**Logotipo

Descripción generada automáticamente**

**UNIVERSIDAD GERARDO BARRIOS, SAN MIGUEL**

**FACULTAD DE CIENCIA Y TECNOLOGÍA**

**MATERIA: PROGRAMACIÓN III**

**DOCENTE: WILLIAN MONTES**

**ALUMNOS:**

**XAVIER ALEXANDER PÉREZ BONILLA**

**BLANCA LETICIA ARGUETA PORTILLO**

**ACTIVIDAD: PARCIAL 1**

**Justificación**

La solución se abordó utilizando POO y un diccionario para almacenar los pacientes porque estas herramientas ofrecen una forma clara y eficiente de modelar el problema. La POO facilita la encapsulación de los datos y la funcionalidad relacionada con el consultorio, lo que hace que el código sea más fácil de entender, mantener y escalar.

También tenemos estructuras de control para un mejor control de los datos

**Planteamiento del ejercicio 1:**

El ejercicio consiste en simular el proceso de registro y asignación de citas en un consultorio médico, donde una secretaria realiza las siguientes tareas:

1. Registrar Pacientes Nuevos: Solicitar los datos de los pacientes, incluyendo el nombre y el motivo de la consulta, y asignarles una fecha para su cita.
2. Manejo de Pacientes ya Registrados: Si un paciente ya tiene una consulta programada, se le informa de la fecha de su cita y se le envía a la sala de espera en lugar de tomar sus datos de nuevo.
3. Mostrar Pacientes Registrados: Permitir que se vea una lista de todos los pacientes registrados, junto con sus motivos de consulta y fechas de cita.

**Solución:**

La POO permite modelar el consultorio como una clase ConsultorioMedico, que contiene métodos para registrar pacientes, mostrar la lista de pacientes y manejar las citas.

**Estructura de Datos:**

* Diccionario para Pacientes: Se utiliza un diccionario para almacenar la información de los pacientes, donde la clave es el nombre del paciente y el valor es otro diccionario con detalles como el motivo de consulta y la fecha de cita.
* Motivo por lo que también pusimos un diccionario: El diccionario facilita la verificación rápida de si un paciente ya tiene una cita asignada y permite acceder fácilmente a sus datos.

**Planteamiento ejercicio 2:**

El ejercicio nos pide hacer préstamos de libros a sus usuarios mediante el uso de tarjetas impresas. Cada tarjeta contiene los datos del usuario y la información de los libros que ha retirado. La biblioteca desea implementar un sistema más eficiente para gestionar los préstamos y las devoluciones de los libros.

**Objetivos del ejercicio:**

* Automatizar el proceso de préstamo y devolución de libros para mejorar la eficiencia de la biblioteca.
* Reducir los errores manuales en el registro de préstamos y devoluciones.
* Asegurar que los usuarios cumplan con las fechas de devolución de los libros y gestionar las sanciones adecuadamente

**Solucion:**

* **Definición de Clases Principales**:
* **Clase Libro**: Representa un libro en la biblioteca. Cada libro tiene un título y un estado que indica si está en préstamo o no.
* **Clase Persona**: Representa a una persona que puede tomar libros prestados. Cada persona tiene un nombre y una lista de libros que ha prestado.
* **Clase SistemaBiblioteca**: Gestiona la lógica de los préstamos y devoluciones de libros. Esta clase maneja el registro de préstamos, la devolución de libros y la aplicación de sanciones por devoluciones tardías.
* **Funcionalidades Clave Implementadas**:
* **Prestar un libro**: Se verifica si el libro está disponible y, si es así, se registra el préstamo, incluyendo la fecha actual y la fecha límite de devolución.
* **Devolver un libro**: Se permite la devolución de un libro, verificando si la devolución es puntual o tardía y aplicando sanciones en consecuencia.

**Planteamiento del ejercicio 3:**

El ejercicio consiste en crear un programa para un hotel de playa donde un recepcionista realiza estas tareas:

1. Mostrar habitaciones disponibles y sus precios a los clientes.
2. Registrar los datos del cliente, incluyendo la habitación elegida y el número de noches que se quedará.
3. Ofrecer servicios extra como piscina o cancha de golf, que tienen un costo adicional.
4. Generar una factura que incluya todos los costos.

**Solución:**

* Usamos Programación Orientada a Objetos para organizar el código, creando una clase Hotel que maneja todas las funciones del hotel
* Utilizamos diccionarios para almacenar los tipos de habitaciones y servicios extra, facilitando el acceso a sus precios
* Proporcionamos una interfaz interactiva con el recepcionista para ingresar los datos del cliente y seleccionar opciones

**Planteamiento ejercicio 4:**

Una empresa de renta de transporte tiene varios tipos de vehículos disponibles para alquilar, cada uno con sus características y coste de renta. La empresa necesita un sistema para registrar los vehículos que ingresan al lote y gestionar su alquiler, tomando en cuenta los datos del cliente.

**Solucion:**

**Registro de Vehículos**: Se utiliza el método registrar\_vehiculo para agregar un nuevo vehículo al inventario de la empresa. Cada vehículo tiene un tipo, modelo y coste de renta diario.

**Renta de Vehículos**: El método rentar\_vehiculo permite a un cliente rentar un vehículo específico, siempre y cuando esté disponible. Se registran los detalles de la renta, incluyendo las fechas de inicio y fin del alquiler.

**Devolución de Vehículos**: Con el método devolver\_vehiculo, un cliente puede devolver un vehículo rentado. Al hacerlo, el estado del vehículo se actualiza a disponible y se elimina el registro de renta correspondiente.

**Manejo de Inventario**: El método mostrar\_vehiculos\_disponibles imprime una lista de todos los vehículos que están disponibles para renta.