INFORME

Nombre Alumnos:
- Pannunzio, Mario Nicolás. DNI: 43.825.227
- Quintana, Ricardo José Antonio. DNI: 41.379.027

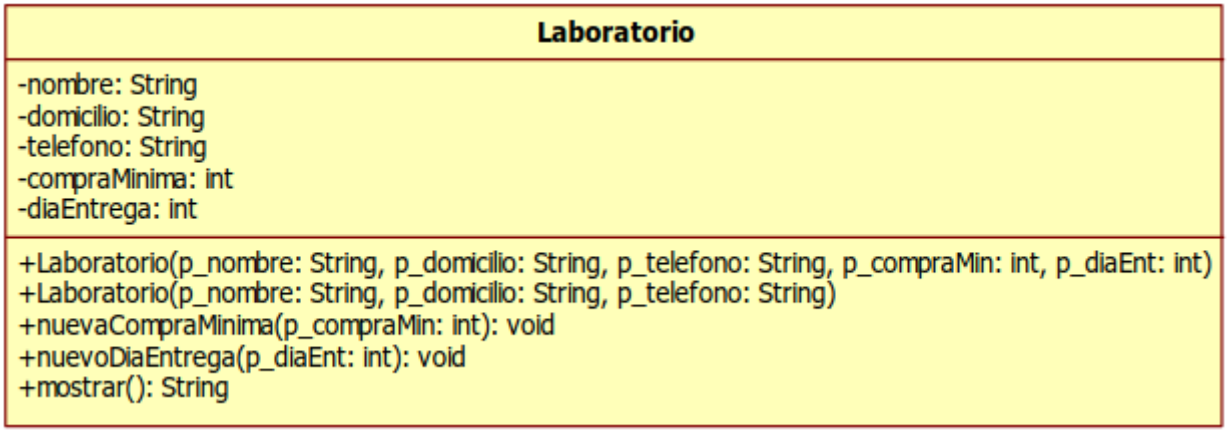
Nombre Profesor:
- Mettini, Aldo.
- Airaldi, Andrea.

Grupo Laboratorio: 3. TP: 2 Fecha de entrega: 24/08/2025

Con el desarrollo de los distintos ejercicios del TP2 pude trabajar con los siguientes conceptos:

- Interpretación de diagramas UML.
- Abstracción
- Encapsulamiento
- Doble encapsulamiento
- Documentación con JavaDoc
- Autorefenciación.
- Sobre carga de métodos.

Ejemplos:



A partir del diagrama UML del ejercicio 2 pude practicar la interpretación de las características de los atributos y métodos, identificando en cada caso si estos son privados o públicos, qué tipo de datos son o retornan, los nombres que deben llevar y también si reciben parámetros y sus nombres y tipos (para los métodos).

En este ejercicio se esta abstrayendo y encapsulando la información de Laboratorios. Abstrayendo porque solo se enfoca en la información que le interesa para realizar las tareas que le interesa, y no se enfoca en otros detalles. Se encapsula porque se agrupan estos datos de interés (atributos) y no quedan visible a simple vista.

```java
//Setters -------------
private void setNombre(String p_nombre){
    this.nombre = p_nombre;
}
private void setDomicilio(String p_domicilio){
    this.domicilio = p_domicilio;
}
private void setTelefono(String p_telefono){
    this.telefono = p_telefono;
}
private void setCompraMinima(int p_compraMin){
    this.compraMinima = p_compraMin;
}
private void setDiaEntrega(int p_diaEnt){
    this.diaEntrega = p_diaEnt;
}
```

Con la implementación de los setters se crea un doble encapsulamiento, los atributos no pueden ser modificados ni por el propio objeto si no es a través de los setters.

```java
/**
 * Constructor con todos los atributos pasados por parametro
 */
public Laboratorio(String p_nombre, String p_domicilio, String p_telefono, int p_compraMin, int p_diaEnt)
{
    this.setNombre(p_nombre);
    this.setDomicilio(p_domicilio);
    this.setTelefono(p_telefono);
    this.setCompraMinima(p_compraMin);
    this.setDiaEntrega(p_diaEnt);
}
/**
 * Constructos con solo 3 parametros, los faltantes se inicializan en cero (0).
 * @param p_nombre
 * @param p_domicilio
 * @param telefono
 */
public Laboratorio(String p_nombre, String p_domicilio, String p_telefono)
{
    this.setNombre(p_nombre);
    this.setDomicilio(p_domicilio);
    this.setTelefono(p_telefono);
    this.setCompraMinima(0);
    this.setDiaEntrega(0);
}
```

En este ejercicio también se trabaja la sobre carga de métodos. En el ejemplo podemos ver como el método constructor esta sobrecargado, son dos métodos que se diferencian por su firma, es decir por el orden, tipo y cantidad de parámetros.

Se evidencia también el concepto de autoreferenciación, los mensajes deben tener un receptor, y se emplea la pseudo variable “this” para indicar que este propio objeto es el receptor del mensaje. Ej: this.setNombre(p_nombre), cuando se quiere instanciar un objeto con el constructor, se envía un mensaje al mismo objeto para que busque el método setNombre con la firma indicada, si se encuentra el método, éste se ejecuta y envía su respuesta.

Por ultimo también se trabajó la documentación con JavaDoc, una cosa que aprendí en particular es que los métodos y variables privadas no aparecen en la documentación, por lo cual vale la pena enfocarse en documentar los métodos públicos, y los privados documentar si es necesario de forma interna.