



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Karina García Morales

Asignatura: Fundamentos de Programación

Grupo: 20

No. de práctica(s): No. 4 - Diagramas de flujo.

Integrante(s): López Olmedo Ernesto Yael

No. de lista o brigada: 27

Semestre: 2024-1

Fecha de entrega: 19 de Septiembre 2023

Observaciones:

CALIFICACIÓN: _____

Introducción

Para esta práctica se llevará a cabo la elaboración de diagramas de flujo, basados en los conocimientos dados en clase acerca de los diagramas de flujo, para finalmente hacer el desarrollo gráfico de un diagrama de flujo con lo que el cual el alumno podrá observar y comprender las posibilidades de desarrollo de un programa de forma adecuada, evitando errores de ambigüedad, optimización, orden y cronología.

Se planea la práctica mediante la elaboración de un diagrama de flujo que represente la solución algorítmica de un problema, en el cual requiere el uso de la estructura de control condicional. A mismo tiempo se hará la representación gráfica de la solución de un problema, a través de un diagrama de flujo, en el cual requiere el uso de la estructura de control iterativa

Objetivo

El alumno elaborará diagramas de flujo que representen soluciones algorítmicas vistas como una serie de acciones que comprendan un proceso gráfico.

Desarrollo

Un diagrama de flujo es una representación gráfica de un proceso computacional para la resolución de un problema específico. Mostrándose la construcción modular específica de componentes a través de figuras, se puede llevar a cabo la correcta función y operatividad del algoritmo a desarrollar. Para ello se debe comprender las formas de los diagramas de flujo:

1. Inicio/Final:

Todo diagrama debe presentar un inicio y un final. Se representa de la forma:



Figura 1.1 - Inicio/Final.

2. Conectores:

Indican la dirección del diagrama, estas deben ser verticales y horizontales (exclusivamente). A cada símbolo solo se le puede asignar una dirección de flujo. Se representa de la forma:



Figura 1.2 - Conectores.

3. Lectura de datos:

Expresa la lectura de datos (datos de entrada). Se representa de la forma:



Figura 1.3 - Lectura de datos.

4. Procesos:

Expresan asignaciones u operaciones. Se representan de la forma:

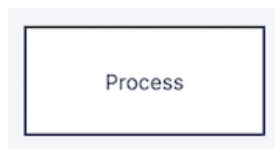


Figura 1.4 Procesos.

5. Decisiones:

Posee dos caminos, se define la dirección a través de una condición que arrojará un valor verdadero o falso (cumple o no). Se representa de la forma:

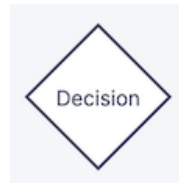


Figura 1.5 Decision.

6. Escritura:

Presenta el o los posibles resultados, pueden ser estos numéricos, escritos o mixtos. Se representa de la forma:

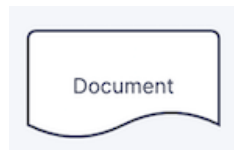


Figura 1.6 - Escritura.

7. Conexión:

Permite hacer la división ya sea en una página horizontal, para respetar la cronología que se debe elaborar en el diagrama(izquierda a derecha y arriba a abajo) o hacer un salto de página.

a. Dentro de la página.

Se representa de la forma:



Figura 1.7 - Conexión dentro de la página.

b. Fuera de la página.

Se representa de la forma:

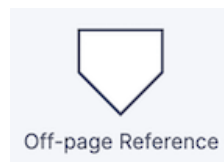


Figura 1.8 - Conexión fuera de la página.

8. Módulo problema:

Permite llamar a otras funciones, módulos o algoritmos que estén creados en otro directorio o archivo. Se representa de la forma:

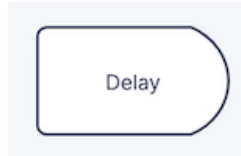


Figura 1.9 - Módulo problema.

9. Decisión múltiple:

Contiene un seleccionador, que permite tomar diversas ramas para continuar o procesar alguna función en el diagrama. Se representa de la forma:



Figura 1.10 - Decisión múltiple.

Estructuras de control de flujo.

- Estructura de control secuencial.

Resultan de la concatenación de una función primaria de la cual se va desencadenando otra(s).

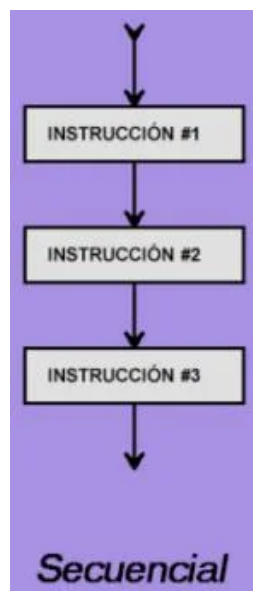


Figura 1.11 - Estructura de control secuencial.

- Estructura de control condicionales (o selectivas).
Son conocidas como cíclicas, ya que involucra una expresión lógica que permitirá el número de veces que se ejecutara la operación.

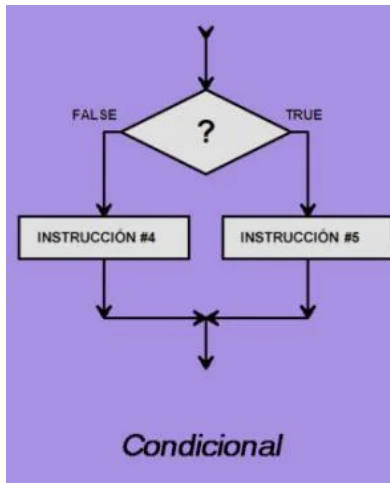


Figura 1.12 - Estructura de control condicional.

- Estructura de control iterativas o repetitivas.

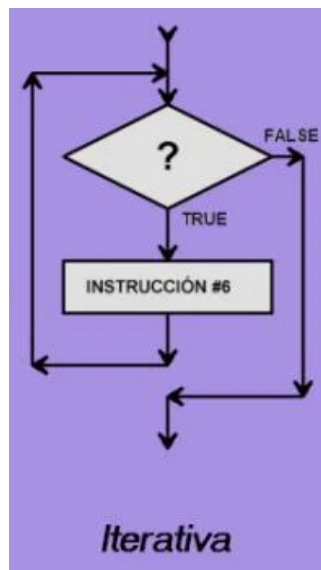


Figura 1.13 - Estructura de control iterativas.

Para esto existen dos tipos:

1. Mientras:

Primero comprueba que la condición (si Sí) se cumple, se hace la acción, se vuelve a repetir hasta que esta condición se deje de cumplir (romper el ciclo) y pasa al siguiente paso (si No).

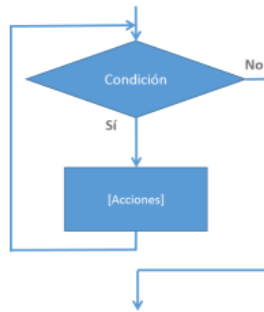


Figura 1.14 - Estructura de control iterativa - Mientras.

2. Hacer-Mientras:

Ejerce la instrucción, luego comprueba si la función es correcta y pasa a realizar la acción nuevamente, de manera cíclica mientras no resulte falsa la instrucción. Siendo una vez falsa se rompe el sistema y pasa a la siguiente acción.

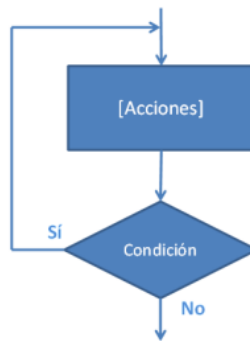


Figura 1.13 - Estructura de control iterativa Hacer-Mientras.

Laboratorio

¿Cómo declarar una variable? (Entrada de variables/valores)

Dependiendo de la variable se debe analizar el tipo de dato que deseamos obtener: entero, real, carácter, cadena, cadena, booleano.

```
var numParking = Entero
```

Corrección de ejemplos.

→ Ejemplo 1:

Ejemplo

La Figura 12 muestra el diagrama de flujo que usa la estructura de control condicional SELECCIONAR-CASO donde, de acuerdo con el valor de la variable a, escribe un mensaje.

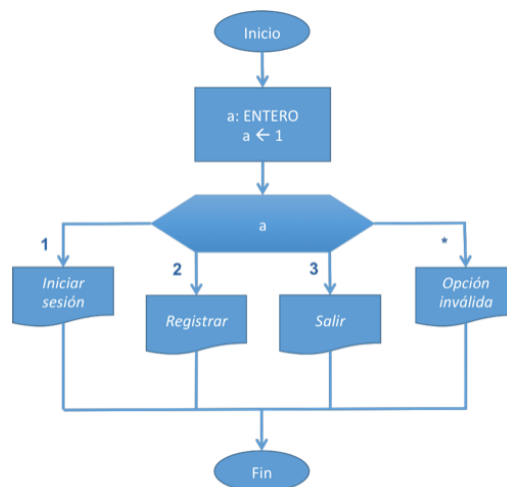


Figura 1.14 - Ejemplo figura 12.

Corrección: En el módulo de ingreso de variables y establecer su tipo, se debe agregar las variables faltantes del menú. Se debe agregar una impresión que pida el ingreso de una variable y agregar otra entrada para la variable a trabajar. Además las impresiones no están entre comillas, esto se puede tomar que su interior es una variable, marcando error.

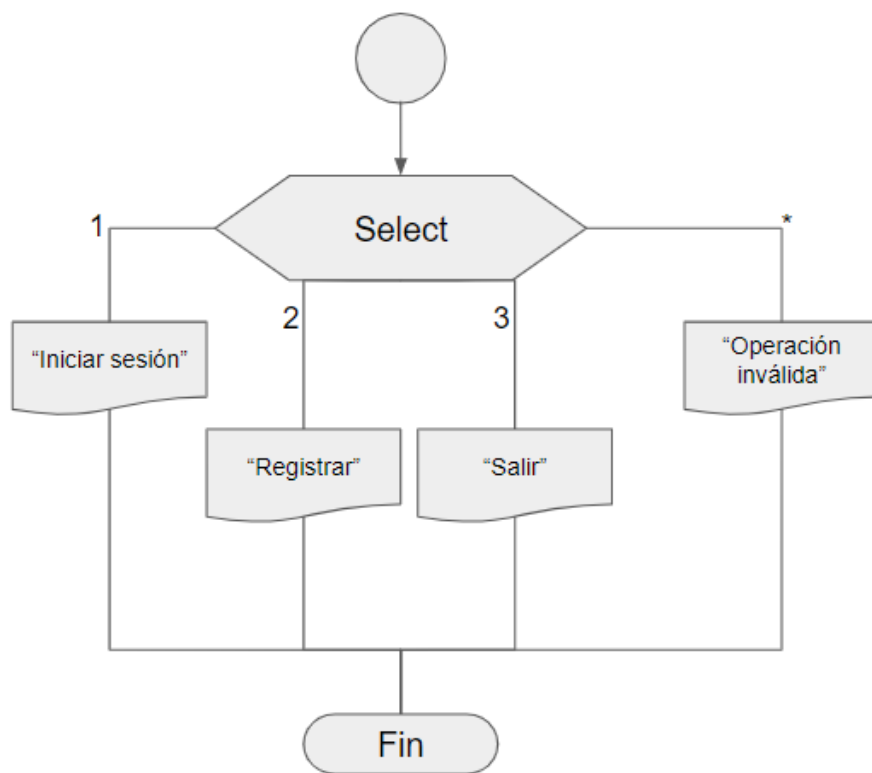
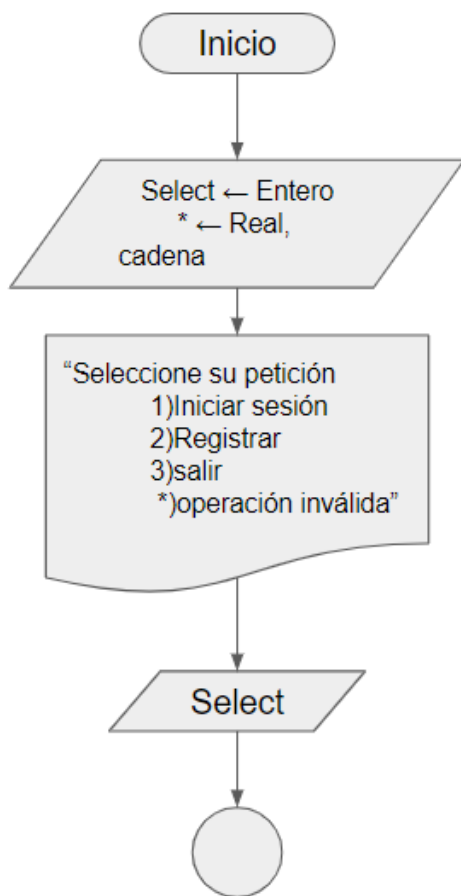


Figura 1.15 - Ejemplo figura 12/ Corrección.

Ejercicio en clase:

Genere un diagrama de flujo de un menú con las siguientes indicaciones:

1. Imprime los números del 1 al 10.
2. Calcula la suma de dos variables diferentes y diferentes de 0.
3. Ingresa a tu menú que trabajaste en clase (Lunes).
4. Salida del menú. (op!=0 fin del proceso, sí sí continúa)

Datos de entrada: Select

Restricciones: Sólo pueden ser caracteres. Solo con la cuarta opción(d) termina el programa, de lo contrario continua en bucle. Tiene que tener funciones hijas.

Datos de salida: Cuenta de números, suma, menú o fin.

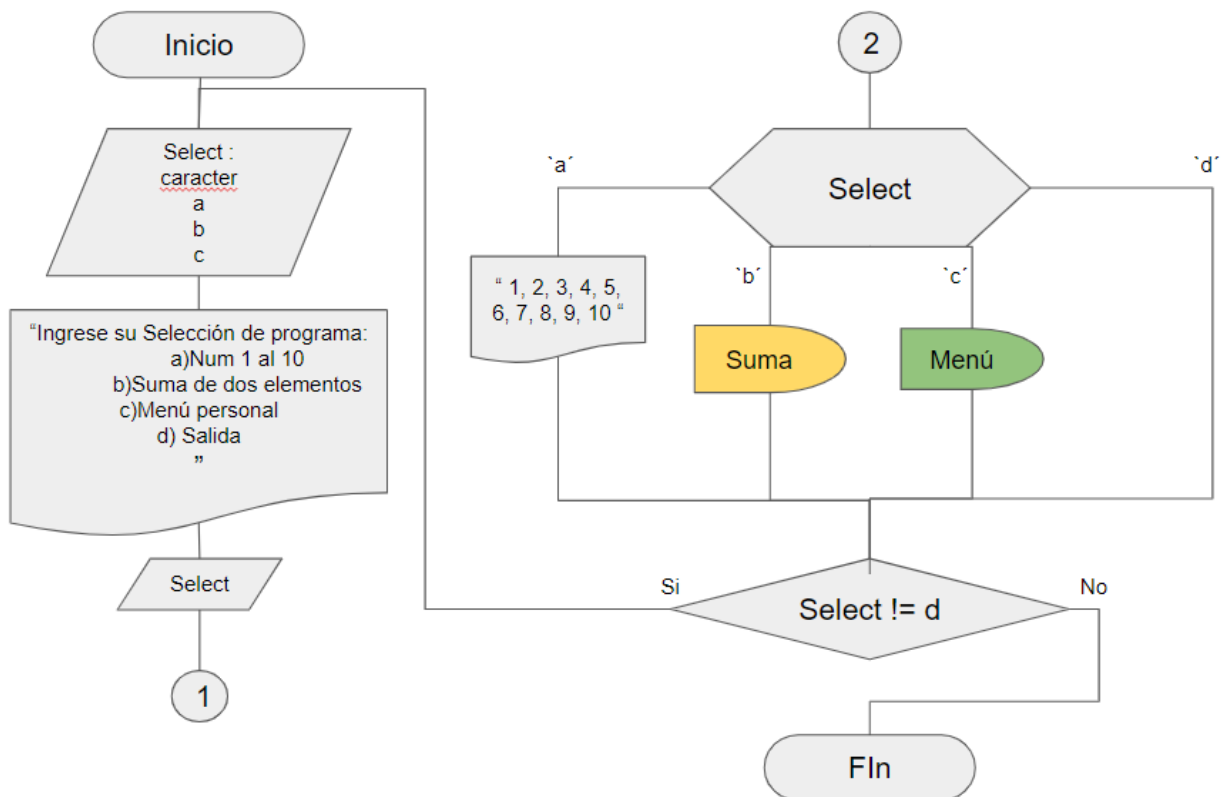


Figura 1.16 - Función padre.

Función Suma y Menú:

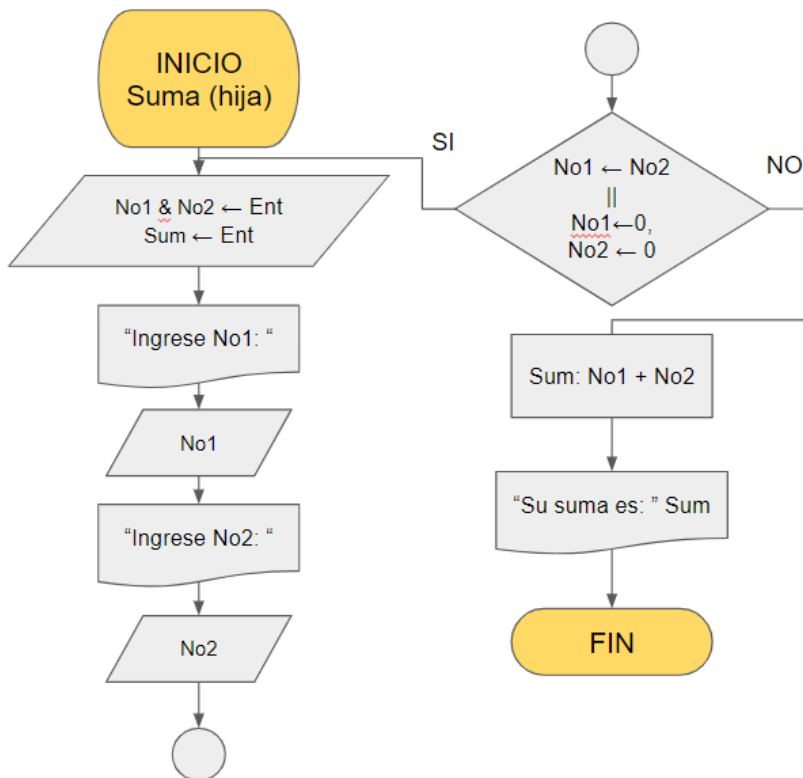


Figura 1.17 - Función hija Suma .

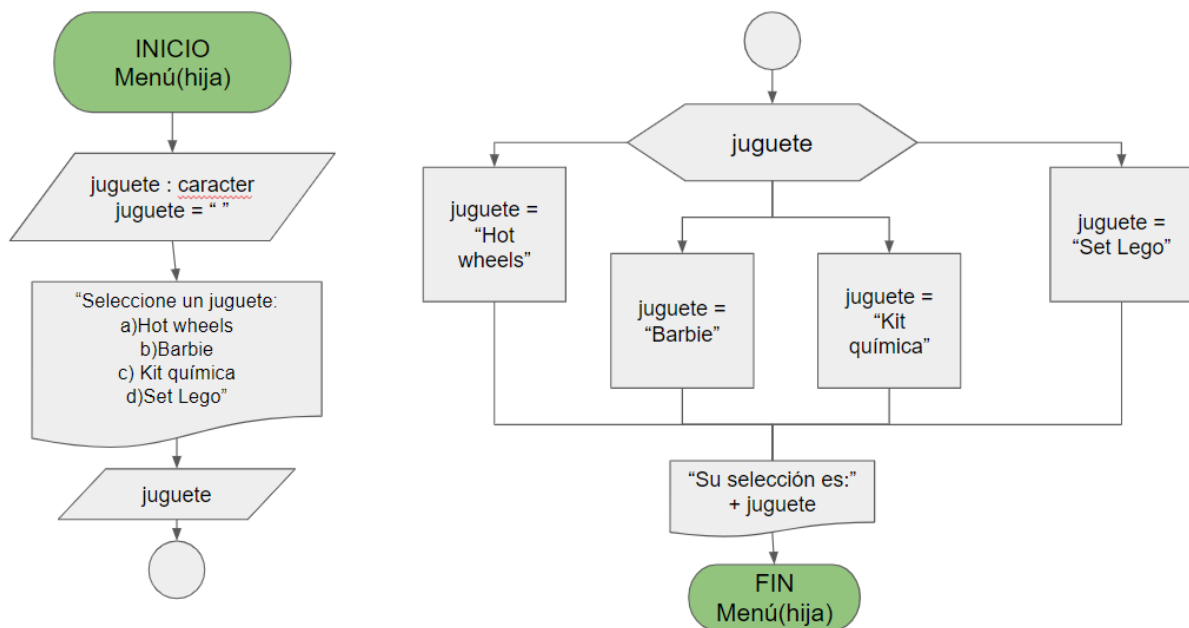


Figura 1.16 - Función hija Menú.

Conclusión

Esta práctica representa ya un cúmulo de los conocimientos previos de las prácticas pasadas, con los cuales resultan pilares para la resolución de este tema, ya que se es necesario para la comprensión lógica que requiere hacer un algoritmo, ser objetivo, dar un order, evitar ambigüedades, bucles sin fin, entender las diferencias de redacción cualitativa y cuantitativa. Por lo que la elaboración de esta práctica consistió en un diseño gráfico de un algoritmo, a través de los cuales se comprendió la forma de crear resoluciones con módulos específicos para la expresión de acciones dentro del algoritmo, permitiendo reconocer las diversas formas en las que se puede desarrollar un algoritmo.

Referencias

- Templos, A.(2022).Manual de prácticas del laboratorio de Fundamentos de programación. México: UNAM.