

**Nombre:**

Daniel Eliseo Martínez Pérez (SMSS058823)

Roberto Carlos Orellana Sorto (SMSS079623)

**Docente:**

Ing. William Montes

**Materia:**

Programación III

**Actividad:**

Parcial 1 C1

**Facultad:**

Ciencias y tecnología

**Año lectivo:**

2024

EJERCICIO 1  
Gestión de Fechas:

datetime.date.today() y datetime.timedelta(): Se utilizan para manejar fechas sin complicaciones adicionales. datetime.date.today() obtiene la fecha actual, y datetime.timedelta(days=...) suma días a la fecha actual para asignar fechas de consulta futuras de manera sencilla.

Generación de Fechas Aleatorias:

random.randint(): Asigna fechas de consulta aleatorias dentro de un rango (1 a 10 días) para simular la variabilidad en la programación de citas.

Almacenamiento en Diccionario:

self.pacientes\_registrados = {}: Un diccionario almacena los pacientes por nombre, permitiendo un acceso rápido y directo para verificar si un paciente ya está registrado.

Lista para Sala de Espera:

self.sala\_espera = []: Una lista mantiene los nombres de pacientes en la sala de espera, lo que facilita agregar y mostrar pacientes que deben esperar.

Métodos para Asignación y Verificación de Fechas:

asignar\_fecha\_consulta() y verificar\_fecha\_consulta(): Estos métodos asignan fechas de consulta y verifican si han pasado, asegurando que los pacientes sean gestionados adecuadamente, incluyendo el envío a la sala de espera si es necesario.

Mostrar Información:

mostrar\_pacientes(): Imprime la información de todos los pacientes y los que están en la sala de espera, proporcionando una visión clara del estado del consultorio.

Interfaz de Usuario en Consultorio:

La clase Consultorio gestiona la interacción con el usuario, pidiendo datos y coordinando el flujo del registro y la gestión de pacientes. Esto mantiene el código modular y organizado.

Razón de Elección: Estos métodos y estructuras fueron elegidos por su simplicidad y efectividad para manejar las tareas requeridas. Usar datetime y random simplifica la manipulación de fechas y la asignación de citas. Los diccionarios y listas permiten un almacenamiento y acceso eficiente de datos. La modularidad en la clase Consultorio organiza el código, haciendo que sea más fácil de entender y mantener.

EJERCICIO 2. Clases:

Book: Representa un libro con título y autor.

Member: Representa a un miembro con nombre.

Loan: Maneja el préstamo de un libro a un miembro, incluyendo la fecha del préstamo y la fecha límite de devolución. También gestiona los retrasos en la devolución.

Métodos:

Loan.process\_return(): Marca el libro como devuelto y calcula si hay retraso, mostrando un mensaje apropiado.

Library.issue\_loan(): Crea un préstamo nuevo y lo almacena en una lista, usando dateparser para interpretar la fecha ingresada.

Library.return\_loan(): Procesa la devolución de un libro buscando el préstamo activo correspondiente.

Flujo Principal:

Se solicitan datos de libros y miembros.

Se ofrece un menú para realizar préstamos, devolver libros o salir del sistema.

Elección de Métodos:

dateparser simplifica el manejo de formatos de fecha diversos.

Listas y objetos permiten una gestión clara de libros, miembros y préstamos.

EJERCICIO 3. Yo creé el código de la tienda usando clases para organizar las funciones de manera clara. La clase Producto representa los productos en la tienda, con atributos para el nombre, precio y cantidad. La clase Tienda gestiona estos productos, permitiendo agregar nuevos productos, realizar ventas y recibir productos de proveedores. Utilicé métodos como vender\_producto para calcular el total de la venta y el vuelto, y recibir\_proveedor para actualizar el inventario con nuevos productos. Esta estructura facilita la gestión del inventario y las transacciones de manera eficiente.

EJERCICIO 4. Yo diseñé el código del zoológico usando clases para manejar la información de los animales de forma ordenada. La clase Animal incluye detalles como nombre, especie, área y estado de salud, mientras que la clase Zoologico gestiona una lista de animales, permitiendo agregar nuevos y listar aquellos en tratamiento. Utilicé métodos como agregar\_animal para actualizar o añadir animales y listar\_animales\_en\_tratamiento para filtrar y mostrar solo los que están en tratamiento. Esta organización ayuda a mantener un registro claro y actualizado de la salud de los animales.