

Técnicas básicas para el desarrollo de programas de computadoras

Programación Convencional

Definición: Técnica de programación enfocada en el desarrollo sin considerar el mantenimiento.

Programas sin estructura.

Instrucciones secuenciales.

Dificultad para depurar: Localizar y corregir errores.

Dificultad para modificar: Ampliar o desarrollar nuevos módulos.

Problemas:

Falta de transportabilidad: Problemas para ejecutar en diferentes entornos.

Falta de documentación: Dificulta la comprensión del programa.

Redundancia en el código: Repetición de secuencias de instrucciones.

Conclusión: No es recomendable para el desarrollo de software moderno.

Técnicas de Programación

Programación Convencional:

Características:

Programas sin estructura.

Instrucciones secuenciales.

Dificultad para depurar y modificar.

Redundancia en el código.

Falta de documentación.

Problemas de transportabilidad.

Problemas:

Programación por Procedimientos:

Definición: Uso de funciones o procedimientos para evitar repetición de código.

Reduce redundancia.

Facilita mantenimiento.

Ventajas:

Dificultad para determinar cuándo crear un procedimiento.

Ejemplo: Uso de funciones para tareas repetitivas.

Desventajas:

Programación Modular

Definición: División del programa en módulos independientes.

Características:

Módulo principal y subordinados.

Comunicación mediante parámetros.

Cada módulo realiza una tarea específica.

Ventajas:

Facilita trabajo en equipo.

Reduce redundancia.

Mejora mantenimiento y verificación.

Requisitos:

Tamaño entre 20 y 200 líneas de código.

Nombre único para cada módulo.

Cabecera y cuerpo definidos.

Técnicas de Recolección de Datos

Definición: Proceso de recopilar información para una investigación.

Tipos:

Datos primarios: Obtenidos directamente de la realidad.

Observación: Observar y registrar fenómenos.

Entrevista: Conversación dirigida con preguntas y respuestas.

Encuesta: Preguntas normalizadas a una muestra o población.

Cuestionario: Conjunto de preguntas para obtener datos.

Datos secundarios: Información ya procesada y recopilada por otros.

Programación Estructurada

Origen: Décadas de 1960 y 1970, popularizada por Niklaus Wirth (creador de Pascal) y E. W. Dijkstra.

Principios:

Estructuras básicas de control:

Secuencial: Ejecución de acciones en orden.

Alternativa: Ejecución condicional (if, if-else, switch).

Repetitiva: Bucles (while, do-while, for).

Iterativa: Ejecución basada en condiciones o valores.

Recursos Abstractos: Descomposición de acciones en sub-acciones.

Metodología descendente (arriba-abajo): Resolución de problemas desde lo general a lo específico.

Características:

División en funciones: Agrupación de tareas en funciones con un propósito definido.

Módulos: Agrupación de funciones en unidades más grandes (archivos o ficheros).

Estructuras de control: Uso de secuencias, selecciones y repeticiones.

Ventajas:

Facilita el mantenimiento y la modificación.

Reduce la redundancia en el código.

Mejora la claridad y comprensión del programa.

Permite trabajar en equipo.

Métodos conocidos:

Bertini.

Michael Jackson.

Lassy (Universidad de Nancy).

Warnier.

Gordfolh.