UNIVERSIDAD NACIONAL ABIERTA

ÁREA DE INGENIERÍA

CARRERA INGENIERÍA DE SISTEMAS

TRABAJO PRÁCTICO

ASIGNATURA: **COMPUTACIÓN I**

CÓDIGO: **323**

FECHA DE PUBLICACIÓN EN EL BLOG DEL SUBPROGRAMA DISEÑO ACADÉMICO: **En las primeras cinco semanas de administración del lapso.**

FECHA DE DEVOLUCIÓN DEL INFORME POR EL ESTUDIANTE: El estudiante contará hasta el día **21/10/2023 sin prórroga** para su realización y envío.

NOMBRE DEL ESTUDIANTE: **CÉSAR ANTONIO TORRES CHANG**

CÉDULA DEL ESTUDIANTE: **V-20.246.713**

CENTRO LOCAL: **METROPOLITANO**

CARRERA: **236**

NÚMERO DE ORIGINALES:

LAPSO: **2023-2**

RESULTADOS DE CORRECCIÓN:

OBJ. N°	II.2	II.3	II.4
0: NL; 1: L			

TRABAJO PRÁCTICO
COMPUTACIÓN I (323)
LAPSO 2023-2

Especificaciones:

El Departamento de Personal de cierta compañía necesita implementar un programa en Pascal que calcule el salario neto semanal de un trabajador en función del número de horas trabajadas y la tasa de impuestos de acuerdo a las siguientes hipótesis:

- Las primeras 35 horas se pagan a tarifa normal
- Las horas que pasen de 35 se pagan 1.5 veces la tarifa normal
- Las tasas de impuestos son:

- a: Los primeros 50 dólares son libres de impuestos
- b: Los siguientes 40 dólares tienen un 25% de impuestos
- c: Los restantes tienen un 45% de impuestos

Orden de las actividades a realizar correspondientes a los objetivos II.2, II.3 y II.4, a fin de llevar una mejor ejecución de la solución.

Solución analítica:

Condiciones y Reglas:

- Las primeras 35 horas se pagan a una tarifa normal (TARIFA_NORMAL).
- Las horas que superen las 35 horas se pagan a 1.5 veces la tarifa normal.
- Las tasas de impuestos son las siguientes:
- Los primeros 50 dólares son libres de impuestos.
- Los siguientes 40 dólares tienen un 25% de impuestos.
- Cualquier ingreso que supere los 90 dólares tiene un 45% de impuestos.

Solución:

- Solicitar al usuario que ingrese el número de horas trabajadas.

- Validar que la entrada sea un número positivo. Si no lo es, mostrar un mensaje de error y volver al paso 1.

- Calcular el salario bruto:

    - Si las horas trabajadas son 35 o menos, el salario bruto es **horasTrabajadas * TARIFA_NORMAL**.
    - Si las horas trabajadas son más de 35, el salario bruto es **35 * TARIFA_NORMAL + (horasTrabajadas - 35) * (TARIFA_NORMAL * 1.5)**.

- Calcular el monto de impuestos a pagar:

    - Si el salario bruto es de 50 dólares o menos, los impuestos son 0.
    - Si el salario bruto es más de 50 dólares, pero 90 dólares o menos, los impuestos son **(salarioBruto - 50) * 0.25**.
    - Si el salario bruto es más de 90 dólares, los impuestos son **40 * 0.25 + (salarioBruto - 90) * 0.45**.

- Calcular el salario neto restando los impuestos al salario bruto:
    - **salarioNeto = salarioBruto - impuestos**.

- Mostrar el salario neto semanal calculado.

TARIFA_NORMAL: representa el valor de la remuneración por hora del trabajador, como dicha tarifa no está especificada en el enunciado, le daremos el valor de **10** al momento de implementar el código en Pascal.

Objetivo II.2

Resuelva el problema planteado algorítmicamente usando la metodología MAPS, a fin de facilitar la conceptualización, diseño, planificación, ejecución de la solución solicitada.

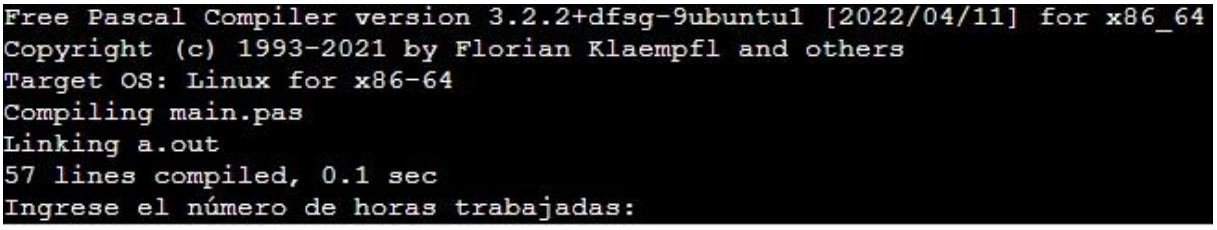
Aplicación de la metodología MAPS

- **Diálogo:**

La etapa de Diálogo es fundamental en la implementación de programas, ya que define la interacción entre el usuario y la aplicación. Este paso es crucial para la entrada de datos necesarios en el programa, como el número de horas trabajadas. La claridad en las instrucciones al usuario y la recopilación precisa de datos son esenciales para garantizar que el programa funcione correctamente.

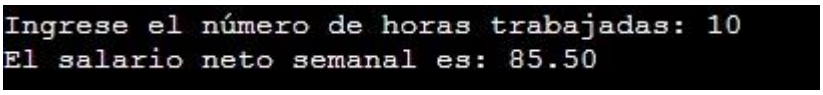
Entradas:

- Número de horas trabajadas.



Salidas:

- Salario neto semanal.



Las imágenes son tomadas de la compilación y ejecución del programa en el IDE de Pascal de la página web: https://www.onlinegdb.com/online_pascal_compiler

- **Especificaciones:**

La etapa de Especificaciones establece las reglas del negocio y los requisitos del programa. En este caso, se especifican claramente las reglas para calcular el salario neto, incluyendo las tarifas de impuestos y las horas de trabajo normales y extras. Definir estas reglas con precisión es esencial para garantizar que el programa

cumpla con los requisitos. Cualquier ambigüedad o falta de claridad en las especificaciones puede llevar a errores en la implementación.

- Las primeras 35 horas se pagan a una tarifa normal (TARIFA_NORMAL).
- Las horas que superen las 35 horas se pagan a 1.5 veces la tarifa normal.
- Las tasas de impuestos son las siguientes:
- Los primeros 50 dólares son libres de impuestos.
- Los siguientes 40 dólares tienen un 25% de impuestos.
- Cualquier ingreso que supere los 90 dólares tiene un 45% de impuestos.

TARIFA_NORMAL: representa el valor de la remuneración por hora del trabajador, como dicha tarifa no está especificada en el enunciado, le daremos el valor de **10** al momento de implementar el código en Pascal.

Ya lo habíamos mencionado anteriormente, los detalles de los requisitos del programa pueden verse en la solución analítica.

- **División:**

La División del problema en partes más pequeñas facilita su resolución, esto implica descomponer el cálculo del salario neto en pasos lógicos y manejables. Por ejemplo:

- Calcular las horas normales y las horas extras.
- Calcular el salario bruto.
- Calcular los impuestos.
- Calcular el salario neto.

Esto lo realizamos detalladamente en la solución analítica:

- Calcular el salario bruto:

    - Si las horas trabajadas son 35 o menos, el salario bruto es **horasTrabajadas * TARIFA_NORMAL**.
    - Si las horas trabajadas son más de 35, el salario bruto es **35 * TARIFA_NORMAL + (horasTrabajadas - 35) * (TARIFA_NORMAL * 1.5)**.

- Calcular el monto de impuestos a pagar:

    - Si el salario bruto es de 50 dólares o menos, los impuestos son 0.
    - Si el salario bruto es más de 50 dólares, pero 90 dólares o menos, los impuestos son **(salarioBruto - 50) * 0.25**.
    - Si el salario bruto es más de 90 dólares, los impuestos son **40 * 0.25 + (salarioBruto - 90) * 0.45**.

- Calcular el salario neto restando los impuestos al salario bruto:
    - **salarioNeto = salarioBruto - impuestos**.

Definición de abstracciones:

La definición de abstracciones implica la creación de funciones y subprocesos que realizan tareas específicas, se definen subprocesos para cálculos complejos, como el cálculo del salario bruto y de impuestos. Esto aumenta la modularidad del programa, lo que lo hace más legible y mantenible. La definición clara de parámetros y valores de retorno es esencial para la correcta implementación de estos subprocesos.

Ejemplo:

```
if horasTrabajadas <= 35 then
  salarioBruto := horasTrabajadas * TARIFA_NORMAL
else
  salarioBruto := 35 * TARIFA_NORMAL + (horasTrabajadas - 35) * (TARIFA_NORMAL * 1.5);

if salarioBruto <= 50 then
  impuestos := 0
else if salarioBruto <= 90 then
  impuestos := (salarioBruto - 50) * 0.25
else
  impuestos := 40 * 0.25 + (salarioBruto - 90) * 0.45;

salarioNeto := salarioBruto - impuestos;
```

Codificación:

La etapa de codificación involucra la escritura real del programa en lenguaje Pascal. Aquí se implementan los subprocesos y se utilizan las abstracciones definidas anteriormente. La precisión en la traducción de los algoritmos en Pascal es crucial, ya que errores en esta etapa pueden llevar a resultados incorrectos.

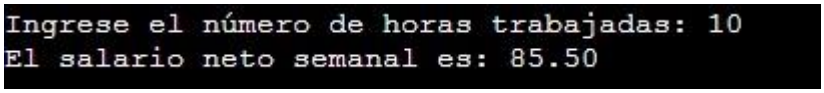
Prueba y verificación:

La prueba y verificación son pasos críticos para garantizar la funcionalidad del programa. Se deben realizar pruebas exhaustivas con diferentes conjuntos de datos de entrada para asegurarse de que el programa funcione correctamente. Las pruebas también deben incluir casos límite y situaciones inusuales. La verificación implica la revisión de código y resultados para detectar y corregir errores antes de la implementación en un entorno de producción. En este paso aplicamos el concepto de robustez.

Lo realizamos en la página: https://www.onlinegdb.com/online_pascal_compiler

Presentación:

Finalmente, la presentación de los resultados al usuario debe ser clara y comprensible. En Pascal, esto implica mostrar el salario neto calculado de una manera legible, como una cadena de texto que incluye el resultado. La presentación adecuada es importante para que el usuario entienda y pueda utilizar los resultados del programa de manera efectiva.



La codificación en algoritmos y en lenguaje Pascal está presentada en las páginas siguientes.

Objetivo II.3

Después de haber aplicado cada una de las etapas de la metodología MAPS en el objetivo II.2, diseñe un algoritmo usando técnicas de programación estructurada, que cumplan con las especificaciones dadas; tenga especial cuidado de hacer una buena declaración de datos y sus tipos.

```
Algoritmo CalculoSalario
    Definir horasTrabajadas, salarioBruto, impuestos, salarioNeto Como Real
    Definir TARIFA_NORMAL Como Real

    Escribir("Ingrese el número de horas trabajadas: ")
    Leer(horasTrabajadas)

    Si horasTrabajadas <= 35 Entonces
        salarioBruto = horasTrabajadas * TARIFA_NORMAL
    Sino
        salarioBruto = 35 * TARIFA_NORMAL + (horasTrabajadas - 35) *
        (TARIFA_NORMAL * 1.5)
    FinSi

    Si salarioBruto <= 50 Entonces
        impuestos = 0
    Sino
        Si salarioBruto <= 90 Entonces
            impuestos = (salarioBruto - 50) * 0.25
        Sino
            impuestos = 40 * 0.25 + (salarioBruto - 90) * 0.45
        FinSi
    FinSi

    salarioNeto = salarioBruto - impuestos

    Escribir("El salario neto semanal es: ", salarioNeto)
FinAlgoritmo
```

Objetivo II.4

Pruebe el algoritmo diseñado en el Objetivo II.3 usando el concepto de Robustez; se requiere que Ud. presente en el informe del trabajo, el algoritmo original, y el ajustado después de aplicarle la definición de la propiedad algorítmica mencionada.

Este algoritmo incorpora el manejo de errores para garantizar que el usuario ingrese datos válidos. Detecta si el usuario ingresa letras, números negativos o caracteres no válidos y muestra mensajes de error.

```
Proceso CalculoSalario
    Definir horasTrabajadas, salarioBruto, impuestos, salarioNeto Como Real
    Definir entradaUsuario Como Caracter
    Definir esNumero Como Logico
    Definir i Como Entero

    Repetir
        esNumero := Verdadero
        Escribir("Ingrese el número de horas trabajadas: ")
        Leer(entradaUsuario)

        i := 1
        Mientras i <= Longitud(entradaUsuario) Hacer
            Si No EsNumero(entradaUsuario[i]) y entradaUsuario[i] <> "." Entonces
                esNumero := Falso
                Romper
            FinSi
            i := i + 1
        FinMientras

        Si esNumero Entonces
            horasTrabajadas := ConvertirACaracterReal(entradaUsuario)
            Si horasTrabajadas < 0 Entonces
                esNumero := Falso
```

```
            Escribir("Error: Ingrese un número válido de horas (no negativo).")
        FinSi
    Sino
        Escribir("Error: Ingrese un número válido de horas.")
    FinSi

  Hasta Que esNumero

  Si horasTrabajadas <= 35 Entonces
    salarioBruto := horasTrabajadas * 10 // Reemplaza con la tarifa normal adecuada
  Sino
    salarioBruto := 35 * 10 + (horasTrabajadas - 35) * (10 * 1.5) // Reemplaza con la
tarifa normal adecuada
  FinSi

  Si salarioBruto <= 50 Entonces
    impuestos := 0
  Sino
    Si salarioBruto <= 90 Entonces
      impuestos := (salarioBruto - 50) * 0.25
    Sino
      impuestos := 40 * 0.25 + (salarioBruto - 90) * 0.45
    FinSi
  FinSi

  salarioNeto := salarioBruto - impuestos

  Escribir("El salario neto semanal es: ", Redondear(salarioNeto, 2))
FinProceso
```

Objetivo II.3

Codifique el algoritmo, obtenido en el objetivo II.4, en lenguaje PASCAL, a fin de obtener un programa estructurado que cumpla con los requerimientos especificados, aplicando tipo de datos y/o procedimientos y funciones y/o métodos de archivos, según se requiera.

```
program CalculoSalario;

var
  horasTrabajadas, salarioBruto, impuestos, salarioNeto: real;
  entradaUsuario: string;
  esNumero: boolean;
  i: integer;

const
  TARIFA_NORMAL = 10; // Reemplaza con la tarifa normal adecuada

begin
  repeat
    esNumero := true;
    Write('Ingrese el número de horas trabajadas: ');
    ReadLn(entradaUsuario);

    for i := 1 to Length(entradaUsuario) do
    begin
      if not (entradaUsuario[i] in ['0'..'9', '.']) then
      begin
        esNumero := false;
        Break;
      end;
    end;

    if esNumero then
    begin
```

```
      Val(entradaUsuario, horasTrabajadas);
      if (horasTrabajadas <= 0) then
      begin
        esNumero := false;
        Writeln('Error: Ingrese un número válido de horas (no negativo).');
      end;
    end
    else
    begin
      Writeln('Error: Ingrese un número válido de horas.');
    end;
  until esNumero;

  if horasTrabajadas <= 35 then
    salarioBruto := horasTrabajadas * TARIFA_NORMAL
  else
    salarioBruto := 35 * TARIFA_NORMAL + (horasTrabajadas - 35) *
(TARIFA_NORMAL * 1.5);

  if salarioBruto <= 50 then
    impuestos := 0
  else if salarioBruto <= 90 then
    impuestos := (salarioBruto - 50) * 0.25
  else
    impuestos := 40 * 0.25 + (salarioBruto - 90) * 0.45;

  salarioNeto := salarioBruto - impuestos;

  Writeln('El salario neto semanal es: ', salarioNeto:0:2);
end.
```