



# Tema 8. Reutilización y Reingeniería

## MODELADO DEL SOFTWARE

Raquel Martínez España

Escuela Politécnica

# Reutilización: ¿Qué es?

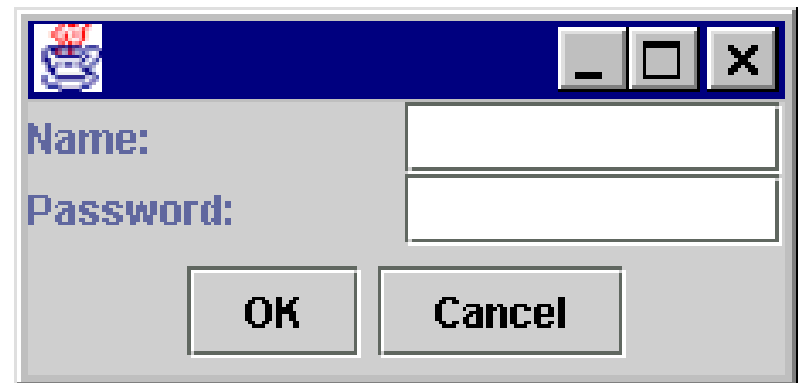
- Literalmente: volver a utilizar
- Actividad o proceso que permite desarrollar un sistema a partir de algo procedente de algún esfuerzo de desarrollo anterior
- ¿Qué es lo que hay que reutilizar?
- ¿Qué proceso es el que hay que seguir?
- ¿Cómo se organizan los componentes?

# Dificultades

- No existen planes de reutilización cuando se aborda un proyecto
- Los programadores no emplean las herramientas ni las técnicas
- Falta información
- Resistencia al cambio
- Los clientes no siempre quieren afrontar el cambio

# Reutilización: un ejemplo

- Para que un componente sea reutilizable tiene que ser lo más general posible (*pueda emplearse en una aplicación distinta de la actual*)
- ¿ Cuantas características configurables ?

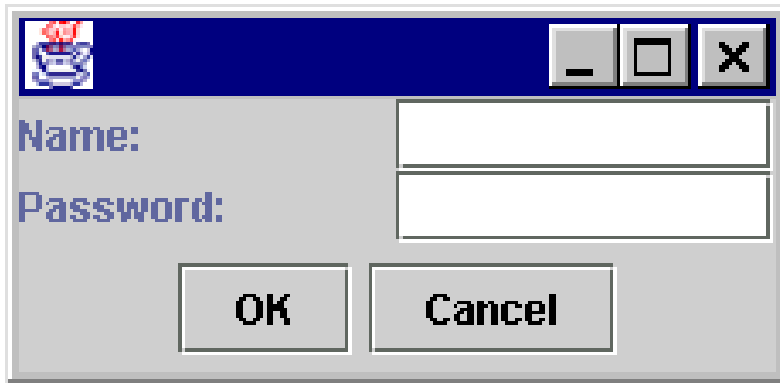


A screenshot of a Windows-style dialog box. The title bar is blue with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main area has a light gray background. It contains two labels, "Name:" and "Password:", in a blue font. To the right of each label is a white text input field. At the bottom of the dialog are two buttons: "OK" and "Cancel".

# Reutilización: un ejemplo

## Directamente con la GUI

- Propiedades de los elementos: *posición, color, valores concretos, visibilidad.*
- Propiedades de la ventana: *título, botones de la barra, maximizable, ....*
- Colores de fondo
- Secuencia de foco ...



## Directamente con el funcionamiento

- Fuente de datos: *base de datos, fichero, via red.*
- Acciones concretas en cada uno de los botones
- Acción en caso de password correcto e incorrecto
- Eventos de ratón y de teclado
- ...

# ¿Qué podemos reutilizar?

- Planes de proyecto
- Estimaciones de coste
- Arquitectura de datos
- Especificaciones y modelos de requisitos
- Diseños
- Código fuente
- Documentación de usuario y técnica
- Interfaces humanas
- Datos
- Casos de prueba

# Ingeniería de dominio

- Efectúa el trabajo necesario para establecer un conjunto de elementos software que sean reutilizables
- *Identificar, construir, catalogar y diseminar un conjunto de elementos software que tienen aplicación dentro de un determinado dominio de aplicación*

# Ingeniería del dominio

- Análisis del dominio
  - Determinar un conjunto de características que permitan clasificar a los dominios de aplicación.
  - Comprobar si esas características están presentes en los componentes de mi catálogo y así establecer qué elementos puedo reutilizar.
- Construcción de componentes
  - Identificar las estructuras de datos estándar
  - Identificar los protocolos de interfaz
  - Construir plantillas de programa



# Clasificación y recuperación de componentes

- Describir los componentes en términos no ambiguos y fáciles de clasificar
- Modelo 3C (Tracz)
  - Concepto: *descripción de lo que hace*
  - Contenido: *como se consigue el concepto*
  - Contexto: *dominio de aplicabilidad del componente*

# Patrones de diseño

- *Descripción de un problema que se repite a menudo y la solución general del mismo. De forma que puede emplearse esta solución una y otra vez.*

- Reutilización del diseño

- Catálogo de patrones:

Design Patterns: *Elements of Reusable Object.Oriented Software*

# Elementos principales

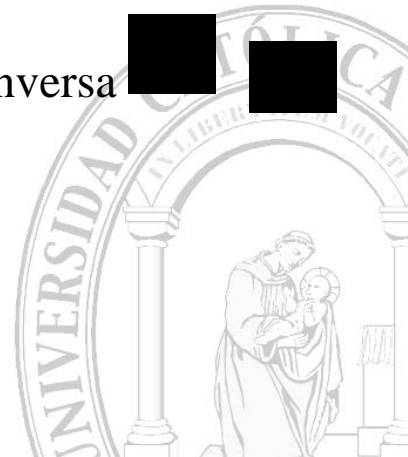
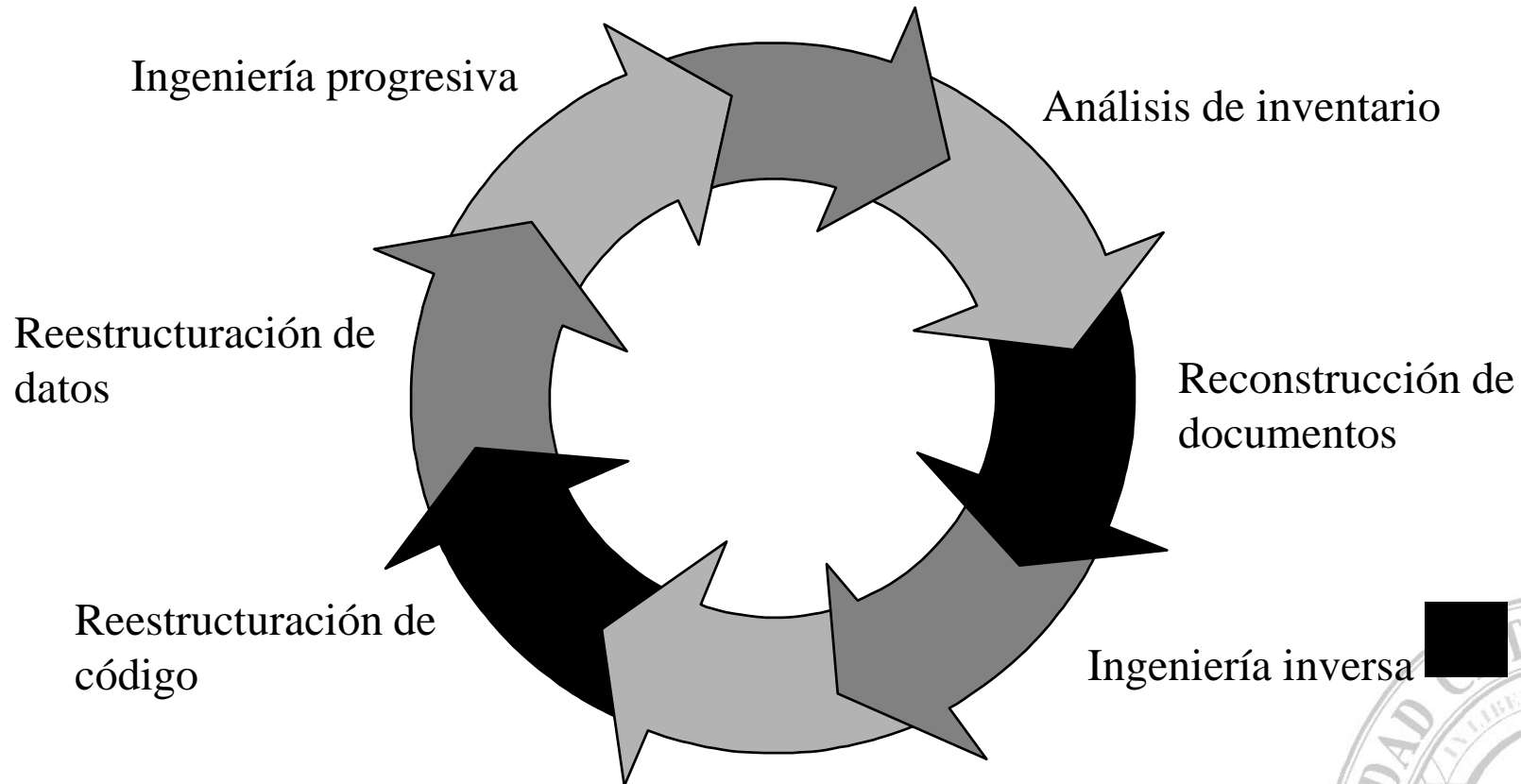
- **Nombre:** descripción del problema en una o dos palabras a lo sumo
- **Problema:** cuando aplicar el patrón (*contexto*)
- **Solución:** elementos que forman parte del patrón, relaciones entre ellos, responsabilidades y colaboradores. (*concepto y contenido*)
- **Consecuencias:** resultado de aplicar el patrón (*contexto*)

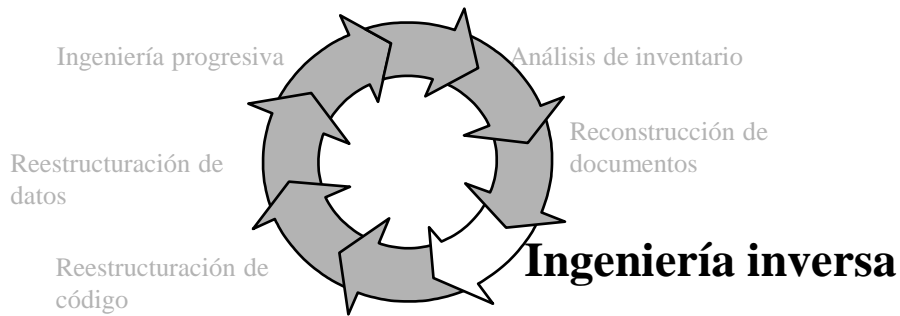
# Reingeniería: ¿Qué es?

- Mecanismo para evaluar, controlar y realizar modificaciones sobre aplicaciones “veteranas”
- Tarea de reconstrucción que requiere tiempo, dinero y recursos.
- Similar a la reconstrucción de una casa *Inspeccionar la casa, verificar la estructura, entender la arquitectura original, utilizar los materiales más modernos, reconstruir con calidad.*

# Proceso de reingeniería

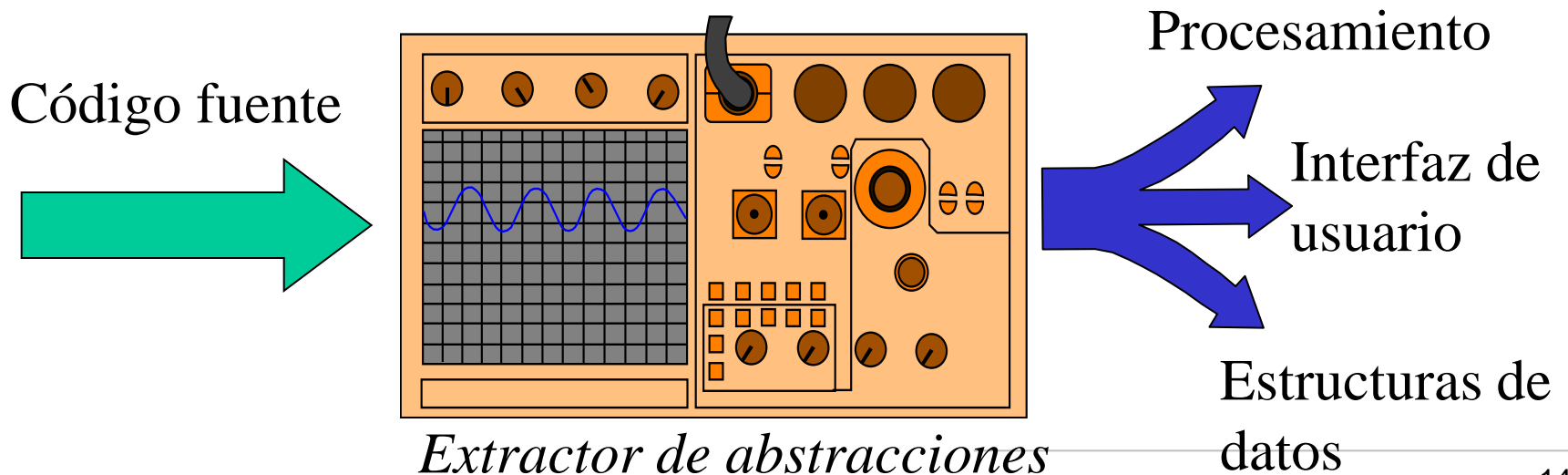
---

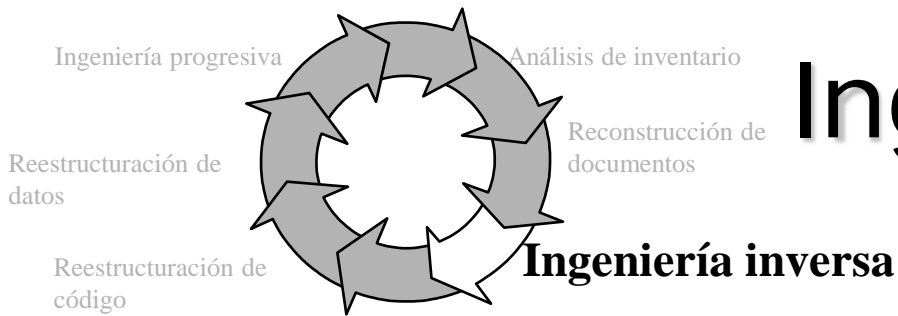




# Ingeniería Inversa

- Proceso de análisis de un programa para crear una representación del mismo con un nivel de abstracción mayor que el código fuente

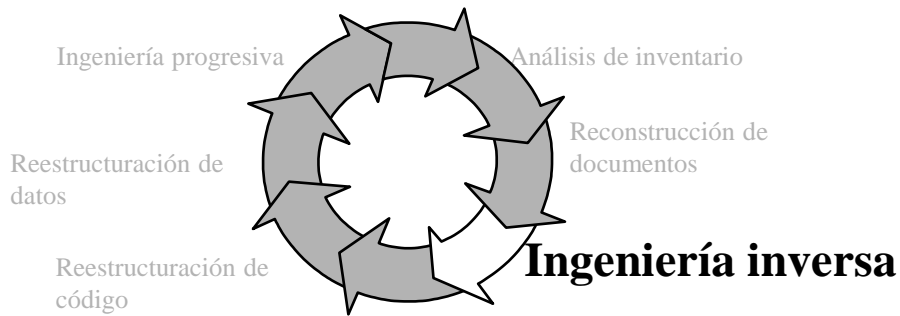




# Ingeniería Inversa: Procesamiento

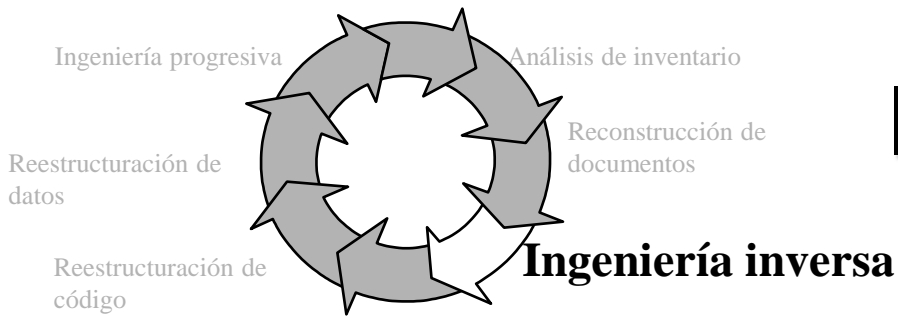
- Comprender la funcionalidad general de la aplicación
- Revisar si el código se corresponde con la especificación.
- Analizar los módulos de los programas
  - Sección de preparación de datos
  - Sección de operación de datos
  - Sección de exportación de datos

# Ingeniería Inversa: Datos



- Estructuras internas (programa)
  - Definir tipos abstractos para determinar las clases de objetos
- Estructuras externas (Bases de datos)
  - Estudiar la estructura actual y determinar si es adecuado construir una nueva base de datos

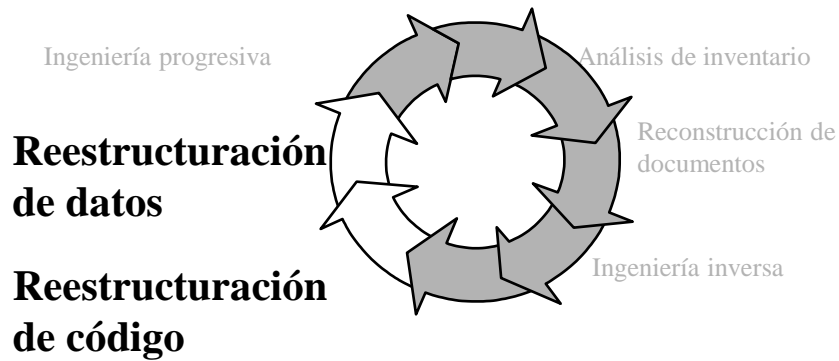




# Ingeniería Inversa: GUI

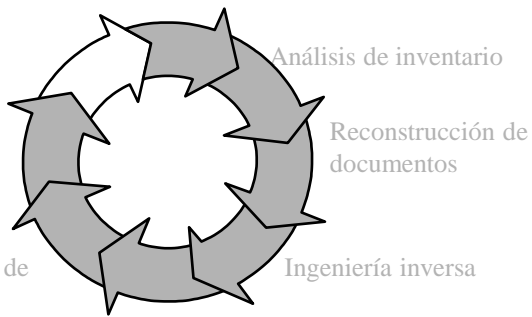
- Determinar las acciones básicas que debe procesar la interfaz (teclado y ratón).
- Describir la respuesta del sistema a estas acciones.
- Determinar que interfaces sustituir y cuales mantener

Diagrama de  
Estados /Actividad



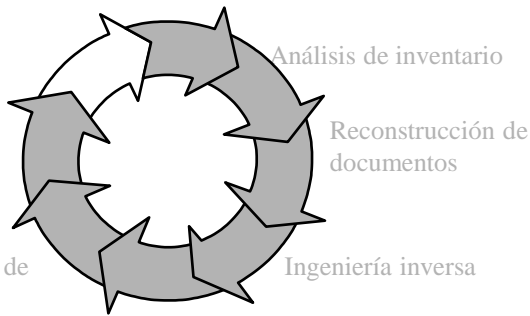
# Reestructuración

- Beneficios
  - Mayor calidad, mejor documentación y menor complejidad
  - Mejora el aprendizaje y aumenta la productividad (más sencillo de comprobar y depurar)
  - Reduce el esfuerzo en el mantenimiento
- Reestructuración de código
- Reestructuración de los datos: *análisis de datos, rediseño de datos (estandarización y/o racionalización )*



# Ingeniería Progresiva: Entornos C/S

- Identificar los elementos que deben estar en el cliente y en el servidor
- Necesario un proceso de ingeniería inversa y reestructuración para adaptar la base de datos a la nueva situación: transacciones atómicas, acceso restringido a usuarios.
- Adaptación al modelo de 3 capas.



# Ingeniería Progresiva: GUI

- Proceso de cambio a una nueva GUI
  - Comprender la interfaz original y el intercambio de datos
  - Remodelar el comportamiento para facilitar el trabajo aprovechando los componentes gráficos
  - Introducir mejoras
  - Construir e integrar la nueva GUI

# Referencias

- Ingeniería del software, un enfoque práctico (Roger S. Pressman).
- Reutilización de Software: Un paso hacia la Reingeniería . 1996.  
Novatica, N° 122. Barcelona. España.