PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

**Proyecto de Programación orientada a objetos**

**Easy Shop**

Nicholas Antonio Espinoza Berroeta

Matías Antonio Pinto Díaz

Franco Antonio Riveros Vivar

Profesor: Claudio Cubillos Figueroa

18 de septiembre 2017

**Índice**

1.- Introducción…………………………………………………………………1

2.- Dominio del problema…………….…………………………………………2

3.- Análisis………………………………………………………………………3

4.- Diseño……..…………………………………………………………………4

4.1.- Clases……………………….………………….…..………………4

4.2.- Diagrama general de organización de las clases…..……………….5

4.3.- Diagrama de clases (UML)…………………………………..……6

4.3.1.- Clases UML……………………………………………..7

4.4.- Interfaz……….……………………………………………………9

5.- Planificación.………………………………………………………...……..12

6.- Conclusión.…………………………………………………………………14

I

**Lista de Figuras**

Figura 1. Diagrama general de clase……..……………………......………..….....5

Figura 2. Diagrama UML……………………………………..…………..….......6

Figura 2.1 Clase Empresa UML.…………………………………..…......7

Figura 2.2 Clase Trabajador UML.………………………………..……..7

Figura 2.3 Clase Shop UML……..………………………………............7

Figura 2.4 Clase Artículo UML...………...………………………..…….7

Figura 2.5 Clase Proyecto UML (Main)……………………………...….8

Figura 2.6 Clase Factura UML……………………………………..…....8

Figura 3. Login………………………………………………………………..….9

Figura 3.1 Menú principal...…………………………………….…….…9

Figura 3.2 Agregar tienda…………………………………………..…..10

Figura 3.3 Eliminar tienda…...................................................................10

Figura 3.4 Mostar/guardar tienda….........................................................11

Figura 3.5 Menú tienda…........................................................................11

Figura 3.6 Agregar Gerente......................................................................11

II

**Lista de Tablas**

Tabla 1. Carta Gantt grupal………...................................................................12

Tabla 1.1 Carta Gantt Matías Pinto………..............................................12

Tabla 1.2 Carta Gantt Nicholas Espinoza.................................................12

Tabla 1.3 Carta Gantt Franco Riveros………..........................................13

III

**1 Introducción**

A estas alturas la tecnología avanza a pasos agigantados, es por eso que las empresas informáticas, cada vez tienen que crear nuevos y más aún sofisticados softwares para cualquier tipo de negocio en la actualidad.

En el siguiente informe se dará a conocer un software llamado “Easy Shop”, que está especializado principalmente en la administración de empresas. Y que busca, en lo posible, de innovar las antiguas herramientas existentes en el mercado laboral actual.

Cuando se mejora tanto el ambiente como el sistema central de una empresa, siempre tiende a la comodidad para sus empleados y aún más importante, a la eficiencia de estos mismos.

Es por esto que la incorporación de nuevas tecnologías informáticas siempre estará en una alta demanda en el mercado.

Uno de los más claros ejemplos de avances, es el de la administración, todas las empresas necesita de un sistema capaz de trabajar de manera eficiente con muchos datos, ya sea nombres, sueldos, cargos, etc. Y manejarlos sin cometer ningún error.

**2 Dominio del problema**

“Easy Shop” es un software de servicios dedicado a la organización y administración de empresas y da soporte a soluciones de tecnología informática, orientado a apoyar la alta eficiencia y seguridad en el manejo de grandes cantidades de datos a sus clientes. El manejo de documentos, representa para la empresa, algunas situaciones que no son las más adecuadas.

En términos generales, existe un problema de dificultad del acceso a los documentos que maneja la empresa y el riesgo de pérdida de estos. Desde el momento en que se genera algún dato hasta que se guarde de manera segura, se hace necesario llevar un control total sobre los mismos.

Cuando un software no es lo suficientemente eficiente o no lleva un debido control cuando se reciben datos, a la hora de buscarlo, el proceso se hace bastante tedioso porque no se sabe dónde está o quien lo tiene. Esto genera pérdida de tiempo y un alto costo para la empresa en cuestión.

Es por esto que surge “Easy Shop”, un software que busca la eficiencia en el manejo de datos y una debida seguridad al momento de proteger estos mismos.

**3 Análisis**

Este software maneja datos de la empresa como locales, artículos, trabajadores, facturas. Y será necesario que el administrador ingrese los datos que el programa pida.

Asumiendo que los datos generales de la empresa los sabemos (nombre, CEO (dueño de la empresa)), esta también presenta una lista de objetos y aquí entramos a la información más específica, como los locales. El administrador estará a cargo de ingresar todos los datos, como lo son: dirección, id (Strings) y gerente (tipo trabajador). Cada local tendrá una lista de artículos, trabajadores y facturas.

El objeto Articulo tiene información como nombre, código (datos tipo String), stock, precio y descuento (datos tipo int).

Por el lado de Trabajador este presenta información como nombre, rut, cargo (datos tipo String) y sueldo (único dato tipo int). No olvidemos que el gerente de cada local es de tipo trabajador, por lo que, presenta estos mismos datos.

En las facturas hay información como ID (String), ganancia (dato tipo int), una Lista de artículos vendidos y un objeto tipo Trabajador llamado vendedor.

Como características propias del negocio tenemos la de calcular las ganancias de cada tienda y calcular el descuento que puede presentar cada producto (mostrándolos del producto con mayor descuento, al menor).

Los locales, artículos como también los trabajadores, podrán ser agregados, eliminados o modificados.

Los datos que presenta el programa son todos los que se pueden encontrar normalmente en cualquier tipo de empresa tales como: trabajadores, sueldos, puestos de trabajo, inventario, rut, facturas, ganancias, precios. También la capacidad de manipularlos (editarlos) y los métodos y clases presentes.

**4 Diseño**

En este punto mostraremos los diagramas, la estructura general del software y las clases presentes.

**4.1 Clases**

Clase Proyecto

Es el main del programa, maneja todo desde arriba, crea el objeto Empresa y lo envía como parámetro a los constructores de las ventanas para poder editar la empresa con los botones. Proporciona un menú por consola que permite al usuario manejarse dentro de las funciones que el programa ofrece.

Contiene: El objeto Empresa e inicializa la ventana LoginAdmin.

Clase Empresa

Esta clase contiene un ArrayList de la clase Shop, en conjunto con los atributos String CEO y String Nombre que para la entrega B tendrán un uso útil. Aparte de esto, contiene un menú para editar las tiendas en específico.

Contiene: ArrayList Shop, String nombre, String CEO.

Clase Shop

La clase más extensa/pesada del programa, posee dos ArrayLists, uno de artículos y otro de trabajadores. Provee de dos menús para cada ArrayList para agregar, modificar, eliminar y mostrar (consola y archivo) los elementos de los ArrayLists ya mencionadas.

Contiene: ArrayList Artículos, ArrayList Trabajadores, String dirección, String ID, Trabajador Gerente (objeto) y ArrayList Facturas.

Clase Artículo

Clase liviana, posee setter y getters correspondientes y un ID para diferenciarlos y hacerlos únicos. Posee una función única llamada getPrecioDscto, que retorna el precio con el descuento que posee.

Contiene: String nombre, String código, int precio, int stock e int descuento.

Clase Trabajador

Clase liviana, posee setter y getters correspondientes y un Rut para diferenciarlos.

Contiene: String nombre, String rut, int sueldo, String cargo.

Clase Facturas

Clase liviana, setter y getters correspondientes, posee una ID única, tiene un objeto de la Clase Trabajador llamada Vendedor, la cual servirá para futuras funciones, y es parte de un ArrayList de la clase Shop.

**4.2 Diagrama general de organización de las clases**

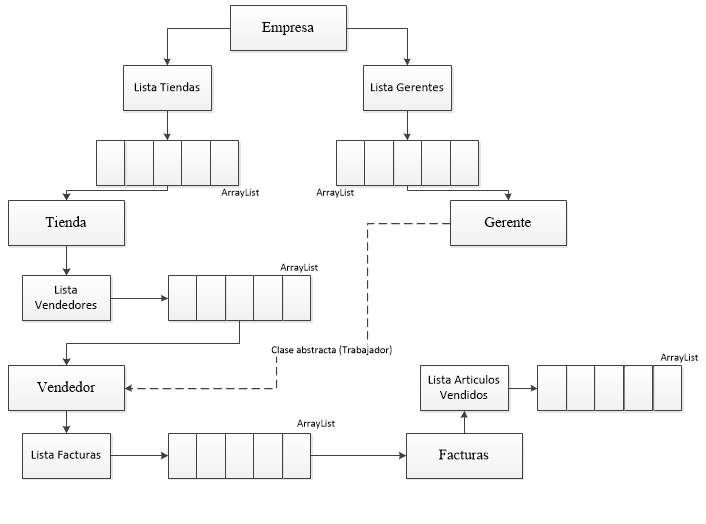
****

Figura 1. Diagrama general de clases.

**4.3 Diagrama de clases (UML)**

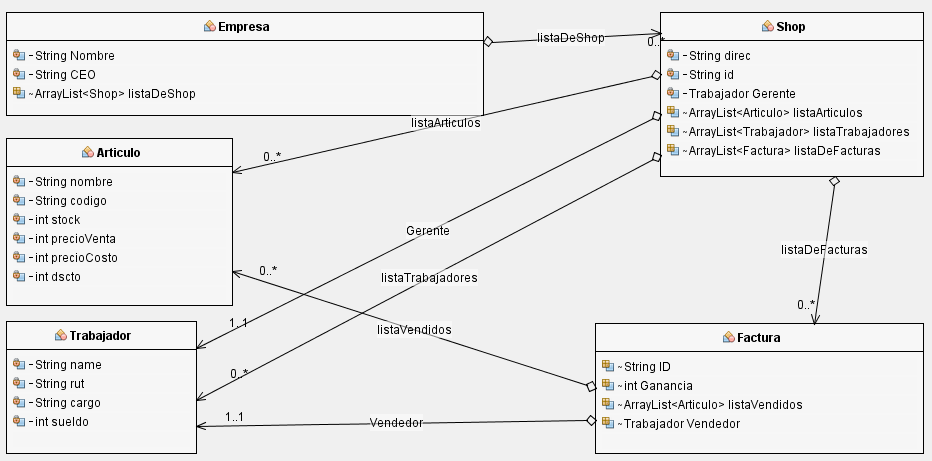
****

Figura 2. Diagrama UML

**4.3.1 Clases UML**

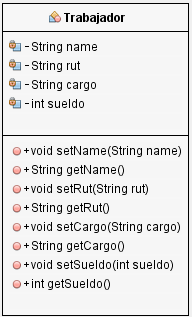
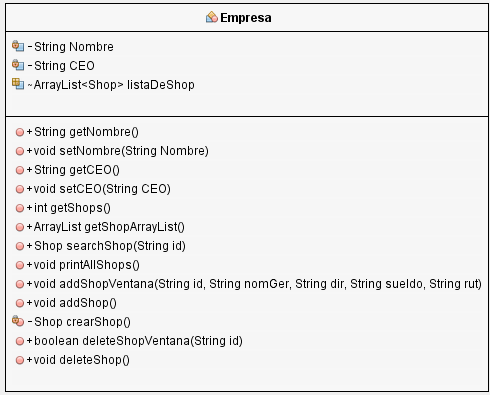


Figura 2.1 Clase Empresa UML Figura 2.2 Clase Trabajador UML

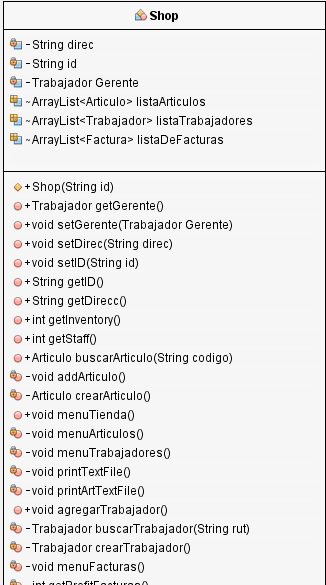
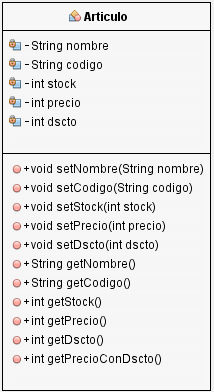
 

Figura 2.3 Clase Shop UML Figura 2.4 Clase Articulo UML

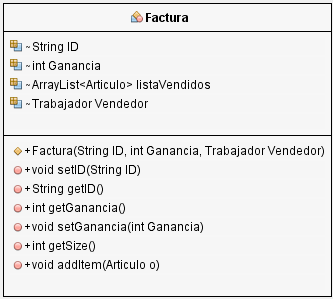
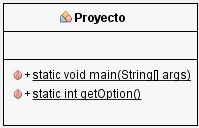


Figura 2.5 Clase Proyecto UML (Main) Figura 2.6 Clase Factura (UML)

**4.4 Interfaz**

Las siguientes interfaces fueron creadas por la herramienta que ofrece NetBeans con la clase JFrame, la cual es una clase creada en swing (biblioteca gráfica) para generar ventanas dinámicas y amigables para el usuario.

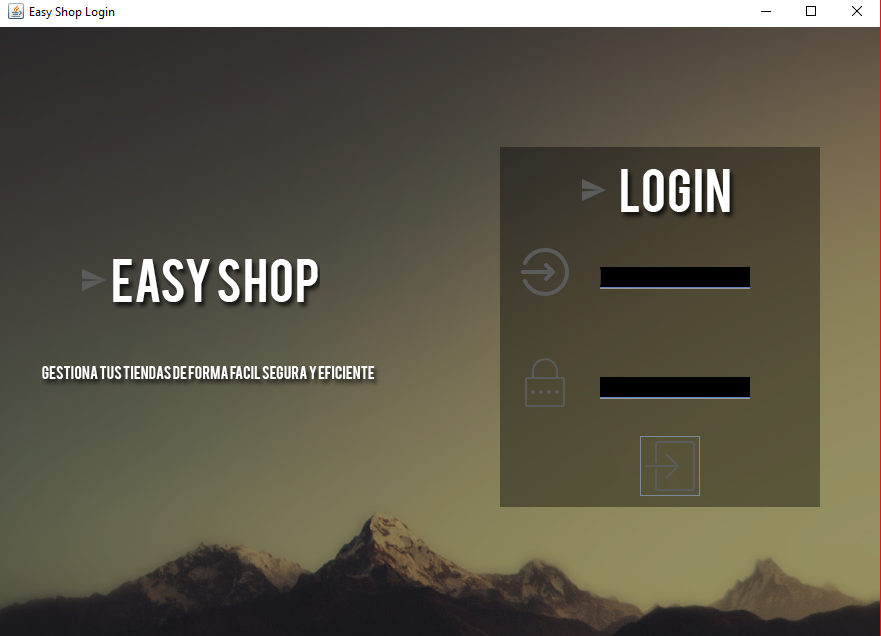


Figura 3. Login

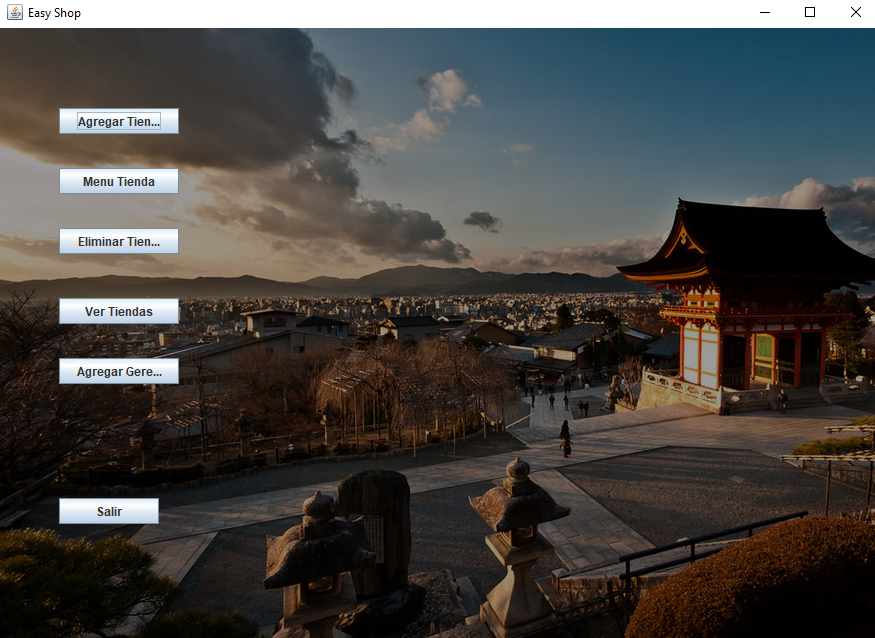


Figura 3.1 Menú principal

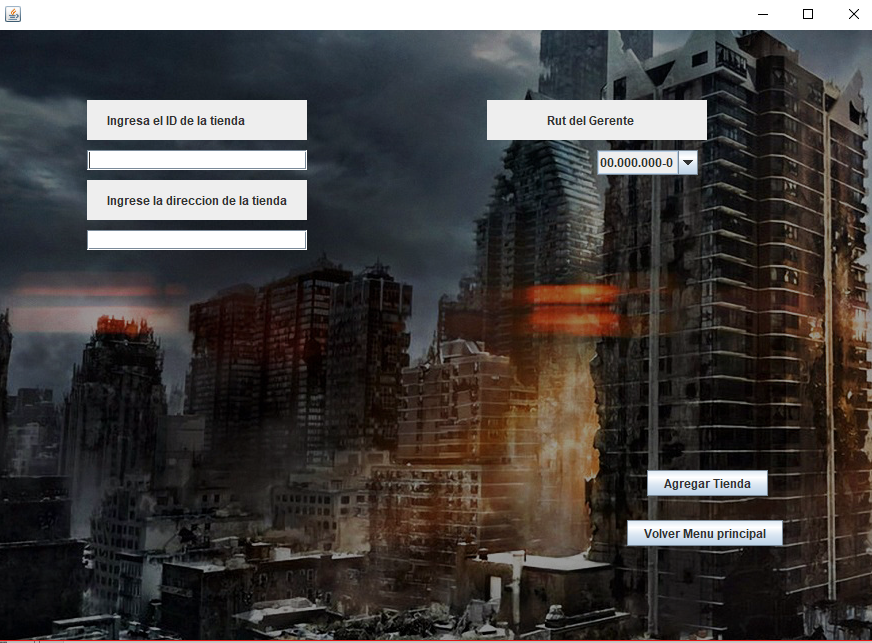


Figura 3.2 Agregar tienda

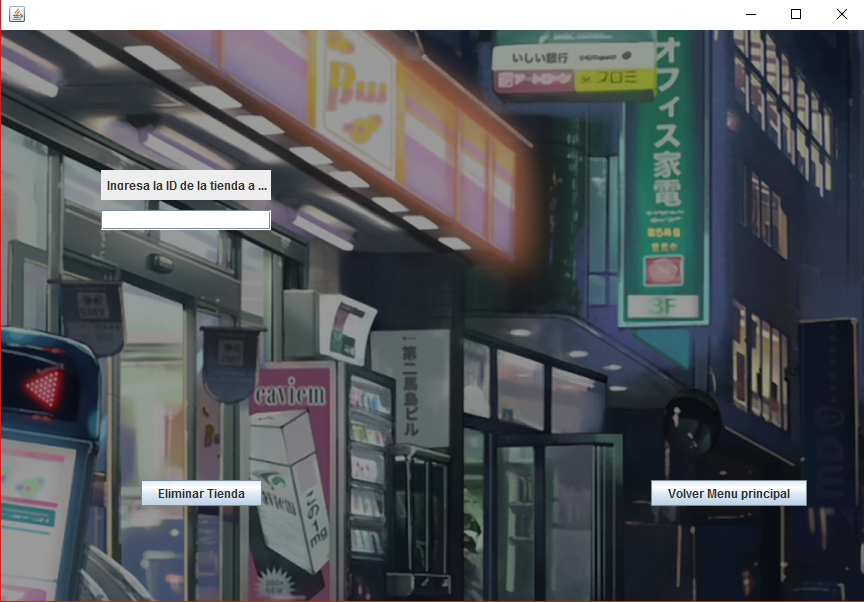


Figura 3.3 Eliminar tienda

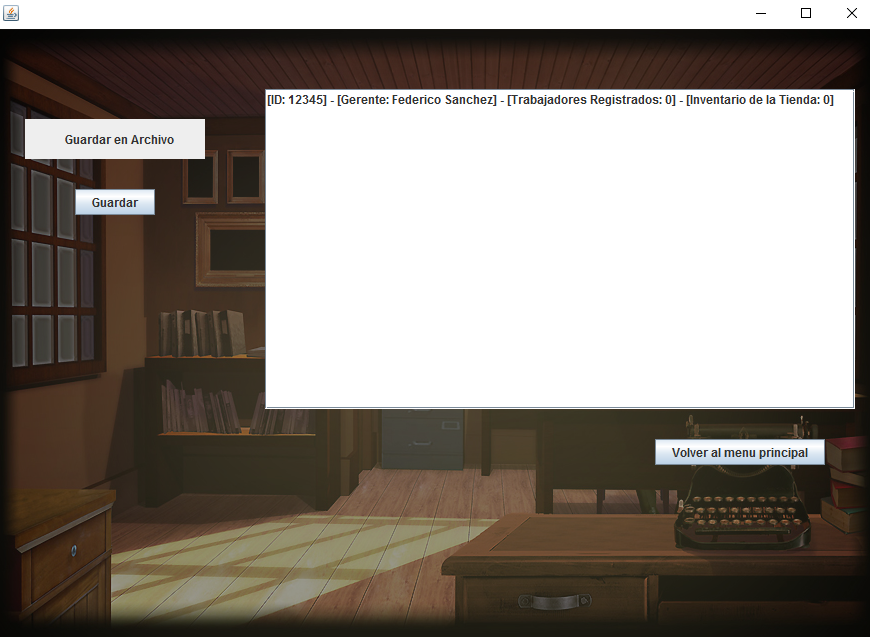


Figura 3.4 Mostrar/Guardar tienda.

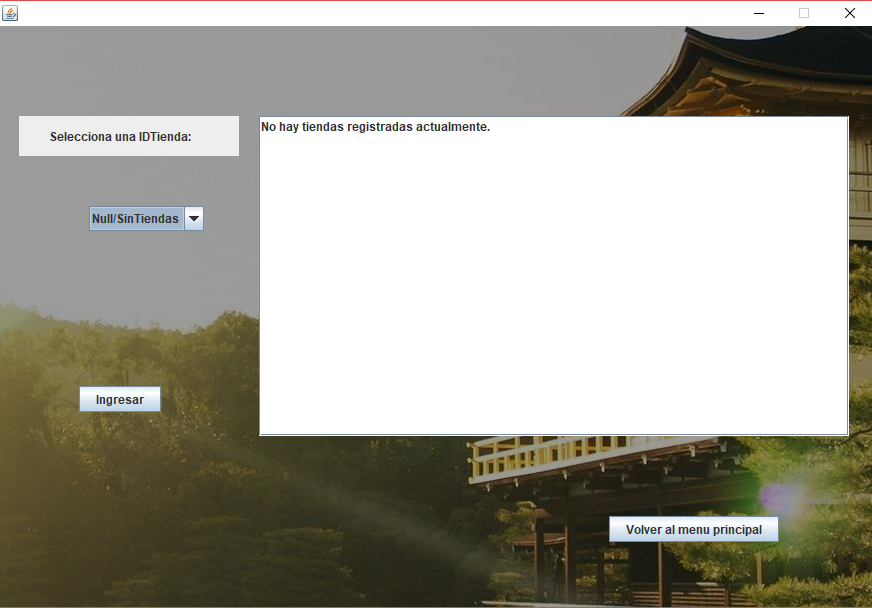


Figura 3.5 Menú tienda.

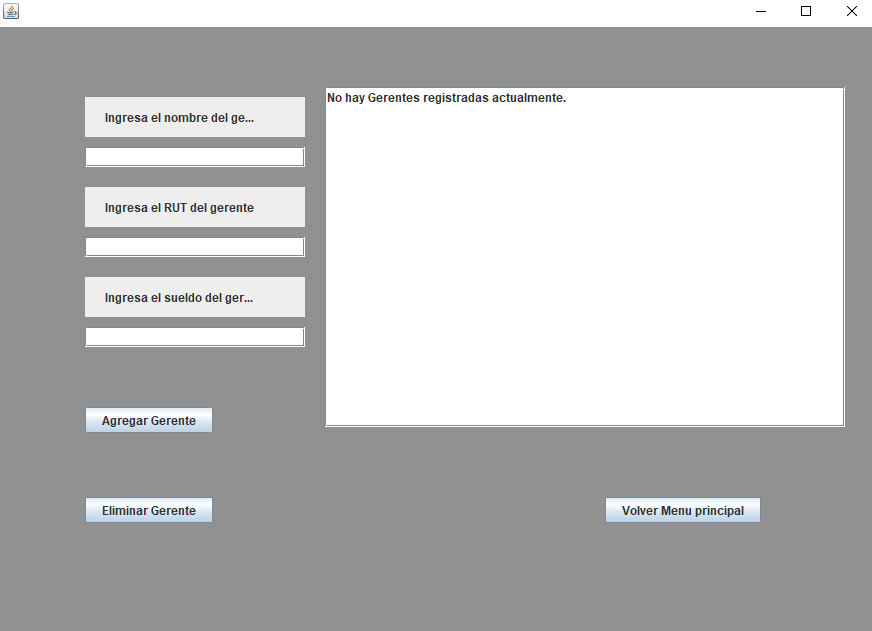


Figura 3.6 Agregar Gerente

**5 Planificación**

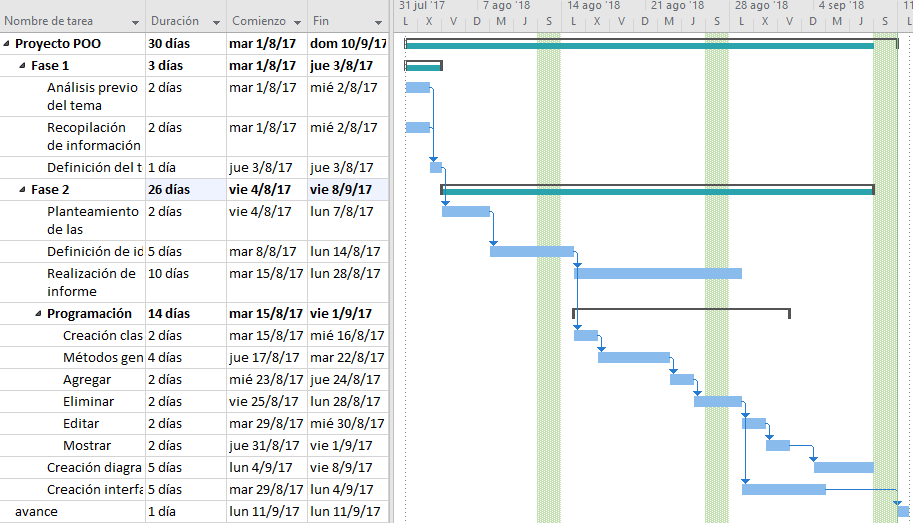


Tabla 1. Carta Gantt Grupal

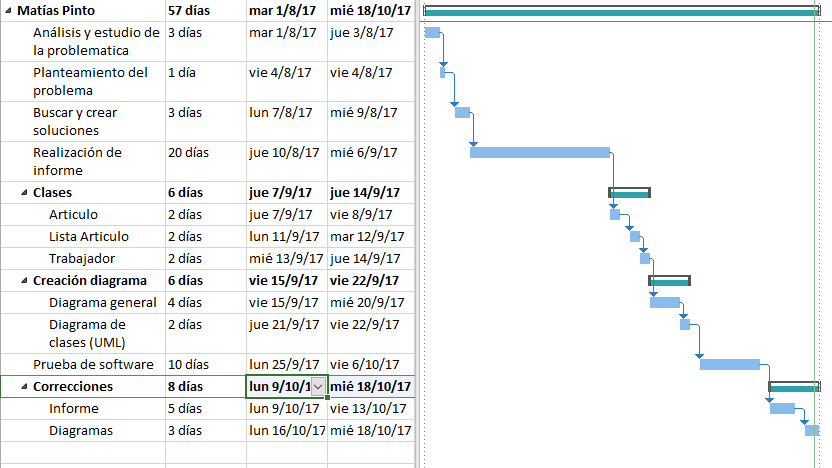


Tabla 1.1 Carta Gantt Matías Pinto

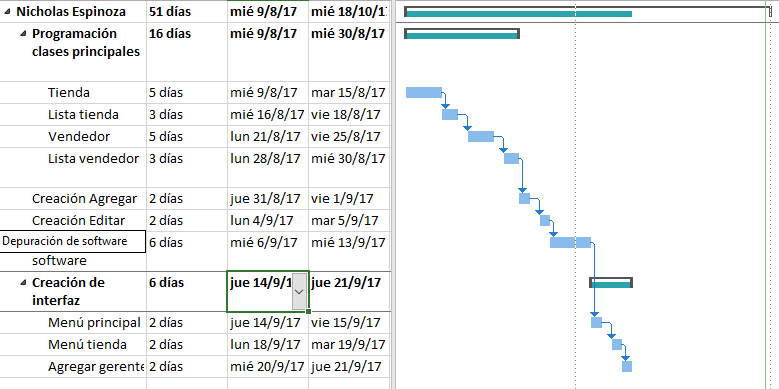


Tabla 1.2 Carta Gantt Nicholas Espinoza

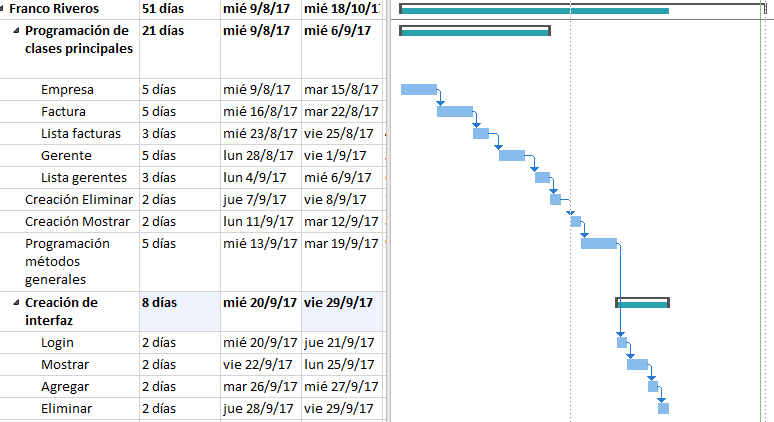


Tabla 1.3 Carta Gantt Franco Riveros

**6 Conclusión**

Como grupo nos vimos muy afectado por el tiempo, ya que topamos con distintos horarios de clases, por ende nos costó mucho el ponernos de acuerdo con todo lo que teníamos que hacer, eso fue lo principal. Asimismo, en trabajos grupales siempre estará la posibilidad de estar en desacuerdo por lo que también fue influyente, pero logramos superarlo.

Por otro lado, con lo investigado y realizado en este proyecto, nos dimos cuenta que la exigencia en trabajos informáticos es muy alta y que la actualización y mejoramiento del software debe estar siempre presente. Hemos obtenido una visión más amplia sobre cómo realizar proyectos, para así no pasar nada por alto y poder cumplir con los objetivos pedidos.

El software que se desarrolló permite llevar un control de datos de manera eficiente, como también protegerlos de manera segura, presentando una interfaz amigable y cómoda para el usuario.

El aprendizaje que nos queda es que tenemos, a como de lugar, organizarnos mejor y con tiempo, para evitar que se forme tiempo muerto al momento de ejecución del proyecto en sí. Y que en este tipo de software debemos fijarnos muy bien en los detalles para no cometer ningún error con datos privados de la empresa en cuestión.