema (en cualquier dirección);
- 2 Una validación para proteger algún objetivo de un *stakeholder*;
- 3 Un cambio interno en el sistema para satisfacer algún objetivo de un *stakeholder*.

**Escenario Principal** Conjunto de pasos que describen el escenario principal.

**Extensiones** Descripción de las posibles extensiones.

# Elementos de un Escenario

**Evento de Disparo (Trigger)** Evento que da lugar al inicio del caso de uso.

**Precondición** Condición necesaria para que el escenario termine con éxito.

**Garantías Mínimas** Condición mínima que ha de satisfacerse cuando el objetivo del escenario no puede alcanzarse.

**Garantías de Éxito** Condición que se cumple siempre y cuando el escenario termina con éxito.

**Paso** Acción con o del sistema que puede ser:

- 1 Una interacción entre un actor y el sistema (en cualquier dirección);
- 2 Una validación para proteger algún objetivo de un *stakeholder*;
- 3 Un cambio interno en el sistema para satisfacer algún objetivo de un *stakeholder*.

**Escenario Principal** Conjunto de pasos que describen el escenario principal.

**Extensiones** Descripción de las posibles extensiones.

# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
- 5 Modelado y Especificación de Escenarios
  - Definición de Escenario
  - Elementos de un Escenario
  - Reglas de Escritura de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Reglas de Escritura de Escenarios

- 1 Usar una gramática simple.
- 2 Indicar claramente quién tiene el control y quién recibe resultado de las acciones.
- 3 Escribir el caso de uso *desde fuera del sistema*.
- 4 Mostrar el proceso avanzando.
- 5 Describir las intenciones del usuario, no sus movimientos.
- 6 Escribir *validar*, en lugar de *el sistema comprueba si*.
- 7 Mencionar si es necesario la temporización.
- 8 Escribir *el usuario requiere que el sistema ...* para acciones con terceros.
- 9 Truco: *Repetir los pasos X-Y hasta que <condición>*.
- 10 Truco: *Los pasos X-Y pueden suceder en cualquier orden*.



# Reglas de Escritura de Escenarios

- 1 Usar una gramática simple.
- 2 Indicar claramente quién tiene el control y quién recibe resultado de las acciones.
- 3 Escribir el caso de uso *desde fuera del sistema*.
- 4 Mostrar el proceso avanzando.
- 5 Describir las intenciones del usuario, no sus movimientos.
- 6 Escribir *validar*, en lugar de *el sistema comprueba si*.
- 7 Mencionar si es necesario la temporización.
- 8 Escribir *el usuario requiere que el sistema ...* para acciones con terceros.
- 9 Truco: *Repetir los pasos X-Y hasta que <condición>*.
- 10 Truco: *Los pasos X-Y pueden suceder en cualquier orden*.

# Reglas de Escritura de Escenarios

- ➊ Usar una gramática simple.
- ➋ Indicar claramente quién tiene el control y quién recibe resultado de las acciones.
- ➌ Escribir el caso de uso *desde fuera del sistema*.
- ➍ Mostrar el proceso avanzando.
- ➎ Describir las intenciones del usuario, no sus movimientos.
- ➏ Escribir *validar*, en lugar de *el sistema comprueba si*.
- ➐ Mencionar si es necesario la temporización
- ➑ Escribir *el usuario requiere que el sistema ...* para acciones con terceros.
- ➒ Truco: *Repetir los pasos X-Y hasta que <condición>*.
- ➓ Truco: *Los pasos X-Y pueden suceder en cualquier orden*.

# Reglas de Escritura de Escenarios

- 1 Usar una gramática simple.
- 2 Indicar claramente quién tiene el control y quién recibe resultado de las acciones.
- 3 Escribir el caso de uso *desde fuera del sistema*.
- 4 Mostrar el proceso avanzando.
- 5 Describir las intenciones del usuario, no sus movimientos.
- 6 Escribir *validar*, en lugar de *el sistema comprueba si*.
- 7 Mencionar si es necesario la temporización.
- 8 Escribir *el usuario requiere que el sistema ...* para acciones con terceros.
- 9 Truco: *Repetir los pasos X-Y hasta que <condición>*.
- 10 Truco: *Los pasos X-Y pueden suceder en cualquier orden*.

# Reglas de Escritura de Escenarios

- ➊ Usar una gramática simple.
- ➋ Indicar claramente quién tiene el control y quién recibe resultado de las acciones.
- ➌ Escribir el caso de uso *desde fuera del sistema*.
- ➍ Mostrar el proceso avanzando.
- ➎ Describir las intenciones del usuario, no sus movimientos.
- ➏ Escribir *validar*, en lugar de *el sistema comprueba si*.
- ➐ Mencionar si es necesario la temporización
- ➑ Escribir *el usuario requiere que el sistema ...* para acciones con terceros.
- ➒ Truco: *Repetir los pasos X-Y hasta que <condición>*.
- ➓ Truco: *Los pasos X-Y pueden suceder en cualquier orden*.

# Reglas de Escritura de Escenarios

- ➊ Usar una gramática simple.
- ➋ Indicar claramente quién tiene el control y quién recibe resultado de las acciones.
- ➌ Escribir el caso de uso *desde fuera del sistema*.
- ➍ Mostrar el proceso avanzando.
- ➎ Describir las intenciones del usuario, no sus movimientos.
- ➏ Escribir *validar*, en lugar de *el sistema comprueba si*.
- ➐ Mencionar si es necesario la temporización
- ➑ Escribir *el usuario requiere que el sistema ...* para acciones con terceros.
- ➒ Truco: *Repetir los pasos X-Y hasta que <condición>*.
- ➓ Truco: *Los pasos X-Y pueden suceder en cualquier orden*.

# Reglas de Escritura de Escenarios

- ➊ Usar una gramática simple.
- ➋ Indicar claramente quién tiene el control y quién recibe resultado de las acciones.
- ➌ Escribir el caso de uso *desde fuera del sistema*.
- ➍ Mostrar el proceso avanzando.
- ➎ Describir las intenciones del usuario, no sus movimientos.
- ➏ Escribir *validar*, en lugar de *el sistema comprueba si*.
- ➐ Mencionar si es necesario la temporización
- ➑ Escribir *el usuario requiere que el sistema ...* para acciones con terceros.
- ➒ Truco: *Repetir los pasos X-Y hasta que <condición>*.
- ➓ Truco: *Los pasos X-Y pueden suceder en cualquier orden*.

# Reglas de Escritura de Escenarios

- ➊ Usar una gramática simple.
- ➋ Indicar claramente quién tiene el control y quién recibe resultado de las acciones.
- ➌ Escribir el caso de uso *desde fuera del sistema*.
- ➍ Mostrar el proceso avanzando.
- ➎ Describir las intenciones del usuario, no sus movimientos.
- ➏ Escribir *validar*, en lugar de *el sistema comprueba si*.
- ➐ Mencionar si es necesario la temporización
- ➑ Escribir *el usuario requiere que el sistema ...* para acciones con terceros.
- ➒ Truco: *Repetir los pasos X-Y hasta que <condición>.*
- ➓ Truco: *Los pasos X-Y pueden suceder en cualquier orden.*

# Reglas de Escritura de Escenarios

- ➊ Usar una gramática simple.
- ➋ Indicar claramente quién tiene el control y quién recibe resultado de las acciones.
- ➌ Escribir el caso de uso *desde fuera del sistema*.
- ➍ Mostrar el proceso avanzando.
- ➎ Describir las intenciones del usuario, no sus movimientos.
- ➏ Escribir *validar*, en lugar de *el sistema comprueba si*.
- ➐ Mencionar si es necesario la temporización
- ➑ Escribir *el usuario requiere que el sistema ...* para acciones con terceros.
- ➒ Truco: *Repetir los pasos X-Y hasta que <condición>*.
- ➓ Truco: *Los pasos X-Y pueden suceder en cualquier orden.*



# Reglas de Escritura de Escenarios

- ➊ Usar una gramática simple.
- ➋ Indicar claramente quién tiene el control y quién recibe resultado de las acciones.
- ➌ Escribir el caso de uso *desde fuera del sistema*.
- ➍ Mostrar el proceso avanzando.
- ➎ Describir las intenciones del usuario, no sus movimientos.
- ➏ Escribir *validar*, en lugar de *el sistema comprueba si*.
- ➐ Mencionar si es necesario la temporización
- ➑ Escribir *el usuario requiere que el sistema ...* para acciones con terceros.
- ➒ Truco: *Repetir los pasos X-Y hasta que <condición>*.
- ➓ Truco: *Los pasos X-Y pueden suceder en cualquier orden*.

# Identificación de Extensiones

- ❶ Identificar, mediante una técnica adecuada, todas las desviaciones del escenario principal que se consigan imaginar. Prestar atención a:
  - ❶ Caminos de éxito alternativos.
  - ❷ Comportamientos erróneos del actor primario y último.
  - ❸ Inactividad del actor primario y último.
  - ❹ Inactividad de un actor secundario con el que interactúa el sistema.
  - ❺ Caminos alternativos para *el sistema válida*.
  - ❻ Fallo interno, detectable y recuperable del sistema.
  - ❼ Fallos inesperados.
  - ❽ Fallos de rendimiento.
- ❷ Descartar aquellas extensiones cuyas condiciones sean indetectables.
- ❸ Fusionar aquellas extensiones cuyas condiciones y efecto sean equivalente.
- ❹ Escribir cada extensión.

# Identificación de Extensiones

- ❶ Identificar, mediante una técnica adecuada, todas las desviaciones del escenario principal que se consigan imaginar. Prestar atención a:
  - ❶ Caminos de éxito alternativos.
  - ❷ Comportamientos erróneos del actor primario y último.
  - ❸ Inactividad del actor primario y último.
  - ❹ Inactividad de un actor secundario con el que interactúa el sistema.
  - ❺ Caminos alternativos para *el sistema válida*.
  - ❻ Fallo interno, detectable y recuperable del sistema.
  - ❼ Fallos inesperados.
  - ❽ Fallos de rendimiento.
- ❷ Descartar aquellas extensiones cuyas condiciones sean indetectables.
- ❸ Fusionar aquellas extensiones cuyas condiciones y efecto sean equivalente.
- ❹ Escribir cada extensión.

# Identificación de Extensiones

- ➊ Identificar, mediante una técnica adecuada, todas las desviaciones del escenario principal que se consigan imaginar. Prestar atención a:
  - ➊ Caminos de éxito alternativos.
  - ➋ Comportamientos erróneos del actor primario y último.
  - ➌ Inactividad del actor primario y último.
  - ➍ Inactividad de un actor secundario con el que interactúa el sistema.
  - ➎ Caminos alternativos para *el sistema válida*.
  - ➏ Fallo interno, detectable y recuperable del sistema.
  - ➐ Fallos inesperados.
  - ➑ Fallos de rendimiento.
- ➋ Descartar aquellas extensiones cuyas condiciones sean indetectables.
- ➌ Fusionar aquellas extensiones cuyas condiciones y efecto sean equivalente.
- ➍ Escribir cada extensión.

# Identificación de Extensiones

- ❶ Identificar, mediante una técnica adecuada, todas las desviaciones del escenario principal que se consigan imaginar. Prestar atención a:
  - ❶ Caminos de éxito alternativos.
  - ❷ Comportamientos erróneos del actor primario y último.
  - ❸ Inactividad del actor primario y último.
  - ❹ Inactividad de un actor secundario con el que interactúa el sistema.
  - ❺ Caminos alternativos para *el sistema válida*.
  - ❻ Fallo interno, detectable y recuperable del sistema.
  - ❼ Fallos inesperados.
  - ❽ Fallos de rendimiento.
- ❷ Descartar aquellas extensiones cuyas condiciones sean indetectables.
- ❸ Fusionar aquellas extensiones cuyas condiciones y efecto sean equivalente.
- ❹ Escribir cada extensión.

# Identificación de Extensiones

- ➊ Identificar, mediante una técnica adecuada, todas las desviaciones del escenario principal que se consigan imaginar. Prestar atención a:
  - ➊ Caminos de éxito alternativos.
  - ➋ Comportamientos erróneos del actor primario y último.
  - ➌ Inactividad del actor primario y último.
  - ➍ Inactividad de un actor secundario con el que interactúa el sistema.
  - ➎ Caminos alternativos para *el sistema válida*.
  - ➏ Fallo interno, detectable y recuperable del sistema.
  - ➐ Fallos inesperados.
  - ➑ Fallos de rendimiento.
- ➋ Descartar aquellas extensiones cuyas condiciones sean indetectables.
- ➌ Fusionar aquellas extensiones cuyas condiciones y efecto sean equivalente.
- ➍ Escribir cada extensión.

# Identificación de Extensiones

- ❶ Identificar, mediante una técnica adecuada, todas las desviaciones del escenario principal que se consigan imaginar. Prestar atención a:
  - ❶ Caminos de éxito alternativos.
  - ❷ Comportamientos erróneos del actor primario y último.
  - ❸ Inactividad del actor primario y último.
  - ❹ Inactividad de un actor secundario con el que interactúa el sistema.
  - ❺ Caminos alternativos para *el sistema válida*.
  - ❻ Fallo interno, detectable y recuperable del sistema.
  - ❼ Fallos inesperados.
  - ❽ Fallos de rendimiento.
- ❷ Descartar aquellas extensiones cuyas condiciones sean indetectables.
- ❸ Fusionar aquellas extensiones cuyas condiciones y efecto sean equivalente.
- ❹ Escribir cada extensión.

# Identificación de Extensiones

- ➊ Identificar, mediante una técnica adecuada, todas las desviaciones del escenario principal que se consigan imaginar. Prestar atención a:
  - ➊ Caminos de éxito alternativos.
  - ➋ Comportamientos erróneos del actor primario y último.
  - ➌ Inactividad del actor primario y último.
  - ➍ Inactividad de un actor secundario con el que interactúa el sistema.
  - ➎ Caminos alternativos para *el sistema válida*.
  - ➏ Fallo interno, detectable y recuperable del sistema.
  - ➐ Fallos inesperados.
  - ➑ Fallos de rendimiento.
- ➋ Descartar aquellas extensiones cuyas condiciones sean indetectables.
- ➌ Fusionar aquellas extensiones cuyas condiciones y efecto sean equivalente.
- ➍ Escribir cada extensión.



# Identificación de Extensiones

- ➊ Identificar, mediante una técnica adecuada, todas las desviaciones del escenario principal que se consigan imaginar. Prestar atención a:
  - ➊ Caminos de éxito alternativos.
  - ➋ Comportamientos erróneos del actor primario y último.
  - ➌ Inactividad del actor primario y último.
  - ➍ Inactividad de un actor secundario con el que interactúa el sistema.
  - ➎ Caminos alternativos para *el sistema válida*.
  - ➏ Fallo interno, detectable y recuperable del sistema.
  - ➐ Fallos inesperados.
  - ➑ Fallos de rendimiento.
- ➋ Descartar aquellas extensiones cuyas condiciones sean indetectables.
- ➌ Fusionar aquellas extensiones cuyas condiciones y efecto sean equivalente.
- ➍ Escribir cada extensión.

# Identificación de Extensiones

- ❶ Identificar, mediante una técnica adecuada, todas las desviaciones del escenario principal que se consigan imaginar. Prestar atención a:
  - ❶ Caminos de éxito alternativos.
  - ❷ Comportamientos erróneos del actor primario y último.
  - ❸ Inactividad del actor primario y último.
  - ❹ Inactividad de un actor secundario con el que interactúa el sistema.
  - ❺ Caminos alternativos para *el sistema válida*.
  - ❻ Fallo interno, detectable y recuperable del sistema.
  - ❼ Fallos inesperados.
  - ❽ Fallos de rendimiento.
- ❷ Descartar aquellas extensiones cuyas condiciones sean indetectables.
- ❸ Fusionar aquellas extensiones cuyas condiciones y efecto sean equivalente.
- ❹ Escribir cada extensión.

# Identificación de Extensiones

- ➊ Identificar, mediante una técnica adecuada, todas las desviaciones del escenario principal que se consigan imaginar. Prestar atención a:
  - ➊ Caminos de éxito alternativos.
  - ➋ Comportamientos erróneos del actor primario y último.
  - ➌ Inactividad del actor primario y último.
  - ➍ Inactividad de un actor secundario con el que interactúa el sistema.
  - ➎ Caminos alternativos para *el sistema válida*.
  - ➏ Fallo interno, detectable y recuperable del sistema.
  - ➐ Fallos inesperados.
  - ➑ Fallos de rendimiento.
- ➋ Descartar aquellas extensiones cuyas condiciones sean indetectables.
- ➌ Fusionar aquellas extensiones cuyas condiciones y efecto sean equivalente.
- ➍ Escribir cada extensión.

# Identificación de Extensiones

- ❶ Identificar, mediante una técnica adecuada, todas las desviaciones del escenario principal que se consigan imaginar. Prestar atención a:
  - ❶ Caminos de éxito alternativos.
  - ❷ Comportamientos erróneos del actor primario y último.
  - ❸ Inactividad del actor primario y último.
  - ❹ Inactividad de un actor secundario con el que interactúa el sistema.
  - ❺ Caminos alternativos para *el sistema válida*.
  - ❻ Fallo interno, detectable y recuperable del sistema.
  - ❼ Fallos inesperados.
  - ❽ Fallos de rendimiento.
- ❷ Descartar aquellas extensiones cuyas condiciones sean indetectables.
- ❸ Fusionar aquellas extensiones cuyas condiciones y efecto sean equivalente.
- ❹ Escribir cada extensión.

# Identificación de Extensiones

- ❶ Identificar, mediante una técnica adecuada, todas las desviaciones del escenario principal que se consigan imaginar. Prestar atención a:
  - ❶ Caminos de éxito alternativos.
  - ❷ Comportamientos erróneos del actor primario y último.
  - ❸ Inactividad del actor primario y último.
  - ❹ Inactividad de un actor secundario con el que interactúa el sistema.
  - ❺ Caminos alternativos para *el sistema válida*.
  - ❻ Fallo interno, detectable y recuperable del sistema.
  - ❼ Fallos inesperados.
  - ❽ Fallos de rendimiento.
- ❷ Descartar aquellas extensiones cuyas condiciones sean indetectables.
- ❸ Fusionar aquellas extensiones cuyas condiciones y efecto sean equivalente.
- ❹ Escribir cada extensión.

# Formas de Terminación de una Extensión

- 1 El caso de uso sigue un camino de éxito alternativo.
- 2 La situación anómala se soluciona y el caso de uso prosigue normal.
- 3 Se da al usuario una nueva oportunidad y el caso de uso prosigue normal.
- 4 Se detecta un fallo insalvable y el caso de uso termina.

# Formas de Terminación de una Extensión

- 1 El caso de uso sigue un camino de éxito alternativo.
- 2 La situación anómala se soluciona y el caso de uso prosigue normal.
- 3 Se da al usuario una nueva oportunidad y el caso de uso prosigue normal.
- 4 Se detecta un fallo insalvable y el caso de uso termina.

# Formas de Terminación de una Extensión

- 1 El caso de uso sigue un camino de éxito alternativo.
- 2 La situación anómala se soluciona y el caso de uso prosigue normal.
- 3 Se da al usuario una nueva oportunidad y el caso de uso prosigue normal.
- 4 Se detecta un fallo insalvable y el caso de uso termina.



# Formas de Terminación de una Extensión

- ❶ El caso de uso sigue un camino de éxito alternativo.
- ❷ La situación anómala se soluciona y el caso de uso prosigue normal.
- ❸ Se da al usuario una nueva oportunidad y el caso de uso prosigue normal.
- ❹ Se detecta un fallo insalvable y el caso de uso termina.

# Reglas de Escritura de Extensiones

- 1 Hacer que la condición exprese *qué* se ha detectado.
- 2 Identar el cuerpo de la extensión.

# Reglas de Escritura de Extensiones

- 1 Hacer que la condición exprese *qué* se ha detectado.
- 2 Identar el cuerpo de la extensión.

# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Historia de Usuario (*User Story*)

## Historia de Usuario

Un *historia de usuario* describe una funcionalidad del sistema que posee valor para algún *stakeholder* del sistema. Una historia de usuario se compone de tres elementos:

**Conversación** Corresponde con las conversaciones (efímeras) mantenidas entre las diferentes personas involucradas en el desarrollo de la historia de usuario, tales como usuarios, clientes, testadores, programadores, etc. Se puede documentar parte de estas conversaciones.

**Carta** Cada historia de usuario se anota en una tarjeta física, que representa la historia de usuario.

**Checklist** Procedimiento para verificar que la historia de usuario ha sido completada.

# Historia de Usuario (*User Story*)

## Historia de Usuario

Un *historia de usuario* describe una funcionalidad del sistema que posee valor para algún *stakeholder* del sistema. Una historia de usuario se compone de tres elementos:

**Conversación** Corresponde con las conversaciones (efímeras) mantenidas entre las diferentes personas involucradas en el desarrollo de la historia de usuario, tales como usuarios, clientes, testadores, programadores, etc. Se puede documentar parte de estas conversaciones.

**Tarjeta** Cada historia de usuario se anota en una tarjeta física, que representa la historia de usuario.

**Confirmación** Procedimiento para verificar que la historia de usuario ha sido realizada.

# Historia de Usuario (*User Story*)

## Historia de Usuario

Un *historia de usuario* describe una funcionalidad del sistema que posee valor para algún *stakeholder* del sistema. Una historia de usuario se compone de tres elementos:

**Conversación** Corresponde con las conversaciones (efímeras) mantenidas entre las diferentes personas involucradas en el desarrollo de la historia de usuario, tales como usuarios, clientes, testadores, programadores, etc. Se puede documentar parte de estas conversaciones.

**Tarjeta** Cada historia de usuario se anota en una tarjeta física, que representa la historia de usuario.

**Confirmación** Procedimiento para verificar que la historia de usuario ha sido realizada.

# Historia de Usuario (*User Story*)

## Historia de Usuario

Un *historia de usuario* describe una funcionalidad del sistema que posee valor para algún *stakeholder* del sistema. Una historia de usuario se compone de tres elementos:

**Conversación** Corresponde con las conversaciones (efímeras) mantenidas entre las diferentes personas involucradas en el desarrollo de la historia de usuario, tales como usuarios, clientes, testadores, programadores, etc. Se puede documentar parte de estas conversaciones.

**Tarjeta** Cada historia de usuario se anota en una tarjeta física, que representa la historia de usuario.

**Confirmación** Procedimiento para verificar que la historia de usuario ha sido realizada.



# Historia de Usuario (*User Story*)

## Historia de Usuario

Un *historia de usuario* describe una funcionalidad del sistema que posee valor para algún *stakeholder* del sistema. Una historia de usuario se compone de tres elementos:

**Conversación** Corresponde con las conversaciones (efímeras) mantenidas entre las diferentes personas involucradas en el desarrollo de la historia de usuario, tales como usuarios, clientes, testadores, programadores, etc. Se puede documentar parte de estas conversaciones.

**Tarjeta** Cada historia de usuario se anota en una tarjeta física, que representa la historia de usuario.

**Confirmación** Procedimiento para verificar que la historia de usuario ha sido realizada.

# Propiedades Deseables de una Historia de Usuario

- 1 Independiente.
- 2 Abierta.
- 3 De valor para algún stakeholder.
- 4 Estimable.
- 5 Pequeñas.
- 6 Verificable (testable).

# Propiedades Deseables de una Historia de Usuario

- 1 Independiente.
- 2 Abierta.
- 3 De valor para algún stakeholder.
- 4 Estimable.
- 5 Pequeñas.
- 6 Verificable (testable).

# Propiedades Deseables de una Historia de Usuario

- 1 Independiente.
- 2 Abierta.
- 3 De valor para algún stakeholder.
- 4 Estimable.
- 5 Pequeñas.
- 6 Verificable (testable).

# Propiedades Deseables de una Historia de Usuario

- 1 Independiente.
- 2 Abierta.
- 3 De valor para algún stakeholder.
- 4 Estimable.
- 5 Pequeñas.
- 6 Verificable (testable).

# Propiedades Deseables de una Historia de Usuario

- 1 Independiente.
- 2 Abierta.
- 3 De valor para algún stakeholder.
- 4 Estimable.
- 5 Pequeñas.
- 6 Verificable (testable).

# Propiedades Deseables de una Historia de Usuario

- 1 Independiente.
- 2 Abierta.
- 3 De valor para algún stakeholder.
- 4 Estimable.
- 5 Pequeñas.
- 6 Verificable (testable).

# Historia Épica

## Historia Épica

Una historia épica (*epic*) es una historia que por su tamaño no puede ser desarrollada en un periodo corto de tiempo (15 días), por lo que ha de ser descompuesta en historias de usuario de menor tamaño.



# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias

# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias
  - Sumario
  - Referencias

# ¿Qué Tengo que Saber de Todo Esto?

- 1 Saber identificar, estructurar y especificar requisitos a diferentes niveles de abstracción.
- 2 Ser capaz de construir modelos orientados a objetivos utilizando los lenguajes KAOS y/o GRL.
- 3 Ser capaz de especificar escenarios de forma completa, incluyendo extensiones.
- 4 Ser capaz de especificar requisitos como historias de usuario.
- 5 Comprender el papel de los modelos orientados a la solución en Ingeniería de Requisitos y cómo se elaboran.

# ¿Qué Tengo que Saber de Todo Esto?

- 1 Saber identificar, estructurar y especificar requisitos a diferentes niveles de abstracción.
- 2 Ser capaz de construir modelos orientados a objetivos utilizando los lenguajes KAOS y/o GRL.
- 3 Ser capaz de especificar escenarios de forma completa, incluyendo extensiones.
- 4 Ser capaz de especificar requisitos como historias de usuario.
- 5 Comprender el papel de los modelos orientados a la solución en Ingeniería de Requisitos y cómo se elaboran.

# ¿Qué Tengo que Saber de Todo Esto?

- 1 Saber identificar, estructurar y especificar requisitos a diferentes niveles de abstracción.
- 2 Ser capaz de construir modelos orientados a objetivos utilizando los lenguajes KAOS y/o GRL.
- 3 Ser capaz de especificar escenarios de forma completa, incluyendo extensiones.
- 4 Ser capaz de especificar requisitos como historias de usuario.
- 5 Comprender el papel de los modelos orientados a la solución en Ingeniería de Requisitos y cómo se elaboran.

# ¿Qué Tengo que Saber de Todo Esto?

- 1 Saber identificar, estructurar y especificar requisitos a diferentes niveles de abstracción.
- 2 Ser capaz de construir modelos orientados a objetivos utilizando los lenguajes KAOS y/o GRL.
- 3 Ser capaz de especificar escenarios de forma completa, incluyendo extensiones.
- 4 Ser capaz de especificar requisitos como historias de usuario.
- 5 Comprender el papel de los modelos orientados a la solución en Ingeniería de Requisitos y cómo se elaboran.

# ¿Qué Tengo que Saber de Todo Esto?

- 1 Saber identificar, estructurar y especificar requisitos a diferentes niveles de abstracción.
- 2 Ser capaz de construir modelos orientados a objetivos utilizando los lenguajes KAOS y/o GRL.
- 3 Ser capaz de especificar escenarios de forma completa, incluyendo extensiones.
- 4 Ser capaz de especificar requisitos como historias de usuario.
- 5 Comprender el papel de los modelos orientados a la solución en Ingeniería de Requisitos y cómo se elaboran.

# Índice

- 1 Introducción
- 2 Modelos Orientados a la Solución del Problema
- 3 Jerarquías de Requisitos Funcionales
- 4 Modelado y Especificación de Objetivos
- 5 Modelado y Especificación de Escenarios
- 6 Historias de Usuario
- 7 Sumario y Referencias
  - Sumario
  - **Referencias**



# Referencias



Pohl, K. (2010).

*Requirements Engineering: Fundamentals, Principles and Techniques.*  
Springer.



van Lamsweerde, A. (2009).

*Requirements Engineering: From System Goals to UML Models to Software Specifications.*  
Wiley.