

	<p align="center"><b>Carátula para entrega de prácticas</b></p>	
<p align="center">Facultad de Ingeniería</p>	<p align="center">Laboratorio de docencia</p>	

# Laboratorios de computación salas A y B

<p align="center"><i>Profesor:</i></p>	<p align="center">Alejandro Pimentel</p>
<p align="center"><i>Asignatura:</i></p>	<p align="center">Fundamentos de programación</p>
<p align="center"><i>Grupo:</i></p>	<p align="center">3</p>
<p align="center"><i>No de Práctica(s):</i></p>	<p align="center">5</p>

<i>Integrante(s):</i>	Villanueva Bustamante Victoria
<i>No. de Equipo de cómputo empleado:</i>	19
<i>No. de Lista o Brigada:</i>	54 8034
<i>Semestre:</i>	2020-1
<i>Fecha de entrega:</i>	16 de septiembre 2019
<i>Observaciones:</i>	Revisa nuevamente cómo se usa el SWITCH, me parece que estás confundido y que intentas usar condiciones generales en los casos.
	Cuida más el formato de la carátula, apégate más al original

CALIFICACIÓN:

Introducción:

El origen etimológico de las dos palabras que le dan forma:

- Pseudo deriva del griego, de “seudo”, que puede traducirse como “falso”.
- Código, por su parte, emana del latín. En concreto, de “códices, codex” que se empleaban para referirse a los documentos o libros donde los romanos tenían escritas todas y cada una de sus leyes.

Se trata de un falso lenguaje, ya que apela a las normas de estructura de un lenguaje de programación aunque se encuentra desarrollado para que pueda ser leído por un ser humano y no interpretado por una máquina.

Objetivo:

Elaborar pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

Actividades:

### Actividad

- Desarrollar pseudocódigo que reciba un número obtenga su factorial

INICIO

```
int i, fact = 1, n
printf “calcular factorial del número ”
for i=1 i <=n i++
fact= fact *i
print “el factorial del numero ingresado es
```

FIN”

**0**

“calcular factorial del número 0”

El factorial del numero ingresado es 1 “

**2**

“calcular factorial del número 2”

El factorial del numero ingresado es 2“

**-4**

“calcular factorial del número -4”

El factorial del numero ingresado es no se puede calcular“

**5**

“calcular factorial del número 5”

El factorial del numero ingresado es 120 “

- Desarrollar un pseudocódigo que lea 2 datos, nivel e ingreso. El programa debe:

Verificar que no se tiene un nivel mayor al ingreso (el ingreso debe ser mayor que la base)

Mostrar el impuesto a pagar

El porcentaje del impuesto se aplica a la diferencia entre el ingreso y la base. Y el impuesto total se calcula con la suma del resultado anterior más la cuota fija.

BEGIN

nivel: INT

ingreso: REAL POSITIVO

READ nivel, ingreso

PI: REAL POSITIVO

I: REAL POSITIVO

base: REAL POSITIVO

CF: REAL POSITIVO

IF nivel <= 11

PRINT "Digite un nivel"

PRINT "Digite un ingreso"

SWITCH nivel:

case 1:

base > 0

base < 6,942.21

CF= 0

ingreso > base

PI= ((Ingreso - base)/1000)\*1.92

I=PI + CF

PRINT "El impuesto es "

case 2:

base > 6,942.21

base < 58,922.16

CF= 133.28

ingreso > base

PI= (Ingreso - base)/100)\*6.4

I=PI + CF

PRINT "El impuesto es "

case 3:

base > 58,922.16

base < 103,550.45

CF= 3,460.00

ingreso > base

PI= (Ingreso - base)/1000)\*10.88

I=PI + CF

PRINT "El impuesto es "

case 4:

base > 103,550.45

Tiene mas sentido aca

Operaciones, comparaciones, y operadores lógicos al aire, no tienen ningún efecto.

Entre 100

```
base < 120,372.84
CF= 8,315.57
ingreso > base
PI= (Ingreso - base)/1000)*16.00
I=PI + CF
PRINT "El impuesto es "
```

case 5:

```
base > 120,372.84
base < 144,119.24
CF= 11,007.14
ingreso > base
PI= (Ingreso - base)/1000)*17.92
I=PI + CF
PRINT "El impuesto es "
```

case 6:

```
base > 144,119.24
base < 290,667.76
CF= 15,262.49
ingreso > base
PI= (Ingreso - base)/1000)*21.36
I=PI + CF
PRINT "El impuesto es "
```

case 7:

```
base > 290,667.76
base < 458,132.30
CF= 46,565.26
ingreso > base
PI= (Ingreso - base)/1000)*23.52
I=PI + CF
PRINT "El impuesto es "
```

case 8:

```
base > 458,132.30
base < 874,650.01
CF= 85,952.92
ingreso > base
PI= (Ingreso - base)/1000)*30.00
I=PI + CF
PRINT "El impuesto es "
```

```

case 9:
    base > 874,650.01
    base < 1,166,200.01
    CF= 210,908.23
    ingreso > base
    PI= (Ingreso - base)/1000)*32.00
    I=PI + CF
    PRINT "El impuesto es "
case 10:
    base > 1,166,200.01
    base < 3,498,600.01
    CF= 304,204.21
    ingreso > base
    PI= (Ingreso - base)/1000)*34.00
    I=PI + CF
    PRINT "El impuesto es "
case 11:
    base > 3,498,600.01
    CF= 1,097,220.21
    ingreso > base
    PI= (Ingreso - base)/1000)*35.00
    I=PI + CF
    PRINT "El impuesto es "
END SWITCH
ELSE
ENDIF
END

```

Actividad 2: Verificar el algoritmo con los pares

```

- (1, 5000)
BEGIN
nivel: 1
ingreso: 5000
READ nivel
PI: REAL POSITIVO
I: REAL POSITIVO
base: REAL POSITIVO
CF:REAL POSITIVO
IF nivel <=11
    PRINT 1

```

```
READ ingreso
PRINT 5000
SWITCH nivel:
  case 1:
    base > 0
    CF= 0
    5000 > 0
    PI= (5000 - 0)/1000)*1.92
    I=9.6 + 0
    PRINT "El impuesto es 9.6"
  ELSE
ENDIF
END
```

- (7,8000): no es posible calcularlo
- (12,5000,000): no es posible calcularlo

Conclusión: Si bien los pseudocódigos nos sirven para dar órdenes a la computadora, no quiere decir que sea lo más eficiente en cuanto a tiempo hablamos, ya que se debe tener un mayor cuidado para escribirlo.