

ECBTI /Zona Caribe / Ingeniería de Sistemas

Pensamiento Computacional y Programación en Python

Fernando J. Diaz – Arellys Correa

Corozal, 20 de Julio 2021

1 Presentación de los Contenidos

Temáticas

- ✓ Introducción Algoritmos
- ✓ Variables y Tipos de Datos
- ✓ Diagramas de Flujo y Pseudocódigo
- ✓ Introducción Software DFD y PSeInt
- ✓ Estructuras Secuenciales
- ✓ Estructuras Condicionales
- ✓ Estructuras Repetitivas
- ✓ Introducción al Leguaje de Programación Python
- ✓ Instalación y configuración de Python
- ✓ Variables, tipos de datos y casting, Operaciones numéricas
- ✓ Estructuras condicionales y flujo de control
- ✓ Funciones, Diccionarios, Tuplas, Listas
- ✓ Ejercicios de Aplicación

Programas

- ✓ DFD
- ✓ PSeInt
- ✓ Python



2 Plan de Trabajo por Semana

Sábado 24 de Julio

- ✓ Introducción Algoritmos
- ✓ Variables y Tipos de Datos
- ✓ Diagramas de Flujo y Pseudocódigo
- ✓ Introducción Software DFD y PSeInt
- ✓ Estructuras Secuenciales
- ✓ Ejercicios de Aplicación

Sábado 24 de Julio

- ✓ Estructuras Condicionales
- ✓ Si – Entonces
- ✓ Según
- ✓ Ejercicios de Aplicación

Sábado 31 de Julio

- ✓ Estructuras Repetitivas
- ✓ Ciclo Para
- ✓ Ciclo Mientras
- ✓ Ciclo Repetir
- ✓ Validación de entrada de datos Numéricos
- ✓ Ejercicios de Aplicación

Encargado: Ing. Fernando J. Diaz M

4 Presentación Docentes

Fernando J Diaz M



Ingeniero de Sistemas
Esp. Administración de la Informática Educativa
Mg. Gestión de la Tecnología Educativa
Docente ECBTI UNAD
Líder de Semillero Investigación
Engine CCAV Corozal

Arellys Correa



Ingeniera de Sistemas
Esp. Informática y Telemática
Mg. In Online Education
Docente ECBTI UNAD
Líder de Semillero Investigación
Innova CCAV Cartagena

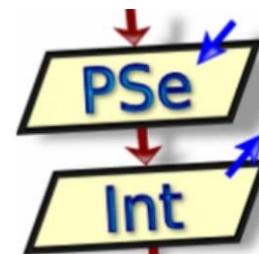
5 Desarrollo de los Contenidos

Sábado 24 de Julio

- ✓ Introducción Algoritmos
- ✓ Variables y Tipos de Datos
- ✓ Diagramas de Flujo y Pseudocódigo
- ✓ Introducción Software DFD y PSeInt
- ✓ Estructuras Secuenciales
- ✓ Ejercicios de Aplicación

Link de Descarga de los Software
PseInt y DFD

<https://bit.ly/2U63dAY>



5 Desarrollo de los Contenidos

¿Qué son los Algoritmos?



“Un algoritmo es una serie de **pasos organizados** que describe el **proceso que se debe seguir**, para dar **solución a un problema** específico”.

La palabra **algoritmo** se deriva de la traducción al latín de la palabra árabe *al-khowarizmi* o *al-Jwārizmī*, nombre de un matemático y astrónomo árabe que escribió un tratado sobre manipulación de números y ecuaciones en el siglo IX.



5 Desarrollo de los Contenidos

Tipos de Algoritmos

- **Cualitativos:** Son aquellos en los que se describen los pasos utilizando palabras.
- **Cuantitativos:** Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso.



5 Desarrollo de los Contenidos

Técnicas para Formular un Algoritmos

- **Gráficos:** Son las representaciones gráficas de las operaciones que realiza un algoritmo (diagrama de flujo).
- **No Gráficos:** Representa en forma descriptiva las operaciones que debe realizar un algoritmo (pseudocódigo).



5 Desarrollo de los Contenidos

Pasos para la Solución de Problemas por Medio de Algoritmos



- ¿Es difícil trabajar con algoritmos?
- ¿Cómo puedo aprender a programar?
- ¿Qué lenguaje se recomienda para iniciar en la programación?
- ¿Cual es el mejor lenguaje de programación?
- ¿Cómo puedo conseguir habilidades en programación?



5 Desarrollo de los Contenidos

Pasos para la Solución de Problemas por Medio de Algoritmos

Definición del Problema

Esta fase está dada por el enunciado del problema, el cual requiere una definición clara y precisa. Es importante que se conozca lo que se desea que realice la computadora; mientras esto no se conozca del todo no tiene mucho caso continuar con la siguiente etapa.



5 Desarrollo de los Contenidos

Pasos para la Solución de Problemas por Medio de Algoritmos

Análisis del Problema

Una vez que se ha comprendido lo que se desea de la computadora, es necesario definir:

- Los datos de entrada.
- Cual es la información que se desea producir (salida)
- Los métodos y fórmulas que se necesitan para procesar los datos.

Una recomendación muy practica es el que nos pongamos en el lugar de la computadora y analicemos que hay la necesidad que nos ordenen y en que secuencia, para producir los resultados esperados.



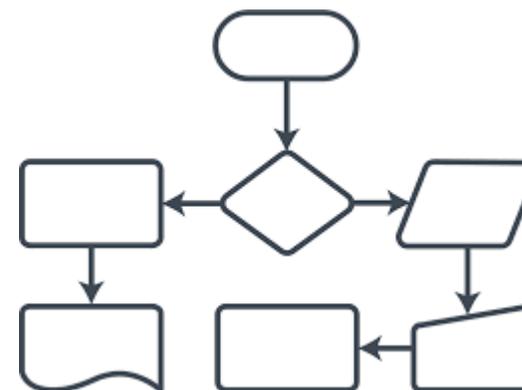
5 Desarrollo de los Contenidos

Pasos para la Solución de Problemas por Medio de Algoritmos

Diseño del Algoritmo

Las características de un buen algoritmo son:

- Tener un punto particular de inicio.
- Ser definido, no debe permitir dobles interpretaciones.
- Ser general, es decir, soportar la mayoría de las variantes que se puedan presentar en la definición del problema.
- Ser finito en tamaño y tiempo de ejecución.



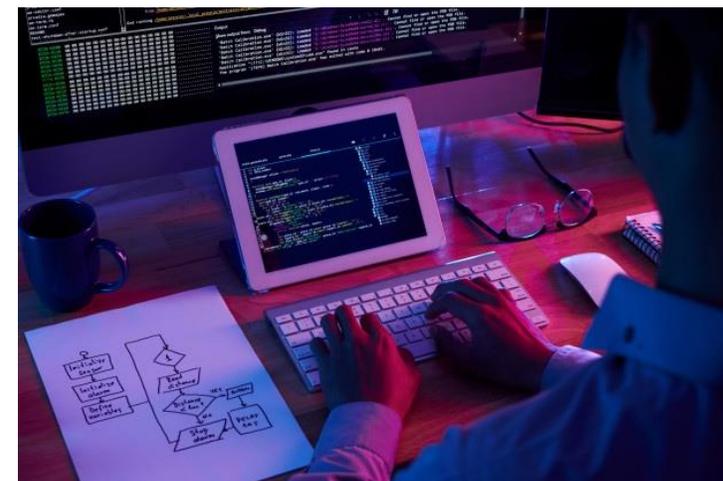
5 Desarrollo de los Contenidos

Pasos para la Solución de Problemas por Medio de Algoritmos

Codificación

La codificación es la operación de escribir la solución del problema (de acuerdo a la lógica del diagrama de flujo o pseudocódigo), en una serie de instrucciones detalladas, en un código reconocible por la computadora, la serie de instrucciones detalladas se le conoce como código fuente.

Lenguajes de Programación: Es un conjunto de símbolos, caracteres y reglas (programas) que le permiten a las personas comunicarse con la computadora. Los lenguajes de programación tienen un conjunto de instrucciones que nos permiten realizar operaciones de entrada/salida, cálculo, manipulación de textos, lógica/comparación y almacenamiento/recuperación.



5 Desarrollo de los Contenidos

Pasos para la Solución de Problemas por Medio de Algoritmos

Prueba y Depuración

Los errores humanos dentro de la programación de computadoras son muchos y aumentan considerablemente con la complejidad del problema. El proceso de identificar y eliminar errores, para dar paso a una solución sin errores se le llama **depuración**.

La depuración o prueba resulta una tarea tan creativa como el mismo desarrollo de la solución, por ello se debe considerar con el mismo interés y entusiasmo.



5 Desarrollo de los Contenidos

Pasos para la Solución de Problemas por Medio de Algoritmos

Documentación

Es la guía o comunicación escrita en sus variadas formas, ya sea en enunciados, procedimientos, dibujos o diagramas.

A menudo un programa escrito por una persona, es usado por otra. Por ello la documentación sirve para ayudar a comprender o usar un programa o para facilitar futuras modificaciones (mantenimiento).

La documentación se divide en tres partes: Interna (Comentarios), Externa (Descripción del Problema, Autor, Diagrama de Flujo, Código Fuente) y Manual de Usuario (Describe paso a paso la manera como funciona el programa, con el fin de que el usuario obtenga el resultado deseado.)



5 Desarrollo de los Contenidos

Pasos para la Solución de Problemas por Medio de Algoritmos

Mantenimiento

Se lleva acabo después de terminado el programa, cuando se detecta que es necesario hacer algún cambio, ajuste o complementación al programa para que siga trabajando de manera correcta. Para poder realizar este trabajo se requiere que el programa este correctamente documentado.

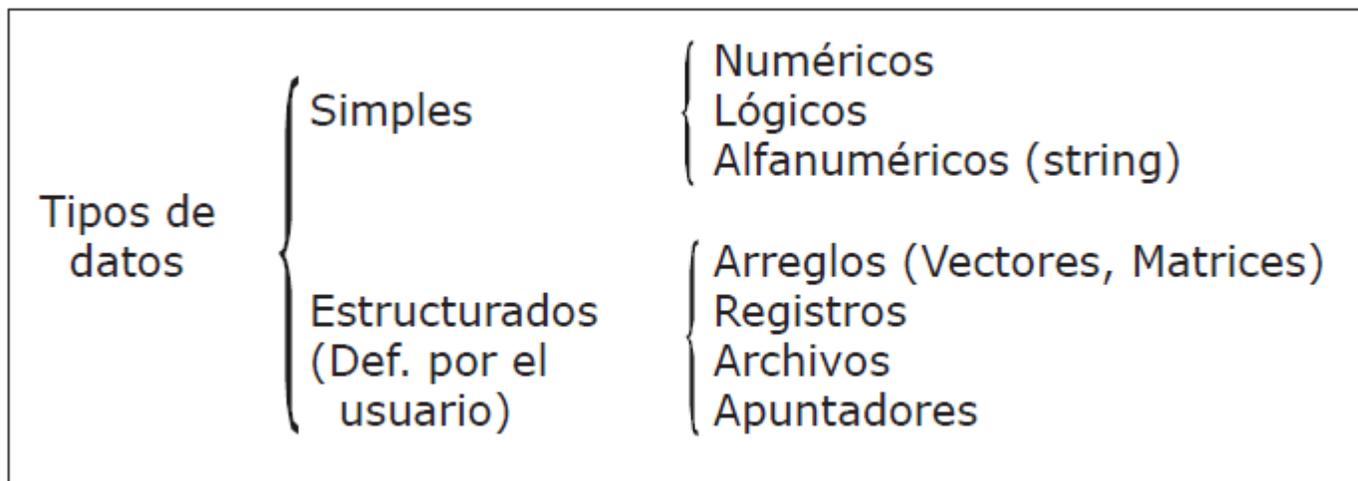


5 Desarrollo de los Contenidos

Variables y Tipos de Datos

Tipos de Datos

Todos los datos tienen un tipo asociado con ellos. Un dato puede ser un simple carácter, tal como "b", un valor entero tal como 35. El tipo de dato determina la naturaleza del conjunto de valores que puede tomar una variable.



5 Desarrollo de los Contenidos

Variables y Tipos de Datos

Operadores y Operandos

Operadores: Son elementos que relacionan de forma diferente, los valores de una o mas variables y/o constantes. Es decir, los operadores nos permiten manipular valores.

Tipos de Operadores	{	Aritméticos Relacionales Lógicos
---------------------	---	--

5 Desarrollo de los Contenidos

Variables y Tipos de Datos

Operadores y Operandos

Operadores Aritméticos: Los operadores aritméticos permiten la realización de operaciones matemáticas con los valores (variables y constantes).

Los operadores aritméticos pueden ser utilizados con tipos de datos enteros o reales. Si ambos son enteros, el resultado es entero; si alguno de ellos es real, el resultado es real.

<i>Operadores Aritméticos</i>	
+	Suma
-	Resta
*	Multiplicación
/	División
Mod	Modulo (residuo de la división entera)

5 Desarrollo de los Contenidos

Variables y Tipos de Datos

Operadores y Operandos

Operadores Relacionales: Se utilizan para establecer una relación entre dos valores. Compara estos valores entre si y esta comparación produce un resultado de certeza o falsedad (verdadero o falso).

Los operadores relacionales comparan valores del mismo tipo (numéricos o cadenas). Tienen el mismo nivel de prioridad en su evaluación.

Operadores Relacionales

>	Mayor que
<	Menor que
> =	Mayor o igual que
< =	Menor o igual que
< >	Diferente
=	Igual

5 Desarrollo de los Contenidos

Variables y Tipos de Datos

Operadores y Operandos

Operadores Lógicos: Estos operadores se utilizan para establecer relaciones entre valores lógicos.

Estos valores pueden ser resultado de una expresión relacional.

<i>Operadores Lógicos</i>	
And	Y
Or	O
Not	Negación

5 Desarrollo de los Contenidos

Variables y Tipos de Datos

Identificadores

Los identificadores representan los datos de un programa (**constantes**, **variables**, **tipos de datos**). Un identificador es una secuencia de caracteres que sirve para identificar una posición en la memoria de la computadora, que nos permite acceder a su contenido.

Ejemplo: Nombre, Num_hrs, Calif2

Reglas para formar un identificador

- Debe comenzar con una letra, mayúsculas o minúsculas, y no deben contener espacios en blanco.
- Letras, dígitos y caracteres como la subraya (_) están permitidos después del primer carácter.
- La longitud de identificadores puede ser de varios caracteres. Pero es recomendable una longitud promedio de 8 caracteres.

5 Desarrollo de los Contenidos

Variables y Tipos de Datos

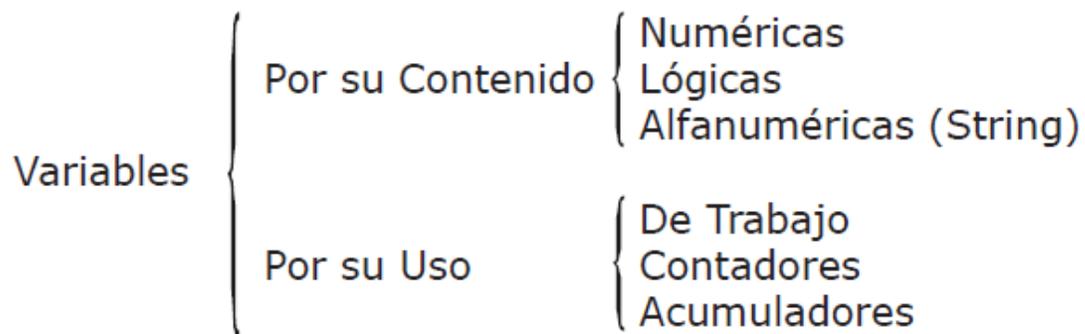
Constante: Una constante es un dato numérico o alfanumérico que no cambia durante la ejecución del programa.

Ejemplo: $\pi = 3.1416$

Variable: Es un espacio en la memoria de la computadora que permite almacenar temporalmente un dato durante la ejecución de un proceso, su contenido puede cambiar durante la ejecución del programa. Para poder reconocer una variable en la memoria de la computadora, es necesario darle un nombre con el cual podamos identificarla dentro de un algoritmo.

Ejemplo: $\text{area} = \pi * \text{radio}^2$

Clasificación de las Variables



5 Desarrollo de los Contenidos

Variables y Tipos de Datos

Por Su Contenido

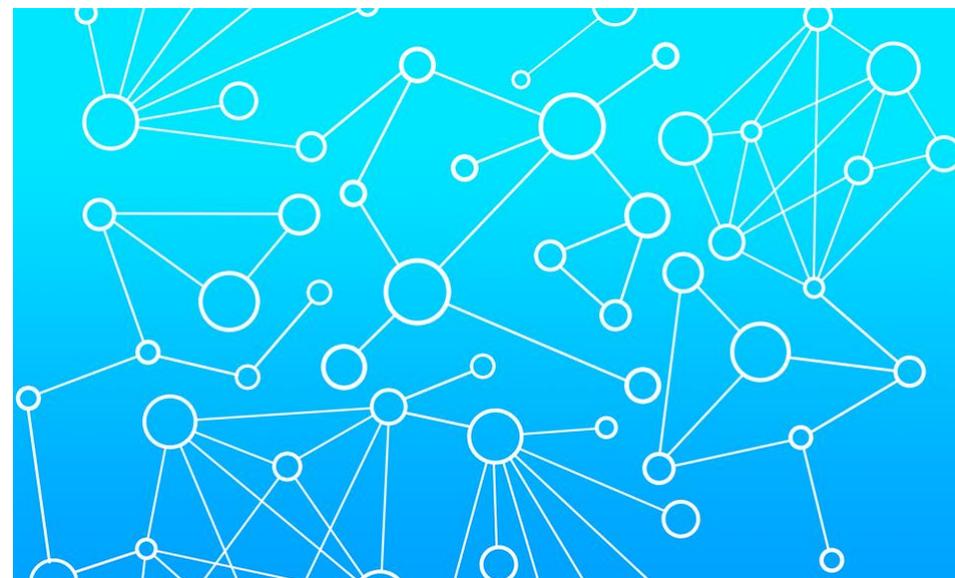
- **Variable Numéricas:** Son aquellas en las cuales se almacenan valores numéricos, positivos o negativos, es decir almacenan números del 0 al 9, signos (+ y -) y el punto decimal.

Ejemplo: `iva=0.15 pi=3.1416 costo=2500`

- **Variables Lógicas:** Son aquellas que solo pueden tener dos valores (cierto o falso) estos representan el resultado de una comparación entre otros datos.

- **Variables Alfanuméricas:** Esta formada por caracteres alfanuméricos (letras, números y caracteres especiales).

Ejemplo: `letra="a" apellido="lopez" direccion="Av. Libertad #190"`



5 Desarrollo de los Contenidos

Variables y Tipos de Datos

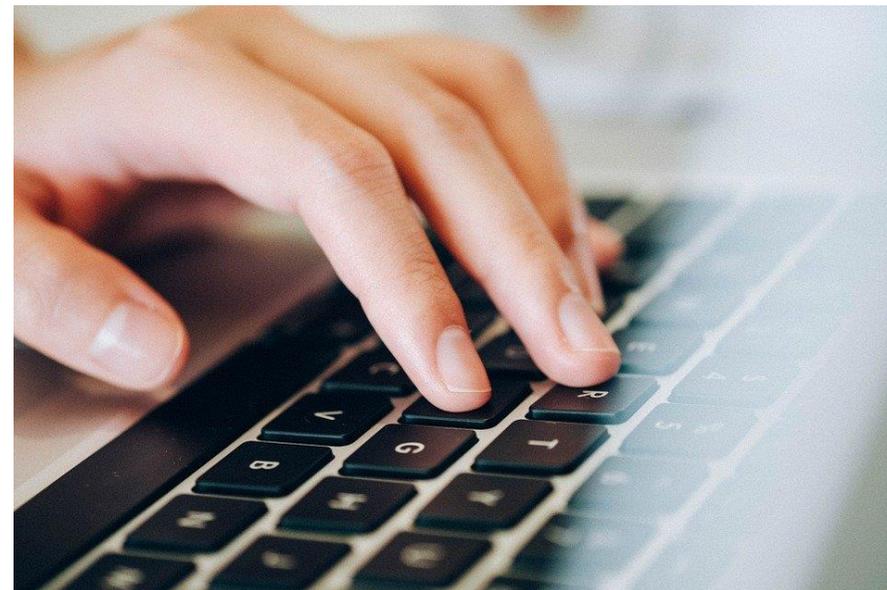
Por Su Uso

- **Variables de Trabajo:** Variables que reciben el resultado de una operación matemática completa y que se usan normalmente dentro de un programa.

Ejemplo: $\text{suma} = a + b/c$

- **Contadores:** Se utilizan para llevar el control del número de ocasiones en que se realiza una operación o se cumple una condición. Con los incrementos generalmente de uno en uno.

- **Acumuladores:** Forma que toma una variable y que sirve para llevar la suma acumulativa de una serie de valores que se van leyendo o calculando progresivamente.



5 Desarrollo de los Contenidos

Diagramas de Flujo y Pseudocódigo

Pseudocódigo

Mezcla de programación y lenguaje (español, inglés o cualquier otro idioma) que se emplea dentro de la programación estructurada, para realizar el diseño de un programa. En esencia, el pseudocódigo se puede definir como un lenguaje de especificaciones de algoritmos.

Es la representación de los pasos que debe seguir un algoritmo para dar solución a un problema determinado. El pseudocódigo utiliza palabras que indican el proceso a realizar.



5 Desarrollo de los Contenidos

Diagramas de Flujo y Pseudocódigo

Ventajas de utilizar un Pseudocódigo a un Diagrama de Flujo

- Ocupa menos espacio en una hoja de papel
- Permite representar en forma fácil operaciones repetitivas complejas
- Es muy fácil pasar de pseudocódigo a un programa en algún lenguaje de programación.
- Se puede observar claramente los niveles que tiene cada operación y/o ciclo.



5 Desarrollo de los Contenidos

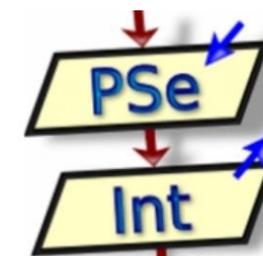
Introducción Software DFD y PSeInt

DFD: es un software diseñado para construir y analizar algoritmos. Se pueden crear diagramas de flujo de datos para la representación de algoritmos de programación estructurada a partir de las herramientas de edición que para éste propósito suministra el programa. Después de haber ingresado el algoritmo representado por el diagrama, podrá ejecutarse, analizarse y depurarse en un entorno interactivo diseñado para este fin.

PSeInt : es una herramienta para asistir a un estudiante en sus primeros pasos en programación. Mediante un simple e intuitivo pseudolenguaje en español (complementado con un editor de diagramas de flujo), le permite centrar su atención en los conceptos fundamentales de la algoritmia computacional, minimizando las dificultades propias de un lenguaje y proporcionando un entorno de trabajo con numerosas ayudas y recursos didácticos.

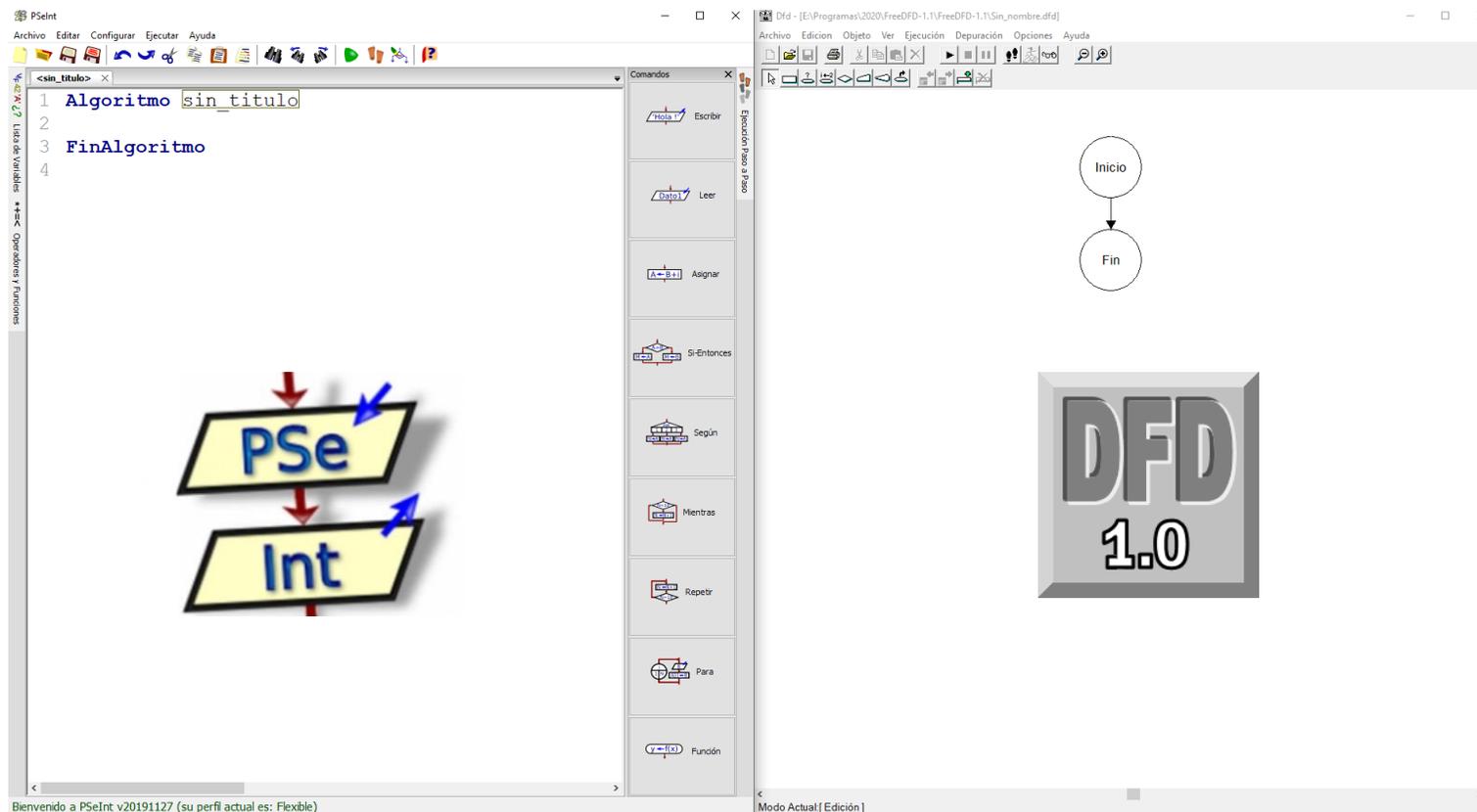
Link de Descarga de los Software
PSeInt y DFD

<https://bit.ly/2U63dAY>



5 Desarrollo de los Contenidos

Introducción Software DFD y PSeInt



The screenshot displays the PSeInt software interface. On the left, a code editor window titled 'sin_titulo' contains the following code:

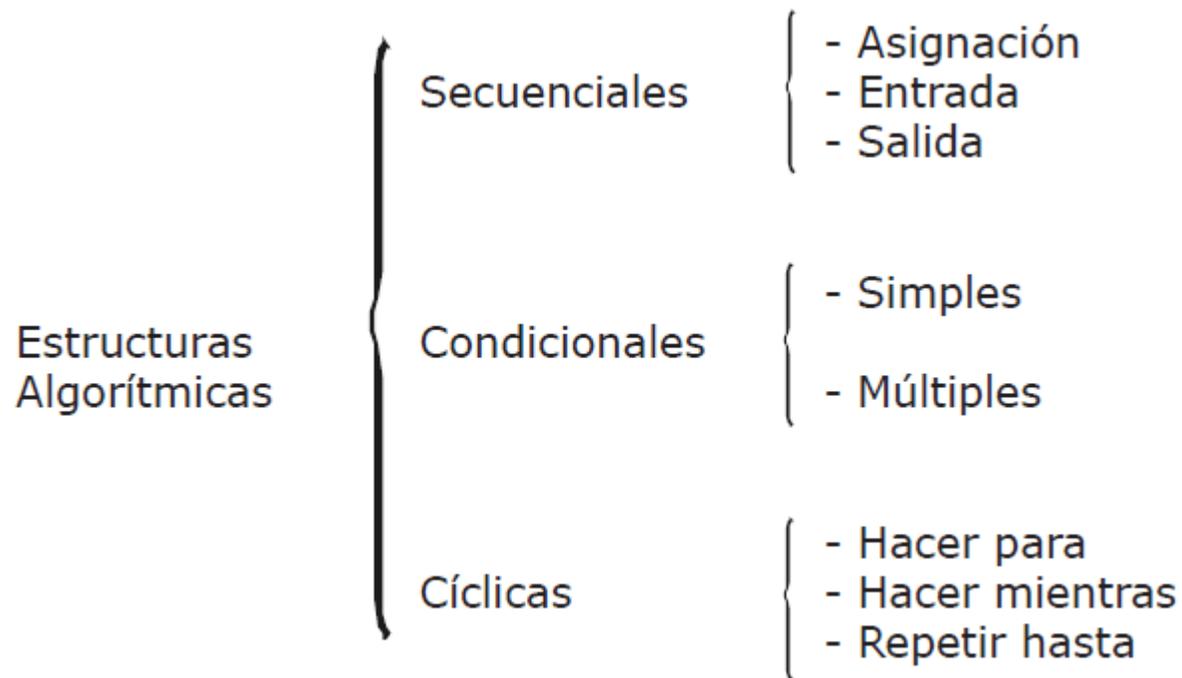
```
1 Algoritmo sin_titulo
2
3 FinAlgoritmo
4
```

Below the code editor, there is a graphic of two yellow parallelograms labeled 'PSe' and 'Int', representing the software's branding. On the right, a diagram window titled 'Dfd - [E:\Programas\2020\FreeDFD-1.1\FreeDFD-1.1\sin_nombre.dfd]' shows a simple Data Flow Diagram (DFD) with two circular nodes: 'Inicio' (Start) and 'Fin' (End), connected by a downward arrow. A large grey box with the text 'DFD 1.0' is overlaid on the diagram. A vertical toolbar on the left side of the diagram window contains various symbols for creating and editing DFD elements.

5 Desarrollo de los Contenidos

Estructuras Algorítmicas

Las estructuras de operación de programas son un grupo de formas de trabajo, que permiten, mediante la manipulación de variables, realizar ciertos procesos específicos que nos lleven a la solución de problemas. Estas estructuras se clasifican de acuerdo con su complejidad en:



5 Desarrollo de los Contenidos

Estructuras Secuenciales

La estructura secuencial es aquella en la que una acción (instrucción) sigue a otra en secuencia. Las tareas se suceden de tal modo que la salida de una es la entrada de la siguiente y así sucesivamente hasta el fin del proceso. Una estructura secuencial se representa de la siguiente forma:

Inicio
Accion 1
Accion 2
.
.
Accion N
Fin

Lectura: La lectura consiste en recibir desde un dispositivo de entrada (p.ej. El teclado) un valor. Esta operación se representa en un pseudocódigo como sigue:

Leer a, b

Donde "a" y "b" son las variables que recibirán los valores

• Escritura: Consiste en mandar por un dispositivo de salida (p.ej. monitor o impresora) un resultado o mensaje. Este proceso se representa en un pseudocódigo como sigue:

Escribe "El resultado es:", R

Donde "El resultado es:" es un mensaje que se desea aparezca y R es una variable que contiene un valor.

5 Desarrollo de los Contenidos

Estructuras Secuenciales

- **Asignación:** La asignación consiste, en el paso de valores o resultados a una zona de la memoria. Dicha zona será reconocida con el nombre de la variable que recibe el valor.
- **Lectura:** La lectura consiste en recibir desde un dispositivo de entrada (p.ej. El teclado) un valor. Esta operación se representa en un pseudocódigo como sigue:

Leer a, b; Donde "a" y "b" son las variables que recibirán los valores

- **Escritura:** Consiste en mandar por un dispositivo de salida (p.ej. monitor o impresora) un resultado o mensaje. Este proceso se representa en un pseudocódigo como sigue:

Escribe "El resultado es:", R; Donde "El resultado es:" es un mensaje que se desea aparezca y R es una variable que contiene un valor.

5 Desarrollo de los Contenidos

Ejercicios de Aplicación



5 Desarrollo de los Contenidos

Sábado 24 de Julio

- ✓ Repaso de Estructuras Secuenciales
- ✓ Estructuras Condicionales
- ✓ Si – Entonces
- ✓ Según
- ✓ Ejercicios de Aplicación

Drive Workshops de Algoritmos y Programación en Python

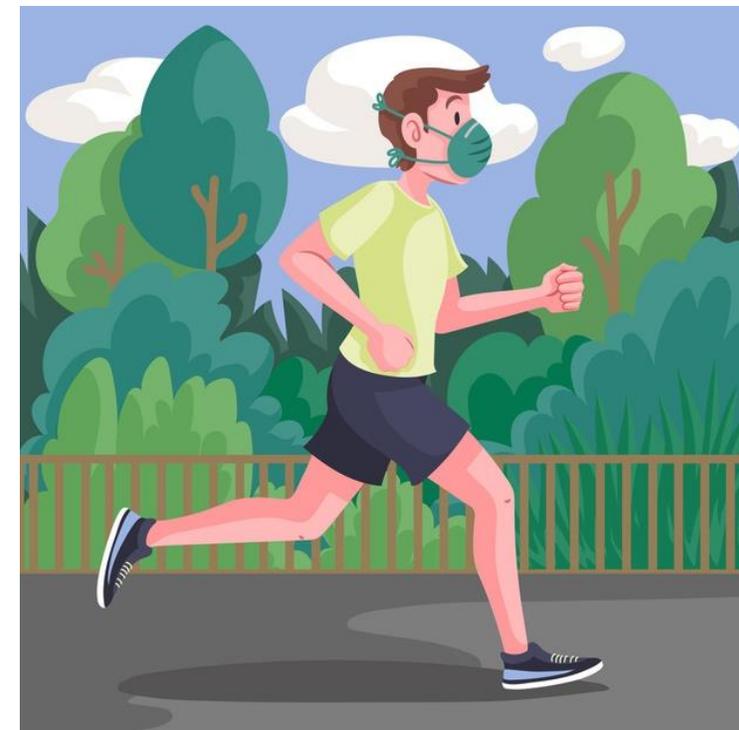
<https://bit.ly/2Umqej7>



5 Desarrollo de los Contenidos

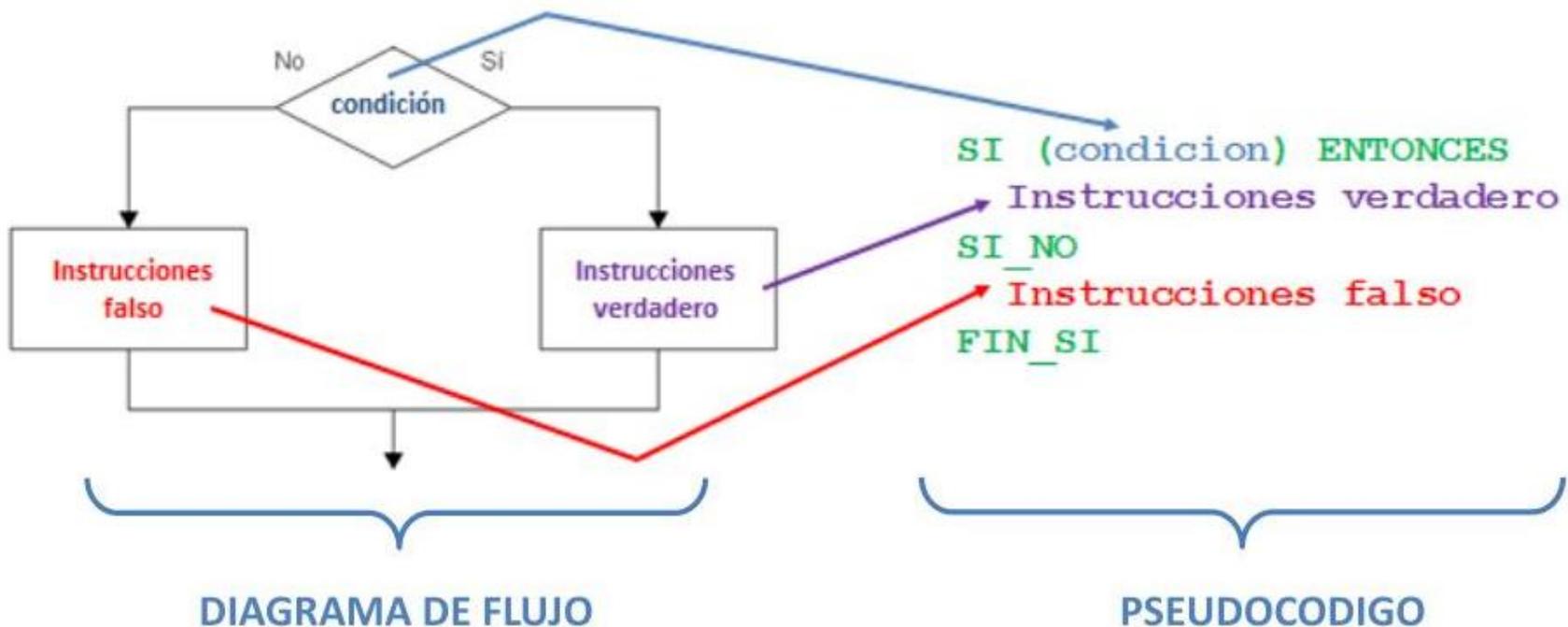
Repaso de Estructuras Secuenciales – Ejercicios de Aplicación

1. Diseñar un algoritmo que permita convertir una unidad de tiempo dada en horas – Minutos y Segundos a su equivalente en solo segundos.
2. La empresa de Taxis XYZ, tiene unas nuevas tarifas para sus clientes a los cuales les cobra por kilómetros y por minutos, el algoritmo debe determinar el total a pagar por una carrera en donde se sabe que por kilómetro se cobran \$ 500, y por minuto \$300. Nota: por ejemplo, si Juan se desplaza 5 Km y demora 10 Min, el valor de la carrera será \$5.500.
3. Un estudiante de la asignatura de algoritmos desea conocer cuál es su nota de final de semestre, se sabe que en la institución se tienen en cuenta las notas de 3 cortes académicos, el corte 1 equivale al 30% del valor de la nota final (15% correspondiente al parcial, y el restante 15% 2 notas de trabajos), el corte 2 equivale al 30% del valor de la nota final (15% correspondiente al parcial, y el restante 15% 3 notas de trabajos), y el corte 3 equivale al 40% de la nota final, (30% correspondiente al parcial, y el restante 10% 2 notas de trabajos).



5 Desarrollo de los Contenidos

Estructuras Condicionales



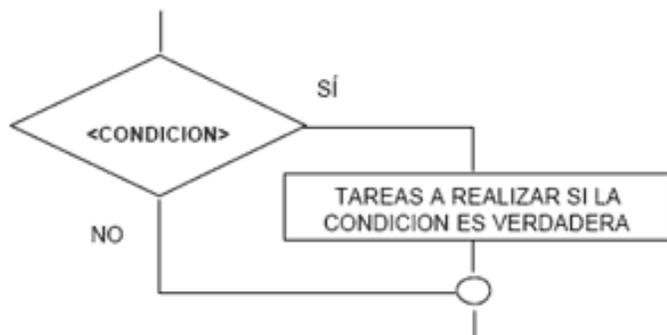
Las estructuras condicionales comparan una variable contra otro(s) valor(es), para que con base al resultado de esta comparación, se siga un curso de acción dentro del programa. Cabe mencionar que la comparación se puede hacer contra otra variable o contra una constante, según se necesite. Los tipos básicos son, simples, dobles y las múltiples.

5 Desarrollo de los Contenidos

Estructuras Condicionales – Si – Entonces

Simples: Las estructuras condicionales simples se les conoce como "Tomas de decisión". Estas tomas de decisión tienen la siguiente forma:

Si <condición> entonces
Acción(es) Verdadera
Fin-si



Dobles: Las estructuras condicionales dobles permiten elegir entre dos opciones o alternativas posibles en función del cumplimiento o no de una determinada condición. Se representa de la siguiente forma:

Si <condición> entonces
Acción(es) Verdadera
si no
Acción(es) Falsa
Fin-si



5 Desarrollo de los Contenidos

Estructuras Condicionales – Si – Entonces

Múltiples: Las estructuras de comparación múltiples, son tomas de decisión especializadas que permiten comparar una variable contra distintos posibles resultados, ejecutando para cada caso una serie de instrucciones específicas. La forma común es la siguiente:

Si <condición> entonces

Acción(es)

si no

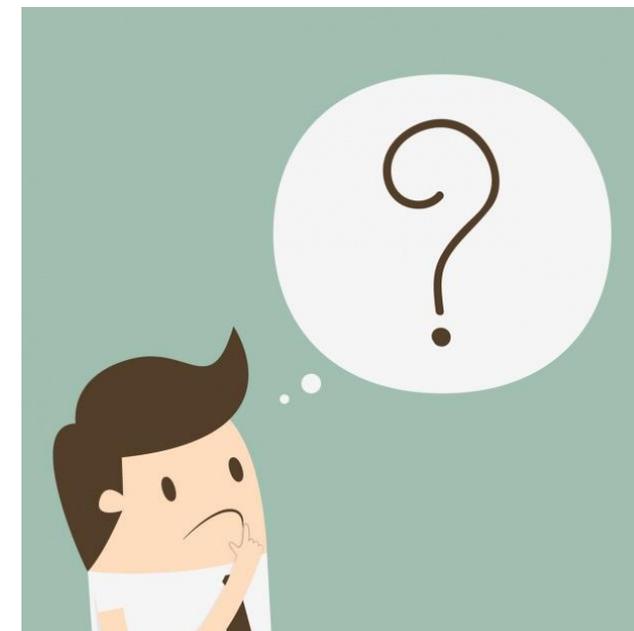
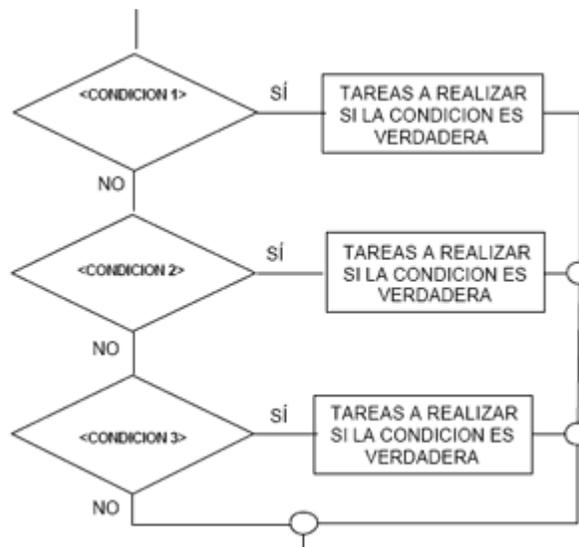
Si <condición> entonces

Acción(es)

si no

•
•
•
} Varias condiciones

Fin-si



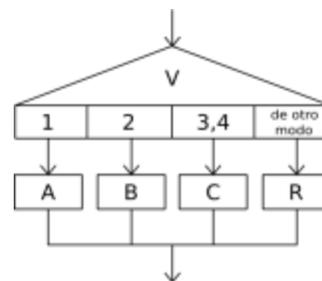
5 Desarrollo de los Contenidos

Estructuras Condicionales – Según

Es frecuente tener que comprobar más de una condición a la vez, o bien varias condiciones consecutivas. En concreto, un caso especialmente habitual es el de que una variable pueda tomar un valor de entre varios.

El bloque "De Otro Modo" es opcional: si detallamos todos los valores posibles, no sería necesario utilizarlo.

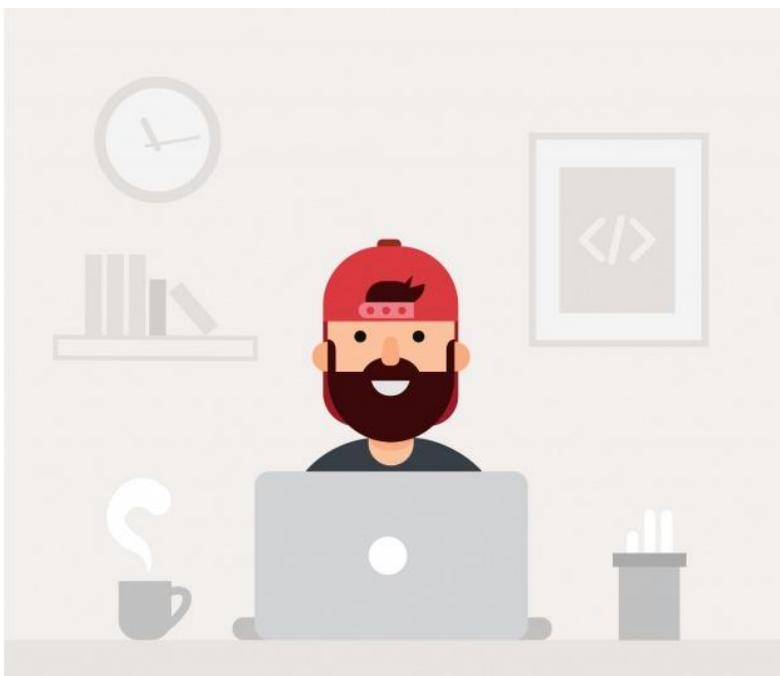
El número de valores que podemos probar es indefinido: no tiene por qué ser 3, sino que pueden ser menos casos o muchos más.



```
Segun variable_numerica Hacer
opcion_1:
    secuencia_de_acciones_1
opcion_2:
    secuencia_de_acciones_2
opcion_3:
    secuencia_de_acciones_3
De Otro Modo:
    secuencia_de_acciones_dom
Fin Segun
```

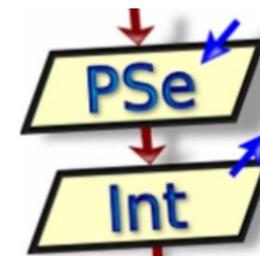
5 Desarrollo de los Contenidos

Ejercicios de Aplicación



Proyecto – Estructuras Condicionales

Drive Pensamiento Computacional y Programación en Python
<https://bit.ly/2Umqej7>



5 Desarrollo de los Contenidos

Sábado 31 de Julio

- ✓ Estructuras Repetitivas
- ✓ Ciclo Para
- ✓ Ciclo Mientras
- ✓ Ciclo Repetir
- ✓ Validación de entrada de datos Numéricos
- ✓ Ejercicios de Aplicación

Drive Workshops de Algoritmos y Programación en Python

<https://bit.ly/2Umqej7>

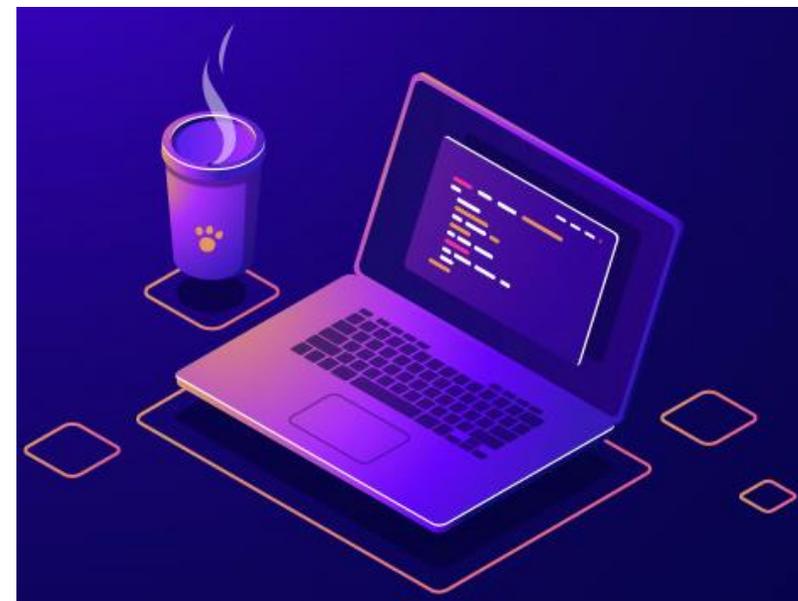


5 Desarrollo de los Contenidos

Estructuras Repetitivas

Se llaman problemas repetitivos o cíclicos a aquellos en cuya solución es necesario utilizar un mismo conjunto de acciones que se puedan ejecutar una cantidad específica de veces. Esta cantidad puede ser fija (previamente determinada por el programador) o puede ser variable (estar en función de algún dato dentro del programa). Los ciclos se clasifican en:

- ✓ Para
- ✓ Mientras
- ✓ Repetir



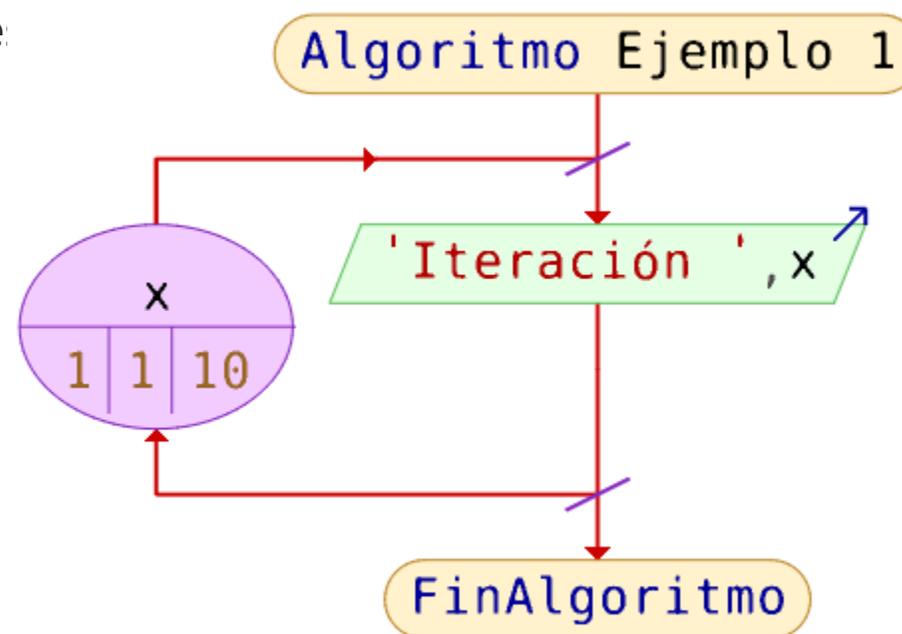
5 Desarrollo de los Contenidos

Estructuras Repetitivas – Ciclo Para

Son aquellos en que el número de iteraciones se conoce antes de ejecutarse el ciclo.

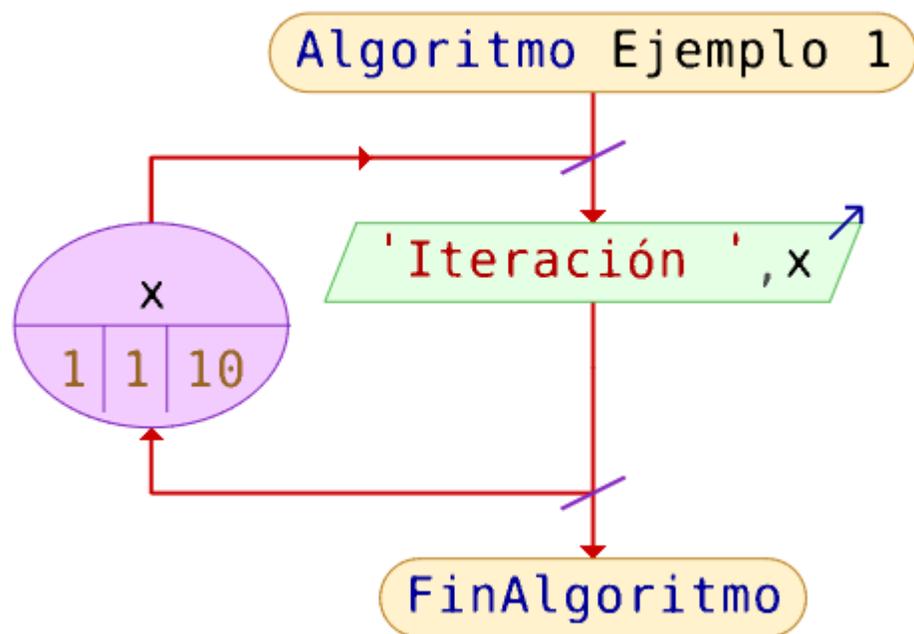
La forma de esta estructura es la siguiente:

1. Inicio de Variables ($x < -1$)
2. Condición ($x \leq 10$)
3. Incrementos o decrementos ($x < -x + 1$)



5 Desarrollo de los Contenidos

Estructuras Repetitivas – Ciclo Para



```
1 Algoritmo Ejemplo_1
2
3 Para x<-1 Hasta 10 Con Paso 1 Hacer
4     Escribir "Iteración ", x
5
6 Fin Para
7
8 FinAlgoritmo
9
```

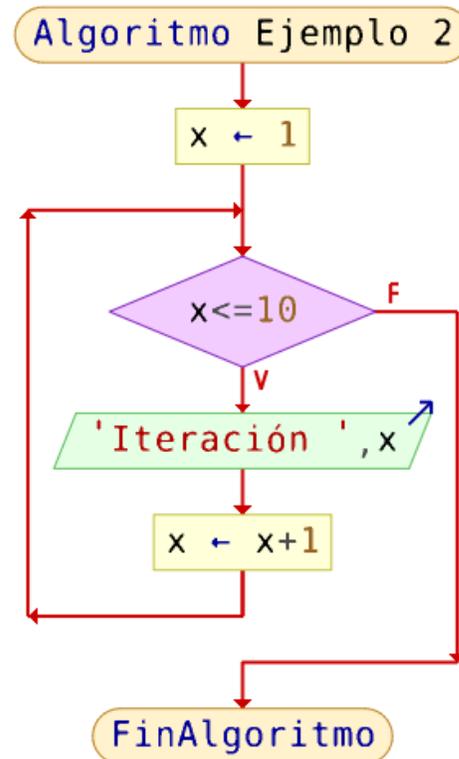
5 Desarrollo de los Contenidos

Estructuras Repetitivas – Ciclo Mientras

Esta es una estructura que repetirá un proceso durante “N” veces, donde “N” puede ser fijo o variable. Para esto, la instrucción se vale de una condición que es la que debe cumplirse para que se siga ejecutando. Cuando la condición ya no se cumple, entonces ya no se ejecuta el proceso.

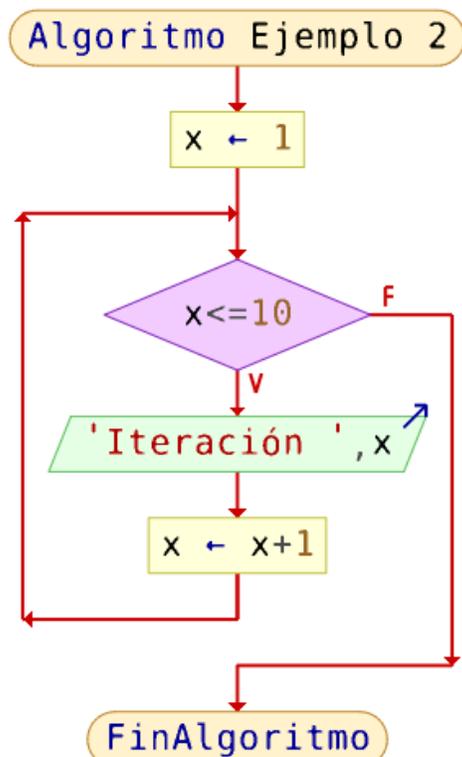
La forma de esta estructura es la siguiente:

1. Inicio de Variables ($x < -1$)
2. Condición ($x \leq 10$)
3. Incrementos o decrementos ($x \leftarrow x + 1$)



5 Desarrollo de los Contenidos

Estructuras Repetitivas – Ciclo Mientras



Algoritmo Ejemplo_2

```
x ← 1
Mientras x ≤ 10 Hacer
    Escribir "Iteración ", x
    x ← x + 1
Fin Mientras
FinAlgoritmo
```

5 Desarrollo de los Contenidos

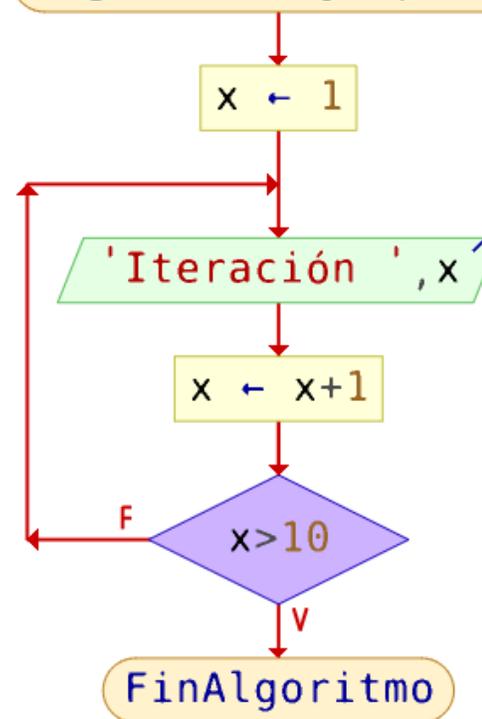
Estructuras Repetitivas – Ciclo Repetir

Esta es una estructura similar en algunas características, a la anterior. Repite un proceso una cantidad de veces, pero a diferencia del Mientras, el Repetir lo hace hasta que la condición se cumple y no mientras, como en el Mientras. Por otra parte, esta estructura permite realizar el proceso al menos una vez, ya que la condición se evalúa al final del proceso, mientras que en el Mientras puede ser que nunca llegue a entrar si la condición no se cumple desde un principio.

La forma de esta estructura es la siguiente:

1. Inicio de Variables ($x < -1$)
2. Incrementos o decrementos ($x \leftarrow x + 1$)
3. Condición ($x > 10$ o $x = 11$)

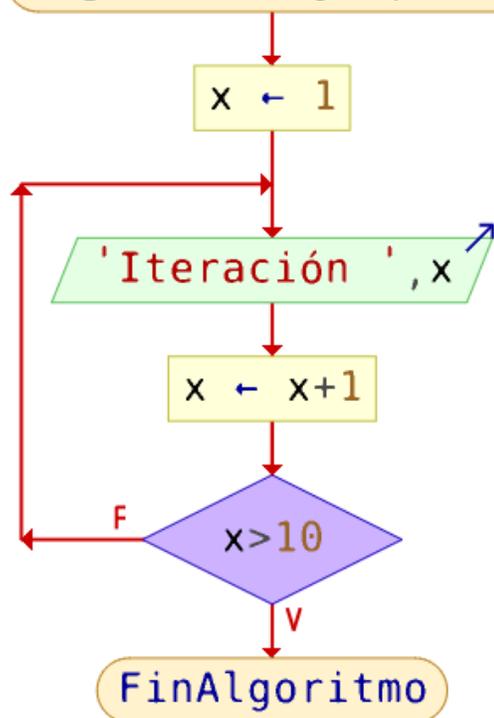
Algoritmo Ejemplo 3



5 Desarrollo de los Contenidos

Estructuras Repetitivas – Ciclo Repetir

Algoritmo Ejemplo 3



Algoritmo Ejemplo_3

```
x ← -1
Repetir
    Escribir "Iteración ", x
    x ← -x + 1
Hasta Que x > 10
FinAlgoritmo
```

5 Desarrollo de los Contenidos

Estructuras Repetitivas – Validación de entrada de datos Numéricos

La validación de datos es un proceso que asegura la entrega de datos limpios y claros a los programas, aplicaciones y servicios que lo utilizan. Comprueba la integridad y validez de los datos que se están introduciendo en diferentes software y sus componentes. La validación de los datos garantiza que los datos cumplen con los requisitos y los parámetros de calidad.

Algunos de los tipos de validación de datos incluyen:

- ✓ Validación de código
- ✓ Validación de tipo de datos
- ✓ Validación del rango de datos
- ✓ Validación de restricciones
- ✓ Validación estructurada



5 Desarrollo de los Contenidos

Estructuras Repetitivas – Validación de entrada de datos Numéricos

Validación de datos Numéricos en un Rango Especifico:

```
Para x<-1 Hasta 10 Con Paso 1 Hacer
    Escribir "Ingrese una nota en el rango de 0 a 5"
    Leer nota
    mientras (nota < 0 o nota > 5) hacer
        Escribir "Error... Notas fuera de Rango"
        Escribir "Ingrese una nota en el rango de 0 a 5"
        Leer nota
    FinMientras
Fin Para
```

5 Desarrollo de los Contenidos

Estructuras Repetitivas – Validación de entrada de datos Numéricos

Validación de datos
Numéricos como Control
del Ciclo:

```
control <-2
num <-1
Mientras (control <> 1) Hacer
    Escribir "Factura No ", num
    //Validación Salida
    Escribir "Desea continuar en el programa (1. No - 2. Si)"
    Leer control
    //Validar Control
    Mientras (control < 1 o control > 2) Hacer
        Escribir "Opción Fuera de Rango"
        Escribir "Desea continuar en el programa (1. No - 2. Si)"
        Leer control
    FinMientras
num <- num + 1
Limpiar Pantalla
FinMientras
```

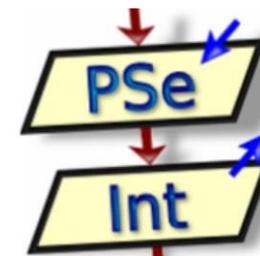
5 Desarrollo de los Contenidos

Ejercicios de Aplicación



Proyecto – Estructuras Repetitivas

Drive Pensamiento Computacional y Programación en Python
<https://bit.ly/2Umqej7>



5 Desarrollo de los Contenidos

Bigliografía

Libros:

Fadul, A. (2004). DISEÑO ESTRUCTURADO DE ALGORITMOS

Imágenes:

<https://www.freepik.es/>

<https://pixabay.com/>





¡GRACIAS POR SU ATENCIÓN!

www.unad.edu.co

 @universidadunad  /universidadunad  universidadunad  @UniversidadUNAD