



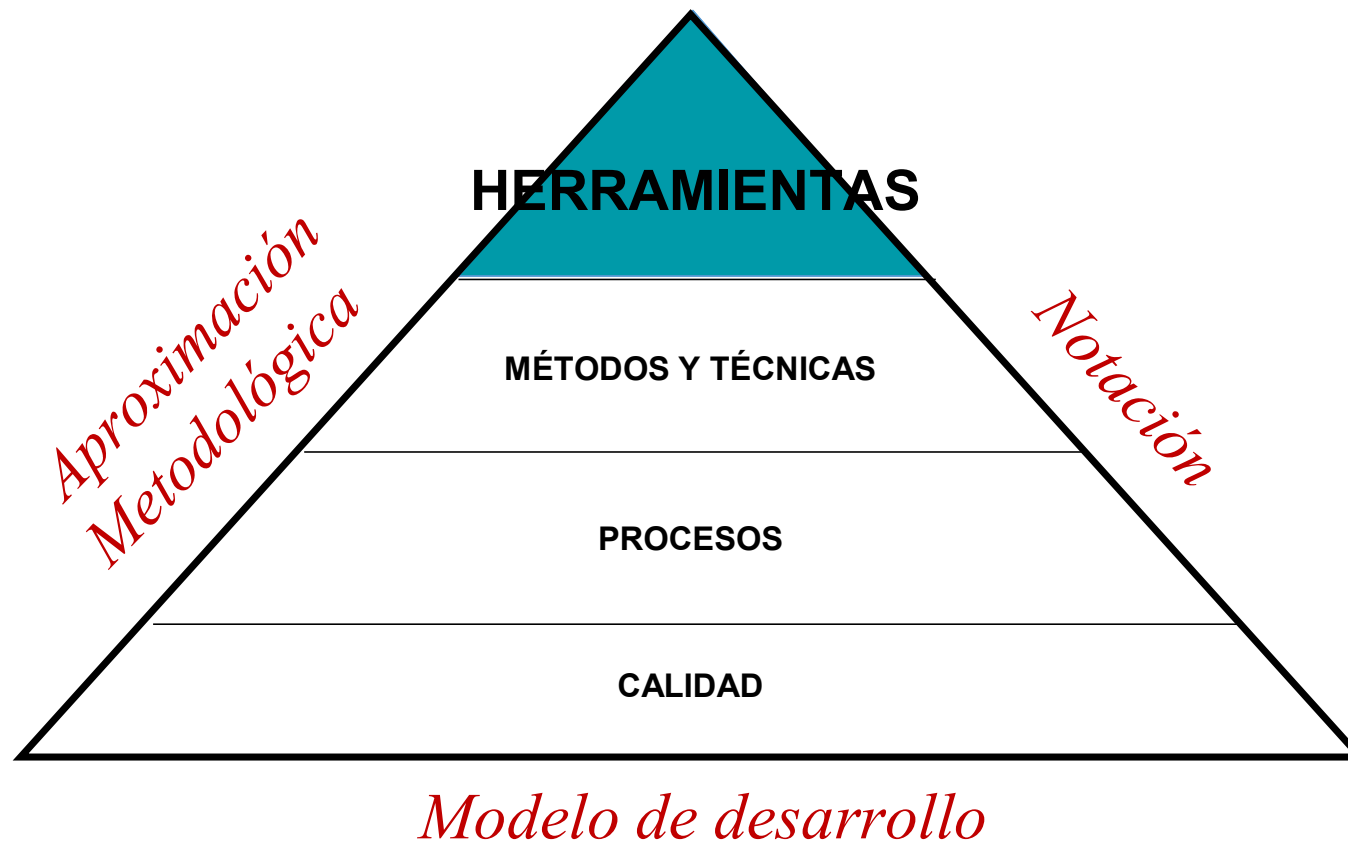
FACULTAD  
**INGENIERÍA  
& ARQUITECTURA**

# Análisis y Diseño de Sistemas

Unidad I: Fundamentos para el análisis y diseño de sistemas

Tema: Herramientas CASE

# Ingeniería del Software



## Concepto de Herramientas CASE (HC)

**CASE** es un acrónimo para **Computer Aided Software Engineering**, aunque existen algunas variaciones para lo que actualmente se entiende por CASE:

**C**

Computer

**A**

Aided

Assisted

Automated

**S**

Software

Systems

**E**

Engineering

## ¿Qué es una Herramienta CASE?

En “Terminology for Software Engineering and Computer-aided Software Engineering”, B.Terry & D.Logee, Software Engineering Notes, Abril 1990, CASE es definido como:

**“Herramientas individuales para ayudar al desarrollador de software o administrador de proyecto durante una o más fases del desarrollo de software (o mantenimiento).”**

## ¿Qué es una Herramienta CASE?

En “The CASE Experience”, Carma McClure, BYTE Abril 1989 p.235 se ofrece la siguiente definición:

**“Una combinación de herramientas de software y metodologías de desarrollo”**

## Definición de Herramienta Case

Conjunto de herramientas y métodos asociados que proporcionan asistencia automatizada en el proceso de desarrollo del software a lo largo de su ciclo de vida.

- ❑ En la gestión del proyecto  
*(planificación, estimación y control)*
- ❑ En el desarrollo del software  
*(análisis, diseño, implementación, validación)*
- ❑ En el mantenimiento del software

## Objetivos de las Herramientas Case

- ❑ Automatizar las actividades de gestión de proyectos y mejorar la visión de la ingeniería.
- ❑ Ayudar en el análisis, diseño y codificación.
- ❑ Garantizar que la calidad se diseñe antes de llegar a construir el producto.
- ❑ Incrementar la reusabilidad del software.
- ❑ Reducir costes de desarrollo y mantenimiento.

## Objetivos de las Herramientas Case

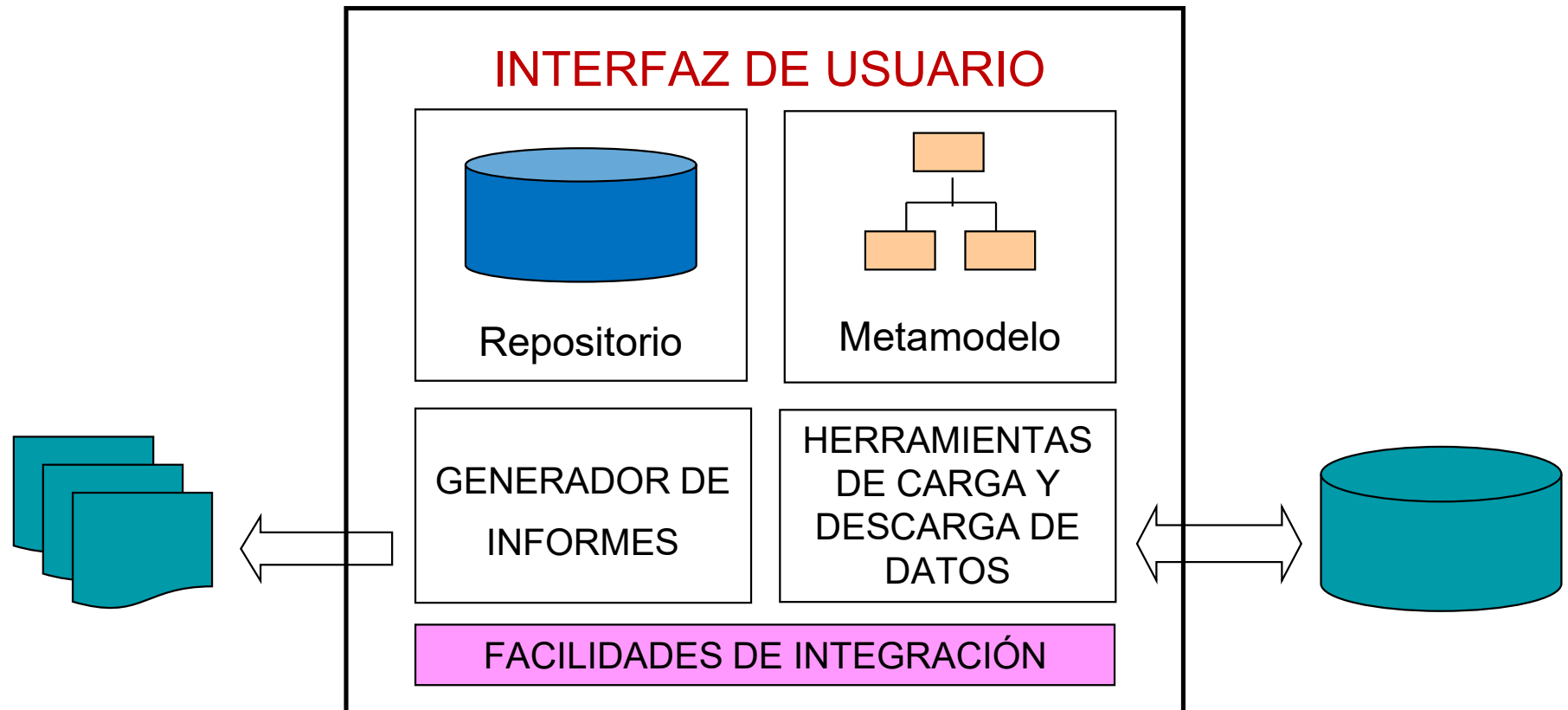
- ❑ Introducir automatización en el desarrollo de software
  - Productividad del equipo
  - Calidad del software
- ❑ Incrementar
  - Reutilización del software
- ❑ Reducir
  - Costes de desarrollo y mantenimiento
- ❑ Automatizar y simplificar
  - Gestión del proyecto
  - Desarrollo del software
    - facilitar la aplicación de métodos
    - prototipos
    - desarrollo “visual”
  - Mantenimiento del software
    - incluyendo la automatización y estandarización de la documentación y de su mantenimiento



## Características deseables

- ❑ Soporte gráfico para varias técnicas (DFD, DER, modelos OO, etc.)
- ❑ Control de errores, unicidad de identificadores, reglas, metodología, etc.
- ❑ Control de documentos y versiones
- ❑ Métricas del software
- ❑ Simulación y prototipado
- ❑ Generación de código
- ❑ Verificación entre diferentes modelos

## Componentes de una Herramienta CASE



## Elementos básicos de las Herramientas CASE

- ❑ Interfaz gráfica
- ❑ Editor de textos y gráficos
- ❑ BD de soporte
  - (BD del proyecto, depósito o repositorio CASE)
- ❑ Mecanismos de control
  - acceso a componentes
  - (datos, código, documentos, dispositivos)
  - Compatibilidad de las herramientas
  - Consistencia de los productos
  - Detección de olvidos
  - Trazado de modificaciones

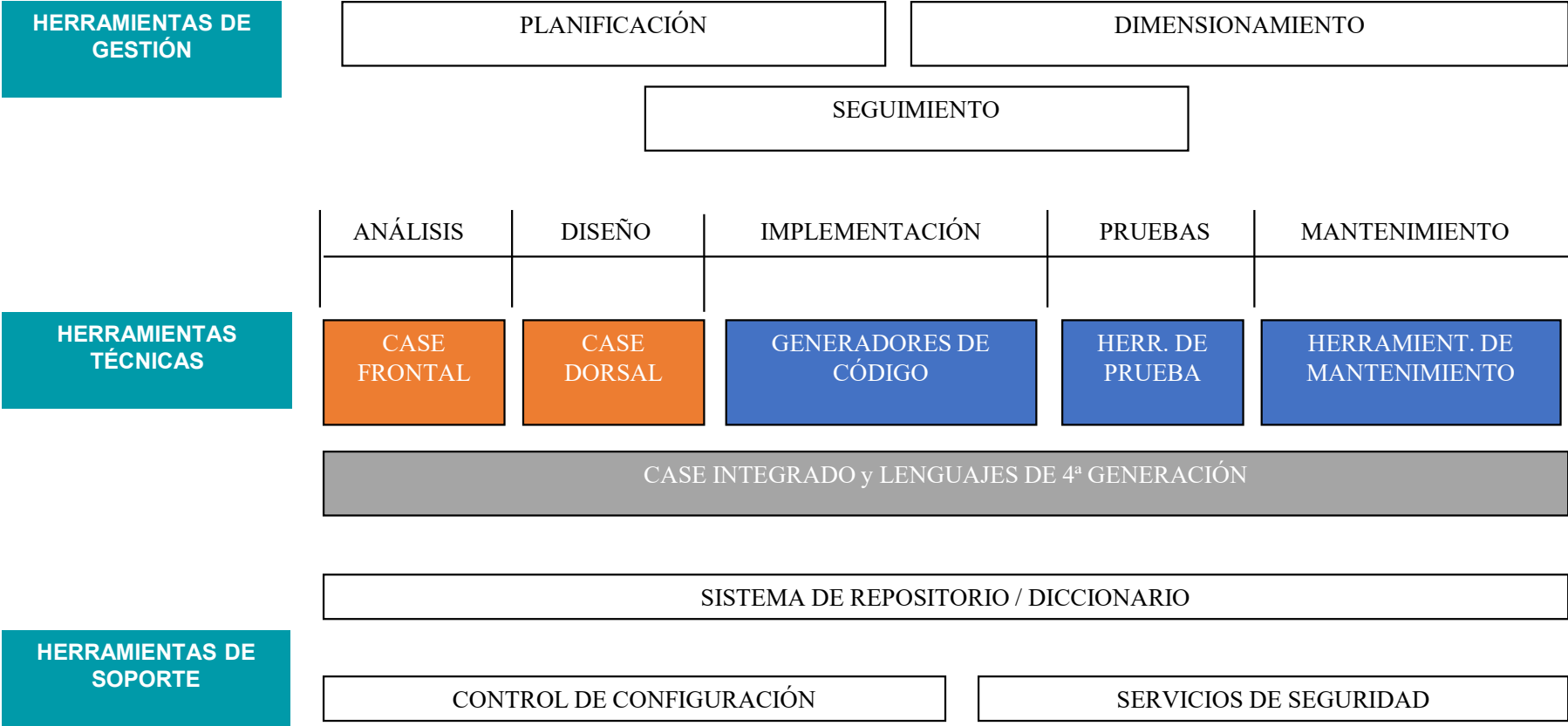
## Taxonomía de las Herramientas Case

- ❑ Herramientas de gestión
- ❑ Herramientas técnicas
- ❑ Herramientas de soporte
- ❑ Herramientas de apoyo a las primeras fases
  - Análisis, diseño
- ❑ Herramientas de apoyo a las últimas fases
  - Implementación (generación de código)
  - Pruebas (caja blanca y caja negra)
  - Mantenimiento

## Taxonomía de las Herramientas Case

- ❑ CASE frontales (front-end) o upper CASE
  - Gestión, Análisis, diseño
- ❑ CASE dorsales (back-end) o Lower CASE
  - Implementación (generación de código)
  - Pruebas (caja blanca y caja negra)
  - Mantenimiento
- ❑ ICASE (Integrated-CASE)
  - Contienen elementos de Upper y Lower CASE: contemplan todo el ciclo de desarrollo.

# Categorías de Herramientas CASE



## Algunas Herramientas CASE

- Erwin
- ER/Studio
- DESIGNER de ORACLE
- EASY CASE
- Rational ROSE
- OBJECT MAKER
- Visual Paradigm
- SYSTEM Architect (IBM)
- Architect (Sparx Systems)
- Visual Studio / VS Code
- Eclipse
- Netbeans
- PyCharm
- Azure Devops
- Junit
- Team Box
- Git / Github
- IntelliJ IDEA
- Jira
- Trello
- Confluence
- Microsoft Teams
- Slack
- Zoom / Meet / Skype

## Criterios de Selección

- ❑ Plataforma soportada
- ❑ Lenguaje y SGBD al que va orientada
- ❑ Metodología y técnicas soportadas
- ❑ Posibilidades de integración con otras plataformas (presente y futuro)
- ❑ Criterios habituales en la selección de software
  - Formación
  - Precio
  - Asistencia técnica
  - Mantenimiento



## Análisis y Diseño de Sistemas 2S2024



### ☐ Architect (Edición Enterprise, Versión 2015)

- Gestión del proyecto
- Requerimientos
- Diagrama de flujo de datos
- Árboles de decisión
- Diagrama entidad-relación
- Prototipos
- Documentación

Architect ofrece mecanismos (extensiones y estereotipos) para incorporar otros resultados del proceso de ciclo de vida de desarrollo de software

