
Practica 1

Mascota Virtual



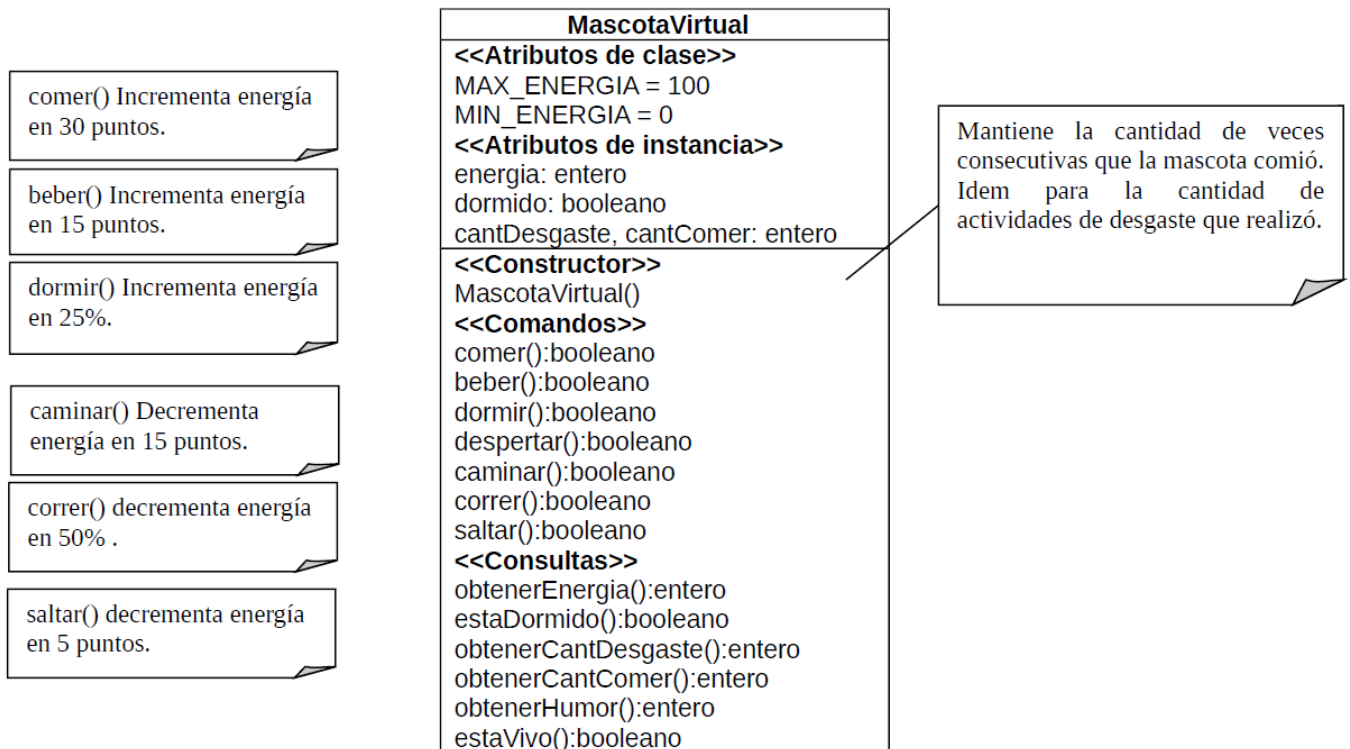
Es un juego donde el objetivo es mantener con vida a una “mascota virtual” la mayor cantidad de tiempo posible. Cada mascota virtual se identifica por medio de su nombre. Pueden realizar diferentes acciones, las que aumentan o disminuyen su nivel de energía, el cual nunca puede disminuir por debajo de 0 ni superar el máximo de 100. Cuando el nivel de energía es cero, la mascota lamentablemente abandona la existencia terrenal.

Las operaciones que incrementan el nivel de energía son *comer*, *beber* y *dormir*. Cuando duerme recupera el 25% de su energía. Mientras duerme no puede hacer ninguna otra acción, a menos que antes se lo despierte. Las acciones que decrementan el nivel de energía de la mascota son *caminar*, *correr* y *saltar*, si bien lo hacen en distinto grado. Ninguna mascota puede realizar más de tres acciones de desgaste en forma consecutiva. Al cuarto intento, ignora la orden dada y directamente se pone a dormir. Cuando se vuelva a despertar, podrá volver a realizar a lo sumo tres acciones de desgaste. Asimismo, si come o bebe cinco veces seguidas morirá al instante de indigestión.

La mascota manifiesta diferentes grados de humor, los que expresa mediante un valor numérico que va de 1 a 5, siendo 1 el más infeliz y 5 el más contento. El humor de la mascota depende de su nivel de energía como sigue:

Energía	Humor
Entre 0 y 20	1
Entre 21 y 40	2
Entre 41 y 60	3
Entre 61 y 80	4
Entre 81 y 100	5

La clase Mascota Virtual se modela con le diagrama:



Todos los comandos retornan true si se ejecutaron correctamente o false en caso contrario (si la mascota está muerta o superó la cantidad de veces que puede realizar la acción de forma consecutiva).

- Implementar en Java la clase MascotaVirtual con todos los atributos y comportamiento necesario.
- Implementar una clase Tester que ejecute un ciclo completo de la vida de estas mascotas: nacimiento, actividades varias, muerte.

Objetivos de la Práctica: Interpretar un diagrama de modelado. Ilustrar los conceptos de objeto, atributo de instancia y de clase. Constructores y Métodos. Establecer la funcionalidad de los métodos. La clase Tester con valores ingresados por el usuario.