

UNIVERSIDAD TECNOLÓGICA NACIONAL
Facultad Regional Reconquista

Programación en Computación
Ciclo Lectivo: 2021

Trabajo Práctico N.º 2

Tema Unidad 2: **Algoritmos**

GRUPO N°: 12

INTEGRANTES: Fulano, Mengano, Zutano

Guía Unidad 2: ALGORITMOS

SITUACIONES PROBLEMÁTICAS

1. Diseñar un algoritmo coloquial que describa el proceso de lavarse los dientes.
2. Diseñar un algoritmo coloquial que describa el proceso de abrir una puerta.
3. Diseñar un algoritmo coloquial que describa el proceso de atarse los cordones de los zapatos.
4. Diseñar un algoritmo coloquial que describa el proceso de hacer una torta de nueces.
5. Diseñar un algoritmo coloquial que describa el proceso de cambiar el neumático pinchado de un automóvil.
6. Diseñar un algoritmo que calcule y muestre el número de segundos que hay en X días.
7. Diseñar un algoritmo que, dados la base y altura de un rectángulo, calcule y muestre la Superficie y el Perímetro.
8. Diseñar un algoritmo que, dados las bases y altura de un trapecio, calcule y muestre la Superficie.
9. Diseñar un algoritmo que, dados la base y altura de un triángulo rectángulo, calcule y muestre la Superficie.
10. Un alumno desea saber cuál será su calificación final en la materia de Algoritmos. Dicha calificación se compone de los siguientes porcentajes:
 - 55% del promedio de sus tres calificaciones parciales.
 - 30% de la calificación del examen final.
 - 15% de la calificación de un trabajo final.
11. Calcular el número de pulsaciones que una persona debe tener, si la fórmula es:
$$\text{num. pulsaciones} = (220 - \text{edad}) / 10$$
12. Ingresar una temperatura en °C (Centígrados o Celsius), mostrar sus valores equivalentes en °K (Kelvin) y en °F (Fahrenheit). *(Problema desarrollado como ejemplo)*

ACTIVIDADES:

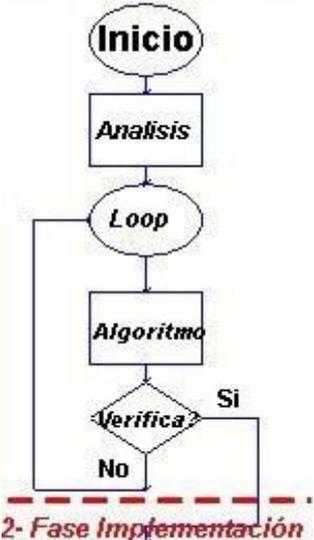
Desarrollar las distintas Fases para la Creación de un Algoritmo que resuelva la situación problemática anterior.

Conocimientos necesarios:

Fases en la Creación de Programas. [Programa = Algoritmo + DATOS]. Programas y Sub - programas. Características de un buen diseño. **Operatoria:**

1- Fase de Resolución

1 - Fase de Resolución de un Programa:



- a) El **Análisis del Problema**, implica Examinar cuidadosamente el problema para tener una idea clara de lo que se solicita y determinar los DATOS (de Entrada, Salida y Auxiliares) necesarios para conseguirlo.
- b) El **Diseño del Algoritmo**, [Secuencia ordenada de pasos (Sin Ambigüedades) que conducen a la solución de un Problema], que debe ser:

- i - Preciso** **ii - Definido** **iii – Finito.**

Esto implica identificar las tareas más importantes y ponerlas en el orden en que deben ejecutarse.

Estos pasos pueden repetirse (Refinamiento, Diseño Descendente o Top-Down) hasta obtener un **Algoritmo Claro, Preciso y Completo.**

Además, un Algoritmo consta de tres partes principales:

- i - Información de Entrada** del Algoritmo.
- ii - Proceso** u operaciones necesarias para la solución.
- iii - Salida** o respuestas dadas por el Algoritmo.

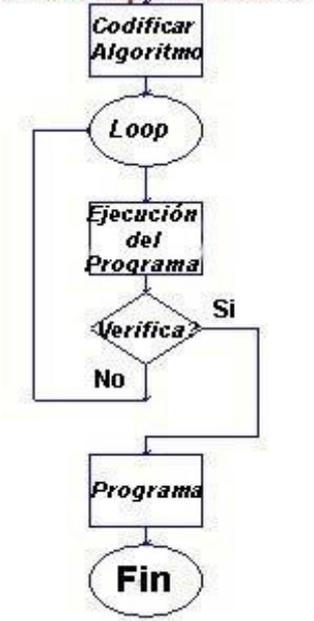


Esto implica contestar las siguientes preguntas:

- i - Para las especificaciones de la Entrada:** ¿Qué DATOS son de Entrada?, ¿Cuántos se necesitan?, ¿Cuándo son DATOS válidos?
- iii - Para las especificaciones de la Salida:** ¿Cuáles son los datos de Salida?, ¿Cuántos datos de salida se producirán?, ¿Qué precisión necesitan los resultados?, ¿Se debe imprimir una cabecera?
- ii - Para el Proceso** se aplica el algoritmo que nos lleva a la Salida.

Hay métodos Normalizados para representar Algoritmos como:

- 1) Diagramas de Flujo, 2) Pseudo-Código, 3) Nassi-Schneiderman



- c) La **Verificación del Algoritmo** implica comprobar que realiza las tareas esperadas y produce los resultados correctos, y esto se consigue:
 - Ejecutando el algoritmo manualmente y verificarlo con datos conocidos
 - Con datos significativos de un amplio rango de valores. (Nulos, negativos, extremos, etc.)
 - Llevando una tabla con anotación de las modificaciones. (es muy conveniente)

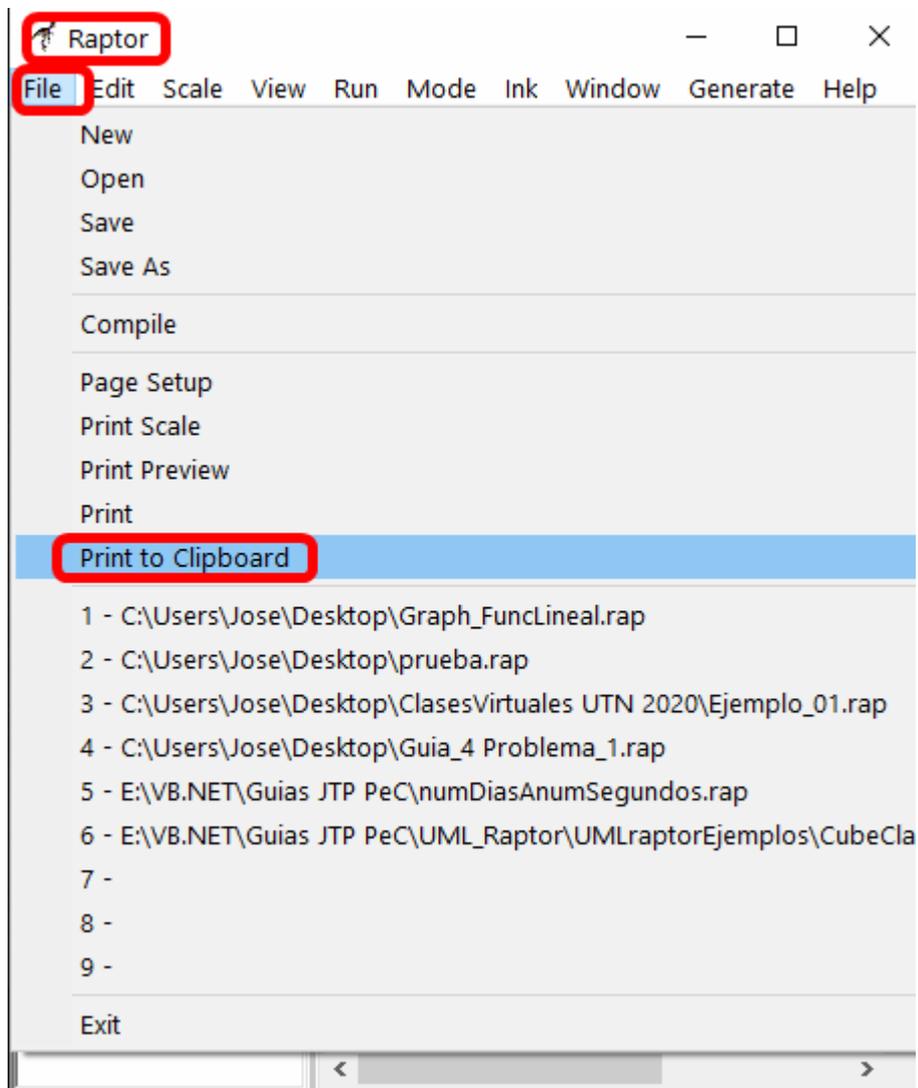
Esto se conoce como **prueba del Algoritmo.**

Si **NO verifica**, se debe retornar al punto **b) El Diseño del Algoritmo.**

2 - Fase de Implementación.

Si el Algoritmo **verifica**, se pasa a la Codificación o Traducción a un Lenguaje de Programación; que se completará con la Ejecución y Verificación del Programa en la Computadora.

Para representar las **Estructuras Secuenciales**, sobre todo en problemas de carácter numérico, ver: “**Diseño_de_Algoritmos.pdf**”, punto **5.1**. En esta Guía, a partir del problema **6**, además de realizar el *Pseudo-Código*, se deberá implementar el *Diagramas de Flujo* (en RAPTOR) y programarlos. Para llevar este diagrama al informe que deben presentar, pueden llevarlo al Word, directamente desde una opción del Menu -> File -> Print to Clipboard. (Ver Foto)



A continuación, realizaremos un problema numérico (digamos que es el problema 12 de esta guía.), cuya consigna es:

12 Ingresar una temperatura en °C (Centígrados o Celsius), mostrar sus valores equivalentes en °K (Kelvin) y en °F (Fahrenheit).

- Siguiendo las directivas dadas en el Diagrama de Flujo de la página 2 de este informe, en el punto **Fase de Resolución de un Programa**, procederemos a realizar:
- El **Análisis del Problema**, lo que implica Examinar cuidadosamente el problema para tener una idea clara de lo que se solicita y determinar los DATOS (de **Entrada, Salida y Auxiliares**).
- Aquí es evidente que el dato de **Entrada** será un valor numérico que representa una temperatura dada en °C, que identificaremos como **grado_Celsius**.
- Los datos de **Salida** son también números, que representarán el valor de la temperatura ingresada en °C pero transformados a sus valores en °K (**grado_Kelvin**) y en °F (**grado_Fahrenheit**).
- En este caso, no son necesarios datos **Auxiliares**.
- El **Diseño del Algoritmo**, [o **Secuencia ordenada de pasos (Sin Ambigüedades) que conducen a la solución de un Problema**], que debe ser: **i – Preciso ii - Definido iii – Finito**.
- El problema, necesita dos **Algoritmos**: uno para transformar los °C en °K; y otro para ir de °C a °F.

Si no recordamos como realizar estas transformaciones, debemos investigar el tema.

Hoy es simple, ir a **Internet -> Google** y preguntar: “**como convertir grados centígrados a kelvin**”:

Google

como convertir grados centígrados a kelvin

Todos Imágenes Videos Maps Noticias Más

Cerca de 47,700 resultados (0.52 segundos)

Temperatura

0 = 273.15

Grado Celsius Kelvin

Fórmula 0 °C + 273.15 = 273.15 K

Más información

Donde, ya está seleccionada la opción de convertir Temperatura, y específicamente la transformación de °C a °K, con indicación de la Fórmula de Cálculo:

$$^{\circ}\text{K} = \text{grado_Kelvin} = \text{grado_Celsius} + 273,15.$$

Para obtener la Fórmula de Cálculo: °F solo debemos seleccionar el **ListBox** que dice **Kelvin** y cambiarlo por (**grado_Fahrenheit**) para obtener la siguiente Fórmula:

$$^{\circ}\text{F} = \text{grado_Fahrenheit} = (\text{grado_Celsius} \times 9 / 5) + 32.$$

- Con estas dos fórmulas de transformación o (algoritmos), ya podemos continuar con:

ii - Proceso u operaciones necesarias para la solución.

Realizados los cálculos indicados por el proceso anterior, tendremos el valor de los **°K = grado_Kelvin** y **°F = grado Fahrenheit** que indica el problema: **mostrar sus valores equivalentes en °K (Kelvin) y en °F (Fahrenheit).**

iii – Salida.

Ahora codificamos todo el proceso en “Pseudo-Código”:

Inicio

Leer: grado_Celsius

Asignar: grado_Kelvin = grado_Celsius + 273,15

Asignar: grado_Fahrenheit = (grado_Celsius * 9 / 5) + 32

Imprimir: “°K = “ grado_Kelvin, “ °F = “ grado_Fahrenheit

Fin

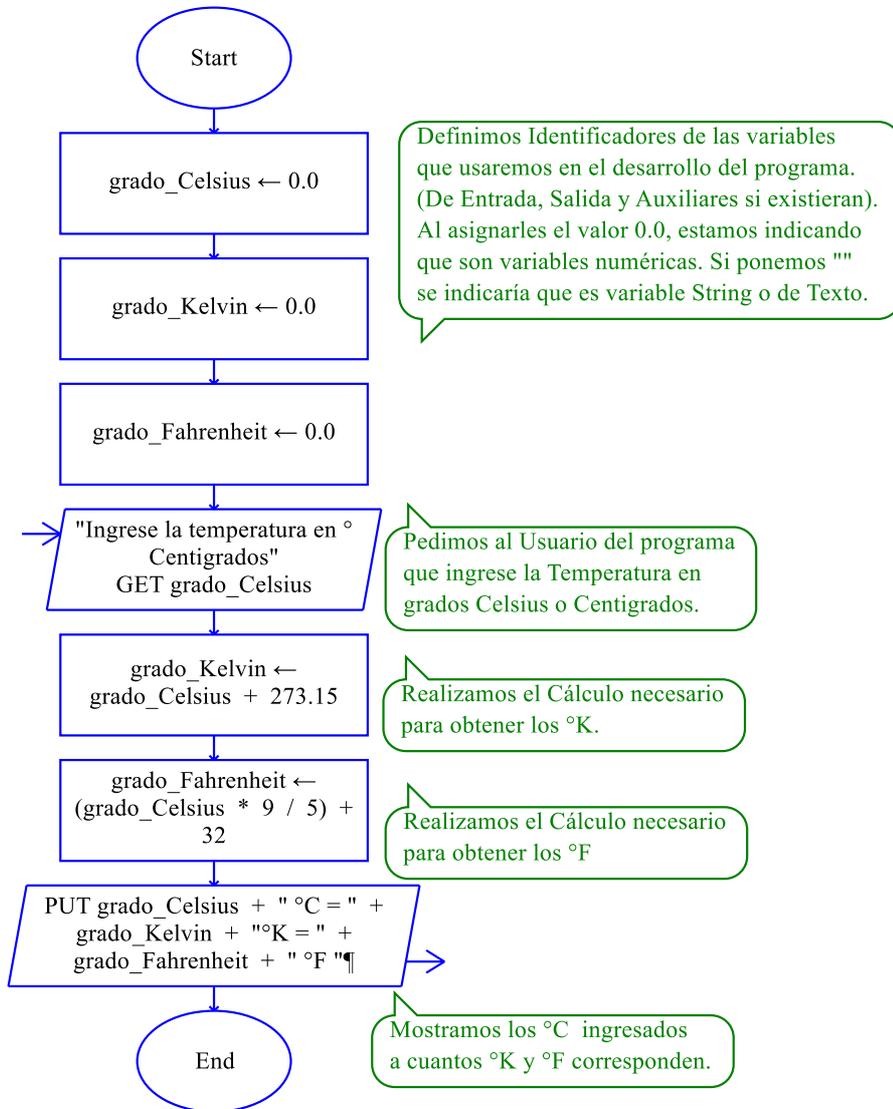
C)- A continuación, se debería verificar que las respuestas dadas son correctas:

°C	°K	°F
0	273,15	32
100	373,15	212
-273,15	0	-459,67
-17,7777....	255,3722...	0
-300	-26,85	-508

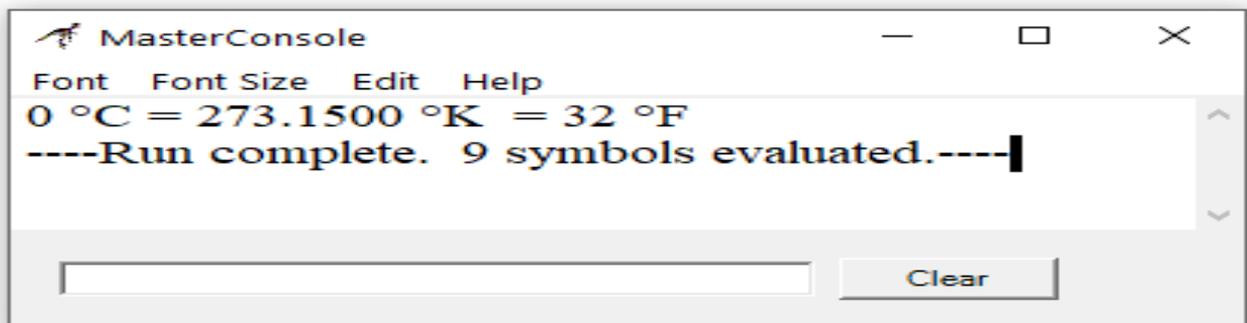
OJO: Resalto el valor **-26,85 °K**, porque sabemos que una temperatura por debajo de los 0 °K es imposible Físicamente, a pesar de que el cálculo es correcto, sería importante en una segunda versión informar al Usuario de esta imposibilidad.

y ya podemos ir a la realización del *Diagramas de Flujo*, que haremos (y programaremos) en RAPTOR.

Diagramas de Flujo realizado en RAPTOR, y en el interior de cada Símbolo tenemos la codificación ingresada, y en las nubes verdes, los comentarios sobre lo que hacemos dentro de cada símbolo.



La salida impresa que podemos ver por la consola de RAPTOR nos muestra lo solicitado en la consigna del problema:



Escritura de la Impresión o Salida:

The image shows a screenshot of an IDE's 'Enter Output' window on the left and a flowchart on the right. The IDE window displays examples of string concatenation and a red prompt 'Ingrese la salida aquí'. The flowchart starts with an input box 'Ingrese la temperatura en °Centigrados' leading to 'GET grado_Celsius'. This is followed by two calculation boxes: 'grado_Kelvin ← grado_Celsius + 273.15' and 'grado_Fahrenheit ← (grado_Celsius * 9 / 5) + 32'. A final output box 'PUT grado_Celsius + "°C=" + grado_Kelvin + "°K=" + grado_Fahrenheit + "°F"' leads to an 'End' terminal box. Green callouts explain each step: 'Pedimos al Usuario del programa que ingrese la Temperatura en grados Celsius o Centigrados.', 'Realizamos el Cálculo necesario para obtener los °K.', and 'Realizamos el Cálculo necesario para obtener los °F'.

Detalle de la Salida:

This is a close-up screenshot of the IDE's output window. It shows three lines of concatenated strings: 'grado_Celsius + "°C=" +', 'grado_Kelvin + "°K=" +', and 'grado_Fahrenheit + "°F"'. Below these is a large red prompt 'Ingrese la salida aquí'.

Aquí es importante comprender que el signo ‘+’ actúa como **Pegar o Concatenar**, y puede pegar texto (*siempre debe estar entre comillas*) y también variables Numéricas, (*de las cuales trae su valor de memoria*) y lo transforma en texto para imprimir.