



**UNIVERSIDAD GERARDO BARRIOS SAN MIGUEL**

**FACULTAD DE CIENCIA Y TECNOLOGÍA**

**ASIGNATURA:**

Programación Computacional 3.

**DOCENTE:**

William Alexis Montes Girón

**Actividad:**

Parcial 1.

**Estudiantes:**

Mario Antonio Barahona Sorto (SMSS032623)

José Isaac Medrano Ventura (SMSS094123)

## **Explicación del ejercicio numero 1 La tienda de Niña Mary:**

**-Clase Producto:** Representa un producto en la tienda con su nombre, precio de costo, cantidad disponible, y precio de venta.

**-Clase Tienda:** Maneja el inventario de productos, las ventas, y la interacción con los proveedores.

Si cumple los métodos de POO:

- **Método agregar\_producto:** Añade un producto al inventario.
- **Método realizar\_venta:** Procesa una venta, calcula el total y el vuelto, y actualiza el inventario.
- **Método recibir\_proveedor:** Recibe un producto del proveedor, lo agrega o actualiza en el inventario.
- **Método mostrar\_inventario:** Muestra los productos disponibles en la tienda.

El código proporciona una base para manejar las operaciones diarias de la tienda de "niña Mary".

## **-Explicación de ejercicio numero 2 centro educativo:**

Este código gestiona la asistencia de estudiantes en un colegio. La clase RegistroAsistencia guarda la fecha, el estado de asistencia (Asistió, Permiso, Inasistencia) y la razón para permisos. La clase Estudiante mantiene un registro de asistencia de cada estudiante. La clase Docente maneja una lista de estudiantes y registra su asistencia. Finalmente, la clase director coordina con varios docentes y muestra la asistencia de todos los estudiantes. Este sistema ayuda a digitalizar el registro de asistencia y gestionar permisos de manera eficiente.

### **-Explicación del ejercicio numero 3:**

Motivo por el que se Abordó la Solución

La solución se abordó con el objetivo de simplificar y automatizar las tareas cotidianas del recepcionista en el hotel. El recepcionista tiene que gestionar múltiples funciones, como mostrar opciones de habitaciones, registrar información de los clientes, calcular el costo total de la estancia, y generar facturas detalladas. Realizar estas tareas manualmente puede ser propenso a errores y tomar tiempo valioso.

### **-Explicación del ejercicio numero 4:**

Motivo por el que se abordó la solución:

Este enfoque utiliza la herencia en la programación orientada a objetos, lo que permite manejar diferentes tipos de empleados con distintas estructuras salariales dentro de una misma clase base. El uso de herencia asegura que el código sea modular, fácil de entender, y extensible si se desea añadir más tipos de empleados en el futuro. Además, centraliza la lógica del cálculo del bono, evitando la duplicación de código.