

PROYECTO FINAL.

Integrantes: Gómez Trejo Gustavo Ali

Zagoya Mellado Roberto Uriel

Romero Lares Luis Daniel

Vázquez Torres Juan Adrián

Profesor: Carlos Alberto Román Zamitiz

Grupo: 01

Materia: Programación Orientada a Objetos

Semestre: 2017 – 2



ENUNCIADO DEL PROYECTO FINAL

“Somos la empresa ManPower S.A. de C.V. que se encarga de llevar consultores outsourcing a otras empresas. Queremos un sistema que nos ayude a gestionar nuestros empleados porque actualmente la chica de Recursos Humanos tiene un relajo, a algunos los tiene registrados en Excel

a otros en papel, pero llega Doña Chonita la señora de la limpieza y le revuelve todos los papeles de

su escritorio.

Cada empleado tiene un identificador o clave de empleado, un departamento al que pertenece y

obviamente su sueldo quincenal. Cada departamento tiene un gerente, el cual es un empleado, pero

con la diferencia de que el gerente tiene además un bono (\$), el cual es variable entre el gerente de

un departamento y el gerente de otro departamento.

Como parte de la gestión de empleados que debe hacer el sistema se encuentra la búsqueda de

empleados por clave, por nombre, por departamento y obviamente todas las operaciones CRUD

que todo sistema debe hacer.

Creo que no estoy olvidando nada, pero si recuerdo algo más, se los hago llegar por correo. Me

tengo que ir porque tengo una reunión con el Director General.”

CODIGO.

CLASE SISTEMA.

```
import java.util.Scanner;
import java.util.*;
import java.io.*;
import java.lang.IllegalStateException;

public class Sistema
{
    public static void main(String[]args)
    {
        Insertar insertar = new Insertar(); //Se crea una referencia a tipo de objeto necesitado
        Buscar buscar = new Buscar();      //Se crea una referencia a tipo de objeto necesitado
        Eliminar eliminar = new Eliminar(); //Se crea una referencia a tipo de objeto necesitado
        String cla="";
        Scanner opcion = new Scanner(System.in); //Se crea una referencia a tipo de objeto
        necesitado
        Scanner entrada; //Se lee la opcion con un Scanner
        do{ //Usamos un do while para iterar el menú que se propuso para relizar el proyecto
        try{ //Se utilizo el try - catch para poder prevenir error al momento de ingresar la opcion
        menu(); cla=opcion.next();
        }catch(Exception ex){
        System.out.println("error");
        }
        switch(cla){ // aqui se compara el valor que el usuario escogio y se procede a entrar y
        realizar lo que se necesita
        case "1" :{
        System.out.println("Elegiste: 1.-INSERTAR\n");
        insertar.agregarRegistro(); //Se llama a la clase y a su metodo respectivamente
        break; }
```

```

case "2": {
System.out.println("Elegiste: 2.-CONSULATAR\n");
buscar.búsqueda(); //Se llama a la clase y a su metodo respectivamente
break;}
case "3": {
System.out.println("Elegiste: 3.-MODIFICAR");
eliminar.elimi(1); //Se llama a la clase y a un metodo en este caso es el modificar un
elemento
break;}
case "4":{
System.out.println("Elegiste: 4.-ELIMINAR");
eliminar.elimi(0); //Se llama a la clase y a un metodo en este caso es eliminar un elemento
break;}
case "5":{ //Se sale del menu
cla="5";
break;
}
default : { //Si la opcion que inserto el usuario no es ninguna de las anteriores entra a este
caso
break;}
}
}while(cla!="5");
} //Se muestra todas las opciones que tiene nuestro menu
public static void menu(){
System.out.println("***** Bienvenido*****\n");
System.out.println("1.-INSERTAR");
System.out.println("2.-CONSULATAR");
System.out.println("3.-MODIFICAR");
System.out.println("4.-ELIMINAR");

```

```

System.out.println("5.-Salir\n");
System.out.print("\tIngresa una opcion:");

    }
}

```

CLASE INSERTAR.

```

import java.io.*;
import java.util.*;
import java.lang.SecurityException;

public class Insertar
{
    public static void insertaDato(String s,String nomA){

        ArrayList<String> cs = new ArrayList<String>(0); //Utilizamos un ArrayList para poder
manejar los nombres en el archivo

        try{

File f = new File(nomA); //Verificamos si no hay un archivo ya existen y si no lo hay lo
creamos

if( !f.exists() ){
f.createNewFile();
}

BufferedReader br = new BufferedReader(

new InputStreamReader(new FileInputStream(nomA))); //Creamos las referencias del
tipo de clase que vamos utilizar para escribir en el archivo

String linea = "";

while( (linea = br.readLine()) != null )

{

    cs.add( linea ); //Verificamos cuanto se leera

}

br.close();

```

```

PrintWriter pw = new PrintWriter(new OutputStreamWriter(new
FileOutputStream(nomA)));

for(int i=0; i<cs.size();i++)

{

pw.println( cs.get(i) );    //Ingresamos el texto escrito en el archivo del tamaño del
ArrayList

}

pw.println( s );

pw.close();                //Cerramos el archivo

} catch( IOException exc ){
exc.printStackTrace();

}

}

public void agregarRegistro(){ //Procedemos con el metodo de agregar un registro

String nombreArchivo="";

int aux=0,bandera=0;

Scanner opcion=new Scanner(System.in); //leemos la opcion que se preguntara

System.out.println("¿Desea insertar empleado(1) o gerente(2)?"); //Preguntamos que tipo
de trabajador se desea agregar

String op=opcion.next();

switch(op){ //Usamos un switch para poder entrar a la opcion seleccionada

case "1":{

nombreArchivo="empleados.txt"; //Si es un tipo empleado lo guardaremos en un archivo
llamado empleados

break;

}

case "2":{

nombreArchivo="gerentes.txt"; //Si es un gerente lo guardaremos en un archivo llamado
gerentes

aux=1;

```

```

break;

}default:{    //En caso de que sea una opcion diferente entra en el default
System.out.println("Opcion invalida");
bandera=1;
break;
}
}

if(bandera==0){
//objeto que se va a escribir en el archivo
Empleado empleado = new Empleado();
Gerente gerente = new Gerente();
Scanner entrada = new Scanner(System.in);
String bandera2="";

System.out.printf("%s\n","Para terminar la incersion presione <ctrl> z y oprima Intro");
//Especificamos el como se debe realizar la insercion del tipo de empleado
System.out.printf("%s\n","Escriba numero de cuenta(>0),Nombre,Departamento,Sueldo");
while(entrada.hasNext()){//itera hasta encontrar el indicador de fin de entrada
try{
//obtiene los datos que se van a almacenar
if(aux==1){
gerente.setNumeroDeCuenta(entrada.nextInt());//lee el numero de cuenta
bandera2="" + gerente.getNumeroDeCuenta();
System.out.println(""+bandera2.length());
gerente.setNombre(entrada.next());//lee el nombre
gerente.setDepartamento(entrada.next());//lee el departamento
gerente.setSueldo(entrada.nextDouble());//lee el sueldo-->bono = saldo*.20 + saldo;
gerente.setBono((gerente.getSueldo()*0.20));

```

```

gerente.setSuelo(gerente.getSuelo()+gerente.getBono());//lee el sueldo-->bono =
saldo*.20 + saldo;

}else{

empleado.setNumeroDeCuenta(entrada.nextInt());//lee el numero de cuenta

bandera2=""+empleado.getNumeroDeCuenta();

empleado.setNombre(entrada.next());//lee el nombre

empleado.setDepartamento(entrada.next());//lee el departamento

empleado.setSuelo(entrada.nextDouble());//lee el sueldo

}

//se verifica que el empleado y el gerente tengan un numero de cuenta mayor a 0 y de 9
digitos

if((empleado.getNumeroDeCuenta()>0||gerente.getNumeroDeCuenta()>0)&&
bandera2.length()==9){

if(aux==1){

insertaDato(gerente.concatenacion(),nombreArchivo);

}else{

insertaDato(empleado.concatenacion(),nombreArchivo);

}

}else{

System.out.println("El numero de ceutna debe ser mayor que cero");

System.out.println("Y debe tener 9 digitos");

}

} catch(NoSuchElementException elementException){

System.err.println("Entrada invalida intente de nuevo");

entrada.nextLine();//descarta la entrada para que el usuario intente de nuevo

} //fin de catch

System.out.printf("%s\n", "Escriba numero de cuenta(>0),Nombre,Departamento,Suelo");

} //fin de while()

} } //fin de agregar registro }

```


CLASE GERENTE.

```
import java.util.Scanner;

public class Gerente extends Empleado //Hacemos que el gerente sea hijo de la clase
empleado

{
    private double bono=0.0;

    public Gerente(){
        super(); //Se llama a todos los tributos y metodos del padre
    }

    public void prueba(){
        String pru="" + getSueldo();
        System.out.println(pru);
    }

    //setters and getters

    public void setBono(double bono){
        this.bono=bono;
    }

    public double getBono(){
        return this.bono;
    }

    //fin de setters and getters
}
```

CLASE EMPLEADO.

```
public class Empleado {  
    private String nombre;  
    private String departamento;  
    private int numeroDeCuenta;  
    private double sueldo;  
  
    //constructor sin valores que llama a otro constructor con valores predeterminados  
    public Empleado(){  
        this(0,"", "",0.0);  
    }  
  
    //inicializa un registro  
    public Empleado(int numC, String nom, String dep, double suel){  
        setNombre(nom);  
        setDepartamento(dep);  
        setNumeroDeCuenta(numC);  
        setSueldo(suel);  
    }  
  
    //setters and getters  
    public void setNombre(String nombre){  
        this.nombre=nombre;  
    }  
  
    public void setDepartamento(String departamento){  
        this.departamento=departamento;  
    }  
  
    public void setNumeroDeCuenta(int numeroDeCuenta){  
        this.numeroDeCuenta=numeroDeCuenta;  
    }  
  
    public void setSueldo(double sueldo){
```

```
this.sueldo=sueldo;

}

public String getNombre(){
return this.nombre;
}

public String getDepartamento(){
return this.departamento;
}

public int getNumeroDeCuenta(){
return this.numeroDeCuenta;
}

public double getSueldo(){
return this.sueldo;
}

//fin de setters and getters

public String concatenacion(){ //Mostamos todos los valores que se metieron
String cs=getNumeroDeCuenta()+",""+getNombre()+",""+
getDepartamento()+",""+getSueldo();
return cs;
}

} //fin de la clase
```

CLASE ELIMINAR.

```
import java.io.*;
import java.util.*;
import java.lang.SecurityException;

public class Eliminar
{
    public void borrar(String id, String nombreA,String nombreB)
    {
        ArrayList<String> contenido = new ArrayList<String>(0); //Creamos la referencia tipo
        ArrayList

        String idElimina = id;

        try //Definimos la parte de lectura del archivo y lo pasamos al array list
        {
            BufferedReader br = new BufferedReader(new InputStreamReader(new
            FileInputStream(nombreA)));

            String linea = "";

            while( (linea = br.readLine()) != null )
            {
                if( !linea.contains( idElimina ) )
                {
                    contenido.add( linea );
                }
            }

            br.close();

            PrintWriter pw = new PrintWriter(new OutputStreamWriter(new
            FileOutputStream(nombreB)));

            for(int i=0; i<contenido.size();i++)
            {
                pw.println( contenido.get(i) );
            }
        }
    }
}
```

```

}
pw.close();
}
catch( IOException exc )
{
System.out.println("IOException generada");
exc.printStackTrace();
}
}

public void elimi(int op1){
//Aqui vamos a procer a eliminar o modificar un archivo
Scanner op=new Scanner(System.in);
String clave="";
String opcion="";
Buscar b=new Buscar(); //Creamos una referecia para buscar el dato
if(op1==0){
System.out.println("Desea eliminar gerente(1) o empleado(2)?");
}else{
System.out.println("Desea modificar gerente(1) o empleado(2)?");
}
//Usamos el switch para entrar a lo que el usuario desea realizar
opcion="" + op.next();
switch(opcion){
case "1":{
System.out.println("Introduce la cave del empleado:");
clave="" + op.next(); //Pedimos la clave y la guardamos en una variable
if(clave.length()==9){ //Verificamos que la longitud de la calve sea del
tamaño que necesitamos

```

```

if(b.busqueda(clave,"gerentes.txt")){ //En caso de que sea procedemos a
    buscar en el archivo la clave
    if(op1==0){
        borrar(clave,"gerentes.txt","gerentes.txt"); //Llama al metodo Borrar y borra el elemento
        System.out.println("Elemento eliminado");
    }else{
        modificar(clave,"gerentes.txt",1); //Llama al metodo modificar y modifica el elemento
        System.out.println("Elemento modificado correctamente");
    }
}
}
}else{
    System.out.println("El numero de cuenta debe ser de 9 digitos");
}
break;
}case "2":{ //Pedimos la clave y la guardamos en una variable
    System.out.println("Introduce la clave del empleado:");
    clave="" + op.next();

    if(clave.length()==9){ //Verificamos que la longitud de la clave sea del tamaño que
        necesitamos
        if(b.busqueda(clave,"empleados.txt")){
            if(op1==0){ //En caso de que sea procedemos a buscar en el archivo la clave
                borrar(clave,"empleados.txt","empleados.txt");
                System.out.println("Elemento eliminado");
            }else{
                modificar(clave,"empleados.txt",0);
            }
        }
    }
}
}else{
    System.out.println("El numero de cuenta debe ser de 9 digitos");
}
}

```

```

    }
    break;
    }default :{
    System.out.println("!!!Opcion invalida!!!");
    break;
    }
    }
    }

//Metodo modificar
public void modificar(String id,String nombreA,int aux){
    ArrayList<String> contenido = new ArrayList<String>(0);
    String idElimina = id;
    String bandera="";
    Gerente gerente = new Gerente();
    Empleado empleado = new Empleado();
    Scanner entrada = new Scanner(System.in);
    try{ //Entra al archivo
    BufferedReader br = new BufferedReader(
    new InputStreamReader(
    new FileInputStream(nombreA)));
    String linea = "";
    while( (linea = br.readLine()) != null ) //Verifica que el archivo no este vacio
    {
    if( !linea.contains( idElimina ) )
    {
    contenido.add( linea );
    }else{
    if(aux==1){

```

```

try{ //Para un gerente

System.out.println("Ingresa el nuevo numero de cuenta:");

int numC=entrada.nextInt();

gerente.setNumeroDeCuenta(numC);//lee el numero de cuenta

bandera="" + gerente.getNumeroDeCuenta();

System.out.println("Ingresa el nuevo nombre:");

String nom=entrada.next();

gerente.setNombre(nom);//lee el nombre

System.out.println("Ingresa el nuevo departamento:");

String dep=entrada.next();

gerente.setDepartamento(dep);//lee el departamento

System.out.println("Ingresa el nuevo sueldo:");

double sal=entrada.nextDouble();

gerente.setSueldo(sal);//lee el sueldo-->bono = saldo*.20 + saldo;

gerente.setBono((gerente.getSueldo()*0.20));

gerente.setSueldo(gerente.getSueldo()+gerente.getBono());//lee el sueldo-->bono =
saldo*.20 + saldo;

if(gerente.getNumeroDeCuenta()>0 && bandera.length()==9){

contenido.add(gerente.concatenacion());

System.out.println("Elemento modificado correctamente");

}else{

System.out.println("El numero de cuenta debe ser mayor que cero");

System.out.println("Y de nueve digitos");

contenido.add( linea );

}

}catch(NoSuchElementException elementException){

System.err.println("Entrada invalida intente de nuevo");

contenido.add( linea );//entrada.nextLine();//descarta la entrada para que el usuario intente
de nuevo

```



```

} //fin de catch

}else{

try{ //Para un empleado

System.out.println("Ingresa el nuevo numero de cuenta:");

int numC=entrada.nextInt();

empleado.setNumeroDeCuenta(numC);//lee el numero de cuenta

bandera="" + empleado.getNumeroDeCuenta();

System.out.println("Ingresa el nuevo nombre:");

String nom=entrada.next();

empleado.setNombre(nom);//lee el nombre

System.out.println("Ingresa el nuevo departamento:");

String dep=entrada.next();

empleado.setDepartamento(dep);//lee el departamento

System.out.println("Ingresa el nuevo sueldo:");

double sal=entrada.nextDouble();

empleado.setSueldo(sal);//lee el sueldo

if(empleado.getNumeroDeCuenta()>0 && bandera.length()==9){

contenido.add(empleado.concatenacion());

System.out.println("Elemento modificado correctamente");

}else{

System.out.println("El numero de cuenta debe ser mayor que cero");

System.out.println("Y de nueve digitos");

contenido.add( linea );

}

} catch(NoSuchElementException elementException){

System.err.println("Entrada invalida intente de nuevo");

contenido.add( linea );

//entrada.nextLine();//descarta la entrada para que el usuario intente de nuevo

```

```
}//fin de catch  
  
}  
  
}  
  
} //Cierra el archivo br.close();  
  
PrintWriter pw = new PrintWriter(new OutputStreamWriter(new  
FileOutputStream(nombreA)));  
  
for(int i=0; i<contenido.size();i++)  
{  
  
    pw.println( contenido.get(i) );  
  
}  
  
pw.close();  
  
}catch( IOException exc ){  
System.out.println("IOException generada");  
exc.printStackTrace();  
  
}  
  
}  
  
}
```

CLASE BUSCAR.

```
import java.io.*;
import java.util.Scanner;

public class Buscar{

    public boolean busqueda(String clave,String nombreA){
        String idConsulta = clave;
        int aux=0;
        boolean temp=false;

        try{           //Abrimos el archivo para poder usarlo
            BufferedReader br = new BufferedReader(new InputStreamReader(new
            FileInputStream(nombreA)));

            String linea = "";
            while( (linea = br.readLine()) != null )//Verificamos que no este vacio
            {
                if( linea.contains( idConsulta ) )
                {
                    System.out.println( linea );
                    aux=1;
                }
            }

            if(aux!=0){ //Vemos si el dato se encontro
                System.out.println("El empleado fue encontrado");
                temp=true;
                aux=0;
            }else{
                System.out.println("El empleado NO fue encontrado");
                temp=false;
                aux=0;
            }
        }
```

```

aux=0;

br.close();          //Cerramos el archivo

return temp;         //Retornamos que encontro el metodo
}

catch( IOException exc )
{
    System.out.println("IOException generada");
    exc.printStackTrace();
}

return temp;
}

public void busqueda(){ //Leemos lo que el usuario desea hacer

String clave="";

String nomA="";

Scanner c = new Scanner(System.in);

System.out.println("\tDesea buscar gerente(1) o empleado(2)?");

String op="" + c.next();

switch(op){ //Usamos el switch para entrar a lo que se realizara

case "1":{ //Si es un gerente se realiza lo siguiente para la busqueda

nomA="gerentes.txt"; //Pasamos el nombre del archivo

System.out.println("\tEscribe el numero de cuenta: ");

clave="" + c.next();

if(clave.length()==9){          //Verificamos que la clave tenga lo que necesitamos

busqueda(clave,nomA);

}else{

System.out.println("El numero de cuenta debe tener 9 digitos");

}

break;

```

```
}case "2":{ //Si es un empleado se realiza lo siguiente para la busqueda
nomA="empleados.txt";    //Pasamos el nombre del archivo
System.out.println("\tEscribe el nuemro de cuenta: ");
clave="" + c.next();
if(clave.length() == 9){ //Verificamos que la clave tenga lo que necesitamos
busqueda(clave,nomA);
}else{
System.out.println("El numero de cuetna debe tener 9 digitos");
}break;
}default:{
System.out.println("opcion invalida");
break;
}
}
}
}
```