

**Carrera:** INFORMÁTICA APLICADA**Docentes:** Prof. Dr. DIEGO CORSI**Materia:** PROGRAMACIÓN II (2.601)**PRIMER PARCIAL (Límite para la entrega: 3/7/2024 – 8:00)****Ejército Argentino (www.argentina.gob.ar/ejercito)**

Desarrolle un sistema para gestionar la información de los soldados que realizan el servicio militar, teniendo en cuenta que:

- Un **soldado** se define por su **código de soldado** (único), su **nombre** y **apellidos**, y su **graduación**.
- Existen varios **cuarteles**, cada uno se define por su **código de cuartel**, **nombre** y **ubicación**.
- Hay que tener en cuenta que existen diferentes **Cuerpos del Ejército** (Infantería, Artillería, Armada...), y cada uno se define por un **código de Cuerpo** y **denominación**.
- Los soldados están agrupados en **compañías**, siendo significativa para cada una de éstas, el **número de compañía** y la **actividad** principal que realiza.
- Se desea controlar los **servicios** que realizan los soldados (correr, limpiar, barrer...), y se definen por el **código de servicio** y **descripción**.

Consideraciones de diseño:

- Un **soldado pertenece** a un único **cuerpo** y a una única **compañía**, durante todo el servicio militar. A una **compañía** pueden **pertenecer** **soldados** de diferentes **cuerpos**, no habiendo relación directa entre compañías y cuerpos.
- Los **soldados** de una misma **compañía** pueden estar **destinados** en diferentes **cuarteles**, es decir, una **compañía** puede estar **ubicada** en varios **cuarteles**, y en un **cuartel** puede **haber** varias **compañías**. Eso sí, un **soldado** sólo **está** en un **cuartel**.
- Un **soldado realiza** varios **servicios** a lo largo de su paso por el servicio militar. Un mismo **servicio** puede ser **realizado** por más de un **soldado** (con independencia de la compañía), siendo significativa la fecha de realización.
- Al sistema podrán acceder tres tipos de usuarios: **soldados (que sólo lo podrán consultar)**, **suboficiales (que administrarán a los soldados y los servicios)** y **oficiales (que administrarán todo)**.

Para ello:

- Analice los requerimientos anteriores
- Determine los objetos requeridos para implementar ese sistema
- Establezca los atributos que deben tener estos objetos
- Fije los comportamientos que exhibirán estos objetos
- Especifique la forma en que los objetos deben interactuar entre sí para cumplir con los requerimientos del sistema

El sistema deberá utilizar abstracción, encapsulamiento, herencia, polimorfismo y persistencia (no BD).

La E/S del sistema será exclusivamente por consola (no GUI).

Se deberán subir a **GitLab** o **GitHub** el ejecutable (en formato *jar*), el código fuente, la documentación (generada con *javadoc*) y los **diagramas UML de caso-uso, de clases y uno de secuencia** (generados con <http://plantuml.com/es> o <http://www.planttext.com> y grabados en formato *png*).