

FASES DEL PROCESO DE LA Programación

1. Definición del problema

Requiere que el problema sea definido y comprendido claramente para que pueda ser analizado con todo detalle .

2. Análisis del problema

El propósito del análisis de un problema es ayudar al programador para llegar a una cierta comprensión de la naturaleza del problema. El problema debe estar bien definido si se desea llegar a una solución satisfactoria.

Para poder definir con precisión el problema se requiere que las especificaciones de entrada, proceso y salida sean descritas con detalles.

- **Entrada**, son todos los datos que hay que ingresar para la resolución del problema.
- **Proceso**, son los diferentes procedimientos en los cuales usare los datos proporcionados por el usuario en el paso anterior para resolver el problema.
- **Salida**, la resolución del problema.

Una buena definición del problema, junto con una descripción detallada de las especificaciones de entrada y salida, son los requisitos mas importantes para llegar a una solución eficaz.

El análisis del problema exige una lectura previa del problema a fin de obtener una idea general de lo que se solicita . la segunda lectura deberá servir para responder a las preguntas:

- ¿Qué información debe proporcionar la resolución del problema?
- ¿Qué datos se necesitan para resolver el problema?

La respuesta a la primera pregunta indicara los resultados deseados o las salidas del problema. La respuesta a la segunda pregunta indicara que datos proporcionan o las entradas del problema.

En esta fase debemos aprender a analizar la documentación de la empresa , investigar, observar todo lo que rodea el problema.

3. Diseño de la solución

Una computadora no tiene la capacidad para solucionar mas que cuanto se le proporcionan los sucesivos pasos a realizar. Estos pasos sucesivos que indican las instrucciones a ejecutar por la maquina constituyen, como ya sabemos el *algoritmo*.

La información proporcionada al algoritmo constituye su entrada y la información producida por el algoritmo constituye su salida.

Los problemas complejos se pueden resolver mas eficazmente con la computadora cuando se rompen en subproblemas que sean mas fáciles de solucionar que el original. La descomposición del problema original en subproblemas mas simples y a continuación dividir estos subproblemas en otros mas simples que puedan ser implementados para su solución en la computadora se denomina *diseño descendente (top – down design)*.

Es preciso representar el algoritmo mediante una determinada herramienta de programación: *diagramas de*

flujo, pseudocódigo o diagramas N – S.

4. Codificación

Es el algoritmo traspasado a la computadora a través de un lenguaje de programación, debe ser escrito de acuerdo a las reglas gramaticales o sintaxis del mismo.

Generamos un algoritmo el cual se denomina código, y al pasarlo a un lenguaje de programación se le llama código fuente.

5. Compilación y ejecución

Compilación, el lenguaje de programación seleccionado revisa que ya no halla errores en el código fuente.

Ejecución, ejecutar el programa después del ser compilado.

6. Verificación y pruebas

Revisar que el programa de los resultados que el usuario quiere, hay que revisar si hay algún error el cual puede ser de tipo lógico, de semántica o ejecución.

7. Depuración

Corregir los errores encontrados en la etapa anterior, si hubiese algún error se tiene que regresar hasta la etapa que sea necesaria para que la solución sea la que el usuario requiere.

8. Documentación

Son todos los comentarios, referencias desde que se dio el problema hasta que se encontró la solución.

a) documentación interna, es el encabezado, descripción, declaración del problema, son todos los comentarios que puedan llegar a servir dentro del código fuente.

b) documentación externa, son los manuales que se hacen para una mejor ejecución del programa, ahí se explicara como usar el programa.

9. Capacitación

Una vez que tenemos los manuales viene la capacitación de usuarios, como no a todos les interesa lo mismo porque hay diferentes tipos de usuarios, se tienen que dividir de acuerdo a sus conveniencias.

10. Implantación o implementación del sistema

Hay diferentes formas de implementar el sistema:

- Depende del tamaño del sistema
- Si existiera un sistema se tiene que poner el nuevo y quitar el que ya estaba.
- Trabajar primero con el sistema viejo (si existe), y después empezar a trabajar con el nuevo.
- Sistemas implantados por regiones.
- Trabajar con el sistema viejo (si existe) y con el sistema nuevo simultáneamente después quitar el sistema viejo.

- Ir quitando el sistema viejo(si existe) poco a poco e ir implantando el sistema nuevo.

11. Mantenimiento

El mantenimiento preventivo es que hagamos lo posible por no caer en errores, la actualización si el usuario tiene la necesidad de quitar o poner algo; téngase en cuenta que cuando surge mantenimiento tenemos que volver a hacer todos los pasos anteriores revisando que todas la condiciones sean favorables alrededor del sistema.

Fases de la programación:

- 1.- ANALISIS: Cuando se tiene y piensa la idea o problema a ejecutar.
- 2.- DISEÑO: Se plantea un algoritmo con los pasos a dibujar o realizarse.
- 3.- IMPLEMENTACION: Los pasos de algoritmo se transforma en un código pascal y se plasman en el programa
- 4.- PRUEBAS: Se corre el programa y se registra su efectividad.
- 5.- DEPURACION: después de haber corrido el programa se hacen las correcciones necesarias.
- 6.- RETROALIMENTACION Y LIBERACION: Después de haber corregido los errores se regresa al paso 4 en caso de haber existido.