Aprendizaje esperado:

AE1: Explicar conceptos básicos de la computación y la programación tales como variable, operador, condicional, ciclo para describir la solución computacional de un problema.

1 Problema a Resolver

Contexto: Fabricación de poleras Un plan de fabricación de poleras de una recocida empresa confección se puede representar matricialmente, las estimaciones de operaciones de las distintas fabricaciones están disponibles en función de los días planificados. Esta representación ayuda a determinar poleras con fabricación máxima, mínima, sumas totales de fabricaciones por poleras, por polera, por día, entre otros.
Para alcanzar tal propósito, escriba un programa en Python que permita al usuario generar un plan de fabricación de poleras para varios tipos de poleras durante un cierto número de días. El programa debe permitir al usuario ingresar el número de tipos de poleras y el número de días de fabricación, y luego mostrar un plan de fabricación en una matriz, donde cada fila representa un tipo de polera y cada columna representa un día de fabricación. Luego, el programa debe permitir al usuario ingresar la cantidad de poleras fabricadas (el valor en pesos de cada tipo polera debe ser ingresado) por tipo, en ningún caso la cantidad poleras a fabricar por tipo debe ser menor a 100 o mayor a 5500) y día, y mostrar el ingreso total de cada tipo de polera y la fabricación total de cada día.

El programa debe tener un menú que permita al usuario seleccionar una de las siguientes opciones:

- Generar plan de fabricación
- Ingresar cantidad y precio de fabricación por tipo de polera y día
- Visualizar el ingreso total de un tipo de polera específica
- Visualizar el ingreso total de todos los tipos de poleras en un día específico
- Salir del programa

Un ejemplo de interacción con el programa:

```
MENÚ

SISTEMA DE PLANIFICACIÓN DE FABRICACIÓN DE POLERAS

1. Generar plan de fabricación
2. Ingresar cantidad y precio de fabricación por tipo de polera y día
3. Visualizar el ingreso total de un tipo de polera específica
4. Visualizar el ingreso total de todos los tipos de poleras en un día específico
5. Salir del programa

Seleccione su opción: 1

Generar plan de fabricación

Ingrese número de tipos de poleras: 2
Ingrese número de días de fabricación: 3

Plan de fabricación
********************************
                 1   2   3
Tipo polera 1    0   0   0
Tipo polera 2    0   0   0
********************************
```

SISTEMA DE PLANIFICACIÓN DE FABRICACIÓN DE POLERAS
1. Generar plan de fabricación
2. Ingresar cantidad y precio de fabricación por tipo de polera y día
3. Visualizar el ingreso total de un tipo de polera específica
4. Visualizar el ingreso total de todos los tipos de poleras en un día específico
5. Salir del programa

Seleccione su opción: 2

Ingresar cantidad y precio de tipos de poleras y día

Tipo polera 1

Ingrese cantidad de poleras a fabricar en el día 1: 2555
Ingrese cantidad de poleras a fabricar en el día 2: 2900
Ingrese cantidad de poleras a fabricar en el día 3: 3990
Ingrese el precio de este tipo de polera: 5420

Tipo polera 2

Ingrese cantidad de poleras a fabricar en el día 1: 2695
Ingrese cantidad de poleras a fabricar en el día 2: 6400
Error, la capacidad máxima por poleras a fabricar es de 5500
Ingrese cantidad de poleras a fabricar en el día 2: 2520
Ingrese cantidad de poleras a fabricar en el día 3: 4985
Ingrese el precio de este tipo de polera: 6568

Plan de fabricación
****

	1	2	3
Tipo polera 1	2555	2900	3990
Tipo polera 2	2695	2520	4985

****

SISTEMA DE PLANIFICACIÓN DE FABRICACIÓN DE POLERAS
1. Generar plan de fabricación
2. Ingresar cantidad y precio de fabricación por tipo de polera y día
3. Visualizar el ingreso total de un tipo de polera específica
4. Visualizar el ingreso total de todos los tipos de poleras en un día específico
5. Salir del programa

Seleccione su opción: 3

Visualizar el ingreso total de un tipo de polera específica

Ingrese el tipo de polera a calcular y visualizar: 1

Tipo polera 1, día 1: 2555 fabricadas para un total de ingresos $: 14103600
Tipo polera 1, día 2: 2900 fabricadas para un total de ingresos $: 16008000
Tipo polera 1, día 3: 3990 fabricadas para un total de ingresos $: 22024800
Total ingresos del tipo de polera 1 $: 52136400

```
SISTEMA DE PLANIFICACIÓN DE FABRICACIÓN DE POLERAS
1. Generar plan de fabricación
2. Ingresar cantidad y precio de fabricación por tipo de polera y día
3. Visualizar el ingreso total de un tipo de polera específica
4. Visualizar el ingreso total de todos los tipos de poleras en un día específico
5. Salir del programa

Seleccione su opción: 4

Visualizar el ingreso total de todos los tipos de poleras en un día específico

Ingrese el día a calcular y visualizar: 2

Tipo polera 1, día 2: 2900 fabricadas para un total de ingresos $:  15718000
Tipo polera 2, día 2: 2520 fabricadas para un total de ingresos $:  16551360
Total ingresos del día 2 $: 32269360

SISTEMA DE PLANIFICACIÓN DE FABRICACIÓN DE POLERAS
1. Generar plan de fabricación
2. Ingresar cantidad y precio de fabricación por tipo de polera y día
3. Visualizar el ingreso total de un tipo de polera específica
4. Visualizar el ingreso total de todos los tipos de poleras en un día específico
5. Salir del programa

Seleccione su opción: 5

Fin de la ejecución del programa
```

2 Aspectos administrativos

- Modalidad: grupos de 3 estudiantes máximo.
- Lugar: Canvas Fecha de Entrega: Domingo 18 de mayo de 2025
- Entregable: archivo .py con todo el código respectivo. El archivo debe ser nombrado: RUT1_RUT2_RUT3_NRC9999.py
- RUT, sin el digito verificador. En el cuerpo del programa en comentarios deben ir el nombre completo y rut de cada integrante del grupo.
- Presentación de los conocimientos en clases Semana 11 y 12

3 Rúbrica

Criterios de la entrega de la solución (.py)	Excelente (10 puntos)	Bueno (7 puntos)	Regular (4 puntos)	Deficiente (1 punto)	
Puntualidad	Entrega a tiempo	Entrega con 1-2 días de atraso	-	-	
Diseño del Algoritmo	El algoritmo está claramente definido, es eficiente y sigue una lógica óptima para resolver el problema. Se consideran todos los casos posibles y se anticipan errores.	El algoritmo es funcional y resuelve el problema, pero podría ser más eficiente o claro en su lógica. Se consideran la mayoría de los casos posibles.	El algoritmo tiene deficiencias en su lógica o no cubre todos los casos posibles. La eficiencia es baja y la claridad es limitada.	El algoritmo es confuso, incompleto o no resuelve el problema planteado. No se consideran los casos posibles.	
Diseño de la solución de los requisitos	La solución implementa todos los requisitos del problema de forma creativa y eficiente. Se demuestra un profundo entendimiento del contexto y la problemática.	La solución implementa la mayoría de los requisitos del problema de forma funcional. Se demuestra un buen entendimiento del contexto.	La solución implementa solo algunos de los requisitos del problema o lo hace de forma incompleta. El entendimiento del contexto es limitado.	La solución no aborda el problema planteado o lo hace de forma incorrecta. No se demuestra entendimiento del contexto.	
Correcto Uso de Estructuras de Datos en Python	El código utiliza de manera eficiente y apropiada las estructuras de datos de Python (listas, strings, sentencias de control, etc.). Se justifica la elección de cada estructura y se optimiza su uso.	El código utiliza las estructuras de datos de Python de manera funcional, pero podría ser más eficiente o justificar mejor su elección.	El código utiliza las estructuras de datos de Python de manera incorrecta o ineficiente. No se justifica su elección o se utilizan estructuras inadecuadas para el problema.	El código no utiliza o utiliza de manera muy deficiente las estructuras de datos de Python.	
Correcto uso de la sintaxis	El código utiliza correctamente la sintaxis de Python, incluyendo estructuras de control, funciones, listas y manejo de strings. No hay errores de sintaxis.	El código utiliza la sintaxis de Python de forma mayormente correcta, pero puede haber errores menores o inconsistencias.	El código presenta errores de sintaxis frecuentes que dificultan su ejecución. El uso de las estructuras de Python es limitado o incorrecto.	El código tiene numerosos errores de sintaxis que impiden su ejecución. El uso de Python es muy deficiente.	
Correcta Ejecución	El programa se ejecuta sin errores y produce los resultados esperados en todos los casos de prueba. La salida es clara y fácil de entender.	El programa se ejecuta con algunos errores menores o produce resultados incorrectos en casos específicos. La salida e	El programa presenta errores que impiden su correcta ejecución o produce resultados incorrectos en la mayoría de los casos. La salida es confusa o incompleta.	El programa no se ejecuta o produce resultados completamente erróneos. La salida es inexistente o incomprensible.	
Documentación de la solución	Correcta aplicación de los comentarios para documentación del código presentado	Aplicación de los comentarios en mediana escala para documentación del código presentado	Aplicación de los comentarios regularmente para documentación del código presentado	Sin documentación el código presentado	
Criterios de la Presentación					
Presentación	Presenta a tiempo	Presenta con 1-2 días de	-	-	

		atraso			
Explicación Técnica	Explica correctamente la sintaxis de Python, incluyendo estructuras de control, funciones, listas y manejo de strings.	Explica medianamente la sintaxis de Python, incluyendo estructuras de control, funciones, listas y manejo de strings.	Explica regularmente la sintaxis de Python, incluyendo estructuras de control, funciones, listas y manejo de strings.	No explica la sintaxis de Python, incluyendo estructuras de control, funciones, listas y manejo de strings.	
Respuesta a preguntas	Responde correctamente las preguntas realizadas	Responde medianamente las preguntas realizadas	Responde regularmente las preguntas realizadas	No Responde las preguntas realizadas	