

<b>Estado</b>	Finalizado
<b>Comenzado</b>	martes, 23 de septiembre de 2025, 17:50
<b>Completado</b>	martes, 23 de septiembre de 2025, 19:34
<b>Duración</b>	1 hora 44 minutos
<b>Calificación</b>	Sin calificar aún

**Pregunta 1**

Correcta

Se puntuá 2,00 sobre 2,00

El paradigma Logico, se caracteriza por hacer hincapié en la aplicación de las funciones y composición entre ellas, más que en los cambios de estados y la ejecución secuencial de comandos.

Seleccione una:

- Verdadero  
 Falso

La respuesta correcta es 'Falso'

**Pregunta 2**

Correcta

Se puntuá 2,00 sobre 2,00

En el **paradigma orientado a objetos**, un programa esta compuesto por un conjunto de "**reglas**" y "**hechos**" que permiten la resolución del problema.

Seleccione una:

- Verdadero  
 Falso

La respuesta correcta es 'Falso'

**Pregunta 3**

Correcta

Se puntuá 2,00 sobre 2,00

En Python, para que una carpeta pueda ser considerada un "paquete", debe contener un archivo de inicio llamado `__init__.py`

- Verdadero   
 Falso

La respuesta correcta es 'Verdadero'

**Pregunta 4**

Correcta

Se puntuá 2,00 sobre 2,00

En Python un paquete, es una carpeta que contiene archivos .pys

- Verdadero  
 Falso 

La respuesta correcta es 'Falso'

**Pregunta 5**

Correcta

Se puntuá 2,00 sobre 2,00

En un diccionario puede haber 2 elementos con la misma clave.

- Verdadero  
 Falso 

La respuesta correcta es 'Falso'

**Pregunta 6**

Correcta

Se puntuá 2,00 sobre 2,00

La sentencia: diccionario['valor'] = 'clave' significa que se le actualizara el valor 'valor' al elemento cuya clave es 'clave'.

- Verdadero  
 Falso 

La respuesta correcta es 'Falso'

**Pregunta 7**

Correcta

Se puntuá 2,00 sobre 2,00

La sentencia "import" en Python se utiliza para incorporar el contenido de un modulo

- Verdadero   
 Falso

La respuesta correcta es 'Verdadero'

**Pregunta 8**

Correcta

Se puntuá 2,00 sobre 2,00

La sentencia: shopping.insert(0, 'Jamon') permite agregar el elemento "Jamon" como "primer" elemento de la lista "shopping".

 Verdadero  Falso

La respuesta correcta es 'Verdadero'

**Pregunta 9**

Incorrecta

Se puntuá 0,00 sobre 2,00

Las tuplas NO se pueden usar como claves de diccionarios

 Verdadero  Falso

La respuesta correcta es 'Falso'

**Pregunta 10**

Correcta

Se puntuá 2,00 sobre 2,00

Para declarar una función se debe utilizar la sentencia "def".

 Verdadero  Falso

La respuesta correcta es 'Verdadero'

**Pregunta 11**

Correcta

Se puntuá 2,00 sobre 2,00

Un error SEMANTICO, podemos capturarlo y controlarlo, evitando así que el programa se cierre de manera "abrupta".

 Verdadero Falso 

La respuesta correcta es 'Falso'

**Pregunta 12**

Correcta

Se puntuá 2,00 sobre 2,00

Un programa interpretado se caracteriza por generar un archivo ejecutable, como consecuencia de su proceso de traducción.

 Verdadero Falso 

La respuesta correcta es 'Falso'

**Pregunta 13**

Correcta

Se puntuá 10,00 sobre 10,00

```
def examen(,B):
    for j in range (2,5,1):
        B[j]=B[j]-B[0];

A=[100,200,300,400,500]
A[0]=10
A[3]=(A[0]+A[1])*2

examen(A)
for i in range(len(A))
    print(A[i], end=", ")
```

Seleccione una:

- a. 100, 200, 300, 400, 500,
- b. 10, 200, 290, 410, 490, 
- c. 2, 25, 35, 54, 55,
- d. 2, 25, 37, 56, 57,
- e. 10, 200, 290, 400, 490,

Respuesta correcta

La respuesta correcta es: 10, 200, 290, 410, 490,

**Pregunta 14**

Correcta

Se puntuá 10,00 sobre 10,00

Se solicita registrar 5 notas para una materia en particular, luego de ello debe obtener el promedio de ellas. Para esto se necesita que la carga de la información sea realizada en una función, al igual que el cálculo del promedio.

Seleccione una:

- a. def cargar\_datos(materias\_local):

```
    nombre_materia = input("Ingrese el Nombre de la Materia: ")
    notas=[]
    for indice_nota in range(5):
        unaNota = int(input("Ingrese una Nota: "))
        notas.append(unaNota)

    materias_local[nombre_materia]=notas

def promedio(materias_local):
    sumatoria = 0
    for notas in materias_local.values():
        for nota in notas:
            sumatoria += nota

    return sumatoria/5

materias = {}

cargar_datos(materias)
resultado_promedio = promedio(materias)
print("El Promedio de Notas es:", resultado_promedio)
```

- b. def cargar\_datos(materias\_local):

```
    nombre_materia = input("Ingrese el Nombre de la Materia: ")
    notas=[]
    for indice_nota in range(5):
        unaNota = int(input("Ingrese una Nota: "))
        notas.append(unaNota)

    materias_local[nombre_materia]=notas

def promedio(materias_local):
    sumatoria = 0
    for notas in materias_local.values():
        for nota in notas:
            sumatoria += nota

    return sumatoria
```

```
materias = {}

cargar_datos(materias)
resultado_promedio = promedio(materias)
print("El Promedio de Notas es:", resultado_promedio)
```

- c. def cargar\_datos(materias\_local):

```
    nombre_materia = input("Ingrese el Nombre de la Materia: ")
    notas=[]
    for indice_nota in range(5):
        unaNota = int(input("Ingrese una Nota: "))
        notas.append(unaNota)

    materias_local[nombre_materia]=notas
```

```
def promedio(materias_local):
    promedio=0
    for notas in materias_local.values():
        for nota in notas:
            promedio+= nota/2

    return promedio
```

```
materias = {}
```

```
cargar_datos(materias)
resultado_promedio = promedio(materias)
print("El Promedio de Notas es:", resultado_promedio)
```

d. def cargar\_datos(materias\_local):
 nombre\_materia = input("Ingrese el Nombre de la Materia: ")
 notas=[]
 for indice\_nota in range(5):
 unaNota = int(input("Ingrese una Nota: "))
 notas.append(unaNota)

```
materias_local[nombre_materia]=notas
```

```
def promedio(materias_local):
    promedio= 0
    for notas in materias_local.values():
        for nota in notas:
            promedio+= nota/2
```

```
return promedio/5
```

```
materias = {}
```

```
cargar_datos(materias)
resultado_promedio = promedio(materias)
print("El Promedio de Notas es:", resultado_promedio)
```

Respuesta correcta

La respuesta correcta es:

```
def cargar_datos(materias_local):
    nombre_materia = input("Ingrese el Nombre de la Materia: ")
    notas=[]
    for indice_nota in range(5):
        unaNota = int(input("Ingrese una Nota: "))
        notas.append(unaNota)
```

```
materias_local[nombre_materia]=notas
```

```
def promedio(materias_local):
    sumatoria = 0
    for notas in materias_local.values():
        for nota in notas:
            sumatoria += nota
```

```
return sumatoria/5
```

```
materias = {}
```

```
cargar_datos(materias)
resultado_promedio = promedio(materias)
print("El Promedio de Notas es:", resultado_promedio)
```

**Pregunta 15**

Correcta

Se puntuá 10,00 sobre 10,00

```
def suma_ventas():
    ventas = {
        'papas': 100000,
        'camote': 110000,
        'zanahoria': 125000,
        'banana': 140000,
    }
    suma_total = 0
    for valor_venta in ventas.values():
        suma_total += valor_venta
    return suma_total

calculo_sumatoria = suma_ventas()
print ("Sumatoria de todas las Ventas devuelta por la Funcion: ", calculo_sumatoria)
```

 suma\_venta() suma  ventas.item()  total  while valor  for valor

Respuesta correcta

La respuesta correcta es:

```
[def] suma_ventas():
[ventas ]= {
    'papas': 100000,
    'camote': 110000,
    'zanahoria': 125000,
    'banana': 140000,
}
suma_total = 0
[for valor_venta] in [ventas.values()]:
    [suma_total ] += [valor_venta]
return suma_total

calculo_sumatoria = [suma_ventas()]
print ("Sumatoria de todas las Ventas devuelta por la Funcion: ", calculo_sumatoria)
```

**Pregunta 16**

Correcta

Se puntuó 14,00 sobre 14,00

**Cree un programa que contenga una Funcion la cual reciba una lista de valores enteros y devuelva otra lista sólo con aquellos valores pares.**

```
def NumerosPares(p_lista):
    lista_pares = []
    for nro in p_lista:
        if nro%2==0:
            lista_pares.append(nro)
    return lista_pares

lista = []
try:
    numero = int(input("Ingrese un numero (cero para terminar de ingresar): "))
    while numero>0:
        lista.append(numero)
    numero = int(input("Ingrese un numero (cero para terminar de ingresar): "))
resultado_lista = NumerosPares(lista)

if len(resultado_lista) >0:
    print("LOS NUMEROS PARES INGRESADOS SON : ",resultado_lista)
else:
    raise Exception ("**NO EXISTEN NUMEROS PARES**")

except ValueError:
    print("SOLO DEBIA INGRESAR NUMEROS ENTEROS")
except Exception as mierror:
    print("ERROR: ", mierror)
```

ZeroDivisionError

raise except	NumerosPares(p_lista)	catch:	{}
--------------	-----------------------	--------	----

Respuesta correcta

La respuesta correcta es:

**Cree un programa que contenga una Funcion la cual reciba una lista de valores enteros y devuelva otra lista sólo con aquellos valores pares.**

```
def NumerosPares(p_lista):
    lista_pares=[[]]
    for nro in [p_lista]:
        if [nro%2==0]:
            lista_pares.append([nro])
    return lista_pares

lista = []
[try:]:
    numero = int(input("Ingrese un numero (cero para terminar de ingresar): "))
    [while numero>0:
        [lista].append([numero])
        numero = int(input("Ingrese un numero (cero para terminar de ingresar): "))

    resultado_lista = [NumerosPares(lista)]
    if [len(resultado_lista)]>0:
        print("LOS NUMEROS PARES INGRESADOS SON : ",resultado_lista)
    else:
```

```
[raise Exception] ("**NO EXISTEN NUMEROS PARES**")
```

```
[except] [ValueError]:  
    print("SOLO DEBIA INGRESAR NUMEROS ENTEROS")  
except [Exception as mierror]:  
    print("ERROR: ", mierror)
```

**Pregunta 17**

Finalizado

Se puntuó como 0 sobre 32,00

Un club de futbol necesita registrar información relacionada a los jugadores que van a participar de una fecha en particular del torneo local. La información que se debe registrar es la siguiente: *edad del jugador, nombre completo del jugador, categoría (primera, segunda, tercera), posición en la que juega (arquero, defensor, mediocampista, delantero)*, estadio donde juega. Dicha información se debe registrar en el siguiente formato:

```
partidos = [  
    {'edad': 10, 'nombre': 'Juan Pérez', 'categoria': 1, 'posicion': 'Delantero', 'estadio': 'Malvinas Argentinas'},  
    {'edad': 11, 'nombre': 'Carlos Gómez', 'categoria': 2, 'posicion': 'Mediocampista', 'estadio': 'La Bombonera'}  
]
```

Las acciones a realizar son:

- Deberá crear un paquete con al menos 2 módulos, en los cuales se deben encontrar las funciones que resuelven los siguientes puntos.
- Carga de partidos (al menos 3).
- Mostrar la “cantidad” partidos que se juegan en un estadio en particular.
- Mostrar el nombre de todos los jugadores que juegan de delanteros.
- Mostrar el promedio de edad de los que juegan en una categoría en particular.

 [Parcial.7z](#)