**Laboratorio #4: Herencia**

***Ejercicio 1: Timbre y alarma***

1. **¿Qué mostrarán las instrucciones?**

Imprimirá en la terminal:

”Ring… Ring“

Seguido de:

“Alarma 1”

“Timbre 2”

1. **¿Por qué sale eso en pantalla?**

* Se imprime la primera línea, ya que se hace un toString de la clase padre.
* La segunda línea “Alarma 1”, se imprime ya que se le hace un override al método de la clase padre, desde la clsae hija “Alarma” el cual cambia lo que se imprime al ser llamado el objeto de dicha clase.
* La úlitma línea, “Timbre 2”, se imprime ya que, al extender una clase, esta posee todos los metodos creados desde la clase Padre y en efecto pueden ser usado sin necesariamente ser declarados o reescritos.

***Ejercicio 2: Timbre y alarma (2da. Ronda)***

1. **¿Qué mostrarán las instrucciones?**

Imprimirá en la terminal:

“Ring… Ring…”

Seguido de:

“alarma 1”

“timbre 1”

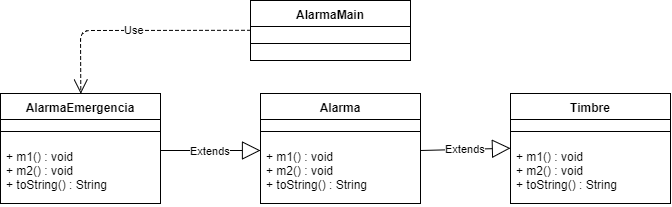
1. **¿Por qué sale eso en la pantalla?**

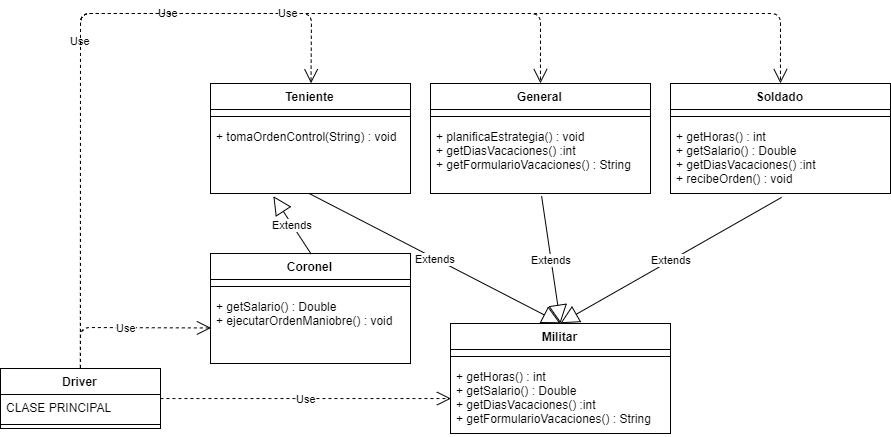
* La primera linea, es un override del método original de la clase padre el cual, ademas de imprimir el toString original del mismo, le concatena tres puntos suspensivos.
* La segunda línea, está llamando al método m1, perteneciente al objeto “Alarma”, el cual hace un override al método original por lo que imprime lo que se le pide.
* La tercera línea se imprime ya que, el método declarado como “m2”, está haciendo un super “sobreponiendo” al método “m1” perteneciente de la clase padre; en efecto se imprime “alarma 1”.

***Ejercicio 3 y 4:***

Tanto el .jar como las clases y el UML, se encuentran en el archivo comprimido. Sin embargo, se adjuntarán los diagramas UML en este documento.

*UML Ejercicio 3*

**

*UML Ejercicio 4*

***¡Puntos extra!***

Para la revisión de los puntos extra, se adjuntará el URL del repositorio en GitHub.

<https://github.com/OJP98/Lab04>