**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 susanno: 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 la estrategiaer
4. de búsqueda clásica (secuencial).
5. Utilizar la interface Comparable.
6. 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**

**1.**

|  |  |
| --- | --- |
| **Nombre** | **R.F. 1. Registrar un clan nuevo** |
| **Resumen** | El usuario debe ingresar el nombre del clan, para que registrar los datos en el sistema. |
| **Entradas** | |
| -Nombre del clan | |
| **Resultados** | |
| Clan registrado. | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 2. Registrar un ninja nuevo** |
| **Resumen** | El usuario debe ingresar el nombre del clan al cual se le asignará el nuevo ninja. Nombre, Fecha de creación, personalidad y poder del ninja, para registrar los datos en el sistema. |
| **Entradas** | |
| -Nombre del clan  -Nombre del ninja  -Fecha de creación del ninja  -Personalidad del ninja  -Poder del ninja | |
| **Resultados** | |
| Ninja registrado. | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 3. Registrar un jutsu nuevo** |
| **Resumen** | El usuario debe ingresar el nombre del clan y del ninja al cual se le asignará el nuevo jutsu. Nombre y poder del jutsu, para registrar los datos en el sistema. |
| **Entradas** | |
| -Nombre del clan  -Nombre del ninja  -Nombre del jutsu  -Poder del jutsu | |
| **Resultados** | |
| Jutsu registrado | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 4. Eliminar un clan** |
| **Resumen** | El usuario debe ingresar el nombre del clan para que el programa lo busque y lo elimine consigo a los ninjas que tenga ligados. |
| **Entradas** | |
| -Nombre del clan | |
| **Resultados** | |
| Clan eliminado. | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 5. Eliminar un ninja** |
| **Resumen** | El usuario debe ingresar el nombre del clan del ninja, y el nombre del ninja para que el programa busque el ninja y lo elimine. |
| **Entradas** | |
| -Nombre del clan  -Nombre del ninja | |
| **Resultados** | |
| Ninja eliminado. | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 6. Eliminar un jutsu** |
| **Resumen** | El usuario debe ingresar nombre del clan del ninja, del ninja y finalmente el nombre del jutsu para que el programa busque el jutsu y lo elimine. |
| **Entradas** | |
| -Nombre del clan  -Nombre del ninja  -Nombre del jutsu | |
| **Resultados** | |
| Jutsu eliminado. | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 7. Leer archivos** |
| **Resumen** | El programa debe poder leer archivos serializados para que su contenido pueda ser utilizado dentro del problema |
| **Entradas** | |
| -Archivos serializados | |
| **Resultados** | |
| Contenido de archivos cargado en el sistema | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 8. Guardar cambios en el programa** |
| **Resumen** | El programa debe poder guardar cambios del modelo del problema para que estos datos puedan ser utilizados de nuevo en el futuro |
| **Entradas** | |
| -Archivos serializados | |
| **Resultados** | |
| Estado final del sistema es guardado en archivos | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 9. Buscar un clan** |
| **Resumen** | El programa debe poder buscar un clan según las entradas dadas por el usuario y bajo que críterio quiere hacer la búsqueda |
| **Entradas** | |
| -Nombre del clan | |
| **Resultados** | |
| Clan encontrado | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 10. Buscar un ninja** |
| **Resumen** | El programa debe poder buscar un ninja de un clan según las entradas dadas por el usuario y bajo que críterio quiere hacer la búsqueda |
| **Entradas** | |
| -Nombre del ninja | |
| **Resultados** | |
| Ninja encontrado | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 11. Buscar un jutsu** |
| **Resumen** | El programa debe poder buscar un jutsu de un ninja de un clan según las entradas dadas por el usuario y bajo que críterio quiere hacer la búsqueda |
| **Entradas** | |
| -Nombre del jutsu | |
| **Resultados** | |
| Jutsu encontrada | |

|  |  |
| --- | --- |
| **Nombre** | **R.N.F.1 Clanes únicos en el programa** |
| **Resumen** | **Cada clan debe tener un nombre diferente** |

|  |  |
| --- | --- |
| **Nombre** | **R.N.F.2 Ninjas únicos en cada clan del programa** |
| **Resumen** | **Cada ninja de un clan debe tener un nombre diferente** |