

Universidad Central Marta Abreu de las Villa (UCLV)

Facultad de Matemática-Física-Computación

2024

**PRÁCTICA LABORAL E INVESTIGATIVA**



Proyecto WebSoft. SA

Autores: Carlos Elías González Valdés   
 René Espinosa Arteaga  
 Mauricio J. Avalo Tamayo

**Tabla de contenido**

[**Resumen** 2](#_Toc156079317)

[**Introducción** 3](#_Toc156079318)

[**Desarrollo** 4](#_Toc156079319)

[**Conclusiones** 11](#_Toc156079320)

[**Bibliografía** 11](#_Toc156079321)

[**Anexos** 12](#_Toc156079322)

**Resumen**

La gestión eficiente de la información en empresas de desarrollo informático es crucial. Mediante este proyecto se desarrolla una aplicación en Java para facilitar el gestionamiento de la información generada en la empresa Websoft S.A. Esta aplicación permite el gestionamiento de una empresa con sus respectivos departamentos proyectos y trabajadores. Es capaz de manejar información con respecto al trabajador como salarios y ausencias, además de otros datos de interés para dicha empresa. A partir del paradigma de programación orientada a objetos y los conocimientos adquiridos en la asignatura, se le da solución óptima y eficaz al problema planteado.

**Introducción**

Gestionar la información generada en las empresas de desarrollo informático es crucial. Una gestión eficiente de la información permite, realizar toma de decisiones informadas, basadas en hechos concretos y precisos. La gestión adecuada de la información en empresas de desarrollo informático es esencial para lograr eficiencia, cumplir con normativas, retener talento y mantenerse competitivas en un entorno empresarial cada vez más dinámico.

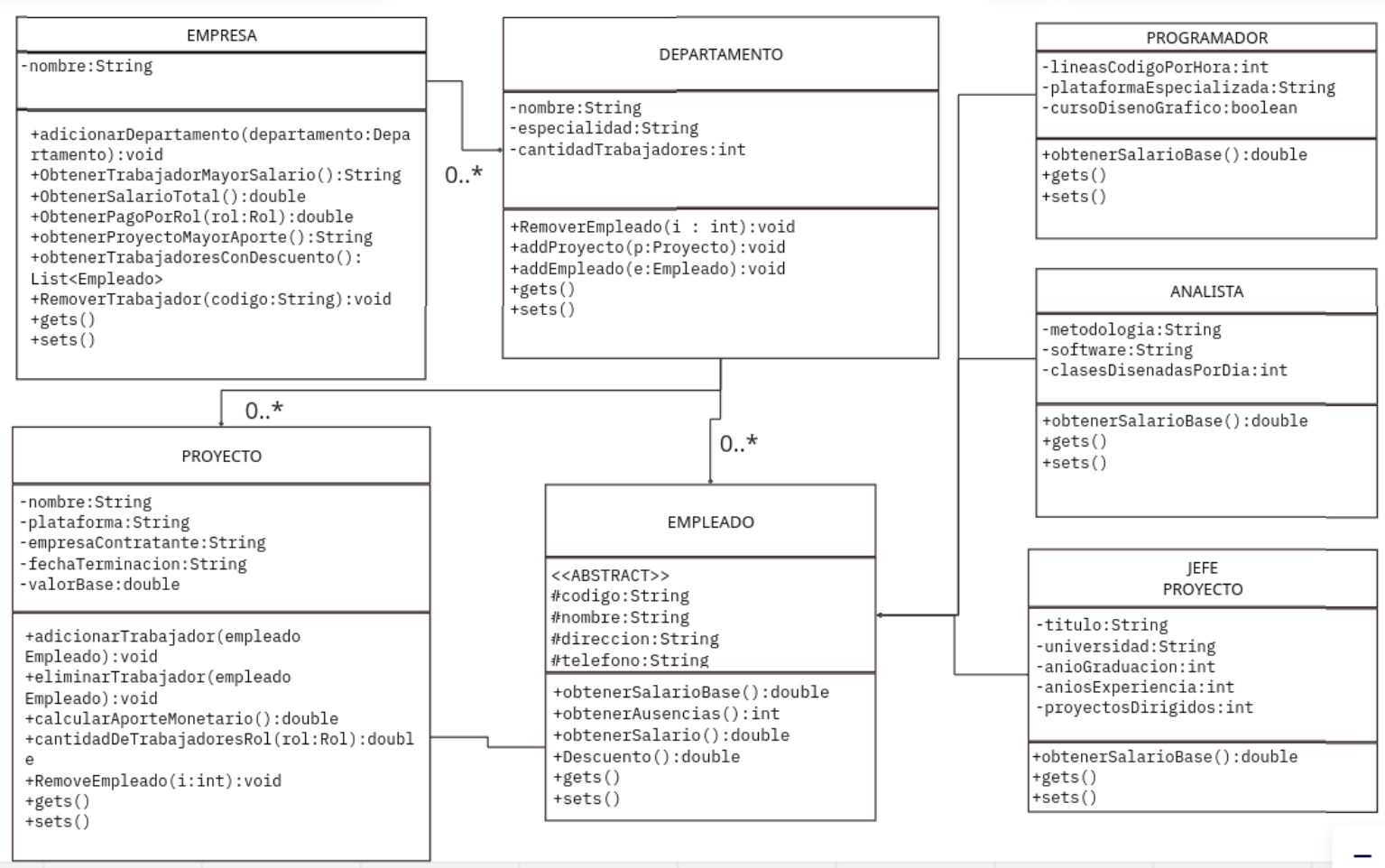
La empresa Websoft S.A. dedicada a la producción de sitios Web, se encuentra dividida en diferentes departamentos, con sus respectivos trabajadores y proyectos. De cada departamento se conoce su nombre, especialidad y cantidad de trabajadores. Se desea implementar un sistema que permita realizar las siguientes funcionalidades:

* Crear una empresa.
* Adicionar los trabajadores a la empresa, cada uno con su rol.
* Adicionar nuevos proyectos.
* Obtener el nombre del trabajador de mayor salario.
* Obtener el salario total a pagar por la empresa.
* Dar baja a un trabajador, lo que implica sacarlo del proyecto en los que estuviera trabajando en un momento determinado y en caso que se desee, sustituirlo en sus funciones por un trabajador determinado.
* Obtener cuánto paga la empresa a todos los trabajadores por concepto de jefe de proyecto, analista o desarrollador según se desee.
* Obtener el proyecto de mayor aporte monetario a los trabajadores de la empresa.
* Obtener las ausencias por trabajador
* Saber cuáles trabajadores tuvieron un determinado descuento por los días dejados de trabajar.

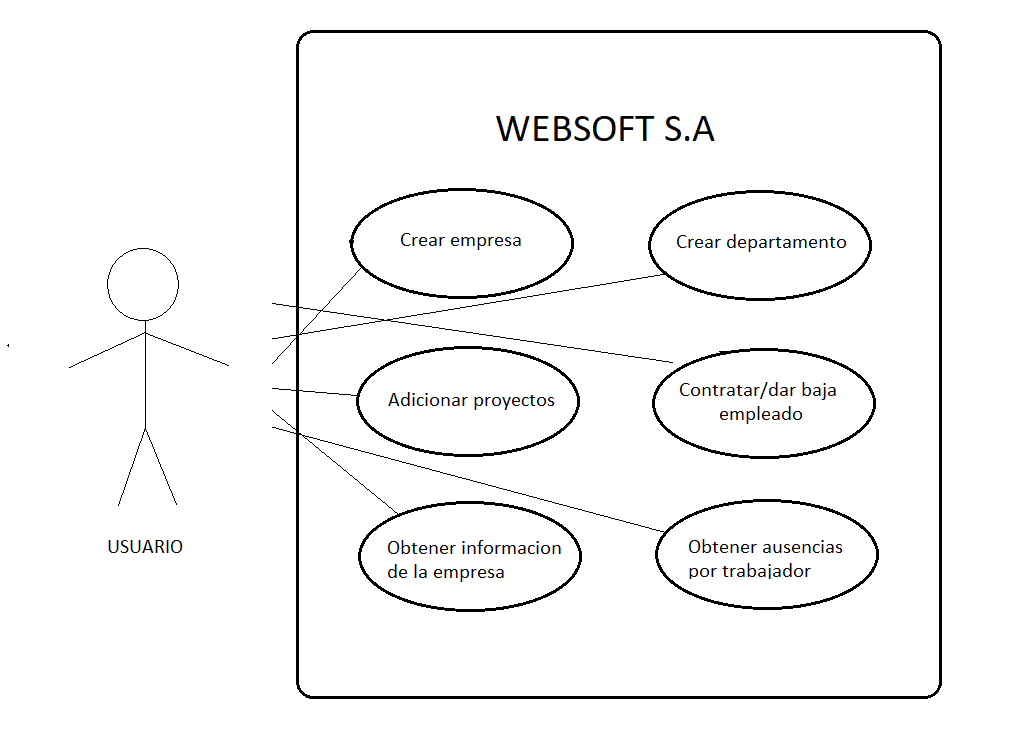
Este proyecto pretende gestionar la información de la empresa mediante el diseño de una aplicación que cumpla con las especificaciones anteriores. Se pretende desarrollar una interfaz gráfica dinámica y fácil de usar, con el propósito de mejorar la interacción del usuario.

Además, se procura la estructuración adecuada de las clases del proyecto para garantizar que el código pueda extenderse y resulte comprensible para otros usuarios. Se busca una organización que asegure la modularidad y claridad del código, facilitando su mantenimiento y posibles expansiones.

**Desarrollo**

**Diagrama de Clases**

**Diagrama de casos de uso**



**Clases definidas que se utilizaron en la solución.**

Clase Empresa

Esta posee el nombre de la empresa que se gestiona y una lista de departamentos en los que se divide esta. En esta clase se encuentran la mayoría de los principales métodos que se utilizan para la solución de la problemática del proyecto

Clase Departamento

En esta clase existen diferentes atributos propios de los departamentos como su nombre, especialidad y la cantidad de trabajadores, así como posee una lista de los trabajadores y los proyectos que se asocian a este departamento.

Clase Proyecto

En esta clase existen diferentes atributos de los proyectos: el nombre, el valor base, la plataforma de programación, el nombre de la empresa que solicita el servicio, la fecha de entrega además de una lista con los empleados que se encuentran trabajando en este proyecto

Clase Empleado

En esta clase abstracta existen diferentes atributos comunes de los diferentes tipos de empleados (programador, analista y jefe), entre ellos el código que le identifica, el nombre, dirección, teléfono, salario y los días trabajados

Clase Programador

Esta clase hereda de la clase Empleado y posee: la cantidad de líneas de códigos que realiza por día, plataforma en la que se especializa y si tiene aprobado el título de diseño gráfico o no

Clase Analista

Esta clase hereda de la clase Empleado y posee: metodología que utilizan para trabajar, software utilizado y estimado de clases que diseñan por día.

Clase JefeProyecto

Esta clase hereda de la clase Empleado y posee: título, universidad en la que estudio, año de graduación, años de experiencia como jefe de proyecto y la cantidad de proyectos dirigidos

Clase ControlDeExcepciones:

Se utiliza para un control más simple de las excepciones

Clase Rol:

Se utiliza para definir un tipo de dato que representa un conjunto fijo de constantes en este caso el Rol que ocupa cada trabajador en la empresa.

Clases Excepciones:

1.- FechaIncorrectaException

2- EmpleadoNoExisteException

3 - GraduacionYearException

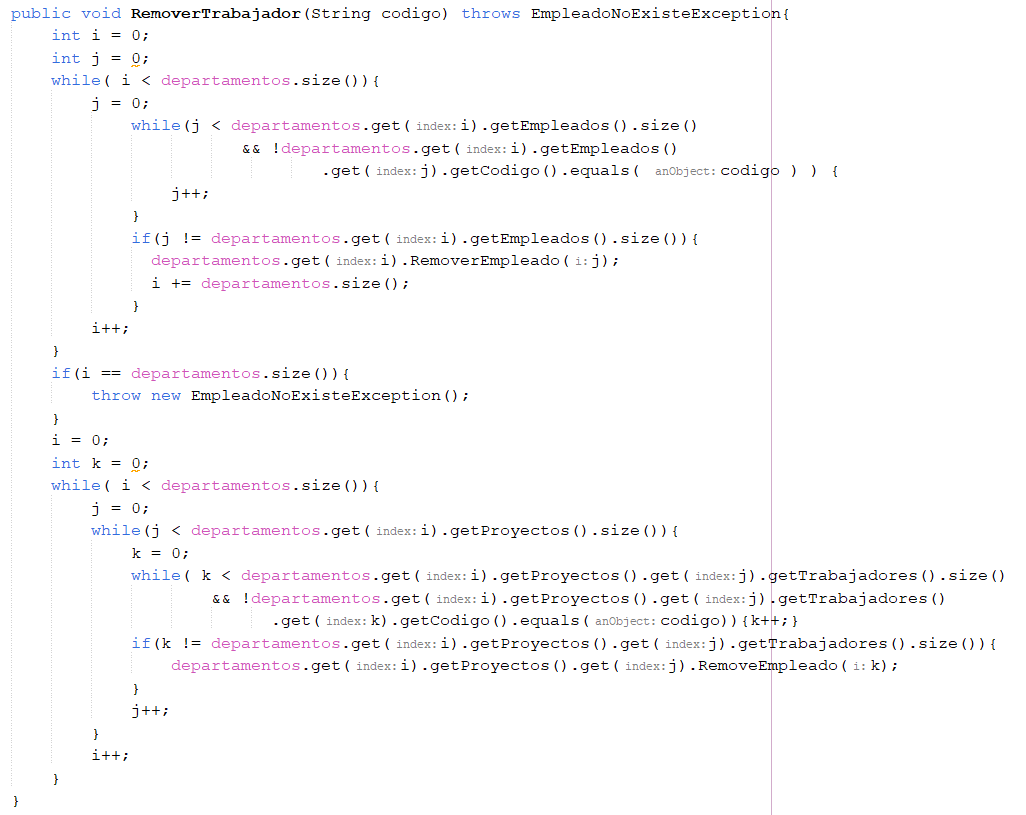
4 - DepartamentoYaCreadoException

5 – IncorrectNumberFormatException

6 – ProyectoYaCreadoException

7 - VariosJefesDeProyectoException

**Algunos métodos de importancia utilizados en la solución del problema**

****

RemoverTrabajador: Este se encarga de eliminar un trabajador específico de los departamentos y proyectos de la empresa. A continuación se muestra una explicación paso a paso de lo que hace:

1. Recibe como parámetro un código que identifica al trabajador que se desea eliminar.

2. Declara dos variables enteras, "i" y "j", que se utilizarán para iterar sobre los departamentos y empleados respectivamente.

3. Comienza un bucle "while" que se ejecutará mientras "i" sea menor que el tamaño de la lista de departamentos.

4. Entra en un segundo bucle "while" que se ejecutará mientras "j" sea menor que el tamaño de la lista de empleados del departamento actual y el código del empleado actual no sea igual al código proporcionado.

5. Si se encuentra un empleado con el código proporcionado, se ejecuta el siguiente bloque de código:

- Llama al método "RemoverEmpleado" del departamento actual, pasando el índice "j" como argumento para eliminar al empleado de la lista de empleados.

- Incrementa el valor de "i" en la cantidad de departamentos existentes, lo que hace que salga del bucle exterior.

6. Incrementa el valor de "i" en 1 para pasar al siguiente departamento y continúa con la siguiente iteración del bucle exterior.

7. Después de salir del bucle exterior, verifica si "i" es igual al tamaño de la lista de departamentos. Si es así, significa que no se encontró ningún empleado con el código proporcionado y se lanza una excepción de tipo "EmpleadoNoExisteException".

8. Reinicia el valor de "i" a 0 y declara una nueva variable entera "k" que se utilizará para iterar sobre los proyectos.

9. Comienza otro bucle "while" que se ejecutará mientras "i" sea menor que el tamaño de la lista de departamentos.

10. Entra en un tercer bucle "while" que se ejecutará mientras "j" sea menor que el tamaño de la lista de proyectos del departamento actual.

11. Dentro del tercer bucle, entra en un cuarto bucle "while" que se ejecutará mientras "k" sea menor que el tamaño de la lista de trabajadores del proyecto actual y el código del trabajador actual no sea igual al código proporcionado.

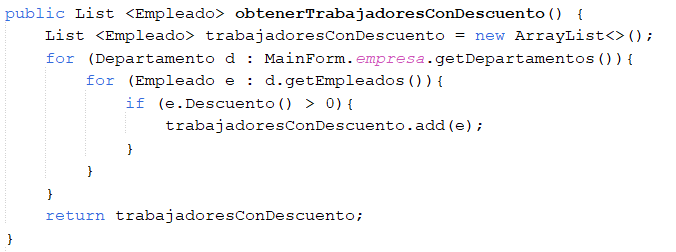
12. Si se encuentra un trabajador con el código proporcionado, se ejecuta el siguiente bloque de código:

- Llama al método "RemoveEmpleado" del proyecto actual, pasando el índice "k" como argumento para eliminar al trabajador de la lista de trabajadores.

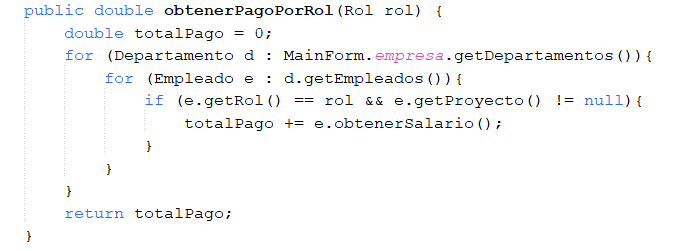
13. Incrementa el valor de "j" en 1 para pasar al siguiente proyecto y continúa con la siguiente iteración del bucle intermedio.

14. Incrementa el valor de "i" en 1 para pasar al siguiente departamento y continúa con la siguiente iteración del bucle exterior.

15. Una vez que se completa el último bucle, el método ha terminado de eliminar al trabajador de todos los departamentos y proyectos en los que estaba involucrado.

****

obtenerTrabajadoresConDescuento: Este método se encarga de añadir a un ArrayList de empleados, todos aquellos empleados de los departamentos de la empresa cuyo método “Descuento” devuelva un valor mayor a 0.



obtenerPagoPorRol: Este método se encarga de sumar los salarios de los empleados de cada departamento según su atributo “Rol” la suma se va almacenando en una variable (totalPago) y finalmente el método devuelve el valor final de la variable.

**Descripción de las pruebas realizadas al programa. Análisis de los resultados de las pruebas.**

A lo largo del desarrollo del proyecto, aprovechando las ventajas de la Programación Orientada a Objetos, se llevaron a cabo pruebas sistemáticas en las distintas clases y métodos a medida que se iban codificando.

Las pruebas se enfocaron en casos de uso específicos, abarcando diversas entradas de datos, la gestión de diálogos de error desencadenados por diversas excepciones y, de manera destacada, el manejo de archivos.

Al concluir, se sometió al software a una prueba integral, simulando numerosos escenarios posibles. Se evaluaron el tiempo necesario para realizar operaciones y el uso de la memoria utilizada.

**Conclusiones**

* Se diseñó una aplicación en Java, que da solución al problema orientado, capaz de gestionar la información de la empresa Websoft,
* Se desarrolló una interfaz que permite una interacción simple con el usuario
* Se estructuraron de manera adecuada las clases de manera que el código puede ser utilizado y extendido por otros programadores
* Se aplicaron y profundizaron los conocimientos y habilidades adquiridas en la asignatura de Programación, específicamente en el lenguaje Java.

**Bibliografía**

*miro*. (2024). Obtenido de https://miro.com/es/plantillas/diagrama-uml/

Vázquez, R. P. (2023). Conferencias Programación . Santa Clara.

Zukowski, J. (2003). *Programación Java 2 .* Habana: Felix Varela.

**Anexos**



