**UNIVERSIDAD DE GUADALAJARA**

**CUCEI**

**DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN**

**DEPARTAMENTO DE CIENCIAS COMPUTACIONALES**

**PRÁCTIA No. 1**

*TEMA***: ESTRUCTURA DE CONTROL SELECTIVA SIMPLE Y LOS OPERADORES RELACIONALES**

*NOMBRE DEL ESTUDIANTE*: **ROBLES PULIDO EFRAIN**

*NOMBRE DE LA MATERIA:* **FUNDAMENTOS FILOSOFICOS DE LA COMPUTACION**

*SECCIÓN:* **D13**  *CALENDARIO:* **2021A**

*NOMBRE DEL PROFESOR:* **LUIS FELIPE MARISCAL LUGO**





**Marco Teórico**

* **¿Para que sirve la estructura de control selectiva simple?**

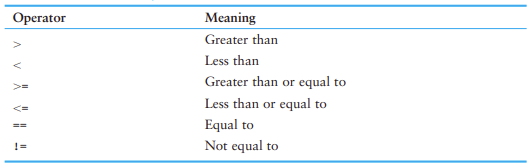
La instrucción **if** se utiliza para crear una estructura de decisión, lo que permite que un programa tenga más de una ruta de ejecución. La sentencia **if** causa una o más declaraciones para ejecutar solo cuando una expresión booleana es verdadera.

Una estructura de control es un diseño lógico que controla el orden en el que un conjunto de declaraciones ejecutar. Hasta ahora en este libro solo hemos utilizado el tipo más simple de estructura de control: la estructura de secuencia. Una estructura de secuencia es un conjunto de declaraciones que se ejecutan en el orden en que ellos aparecen.

La cláusula **if** comienza con la palabra si, seguida de una condición, que es una expresión que se evaluará como verdadero o falso. Aparecen dos puntos después de la afección. Comenzando en la siguiente línea hay un bloque de declaraciones. (En la siguiente línea se agrega una sangría que es necesaria porque el intérprete de Python la usa para indicar dónde el bloque comienza y termina.)

Cuando se ejecuta la instrucción **if**, se prueba la condición. Si la condición es verdadera, se ejecutan las sentencias que aparecen en el bloque que sigue a la cláusula **if**. Si la condición es falsa, se omiten las declaraciones del bloque.

* **Tabla de los operadores relacionales**



**Bibliografía:**

Comenzando con Python (3ª edición) PDF - Firebase Python Programming (2ª edición) Inventa sus propios juegos de computadora con Python: ... Hacking: Hacking Made Easy 1: Principiantes: - [Documento PDF]. (2021). Consultado el 13 de marzo de 2021 en <https://fdocuments.in/document/starting-out-with-python-3rd-edition-pdf-firebase-python-programming-2nd-edition.html>

**Resolver los problemas**

**1.-Definición del problema**

Solicitar al usuario dos números enteros, aplicar todos los operadores relacionales (>,<,>=,<=,==,!=) para comparar los datos de entrada y mostrar la salida correspondiente de acuerdo con el operador relacional. Observa la siguiente imagen para darte una idea más amplia del problema a resolver.

**2.- Análisis del problema**

**Entrada:** Solicitar al usuario dos datos de tipo entero, vamos a utilizar dos variables, en cada una de ellas almacenar un dato.

**Proceso:** Después de solicitar los datos, utilizaremos 6 condiciones (if) en donde si es verdadero una operación relacional (>,<,>=,<=,==,!=) se imprimirá un mensaje.

(d1 < d2), (d1 >= d2), (d1 <= d2), (d1 == d2), (d1 <> d2)

**Salida:** Mostrar el mensaje correspondiente de las operaciones relacionales, es decir, imprimir el mensaje correspondiente de cada operación relacional en donde sea verdadero el valor.

**3.-Diseño del algoritmo**

**Diagrama de flujo**

Plataforma: Lucidchart

**Diagrama

Descripción generada automáticamente**

**Pseudocódigo**

# Robles Pulido Efrain

#Algoritmo donde se utiliza los operadores relacionales y la selectiva simple

#Entrada

Entrada (“Introduce un dato:”)

Leer (d1)

Entrada (“Introduce otro dato:”)

Leer (d2)

#Estructura de control simple

Si d1 > d2:

Imprimir (d1,“ es mayor a “,d2) #Salida

Si d1 < d2:

Imprimir (d1,“ es menor a “,d2) #Salida

Si d1 >= d2:

Imprimir (d1,“ es mayor o igual a “,d2) #Salida

Si d1 <= d2:

Imprimir (d1,“ es menor o igual a “,d2) #Salida

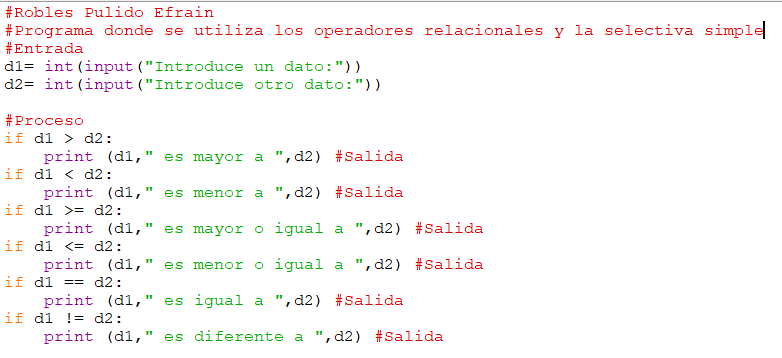
Si d1 == d2:

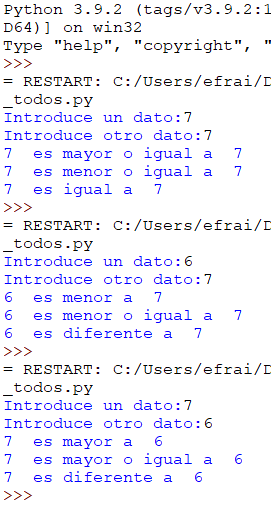
Imprimir (d1,“ es igual a “,d2) #Salida

Si d1 <> d2:

Imprimir (d1,“ es diferente a “,d2) #Salida

**4.-Codificación**





**Conclusión:**