|  |  |  |
| --- | --- | --- |
| | GUÍA 2.2.4:Material Complementario Ejercicios Resueltos | | |
| Sigla | Asignatura | Experiencia de Aprendizaje |
| FPY1101 | Fundamentos de Programación | EA2: Programación de aplicaciones en Python |
| Tiempo | Modalidad de Trabajo | Indicadores de logro |
| 4 h | Parejas | IL 2.1 |

|  |
| --- |
| **Código QR con relleno sólido**  **Antecedentes generales** |

## Esta guía tiene como objetivo conocer los aspectos generales en Python

### Ser una guía paso a paso para la construcción de programas en Python

### Todos los ejercicios tienen la solución incluida, pero antes de ver la solución, debes resolver por tu cuenta el ejercicio, de esa manera podrás reforzar y aprender. Las soluciones utilízalas para comparar con tus resultados, tomar nota o simplemente como revisión. debes ser consiente y responsable en tu autoaprendizaje.

### Debate con tu docente las respuestas obtenidas, si tienes dudas, recuerda dar aviso y argumentar, los programas no tienen soluciones únicas, teniendo derivaciones o mecanismos distintos de funcionamiento.

### Esta guía puede ser desarrollada en casa, o guiada por el docente, con el fin de seguir un paso a paso y comprender las explicaciones de la o el docente.

|  |
| --- |
| **Lista con relleno sólido Requerimientos para esta actividad** |

## Para el desarrollo de esta actividad deberás disponer de:

## Computador

## Visual Studio Code

|  |  |
| --- | --- |
| **Inteligencia artificial con relleno sólidoActividad** |  |

**EJERCICIO 1**

Analiza el siguiente código que se encuentra en desorden sintáctico:

print(f"-----LIQUIDACIÓN EMPLEADO----")

print(f"NOMBRE EMPLEADO: {nom}")

print(f"RUT EMPLEADO: {rut}")

#Ingreso de liquidación de sueldo

nom = input ("Ingrese nombre empleado: ")

rut = input ("Ingrese rut empleado: ")

horasTrabajadas = int(input ("Ingrese las horas trabajadas: "))

valorHora = int(input("Ingrese el valor hora del empleado: "))

print(f"MOVILIZACIÓN: {movilizacion}")

print(f"ALIMENTACIÓN: {colacion}")

print(f"PAGO A EMPLEADO: {resultado}")

colacion = int(5500)

movilizacion = int(3000)

resultado = (valorHora \* horasTrabajadas)+colacion+movilizacion

|  |  |
| --- | --- |
| Pregunta | Ordenar los códigos para que el sistema muestre por consola el siguiente resultado. |
| Resultado | #ingreso de liquidación de sueldo  nom = input ("Ingrese nombre empleado: ")  rut = input ("Ingrese rut empleado: ")  horasTrabajadas = int(input ("Ingrese las horas trabajadas: "))  valorHora = int(input("Ingrese el valor hora del empleado: "))  colacion = int(5500)  movilizacion = int(3000)  resultado = (valorHora \* horasTrabajadas)+colacion+movilizacion  print(f"-----LIQUIDACIÓN EMPLEADO----")  print(f"NOMBRE EMPLEADO: {nom}")  print(f"RUT EMPLEADO: {rut}")  print(f"MOVILIZACIÓN: {movilizacion}")  print(f"ALIMENTACIÓN: {colacion}")  print(f"PAGO A EMPLEADO: {resultado}") |
| Feedback | Los códigos de programación deben siempre considerar el orden de Entrada -> Proceso -> Salida En Python y cualquier otro lenguaje de programación debemos programar en el orden:  1.- Solicitar datos de entrada  2.- Procesar la información (ejemplo: operaciones aritméticas, sentencias de lógica)  3.- Mostrar datos de salida |

**EJERCICIO 2**

|  |  |
| --- | --- |
| **Enunciado** | La empresa de seguridad “Seguronet” está desarrollando un sistema para facilitar la creación de contraseñas de sus usuarios.  La idea central es pedir una cierta cantidad de letras o dígitos de información que maneja una persona y luego mezclarlas para generar opciones de contraseña.    Un ejemplo de la utilización se visualiza en la siguiente imagen:  Elija la opción correcta: |
| **Algoritmo** **o resultado** | Algoritmo:  print("Ingrese los siguientes datos:")  nombre = input("Las 2 primeras letras de su primer nombre: ")  apellido = input("Las 2 primeras letras de su segundo apellido: ")  rut = input("Los 2 primeros números de su rut: ")  mes = input("Las 3 letras iniciales del mes de su nacimiento: ")  ciudad = input("Las 2 últimas letras de la ciudad donde vive: ")  opcion1 = nombre + apellido + rut + mes + ciudad  opcion2 = nombre + rut + apellido + mes + ciudad + mes  opcion3 = rut + nombre + mes + ciudad + apellido  opcion4 = apellido + rut + nombre + mes + ciudad + rut  opcion5 = ciudad + apellido + nombre +rut + mes + ciudad  print("")  print(f"La opción 1 de contraseña es: {opcion1}")  print(f"La opción 2 de contraseña es: {opcion2}")  print(f"La opción 3 de contraseña es: {opcion3}")  print(f"La opción 4 de contraseña es: {opcion4}")  print(f"La opción 5 de contraseña es: {opcion5}") |
| **Resultado** | opcion1 = nombre + apellido + rut + mes + ciudad  opcion3 = rut + nombre + mes + ciudad + apellido  opcion5 = ciudad + apellido + nombre + rut + mes + ciudad |
| **Feedback** | -El orden de ejecución es muy importante en un algoritmo. El orden es línea por línea de manera secuencial de arriba hacia abajo. Como sea el orden del programa, eso hará el computador.  -La sintaxis para print en python solicita un f (print(f”texto {var}”)) para indicar que el nombre de la variable entre llaves debe ser sustituida por su valor. De todas formas si existen otras formas de concatenar datos durante la impresión.    Algoritmo correcto:    print("Ingrese los siguientes datos:")  nombre = input("Las 2 primeras letras de su primer nombre: ")  apellido = input("Las 2 primeras letras de su segundo apellido: ")  rut = input("Los 2 primeros números de su rut: ")  mes = input("Las 3 letras iniciales del mes de su nacimiento: ")  ciudad = input("Las 2 últimas letras de la ciudad donde vive: ")    opcion1 = nombre + apellido + rut + mes + ciudad  opcion2 = nombre + rut + apellido + mes + ciudad + mes  opcion3 = rut + nombre + mes + ciudad + apellido  opcion4 = apellido + rut + nombre + mes + ciudad + rut  opcion5 = ciudad + apellido + nombre +rut + mes + ciudad    print("")  print(f"La opción 1 de contraseña es: {opcion1}")  print(f"La opción 2 de contraseña es: {opcion2}")  print(f"La opción 3 de contraseña es: {opcion3}")  print(f"La opción 4 de contraseña es: {opcion4}")  print(f"La opción 5 de contraseña es: {opcion5}") |

**EJERCICIO 3**

|  |  |
| --- | --- |
| **Enunciado** | La empresa de videojuegos “Retro game” está implementando el conocido juego “El gato”.  Se está realizando la implementación en Python vía terminal como se presenta en la siguiente imagen.    Se le solicita establecer cuál es el algoritmo correcto para generar el resultado de la imagen. |
| **Algoritmo** | Algoritmo:  posicionA = "X"  posicionB = ""  posicionC = ""  posicionD = ""  posicionE = "X"  posicionF = ""  posicionG = ""  posicionH = ""  posicionI = "X"  print("")  print("\t\t|\t\t |\t")  Línea Incógnita 1  print("\t\t|\t\t |\t")  print("----------------------------------------------------")  print("\t\t|\t\t |\t")  Línea Incógnita 2  print("\t\t|\t\t |\t")  print("----------------------------------------------------")  print("\t\t|\t\t |\t")  Línea Incógnita 3  print("\t\t|\t\t |\t")  print("") |
| **Resultado** | Línea incógnita 1: print(f"\t{posicionA}\t|\t{posicionB}\t |\t{posicionC}")  Línea incógnita 2: print(f"\t{posicionD}\t|\t{posicionE}\t |\t{posicionF}")  Línea incógnita 3: print(f"\t{posicionG}\t|\t{posicionH}\t |\t{posicionI}") |
| **Feedback** | -Las tabulaciones dentro de la impresión permiten que el texto de salida sea parejo, a la manera de una tabla. Un pequeño error en el código se manifiesta de forma clara durante la impresión.  -Las variables posiciones de la A a la I reflejan donde se imprimirá el carácter dentro del gráfico.    Algoritmo correcto:  posicionA = "X"  posicionB = ""  posicionC = ""  posicionD = ""  posicionE = "X"  posicionF = ""  posicionG = ""  posicionH = ""  posicionI = "X"    print("")  print("\t\t|\t\t |\t")  print(f"\t{posicionA}\t|\t{posicionB}\t |\t{posicionC}")  print("\t\t|\t\t |\t")  print("----------------------------------------------------")  print("\t\t|\t\t |\t")  print(f"\t{posicionD}\t|\t{posicionE}\t |\t{posicionF}")  print("\t\t|\t\t |\t")  print("----------------------------------------------------")  print("\t\t|\t\t |\t")  print(f"\t{posicionG}\t|\t{posicionH}\t |\t{posicionI}")  print("\t\t|\t\t |\t")  print("") |

**EJERCICIO 4**

|  |  |
| --- | --- |
| **Enunciado** | El almacén, “Diego’s” en su afán de incorporar nuevas tecnologías en sus procesos de negocio, le solicita un prototipo de sistema que le permita ingresar los datos de 3 ventas del almacén.  Antes de usted, el almacén le había pedido a un informático comenzar a hacer el sistema, pero este lo dejó botado.  Ahora los dueños desean que usted siga con el desarrollo.  ¿Cuál sería el algoritmo que permite obtener el siguiente resultado (de la imagen)? |
| **Algoritmo** | Algoritmo:  print("Ingresar los datos de la venta")  cliente = input("Ingrese el nombre del cliente: ")  precio1 = int(input("Ingrese el precio del producto1: "))  cantidad 1 = Incógnita 1  precio2 = int(input("Ingrese el precio del producto2: "))  cantidad2 = int(input("Ingrese la cantidad del producto2: "))  precio3 = int(input("Ingrese el precio del producto3: "))  cantidad3 = int(input("Ingrese la cantidad del producto3: "))  descuento = int(input("Ingrese el decuento: "))  total\_bruto = (precio1 \* cantidad1) + (precio2 \* cantidad2) + (precio2 \* cantidad2)  precio\_con\_descuento = round(total\_bruto - (total\_bruto \* descuento/100))  Línea incógnita 2  print("")  print(f"Cliente: \t{cliente}")  print(f"Total bruto: \t${total\_bruto}")  Línea incógnita 3  print(f"Iva: \t\t${iva}")  print(f"Total: \t\t${precio\_con\_descuento + iva}") |
| **Resultado** | Incógnita 1: int(input("Ingrese la cantidad del producto1: "))  Línea incógnita 2: iva = round(precio\_con\_descuento \* 0.19)  Línea incógnita 3: print(f"Iva: \t\t${iva}") |
| **Feedback** | -El orden de ejecución debe ser el adecuado para obtener los resultados esperados.  -Todos los números ingresados deben ser convertidos al tipo de datos entero (int) para poder utilizarlos en fórmulas matemáticas.  -Se deben agregar las tabulaciones para dar un orden apropiado al informe de salida.    Algoritmo correcto:  print("Ingresar los datos de la venta")  cliente = input("Ingrese el nombre del cliente: ")  precio1 = int(input("Ingrese el precio del producto1: "))  cantidad1 = int(input("Ingrese la cantidad del producto1: "))  precio2 = int(input("Ingrese el precio del producto2: "))  cantidad2 = int(input("Ingrese la cantidad del producto2: "))  precio3 = int(input("Ingrese el precio del producto3: "))  cantidad3 = int(input("Ingrese la cantidad del producto3: "))  descuento = int(input("Ingrese el decuento: "))    total\_bruto = (precio1 \* cantidad1) + (precio2 \* cantidad2) + (precio2 \* cantidad2)  precio\_con\_descuento = round(total\_bruto - (total\_bruto \* descuento/100))  iva = round(precio\_con\_descuento \* 0.19)  print("")  print(f"Cliente: \t{cliente}")  print(f"Total bruto: \t${total\_bruto}")  print(f"Total desc.: \t${precio\_con\_descuento}")  print(f"Iva: \t\t${iva}")  print(f"Total: \t\t${precio\_con\_descuento + iva}") |

**EJERCICIO 5**

|  |  |
| --- | --- |
| **Enunciado** | El Art ASCII, es un medio artístico que utiliza recursos computarizados fundamentados en los caracteres de impresión del Código Estándar Estadounidense de Intercambio de Información. (Wikipedia)    Debe elegir la alternativa que contiene la secuencia correcta de variables para formar la figura de Luke sobre el dromedario. |
| **Algoritmo o resultado** | Código:    linea18 =(" [\_,\_]) \ / \|")  linea1 = (" \_\_\_ |\\_\_\_\_\_\_\_\_/)")  linea15 =(" ||'-'/--. / /\\ =| \|\\ \\")  linea11 =(" /|=T=|] / \_\_ \_\_\\")  linea13 =(" |\ ' // |\_ 9 p ]\\")  linea6 =(" | | \_H\_\_/ \_| : |")  linea16 =(" /|| <\/> |\ | '.\_, @ @)<\_)")  linea21 =(" | |\ | | \.\_\_/(\_;\_)")  linea31 =(" | . H | | : '='|")  linea2 =(" .-' | | | | ':")  linea7 =(" \ '.\_\_ \ / ; ';")  linea19 =(" \_\_'-.\_(\_}==.' : ;")  linea71 =(" (\_\_\_| /-' | :. :")  linea371 =(" [.-' \ | \ \ ; :")  linea414 =("| .' '-. | \ /")  linea22 =(" / |==| \ \ / \\_")  linea661 =(" / [ | '.\_\\_ -.\_ \\")  linea517 =(" / \\_\_) \_\_.- \ \ )\\")  linea81 =("/ | /.' >>)")  linea61 =("| \ |\ |")  linea651 =("| / / / / /")  linea51 =(" | /")  linea41 =("") |
| **Resultado** | linea1 linea18 linea11 linea13 linea15 linea16 linea21 linea31  linea6 linea7 linea19 linea71 linea371 linea2 linea22 linea661  linea517 linea81 linea61 linea414 linea651 linea51 linea41 |
| **Feedback** | -La secuencia de las órdenes dentro del algoritmo es de suma importancia.    Algoritmo correcto:  (La figura está deformada debido al procesador de texto, pero en el terminal se ven bien)    print("")  linea1 = (" \_\_\_ |\\_\_\_\_\_\_\_\_/)")  linea18 =(" [\_,\_]) \ / \|")  linea11 =(" /|=T=|] / \_\_ \_\_\\")  linea13 =(" |\ ' // |\_ 9 p ]\\")  linea15 =(" ||'-'/--. / /\\ =| \|\\ \\")  linea16 =(" /|| <\/> |\ | '.\_, @ @)<\_)")  linea21 =(" | |\ | | \.\_\_/(\_;\_)")  linea31 =(" | . H | | : '='|")  linea6 =(" | | \_H\_\_/ \_| : |")  linea7 =(" \ '.\_\_ \ / ; ';")  linea19 =(" \_\_'-.\_(\_}==.' : ;")  linea71 =(" (\_\_\_| /-' | :. :")  linea371 =(" [.-' \ | \ \ ; :")  linea2 =(" .-' | | | | ':")  linea22 =(" / |==| \ \ / \\_")  linea661 =(" / [ | '.\_\\_ -.\_ \\")  linea517 =(" / \\_\_) \_\_.- \ \ )\\")  linea81 =("/ | /.' >>)")  linea61 =("| \ |\ |")  linea414 =("| .' '-. | \ /")  linea651 =("| / / / / /")  linea51 =(" | /")  linea41 =("")    print(linea1)  print(linea18)  print(linea11)  print(linea13)  print(linea15)  print(linea16)  print(linea21)  print(linea31)  print(linea6)  print(linea7)  print(linea19)  print(linea71)  print(linea371)  print(linea2)  print(linea22)  print(linea661)  print(linea517)  print(linea81)  print(linea61)  print(linea414)  print(linea651)  print(linea51)  print(linea41) |

|  |  |
| --- | --- |
| **Feedback** | - La reutilización de variables es muy utilizado en la programación cuando se necesita que el valor cambie bajo ciertas condiciones.  - Aún cuando las máquinas no se equivocan en los cálculos, la programación del algoritmo da cabida al defecto del programa generando resultados erroneos.    Secuencia correcta:  0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 |