Tienda RPG

Luis Carlos Araya Mata
Escuela de Computación
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
luistec2309@estudiantec.cr

Sebastián Díaz Obando
Escuela de Computación
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
sebastian.diaz@estudiantec.cr

***Resumen*—El paradigma de la programación orientada a objetos ha sido desarollado con amplitud a lo largo de la historia de la computación. Debido a esta razón, la importancia del mismo en el mundo laboral actual es crucial. El realizar una Tienda RPG de dinámica de compra y venta por medio de la programación en este paradigma corresponde a un ejemplo perfecto de una implementación al mismo al crear una aplicación.**

***Palabras clave*—clase, método, atributos, interfaz, objeto, json, software, programación, git, API.**

INTRODUCCIÓN

A. Planteamiento del problema

El presente proyecto tiene como objetivo la realización de una aplicación que contenga las funcionalidades de una típica "Tienda RPG". Para que esto sea posible, un requisito del proyecto fue implementar el paradigma de la programación orientada a objetos junto con la utilización del lenguaje de programación Java.

Seguidamente, las funcionalidades presentes en la aplicación corresponden a la posibilidad de poder comprar,vender,equipar y desequipar ítems. El jugador posee una cantidad de dinero inicial para poder utilizar las funcionalidades citadas. La información que determina el nombre, el precio de compra y de venta proviene de una API. Se implementó una rutina específica para poder traducir el archivo de formato JSON(Javascript Object Notation), a un objeto en el cual se pudiese manipular y gestionar la información requerida para el diseño de los ítems. Existe un personaje en el cual los ítems se pueden equipar y desequipar. Estas últimas dos funcionalidades modifican los stats que posee el personaje, los cuales son:

- Salud
- Ataque
- Defensa
- Fortuna
- Fuerza
- Dinero

A raíz de las funcionalidades citadas, la interfaz gráfica a diseñar debe facilitar el uso de estas, siendo así de agrado e intuitiva para el usuario. De esta manera, la aplicación debe de poder procesar las transacciones y acciones realizadas por el usuario incluyendo todo tipo de errores comunes que pueden llevar al colapso de la aplicación o a tener un "bug".

I. DISEÑO

A continuación, se explicarán las diferentes áreas que se involucran al realizar un diseño de software para la aplicación.

A. Diagrama de clases

Se utilizó UML(Lenguaje unificado modelado) para generar el diagrama correspondiente a las clases requeridas de la aplicación.

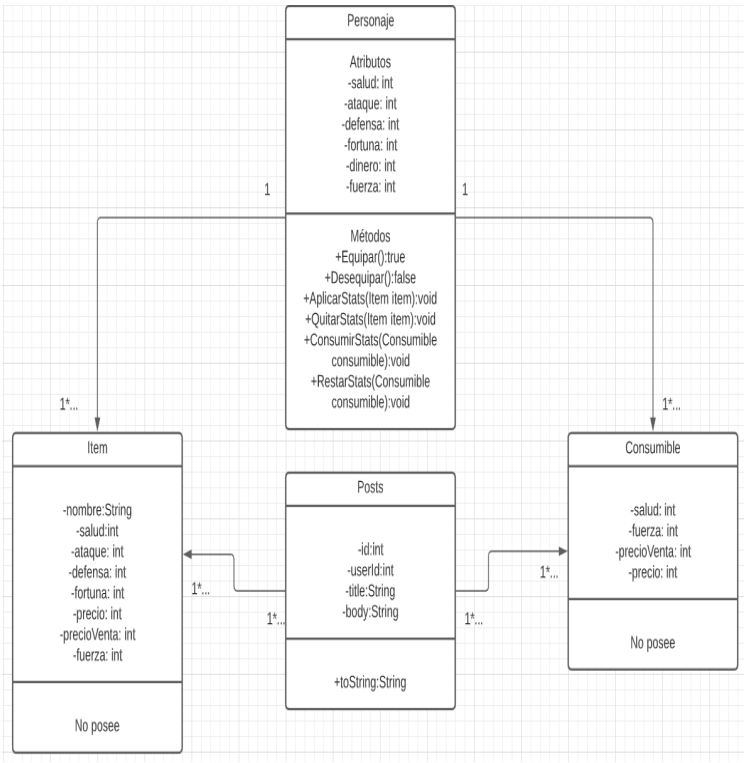

Fig. 1. Imagen correspondiente al diagrama de las clases involucradas en la aplicación.

B. División de Actividades

Luis Carlos Araya Mata:

- Realizar el diseño de software.
- Pasar datos entre las interfaces.
- Creación de la interfaz gráfica de Inicio y sus funciones de ingresar nombre junto con las validaciones respectivas.
- Implementar el API.

- Creación de las clases y los métodos respectivos.
- Complementar el JSON.
- Función de venta y compra de items.
- Validaciones y funciones principales.

Sebastián Díaz Obando:

- Analizar requerimientos de software tanto funcionales como no funcionales.
- Creación de la interfaz gráfica de Héroe y villano.
- Comunicación entre interfaces.
- Implementación de los widgets de las ventanas gráficas.
- Creación de las clases y los métodos respectivos.
- Creación de la ventana gráfica de la tienda.
- Función de venta y compra de items.
- Preview de cada ítem.
- Validaciones y funciones principales.

C. Cronograma

Primera Semana:

- Realizar el diseño de software.
- Creación de interfaz de inicio.
- Implementar el API.
- Creación de la interfaz héroe y villano.
- Realizar la conversión del JSON.

Segunda Semana:

- Creación de las clases y sus métodos.
- Comunicación entre las diferentes interfaces.
- Implementar los widgets a la interfaz.
- Detalles gráficos y validaciones de datos.

Tercera Semana:

- Creación de la ventana Tienda.
- Funciones de compra y venta de items.
- El preview de cada item.
- Validaciones principales.
- Solución de errores.
- Documentación del proyecto.

Cuarta Semana:

- Periodo de Testing de la aplicación.
- Resolución de problemas que surjan dentro del Testing.

II. Métodos

En esta sección se presenta la descripción detallada de todas las pautas y procesos que se siguieron para la realización de la aplicación. Los mismos están separados en diferentes secciones que corresponden a los diferentes ámbitos que hay a la hora de realizar un proyecto de creación de software.

A. Desarollo de Software y Workflow

El modelo de desarrollo de software utilizado fue el "Cascada". Con respecto al control de versiones, la utilización de un repositorio git en la plataforma GitHub junto con la herramienta Git Kraken, fueron cruciales para tener un desarrollo controlado de la aplicación. Para llevar un control de las tareas se utilizó el sistema Kanban y la pizarra de proyecto que provee GitHub.

B. Diseño del Software

Se recurrió a una reunión del equipo de trabajo para poder diseñar en conjunto la aplicación. El modelo de desarrollo de software escogido fue el de "Cascada". Esto se debe a que el equipo de trabajo no poseía con anterioridad experiencia en el lenguaje de programación solicitado. Después del análisis de requerimientos de software, se dio el proceso de investigación de todos los aspectos del proyecto que no fueran conocidos a fondo con anticipación.

C. Lenguaje de Programación e IDE

El lenguaje de programación implementado fue Java. La versión del JDK utilizado fue: jdk-11.0.8 . El IDE corresponde a Apache NetBeans IDE 12.1. El tipo de proyecto corresponde a una "Java Application". No se utilizó ningún framework. Se utilizaron 3 archivos .jar externos con el fin de poder hacer posible el "parse" del archivo JSON. Los nombres de los archivos .jar implementados son:

- jackson-annotations-2.11.1
- jackson-core-2.11.1
- jackson-databind-2.11.1

D. Interfaz Gráfica

El tema de la misma está ambientado en el marco medieval y fantástico. Los colores, nombres de etiquetas e imágenes implementadas son alusivos al tema. A la hora de construirla, se utilizó el editor gráfico que posee el IDE ya integrado, por lo cual el arrastrar, modificar y quitar de los widgets fue bastante dinámico e intuitivo.

E. Arquitectura del proyecto

Debido al paradigma utilizado, se implementó un módulo por cada clase de java que se necesitó. Se siguió pasos con cuidado para poder cumplir con los principios de evasión de "Olores de Software",esto con el fin de que el código fuese lo más limpio posible. Se trabajó con la indicación de preferencia personal por parte del equipo de trabajo de que cada función dentro del código, debía tener solo una responsabilidad, esto con el fin de poder aprovechar el concepto de estas y evitar la repetición del código.

F. Control de Versiones

Se utilizó la plataforma propuesta por el profesor (GitHub), para llevar el control de versiones del archivo. Se tuvo el apoyo de la herramienta de Git Kraken para poder ejecutar los "pulls" y los "push". Todo el proyecto está contenido en un repositorio Git.

G. Periodo de Pruebas

En este periodo el equipo de trabajo se empeñó en probar la aplicación para asegurarse de que no hubiera ningún tipo de errores que llevaran al colapso a la aplicación. Para este proceso se trabajó con el ideal de simular como si se fuese un jugador que no haya participado del desarrollo de la aplicación, y empezar a tratar de encontrar "bugs". En el momento en que se encontrara uno, se retoma la sesión de producción con el objetivo de encontrar una solución al error.

CONCLUSIÓN

Con este proyecto se tenía el propósito de aprender las bases del lenguaje de programación Java , además conocer más acerca de las interfaces gráficas del lenguaje y su implementación. Al finalizar el proyecto se cumplieron con esos propósitos satisfactoriamente, luego de terminado se inició el periodo de testing donde se descubrió algunas validaciones que mejoren el trabajo , algunas de ellas fueron la validación de solo poder equipar un item una única vez, separar los articulos consumibles de los normales para así mejorar la calidad del software y se realizaron de igual manera mejoras gráficas al programa.

Uno de los mayores retos en este trabajo era la conexión con el API, para la cual se utilizaron distintos tutoriales y recursos encontrados en la web , gracias a ello se logró hacer la conexión correctamente y así trabajar con el JSON , se utilizó una API escrita en latín debido a la temática del proyecto se trabajó con esta ya que sería de utilidad para la aplicación.

Para la coordinación del proyecto se utilizó la herramienta de github y Git Kraken las cuales permiten llevar en orden y con documentación cada uno de los cambios realizados al programa lo que es de gran ayuda para mejorar la calidad del trabajo y la comunicación. Estas herramientas analizan los repositorios en tiempo real y notifican de cambios en la aplicación por lo que se facilita que el documento siempre esté en su versión más reciente y no generar archivos secundarios de basura. Además se utilizó el método Kanban que consiste en escribir todos los requerimientos y pasos a seguir para realizar el proyecto estos se colocan en tres columnas, por hacer, en proceso y terminada. Este método mejoró increíblemente el orden en el proyecto ayudando a que se concluya el programa de la mejor manera.

REFERENCIAS

[1]Oracle. (2020). How to Use Labels. Volumen (1). Disponible en :https://docs.oracle.com/javase/tutorial/uiswing/components/label.html. (2020, 7 ,21).

[2]NetBeans. (2012, 01, 7). API Design. Volumen (1). Disponible en: http://wiki.netbeans.org/API$_{D}esign.(2020, 7, 19)$.

[3]JSON-API. (2015, 05, 29). Implementations. Volumen (1.0). Disponible en: https://jsonapi.org/implementations/ (2020, 7, 20).

[4]J.G. Nieva Paredes. (2016, 04, 9). Interfaz Gráfica de Usuario con Netbeans. Volumen (2.0). Disponible en: https://dcodingames.com/interfaz-grafica-de-usuario-con-netbeans/ (2020, 7, 25).

[5]N. Ha Minh. (2019, 06, 06). JRadioButton basic tutorial and examples. Volumen (1). Disponible en: https://www.codejava.net/java-se/swing/jradiobutton-basic-tutorial-and-examples (2020, 7, 24).

[6]P. Samsotha. (2014, 05, 20). Interfaz Gráfica de Usuario con Netbeans. Volumen (1). Disponible en: https://stackoverflow.com/questions/23767763/jlist-adding-and-removing-items-netbeans (2020, 7, 25).

[7]M. Callejón Berenguer. (2019, marzo, 22). Aprende a utilizar el método toString() en java. Volumen (1). Disponible en: https://javautodidacta.es/metodo-tostring-java/ (2020, 7, 25).

[8]T. Zezula y A.Stashkova. (-). Using Oracle Java SE Embedded Support in NetBeans IDE. Volumen (1). Disponible en: https://netbeans.org/kb/docs/java/javase-embedded.html(2020, 7, 19).