

Criterios de evaluación

1. Conocimiento y entendimiento de los conceptos.

- Resolución de problemas: Evaluación de la capacidad del alumno para utilizar los conceptos y las estructuras de Python para resolver problemas específicos. Esto incluye la implementación de algoritmos, la manipulación de datos, el uso de estructuras de datos, la implementación de funciones, entre otros.

2. Sintaxis utilizada de manera correcta.

3. Indentación consistente.

- Indentar el código de manera de respetar los bloques de código.

4. Nombre descriptivo de variables, funciones y clases.

- Utilizar nombres que faciliten la comprensión del código.

5. Uso de comentarios.

- Incluir comentarios que expliquen las funcionalidades del código.

Ejercicio N° 1 (1p):

Desarrollar un programa para gestionar una lista de tareas pendientes. Implementar una función que tome dos listas de tareas y las combine en una sola lista,

Implementar una función llamada `combinar_tareas()` que reciba dos listas de tareas como parámetros.

Retornar la lista combinada resultante.

Ejercicio N° 2 (1.5p):

Programa para gestionar una lista de tareas pendientes. Implementar un programa que utilice diccionarios para almacenar y manipular la información de las tareas.

La solución debe cumplir los siguientes requisitos:

Crea un diccionario que almacenará la información de las tareas pendientes. Cada clave del diccionario representará el nombre de una tarea y cada valor será un booleano que indicará si la tarea está completada o no.

- Implementar una función que reciba como parámetro el nombre de una tarea y la agregue al diccionario tareas con un valor inicial de `False` para indicar que la tarea está pendiente.
- Implementar una función que reciba como parámetro el nombre de una tarea y actualice el valor correspondiente en el diccionario tareas a `True` para indicar que la tarea está completada.

- Implementa una función que reciba como parámetro el nombre de una tarea y elimine dicha tarea del diccionario tareas.
- Verificar el correcto funcionamiento, agregando, actualizando y eliminando al menos una tarea.

Ejercicio N° 3 (2p):

Imagina que estás desarrollando un programa para una máquina expendedora de bebidas. Implementar un programa que solicite al usuario seleccionar una opción y muestre el precio correspondiente de la bebida seleccionada.

Implementa una solución que cumpla con los siguientes requisitos:

Muestra al usuario un menú con las siguientes opciones de bebidas:

- Opción 1: Coca-Cola - Precio: \$2.50
- Opción 2: Sprite - Precio: \$2.00
- Opción 3: Fanta - Precio: \$1.75 Solicita al usuario que seleccione una opción ingresando el número correspondiente.

Verificar la opción seleccionada por el usuario y mostrar por pantalla el precio correspondiente.

Si el usuario selecciona una opción inválida, mostrar un mensaje de error.

Ejercicio N° 4 (4p):

Imagina que estás creando un programa para una biblioteca que gestiona la información de los libros que tiene disponibles. Tu tarea es implementar un sistema de gestión de libros que permita a los usuarios realizar consultas y obtener información sobre los libros.

Utilizando los conceptos de Programación Orientada a Objetos, implementa una solución que cumpla con los siguientes requisitos:

1. Crea una clase llamada Libro que tenga los siguientes atributos:
 - titulo: una cadena que representa el título del libro.
 - autor: una cadena que indica el autor del libro.
 - precio: un número decimal que indica el precio del libro.
 - disponible: un valor booleano que indica si el libro está disponible para préstamo.
2. La clase Libro debe tener los siguientes métodos:
 - informacion(): muestra por pantalla la información del libro, incluyendo el título, autor y precio.
 - prestar(): cambia el estado del libro a no disponible.
 - devolver(): cambia el estado del libro a disponible.
3. Crea una clase derivada llamada LibroFisico que herede de la clase Libro. La clase LibroFisico tiene un atributo adicional:
 - ubicacion: una cadena que indica la ubicación física del libro en la biblioteca.

4. Sobrescribe el método `informacion()` en la clase `LibroFisico` para incluir la ubicación física del libro al mostrar la información.