

**Universidad Nacional de San Agustín**  
**Escuela Profesional de Ingeniería de Sistemas**  
**Fundamentos de Programación II**  
**Practica de Laboratorio 19:**  
**Definición de Clases de Usuario**  
**Herencia y Polimorfismo**

**I**

---

**OBJETIVOS**

- Que el alumno demuestre poder crear “clases definidas por el programador”
- Implementar métodos para las clases definidas por el programador
- Utilizar los mecanismos de Herencia de clases y de Polimorfismo

**II**

---

**ACTIVIDADES**

- En la historia, los ejércitos estaban conformados por diferentes tipos de soldados, que tenían similitudes, pero también particularidades.
- Basándose en la clase Soldado crear las clases Espadachín, Arquero y Caballero. Las tres clases heredan de la superclase Soldado, pero aumentan atributos y métodos, o sobrescriben métodos heredados.
- Los espadachines tienen como atributo particular "longitud de espada" y como acción "crear un muro de escudos" que es un tipo de defensa en particular.
- Los caballeros pueden alternar sus armas entre espada y lanza, además de desmontar (sólo se realiza cuando está montando e implica defender y cambiar de arma a espada), montar (sólo se realiza cuando está desmontado e implica montar, cambiar de arma a lanza y investir). El caballero también puede investir, ya sea montando o desmontando, cuando es desmontado equivale a atacar 2 veces, pero cuando está montando implica a atacar 3 veces.
- Los arqueros tienen un número de flechas disponibles las cuales pueden dispararse y se gastan cuando se hace eso.
- Basándose en los laboratorios anteriores.
- Realizar diagrama de clases de UML.
- Añadir un nuevo tipo de soldado, el lancero. Tiene como atributo particular, "longitud de lanza" y como acción "schiltrom" (como una falange que es un tipo de defensa en particular y que aumenta su nivel de defensa en 1)
- Tendrá 2 Ejércitos que pueden ser constituidos por espadachines, caballeros, arqueros y lanceros (usar una sola estructura de datos donde se almacenen

**todos** los soldados creados de un ejército). Crear una estructura de datos conveniente para cada ejército y para el tablero. Cada ejército tendrá n soldados aleatorios entre 1 y 10. Cada soldado tendrá un nombre autogenerado: Espadachin0X1, Arquero1X1, Caballero2X2, Lancero3X2, etc., un valor de puntos de vida autogenerado aleatoriamente: Espadachín [3..4], Arquero [1..3], Caballero [3..5] y Lancero [1..2], la fila y columna también autogenerados aleatoriamente (no puede haber 2 soldados en el mismo cuadrado) y valores autogenerados para el resto de atributos. Mostrar el tablero, distinguiendo los soldados de ambos ejércitos y los tipos de soldados creados (incluido su nivel de vida). Además, se debe mostrar todos los datos de todos los soldados creados para ambos ejércitos. Además de los datos del soldado con mayor nivel de vida de cada ejército, el promedio de puntos de vida de todos los soldados creados por ejército, los datos de todos los soldados por ejército en el orden que fueron creados y un ranking de poder de todos los soldados creados por ejército (del que tiene más nivel de vida al que tiene menos) usando algún algoritmo de ordenamiento. Finalmente, que muestre qué ejército ganará la batalla (indicar la métrica usada para decidir al ganador de la batalla). Hacerlo programa iterativo.



	A	B	C	D	E	F	G	H	I	J
1							E3			
2		A3		A1						
3							E3			A1
4				L2						
5										
6	C5					L1		L2		
7				E4						
8	C5									
9										
10						E4				A1