Análisis y Diseño

Ana LAUREANO-CRUCES
Alberto YÁÑEZ-CASTILLO

Universidad Autónoma Metropolitana-Azcapotzalco

1. Entender el problema o la situación:

- Este paso parece banal pero es muy importante; ya que la comprensión tiene que pasar por *un proceso cognitivo donde se comprende el problema*.

- Si no sabemos que es lo que vamos a programar no podemos codificarlo.

- Lo anterior implica al siguiente paso como uno de los pasos más importantes y que *competen al analista de sistemas*.

2. Análisis del sistema

- Un sistema es un conjunto estructurado de elementos interrelacionado de alguna forma que puede hacerse explícita.
- Lo anterior con el fin de poder analizar el problema que se va a llevar a una realidad computacional.
- El resultado de esta etapa consiste en diagramas que muestren el flujo de la información.

2. Análisis del sistema

- La función del analista de sistemas consiste en describir el modelo que mejor se adapte a la estructura del problema:

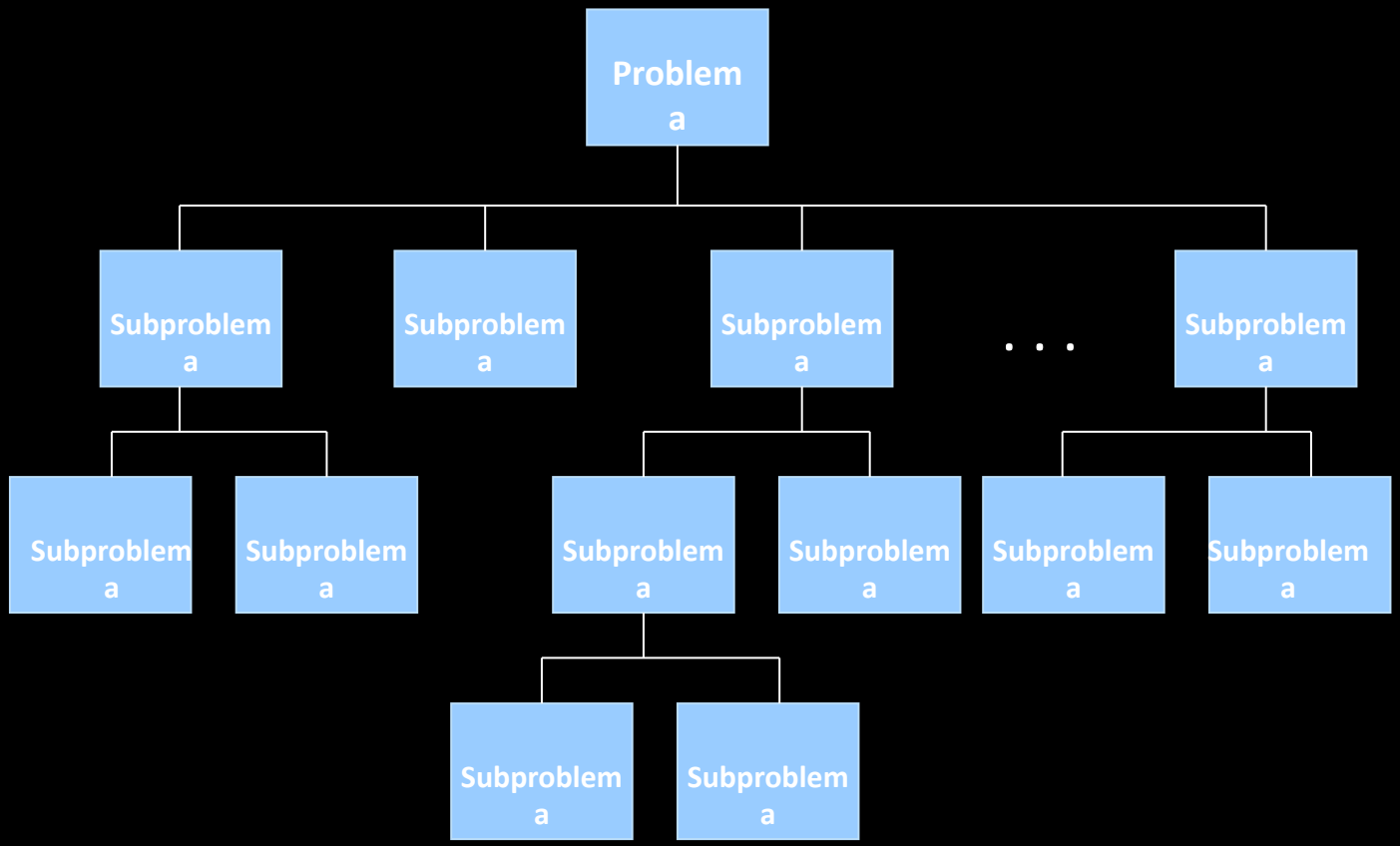
  - *Un enfoque funcional:* permite hacer un análisis partiendo de la función que desarrolla cada componente desempeña en el sistema como un todo.
  - *Un enfoque dirigido por datos:* implica conocer el tipo de datos que utiliza el sistema.

Análisis del sistema

- En esta clase utilizamos un análisis modular descendente, *de arriba hacia abajo*; partiendo de lo general a lo particular. Tiene un enfoque funcional y dirigido por datos.

  - El análisis descendente (*top down*) es un procedimiento *de refinamiento iterativo* de un problema, en el cual se parte del mayor nivel de abstracción del problema (sistema o tarea) y se prosigue hacia los niveles inferiores a través de un proceso de descomposición del problema en subproblemas.
  - Se parte de una visión estructural del problema sin especificar detalles para ninguna de sus partes componentes. Cada componente del sistema es entonces refinado, mostrando más detalles en cada nivel de refinamiento.

2.1. Análisis modular descendente



3. Diseño

- Mencionamos que la etapa de análisis esta formada por módulos, que implican descomponer problemas en sub-problemas más simples que se puedan resolver y comprender de una manera relativamente independiente.

- El diseño de una descomposición apropiada es la parte más difícil de ésta técnica; debido a que la descomposición no es obvia en un principio.

- Entran en juego el paso de datos y las funciones específicas de cada uno de los módulos.

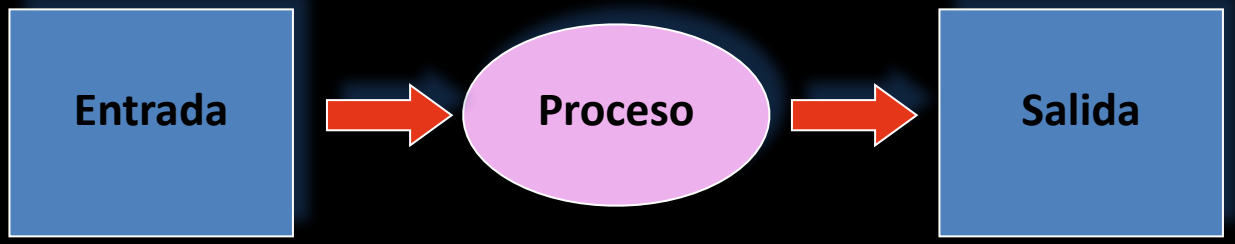
4. Programar

- Es poco probable el establecimiento de la corrección de un programa mediante pruebas, a menos que se tome en cuenta su estructura interna Dijkstra [1972] .
- La única esperanza es diseñar cuidadosamente un programa de manera que su corrección pueda entenderse en términos de su estructura.
- El arte de la programación es el arte de organizar la complejidad.
- Debemos organizar los cálculos de manera que *nuestros limitados sentidos sean suficientes para garantizar que el cómputo arroje los resultados esperados.*

4. Programar

- En este punto de la etapa del diseño, utilizamos *las estructuras de control y las estructuras de datos apropiadas*; que nos permiten llevar a cabo los distintas tareas y sub-tareas de cada uno de los módulos en los que se ha descompuesto el problema.

- Como se mencionó anteriormente cada módulo estará formado por un *algoritmo relativamente independiente*. El cuál dependerá de los datos de entrada y producirá otros datos necesarios para otros módulos.



4. Programar - algoritmo

- Hemos mencionado que cada módulo esta representado por un algoritmo. Dicho algoritmo esta compuesto por las instrucciones que permitirán desarrollar su tarea.

- Un algoritmo es un procedimiento para resolver un problema. Éste describe un conjunto finito y ordenado de pasos, reglas o instrucciones para producir la solución de un problema dado.

- Un algoritmo puede ser definido como una secuencia de instrucciones bien definidas y efectivas, y finaliza con la producción del resultado esperado a partir de las entradas de datos dadas.

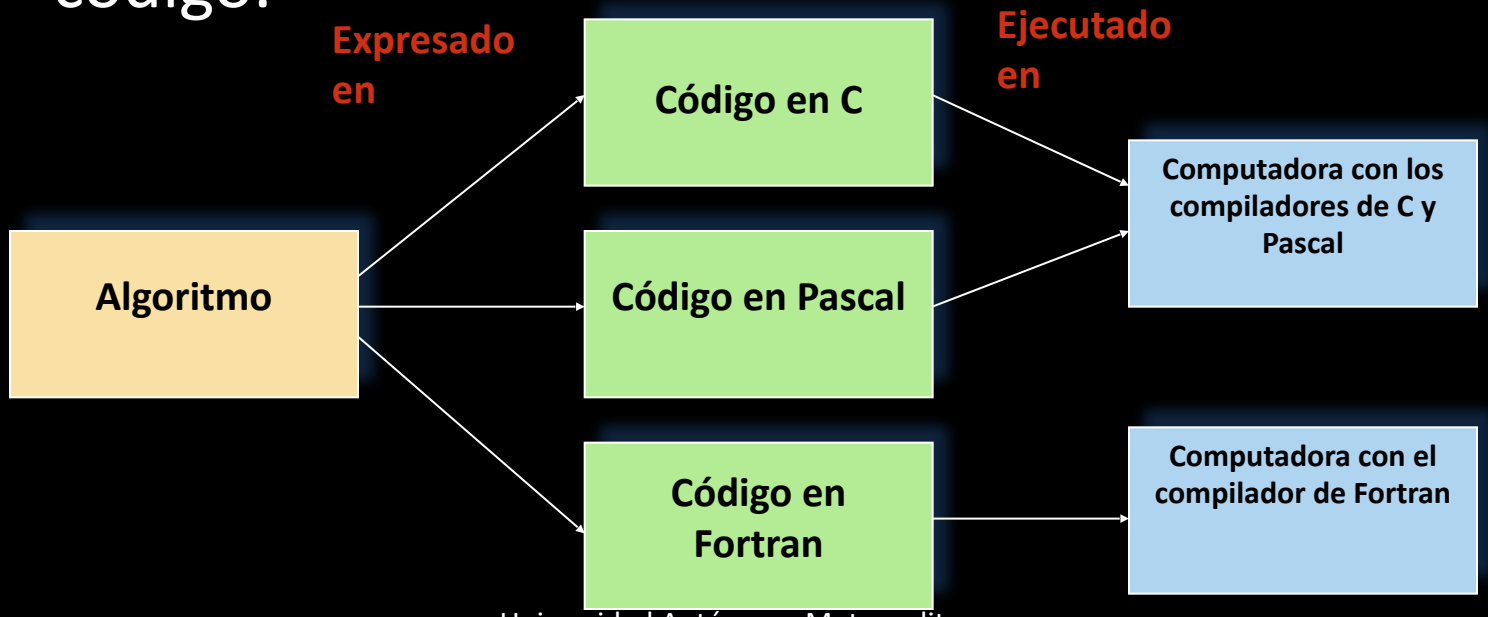
Concepto de Algoritmo

4. Programar - algoritmo

- **Sus características son:**
- Precisos, deben indicar el orden de realización para cada paso, así como especificar con precisión las entradas y cada paso o etapa.
- Bien definidos, las etapas deben ser correctamente delimitadas.
- Finitos, deben tener un número límite de pasos los cuales deben terminar en cierto tiempo.
- Exactitud y corrección, se debe demostrar que el algoritmo resuelve el problema para el cual fue escrito.
- Deben describir el resultado o efecto final. La salida esperada del algoritmo debe estar completamente especificada.

5. Codificación

- La codificación es la representación del modelo en un lenguaje computacional.
  - Se le llama construcción a la actividad de teclear el código.



6. Comprobación

- En este punto se distinguen tres actividades:
  - Integración y pruebas: implica corregir detalles operativos, cuando se ejecuta el programa por vez primera.
  - Liberación al usuario: implica la realización de un *manual para el usuario* y la utilización del sistema en forma paralela con el sistema manual.
    - Con el fin de detectar errores y afinar detalles.

7. Mantenimiento

- En este punto es importante haber desarrollado un manual técnico; que representa la estructura del diseño asi como los detalles de cada uno de los sub-módulos.
  - Con el fin de actualizar diferentes partes de él:
    - Correctez: mejorar algún algoritmo, modificar campos a archivos.
    - Escalabilidad: agregar o retirar submódulos

Fin de la presentación

- Gracias por su atención.

- Usted puede ampliar los conceptos vistos consultando la bibliografía, o bien algún otro medio de información donde consulte los conceptos mencionados en estas diapositivas.