

Mascota virtual grupo 3

Vicente Neira, Gabriel Lara, Diego Moyano, Alvaro Soto

1 Introducción

Durante el siguiente informe se dará a conocer la manera en como se pudo desarrollar un programa que simula a una mascota virtual mediante la utilización de programación orientada a objetos. Para esto se crearán distintas clases, cada una con métodos y atributos, que interactuarán entre si para poder componer a una gran entidad como la mascota.

2 Descripción de la Solución

La solución de la problemática se desarrollará en distintas etapas, en las que en cada una se desarrollarán distintas clases. En la primera etapa se inicializará la clase main y se creará la clase Mascota, en la segunda etapa se hará el diseño y se implementarán los Items en su respectiva clase "Items", posteriormente, en la tercera etapa , se creará la clase inventario, clase en la que se almacenarán las diversas instancias de la clase Items, finalmente en la cuarta etapa se inicializará la clase inventario por medio de la lectura de un archivo entregado, que contendrá información de los items a guardar en el inventario.

A continuación se hará un breve resumen de las características de cada clase utilizada para la creación e interacción de la mascota.

2.1 Clase Main

Desde esta clase se invocarán las demás clases que interactuarán entre si, considerando tambien el paso del tiempo por cada acción, además proveerá al usuario de una interfaz con la cual sera capaz de interactuar con la mascota, dandole de comer, medicina o incluso jugar con ella.

2.2 Clase Mascota

Clase en la que se almacenarán los atributos de la mascota virtual asi como la edad,salud,energia y felicidad. Ademas contiene métodos que podrán ser invocados por la clase main y que modificarán sus propios atributos.

2.3 Clase Inventario

Es en donde se almacenarán los objetos con los cuales se modificarán los atributos anteriormente definidos de la mascota virtual, cada objeto tendrá un efecto distinto sobre los atributos dependiendo de su identificador.

2.4 Clases Item, Comida, Medicina, Juguete

Estas clases definirán las propiedades específicas de cada ítem y cómo afectarán específicamente a los atributos de la mascota. Cada vez que un ítem sea utilizado, la clase Main invocará la Clase Inventario para aplicar estos cambios en los atributos la Clase Mascota.

3 Diagrama UML

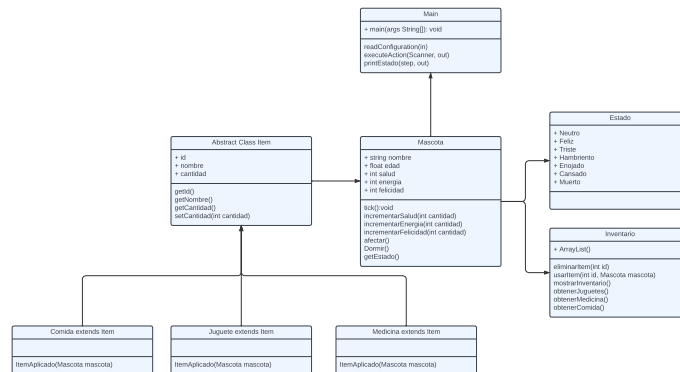


Figure 1: Diagrama UML de las clases del proyecto.

4 Dificultades y Soluciones

A continuación se presentarán las dificultades que como equipo se encontraron durante el desarrollo de la tarea y las respectivas soluciones que se le dieron a las problemáticas.

4.1 Problemas con el formato

Como grupo no teníamos experiencia del formato en el que se desarrolla el programa, el modelo de etapas era confuso en un principio, por lo que tuvimos que adaptar las clases que ya teníamos construidas para poder adaptarlas al formato solicitado. Además desconocíamos de la herramienta utilizada para el testeado del código, por lo que también probar el correcto funcionamiento de este fue laborioso.

4.2 Inexperiencia con el lenguaje

Uno de los integrantes de nuestro equipo desconocía parcialmente el lenguaje en el que se programó esta tarea, afortunadamente el resto del equipo estaba familiarizado con Java, por lo que el integrante que desconocía el programa aprendió rápidamente gracias a su propio estudio y la disposición del equipo en ayudarlo.

4.3 Gestión del tiempo

Debido a problemas relacionados con la organización como equipo para desarrollar la tarea, terminamos encontrándonos con obstáculos que no teníamos previstos cercanos a la fecha límite de entrega, por lo que nos faltó tiempo para un desarrollo más prolijo del código, afortunadamente llevábamos la suficiente cantidad de código escrito como para poder sortear estas dificultades y entregar un desarrollo completo de la tarea.