



Sesión 2: Problema de Diseño

TOPICOS ESPECIALES SISTEMAS OPERATIVOS

DOCENTE

Ing. Jorge Bravo

<https://sites.google.com/site/jorgedavidbravoe>



UNIVERSIDAD NACIONAL
DE TRUJILLO
*Rumbo a la
Excelencia Académica*



Escuela de
INFORMÁTICA

INDICE

TEMAS

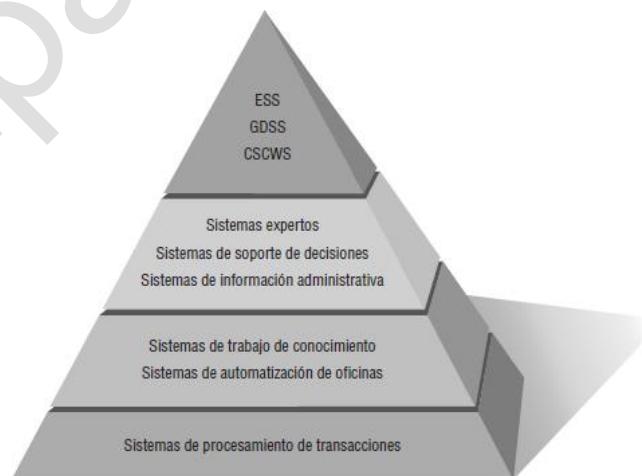
- CONTEXTO
- PROBLEMA DEL D.S.O
- OBJETIVOS DEL D.S.O
- TRABAJO GRUPAL 01



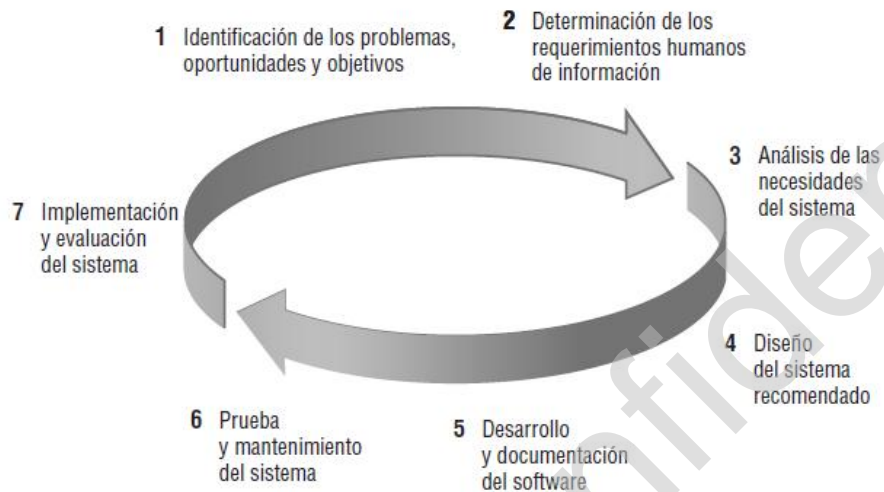
CONTEXTO

EN EL DISEÑO DE SISTEMAS

TIPOS DE SISTEMAS



CICLO DE DESARROLLO



PROBLEMA

EN EL DISEÑO DE SISTEMAS OPERATIVOS

PROBLEMA

- Para poder diseñar un sistema operativo exitoso, los diseñadores deben tener *una clara idea* de lo que quieren.
- La falta de *un objetivo* dificulta de manera considerable el proceso de tomar las decisiones subsiguientes.

EJEMPLO-lenguajes de programación: PL/I y C

- IBM diseñó PL/I en la década de 1960, debido a que era una molestia dar soporte tanto a FORTRAN como a COBOL, y era vergonzoso que los académicos empezaran a parlotear en trasfondo que Algol era mejor que ambos.
 - Por lo tanto, se estableció un comité para producir un lenguaje que cumpliera con las necesidades de todas las personas: PL/I. Tenía un poco de FORTRAN, un poco de COBOL y un poco de Algol.
- Fracásó debido a que carecía de una visión unificadora.
 - Era tan sólo una colección de características en guerra unas con otras, y demasiado voluminoso como para poder compilarse con eficiencia, para iniciarse.

EJEMPLO-lenguajes de programación: PL/I y C

- Ahora consideremos C.
- Este lenguaje fue diseñado por Dennis Ritchie con un propósito: *la programación de sistemas*.
 - Fue un enorme éxito, en gran parte debido a que Ritchie *sabía lo que quería y lo que no*.
 - Como resultado, se sigue utilizando de manera extensa más de tres décadas después de su aparición.
- Tener una visión clara de lo que uno quiere *es algo imprescindible*.

OBJETIVOS

EN EL DISEÑO DE SISTEMAS OPERATIVOS

¿Qué desean los diseñadores de sistemas operativos?

1. Definir las abstracciones.
2. Proveer operaciones primitivas.
3. Asegurar el aislamiento.
4. Administrar el hardware.

DEFINIR LAS ABSTRACCIONES

- La tarea más importante (y tal vez la más difícil) de un sistema operativo es definir las abstracciones correctas.
- Algunas de ellas, como los procesos, los espacios de direcciones y los archivos, han estado presentes tanto tiempo que pueden parecer obvias.
- Otras, como los hilos, proceso multihilo, (hilo bloqueado, bifurcación)
- Otras abstracciones se relacionan con la sincronización, las señales, el modelo de memoria, el modelado de la E/S y muchas otras áreas.

PROVEER OPERACIONES PRIMITIVAS

- Cada una de las abstracciones se puede instanciar en forma de estructuras de datos concretas.
 - Los usuarios pueden crear procesos, archivos, semáforos, etc.
- Las operaciones primitivas manipulan estas estructuras de datos.
 - Por ejemplo, los usuarios pueden leer y escribir archivos.
 - Las operaciones primitivas se implementan en la forma de llamadas al sistema.
- Desde el punto de vista del usuario, *el corazón del sistema operativo* se forma mediante *las abstracciones y las operaciones* disponibles en ellas, a través de las llamadas al sistema.