



UAI

Universidad Abierta Interamericana

EVALUACIÓN PARCIAL – RESOLUCIÓN DE CASOS

ALUMNO:

- Di Salvio Tiago
 - 42727210
 - tiago.disalvio@alumnos.uai.edu.ar

CARRERA: Ingeniería en Sistemas Informáticos

ASIGNATURA: Programación Estructurada

FECHA: 28/05/2021

1. ¿Qué es un programa?

Un programa no es nada más que una serie de instrucciones dadas al ordenador en un lenguaje entendido por él, para decirle exactamente lo que queremos hacer. Si el ordenador no entiende alguna instrucción, lo comunicara mediante un mensaje de error.

2. ¿Por qué cree conveniente separar en fases el proceso de desarrollo de software?

El proceso de desarrollo de software es conveniente separarlo en fases por lo siguiente:

- Se analiza el problema, teniendo presente las especificaciones y requisitos del cliente o encargado.
- Se pasa a la fase de diseño, (previamente habiendo analizado el problema) se diseña una solución que será un algoritmo el cual resolverá el problema.
- Luego pasamos a la codificación, que es el resultado de las dos fases anteriores (análisis y diseño) se escribe en la sintaxis de un lenguaje de alto nivel (por ej.: Lenguaje en C) y se logra un programa.
- Teniendo el programa listo, se ejecutará, comprobará exhaustivamente y se eliminarán todos los errores (“bugs”) que puedan llegar a ocurrir.
- Se tiene en cuenta que los requisitos y especificaciones del cliente o encargado puedan cambiar, este se actualizara y/o modificara el programa cada vez que sea necesario, cumpliendo con las necesidades del usuario.
- En esta fase “documentación” se escribirán las diferentes fases del ciclo de vida del software, esencialmente el análisis, diseño y codificación, unidos a manuales de usuario y referencia, así como normas para el mantenimiento.
- En la última fase: ciclo de vida de un sistema, comienza desde que se recibe los requerimientos por parte del cliente hasta que este se vuelva obsoleto, puede ser por cambios en el negocio, avances tecnológicos o dejar de utilizarlos.

3. ¿Qué dice el teorema de la programación?

El teorema de Böhm y Jacopini cuenta que:

Un programa propio puede ser escrito utilizando únicamente tres tipos de estructuras de control, estas son:

- Estructuras secuenciales.
- Estructuras de selección.
- Estructuras de repetición.

Este teorema afirma que, si los algoritmos se diseñan utilizando exclusivamente dichas estructuras de control, los algoritmos y por resultado los programas derivados de ellos *serán implementables*.

Tomando en base al teorema de Böhm y Jacopini, definimos las siguientes instrucciones de control que nos permiten variar o alterar la secuencia normal de ejecución de un programa.

La gran mayoría de los lenguajes de programación de alto nivel soportan tres tipos de instrucciones de control:

- ❖ Instrucciones condicionales o alternativas.
- ❖ Instrucciones de salto.
- ❖ Instrucciones repetitivas.

4. Compare y relacione el concepto de compilar con el de interpretar.

Compilar: Es un programa que tomará como datos nuestro programa escrito en el lenguaje de alto nivel (programa fuente) y dará como resultado el mismo programa, pero escrito en lenguaje máquina, lenguaje que entiende el ordenador. Al finalizar la traducción se ejecutará automáticamente un programa denominado “enlazador” el cual es el encargado de incorporar las funciones de la biblioteca del lenguaje utilizado necesaria para nuestro programa.

Como por ejemplo lenguajes de compiladores:

- Pascal
- C/C++
- Cobol
- .net

Interpretar: No genera un programa escrito en lenguaje máquina a partir del programa fuente, sino que efectúa la traducción y ejecución simultáneamente para cada una de las sentencias del programa.

A diferencia de un compilador, un intérprete verifica cada línea del programa cuando se escribe, lo cual facilita la puesta a punto del programa. En cambio, la ejecución resulta mas lenta ya que encamina una traducción simultánea.

Como por ejemplo lenguajes de interpretes:

- Ruby
- Python
- JavaScript

5. Clasifique los lenguajes de programación según su nivel.

Los lenguajes de programación son aquellos que nos permiten comunicarnos con las computadoras; son símbolos y palabras que le permiten al usuario darle ordenes a la computadora para que las ejecute.

Esos se clasifican según su nivel:

Lenguaje de maquina: Este lenguaje fue el inicial, se dejó de usar por su dificultad y facilidad de cometer errores para escribirlo, utiliza el alfabeto binario (0,1) se conoce como bits, con ellos se forman cadenas binarias que escriben las instrucciones que el microprocesador entenderá y ejecutará.

Lenguaje de bajo nivel: El assembler fue el primer lenguaje de estos, son más difíciles de escribir, depurar, mantener, de aprender, específicos para un procesador, la ventaja que posee es generar programas rápidos

Lenguaje de alto nivel: Son independientes de la maquina a donde se implementará, es el frecuentemente utilizado. el inconveniente es que necesita una traducción a lenguaje máquina por medio de compilador e interprete. Presenta una superioridad de mantener, escribir y depurar.

6. ¿Qué es un tipo de datos? Indique los que conozca y explique para que sirve.

Es la propiedad de un valor que determina su dominio (valores que puede tomar), que operaciones se le pueden aplicar y como es representado internamente por el procesador.

- Lenguaje en C, no soporta un gran numero de tipos de datos predefinidos, pero tiene la capacidad para crear sus propios tipos de datos.

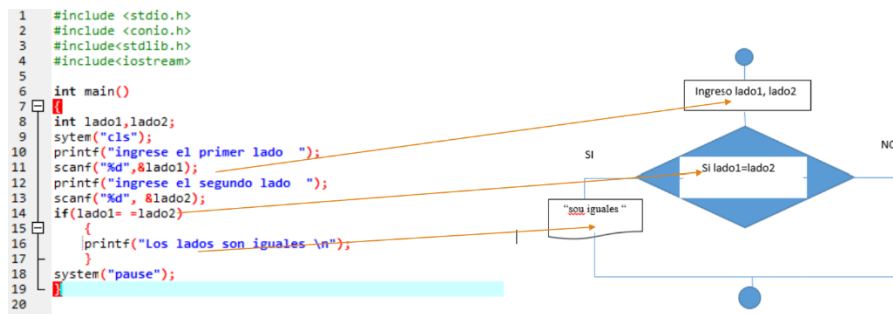
Los tipos de datos básicos en lenguaje C son:

- Enteros: (números completos y sus negativos), de tipo int. Son convenientes para aplicaciones que trabajen con datos numéricos. Se almacenan internamente en 2 bytes (16 bits) de memoria.
- Variantes de enteros: El lenguaje C simula este tipo de dato tan importante en la estructura de control (if, while...). Para ello utiliza el tipo de dato int. C interpreta todo valor distinto de cero como “verdadero” y el valor 0 “falso”.
- Reales: El punto flotante representan número reales que contienen una coma o punto decimal.
- Caracteres: Es cualquier elemento de un conjunto de caracteres predefinidos o alfabeto. Procesa datos utilizando el tipo de dato char. En la estructura array, se utiliza para almacenar cadenas de caracteres. Existen caracteres que tienen un propósito especial, no se pueden escribir utilizando el método normal. C proporciona secuencias de escape.

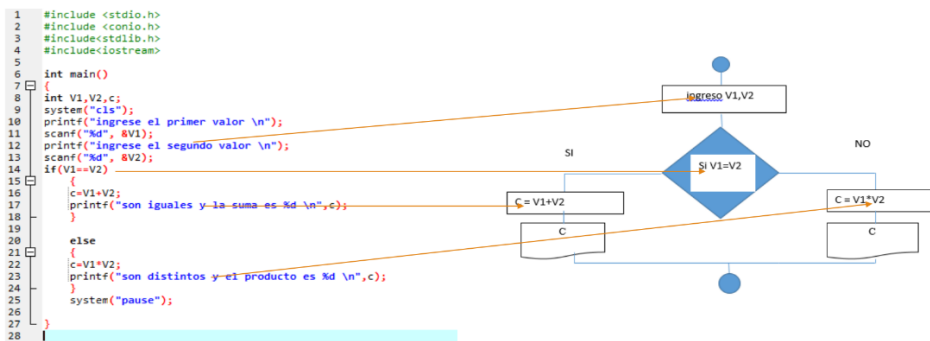
7. ¿Qué tipos de estructuras conoce? Ejemplifique.

Los tipos de estructuras que conozco son:

Alternativa simple; Se evalúa una condición, ejecutándose un grupo de sentencias si el resultado es “verdadero”, y no ejecutándose este grupo de sentencias si el resultado es “falso”.

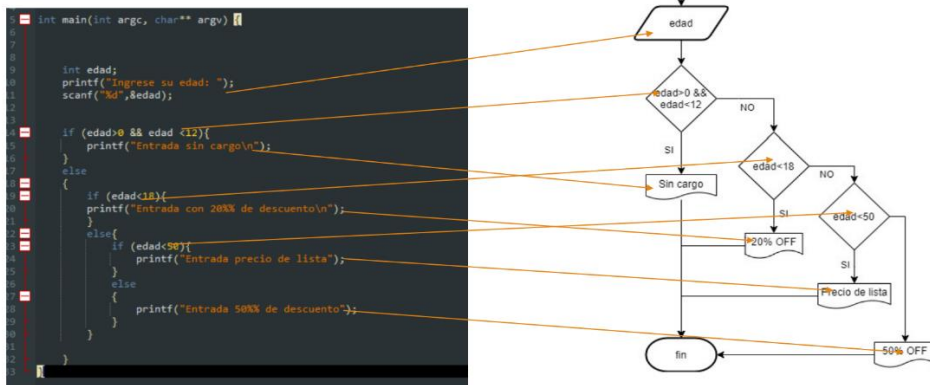


Alternativa doble; Se evalúa la condición, ejecutándose un grupo de sentencias si el resultado es “verdadero”, y ejecutándose otro grupo alternativo de sentencias si el resultado es “falso”.

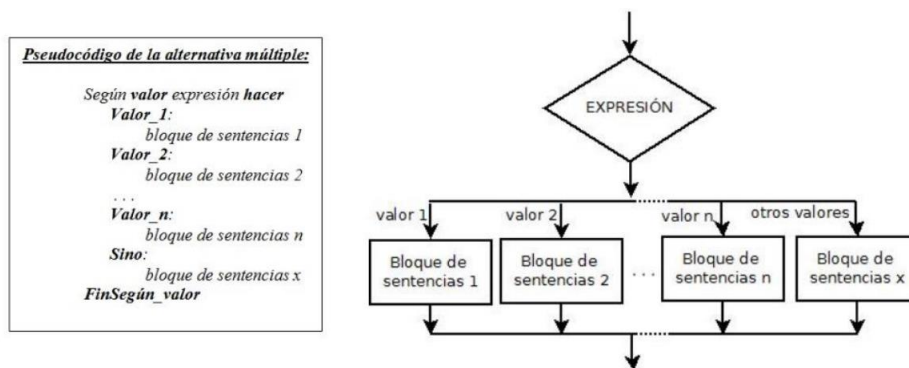


Alternativas anidadas; También es posible utilizar la instrucción **Si-Sino** para diseñar estructuras de selecciones entre más de dos alternativas. Esto se consigue mediante las estructuras anidadas, donde tanto la rama **Si** como la **Sino** pueden contener a su vez otra instrucción **Si-Sino**, y así sucesivamente un número determinado de veces.

Alternativas anidadas



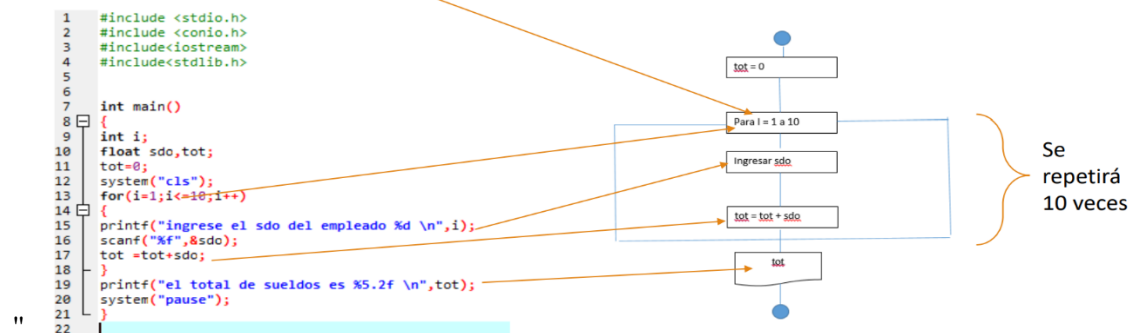
Alternativa múltiple; Se evalúa una expresión con múltiples pero finitos resultados, ejecutándose en función del resultado de la expresión, un grupo de sentencias entre múltiples posibles.



Ciclo exacto (FOR); Posee internamente un “contador” que incrementa la cantidad de veces que se ejecutó el programa y con un condicional que al llegar al número deseado me saca del ciclo.

CICLO EXACTO (FOR)

- Posee internamente un **contador** que va contando la cantidad de veces que se ejecutó el programa y con un condicional que al llegar al número deseado me saca del ciclo



Cabe importante mencionar que también existen estructuras de repetición denominadas (while-do) (do-while).

8. ¿Qué es una función? Compárela con un procedimiento

Una función es una colección de sentencias que ejecutan una tarea específica. En un programa en C se distinguen dos clases de funciones, funciones definidas por el usuario y las funciones de bibliografía.

Los procedimientos solo procesan datos, pero no devuelven ningún resultado. Tanto las funciones como los procedimientos deben ser declarados primero, para luego ser definidos.

La petición corresponde al “llamado” de la función para hacer una tarea en particular.