Fundamentos de informática

Proyecto #2 Software para la administración de recursos humanos

Estudiantes: Ian Villegas Jiménez ID: 207800171

Olman Ugarte Vega ID: 305320681

**1.Descripción del problema**

Se solicita la creación de un programa en C++, usando programación orientada a objetos, el cual es un software para administrar el recurso humano. Este sistema permite llenar el vector con los datos de los empleados. Si no se registran los 50 empleados se debe controlar el límite del tamaño del vector, para todas las operaciones que se hacen con él. Si se borra un empleado se debe eliminar del vector. Si el usuario está en el medio, los otros elementos se deben desplazar y no dejar “huecos” en el vector.

**2.Descripción de los aspectos más importantes a utilizar**

**Sistema de datos de RRHH SIUA**

**Mantenimiento de empleados.**

**0 - Lista de empleados.**

**1 - Registrar empleado.**

**2 - Eliminar por ID.**

**3 - Modificacion de empleado por ID.**

**Mantenimiento de salarios.**

**4 - Aumentar salario de todos los empleados segun porcentaje de aumento.**

**5 - Aumentar salario de un empleado por ID.**

**Reportes**

**6 - Lista de los mejores 5 salarios.**

**7 - Lista de los empleados por orden segun su salario.**

**8 - Lista de los empleados por orden segun su ID.**

**9 - Buscar informacion de un empleado por ID.**

**10 - Buscar informacion de un empleado por nombre.**

**11 - Promedio de los salarios.**

**12 - Salarios con los totales de salario, deducciones y ahorros.**

**13 - Salir.**

**La opción número 0, está relacionado con listaEmpleados0()**

Se crearonobjetos para ser mostrados en el **toString** de **ColeccionEmpleados** y a su vez se guardan los objetos para ser mostrados en el **toString** de **ColeccionEmpleados.**

**baseDatosEmpleados(ColeccionEmpleados lista){**

Muestra la lista de objetos predeterminados

**La opción 1, está relacionada con registrarEmpleado1()**

Este método solicita al usuario información para registrar un nuevo empleado

**La opción 1, está relacionada con vectorEmpleados[cantidad++] = lista;**

Añade el nuevo empleado al vector enlazándolo a lista que es de la opción 0

**La opción 2, está relacionada con eliminarPorID2()**

Este método elimina un empleado al darle un identificador particular

**La opción 3, está relacionada con modificacionEmpleadosID3()**

Este método permite al usuario modificar un empleado por su número de identificador

**La opción 4, está relacionada con aumentarSalarioTodosPorcentaje4()**

El método aumenta el salario general de los empleados indicando un porcentaje

**La opción 5, está relacionada con aumentarSalarioUnoPorID5()**

Este método aumenta el salario de un empleado particular elegido por el usuario

**La opción 6, está relacionada con listaMejoresSalarios6()**

Este método muestra una lista con los mejores 5 salarios

**La opción 7, está relacionada con listaEmpleadosPorOrdenSalario7()**

Este método muestra una lista ordenada de los salarios de menor a mayor

**La opción 8, está relacionada con listaEmpleadosPorOrdenID8()**

Este método muestra una lista ordenada de los empleados de menor a mayor por ID

**La opción 9, está relacionada con buscarInfoEmpleadoPorID9()**

El método permite al usuario buscar información de un empleado particular por ID

**La opción 10, está relacionada con buscarInfoEmpleadoPorNombre10()**

El método permite al usuario buscar información de un empleado particular por nombre

**La opción 11, está relacionada con calcularPromedioSalarios11()**

Este método calcula el promedio de salarios total

**La opción 12, está relacionada con calcularTodosSalarios12()**

Este método calcula el total de salarios, deducciones y de ahorros

**toString()**

El método itera ciertos datos del empleado, sin esto no imprimiría todos los métodos anteriores que hacen uso del vector

**int main ()**

En la función main se encuetra un ciclo do while que contiene el menú del programa; y un switch en el que cada case representa una opción del menú

**3. Descripción de datos de prueba:**

Se le pide al usuario digitar un número correspondiente con las opciones; en las opciones 1,10 se tienen que usar letras y en las opciones 2,3,4,5,9 solo se tiene que usar números.

**4.Limitaciones del programa:**

Si se digita una letra en las opciones 2,3,4,5,9 el programa no va a funcionar debido a que en estas opciones solo se pueden usar números, y en las opciones 1,10 si se digita un número el programa no va a funcionar porque en dichas opciones solo se puede agregar letras.

**5.Observaciones generales:**

Se usaron colores distintos en ciertas partes del menú para hacerlo más atractivo y entendible al usuario.

**6. Manual de usuario:**

El programa es para llevar un control de recurso humano, el cual posee un menú listado de 13 opciones, cada una de ellas con su respectiva uso; el usuario elige la opción que necesita llevar a cabo y digita el número correspondiente.