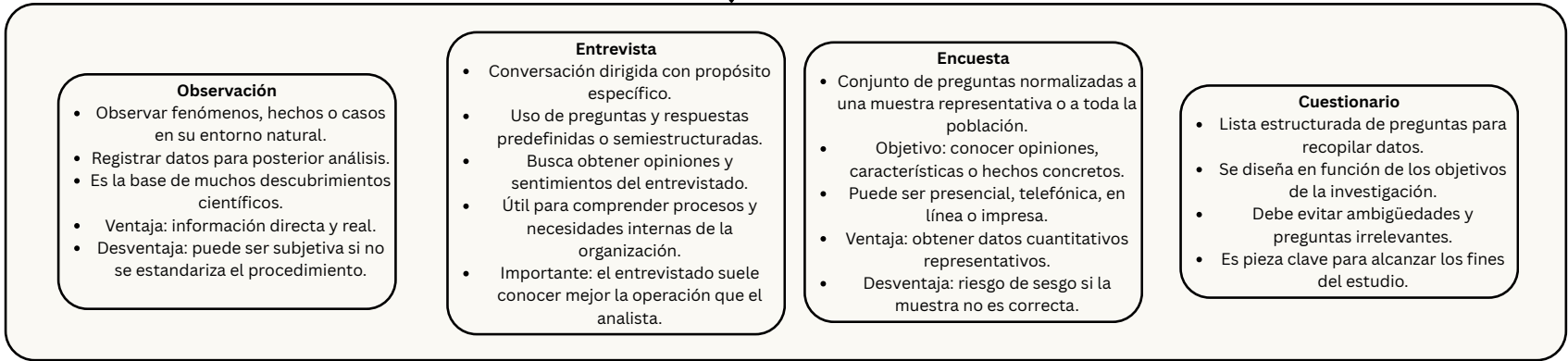


Técnicas Básicas para el Desarrollo de Programas de Computadoras

Métodos usados por el analista para obtener información útil antes de desarrollar un sistema. Pueden ser datos primarios (recogidos directamente) o secundarios (obtenidos y procesados por otros previamente).



Enfoque Monolítico (Programación Convencional)

Características:
Código sin estructura.
Dependencia total del programador original para mantenimiento.

Problemas:
Difícil depuración y localización de errores.
Modificaciones costosas.
Baja transportabilidad entre sistemas.
Redundancia de código.
Falta de documentación.

Programación Procedimental

Basada en funciones o procedimientos que agrupan secuencias repetidas.

Ventajas:

- Reduce redundancia.
- Facilita el mantenimiento y organización.
- Adecuada para grandes proyectos.

Desventajas:

- Difícil determinar el punto óptimo para encapsular en funciones.
- Ejemplo: función que recibe un número y muestra el anterior, el mismo y el siguiente.

Programación Modular

División del programa en módulos con tareas concretas.

Características de un módulo:

- Nombre único.
- Comunicación por parámetros.
- Puede compilarse y almacenarse por separado.
- Contiene cabecera (nombre y datos) y cuerpo (código).
- Tamaño ideal: 20–200 líneas de código.
- Una tarea única y completa.

Ventajas:

- Trabajo en equipo.
- Menos redundancia.
- Mantenimiento más sencillo.

Otras Técnicas de Diseño

Programación Estructurada

- Introducida por Dijkstra (años 70).
- Basada en estructuras de control básicas:
 - Secuencial.
 - Alternativa (if, if-else, switch).
 - Repetitiva (while, do-while, for).
- Uso de descomposición funcional y diagramas en árbol.
- Lenguajes comunes: Pascal, C, Ada.
- Métodos: Jackson, Bertini, Lassy, Warnier, Gordfolh.

Línea Recta

- Lenguajes independientes de la máquina, fáciles de aprender.
- Ejemplos históricos: FORTRAN, COBOL, BASIC.
- Clasificación:
 - Imperativos (C, Pascal).
 - Declarativos (Prolog, Lisp).
 - Orientados a Objetos (C++, Smalltalk).
 - Orientados al Problema (SQL, NATURAL).
 - Naturales (lenguaje cercano al humano).
- Por generación:
 - 1ª: máquina y ensamblador.
 - 2ª: FORTRAN, COBOL.
 - 3ª: Pascal, C.
 - 4ª: SQL, bases de datos.
 - 5ª: IA, procesamiento de lenguaje natural.

Enfoque E-P-S (Entrada – Proceso – Salida)

- **Entrada:**
 - Datos desde usuario, archivos o bases de datos.
 - Puede ser en línea, fuera de línea o por lotes.
- **Proceso:**
 - Diseño del modelo + algoritmo de resolución.
 - Representación: pseudocódigo, diagrama de flujo, diagrama N-S.
- **Salida:**
 - En pantalla, papel, disco o gráficos.
 - Debe cumplir formato y frecuencia definidos.

Programación Orientada a Objetos (POO)

- Paradigma actual dominante.
- Basada en objetos con atributos y métodos.
- **Ventajas:**
 - Encapsulación.
 - Reutilización.
 - Facilidad de mantenimiento.
- Ejemplos: Java (100% POO), C++ (híbrido).
- Objetos interactúan entre sí, cada uno con su propio ciclo de vida.

Programación Procedimental

Método Descendente (Top-Down)

- Se diseña primero el módulo principal y luego los específicos.
- Organización jerárquica.
- Único módulo con jerarquía 0 (principal).
- Se crean diagramas de diseño estructurado que muestran orden, control y responsabilidad.

Programación Procedimental

Método Ascendente (Bottom-Up)

- Se comienza con módulos de bajo nivel y se avanza al superior.
- La estructura visual es similar al descendente, pero cambia el orden de programación.