	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:	Edgar Tista García
Asignatura:	Programación Orientada a Objetos.
Grupo:	Grupo 3
No de Práctica(s):	Práctica 5 - 6 – Abstracción y Encapsulamiento, Organización de clases.
Integrante(s):	Edwin Jaret Santiago Díaz
No. de Equipo de cómputo empleado:	38
No. de Lista o Brigada:	
Semestre:	2022 - 2
Fecha de entrega:	23 marzo 2022
Observaciones:	

CALIFICACIÓN: _____

Abstracción y Encapsulamiento, Organización de clases.

Objetivos

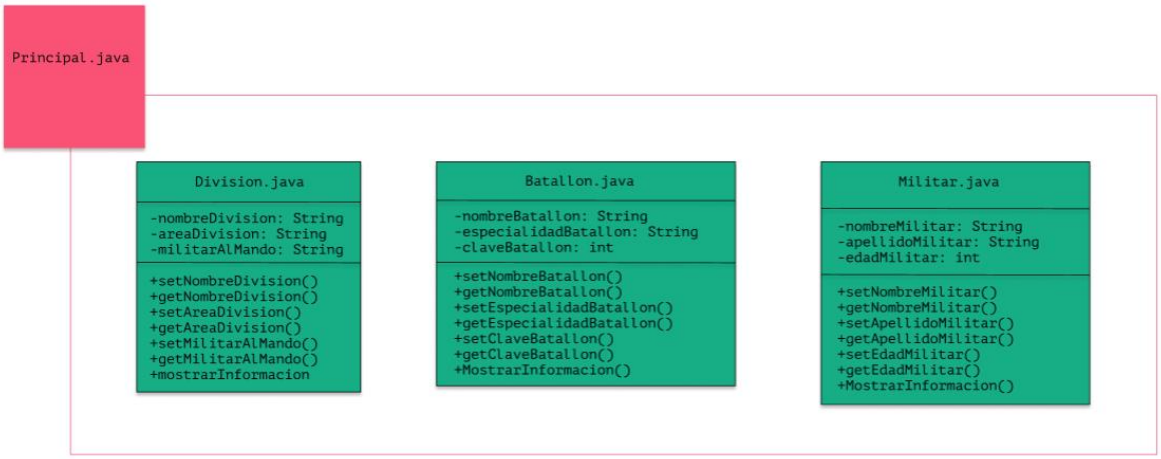
Objetivo (P5): Aplicar el concepto de abstracción para el diseño de clases que integran una solución, utilizando el encapsulamiento para proteger la información y ocultar la implementación.
Objetivo (P6): Organizar adecuadamente las clases según su funcionalidad o propósito bajo un namespace o paquete.
Objetivo de clase: conocer las clases de uso general y sus métodos más importantes y comprender la importancia de su uso como capa de abstracción para el programador.

Desarrollo

Ejercicio 1. Composición de clases

Dentro de la carpeta **Ejercicio1** se encuentran las clases **Batallón**, **División** y **Militar** las cuales se utilizan en la clase **Principal** para crear una división con dos batallones y por cada batallón crear 3 militares. No hay un límite de Divisiones para crear.

La siguiente imagen muestra el diagrama UML:



Las divisiones creadas se guardan en un **ArrayList**. Cada división contiene sus atributos y métodos y un **ArrayList** que guarda los batallones creados. Cada batallón contiene sus atributos y métodos y un **ArrayList** para guardar a los militares creados.

Cuando se ejecuta el programa se le muestra al usuario un menú con dos opciones, la primera opción es para crear una división y la segunda opción es para mostrar la información guardada de las divisiones creadas.

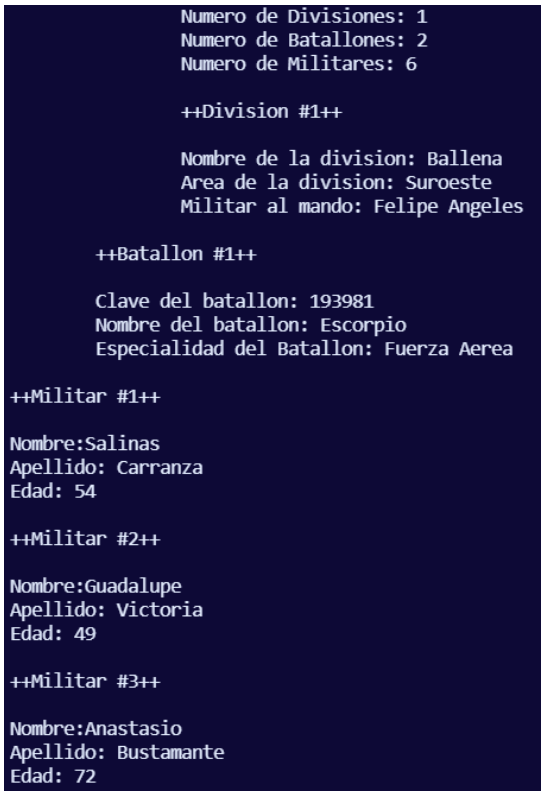
```
**Practica 5-6**

1. Crear una division
2. Ver la informacion de una division
```

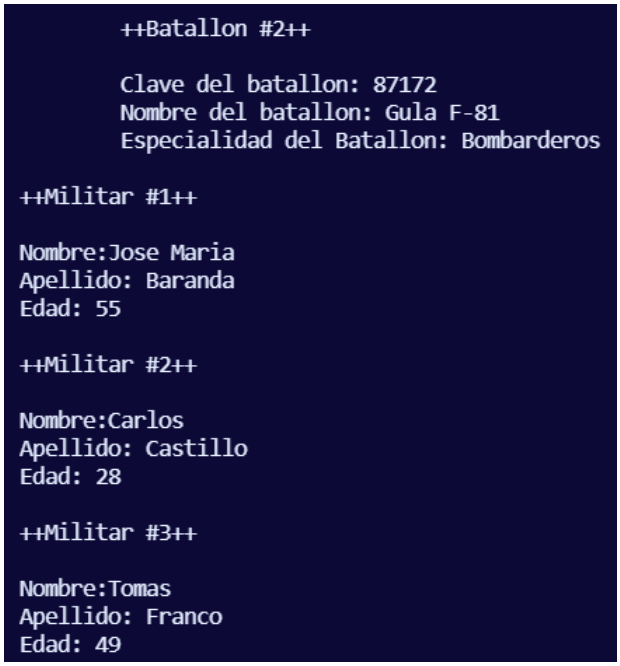
Para este ejemplo, se creó una división con su respectiva información y después de esto se mostrará la información de esta con la segunda opción del menú.

Este es el procedimiento para llenar la división.

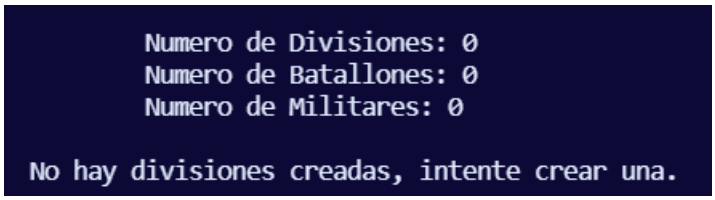
```
        **DIVISION**
Cual es el nombre de la division?
Ballena
Cual es el area de la division?
Suroeste
Quien es el militar al mando?
Felipe Angeles

        **BATALLON #1**
Cual es el nombre del batallon?
Escorpio
Cual es la especialidad del batallon?
Fuerza Aerea
Cual es la clave del batallon?
193981

        **MILITAR #1**
Cual es el nombre del militar?
Salinas
Cual es el apellido del militar?
Carranza
Cual es la edad del militar?
54

        **MILITAR #2**
Cual es el nombre del militar?
Guadalupe
Cual es el apellido del militar?
Victoria
Cual es la edad del militar?
49

        **MILITAR #3**
Cual es el nombre del militar?
Anastasio
Cual es el apellido del militar?
Bustamante
Cual es la edad del militar?
72

        **BATALLON #2**
Cual es el nombre del batallon?
Gula F-81
Cual es la especialidad del batallon?
Bombarderos
Cual es la clave del batallon?
87172

        **MILITAR #1**
Cual es el nombre del militar?
Jose Maria
Cual es el apellido del militar?
Baranda
Cual es la edad del militar?
55

        **MILITAR #2**
Cual es el nombre del militar?
Carlos
Cual es el apellido del militar?
Castillo
Cual es la edad del militar?
28

        **MILITAR #3**
Cual es el nombre del militar?
Tomas
Cual es el apellido del militar?
Franco
Cual es la edad del militar?
49
```

Este es el resultado que imprime en la pantalla al solicitar la información de la división.





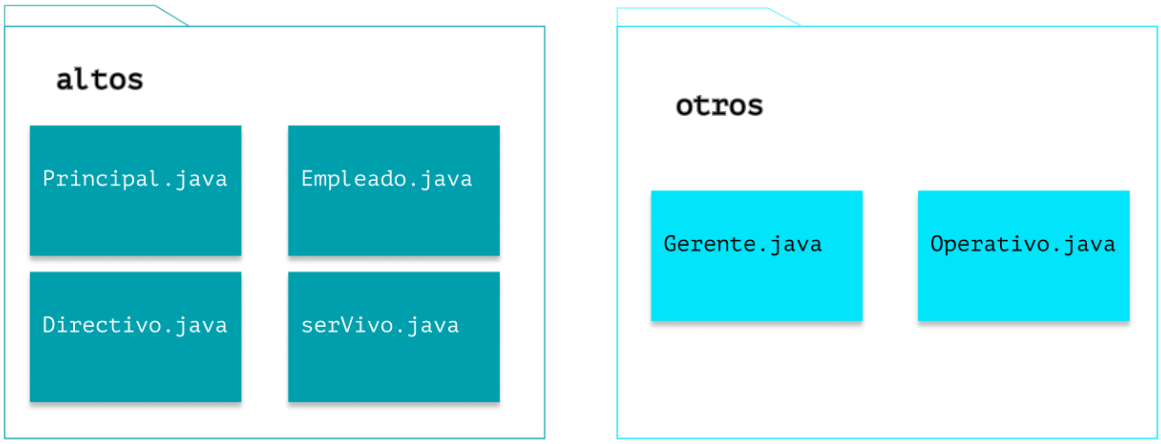
Si fuera el caso de que no se ha creado ninguna división y el usuario escoge la segunda opción, se le notifica que debe crear una división.



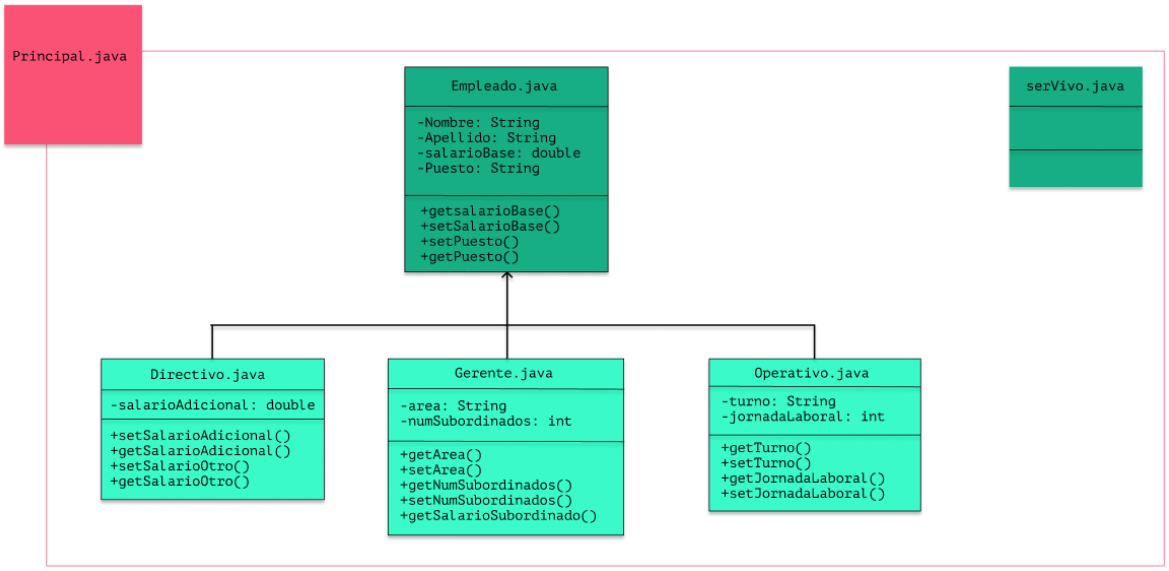
Ejercicio 2. Modificadores de acceso.

En el proyecto **Empleados** proporcionada por el profesor se encuentran las carpetas **build**, **empresa**, **nbproject**, **src**, **test** y los archivos **build** y **manifest.mf**.

Dentro de la carpeta **empresa** se van a encontrar el paquete **altos** que contiene 4 clases (**Directivo**, **Empleado**, **Principal** y **SerVivo**) y el paquete **otros** que contiene 2 clases (**Gerente** y **Operativo**), como se muestra en la imagen siguiente:



La jerarquía de la herencia de este proyecto es la siguiente:



Esto significa que:

- Las clases **Directivo**, **Gerente** y **Operativo** son clases hijas de la clase **Empleado**.
- La clase **serVivo** no hereda a otra clase.
- La clase **Principal** no hereda a otra clase pero en ella se encuentra el método *main()* en donde va a hacer el uso de otras clases.

Los elementos de cada clase son:

- Empleado:
  - Variables privadas de tipo String: Nombre, Apellido, Puesto.
  - Variable privada de tipo double: salarioBase.
  - Método constructor con parámetros Nombre y Apellido.
  - Métodos de acceso para el salario y el puesto.
- Directivo:
  - Es clase hija de la clase Empleado
  - Variable privada de tipo double: salarioAdicional
  - Es el mismo método constructor de la clase Empleado, esto debido al uso del método *super()*.
  - Métodos de acceso para salarioAdicional.
  - Métodos de acceso para asignar el salarioAdicional a un objeto Empleado.

- Gerente:
    - Variable privada de tipo String: area.
    - Variable privada de tipo int: numSubordinados.
    - Mismo método constructor de la clase Empleado, esto debido al uso del método *super()*.
- Operativo:
    - Variable privada de tipo String: turno.
    - Variable privada de tipo int: jornadaLaboral.
    - Mismo método constructor de la clase Empleado, debido al uso del método *super()*.
    - Métodos de acceso para el turno y jornadaLaboral.
- SerVivo:
    - No contiene atributos ni métodos.

En la clase **Principal** se instancia objetos de la clase **Directivo**, **Gerente** y **Operativo** en donde se les asigna un nombre y apellido, un salario base (para el objeto de la clase directivo), un puesto y un numero de subordinados (para el objeto de la clase gerente), un turno y una jornada laboral (para el objeto de la clase operativo).

Este es el resultado de ejecutar el programa:

```
desde operativo 1 5000.0
desde operativo2  5500.0
desde gerente, salario gerente 20000.0
desde gerente salario operativo 5000.0
desde directivo, salario gerente20000.0
desde directivo, salario operaivo 5500.0
```

El directivo de la empresa puede seleccionar los salarios a través de su método *setSalarioOtro()* en donde recibe de parámetro a la persona que se le asigna un salario (debe ser la clase Empleado o una clase hija de esta) y el sueldo. Al objeto/persona que se le asigna el salario contiene una variable privada heredada de la clase Empleado en donde se guarda el salario, para modificar u obtener la información de esta variable se usa los métodos de acceso de la clase **Empleado**.

Si la clase **Principal** se cambiara de ubicación encontrándose en otra carpeta marcaria errores (no encuentra algunas clases, al compilar el comando es erróneo) y estos se solucionarían:

- Al escribir en la primera línea del código el nombre correcto del paquete (en donde se encuentra)
- Importar las clases de los otros paquetes
- En el momento de compilar y ejecutar, escribir el nombre del paquete seguido con el nombre del programa (desde afuera del paquete).

Conclusiones

Los ejercicios de la práctica se completaron en su totalidad, sin embargo, no fueron incluidos los ejemplos de las prácticas.

Para la práctica se implementó el conocimiento sobre colecciones, métodos y modificadores de acceso, análisis de código, herencia y manejo de paquetes.

No tuve errores o conflicto con los ejercicios, pero si me tomó algo de tiempo en analizar qué implementar, cómo implementar y hacer pruebas y errores de código para el primer ejercicio pues la mayor parte del código fue libre aunque, fue de gran ayuda ya que tengo una nueva experiencia en el desarrollo y análisis de programas.

Ver un proyecto creado por otra persona y realizar un análisis sobre sus clases y los elementos que estos incluyen ayuda a mejorar al desarrollo del análisis, a la creatividad para crear diagramas o elementos visuales y si se explica el proyecto (escrito o verbalmente) se refuerza la comprensión que se tiene sobre este.