

UNIVERSIDAD GERARDO BARRIOS
FACULTAD DE CIENCIA Y TECNOLOGIA



Parcial 1

PRESENTADO POR:

JENNIFER TATIANA RIOS VILLATORO - SMSS179923

ALAN ALEXANDER PEÑATE CHICAS - SMSS187322

ASIGNATURA:

Programacion Computacional III

DOCENTE:

William Alexin Montes Giron

FECHA DE PRESENTACIÓN:

30/08/2024

I. Sistema de Registro de Asistencia en una Escuela

Planteamiento:

Una escuela necesita digitalizar el proceso de registro de asistencia de los estudiantes. Actualmente, los docentes registran la asistencia en papel, lo que se recopila al final del día. El director también necesita conocer la razón de las ausencias cuando los estudiantes tienen un permiso.

Solución Propuesta:

La solución consiste en crear una aplicación sencilla en Python que permita a los docentes registrar la asistencia directamente desde un dispositivo electrónico. Este registro incluirá:

El nombre del estudiante.

La fecha de asistencia.

El estado de asistencia (asistió, inasistió, permiso).

Una razón para las ausencias con permiso.

La información se almacena en un archivo CSV para su fácil acceso y consulta por el director.

Justificación:

Elegí esta solución porque:

Simplicidad: Es fácil de implementar con herramientas básicas de Python, sin necesidad de una infraestructura compleja.

Acceso directo: Facilita al director obtener información en tiempo real sin esperar a que los docentes entreguen las hojas de papel.

Escalabilidad: Se puede expandir en el futuro, por ejemplo, integrándolo con una base de datos o creando una interfaz gráfica.

II. Sistema de Reservas y Facturación en un Hotel

Planteamiento:

Un hotel de playa necesita un sistema para gestionar las reservas de habitaciones y servicios adicionales, como el uso de la piscina o la cancha de golf. El recepcionista debe registrar los datos personales del cliente, el tipo de habitación, la cantidad de noches y cualquier servicio adicional solicitado, y luego generar una factura detallada.

Solución Propuesta:

La solución consiste en un programa Python que:

Muestra las opciones de habitaciones y servicios disponibles.

Registra los datos del cliente y su elección de habitación.

Permite al cliente agregar servicios adicionales.

Calcula el costo total de la estancia y genera una factura detallada.

Justificación:

Elegí esta solución porque:

Organización y eficiencia: Permite al recepcionista manejar múltiples tareas (reservas, servicios, facturación) en un solo lugar, evitando errores manuales.

Flexibilidad: Los clientes pueden elegir diferentes servicios adicionales, lo que es manejado dinámicamente por el programa.

Fácil implementación: Es un sistema que puede ser desarrollado rápidamente y, si es necesario, ampliado para manejar mayores volúmenes de datos o integrarse con otros sistemas.

III. Sistema de Planilla de Pago para Empleados

Planteamiento:

Una empresa necesita generar la planilla de pago para dos tipos de empleados: los de plaza fija, que reciben un salario base más comisiones, y los empleados por horas, que son pagados según las horas trabajadas. Además, si un empleado ha trabajado más de 5 años, se le otorga un bono adicional, sin importar su tipo de contrato.

Solución Propuesta:

La solución incluye la creación de un programa Python con:

Clases que representen a los empleados de plaza fija y por horas.

Métodos para calcular el salario, incluyendo comisiones y horas trabajadas.

Un método adicional para calcular un bono basado en la antigüedad del empleado.

Una función para generar la planilla de pago mostrando el salario total de cada empleado.

Justificación:

Elegí esta solución porque:

Modularidad: Usar clases para cada tipo de empleado permite una clara separación de responsabilidades y hace que el código sea más mantenible.

Flexibilidad: La solución se adapta fácilmente a nuevos requisitos, como la inclusión de otros tipos de empleados o beneficios adicionales.

Escalabilidad: Si la empresa crece, este sistema puede integrarse con bases de datos o sistemas de nómina más avanzados sin requerir una reescritura significativa del código.

IV. Sistema de Registro de Ventas para una Tienda

Planteamiento:

Una tienda necesita un sistema para gestionar el registro de ventas. Los empleados deben poder registrar cada venta con detalles del producto, la cantidad vendida, el precio unitario y calcular el total de la venta. Además, se requiere un reporte diario que incluya todas las ventas realizadas.

Solución Propuesta:

La solución consiste en desarrollar un programa en Python que:

Permita a los empleados ingresar los detalles de cada venta.

Calcule automáticamente el total de cada venta.

Almacene todas las ventas del día en un archivo CSV o base de datos.

Genere un reporte diario que pueda ser revisado por el administrador de la tienda.

Justificación:

Elegí esta solución porque:

Automatización del proceso: El cálculo automático de las ventas reduce errores manuales y agiliza el proceso de registro.

Organización de datos: Almacenar las ventas en un archivo o base de datos facilita la gestión y revisión de la información diaria.

Simplicidad en la implementación: Es un sistema básico que no requiere una infraestructura compleja, pero que resuelve eficazmente las necesidades de la tienda.

Escalabilidad: Este sistema puede evolucionar fácilmente para incluir funciones adicionales, como inventarios, integración con sistemas de facturación, o generación de reportes más detallados.