

Estructura de Datos

UNIDAD 0: INTRODUCCIÓN



¿Qué es el Software?

- Es el **componente lógico** de los sistemas de computación que facilita a las personas el uso de los recursos hardware de una computadora.
- El software comprende:
 - **instrucciones** (secuencia de operaciones)
 - **datos** (representación de objetos del mundo real)
 - **documentación** (manuales de usuario, instalación, documentos de diseño, etc.)
- **¿Cómo se construye el software?**

Desarrollo de Programas



Ingeniería de Software

- La **Ingeniería de Software** es una disciplina que integra procesos, métodos y herramientas para el desarrollo de software de alta calidad.
- El desarrollo de software se realiza siguiendo diferentes **modelos de proceso** (ciclos de vida). Estos modelos comprenden una serie de fases genéricas que guían el desarrollo.

Modelos de Proceso. Fases (1)

○ Fases Genéricas de Desarrollo

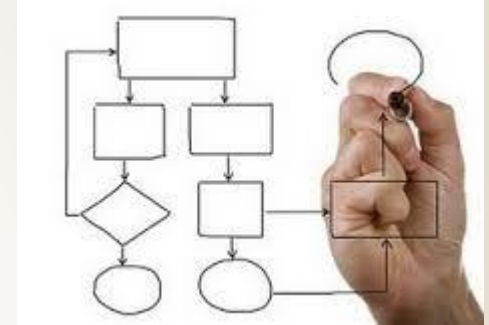
- **Comunicación:** comunicación y colaboración con el cliente para la determinación de requisitos.
- **Planeación:** planificación de tareas y recursos necesarios, productos intermedios y estudio de riesgos.



Modelos de Proceso. Fases (2)

- Fases Genéricas de Desarrollo

- **Modelado:** se formulan modelos (del sistema), se depuran requisitos y se diseñan soluciones.
- **Construcción:** generación de código y realización de pruebas de software.

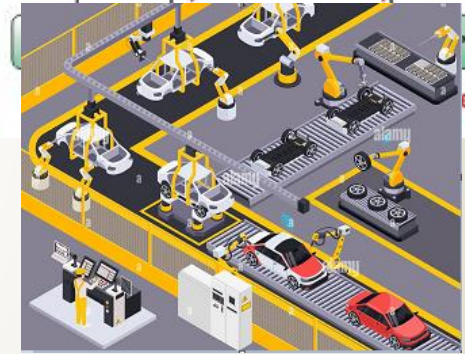


Modelos de Proceso. Fases (3)

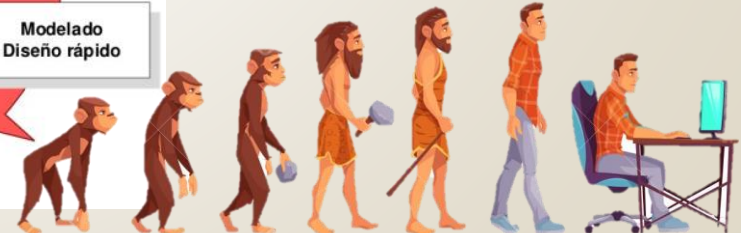
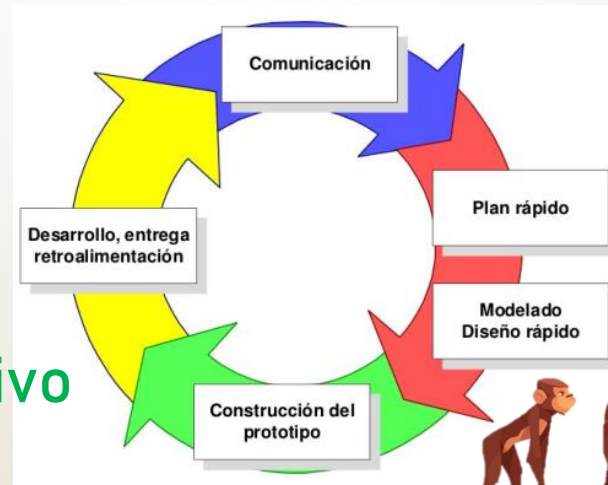
- Fases Genéricas de Desarrollo
 - **Despliegue:** el software (en versiones parciales o producto terminado) se entrega al usuario para su evaluación.



Modelos de Proceso. Fases (4)



Evolutivo

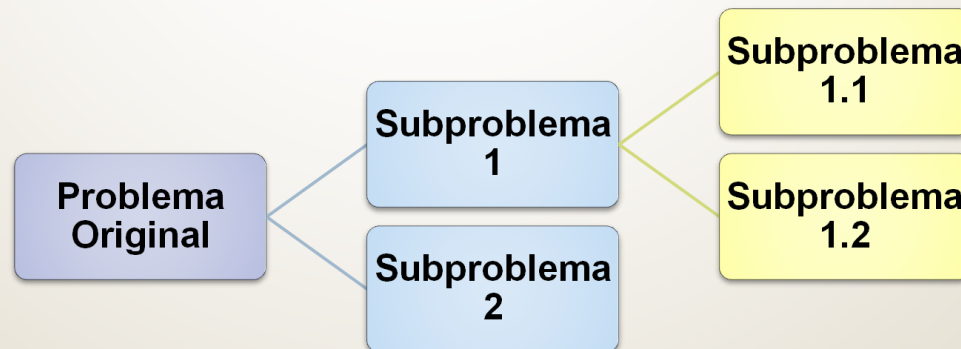


Diseño de Software

- En el diseño de software se utilizan técnicas que ayudan a **comprender el problema** a resolver. Estas técnicas **reducen la complejidad** estudiando problemas más pequeños y evaluando sólo los **aspectos importantes**.
- Durante el diseño se aplican:
 - la descomposición de problemas,
 - el refinamiento sucesivo y
 - la abstracción.

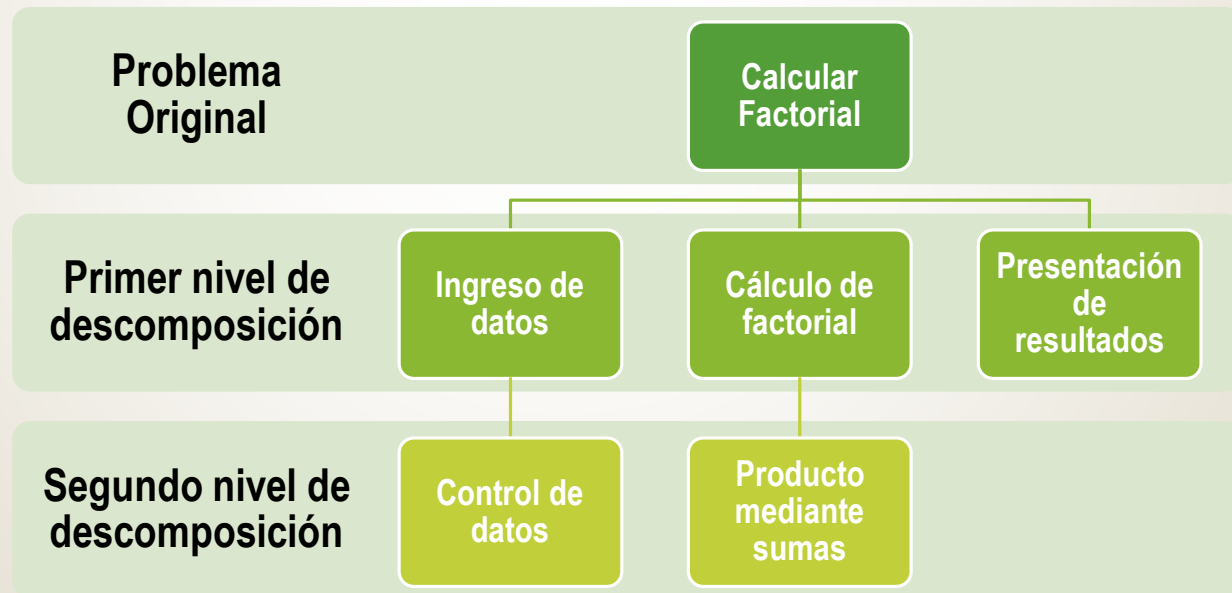
Descomposición y Refinamiento (1)

- La **descomposición** de problemas permite dividir un problema complejo en pequeños problemas de solución más simple.
- Por cada nivel de descomposición, el **refinamiento sucesivo** permite especificar con mayor detalle los pasos para solución de cada subproblema identificado.
- En el nivel más bajo, las soluciones se especifican mediante las instrucciones del lenguaje de programación adecuado.



Descomposición y Refinamiento (2)

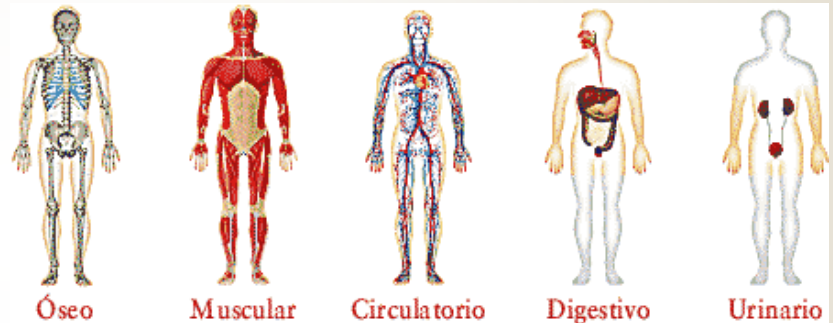
- Ejemplo: Diseñe un algoritmo que calcule el factorial de un número ingresado por el usuario. Recuerde que el cálculo del factorial sólo está definido para números mayores o iguales a cero. Considere que los productos se resuelven mediante sumas sucesivas.



Abstracción

- La **abstracción** consiste en enfocarse sólo en los aspectos relevantes del problema, situación u objeto que se analiza.

El mismo sistema puede analizarse desde distintos puntos de vista según el problema que se intente resolver.



- En el desarrollo de software, pueden distinguirse 2 tipos de abstracción:
 - abstracción de procesos
 - abstracción de datos

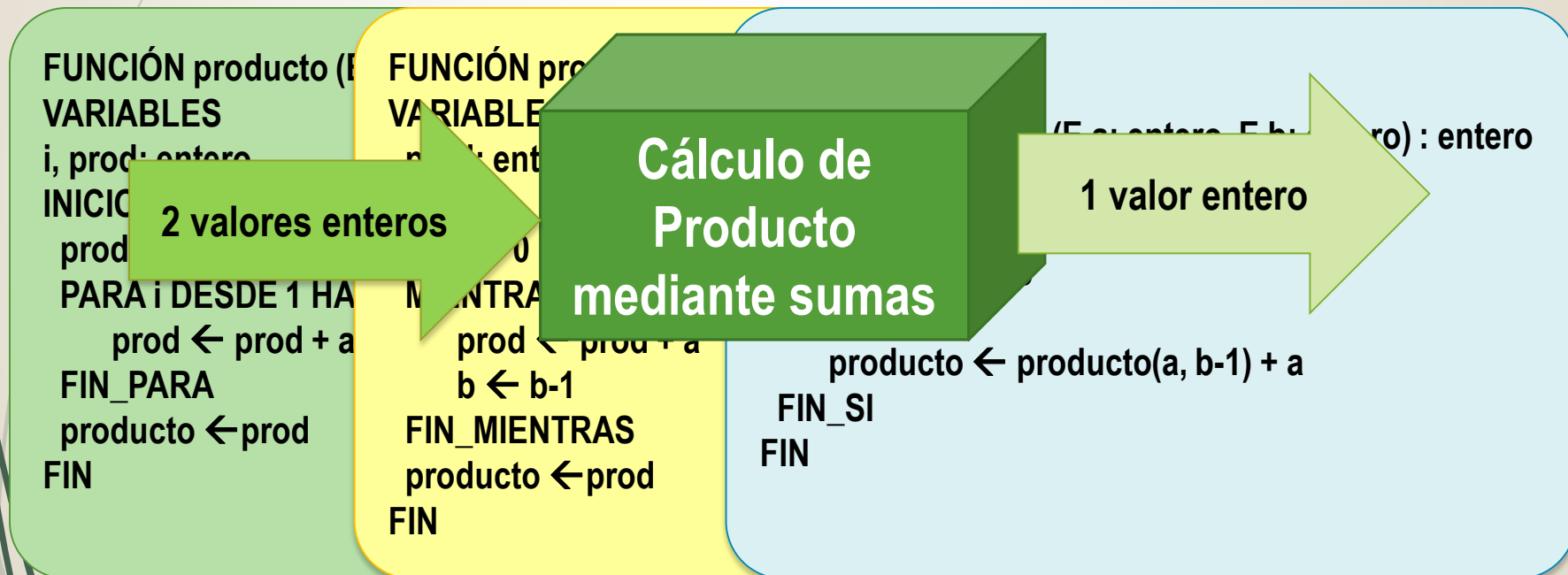
Abstracción de Procesos (1)

- Un programa puede organizarse en unidades o **módulos** que realizan tareas específicas.
- Cada módulo puede tratarse como una **caja negra** que puede usarse conociendo únicamente:
 - **propósito** del módulo (qué hace),
 - datos de **entrada** (parámetros de entrada del módulo) y
 - **resultados** (salidas o datos generados por el módulo).



Abstracción de Procesos (2)

- Ejemplo: diseñe un módulo que calcule el producto mediante sumas.



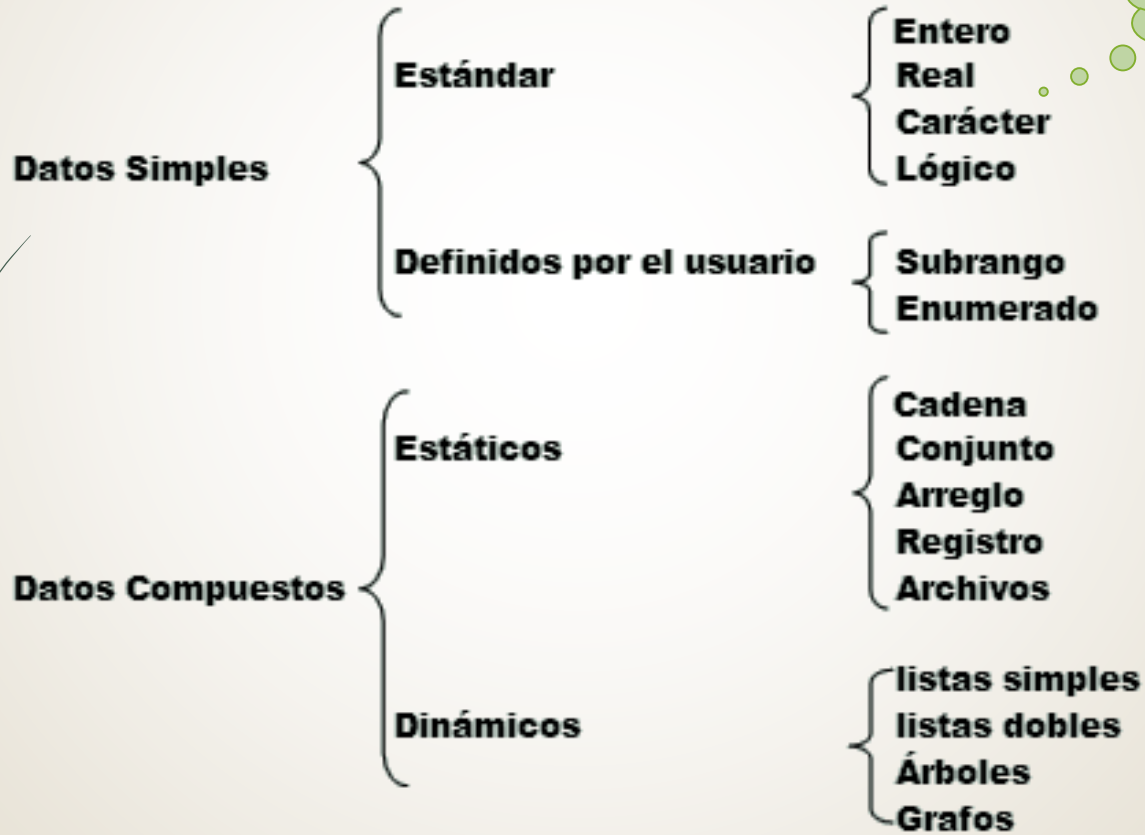
Abstracción de Datos (2)

- Un **tipo de dato** hace referencia a un **conjunto de valores**.
- Un **tipo de dato abstracto** (TDA) comprende tanto el **conjunto de valores** como las **operaciones** que pueden aplicarse a ese conjunto.
- Una **estructura de datos** se refiere a la **implementación** física de un tipo de dato abstracto.



Abstracción de Datos (3)

- Tipos de datos estándar



puntero

Bibliografía

- Pressman, Roger S. Ingeniería de Software: Un enfoque práctico. 6^{ta} Edición. McGraw Hill. 2005.
- Joyanes Aguilar *et al.* Estructuras de Datos en C++. Mc Graw Hill. 2007.
- De Giusti, Armando *et al.* Algoritmos, datos y programas, conceptos básicos. Editorial Exacta. 1998.
- Joyanes Aguilar, Luis. Fundamentos de Programación. Mc Graw Hill. 1996.