**Objetivos**

**Unidad 3: Estructuras Lineales Enlazadas**

Al finalizar esta unidad, el estudiante estará en capacidad de:

OE5.1 Utilizar estructuras enlazadas de objetos para modelar grupos de atributos no primitivos de tamaño flexible.

OE5.2 Escribir los algoritmos necesarios para manipular estructuras lineales que almacenan sus elementos enlazándolos entre ellos.

**Preparación**

* Lea cuidadosamente el enunciado, la documentación suministrada y cada uno de los puntos que debe desarrollar antes de empezar su desarrollo. Pregunte a su profesor cualquier duda respecto al enunciado o a los requerimientos funcionales que debe desarrollar.
* El trabajo debe ser realizado individualmente.
* El trabajo será entregado en la fecha y hora establecida en Moodle.
* Usted debe utilizar git para manejar el versionamiento de su trabajo desarrollado localmente y manejar como repositorio remoto **privado**, un repositorio en GitHub o un proyecto GitLab. Su trabajo debe ser gestionado con git desde el inicio del desarrollo, los commit deben hacerse regularmente, así como los push al remoto. Esto se verificará con las fechas de los commits. En el momento de la fecha de entrega máxima, usted debe hacer público su repositorio. Recuerde las convenciones de nombre y estructura de directorios indicadas en esta [presentación de Git](https://drive.google.com/open?id=1RmergYAoyEvs-Q8g6gklipoKqeHyRhWNW8N77ewTQDQ).

**Enunciado**

Naruto es un manga escrito e ilustrado por Masashi Kishimoto y su adaptación al anime es dirigida por Hayato Date. La obra narra la historia de un ninja adolescente hiperactivo, posteriormente adulto llamado Naruto Uzumaki, quien aspira a convertirse en Hokage, con el propósito de ser reconocido como alguien importante dentro de su aldea y entre sus amigos.

La universidad tiene un proyecto de crear un videojuego y para ello le ha solicitado a usted que le ayude a construir algunas de las funcionalidades. Para esto es importante saber que todos los personaje de la Naruto shippuden tiene un puntaje para determinar su poder, por ejemplo: Naruto: 899, Naruto sennin: 1200, Naruto kyubi mode: 3.000m, Naruto sabio de los 6 caminos: 4.200, Naruto sabio de los 6 caminos Fusionado con kurama:14.500, Sasuke: 900, Sasuke susano’O: 1.900, Sasuke sharingan eternal: 2.300, Kakachi: 1.400, Yiraiya: 3.000, kiba: 740, akamaru: 230, konohamaru: 101, tsunade: 890, Neji: 830, etc.

Del juego se desea tener clanes, por cada clan se tienen los personajes y de ellos se sabe su nombre, su personalidad, la fecha de creación del personaje, su poder y las técnicas usadas para este personaje. De las técnicas nos interesa saber el nombre de la técnica y el factor que influye en el poder de cada personaje, también es importante tener en orden ascendentemente todos las técnicas de un personaje por el factor. Se debe tener en cuenta que el nombre del clan, del personaje y de la técnica no se repitan. El inversionista del proyecto ha solicitado que los personajes sean implementados usando una lista doblemente enlazada y las técnicas una lista sencilla.

En su programa usted deberá:

1. Realizar el CRUD de cada una de las clases del sistema (del original en inglés: Create, Read, Update and Delete) y realizar la respectiva persistencia.
2. Implementar y utilizar los tres métodos de ordenamiento clásicos: burbuja, selección e inserción.
3. Implementar y utilizar búsqueda clásicas.
4. Utilizar la interface Comparable.
5. Utilizar la interface Comparator.

Ya que en el programa hay que hacer diferentes ordenamientos y búsquedas, para cumplir con los requisitos anteriores, usted utilizará un algoritmo de ordenamiento y búsqueda en un caso y otro algoritmo ordenamiento búsqueda en otro caso, implementando todos los vistos hasta la fecha.

**Entregables.**

**1.** Requerimientos Funcionales y no funcionales.

**2.** Diagrama de clases de modelo y de la interfaz del usuario (no generado automáticamente).

**3.** Implementación completa de todos los requerimientos en Java.

**4.** Tabla de trazabilidad de requerimientos vs métodos (tabla con una columna de los requerimientos, tal que, por cada requerimiento se indica en la columna siguiente todos los métodos que contribuyen a resolverlo).

**5.** Investigue e incluya en su reporte de laboratorio qué es trazabilidad en desarrollo de software.

**Fecha de Entrega Máxima:** martes 24 de septiembre de 2019 a las 11:59 AM a través de Moodle. El laboratorio debe trabajarse y entregarse **individualmente**. Lo que usted debe entregar de su trabajo es la url de su repositorio en GitHub o proyecto en GitLab. Recuerde que el repositorio o proyecto debe ser privado durante el desarrollo del laboratorio y hacerse público solo en el momento justo de la entrega máxima indicada aquí.

Desarrollo

Clases: --------

Atributos: -------

Requerimientos: ------

R No Funcionales: ------

|  |  |
| --- | --- |
| **Nombre** | **R.F. 1. Agregar clan** |
| **Resumen** | El usuario debe ingresar los datos del clan que desee crear y este se agrega a un arraylist de clanes |
| **Entradas** | |
| * Nombre del clan | |
| **Resultados** | |
| Se agrega el clan a el arraylist. | |
| **Nombre** | **R.F. 2. Agregar personaje** |
| **Resumen** | El usuario debe ingresar los datos del personaje que desee crear para agregarlo a la lista de personajes |
| **Entradas** | |
| * Nombre del personaje * Clan al que pertenece * Puntaje * Personalidad * Fecha de creación * Poder * Técnicas | |
| **Resultados** | |
| Se agrega el personaje a la lista. | |
| **Nombre** | **R.F. 3. Agregar técnica** |
| **Resumen** | El usuario debe ingresar los datos de la técnica que desee agregar a un personaje. |
| **Entradas** | |
| * Nombre de la técnica * Factor * Nombre del personaje al que va a pertenecer | |
| **Resultados** | |
| Se agrega la técnica a la lista del personaje ingresado. | |
| **Nombre** | **R.F. 4. Mostrar los clanes** |
| **Resumen** | Se muestran en pantalla los clanes que están guardados en el sistema hasta el momento |
| **Entradas** | |
|  | |
| **Resultados** | |
| Se muestra el arraylist de clanes | |
| **Nombre** | **R.F. 5. Mostrar personajes** |
| **Resumen** | Se muestran en pantalla los personajes que están guardados en un clan que debe especificar el usuario |
| **Entradas** | |
| * Nombre del clan del que se quieren ver los personajes | |
| **Resultados** | |
| Se muestra la lista de personajes | |
| **Nombre** | **R.F. 6. Mostrar técnicas** |
| **Resumen** | Se muestran en pantalla las técnicas del personaje que especifique el usuario |
| **Entradas** | |
| * Nombre del personaje * Nombre del clan al que pertenezca el personaje | |
| **Resultados** | |
| Se muestra la lista de técnicas. | |
| **Nombre** | **R.F. 7. Eliminar clan** |
| **Resumen** | El usuario debe ingresar el nombre del clan que desee eliminar y este se quita del arraylist de clanes |
| **Entradas** | |
| * Nombre del clan | |
| **Resultados** | |
| Se elimina el clan especificado | |
| **Nombre** | **R.F. 8. Eliminar personajes** |
| **Resumen** | El usuario debe ingresar el nombre del personaje que desee eliminar y el nombre del clan al que este pertenezca; este se quita de la lista de personajes |
| **Entradas** | |
| * Nombre del clan * Nombre del personaje | |
| **Resultados** | |
| Se elimina el personaje especificado | |
| **Nombre** | **R.F. 9. Eliminar técnicas** |
| **Resumen** | El usuario debe ingresar el nombre del clan, personaje al que pertenezca y nombre de la técnica que desee eliminar; esta se quita de la lista de técnicas |
| **Entradas** | |
| * Nombre del clan * Nombre del personaje * Nombre de la técnica | |
| **Resultados** | |
| Se elimina la técnica especificada | |
| **Nombre** | **R.F. 10. Ordenar los clanes** |
| **Resumen** | Se ordenan los clanes ascendentemente, según el nombre, mediante uno de los tres métodos de ordenamiento (burbuja, inserción o selección) |
| **Entradas** | |
|  | |
| **Resultados** | |
| El arraylist queda ordenado ascendentemente | |
| **Nombre** | **R.F. 11. Ordenar los personajes** |
| **Resumen** | Se ordenan los personajes ascendentemente, según el nombre, mediante uno de los tres métodos de ordenamiento (burbuja, inserción o selección) |
| **Entradas** | |
|  | |
| **Resultados** | |
| La lista queda ordenada ascendentemente | |
| **Nombre** | **R.F. 12. Ordenar las técnicas** |
| **Resumen** | Se ordenan las técnicas ascendentemente, según el factor, mediante uno de los tres métodos de ordenamiento (burbuja, inserción o selección) |
| **Entradas** | |
|  | |
| **Resultados** | |
| La lista queda ordenada ascendentemente | |
| **Nombre** | **R.F. 13. Realizar persistencia** |
| **Resumen** | Mediante serialización, el programa hace persistir los últimos datos ingresados al sistema |
| **Entradas** | |
|  | |
| **Resultados** | |
| Se hacen persistir los últimos datos y quedan guardados en un archivo .dat | |
| **Nombre** | **R.F. 14. Buscar un personaje en el sistema** |
| **Resumen** | El usuario debe ingresar el nombre del personaje que desee encontrar y el clan al que este pertenezca, el programa muestra en pantalla el personaje buscado o un error si este no esta guardado en el sistema |
| **Entradas** | |
| * Nombre del clan * Nombre del personaje | |
| **Resultados** | |
| Se muestra el personaje buscado | |

|  |  |
| --- | --- |
| **Nombre** | **R.N.F. 1. Repetición del nombre de los clanes** |
| **Resumen** | El nombre de los clanes no puede estar repetido (no debe haber más de un clan con el mismo nombre) |
| **Nombre** | **R.N.F. 2. Repetición del nombre de los personajes** |
| **Resumen** | El nombre de los personajes no puede estar repetido (no debe haber más de un personaje con el mismo nombre) |
| **Nombre** | **R.N.F. 3. Repetición del nombre de las técnicas** |
| **Resumen** | El nombre de las técnicas no puede estar repetido (no debe haber más de una técnica con el mismo nombre) |
| **Nombre** | **R.N.F. 4. Uso de una lista doblemente enlazada en los personajes** |
| **Resumen** | Los personajes deben ser agregados a una clase que sea una lista doblemente enlazada |
| **Nombre** | **R.N.F. 5. Uso de una lista enlazada simple en las técnicas** |
| **Resumen** | Las técnicas deben ser agregadas a una clase que sea una lista enlazada simple |