

UNIDAD III.

TÉCNICA DE PROGRAMACIÓN ESTRUCTURADA



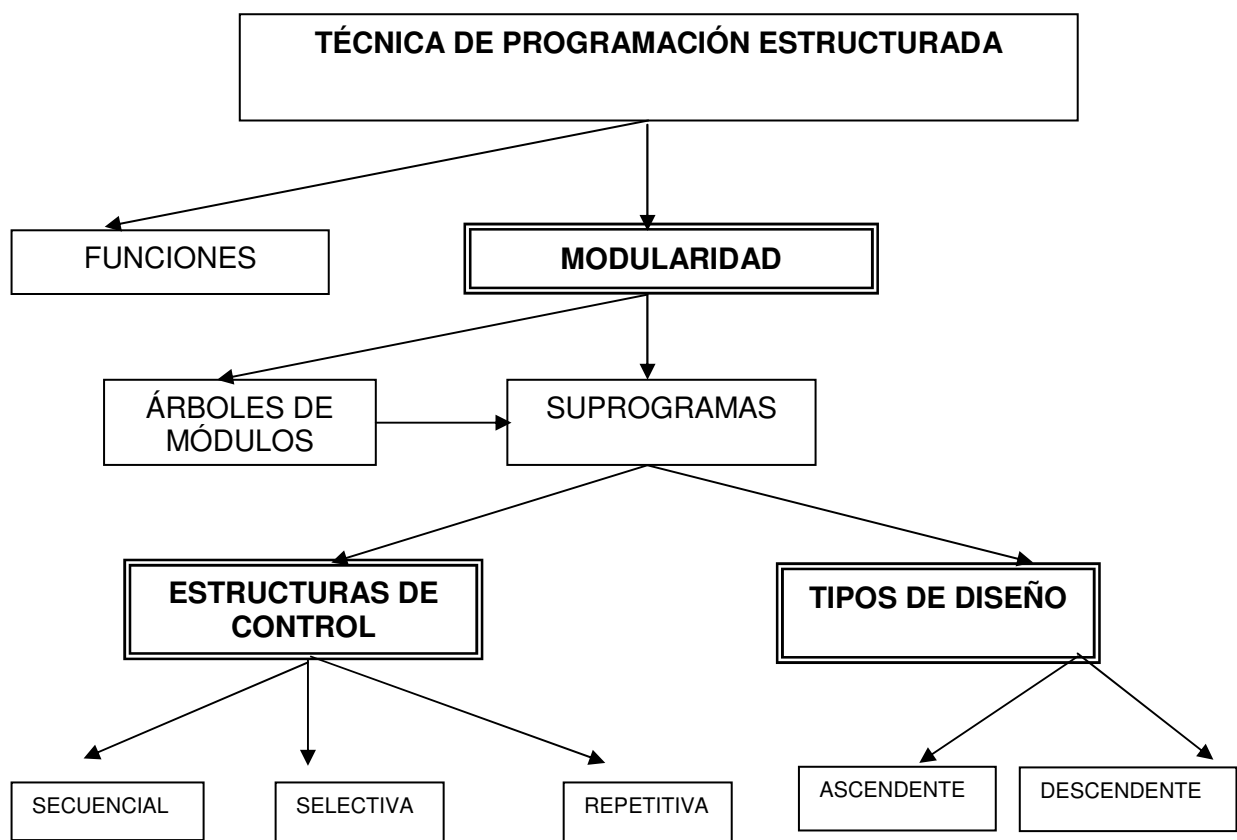
UNIDAD III.

TÉCNICA DE PROGRAMACIÓN ESTRUCTURADA

OBJETIVO: Aplicarás la técnica de programación estructurada, analizando su concepto, estructura y función en la resolución de problemas, lo que te permitirá implementar el proceso de estructuración de la información, con el fin de que desarrolles programas informáticos elementales.

Para comprender los contenidos de esta unidad, requieres **ejercitar la agrupación, el detalle, la agregación y desagregación de tareas; haciendo uso de una relación lógica de jerarquía** (de mayor a menor).

A continuación te presentamos los conceptos e ideas clave de la Unidad.



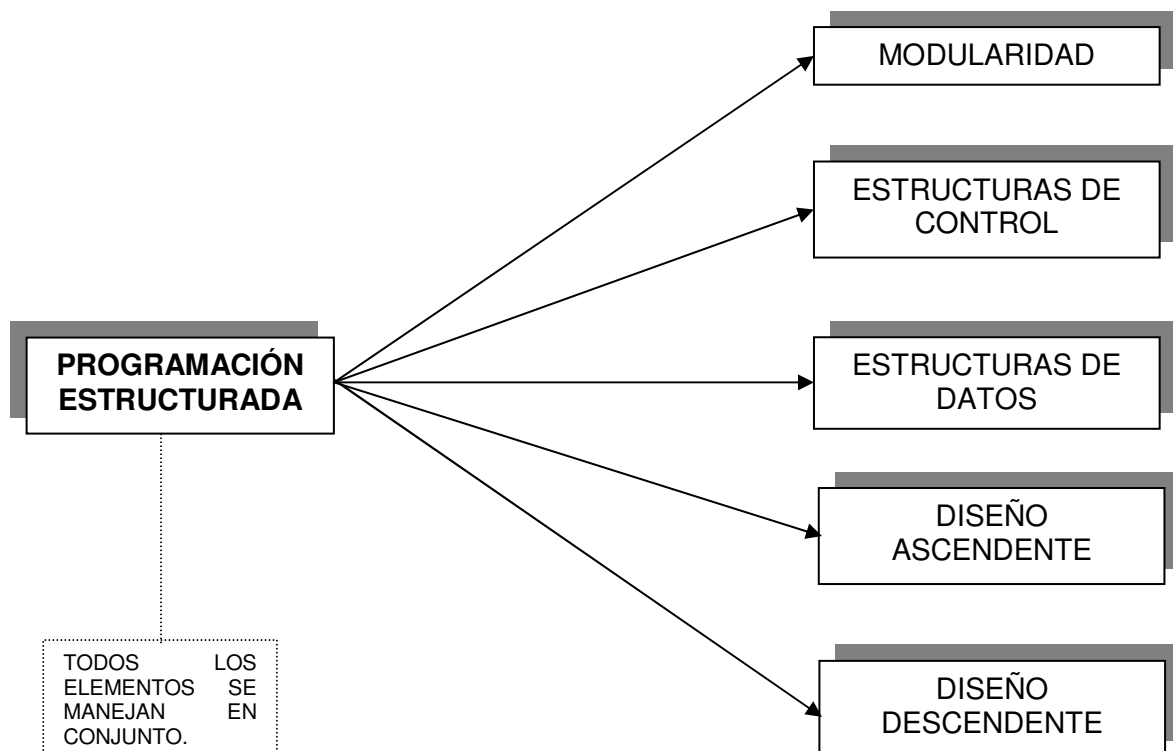
I. PROGRAMACIÓN ESTRUCTURADA.

En los años cincuenta, en la programación no existían reglas o normas para la construcción de programas, con la evolución de nuevas aplicaciones informáticas los problemas se incrementaban. La *programación estructurada* se empieza a manejar a finales de los años sesenta. El nacimiento de esta herramienta es sustancial para la evolución dentro de la programación, ya que inicia la construcción de programas siguiendo técnicas y no una construcción convencional como la que se tenía anteriormente.

Los objetivos de la programación estructurada son:

- ❑ Establecer procesos de diseño.
- ❑ Establecer lógicas sencillas y comprensibles.
- ❑ Realizar estructuras fáciles de modificar.
- ❑ Elaborar objetivos de programa que resuelva las necesidades.
- ❑ Realizar programas de alta calidad.

Los elementos que forman la Programación Estructurada son los siguientes:



DISEÑO ESTRUCTURADO.

Define que un programa puede ser dividido en módulos para lograr mejor calidad dentro de la programación. Este diseño se le llama **PROGRAMACIÓN MODULAR**, así lo definen [PAGE-JONES, 1980], [MYERS, 1975] Y [YOURDON Y CONSTANTINE, 1975].

Un **MÓDULO** es¹:

Una secuencia de instrucciones que se agrupan con un objetivo único: desarrollar una tarea específica.

MODULARIDAD.

Es un método que tiene la función de dividir un sistema o programa de mayor tamaño en pequeñas unidades o subrutinas, que desarrollen procesos individuales para lograr la resolución de problemas.

El Diseño Modular se fundamenta en la siguiente estructura:

✕ **COHESIÓN.**

Hace referencia al grado de integración de las instrucciones de un módulo.

- ❑ Cada módulo debe realizar un sólo proceso o función.
- ❑ El proceso en cada módulo debe resolver el problema planteado.
- ❑ Cada módulo se debe comprender fácilmente, si el proceso que realiza el módulo es complicado se puede dividir en submódulos más pequeños.

✕ **ACOPLAMIENTO.**

Muestra y precisa el grado de relación entre módulos.

- ❑ Se busca diseñar módulos que tengan un amplio margen de independencia, pero que de alguna forma se relacionen entre sí.

Este diseño permite que los sistemas tengan más lógica y sean más comprensibles manejando solamente los siguientes parámetros:

- **Datos de Entrada.**
- **Datos de Salida.**
- **Proceso que realiza el módulo.**

¹ COLEGIO DE BACHILLERES: *Apuntes de técnicas para el diseño de sistemas. Unidad IV Técnicas de diseño.* Agosto, 1990, pág. 67.

Entre más cohesión y menor acoplamiento tenga un módulo, estará mejor construido.

La definición de cada Módulo dentro de un programa o sistema debe tener las siguientes características:

- ❑ **Nombre asignado al Módulo.**
- ❑ **Proceso que realiza el Módulo.**
- ❑ **Comunicación que tiene con otros Módulos.**

Ejemplo:

Módulo: Actualiza datos de los empleados.
Función: Da de alta los datos de nuevos empleados en el archivo maestro.
Entrada: Puesto, sueldo y departamento.
Salida:

El diagrama anterior muestra un ejemplo de un sistema actualizador de Control de Personal, indicando todos los módulos y funciones. El sistema tiene inicialmente un menú:

ALTAS, BAJAS, CAMBIOS, CONSULTAS, REPORTES Y FINALIZACIÓN.

- Muestra los módulos y la jerarquía que existen entre ellos.
- Cada módulo ejecuta una función específica.
- Si faltara, por ejemplo, la rutina de reportes, no estaría completo el proceso.
- Los módulos que no tienen conexión, son módulos compartidos por otros procesos.
- En la implantación del sistema o programa los módulos compartidos son integrados en bibliotecas, para ser utilizados por otros procesos en cualquier momento.

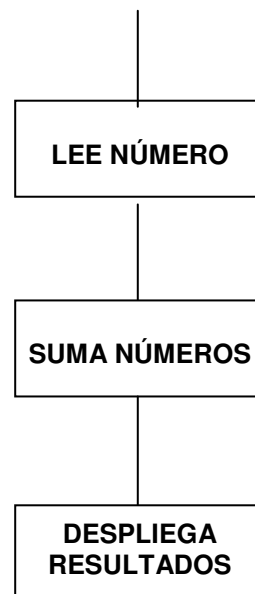
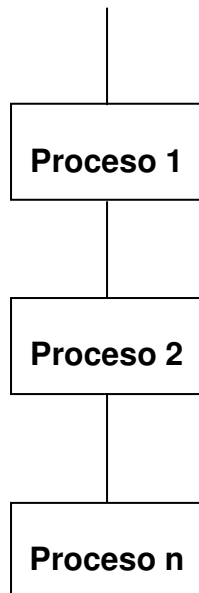
ESTRUCTURAS DE CONTROL

Permiten establecer el flujo de la Información, a través de condiciones y acciones que al ejecutarse, nos dan un resultado. La programación estructurada define la solución a cualquier problema utilizando las siguientes estructuras: **secuencial, selectiva y repetitiva.**

SECUENCIAL.

Ejecuta los procesos en orden jerárquico, uno tras otro, en una secuencia establecida con anterioridad. Cada proceso se define como el conjunto de instrucciones unidas por la estructura dada.

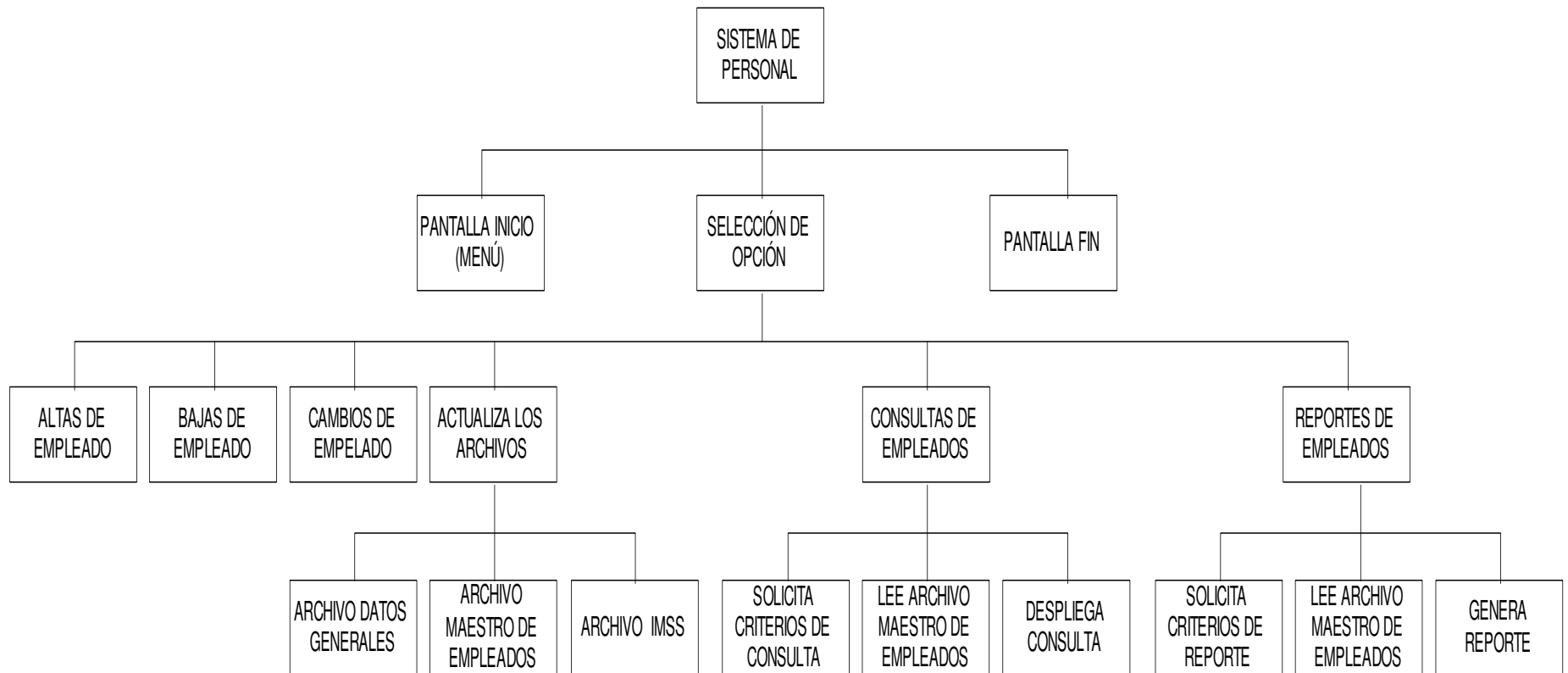
Ejemplo: suma de dos números enteros.



ÁRBOLES DE MÓDULOS.

Es un diagrama que representa la estructura de un sistema y la jerarquía de los módulos. Los módulos son representados por rectángulos y con líneas de conexión entre cada uno de ellos.

A continuación te presentamos el diagrama de módulos referente a un sistema de personal.

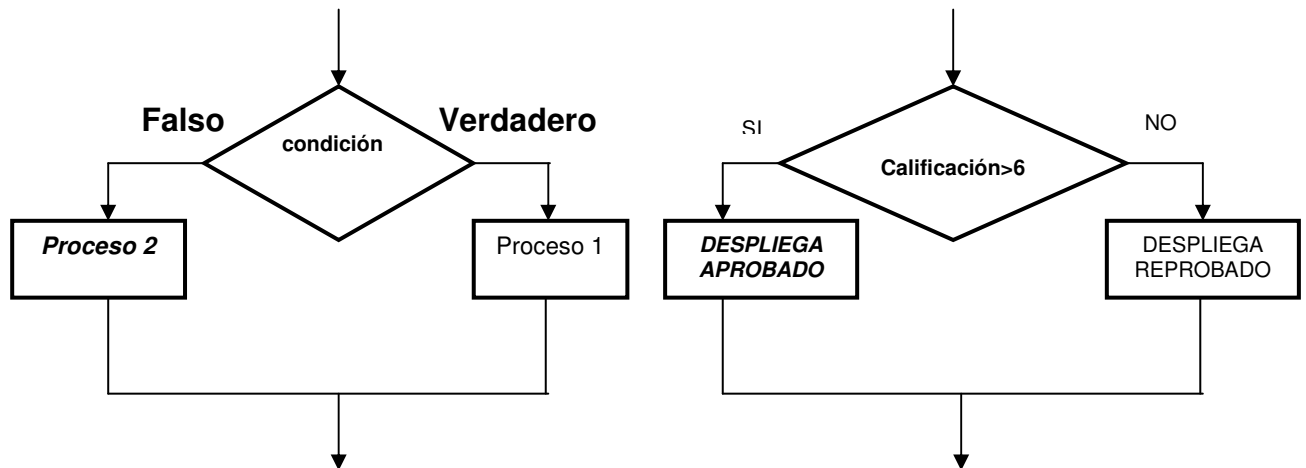


SELECTIVA.

Condición simple (IF-THEN-ELSE)

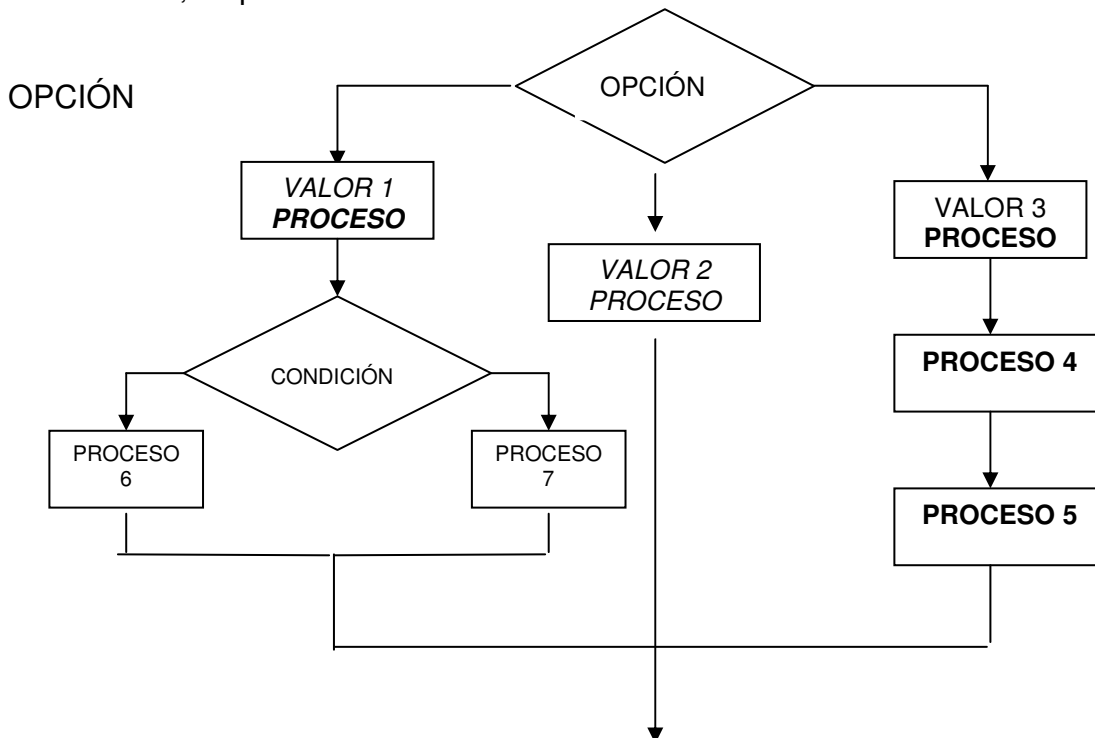
Ejecuta un proceso dependiendo del resultado de la condición, si la condición fue verdadera ejecuta el proceso 1, pero si el valor de la condición es falso, ejecuta el proceso 2, uniéndose posteriormente en un proceso 3.

Ejemplo: dada una calificación numérica desplegar aprobado o reprobado.



CONDICIÓN COMPUESTA (CASE).

Dependiendo del valor de la Opción ejecuta uno de varios procesos. Como se ve, es posible combinar todas las estructuras.

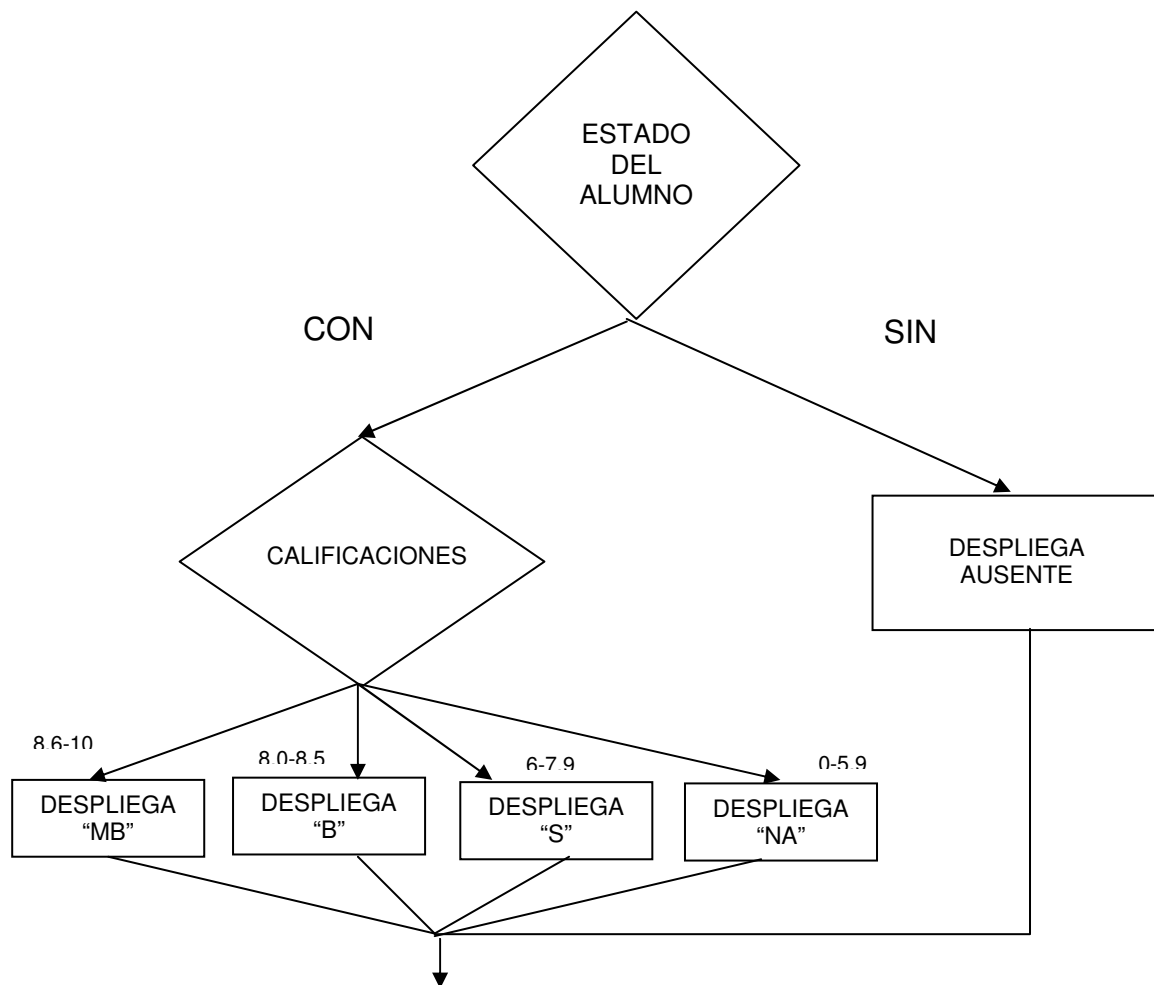


Ejemplo.

Utilizando la estructura de control selectiva compuesta (CASE) representa el siguiente caso:

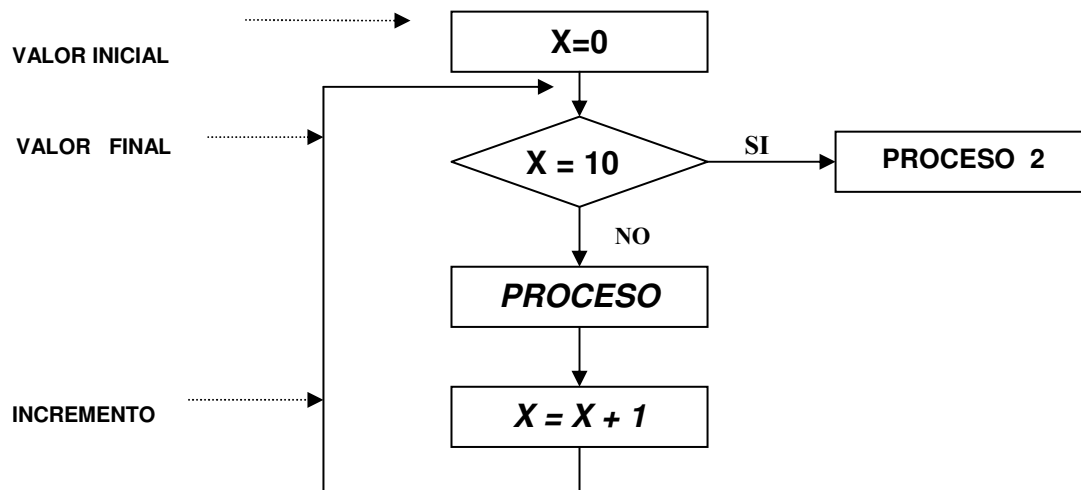
Asignar una calificación con letra a un alumno, a partir de la siguiente tabla:

CALIFICACIÓN NUMÉRICA	CALIFICACIÓN CON LETRA
8.6 – 10	MB
8.0 - 8.5	B
6.0 - 7.9	S
0.0 - 5.9	NA



ESTRUCTURA REPETITIVA

Esta estructura se utiliza para generar un ciclo dentro de un diagrama. El fin del ciclo lo controla una condición y el valor máximo asignado a la variable. Para el manejo de esta estructura se necesitan 3 elementos principales: **VALOR INICIAL**, **VALOR FINAL** e **INCREMENTO**



TIPOS DE DISEÑO

Es una Técnica que se emplea en la creación de Programas utilizando las **Estructuras de Control** y la **Modularidad**. Dentro de esta técnica se manejan el diseño **TOP-DOWN** y **BOTTOM-UP**.

TOP-DOWN.

En este diseño **se trabajan los módulos de arriba hacia abajo**, iniciando por el modulo principal o programa principal. Este diseño se aplica a cualquier tipo de problema, en especial cuando no se tiene conocimiento profundo del mismo.

CARACTERÍSTICAS:

- Definir el diseño de cada módulo para la resolución del problema, a través de sus **Instrucciones y Estructuras de Control**.
- Definir qué función realiza cada **Módulo** sin entrar a **detalle, esto es**, cómo realizarán las operaciones.
- En el momento de definir a detalle la función de cada modulo, también se define la **comunicación** que tendrá con los demás módulos.

BOTTOM-UP.

En este diseño ***se trabajan los módulos de abajo hacia arriba***, inicia por identificar los procesos elementales.

CARACTERÍSTICAS:

- Identificar cada módulo y las funciones que realizará para atacar cada problema.
- Manejar a detalle los procesos de cada módulo que se está creando y la comunicación que tendrá con los elementos externos.
- Esta técnica no es recomendada cuando no se tiene un conocimiento absoluto de los problemas.
- Sé obtiene buen resultado si se combina con la técnica **TOP-DOWN**.