**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 las dos estrategias de búsqueda clásicas (secuencial).
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.

**Requerimientos funcionales**

|  |  |
| --- | --- |
| **Nombre:** | R. #1. Registrar un clan |
| **Resumen:** | Permite al usuario registrar un clan al sistema |
| Entradas: | Nombre |
|  |  |
| Resultados: | Clan registrado con éxito |
|  | No fue posible registrar el clan |

|  |  |
| --- | --- |
| **Nombre:** | R. #2. Registrar un personaje en un clan |
| **Resumen:** | Permite al usuario registrar en un clan especifico un personaje |
| Entradas: | Nombre, personalidad, fecha creación, poder, nombre del clan |
|  |  |
| Resultados: | Personaje registrado con éxito |
|  | No fue posible registrar el personaje |

|  |  |
| --- | --- |
| **Nombre:** | R. #3. Registrar una técnica a un personaje en especifico |
| **Resumen:** | Permite al usuario registrarle una técnica a un personaje en especifico |
| Entradas: | Nombre del personaje, nombre de la técnica, poder |
|  |  |
| Resultados: | Técnica registrada con éxito |
|  | No fue posible registrar la técnica |

|  |  |
| --- | --- |
| **Nombre:** | R. #4. Eliminar un clan |
| **Resumen:** | Permite al usuario eliminar un clan en específico teniendo el nombre de este |
| Entradas: | Nombre del clan |
|  |  |
| Resultados: | Clan eliminado con éxito |
|  | No fue posible eliminar el clan |

|  |  |
| --- | --- |
| **Nombre:** | R. #5. Eliminar un personaje |
| **Resumen:** | Permite al usuario eliminar un personaje en específico teniendo el nombre de este |
| Entradas: | Nombre del personaje |
|  |  |
| Resultados: | Personaje eliminado con éxito |
|  | No fue posible eliminar el personaje |

|  |  |
| --- | --- |
| **Nombre:** | R. #6. Eliminar una técnica |
| **Resumen:** | Permite al usuario eliminar una técnica de un personaje en específico teniendo el nombre de este y el nombre de la técnica |
| Entradas: | Nombre del personaje, nombre de la técnica |
|  |  |
| Resultados: | Técnica eliminada con éxito |
|  | No fue posible eliminar la técnica |

|  |  |
| --- | --- |
| **Nombre:** | R. #7. Eliminar un clan |
| **Resumen:** | Permite al usuario eliminar un clan en específico teniendo el nombre de este |
| Entradas: | Nombre del clan |
|  |  |
| Resultados: | Clan eliminado con éxito |
|  | No fue posible eliminar el clan |

|  |  |
| --- | --- |
| **Nombre:** | R. #8. Modificar el nombre de un clan |
| **Resumen:** | Permite al usuario modificar el nombre de un clan en específico. |
| Entradas: | Nombre del clan actual, nuevo nombre del clan |
|  |  |
| Resultados: | Nombre del clan modificado con éxito |
|  | No fue posible cambiar el nombre del clan |

|  |  |
| --- | --- |
| **Nombre:** | R. #9. Modificar el nombre de un personaje |
| **Resumen:** | Permite al usuario modificar el nombre de un personaje en especifico |
| Entradas: | Nombre del personaje actual, nuevo nombre del personaje |
|  |  |
| Resultados: | Nombre del clan personaje con éxito |
|  | No fue posible cambiar el nombre del personaje |

|  |  |
| --- | --- |
| **Nombre:** | R. #10. Modificar el poder de un personaje |
| **Resumen:** | Permite al usuario modificar el poder de un personaje en especifico |
| Entradas: | Nombre del personaje, nuevo poder del personaje |
|  |  |
| Resultados: | Poder del personaje modificado con éxito |
|  | No fue posible cambiar el poder del personaje |

|  |  |
| --- | --- |
| **Nombre:** | R. #11. Modificar el nombre de una técnica |
| **Resumen:** | Permite al usuario modificar el nombre de una técnica en especifico |
| Entradas: | Nombre del personaje que tiene la técnica, nombre de la técnica, nuevo nombre de la técnica |
|  |  |
| Resultados: | Nombre de la técnica modificado con éxito |
|  | No fue posible cambiar el nombre de la técnica |

|  |  |
| --- | --- |
| **Nombre:** | R. #12. Modificar el poder de una técnica |
| **Resumen:** | Permite al usuario modificar el poder de un personaje en especifico |
| Entradas: | Nombre del personaje que tiene la técnica, nombre de la técnica, poder nuevo de la técnica |
|  |  |
| Resultados: | Poder de la técnica cambiado con éxito |
|  | No fue posible cambiar el poder de la técnica |

|  |  |
| --- | --- |
| **Nombre:** | R. #13. Ordenar los clanes |
| **Resumen:** | El programa debe tener la opción de ordenar los clanes según un criterio |
| Entradas: |  |
|  |  |
| Resultados: | Clanes ordenados correctamente |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #14. Ordenar los personajes |
| **Resumen:** | El programa debe tener la opción de ordenar los personajes según un criterio |
| Entradas: |  |
|  |  |
| Resultados: | Personajes ordenados correctamente |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #15. Ordenar las técnicas |
| **Resumen:** | El programa debe tener la opción de ordenar las técnicas según un criterio |
| Entradas: |  |
|  |  |
| Resultados: | Técnicas ordenadas correctamente |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #16. Buscar un clan |
| **Resumen:** | El programa debe tener la opción de que se busque un clan por determinado criterio |
| Entradas: |  |
|  |  |
| Resultados: | Clan pedido buscado con éxito |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #17. Buscar un personaje |
| **Resumen:** | El programa debe tener la opción se busque un personaje por determinado criterio |
| Entradas: |  |
|  |  |
| Resultados: | Personaje pedido buscado con éxito |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #18. Buscar una técnica |
| **Resumen:** | El programa debe tener la opción de buscar una técnica por determinado criterio |
| Entradas: |  |
|  |  |
| Resultados: | Técnica pedida buscada con éxito |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #19. Serializar |
| **Resumen:** | El programa debe serializar la información del programa |
| Entradas: |  |
|  |  |
| Resultados: | Serialización realizada con éxito |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #20. Deserialización |
| **Resumen:** | El programa debe deserializar los datos al cargar el programa |
| Entradas: |  |
|  |  |
| Resultados: | Deserialización realizada con éxito |
|  |  |

**Documentación pruebas**

**Objetivo:** Comprobar que el método añadir un personaje, agrega correctamente el personaje pasado por parámetro teniendo en cuenta como no habían antes personajes al agregar este el primer personaje debe ser distinto de null

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | addCharacter(String,String ); | El escenario inicializa un clan con:  *Nombre = Prueba*. | boolean esperado = true;  boolean resultado = true; | Verdadero, el método agrega correctamente al personaje |
|
|
|
|
|
|
|
|
|
|

**Objetivo:**  Comprobar que el método añadir una técnica a un personaje, añade correctamente la técnica pasada por parámetro al personaje cuyo nombre es también pasado por parámetro, se verifica teniendo en cuenta que el personaje ya ingresado previamente no tenía ninguna técnica por lo tanto la primera técnica después del método debe ser distinta de null

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | addTechnique(String, String); | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | boolean esperado = true;  boolean resultado = true; | Verdadero, el método agrega correctamente la tecnica |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** Comprobar que el método realmente muestra el nombre de todos los personajes que se encuentran en el clan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | showNameCharacters(); | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | String esperado = “PersonajeTest1 \n”;  String resultado = “PersonajeTest1 \n”; | Verdadero, el método muestra correctamente el nombre de los personajes que están en el club |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** Comprobar que el método elimina correctamente al personaje cuyo nombre se pasa por parámetro, para esto eliminamos el único personaje que tiene el club por lo tanto el primer personaje del club debe ser igual a null

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | deleteCharacter(String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el método elimina correctamente el personaje del club. |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** Comprobar que el busca y encuentra correctamente al personaje pedido por parámetro para saber si este existe o no

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | findCharacter(String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el método encuentra correctamente el personaje pedido. |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** Comprobar que el método muestra correctamente el nombre de las técnicas de un personaje en especifico cuyo nombre es pasado por parámetro

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | deleteCharacter(String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.*  Por último inicializa una técnica con:  *Nombre = Prueba, Poder = 500.* | String esperado = “Prueba \n”;  String resultado = “Prueba \n”; | Verdadero, el método muestra correctamente el nombre de las técnicas. |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** Comprobar que el método elimina correctamente una técnica especifica pasado el nombre de esta por parámetro, de un personaje cuyo nombre también se pasa por parámetro. Como el personaje únicamente tiene una técnica, este método se dice que funciona si la primera técnica es igual a null

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | deleteTechnique(String, String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.*  Por último inicializa una técnica con:  *Nombre = Prueba, Poder = 500.* | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el método elimina correctamente la técnica. |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** Comprobar que el método cambia correctamente el nombre del personaje

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | changeCharacterName(String, String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | String esperado = “Cambio”;  String resultado = “Cambio”; | Verdadero, el método cambia correctamente el nombre del personaje. |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** Comprobar que el método cambia correctamente el poder de un personaje

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | changeCharacterPower(String, int) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | int esperado = 1;  int resultado = 1; | Verdadero, el método cambia correctamente el poder del personaje. |

**Objetivo:** Comprobar que el método cambia correctamente el nombre de la técnica

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | changeTechniqueName(String, String, String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.*  Por último, inicializa una técnica con:  *Nombre = Prueba, Poder = 500.* | String esperado = “Cambio”;  String resultado = “Cambio”; | Verdadero, el método cambia correctamente el nombre de la técnica. |

**Objetivo:** Comprobar que el método cambia correctamente el poder de la técnica

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | changeTechniquePower(String, String, String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.*  Por último, inicializa una técnica con:  *Nombre = Prueba, Poder = 500.* | Int esperado = 1;  Int resultado = 1; | Verdadero, el método cambia correctamente el poder de la técnica. |

**Objetivo:** Comprobar que el método ordena correctamente la lista de tecnicas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | ordenarTecnicas() | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.*  Por último, inicializa una técnica con:  *Nombre = Prueba, Poder = 500.* | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el método ordena correctamente las técnicas. |

**Objetivo:** Comprobar que el método ordena correctamente los personajes por el nombre

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | ordenarCharacterName() | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el método ordena correctamente los personajes por el nombre. |

**Objetivo:** Comprobar que el método busca correctamente la técnica pedida cuyo nombre pasa por parámetro.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | searchTecnique(String, String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.*  Por último, inicializa una técnica con:  *Nombre = Prueba, Poder = 500.* | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el busca correctamente la técnica pedida. |

**Objetivo:** Comprobar que el método busca correctamente el personaje pedido.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Clan | searchTecnique(String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el busca correctamente el personaje pedido. |

**Objetivo:** Comprobar que el método verifica correctamente que no existe otro clan con el mismo nombre.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | verificarNombreClan(String); | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | Boolean esperado = false;  Boolean resultado = false; | Verdadero, el verifica correctamente que no exista ningún otro clan con el mismo nombre |

**Objetivo:** Comprobar que el método verifica correctamente que no existe otro clan con el mismo nombre.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | verificarNombrePersonaje(String, String); | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | Boolean esperado = false;  Boolean resultado = false; | Verdadero, el verifica correctamente que no exista ningún personaje con el nombre de un clan. |

**Objetivo:** Comprobar que el método añade correctamente un clan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | addClan(Clan); | El escenario inicializa únicamente con la relación al modelo: clanManagerTest | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el método agrega correctamente un clan. |

**Objetivo:** Comprobar que el método agrega un personaje al club especificado por parámetro.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | addCharacter(String, Character) | El escenario inicializa un clan con:  *Nombre = Prueba*. | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el método agrega correctamente un personaje al club especificado. |

**Objetivo:** Comprobar que el método elimina correctamente el personaje especificado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | deleteCharacter(String); | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el método elimina correctamente el personaje especificado. |

**Objetivo:** Comprobar que el método muestra correctamente las técnicas de los personajes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | showTechniques(String); | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | String esperado = “Prueba \n”;  String resultado = “Prueba \n” | Verdadero, el método muestra correctamente las técnicas. |

**Objetivo:** Comprobar que el método elimina correctamente una técnica especificada, teniendo en cuenta que el clan previamente asignado tiene únicamente un personaje con una sola técnica, por lo tanto si la primera técnica es igual a null después de ejecutar el método, el método funciona correctamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | deleteTechnique(String, String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | String esperado = “Prueba \n”;  String resultado = “Prueba \n” | Verdadero, el método muestra correctamente las técnicas. |

**Objetivo:** Comprobar que el método cambia correctamente el nombre del clan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | changeNameClan(String, String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | String esperado = “Cambio”;  String resultado = “Cambio”; | Verdadero, el método cambia correctamente el nombre del clan |

**Objetivo:** Comprobar que el método cambia correctamente el nombre del personaje.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | changeCharacterName(String, String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | String esperado = “Cambio”;  String resultado = “Cambio”; | Verdadero, el método cambia correctamente el nombre del personaje. |

**Objetivo:** Comprobar que el método cambia correctamente el poder de una técnica.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | changeTechniquePower(String, int) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | Int esperado = 1;  Int resultado = 1; | Verdadero, el método cambia correctamente el poder de la técnica. |

**Objetivo:** Comprobar que el método cambia correctamente el poder de una técnica.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | changeTechniqueName(String, String) | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | String esperado = “Cambio”;  String resultado = “Cambio”; | Verdadero, el método cambia correctamente el nombre de la técnica. |

**Objetivo:** Comprobar que el método ordena correctamente las tecnicas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | ordenarTecnicas() | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el método ordena correctamente las técnicas. |

**Objetivo:** Comprobar que el método ordena correctamente los personajes teniendo en cuenta su nombre.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | ordenarNombreCharacter() | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el método ordena correctamente los personajes por el nombre. |

**Objetivo:** Comprobar que el método ordena correctamente los clanes por el nombre.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | ordenarClanNombre() | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el método ordena correctamente los clanes teniendo en cuenta su nombre. |

**Objetivo:** Comprobar que el método muestra correctamente la técnica pedida.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| ClanManager | searchTechnique() | El escenario inicializa un clan con:  *Nombre = Prueba*. También inicializa un personaje con: *Nombre =PersonajeTest1, Personalidad = Prueba, Fecha creación = 2019/09/09, Poder = 500.* | Boolean esperado = true;  Boolean resultado = true; | Verdadero, el método busca correctamente la técnica pedida |

**Trazabilidad**

|  |  |  |
| --- | --- | --- |
| **Requerimientos funcionales** | **Clase** | **Método** |
|
| 1. Registrar un clan | ClanManager | +addClan() |
| 2. Registrar un personaje en un clan | ClanManager | +addCharacter() |
| Clan | +addCharacter() |
| 3. Registrar una tecnica a un personaje en especifico | ClanManager | +addTechnique() |
| Clan | +addTechnique() |
| Character | +addTechnique() |
| 4. Eliminar un clan | ClanManager | +deleteClan() |
| 5. Eliminar un personaje | ClanManager | +deleteCharacter() |
| Clan | +deleteCharacter() |
| 6. Eliminar una tecnica | ClanManager | +deleteTechnique() |
| Clan | +deleteTechnique() |
| Character | +deleteTechnique() |
| 7. Modificar el nombre de un clan | ClanManager | +changeClanName() |
| 8. Modificar el nombre de un personaje | ClanManager | +changeCharacterName() |
| Clan | +changeCharacterName() |
| 9. Modificar el poder de un personaje | ClanManager | +changeCharacterPower() |
| Clan | +changeCharacterPower() |
| 10. Modificar el nombre de una tecnica | ClanManager | +changeTechniqueName() |
| Clan | +changeTechniqueName() |
| Character | +changeTechniqueName() |
| 11. Modificar el poder de una tecnica | ClanManager | +changeTechniquePower() |
| Clan | +changeTechniquePower() |
| Character | +changeTechniquePower() |
| 12. Ordenar los clanes | ClanManager | +ordenarClanNombre() |
| 13. Ordenar los personajes | ClanManager | +ordenarPersonajeNombre() |
| Clan | +ordenarNombreCharacter() |
| 14. Ordenar las tecnicas | ClanManager | +ordenarTechniques() |
| Clan | +ordenarTechniques() |
| Character | +ordenarTechniques() |
| 15. Buscar un clan | ClanManager | +searchClan() |
| 16. Buscar un personaje | ClanManager | +searchCharacter() |
| Clan | +searchCharacter() |
| 17. Buscar una tecnica | ClanManager | +searchTenique() |
| Clan | +searchTenique() |
| Character | +searchTenique() |
| 18. Serializar | ClanManager | +serialization() |
| 19. Deserializar | ClanManager | +deserialization() |