

**Interacción persona-ordenador**

# **Ingeniería de la interfaz**





# Objetivos

- ⌘ Conocer el proceso de diseño de sistemas interactivos
- ⌘ Apreciar la importancia de realizar un diseño centrado en el usuario
- ⌘ Presentar notaciones y métodos para el análisis de la interfaz de usuario



# Contenido

- ⌘ Introducción
- ⌘ Ciclo de vida de un sistema interactivo
- ⌘ El diseño centrado en el usuario
- ⌘ Prototipos
- ⌘ Análisis de tareas



# Introducción

- ⌘ Los sistemas interactivos se caracterizan por la importancia del diálogo con el usuario
- ⌘ La interfaz es por tanto una parte fundamental en el proceso de desarrollo y debe tenerse en cuenta desde el principio
- ⌘ Además, la interfaz determina en gran medida la percepción e impresión que el usuario posee de una aplicación
- ⌘ El usuario no está interesado en la estructura interna de una aplicación, sino en cómo usarla



# Introducción

- ⌘ **Conclusión:** no se puede realizar la especificación, diseñar las funciones y estructuras de datos y escribir el código y una vez casi terminado el proceso de desarrollo de la aplicación plantearse el diseño de la interfaz de usuario
- ⌘ De esta forma se obtienen interfaces muy dependientes del diseño de los datos y las funciones, **sin tener en cuenta al usuario** que va a utilizar esos datos y esas funciones



# Introducción

- ⌘ Una vez hecha la especificación, propuesto un diseño y desarrollado el código, es muy difícil cambiar las características de la interacción y presentación de la información, salvo pequeños detalles
- ⌘ Por tanto, **debemos empezar con una idea clara de cómo queremos la interfaz** y cómo serán las interacciones con el usuario para después desarrollar las especificaciones funcionales que sirvan de guía al diseño posterior

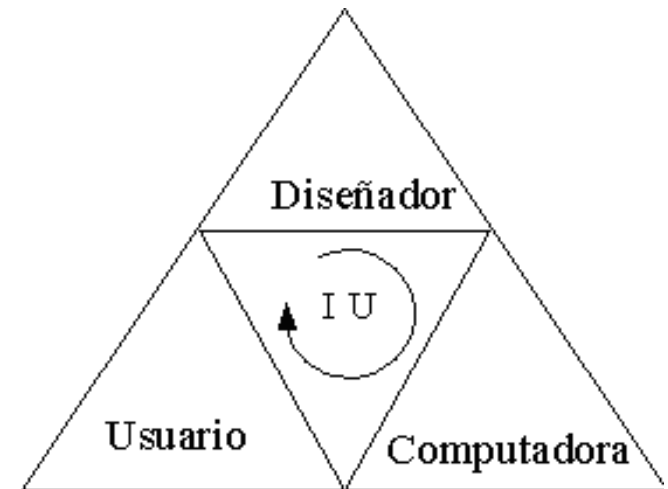


# Introducción

⌘ En el desarrollo de sistemas interactivos se pueden aplicar técnicas de Ingeniería del Software, pero modificando algunos aspectos de los métodos de diseño clásico para adaptarlos a estos sistemas

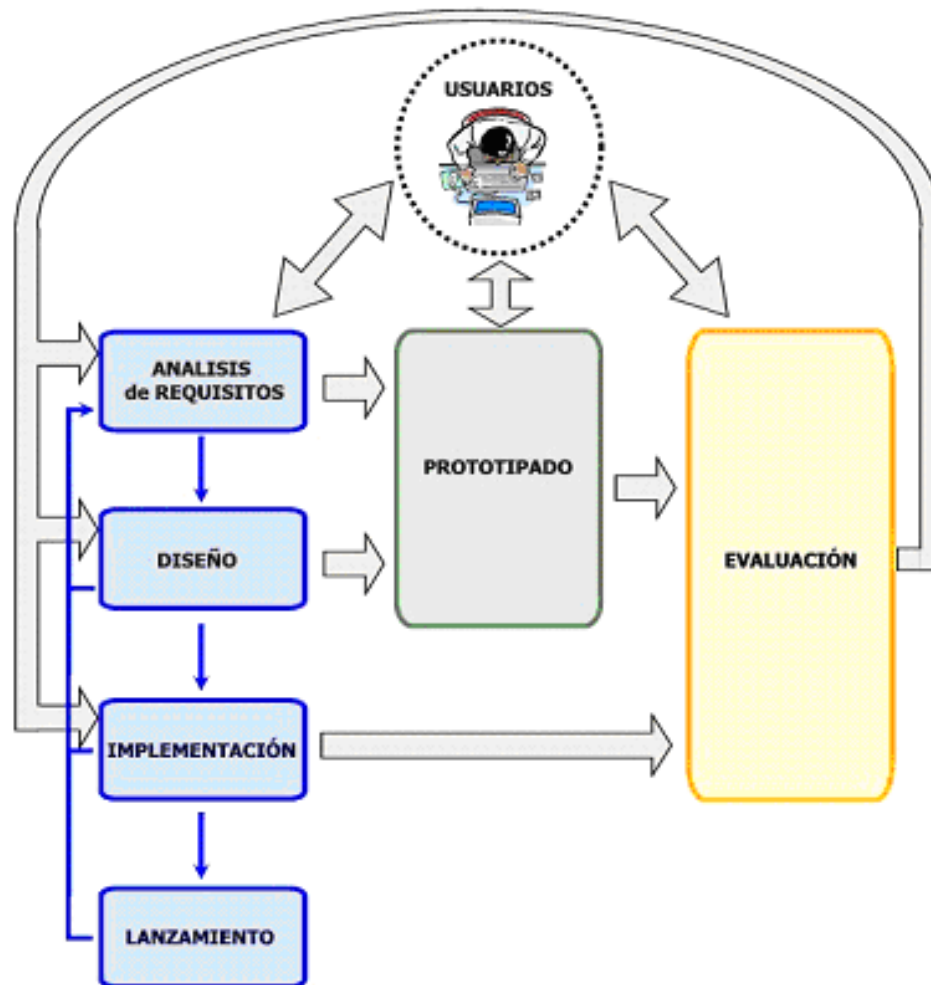
⌘ Aspectos a considerar:

- ☑ Captura de requisitos de interacción
- ☑ Análisis de tareas
- ☑ Realización de prototipos
- ☑ Evaluación





# Ciclo de vida







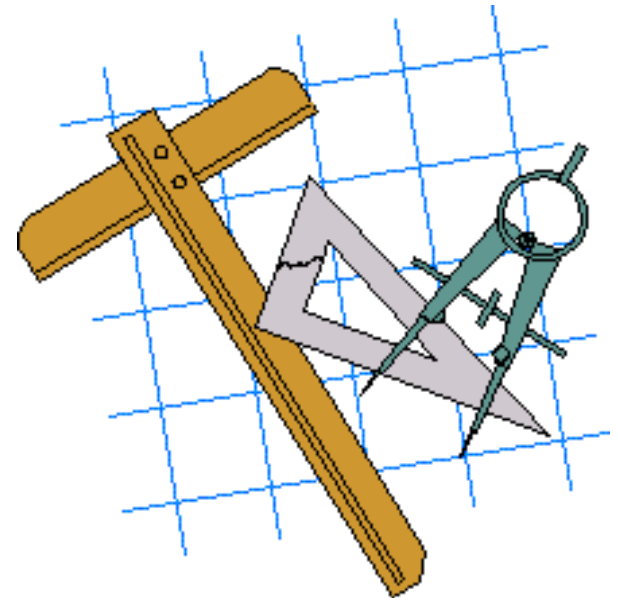
# Diseño centrado en el usuario

- ⌘ El proceso de diseño debe estar centrado en el usuario para recoger sus necesidades y mejorar su utilización
- ⌘ El objetivo del sistema interactivo es permitir al usuario conseguir un objetivo concreto en un dominio de aplicación
- ⌘ El diseño debe responder a las siguientes cuestiones:
  - ☑ Cómo debe ser desarrollado el sistema interactivo para asegurar la **usabilidad**
  - ☑ Cómo puede la usabilidad de un sistema interactivo ser **evaluada** o medida



# Prototipos

- ⌘ Son documentos, diseños o sistemas que simulan o tienen implementadas partes del sistema final
- ⌘ Son herramientas muy útiles para hacer participar al usuario en el diseño y poder evaluarlo ya en las primeras fases del desarrollo





**Prototipos**

# **Características**

- ⌘ Funcionan, no son ni una idea ni un dibujo
- ⌘ Tienen un tiempo de vida corto
- ⌘ Pueden servir para diferentes objetivos
- ⌘ Han de poder ser contruidos rápida y eficientemente



# Prototipos

## Tipos

### ⌘ Maqueta 'para tirar'

- ☑ Sirve para realizar una evaluación con el usuario y posteriormente se desecha

### ⌘ Incremental

- ☑ Se construye con componentes separados

### ⌘ Evolutivo

- ☑ Continúa utilizándose en un proceso evolutivo



## Prototipos

# Escenarios

- ⌘ Un escenario es “*una historia de ficción con representación de personajes, sucesos, productos y entornos*”
- ⌘ Ayuda al diseñador a explorar ideas y las ramificaciones de decisiones de diseño en situaciones concretas
- ⌘ “El uso de los escenarios nos permite definir y desarrollar conocimientos sobre el entorno del usuario y su espacio de trabajo” (*Bruce Toganizzini*)
- ⌘ Es interesante pensar en **varios escenarios** para reflejar las diferentes situaciones y puntos de vista
- ⌘ Es importante ser **consistente** con la representación para ver qué pasa en situaciones concretas



# Escenarios Tipos

## ⌘ Narrativa

- ☑ Historia completa de la interacción hecha con la existente o con un diseño nuevo

## ⌘ *Flowchart*

- ☑ Representación gráfica de las acciones y decisiones extraídas de la narrativa

## ⌘ Texto procedural

- ☑ Descripción paso a paso de las acciones del usuario y las respuestas del sistema

## ⌘ *Storyboard*

## ⌘ Prototipo de papel

## ⌘ Vídeo



**Escenarios**

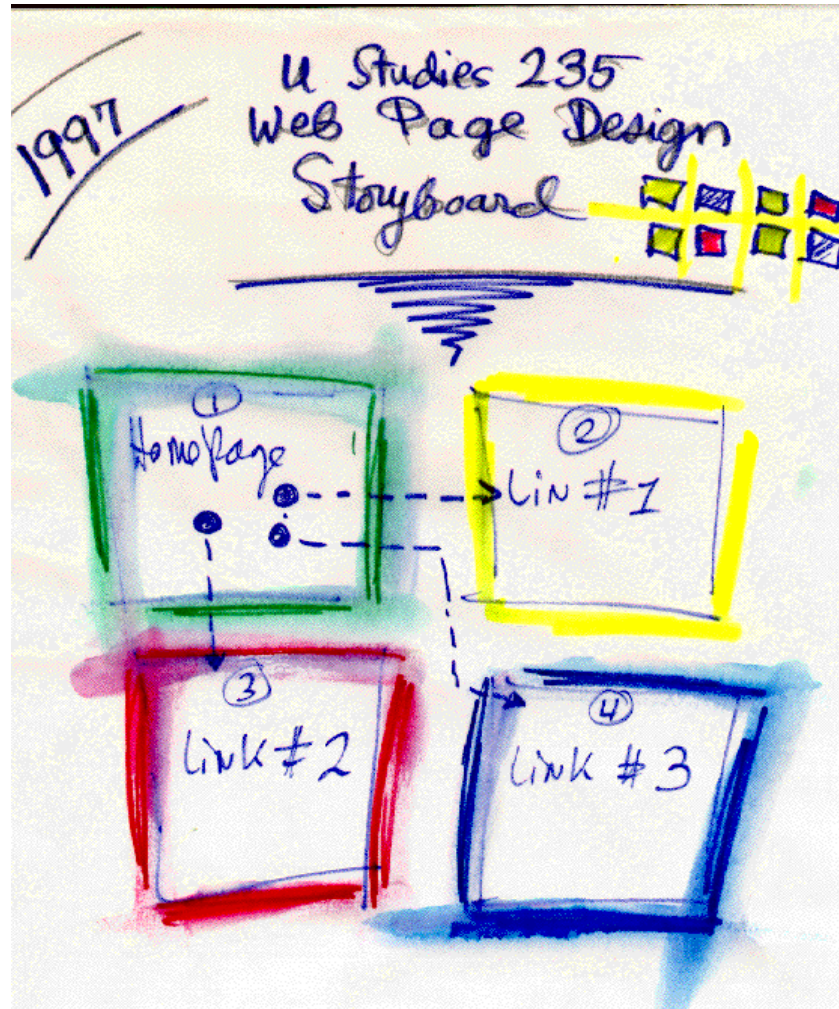
# ***Storyboard***

- ⌘ Es una narración gráfica de una historia en cuadros consecutivos
- ⌘ Podemos utilizar este concepto que se utiliza en el cine o el teatro para la realización de un escenario de interacción que puede ser evaluado con diferentes técnicas
- ⌘ El *storyboard* nos permite indicar los enlaces a diferentes páginas a partir de los resultados de las interacciones del usuario



Escenarios

# Storyboard









**Escenarios**

# **Prototipo de papel**

- ⌘ Este tipo de prototipo se basa en la utilización de papel, tijeras, lápiz o instrumentos que se puedan utilizar para describir un diseño en un papel
- ⌘ Este sistema nos permite una gran **velocidad** y **flexibilidad**





**Escenarios**

# **Prototipo de papel – cómo se hace**

- ⌘ Para poder simular las diferentes interacciones que vamos a realizar con el sistema, realizaremos una **hoja para cada uno de los diferentes escenarios** que vamos a tener como resultado de las diferentes interacciones que podemos realizar
- ⌘ Apilaremos estas hojas que nos permitirán simular la aplicación



## Escenarios

# Prototipo de papel – cómo se usa

- ⌘ Para utilizar el prototipo de papel nos situaremos en un **escenario de uso** de futuro en el que el diseñador actúa como coordinador
- ⌘ El prototipo será analizado por un **posible usuario** e intentará realizar algunas de las tareas que se pretende diseñar
- ⌘ En voz alta **se irán realizando las interacciones** y le iremos cambiando las hojas de papel en función de las interacciones que vaya realizando



## Escenarios

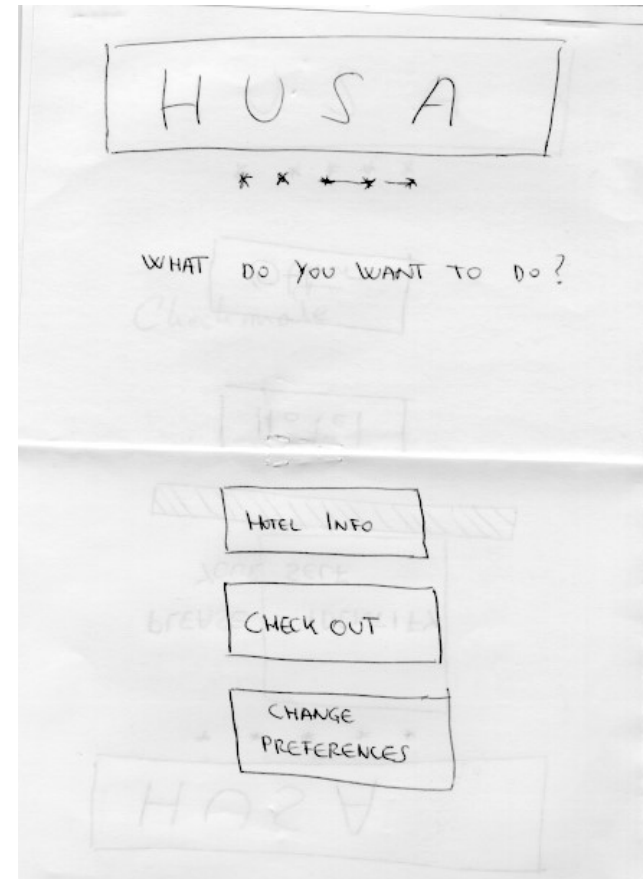
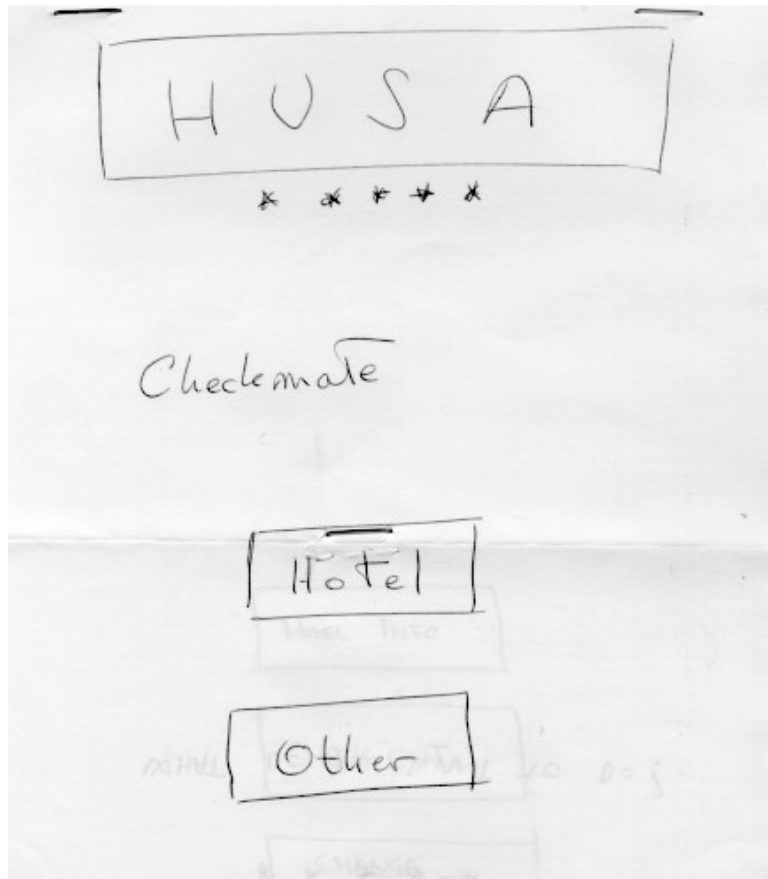
# Prototipo de papel – ventajas

- ⌘ El coste es muy reducido, necesitando únicamente los recursos humanos dedicados a la realización del prototipo
- ⌘ Los cambios se pueden realizar muy rápidamente y sobre la marcha. Si el diseño no funciona se pueden reescribir las hojas erróneas o rediseñarlas y volver a probar la tarea a realizar
- ⌘ Los usuarios o los actores se sienten más cómodos para poder realizar críticas al diseño debido a la sencillez del mismo por lo que no se sienten cohibidos a dar sus opiniones



**Escenarios**

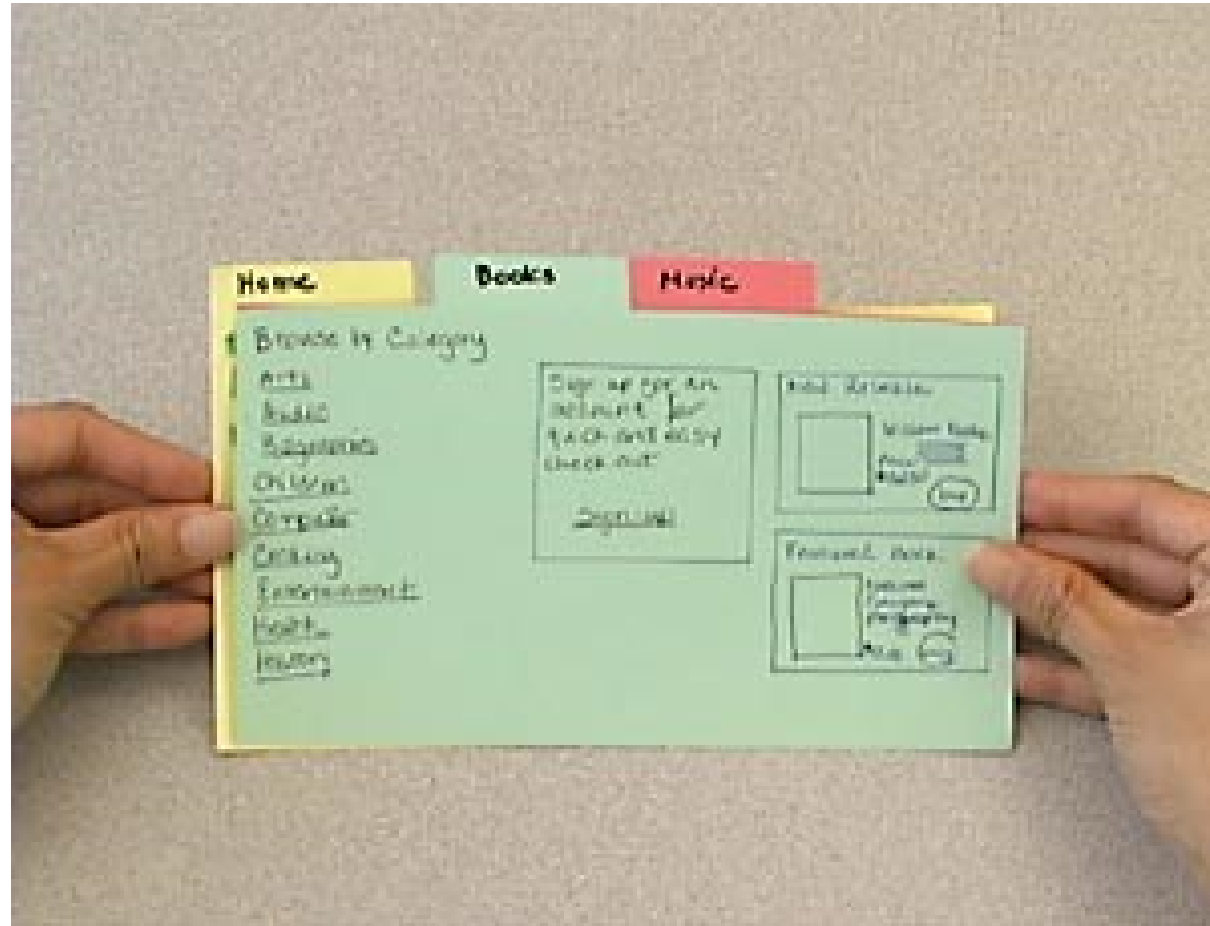
# Prototipo de papel – ejemplo





Escenarios

# Prototipo de papel – ejemplo





## Escenarios Vídeo

⌘ El vídeo permite rodar escenarios de futuro en los que se pueden realizar manipulaciones durante el postproceso para **simular características del diseño que aún no están disponibles**

⌘ Ejemplos:

☒ Poblado íbero de Els Vilars

☒ *Starfire*, de Sun Microsystems, sobre la interacción en 2004



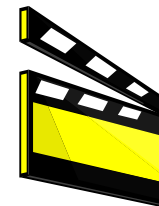


## Escenarios

# Starfire (1)



Vídeo 1



Vídeo 2



## Prototipos

# Problemas

- ⌘ Trabajar con prototipos requiere tiempo y experiencia en la planificación
- ⌘ Las características más importantes del sistema pueden ser las que se sacrifican en el prototipo (seguridad, fiabilidad)



# Análisis de tareas

- ⌘ Una de las premisas de cualquier aproximación con la que abordemos el diseño es la de *conocer al usuario* y por tanto cómo realiza las tareas
- ⌘ Esta información se recoge en la fase de **análisis de las tareas** con una notación que permita su formalización y estudio
- ⌘ **Tarea**: *Unidad significativa de trabajo en la actividad de una persona* (sobre una aplicación)
- ⌘ **Beneficios** del análisis de tareas:
  - ☑ Proporciona un diseño de la aplicación consistente con el modelo conceptual del usuario
  - ☑ Facilita el análisis y evaluación de usabilidad. Se puede predecir el rendimiento humano e identificar problemas de uso



# Análisis de tareas

- ⌘ El análisis de tareas consiste en el estudio de:
  - ☑ Información que necesita el usuario para realizar la tarea (qué hacer)
  - ☑ Terminología y símbolos del dominio del problema (elementos)
  - ☑ Descripción de cómo esas tareas se realizan actualmente (cómo)
- ⌘ Es el proceso de analizar la manera en que las personas realizan sus trabajos
  - ☑ Lo que hacen
  - ☑ Sobre qué cosas actúan
  - ☑ Qué necesitan saber



## Análisis de tareas

# Ejemplo

### ⌘ Análisis de tareas de un vídeo:

- ☒ ¿Qué quiere realizar el usuario?
- ☒ ¿Qué información se necesita?
- ☒ ¿Qué acciones debe llevar a cabo?





## Análisis de tareas

# Ejemplo

### ⌘ Objetivos del usuario:

- ☑ Ver un vídeo
- ☑ Grabar la telenovela de todas las tardes
- ☑ Grabar una película esta noche y no estoy en casa



## Análisis de tareas

# Ejemplo

### ⌘ Información requerida:

- ☑ Lista de programas
- ☑ Tiempo de inicio, duración, canal
- ☑ Día de la semana para la grabación



## **Análisis de tareas**

# **Ejemplo**

### ⌘ Acciones necesarias:

- ☑ Lista de programas (identificar el programa que se quiere grabar)
- ☑ Seleccionar la cinta (de duración adecuada)
- ☑ Iniciar el proceso de grabación (seleccionando ajustes adecuados)





## **Análisis de tareas**

# **Métodos**

### ⌘ Descomposición de tareas

- ☑ Ver el modo en el cual una tarea se puede descomponer en otras más simples

### ⌘ Análisis basado en conocimiento

- ☑ Identificar el conocimiento del usuario para llevar a cabo dicha tarea y cómo está organizado este conocimiento

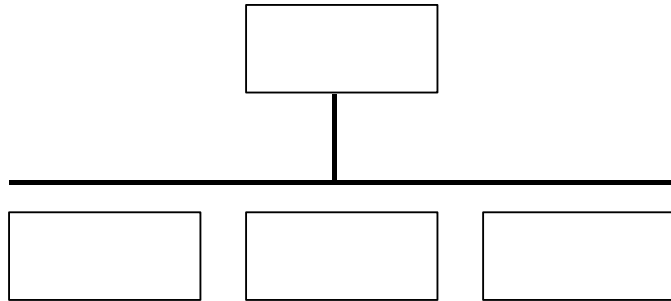
### ⌘ Análisis de relaciones entre entidades

- ☑ Aproximación orientada a objetos que enfatiza los actores y objetos, las relaciones entre los mismos y las acciones que pueden realizar

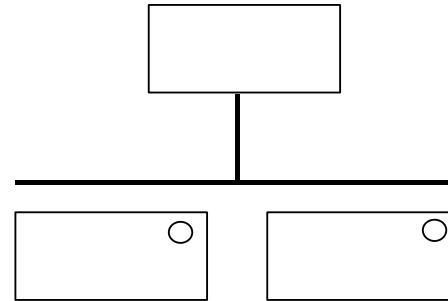


## Análisis de tareas

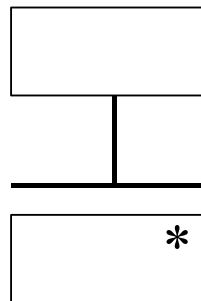
# Análisis jerárquico



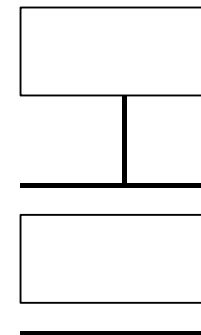
Secuencia de Tareas



Selección de Tareas



Iteración de Tareas

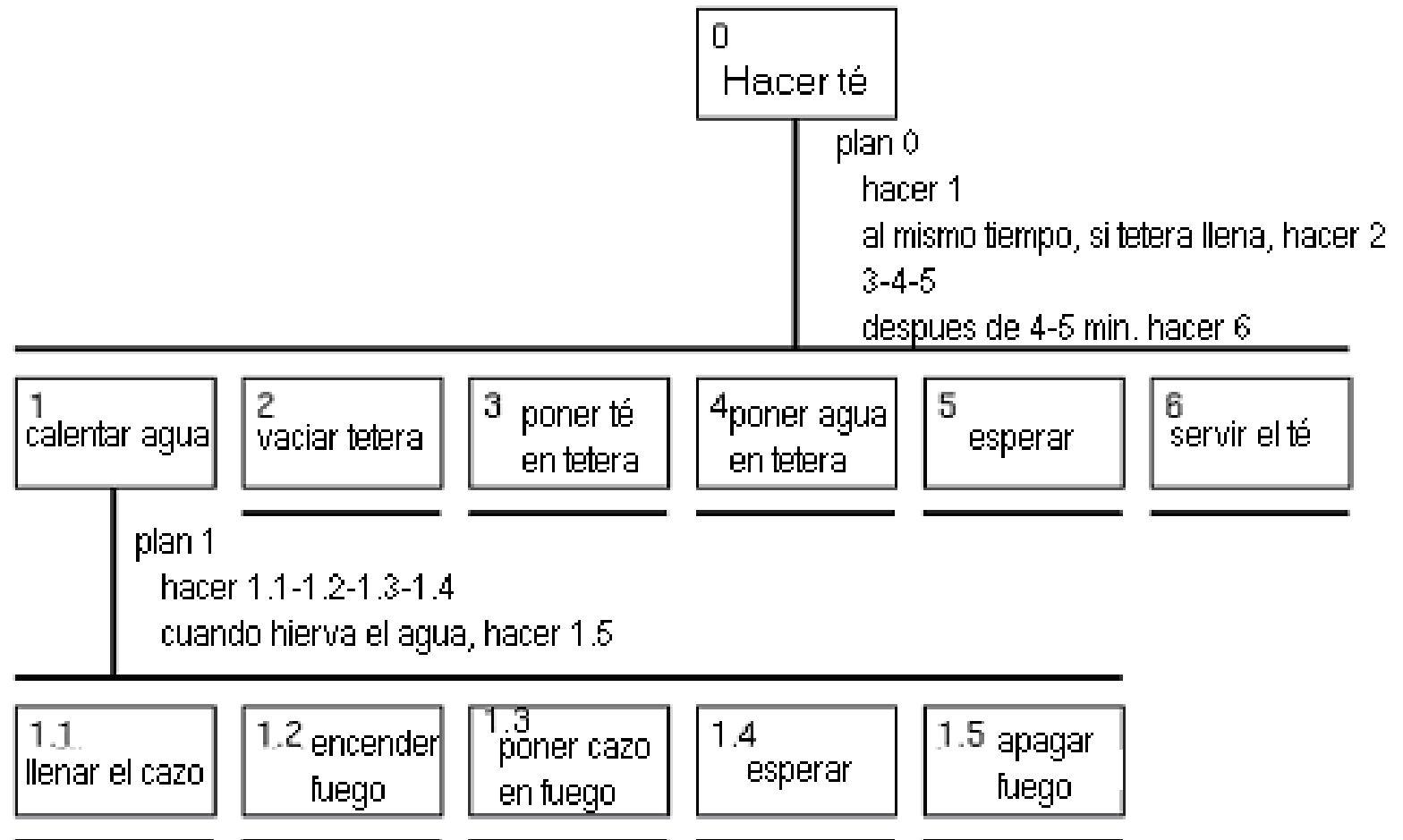


Tarea Unitaria



## Análisis de tareas

# Análisis jerárquico





## Análisis de tareas

# GOMS

- ⌘ Familia de técnicas propuesta por Card, Moran, y Newell (1983) para modelar y describir las prestaciones de las tareas desde el punto de vista humano
- ⌘ **GOMS** es un acrónimo que significa Objetivos (*Goals*), Operadores (*Operators*), Métodos (*Methods*) y Reglas de selección (*Selection rules*)



# Análisis de tareas

# **GOMS**

## ⌘ Objetivos

- ☑️ Objetivos del usuario, describen lo que pretende conseguir

## ⌘ Operadores

- ☑️ Acciones básicas que se deben llevar a cabo para utilizar el sistema

## ⌘ Métodos

- ☑️ Existen diferentes alternativas para conseguir un objetivo.  
P.ej. una ventana se puede cerrar mediante una combinación de teclas (Alt-F4) o con el ratón (Archivo-cerrar)

## ⌘ Reglas de selección

- ☑️ Elección entre posibles alternativas para alcanzar un objetivo



## Análisis de tareas

# GOMS

### ⌘ ***Ejemplo: Cerrar ventana***

⌘ GOAL: CERRAR-VENTANA  
[select GOAL: USAR-MÉTODO-RATÓN  
MOVER-RATÓN-A-MENÚ-VENTANA  
ABRIR-MENÚ  
CLICK-SOBRE-OPCIÓN-CERRAR  
GOAL: USAR-MÉTODO-TECLADO  
PULSAR-TECLAS-ALT-F4  
GOAL: USAR-MÉTODO-CERRAR-APLICACIÓN  
PULSAR CONTROL-ALT-DEL  
SELECCIONAR CERRAR-APLICACIÓN]

Rule 1: IF (USUARIO-EXPERTO) USAR-MÉTODO-TECLADO  
ELSE USAR-MÉTODO-RATÓN

Rule 2: USAR-MÉTODO-CERRAR-APLICACIÓN si se queda bloqueado el sistema



## Análisis de tareas

# Diálogo

- ⌘ El diálogo es el proceso de comunicación entre dos o más participantes
- ⌘ En el diseño de interfaces de usuario, el diálogo representa la estructura de la conversación entre el usuario y el ordenador



**Notaciones para el diálogo**

## ***User Action Notation***

- ⌘ UAN es una especificación mediante un lenguaje para la descripción de las tareas del usuario
- ⌘ Una especificación en UAN se realiza en una tabla dividida en 3 columnas:
  - ☒ acciones del usuario
  - ☒ realimentación de la interfaz
  - ☒ estado de la interfaz





## Notaciones para el diálogo

# UAN

⌘ icon!

☑ Respuesta del sistema: iluminar el icono

⌘ icon-!

☑ Dejar de iluminar el objeto icono

⌘ icon >~

☑ Movimiento de arrastre del objeto icono



# Notaciones para el diálogo **UAN**

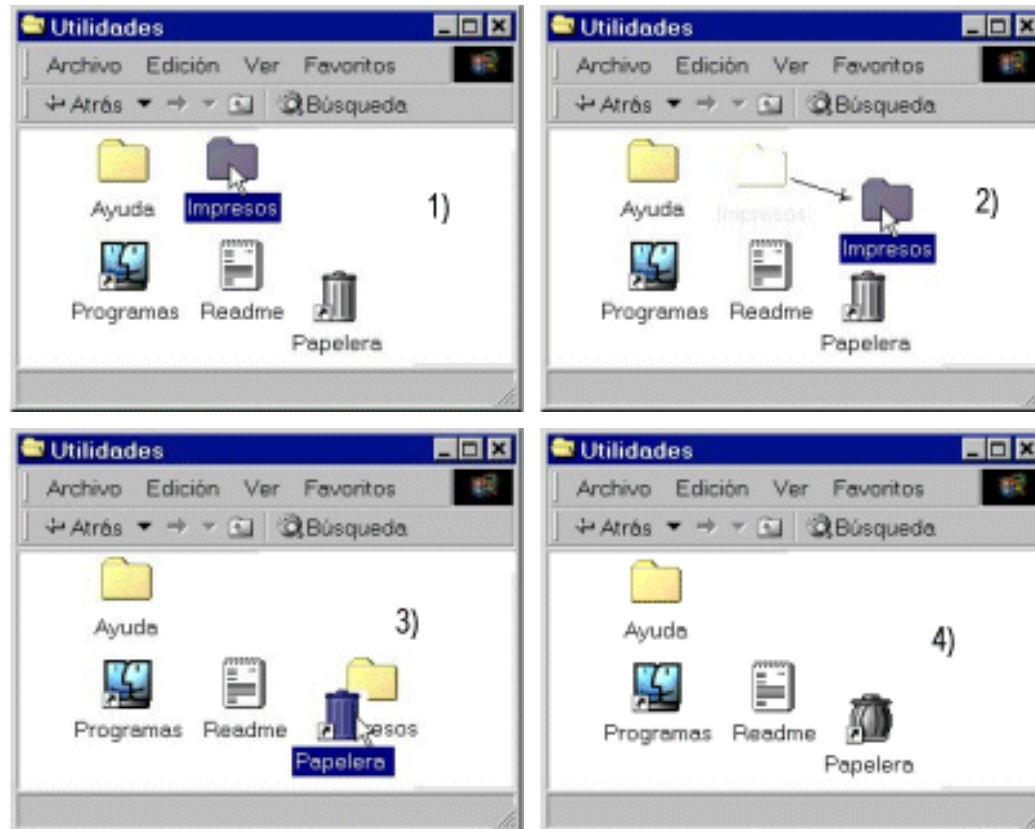
Ejemplo: Tarea "borrar un fichero enviándolo a la papelera de reciclaje"

	UAN	Realimentación	Estado de la interfaz
1)	~[file] Mv	File!, forall(file!): file-!	Selected = file
2)	~[x,y]*	Outline(file) > ~	
3)	~[trash]	Outline(file) > ~	trash!
4)	M^	Delete(file), trash!!	Selected = null



# Notaciones para el diálogo UAN

Ejemplo: Tarea "borrar un fichero enviándolo a la papelera de reciclaje"

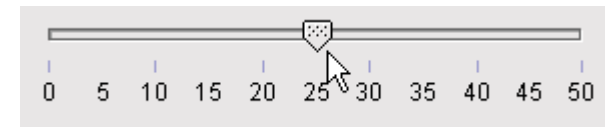
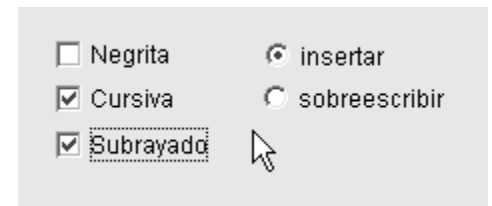




**Análisis de tareas**

# **Implementación**

- ⌘ Una vez modeladas las tareas debe obtenerse una implementación correcta de las mismas
- ⌘ Para ello hay que tener en cuenta varios factores:
  - ⏏ Tipos de interacción
    - ⏏ Posicionamiento, valor, texto, selección, arrastre
  - ⏏ Principios, guías de estilo, estándares
  - ⏏ Gestión de entradas del usuario
    - ⏏ Petición, muestreo, evento
  - ⏏ Diseño de la presentación
  - ⏏ Gestión de errores





# Conclusiones

- ⌘ El diseño de la interfaz es parte fundamental del proceso de desarrollo del software y debe ser considerado desde el principio
- ⌘ El usuario debe tomar parte en el diseño y no ser mero espectador
- ⌘ Existen metodologías y notaciones para el diseño que deben ser utilizadas
- ⌘ La evaluación del diseño tiene una gran importancia