

## Ejercitación

**Aclaración:** Antes de implementar cualquier ejercicio, debe desarrollar el diagrama de clases correspondiente.

### Ejercicio 1: Tom y Jerry

Realizar un diagrama de clase de un objeto que represente a un gato. Un gato tiene un **nombre**, una **cantidad de energía** y una **velocidad**. La **energía** inicial de cualquier gato es **ceros** y la **velocidad** es **5 metros/s**.

1. Implementar la clase Gato con los atributos necesarios.
2. Crear los getters y setters para los atributos.
3. Realizar un diagrama de clase de un objeto que represente a un ratón. Un ratón tiene **un nombre y un peso**.
4. Implementar la clase Ratón con los atributos necesarios.
5. Crear los getters y setters para los atributos.
6. Crear en la clase Gato el método: **public void comerRaton(Raton unRaton)**.  
Cada vez que un gato come a un ratón, su energía aumenta el peso del ratón.
7. Crear en la clase Gato el método: **public void comerRaton(List<Raton> ratones)**.  
En este caso, el gato se come una lista de ratones. Su energía aumenta con el peso de cada ratón.
8. Crear en la clase Gato el método: **public Double correr(Integer cantidadMetros)**.  
Cuando el gato corre, su energía disminuye en 0.5 por la cantidad de metros que corrió. El método debe devolver cuánto tardó en recorrer esa distancia (utilizar la velocidad del gato).
9. Crear en la clase Gato el método: **public Boolean meConvieneComerA(Raton unRaton, Integer unaDistancia)**  
Que dado un ratón y una distancia devuelve **true** si la energía que gana por comer al ratón es mayor a la que consume corriendo la distancia o **false** en caso contrario.

### Ejercicio 2: TGI Thursdays

Un restaurante desea poner en marcha un sistema de organización de la limpieza. En el restaurante trabajan empleados de limpieza.

Algunos empleados de limpieza están **encargados de la limpieza general** del restaurante, mientras que otros son **lavadores de vajilla**.

Ambos trabajadores cobran un sueldo fijo de **X frijoles mágicos**, pero los segundos suelen romper vajillas durante su trabajo; estas roturas **son contabilizadas** para luego ser descontadas de su sueldo.

Del empleado de limpieza nos interesa conocer:

- Nombre
- Apellido
- Sueldo mensual

Además, nos interesa conocer de los empleados lavadores de vajilla, cuánta vajilla rompieron. Es muy importante tener en cuenta que para dichos empleados, al cobrar su sueldo se le descontará **\$5 frijoles mágicos** por cada vajilla que haya roto.

**Aclaración:** Para el restaurante es indistinto si el empleado de limpieza rompe un plato, un vaso o algún otro utensilio, sólo le importa saber la cantidad de vajilla que rompió y que tiene que reponer.

1. Realizar un **diagrama de clase** que modele lo propuesto anteriormente. Recordar que el objetivo del sistema es que el empleado de limpieza cobre su sueldo a fin de mes.
2. **Programar el sistema** teniendo en cuenta todas las consideraciones solicitadas por la empresa.
3. Crear una clase Main y **probar que el sistema funcione** como es esperado.

### Ejercicio 3: World Of Digital House (WODH)

Estamos modelando un novedoso juego de rol en donde participan distintos personajes. Los personajes de este juego están divididos en dos mundos, el mundo de los héroes y el mundo de las criaturas.

- **Mundo de los héroes:**  
Todos los héroes de este juego tienen un **nivel de experiencia** y son capaces de **luchar contra una criatura**. Sin embargo, existen dos tipos de héroes:
  - **Magos:** Los magos son invencibles y tienen la **habilidad de resucitar a un guerrero** cuyo nivel de experiencia sea menor o igual al suyo.
  - **Guerreros:** Los guerreros además tienen un **nivel de vida**.
- **Mundo de las criaturas:**  
Todas las **criaturas** de este juego tienen **nivel de ataque**. Sin embargo, existe un tipo de criatura, los dragones, que además de tener las mismas características de cualquier criatura, tienen la **habilidad de volar y de lanzar fuego**. El héroe es el que tiene la habilidad de luchar contra una criatura. El resultado de la lucha entre un héroe y una criatura dependerá de las siguientes reglas.
  - Si un guerrero lucha contra una criatura cuyo nivel de ataque es menor que su nivel de experiencia, entonces el guerrero resulta ganador y aumenta su experiencia en 1. En este caso, debe imprimir por pantalla **"El guerrero ganó"**.
  - Si un guerrero lucha contra una criatura cuyo nivel de ataque es mayor o igual que su nivel de experiencia, entonces el guerrero resulta perdedor y automáticamente se quedará sin vida. En este caso, debe imprimir por pantalla **"El guerrero perdió y se quedó sin vida"**.
  - Si un guerrero con vida igual a 0 no puede luchar. En este caso, debe imprimir por pantalla **"El guerrero no puede luchar y deberá esperar su resurrección"**.
  - Si un mago lucha contra una criatura cuyo nivel de ataque es menor que su nivel de experiencia, entonces el mago resulta ganador y aumenta su experiencia en 2. En este caso, debe imprimir por pantalla **"El mago ganó"**.
  - Si un mago lucha contra una criatura cuyo nivel de ataque es mayor o igual que su nivel de experiencia, entonces el mago resulta perdedor y disminuye su experiencia en 1. En este caso, debe imprimir por pantalla **"El mago perdió"**.

**Adicional:** Considerar que la lucha sea aleatoria. Es decir, en vez de considerar el nivel de experiencia del héroe y el nivel de ataque de la criatura. Cada uno elegirá un valor aleatorio y el que haya sacado el máximo será el ganador.