

NIVEL 1

SEPARACIÓN DE LA INTERFAZ & LA LÓGICA DEL PROGRAMA
(CREACIÓN DE UNA INTERFAZ BASADA EN CONSOLA)



CONTEXTO



El programa debe ser correcto

- Seguro
- Tener buen desempeño
- Fácil de usar (usabilidad)

Este último, está directamente relacionado con la **Interfaz de usuario**

- Escalable
- Tolerante a fallos
- Pero lo más importante es qué tan fácil es hacer un cambio o en otras qué tan **mantenible**

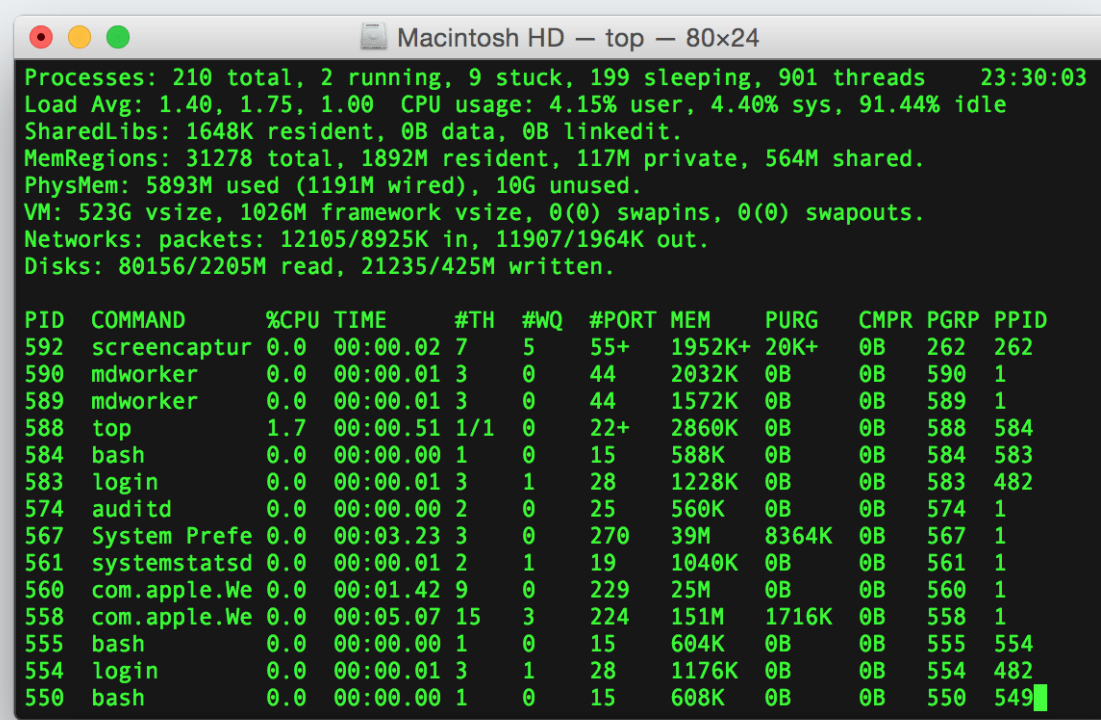
INTERFAZ DE USUARIO

- ✓ La **interfaz de usuario** es el medio a través del cual el usuario puede comunicarse con un programa
- ✓ Existen varios tipos, pero en este curso nos centraremos únicamente en las Interfaces basadas en consola



INTERFAZ BASADA EN CONSOLA

Las interfaces basadas en consola permiten a los usuarios dar instrucciones o ejecutar acciones sobre un programa por medio de una línea de texto simple, y su aspecto es el de una consola de comandos



```

Macintosh HD — top — 80x24
Processes: 210 total, 2 running, 9 stuck, 199 sleeping, 901 threads   23:30:03
Load Avg: 1.40, 1.75, 1.00  CPU usage: 4.15% user, 4.40% sys, 91.44% idle
SharedLibs: 1648K resident, 0B data, 0B linkedit.
MemRegions: 31278 total, 1892M resident, 117M private, 564M shared.
PhysMem: 5893M used (1191M wired), 10G unused.
VM: 523G vsize, 1026M framework vsize, 0(0) swapins, 0(0) swapouts.
Networks: packets: 12105/8925K in, 11907/1964K out.
Disks: 80156/2205M read, 21235/425M written.

  PID  COMMAND      %CPU  TIME    #TH   #WQ   #PORT  MEM    PURG    CMPR  PGRP  PPID
  ---  -
592  screencaptur  0.0   00:00.02  7     5    55+   1952K+ 20K+   0B    262   262
590  mdworker      0.0   00:00.01  3     0    44    2032K  0B     0B    590   1
589  mdworker      0.0   00:00.01  3     0    44    1572K  0B     0B    589   1
588  top           1.7   00:00.51  1/1   0    22+   2860K  0B     0B    588   584
584  bash          0.0   00:00.00  1     0    15    588K   0B     0B    584   583
583  login         0.0   00:00.01  3     1    28    1228K  0B     0B    583   482
574  auditd        0.0   00:00.00  2     0    25    560K   0B     0B    574   1
567  System Prefe  0.0   00:03.23  3     0    270   39M    8364K  0B     0B    567   1
561  systemstatsd  0.0   00:00.01  2     1    19    1040K  0B     0B    561   1
560  com.apple.We  0.0   00:01.42  9     0    229   25M    0B     0B    560   1
558  com.apple.We  0.0   00:05.07  15    3    224   151M   1716K  0B     0B    558   1
555  bash          0.0   00:00.00  1     0    15    604K   0B     0B    555   554
554  login         0.0   00:00.01  3     1    28    1176K  0B     0B    554   482
550  bash          0.0   00:00.00  1     0    15    608K   0B     0B    550   549
  
```

SEPARACIÓN ENTRE INTERFAZ DE USUARIO Y LÓGICA DEL PROGRAMA

Para que un programa sea fácil de entender, mantener y extender es recomendable que **su lógica** sea independiente de **la interfaz**.

- ✓ **La interfaz** se encarga de la comunicación con el usuario (entrada y salida de datos) y se implementa en un archivo separado que contiene el programa principal
- ✓ **La lógica** se encuentra implementada en funciones que componen módulos, los cuales se usan o llaman desde la interfaz





Conversión dólares ↔ pesos

RECORDEMOS EL EJEMPLO...

CONVERSIÓN DÓLARES ↔ PESOS

Lógica del problema

La lógica del programa está implementada en un único módulo llamado «**libreria**»

```
libreria.py
1 |
2 def convertir_a_dolares(pesos: float, trm: float)-> float:
3     return pesos / trm
4
5
6 def convertir_a_pesos(dolares: float, trm: float)-> float:
7     return dolares * trm
8
```

Interfaz

```
interfaz_convertidor_monedas.py
1 import libreria as lb
2
3 def ejecutar_convertir_a_dolares(trm: float)->None:
4     pesos = float(input("Ingrese la cantidad de pesos: "))
5     dolares = lb.convertir_a_dolares(pesos, trm)
6     print(pesos, "pesos son", round(dolares,2) , "dólares")
7
8 def ejecutar_convertir_a_pesos(trm: float)->None:
9     dolares = float(input("Ingrese la cantidad de dólares: "))
10    pesos = lb.convertir_a_pesos(dolares, trm)
11    print(dolares, "dólares son", round(pesos,0) , "pesos")
12
13 def iniciar_aplicacion ()->None:
14     trm = float(input("Ingrese la TRM: "))
15     ejecutar_convertir_a_dolares(trm)
16     ejecutar_convertir_a_pesos(trm)
17
18 iniciar_aplicacion()
```

La interfaz está implementada, en el módulo:«**interfaz_convertidor_monedas**». Note que este módulo tiene una función **iniciar_aplicacion()** y las demás funciones se llaman **ejecutar_**

INTERFAZ BASADA EN CONSOLA



- La interfaz de usuario se debe implementar en su propio archivo
- Todos los **print** o **input** de nuestro programa se harán en dicho
- Como la lógica o "**inteligencia**" del programa se encuentra en otro módulo (en las funciones definidas en este), desde la interfaz de consola se debe importar dicho módulo y darle un nombre para poder invocar las funciones cuando se necesiten
- A través del llamado de estas funciones, la interfaz puede procesar los datos recibidos del usuario y luego mostrar los resultados en la consola

```
1 import libreria as lb
2
3 def ejecutar_convertir_a_dolares(trm: float)->None:
4     pesos = float(input("Ingrese la cantidad de pesos: "))
5     dolares = lb.convertir_a_dolares(pesos, trm)
6     print(pesos, "pesos son", round(dolares,2) , "dólares")
7
8 def ejecutar_convertir_a_pesos(trm: float)->None:
9     dolares = float(input("Ingrese la cantidad de dólares: "))
10    pesos = lb.convertir_a_pesos(dolares, trm)
11    print(dolares, "dólares son", round(pesos,0) , "pesos")
12
13 def iniciar_aplicacion ()->None:
14     trm = float(input("Ingrese la TRM: "))
15     ejecutar_convertir_a_dolares(trm)
16     ejecutar_convertir_a_pesos(trm)
17
18 iniciar_aplicacion()
```


CONVERSIÓN DÓLARES ↔ PESOS CON CONDICIONALES

Lógica del problema

```

libreria.py
1 |
2 def convertir_a_dolares(pesos: float, trm: float)-> float:
3     return pesos / trm
4
5
6 def convertir_a_pesos(dolares: float, trm: float)-> float:
7     return dolares * trm
8
  
```

El módulo que contiene la lógica del problema no ha cambiado pero analicemos el módulo de la interfaz



Interfaz

```

interfaz_if.py
1
2 import libreria as lb
3
4 def ejecutar_convertir_a_dolares()-> None:
5     pesos = float(input("Ingrese la cantidad de pesos: "))
6     trm = float(input("Ingrese la TRM: "))
7     dolares = lb.convertir_a_dolares(pesos, trm)
8     print(pesos, "pesos son", round(dolares,2) , "dolares")
9
10
11 def ejecutar_convertir_a_pesos()-> None:
12     dolares = float(input("Ingrese la cantidad de dolares: "))
13     trm = float(input("Ingrese la TRM: "))
14     pesos = lb.convertir_a_pesos(dolares, trm)
15     print(dolares, "dolares son", round(pesos,0) , "pesos")
16
17
18 def mostrar_menu()-> str:
19     print("Seleccione una opción")
20     print("1. Convertir de dólares a pesos")
21     print("2. Convertir de pesos a dólares")
22     opcion = input("Opción seleccionada: ")
23     return opcion
24
25
26 def iniciar_aplicacion()-> None:
27     opcion = mostrar_menu()
28     if opcion == "1":
29         ejecutar_convertir_a_pesos()
30     elif opcion == "2":
31         ejecutar_convertir_a_dolares()
32     else:
33         print("La opción seleccionada no es válida")
34
35 iniciar_aplicacion()
36
  
```

CONVERSIÓN DÓLARES ↔ PESOS CON CONDICIONALES

```

interfaz_if.py
1
2 import libreria as lb
3
4 def ejecutar_convertir_a_dolares()-> None:
5     pesos = float(input("Ingrese la cantidad de pesos: "))
6     trm = float(input("Ingrese la TRM: "))
7     dolares = lb.convertir_a_dolares(pesos, trm)
8     print(pesos, "pesos son", round(dolares,2) , "dolares")
9
10
11 def ejecutar_convertir_a_pesos()-> None:
12     dolares = float(input("Ingrese la cantidad de dolares: "))
13     trm = float(input("Ingrese la TRM: "))
14     pesos = lb.convertir_a_pesos(dolares, trm)
15     print(dolares, "dolares son", round(pesos,0) , "pesos")
16
17
18 def mostrar_menu()-> str:
19     print("Seleccione una opción")
20     print("1. Convertir de dólares a pesos")
21     print("2. Convertir de pesos a dólares")
22     opcion = input("Opción seleccionada: ")
23     return opcion
24
25
26 def iniciar_aplicacion()-> None:
27     opcion = mostrar_menu()
28     if opcion == "1":
29         ejecutar_convertir_a_pesos()
30     elif opcion == "2":
31         ejecutar_convertir_a_dolares()
32     else:
33         print("La opción seleccionada no es válida")
34
35 iniciar_aplicacion()
36

```

- Tenemos la función **iniciar_aplicación**, que se encarga de mostrar el menú y llamar a la función que corresponda con la opción digitada por el usuario
- Tenemos la función **mostrar_menú**, que imprime las opciones del programa y retorna la opción tecleada por el usuario
- Las funciones **ejecutar_** que se encargan de hacer la conexión con la lógica del programa retornando la respuesta que se mostrará al usuario

CONVERSIÓN DÓLARES ↔ PESOS CON INSTRUCCIONES REPETITIVAS

Lógica del problema

```

1 |
2 def convertir_a_dolares(pesos: float, trm: float)-> float:
3     return pesos / trm
4
5
6 def convertir_a_pesos(dolares: float, trm: float)-> float:
7     return dolares * trm
8

```

El módulo que contiene la lógica del problema no ha cambiado pero analicemos el módulo de la interfaz



Interfaz

```

interfaz_while.py
1 |
2 import libreria as lb
3
4 def ejecutar_convertir_a_dolares()-> None:
5     pesos = float(input("Ingrese la cantidad de pesos: "))
6     trm = float(input("Ingrese la TRM: "))
7     dolares = lb.convertir_a_dolares(pesos, trm)
8     print(pesos, "pesos son", round(dolares,2) , "dolares")
9
10
11 def ejecutar_convertir_a_pesos()-> None:
12     dolares = float(input("Ingrese la cantidad de dolares: "))
13     trm = float(input("Ingrese la TRM: "))
14     pesos = lb.convertir_a_pesos(dolares, trm)
15     print(dolares, "dolares son", round(pesos,0) , "pesos")
16
17
18 def mostrar_menu()-> str:
19     print("Seleccione una opción")
20     print("1. Convertir de dólares a pesos")
21     print("2. Convertir de pesos a dólares")
22     print("3. Salir")
23     opcion = input("Opción seleccionada: ")
24     return opcion
25
26
27 def iniciar_aplicacion()-> None:
28     continuar = True
29     while continuar:
30         opcion = mostrar_menu()
31         if opcion == "1":
32             ejecutar_convertir_a_pesos()
33         elif opcion == "2":
34             ejecutar_convertir_a_dolares()
35         elif opcion == "3":
36             continuar = False
37         else:
38             print("La opción seleccionada no es válida")
39
40
41 iniciar_aplicacion()
42

```

CONVERSIÓN DÓLARES ↔ PESOS CON INSTRUCCIONES REPETITIVAS

```
interfaz_while.py
1|
2| import libreria as lb
3|
4| def ejecutar_convertir_a_dolares()-> None:
5|     pesos = float(input("Ingrese la cantidad de pesos: "))
6|     trm = float(input("Ingrese la TRM: "))
7|     dolares = lb.convertir_a_dolares(pesos, trm)
8|     print(pesos, "pesos son", round(dolares,2) , "dolares")
9|
10|
11| def ejecutar_convertir_a_pesos()-> None:
12|     dolares = float(input("Ingrese la cantidad de dolares: "))
13|     trm = float(input("Ingrese la TRM: "))
14|     pesos = lb.convertir_a_pesos(dolares, trm)
15|     print(dolares, "dolares son", round(pesos,0) , "pesos")
16|
17|
18| def mostrar_menu()-> str:
19|     print("Seleccione una opción")
20|     print("1. Convertir de dólares a pesos")
21|     print("2. Convertir de pesos a dólares")
22|     print("3. Salir")
23|     opcion = input("Opción seleccionada: ")
24|     return opcion
25|
26|
27| def iniciar_aplicacion()-> None:
28|     continuar = True
29|     while continuar:
30|         opcion = mostrar_menu()
31|         if opcion == "1":
32|             ejecutar_convertir_a_pesos()
33|         elif opcion == "2":
34|             ejecutar_convertir_a_dolares()
35|         elif opcion == "3":
36|             continuar = False
37|         else:
38|             print("La opción seleccionada no es válida")
39|
40|
41| iniciar_aplicacion()
42|
```

La función **iniciar_aplicación** luce diferente! Ahora el usuario puede probar varias opciones sin tener que volver a ejecutar el programa