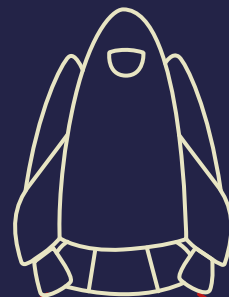


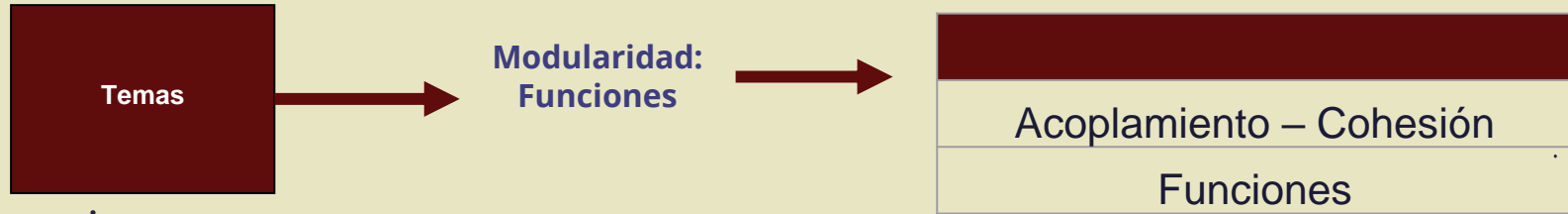
Programa académico CAMPUS



Ciclo 1:
Fundamentos de
Programación



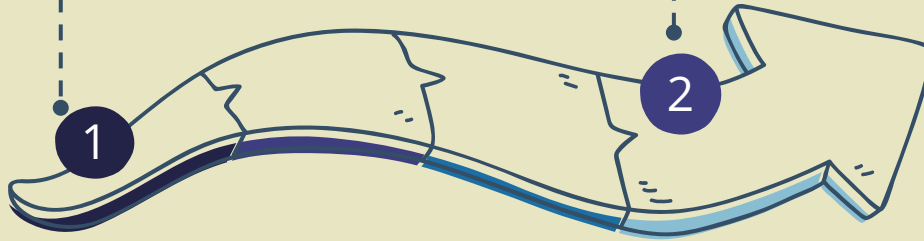
Presentación Ciclo 1 – Fundamentos de Programación



Funciones

**Modularidad:
Funciones**

Taller



Funciones

Conceptualización



¿Has visto alguna vez una carrera de autos de fórmula uno? Pues bien, hay un momento en la competición en la que los autos deben entrar a pits. La razón es que al auto se le debe hacer un mantenimiento a las llantas y se le debe suministrar combustibles. Ambas funciones deben llevarse a cabo luego de un determinado número de vueltas, cuando el ingeniero automovilístico encargado lo determine. Supongamos que la función general de la entrada a pits es realizar ambas tareas (cambio de llantas y suministrar combustible) ejecutadas una seguida de la otra. Así pues, una función se puede definir como una secuencia de instrucciones que tiene como finalidad llevar a cabo una tarea específica; como por ejemplo, realizar la suma de dos números, contar las palabras de una cadena de caracteres, etc.

Funciones

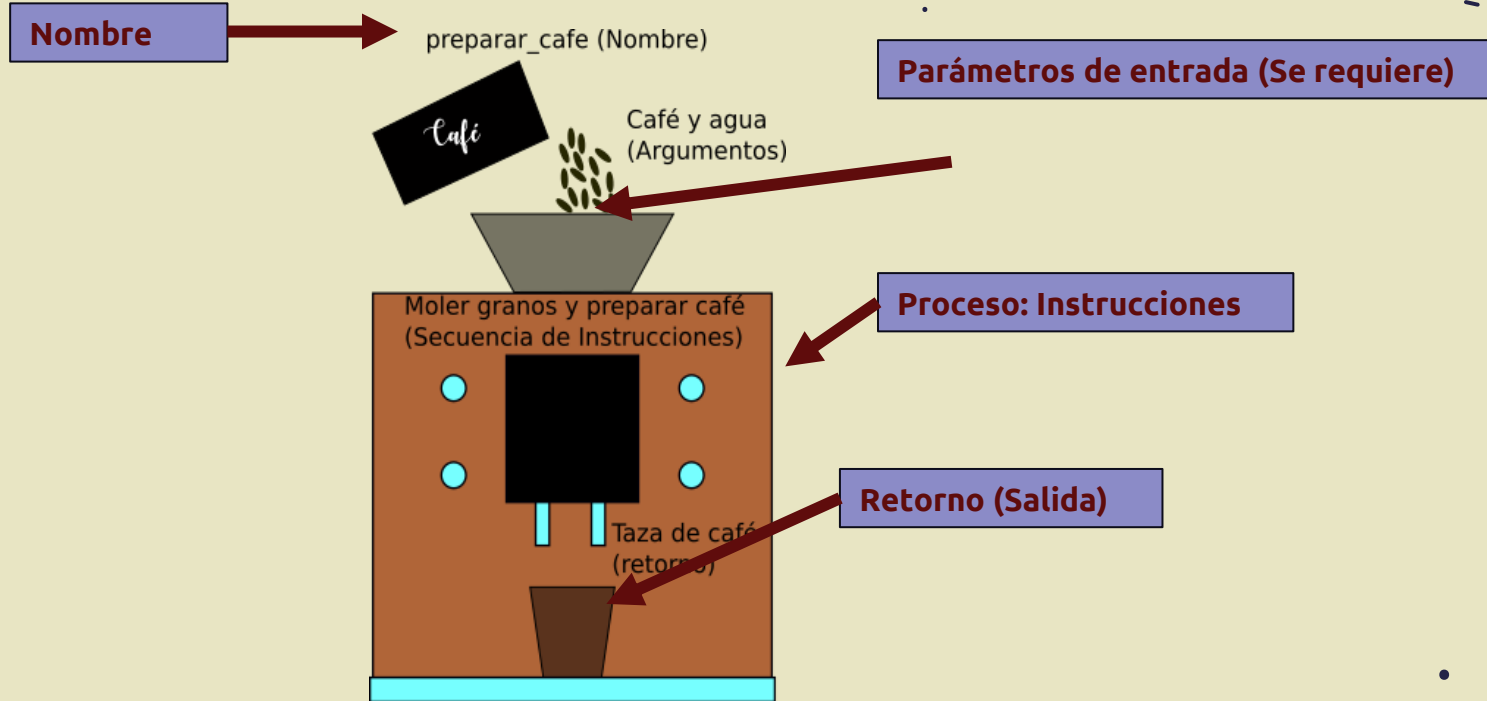
Conceptualización



- ✓ Las **funciones en programación reciben un nombre que debe ser coherente con su función.**
- ✓ Por lo general, un programa es dividido por diferentes tipos de funciones, que llevan a cabo diferentes tipos de tareas,
- ✓ Se logra la solución de un problema más grande.

Funciones

Estructura de una Función



Estructura básica de una función haciendo analogía a la preparación de una taza de café.

Funciones

Modularidad



- Uno de los aspectos fundamentales de la programación moderna, base de los nuevos paradigmas, es sin duda alguna la **modularidad**, entendida
- como la generación **de módulos** o **segmentos funcionales** e **independientes** que permitan una mejor organización y comprensión de un programa. Este aspecto se basa en la aplicación de dos técnicas propias de la ingeniería del software, denominadas **Acoplamiento de módulos** y **Cohesión de módulos** que definen unas guías en la definición de un módulo.

Funciones

Acoplamiento – Cohesión de Módulos



Cohesión de módulos

La técnica de la ingeniería del software, denominada Cohesión de Módulos busca medir el grado de **relación o dependencia** que existe entre las **actividades propias** de un **proceso o módulo**. La finalidad es generar módulos que realicen un proceso determinado y por consiguiente las actividades o instrucciones que contenga están todas relacionadas con el objetivo del módulo. Por ejemplo, un módulo de liquidación de comisiones, solo debe contener las instrucciones que permitan el cálculo del valor de la comisión y no incluir otro tipo de instrucciones, como las de incrementar contadores y sumadores.

Acoplamiento de módulos

La técnica del Acoplamiento de Módulos que se aplica después de la cohesión, tiene como objetivo la generación de **módulos independientes** dentro de un proceso, en los cuales, **cada uno de ellos define sus propias variables y la comunicación con ellos se realice a través de parámetros**, o sea, variables (argumentos) que recibe el módulo que le permitan realizar la función específica para lo que fue definido. Los módulos independientes, que reciben parámetros de entrada y retornan una salida específica, permiten su reutilización en otros programas y procesos, lo que facilita el desarrollo de software.

Funciones

Ejercicio



- La empresa de teléfonos de la ciudad necesita realizar su proceso de facturación en forma automática, contando con los N abonados, de los cuales conoce el nombre, estrato, que puede ser (1, 2, 3, 4, 5), cantidad de impulsos del mes (N es suministrado). Además la empresa nos informa que para la liquidación de la factura se debe tener en cuenta el valor de la tarifa básica, de acuerdo al estrato, que depende de la siguiente tabla:

Estrato	Ttarifa Básica
1	\$10.000
2	\$15.000
3	\$20.000
4	\$25.000
5	\$30.000

Además se debe calcular el valor de los impulsos, con base en la cantidad de impulsos del mes, conociendo que cada impulso tiene un valor de \$100. Con esta información, se desea:

- ☒ Valor a pagar de cada abonado con el nombre. **También se debe visualizar la tarifa básica y el valor de los impulsos**
- ☒ Valor total a pagar(Todos los abonados)

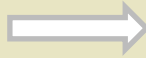
Funciones

Ejercicio



Metodología -> Pensamiento lógico estructurado

Análisis



Construcción

Método
Entrada – Proceso - Salida

Programa

Funciones

Ejercicio

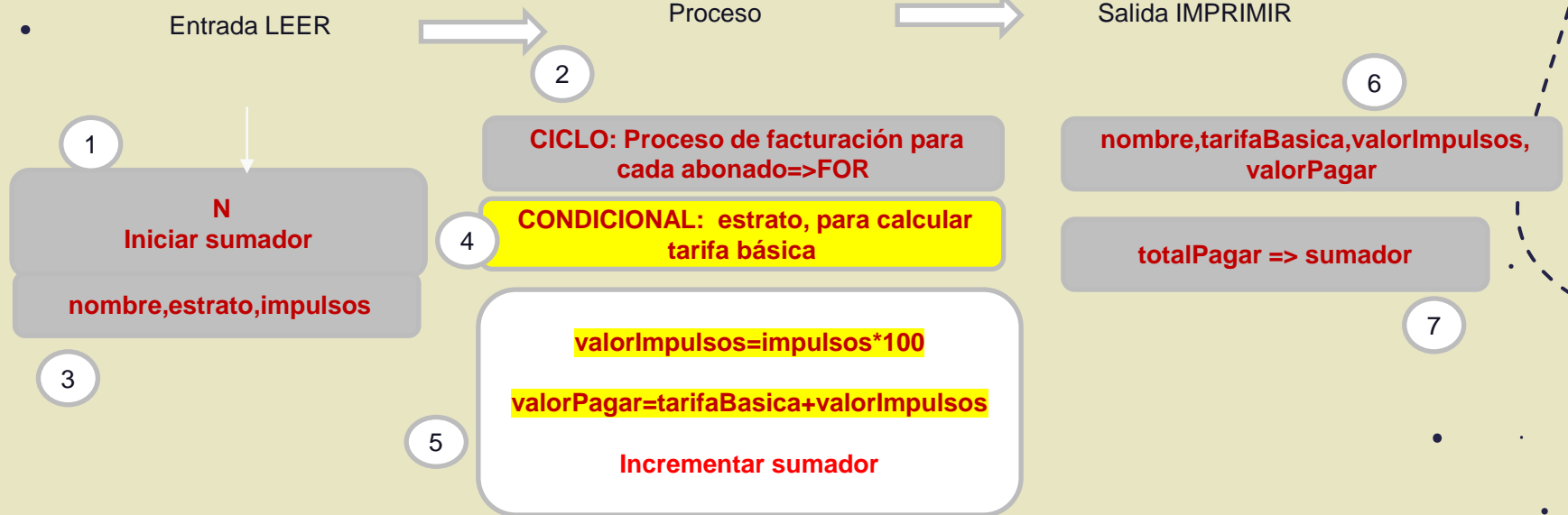


Análisis -> Ejercicio funciones

Entrada LEER

Proceso

Salida IMPRIMIR



Funciones

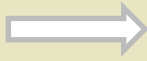
Ejercicio



Análisis - Modularidad

Parámetros de entrada

estrato
impulsos



Nombre: facturación:

4



Parte de

5

Parámetros de salida

valorPagar

FUNCIÓN: Retorno

Funciones

Ejercicio

Programa -Versión 1 -Función



```
# Programa para manejo de funciones
# Autor: Sergio Medina
# Fecha: 15/06/2022

# Definición de las funciones
def facturacion_abonado(estrato,impulsos):
    if estrato==1:
        tarifa_basica=10000
    elif estrato==2:
        tarifa_basica=15000
    elif estrato==3:
        tarifa_basica=20000
    elif estrato==4:
        tarifa_basica=25000
    else:
        tarifa_basica=30000
    valor_impulsos=impulsos*100
    valor_abonado=tarifa_basica+valor_impulsos
    return valor_abonado
#Programa principal
```

Funciones

Programa Versión 1: Principal



Ejercicio

```
#Programa principal
N=int(input("Cantidad de abonados: "))
total_abonados=0
for i in range(N):
    nombre=input("Nombre abonado: ")
    estrato=int(input("Estrato(1,2,3,4,5,6): "))
    impulsos=int(input("Impulsos: "))
    #Llamado o la ejecución de la función
    valor_abonado=facturacion_abonado(estrato,impulsos)
    total_abonados+=valor_abonado
    print("Nombre abonado: ",nombre)
    print("Valor a pagar abonado: ", "{:,.2f}".format(valor_abonado))
print("Toal a pagar por todos los abonados: ", "{:,.2f}".format(total_abonados))
```

Ejercicio

```
# Definición de las funciones
def facturacion_abonado(estrato, impulsos):
    if estrato==1:
        tarifa_basica=10000
    elif estrato==2:
        tarifa_basica=15000
    elif estrato==3:
        tarifa_basica=20000
    elif estrato==4:
        tarifa_basica=25000
    else:
        tarifa_basica=30000
    valor_impulsos=impulsos*100
    valor_abonado=tarifa_basica+valor_impulsos
    return valor_abonado
```

```
def valida_entero(etiqueta):
    while True:
        try:
            dato=int(input(etiqueta))
            break
        except ValueError:
            print(etiqueta, " debe ser un dato entero")
    return dato

def valida_estrato(etiqueta):
    while True:
        try:
            estrato=int(input(etiqueta))
            if estrato<1 or estrato>5:
                print(etiqueta, " debe estar entre 1 y 5")
                continue
            break
        except ValueError:
            print(etiqueta, " debe ser un dato entero")
    return estrato
```


Funciones

Programa Versión 2 (Validaciones) Principal



Ejercicio

```
#Programa principal
N=valida_entero("Cantidad de abonados ")
total_abonados=0
for i in range(N):
    nombre=input("Nombre abonado: ")
    estrato=valida_estrato("Estrato(1,2,3,4,5) ")
    impulsos=valida_entero("Impulsos ")
    #Llamado o la ejecución de la función
    valor_abonado=facturacion_abonado(estrato,impulsos)
    total_abonados+=valor_abonado
    print("Nombre abonado: ",nombre)
    print("Valor a pagar abonado: ", "{:,.2f}".format(valor_abonado))
print("Toal a pagar por todos los abonados: ", "{:,.2f}".format(total_abonados))
```

Funciones

Ejercicio

Programa Versión 3 (Retorno varios valores) Funciones



```
# Programa para manejo de funciones
# Autor: Sergio Medina
# Fecha: 15/06/2022

# Definición de las funciones
def facturacion_abonado(estrato,impulsos):
    if estrato==1:
        tarifa_basica=10000
    elif estrato==2:
        tarifa_basica=15000
    elif estrato==3:
        tarifa_basica=20000
    elif estrato==4:
        tarifa_basica=25000
    else:
        tarifa_basica=30000
    valor_impulsos=impulsos*100
    valor_abonado=tarifa_basica+valor_impulsos
    return tarifa_basica,valor_impulsos,valor_abonado
```

```
def valida_entero(etiqueta):
    while True:
        try:
            dato=int(input(etiqueta))
            break
        except ValueError:
            print(etiqueta," debe ser un dato entero")
    return dato

def valida_estrato(etiqueta):
    while True:
        try:
            estrato=int(input(etiqueta))
            if estrato<1 or estrato>5:
                print(etiqueta," debe estar entre 1 y 5")
                continue
            break
        except ValueError:
            print(etiqueta," debe ser un dato entero")
    return estrato
```

Funciones

Programa Versión 3 ((Retorno varios valores)) Principal



Ejercicio

```
#Programa principal
N=valida_entero("Cantidad de abonados ")
total_abonados=0
for i in range(N):
    nombre=input("Nombre abonado: ")
    estrato=valida_estrato("Estrato(1,2,3,4,5) ")
    impulsos=valida_entero("Impulsos ")
    #Llamado o la ejecución de la función
    tarifa_basica,valor_impulsos,valor_abonado=facturacion_abonado(estrato,impulsos)
    total_abonados+=valor_abonado
    print("Nombre abonado: ",nombre)
    print("Tarifa Básica: ", "{:,.2f}".format(tarifa_basica))
    print("Valor impulsos: ", "{:,.2f}".format(valor_impulsos))
    print("Valor a pagar abonado: ", "{:,.2f}".format(valor_abonado))
print("Toal a pagar por todos los abonados: ", "{:,.2f}".format(total_abonados))
```

Talleres:

Realizar los ejercicios usando Funciones

Funciones

Ejercicio



Dado el nombre y estrato (1,2,3,4,5) de un usuario del servicio de energía eléctrica, calcular lo que pagaría de tarifa básica del servicio de energía eléctrica, que depende del estrato, así

Estrato	Tarifa Básica
1	\$10.000
2	\$15.000
3	\$30.000
4	\$50.000
5	\$65.000

Se pide visualizar el nombre y tarifa básica

Funciones

Ejercicio



Situación problema Liquidación servicio de matrícula

Se tiene una la información sobre 1 estudiante de una institución de educación para el trabajo, que realizará su proceso de matrícula financiera. La información que se conoce del estudiante es la siguiente:

- Código
- Nombre
- Programa a académico al cual pertenece, que puede ser
 - 1: Técnico en Sistemas
 - 2: Técnico en Desarrollo de videojuegos
 - 3: Técnico en Animación Digital
- Indicador de Beca, puede ser:
 - 1: Beca por rendimiento académico. Descuento del 50% sobre el valor matrícula.
 - 2: Beca Cultural – Deportes. Descuento del 40% sobre el valor matrícula
 - 3: Sin Beca.

También nos suministran el cuadro de valores de matrícula que depende del programa académico que cursa el estudiante, así:

Programa académico	Valor Matrícula
Técnico en Sistemas	\$800.000
Técnico en Desarrollo de videojuegos	\$1.000.000
Técnico en Animación Digital	\$1.200.000

Se pide calcular el valor neto a pagar de matrícula para el estudiante e imprimir el nombre y el valor a pagar por matrícula.

Funciones

Ejercicio



Situación problema 2: Comisión Vendedores

Se tiene una lista de vendedores de una empresa , sobre las ventas realizadas en el mes. La información que se conoce de cada vendedor es la siguiente:

- Cédula de ciudadanía
- Nombre
- Tipo de vendedor, que puede ser
 - 1: Puerta a Puerta
 - 2: Telemarketing
 - 3: Ejecutivo de ventas
- Valor ventas realizadas en el mes

NOTA: La lista termina cuando la cédula de ciudadanía es -1

También nos suministran el porcentaje de comisión que se le aplica a las ventas realizadas en el mes, para el cálculo de la comisión, de acuerdo al tipo de vendedor así:

- Si el vendedor es puerta a puerta, el porcentaje de comisión es del 20% sobre el valor de las ventas realizadas en el mes.
- Si el vendedor es telemarketing, el porcentaje de comisión es del 15% sobre el valor de las ventas realizadas en el mes.
- Si el vendedor es ejecutivo de ventas, el porcentaje de comisión es del 25% sobre el valor de las ventas realizadas en el mes.

Se pide calcular el valor a pagar por comisión de cada vendedor, el valor total de las ventas del mes, el valor total a pagar por comisiones

Funciones

Ejercicio



La empresa TRASRAPIDO desea liquidar en forma automática la comisión de N conductores por el servicio realizado en el mes. Para ello suministra la siguiente información del conductor: Cédula, Nombre, Clase conductor: 1: Experto, 2: Novato, Valor total por concepto de pasajes del mes, Valor total por concepto de encomiendas del mes

Se pide calcular el pago al conductor, que depende de un pago por concepto de pasajes más el un pago por concepto encomiendas, así:

Clase conductor	Porcentaje comisión (Sobre el valor total de pasajes del mes)
1 – Experto	30%
2 – Novato	20%

Clase conductor	Porcentaje comisión (Sobre el valor total de Encomiendas del mes)
1 – Experto	20%
2 – Novato	15%

También se debe calcular el valor total a pagar (Todos los conductores) y la cantidad de conductores novatos y expertos

Funciones

Variables de Control - Acumuladores



Situación problema: Liquidación de Honorarios Docente

Se tiene la siguiente información de los N docentes de una institución educativa:

- Documento de identidad
- Nombre
- Categoría docente(A,B o C)
- Horas laboradas en el mes



También suministran el valor de la hora que la institución paga a los docentes, dependiendo de su categoría, así:
(Categoría – Valor hora): (A - \$25.000, B - \$35.000, C - \$50.000)

Con base en la información suministrada se pide:

- Valor a pagar por honorarios para cada docente
- Valor total a pagar (Todos los docentes)
- Cantidad de docentes de cada una de las categorías.

Funciones

Situación problema: Liquidación comisión vendedores

Se tiene la siguiente información de los N vendedores de una organización. De cada uno suministran:

- Documento de identidad
- Nombre
- Tipo vendedor (1=Puerta a puerta, 2=Telemercadeo, 3=Ejecutivo de ventas)



Además, para cada vendedor se suministra información sobre las M ventas que realiza. De cada venta se conoce:

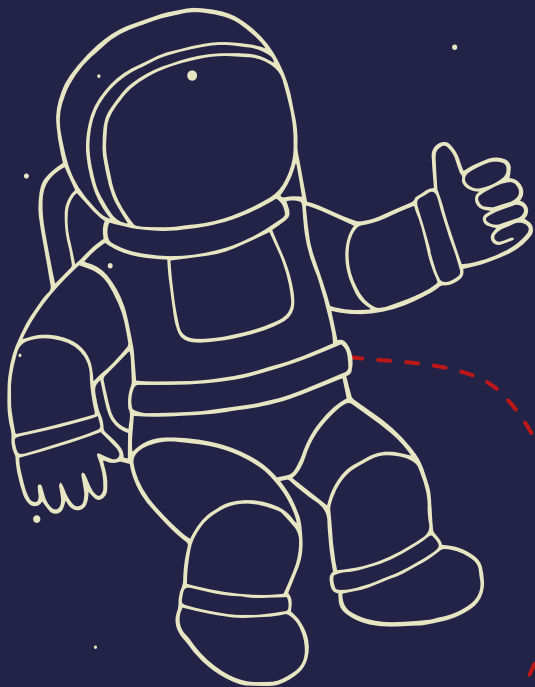
- Nombre del cliente
- Código del cliente
- Tipo Venta (1=Contado, 2=Crédito)
- Valor de la venta

También se suministra la tabla para liquidar comisiones, teniendo en cuenta el tipo de vendedor y el tipo de venta:

	Contado	Crédito
Puerta a Puerta	25%	20%
Telemercadeo	15%	10%
Ejecutivo de venta	20%	15%

Con base en la información suministrada se pide:

- Valor a Pagar a cada vendedor por concepto de comisiones
- Valor total a pagar por comisiones (Todos los vendedores)
- Valor total ventas por cada vendedor.



Programa académico CAMPUS



Ciclo 1:
Fundamentos de
Programación

