

# DISEÑO CENTRADO EN EL USUARIO

# DESARROLLO

- 1. Introducción**
2. Modelo conceptual
3. Análisis de tareas: HTA
4. Modelo objeto-acción

# INTRODUCCIÓN

- Los sistemas interactivos se caracterizan por la importancia del diálogo con el usuario.
- La interfaz resulta fundamental y debe tenerse en cuenta desde el principio.
- 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

- 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, **se debe empezar con una idea clara de cómo se desea la interfaz** y cómo serán las interacciones con el usuario para después desarrollar las especificaciones funcionales que sirvan de guía al diseño posterior.
- 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.

# INTRODUCCIÓN

- El diseño de un sistema interactivo debe satisfacer las demandas de lo usuarios.
- Para lograr un buen diseño se deben analizar:
  - Las características del usuario: físicas, cognitivas, culturales, sociales...
  - Las actividades que realiza y cómo. Lograr objetivos implica llevar a cabo una serie de **TAREAS** en una secuencia dada.
  - El escenario (físico y social) dónde realiza su actividad.
- El diseño se debe basar en criterios consistentes basados en la experiencia y no en juicios intuitivos: principios generales de diseño, guías, estándares...

# DESARROLLO

1. Introducción
- 2. Modelo conceptual**
3. Análisis de tareas: HTA
4. Modelo objeto-acción

# MODELO CONCEPTUAL

- **Modelo conceptual:** Abstracción externa que describe el conocimiento que debe poseer un usuario acerca de un sistema mediante diagramas y descripciones. Se usan métodos formales y se pretende lograr **completitud**, **consistencia** y **exactitud**.
- **Modelo mental:** Abstracción del conocimiento interno que posee el usuario. Da una medida real de lo que piensa y conoce acerca del sistema, guía sus intenciones para realizar una tarea y puede modificarse con la interacción.

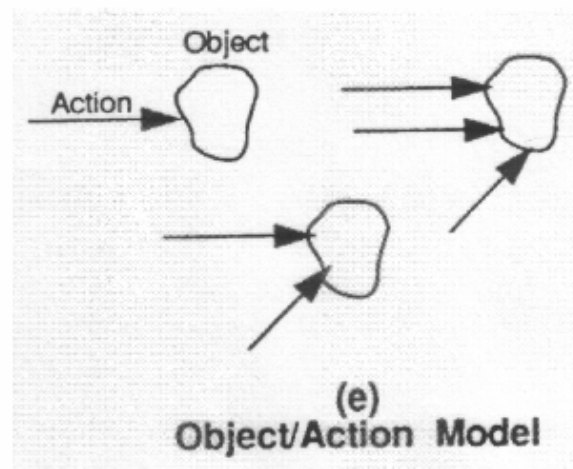
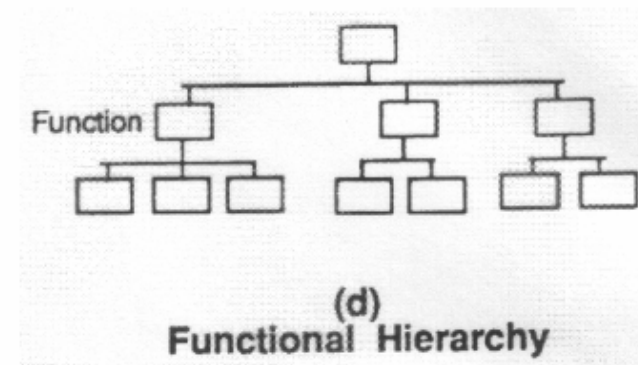
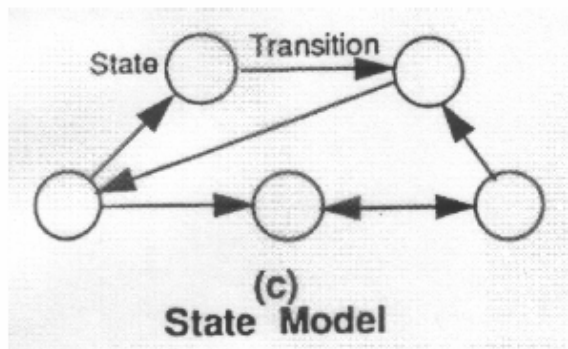
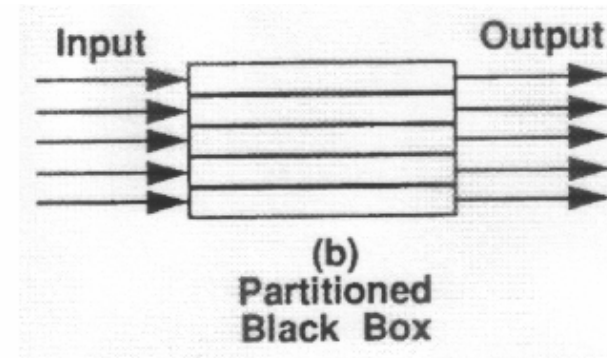
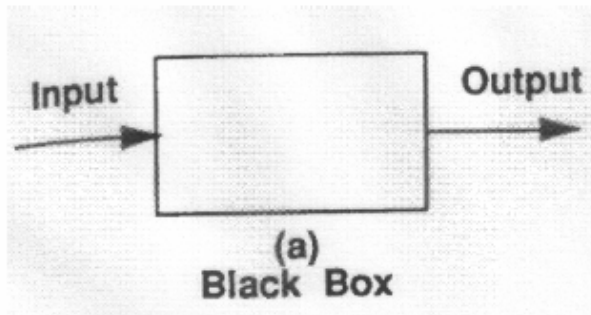
# MODELO CONCEPTUAL

- Informa al usuario de **qué** hace el sistema y los mecanismos para llevarlo a cabo.
- Debe favorecer el aprendizaje del sistema y el establecimiento de estrategias para resolver problemas por parte del usuario.
- Algunos principios en los que basarse:
  - *Asimilable*: usando conceptos familiares.
  - *Consistente*: usando coherencia.
  - Simple: al alcance de un usuario medio.



# MODELO CONCEPTUAL

- El modelo conceptual resulta clave en el diseño.
- Modelos de descripción conceptual:
  - **Caja negra:** el usuario desconoce por completo el funcionamiento interno del sistema.
  - **Funcional jerárquico:** se reduce la complejidad del sistema agrupando sus funciones en jerarquías y usando técnicas de partición en el dominio del problema.
  - **Basado en estados:** El sistema como conjunto de estados. Las transiciones son provocadas por eventos claramente definidos y el usuario observa esos cambios.
  - **Basado en objetos y acciones:** Trabajo directo con entidades (físicas o abstractas) sobre las que se realizan acciones. El usuario conoce los objetos, sus atributos y acciones.



# DESARROLLO

1. Introducción
2. Modelo conceptual
- 3. Análisis de tareas: HTA**
4. Modelo objeto-acción

# ANÁLISIS DE TAREAS

- Parte del conocimiento del usuario y de las actividades o tareas que realiza.
- La información se recoge en la fase de **análisis de las tareas** con una notación que permita su formalización.
- **Tarea**: Unidad significativa de trabajo en la actividad de una persona (sobre una aplicación) para conseguir un objetivo. Puede descomponerse en acciones.
- **Acción**: Cada uno de los pasos para realizar una tarea.
- **Objetivo**: El estado que el usuario quiere alcanzar.
- Beneficios del análisis de tareas:
  - Proporciona un diseño de la aplicación consistente con el modelo mental 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).
- **DEF:** Es el proceso de analizar la manera en que las personas realizan sus trabajos
  - Lo que hacen.
  - Sobre qué cosas actúan.
  - Qué necesitan saber.

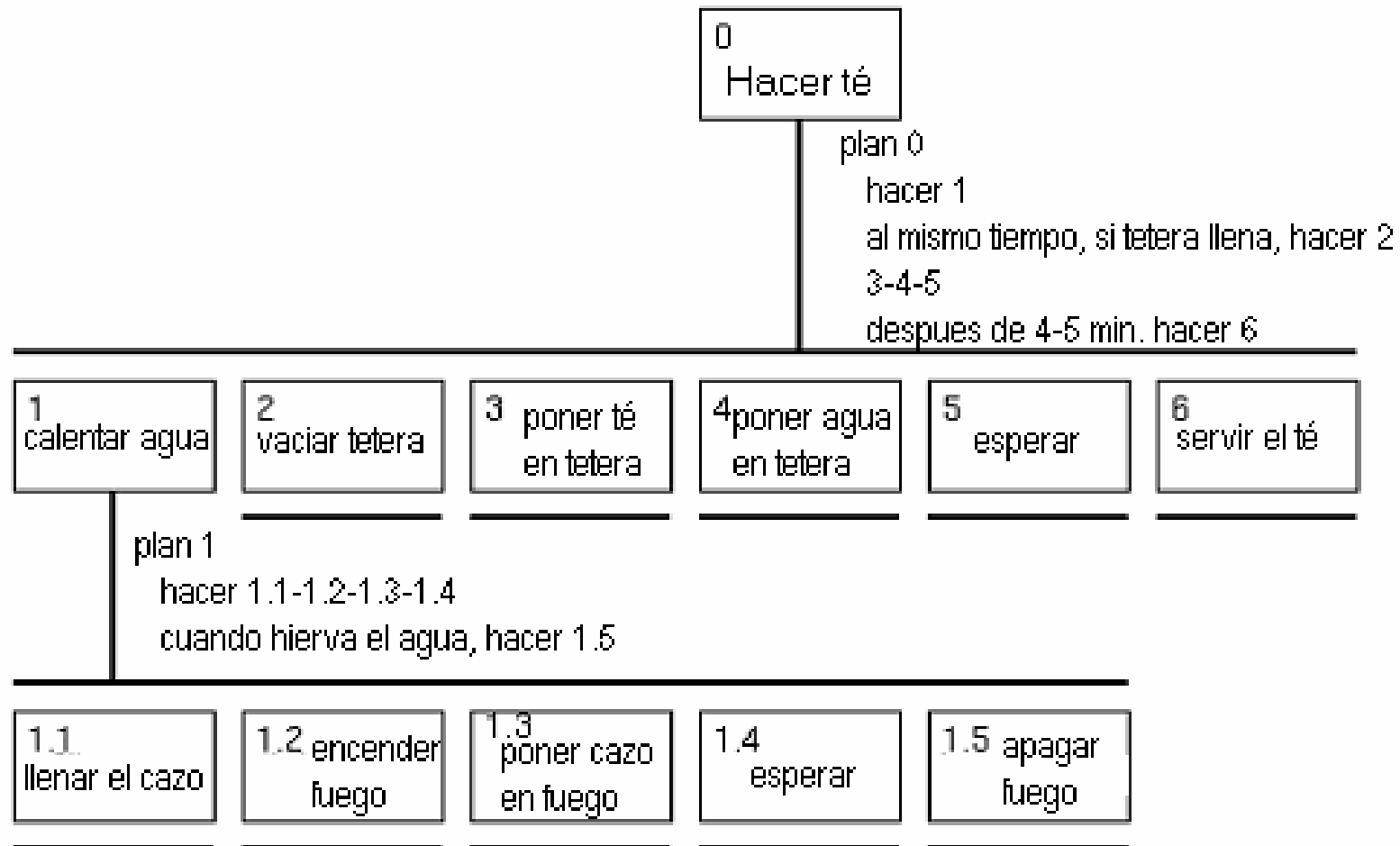
# ANÁLISIS DE TAREAS

- El resultado del análisis es una lista de tareas relevantes con algún tipo de información adicional (atributos, restricciones...). Algunos conceptos relevantes:
  - Modelo de diálogo persona-ordenador.
  - Modelo de tareas.
  - Dominio del sistema.
  - Modelo de usuarios.
  - Propiedades del sistema (seguridad, robustez...)
- Métodos:
  - Descomposición de tareas: implica jerarquía de tareas.
  - 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 JERÁRQUICO DE TAREAS (HTA)

- HTA describe tareas en términos de operaciones y planes.
- **Operación:** actividades a realizar para alcanzar un objetivo. Se descomponen jerárquicamente.
- **Plan:** conjunto de condiciones que se tienen que dar cuando se realiza una operación.
- Cabe hablar de cuatro tipos de ramificaciones:
  - **SECUENCIA:** Conjunto ordenado temporalmente de una secuencia de tareas.
  - **SELECCIÓN:** Se elige una tarea de un conjunto.
  - **ITERACIÓN:** Repetición de un conjunto de tareas.
  - **TAREA UNITARIA:** Actividad indivisible.

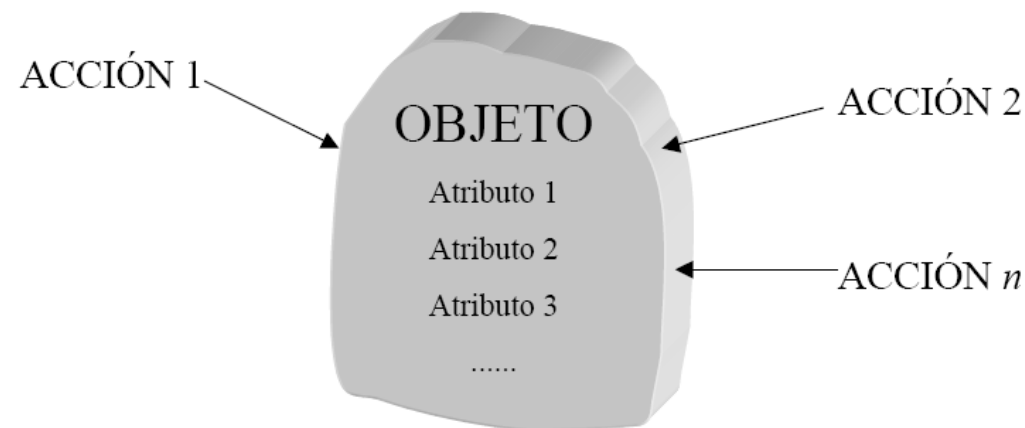
# HTA - EJEMPLO





# DESARROLLO

1. Introducción
2. Modelo conceptual
3. Análisis de tareas: HTA
- 4. Modelo objeto-acción**



# MODELO OBJETO-ACCIÓN

- Se trabaja con objetos físicos o conceptuales sobre los que podemos ejecutar acciones.
- No se hace ningún tipo de indicación sobre la forma en la que se van a realizar las acciones o tareas.
- La gran mayoría del software más reciente desarrollado para ordenadores personales se apoya en este modelo.
- Aportaciones del modelo objeto-acción
  - Representación visual del sistema, sencilla y fácil de aprender y manipular.
  - Obtiene modelos sencillos de sistemas complejos, con un alto grado de flexibilidad para mejoras y cambios futuros.
- Modelo de diálogo:
  - Conjunto de objetos que representan elementos de información.
  - Acciones que el usuario puede ejecutar sobre los objetos.

# MODELO OBJETO-ACCIÓN

- **Lenguajes de órdenes.** Para manipular objetos:
  - Selección/de-selección de un objeto.
  - Búsqueda/Identificación de un objeto.
  - Creación/eliminación de objetos.
  - Mover/copiar objetos.
  - Obtener/cambiar valores de los atributos del objeto.
  - Visualización del objeto.
- **Diálogos modales.** Un modo es un estado (o conjunto de estados) en el que se puede realizar un conjunto de posibles tareas. Ejemplos de diálogos modales:
  - Estado en el que se permite que un conjunto de órdenes se pueda aplicar únicamente al objeto/s seleccionado/s.
  - Estado en el cual se debe completar un cuadro de diálogo antes de hacer otra operación (ventana modal).
  - Estado en el cual se usa un programa externo para modificar un objeto (gráfico, diagrama, etc.).

# 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.

