

Caso de Estudio #1: Gestión de Empleados

Introducción

Usted trabaja en una pequeña empresa y se le ha asignado la tarea de crear un programa simple en Python para gestionar la información de los empleados. La empresa necesita un sistema para realizar cálculos de salarios y evaluar el desempeño de los empleados.

Objetivos

Los objetivos consisten en evaluar los temas de la unidad 1, 2 y 3. Así como:

1. Practicar conceptos fundamentales de programación en Python.
2. Desarrollar habilidades en la gestión de datos y algoritmos.
3. Resolver un problema del mundo real.
4. Fomentar la repetición y la iteración.
5. Promover el pensamiento lógico y la resolución de problemas.

Alcance

El alcance o requerimientos esperados por la solución son:

1. El programa debe consultarle al usuario si desea ingresar los datos de un empleado, cada empleado debe tener:
 - Nombre del empleado.
 - Número de horas trabajadas en un mes.
 - Tarifa por hora.
 - Evaluación de desempeño (puede ser "Excelente", "Bueno" o "Regular").
2. El programa debe calcular el salario del empleado registrado utilizando la fórmula:
 - $\text{Salario} = \text{Número de horas trabajadas} * \text{Tarifa por hora}$
3. El programa debe evaluar el desempeño del empleado registrado y asignar un bono adicional al salario según la siguiente tabla:
 - "Excelente": 10% de bono.
 - "Bueno": 5% de bono.
 - "Regular": Sin bono adicional.
4. El programa debe mostrar en pantalla la información del empleado registrado, incluyendo su salario total con el bono.
5. El programa debe permitir al usuario repetir las operaciones hasta que decida salir. (Utilice un menú para proporcionar al usuario las opciones anteriores)
6. **Por considerar:** Si el usuario no ha ingresado los datos de un empleado, no puede realizar los puntos 2, 3, y 4.

Nota:

- La solución **solamente puede implicar elementos vistos en el curso.**

Entregables

- Se debe realizar un análisis del problema, indicando los pasos necesarios para resolver el problema (algoritmo).
- Se debe diseñar la posible solución indicando las variables y estructuras que se utilizarán en la solución del problema y para qué sirve cada una de ellas.
 - Es decir, debe existir un comentario por bloque de código, puede ser una oración que explique rápidamente qué objetivo realiza el bloque de código.
- Se debe desarrollar en el programa indicado por el profesor la solución programada (.py) y anexarla a la entrega.
- Se realizan las pruebas de la solución desarrollada (screenshot del funcionamiento del programa ante cada escenario).

Evaluación

La calificación del caso será evaluada con 100pts los cuales tienen la siguiente distribución:

- **20pts:** Algoritmo de la solución (se entrega en un bloque de comentarios al inicio del programa),
 - ej:
''' Los pasos del algoritmo son:

1. Paso uno
2. Paso Dos
3. Etc...

'''
- **40pts:** Solución .py
 - **25pts** - Funcionamiento esperado.
 - **10pts** - Ejecución sin errores.
 - **5 pts** - Comentarios por bloque.
- **40pts:** Screenshot (pantallazo .png)
 - **10pts** - Formato .png
 - **30pts** - Contempla los escenarios requeridos (aún si estos no funcionan).