**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:

* 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.
* Implementar y utilizar los tres métodos de ordenamiento clásicos: burbuja, selección e inserción.
* Implementar y utilizar búsqueda clásica.
* Utilizar la interface Comparable.
* 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: | **RF1** Hacer la creación clanes |
| Resumen: | Poder agregar un clan nuevo dentro del sistema |
| Entradas: | -Nombre del clan |
| Salida: | Se agrega un clan dentro del sistema del juego de la universidad |

|  |  |
| --- | --- |
| Nombre: | **RF2** Agregar un personaje dentro de los clanes |
| Resumen: | Agrega a los personajes dentro de un clan especifico |
| Entradas: | -Nombre del personaje  -La personalidad del personaje  -La fecha de creación del personaje  -El poder del personaje |
| Salida: | Se agrega al personaje dentro de un determinado clan |

|  |  |
| --- | --- |
| Nombre: | **RF3** Agregar una técnica a un determinado personaje |
| Resumen: | Agrega técnicas a un determinado personaje de Naruto |
| Entradas: | -Nombre de la técnica  -El influenciador del poder |
| Salida: | Se agrega una técnica a un determinado personaje |

|  |  |
| --- | --- |
| Nombre: | **RF4** Serializar los clanes, de los personajes y de las técnicas |
| Resumen: | Poder serializar a cada uno de los objetos dentro del sistema |
| Entradas: | Ninguna |
| Salida: | Serializar los objetos del sistema |

|  |  |
| --- | --- |
| Nombre: | **RF5** Mostrar la información de los clanes |
| Resumen: | Mostrar la información de los clanes |
| Entradas: | Ninguna |
| Salida: | Mostrar la información de los clanes de todos sus atributos |

|  |  |
| --- | --- |
| Nombre: | **RF6** Mostrar la información de los personajes |
| Resumen: | Mostrar la información de todos los personajes |
| Entradas: | -Nombre del clan |
| Salida: | Mostrar la información de todos los personajes dependiendo del clan |

|  |  |
| --- | --- |
| Nombre: | **RF7** Mostrar la información de las técnicas de los personajes |
| Resumen: | Mostrar la información de las técnicas dependiendo de un determinado personaje |
| Entradas: | -Nombre del clan  -Nombre del personaje |
| Salida: | Mostrar la información de las técnicas con todos sus atributos |

|  |  |
| --- | --- |
| Nombre: | **RF8** Modificar el nombre del personaje |
| Resumen: | Permitir modificar el nombre del clan |
| Entradas: | -Nombre del clan  -Nombre nuevo a modificar |
| Salida: | Modifica el nombre del clan |

|  |  |
| --- | --- |
| Nombre: | **RF9** Modificar el nombre del personaje |
| Resumen: | Poder modificar el nombre del personaje determinado |
| Entradas: | -Nombre del personaje |
| Salida: | Modifica el nombre del personaje |

|  |  |
| --- | --- |
| Nombre: | **RF10** Modificar la personalidad del personaje |
| Resumen: | Poder modificar la personalidad del personaje determinado |
| Entradas: | -Personalidad del personaje |
| Salida: | Modifica la personalidad de un personaje determinado |

|  |  |
| --- | --- |
| Nombre: | **RF11** Modificar la fecha de creación |
| Resumen: | Poder modificar la fecha de creación de un personaje |
| Entradas: | -La fecha de creación de un personaje |
| Salida: | Modifica la fecha de creación de un personaje determinado |

|  |  |
| --- | --- |
| Nombre: | **RF12** Modificar el poder de un personaje |
| Resumen: | Modifica el poder de un personaje determinado |
| Entradas: | -El poder de un personaje |
| Salida: | Modifica le poder de un personaje determinado |

|  |  |
| --- | --- |
| Nombre: | **RF13** Modificar el nombre de la técnica |
| Resumen: | Modifica el nombre de la técnica de un personaje |
| Entradas: | -El nombre de la técnica |
| Salida: | Modifica el nombre de una técnica |

|  |  |
| --- | --- |
| Nombre: | **RF14** Modifica el poder de una técnica |
| Resumen: | Modifica el poder de la técnica de un personaje |
| Entradas: | -El poder |
| Salida: | Modifica el poder de una técnica |

|  |  |
| --- | --- |
| Nombre: | **RF15** Eliminar un clan |
| Resumen: | Poder eliminar un clan determinado |
| Entradas: | -El nombre del clan |
| Salida: | Se elimina a un personaje |

|  |  |
| --- | --- |
| Nombre: | **RF16** Eliminar un personaje |
| Resumen: | Poder eliminar a un personaje determinado dentro de un clan |
| Entradas: | -Nombre del clan  -Nombre del personaje |
| Salida: | Se elimina el personaje dentro de un clan |

|  |  |
| --- | --- |
| Nombre: | **RF17** Eliminar una técnica |
| Resumen: | Permitir eliminar un técnica de un determinado personaje dentro de un clan |
| Entradas: | -nombre del clan  -nombre del personaje  -nombre de la técnica |
| Salida | Se elimina la técnica del personaje |

|  |  |
| --- | --- |
| Nombre: | **RFN1** Implementar los tres métodos de ordenamiento: burbuja, inserción y selección |
| Resumen | Se utiliza cada uno de los métodos de ordenamientos |
| Entradas: | Ninguna |
| Salida: | Utilizar cada uno de los métodos de ordenamiento |

|  |  |
| --- | --- |
| Nombre: | **RFN2** Utilizar las búsquedas clásicas dentro del programa |
| Resumen: | Utilizar cada una de las técnicas de búsqueda secuencial |
| Entradas: | Ninguna |
| Salida: | Se utiliza cada uno de los métodos |

|  |  |
| --- | --- |
| Nombre: | **RFN3** Utilizar la interface Comparable |
| Resumen: | Utilizar la interfaz comparable |
| Entradas: | Ninguna |
| Salida: | Se utiliza la interfaz de la Comparable |

|  |  |
| --- | --- |
| Nombre: | **RNF4** Utiliza la interfaz Comparator |
| Resumen: | Utilizar la interfaz comparator |
| Entradas: | Ninguna |
| Salida: | Se utiliza la interfaz Comparator |

|  |  |
| --- | --- |
| Nombre: | **RNF5** Ordenar ascendentemente las técnicas de un personaje por un factor |
| Resumen: | Utilizar los métodos de ordenamiento para organizar las técnicas |
| Entradas: | Ninguna |
| Salida: | Se organiza las técnicas de un personaje ascendentemente por el factor |

|  |  |
| --- | --- |
| Nombre: | **RNF 6** Utilizar diferentes métodos de ordenamiento y búsquedas para un caso |
| Resumen: | Usar diferentes tipos de ordenamientos para cada caso |
| Entradas: | Ninguna |
| Salida: | Utiliza métodos de ordenamientos para cada uno de los casos |

4.

|  |  |  |
| --- | --- | --- |
| RF Y RNF | Código | |
| Clase | Método |
| Hacer la creación clanes | University | +addClan(String):void |
| Agregar un personaje dentro de los clanes | University,Clan | +callTheMethodLookForTheNoRepeatSamurai(String,Samurai).String  +lookForNotRepeatCharacters(Samurai) String |
| Agregar una técnica a un determinado personaje | University,Clan,Samurai | +callTheMethodsLookForNotRepeatTechnique(String,String,Technique):String  +callTheMethodNotRepeatTechniques(String,Technique): String  +lookForNotRepeatTechniques(Technique) String |
| Serializar los clanes, de los personajes y de las técnicas | University | +serializableClan():void  +desSerializableClan()void |
| Mostrar la información de los clanes | University | +infoClan():String |
| Mostrar la información de los personajes | University,Clan | +showInformationSamurai(String) String  +mostrarInfo():String |
| Mostrar la información de las técnicas de los personajes | University,Clan,Samurai | +showInformationSamurai(String) String  +mostrarInfo():String  +mostrarInfo():String |
| Modificar el nombre del personaje | University,Clan | +modificarNameCharacter(String, String):void  +modifiqueName(String, String, String):void |
| Modificar la personalidad del personaje | University,Clan | +modifiqueNameInfluencer(String, String, String):void  +modificarNameInfluencer(String, String):void |
| Modificar la fecha de creación | University,Clan | +modifiqueDateCreation(String,String,String) void  +modificarDateSamurai(String,String):void |
| Modificar el poder de un personaje | University,Clan | +modifiquePower(String String int) void  +modificarPowerSamurai(String.int):void |
| Modificar el nombre de la técnica | University,Clan,Samurai | +modifiqueNameTechnique(String, String, String, String):void  +modificarNameTechnique(String.String,String):void |
| Modifica el poder de una técnica | University,Clan,Samurai | +modifiquePowerTechnique(String,String,String,int):void  +modificarPowerSamurai(String.int):void |
| Eliminar un clan | University | +EliminarClan(String).void |
| Eliminar un personaje | University,Clan | +callDelatedSamurai(String,String) void  +delatedSamurai(String):void |
| Eliminar una técnica | University,Clan,Technique | +calDelatedTechnique(String, String, String) void  +callMethodDelatedSamurai(String,String):void  +delatedTechniqueNameTechnique(String) void |
| Implementar los tres métodos de ordenamiento: burbuja, inserción y selección | University,Clan,Technique | +ordenarPorInsercion() void  +ordenarPorSeleccion():void  +callSortBurblePorNombre(String) void +callSortBurblePorPower(String) void  +callMethodSelection(String) :void +callMethodInsercion(String) :void  +ordenarPorBurbujaMejorada():void +ordenarPorBurbujaMejorada1():void  +longitud():int +indiceElement(int): Technique +ordenarPorInsercion influencer()void  +ordenarPorSeleccionNombre():void  +modificarPosición(int, Technique):void |
| Utilizar las búsquedas clásicas dentro del programa | University,Clan,Technique | +callMethodSecuencialTechniqueName(String, String, String):String +calMethodSecuencialTechniquelinfluencer(String,String,int):String +calMethodSecuencialSamuName(String,String):String +callMethodSecuencialPower(String.int):String  +searchObjectClan(String): String  +getSamuName(String) String  +getSamuPower(int) String  +callMethodSecuencialSearchinfluencer(String, int) :String +callMethodSecuencialSearchName(String, String):String  +getNameTechnique Object(String): String +getinfluencer0bject(int):String |
| Utilizar la interface Comparable | University,Clan,Samurai | +compare(Technique, Technique) int  +compare(Clan Clan) int  +compare(Samurai,Samurai) int |
| Utiliza la interfaz Comparator | University,Clan,Samurai | +compareTo(Clan) int  +compareTo(Samurai) int  +compareTo(Technique) int |
| Ordenar ascendentemente las técnicas de un personaje por un factor | Samurai | +ordenarPorBurbujaMejorada():void +ordenarPorBurbujaMejorada1():void |
| Utilizar diferentes métodos de ordenamiento y búsquedas para un caso | University,Clan,Samurai | +ordenarPorInsercion() void  +ordenarPorSeleccion():void  +callSortBurblePorNombre(String) void +callSortBurblePorPower(String) void  +callMethodSelection(String) :void +callMethodInsercion(String) :void  +ordenarPorBurbujaMejorada():void +ordenarPorBurbujaMejorada1():void  +longitud():int +indiceElement(int): Technique +ordenarPorInsercion influencer()void  +ordenarPorSeleccionNombre():void  +modificarPosición(int, Technique):void |

5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| ordenarPorInsercion(); | University | El escenario le entran varios objetos de los cuales tienen su respectivo nombre:e1 = “Uzumaki” y e2 = “Uchiha” | -------------------- | Ordena de primero a e2 antes que a e1 |
| ordenarPorSeleccion() | University | El escenario le entran varios objetos de los cuales tienen su respectivo nombre:e1 = “Uzumaki” y e2 = “Uchiha” | -------------------- | Ordena de primero a e2 antes que a e1 |

Objetivo: Saber si el método ordena de manera correcta los objetos de la clase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| addClan() | University | El escenario le entra un atributo de tipo String el cual se convertirá en un objeto | e1 = “Konahaku” | Agrega al objeto dentro de la lista de clanes |
| addClan() | University | El escenario le entra un atributo de tipo String el cual se convertirá en un objeto | e1 = “Uchiha” | No se agrego el objeto puesto que ya exite uno con ese mismo nombre |

Objetivo: Saber si el método de añadir clan lo hace de manera que vaya con la logica del método

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| serializableClan(); | University | El escenario tiene varios objetos dentro de la lista de clanes | -------------------- | Serializa los objetos en un archivo.dat |
| serializableClan(); | University | El escenario no tiene objetos dentro de la lista de clanes | -------------------- | no serializa ningún objeto. |

Objetivo: Saber si el método serializable hace su funcion de acuerdo a la logica de los métodos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| deserializableClan(); | University | El escenario tiene varios objetos nuevos dentro del archivo.dat de clanes | -------------------- | deserializar a los objetos para añadirlos a la lista de clanes |
| deserializableClan(); | University | El escenario no tiene objetos dentro del archivo.dat de clanes | -------------------- | no deserializa ningun objeto del archivo |

Objetivo: Saber si el método permite hacer la deserialización de los objetos dentro del archivo.dat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| EliminarClan() | University | El escenario le entran varios objetos de los cuales tienen su respectivo nombre:e1 = “Uzumaki” y e2 = “Uchiha” y le entra por parámetro un atributo de tipo string | e1 = “Uchiha” | Elimna al objeto del cual se pedia eliminar |
| EliminarClan() | University | El escenario le entran varios objetos de los cuales tienen su respectivo nombre:e1 = “Uzumaki” y e2 = “Uchiha” y le entra por parámetro un atributo de tipo string | e2 = “Nara” | No elimina ningún objeto de tipo Clan porque no encontro |

Objetivo: Saber si el método elimina a un determinado clan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| callMethodSecuencialTechniqueName(String nameClan,String nameSamu,String nameTechnique) | university | El escenario tiene unos objetivos iniciales de clanes,de samurai y de técnicas. e1 = “Uzumaki”,e2 = “Sasuke”,e = “Rasengan” y le entran unos parámetros | nameClan = “Uzumaki”, nameSamu = “Sasuke”, nameTechnique = “Rasengan” | Se encontro el objeto y se llama al metodo |
| callMethodSecuencialTechniqueName(String nameClan,String nameSamu,String nameTechnique) | university | El escenario tiene unos objetos iniciales de clanes,de samurai y de técnicas. e1 = “Uzumaki”,e2 = “Sasuke”,e = “Rasengan” y le entran unos parámetros | nameClan = “Otsusuki”, nameSamu = “Sasuke”, nameTechnique = “Rasengan | No se encontró el objeto, por lo tanto no se puede invocar el método |

Objetivo: Saber si se llama correctamente el método.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| callMethodSecuencialTechniqueInfluencer(String nameClan, String nameSamu,int influencer) | University | El escenario tiene unos objetos iniciales, e1 = “Uzumaki”,e2 = “Sasuke”,e = 1231 y le entran unos parámetros | nameClan = “Uzumaki”, nameSamu = “Sasuke”, influencer= 1231 | Se encontró el objeto por lo cual llamará el objeto que tenga ese influenciador |
| callMethodSecuencialTechniqueInfluencer(String nameClan, String nameSamu,int influencer) | University | El escenario tiene unos objetos iniciales, e1 = “Uzumaki”,e2 = “Sasuke”,e = 12311 y le entran unos parámetros | nameClan = “Uzumaki”, nameSamu = “Sasuke”, influencer= 12311 | No se encontró el objeto, por lo cual no llamará el objeto que tenga ese influenciador |

Objetivo: Saber si se llama correctamente al método de búsqueda secuencial

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| callMethodSecuencialSamuName(String nameClan, String nameSamu) | University | El escenario tiene unos objetos iniciales, e1 = “Uzumaki”,e2 = “Sasuke” | nameClan = “Uzumaki”, nameSamu = “Sasuke” | Se llama al método el cual le dará el objeto de la secuencia |
| callMethodSecuencialSamuName(String nameClan, String nameSamu) | University | El escenario tiene unos objetos iniciales, e1 = “Uzumaki”,e2 = “Naruto” | nameClan = “Uzumaki”, nameSamu = “Sasuke” | No se llama al método el cual no le dará el objeto de la secuencia |

Objetivo: Saber si se llama correctamente al metodo de busqueda secuencial del nombre del ninja.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| String callMethodSecuencialPower(String nameClan,int influencer) | University | El escenario tiene unos objetos iniciales, e1 = “Uzumaki”,e2 = 12313 | nameClan = “Uzumaki”, influencer= 12313 | Se llama al método el cual dará el objeto de la secuencia |
| String callMethodSecuencialPower(String nameClan,int influencer) | University | El escenario tiene unos objetos iniciales, e1 = “Uzumaki”,e2 = 123133 | nameClan = “Uzumaki”, influencer= 123133 | No se llama al método el cual no le dará el objeto de la secuencia |

Objetivo: Saber si se llama correctamente el metodo de busqueda secuencial por el poder del ninja

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| callMethodSelectionMethod(String nameClan,String nameSamu) | University | El escenario tiene unos objetos iniciales, e1 = “Uzumaki”,e2 = “Orochimaru” | nameClan = “Uzumaki”, influencer= “Orochimaru” | Se llama al método que hace la selección en samurai |
| callMethodSelectionMethod(String nameClan,String nameSamu) | University | El escenario tiene unos objetos iniciales, e1 = “Uzumaki”,e2 = “Sasuke” | nameClan = “Uzumaki”, influencer= “Orochimaru” | No se puede dar el método de selección de la clase samurai |

Objetivo: Permitir conocer si el método llama al método de selección.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| searchObjectClan(String nameC) | University | El escenario tiene unos clanes iniciales de los cuales unos son: e1= “Uzumaki”,e2 = “Boruto” | nameClan = “Uzumaki” | Se llama al método para que le retorne el clan |
| searchObjectClan(String nameC) | University | El escenario tiene unos clanes iniciales de los cuales unos son: e1= “Uzumaki”,e2 = “Boruto” | nameClan = “Uchiha” |  |

Objetivo: Permitir conocer si el método hace correctamente la busqueda secuencial en el clan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modifiquePowerTechnique(String nameClan,String nameSamurai,String nameTechnique,int powerInfluencer) | University | El escenario tiene unos objetos iniciales  e1 = “Uzumaki”  e2 = “Uchiha”...  e3 = “Chidori”.... y demás objetos | nameClan = “Uzumaki”  nameSamurai = “Uchiha”  nameTechnique = “ Chidori”  powerInfluencer = 12313 | Se cambia el poder la técnica de determinado personaje en un clan |
| modifiquePowerTechnique(String nameClan,String nameSamurai,String nameTechnique,int powerInfluencer) | University | El escenario tiene unos objetos iniciales  e1 = “Uzumaki”  e2 = “Uchiha”...  e3 = “Chidori”.... y demás objetos | nameClan = “Uzumaki”  nameSamurai = “Uchiha1”  nameTechnique = “Rasengan”  powerInfluencer = 11231231 | No se cambia el poder técnica de determinado personaje en un clan |

Objetivo: Saber si el método modifica de manera efectiva el nombre de la técnica.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modifiqueNameTechnique(String nameClan,String nameSamurai,String nameTechnique,String newTechnique) | University | El escenario tiene unos objetos iniciales  e1 = “Uzumaki”  e2 = “Uchiha”...  e3 = “Chidori”.... y demás objetos | nameClan = “Uzumaki”  nameSamurai = “Uchiha”  nameTechnique = “Rasengan”  newTechnique = “Chidori” | Se cambia el nombre la técnica de determinado personaje en un clan |
| modifiqueNameTechnique(String nameClan,String nameSamurai,String nameTechnique,String newTechnique) | University | El escenario tiene unos objetos iniciales  e1 = “Uzumaki”  e2 = “Uchiha”...  e3 = “Chidori”.... y demás objetos | nameClan = “Uzumaki”  nameSamurai = “Uchiha1”  nameTechnique = “Rasengan1”  newTechnique = “Chidori” | No se cambia el nombre la técnica de determinado personaje en un clan |

Objetivo: Saber si el método modifica el nombre de la técnica correctamente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modifiquePower(String nameClan,String nameSamurai,int nameNew) | University | El escenario tiene unos objetos iniciales de los cuales uno: e1 = “Uzumaki”,e2 = “Uchiha” y los demás. | nameClan = “Uzumaki”  nameSamurai = “Uchiha”  nameNew = 123123 | Se cambia el poder del samurai que se le asignó |
| modifiquePower(String nameClan,String nameSamurai,int nameNew) | University | El escenario tiene unos objetos iniciales de los cuales uno: e1 = “Uzumaki”,e2 = “Uchiha” y los demás. | nameClan = “Uzumaki”  nameSamurai = “Uchiha1”  nameNew = 12312 | No se cambia el poder del samurai que se le asignó |

Objetivo: Saber si el método hace de manera adecuada la ejecución de este modificación de poder.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modifiqueDateCreation(String nameClan,String nameSamurai,String nameNew) | University | El escenario tiene unos objetos iniciales de los cuales uno: e1 = “Uzumaki”,e2 = “Uchiha” y los demás. | nameClan = “Uzumaki”  nameSamurai = “Uchiha”  nameNew = “2019/06/13” | Se cambia la fecha del ninja |
| modifiqueDateCreation(String nameClan,String nameSamurai,String nameNew) | University | El escenario tiene unos objetos iniciales de los cuales uno: e1 = “Uzumaki”,e2 = “Uchiha” y los demás. | nameClan = “Uzumaki”  nameSamurai = “Uchiha1”  nameNew = “2019/06/13” | No se cambia la fecha del ninja |

Objetivo: Saber si el método modifica la fecha creacion del samurai

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modifiqueNameClan(String nameClan,String nameNew) | University | El escenario tiene unos clanes iniciales de los cuales uno: e1 = “Uzumaki” | nameClan = “Uzumaki”  nameNew = “Uzumaki Shippuden” | Se cambia el nombre del clan determinado |
| modifiqueNameClan(String nameClan,String nameNew) | University | El escenario tiene unos clanes iniciales de los cuales uno: e1 = “Uzumaki” | nameClan = “Uzumaki1”  nameNew = “Uzumaki Shippuden” | No se cambia el nombre del clan determinado |

Objetivo: Permitir cambiar el nombre de un determinado clan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modifiqueNameInfluencer(String nameClan,String nameSamurai,String nameNew) | University | El escenario tiene unos clanes, ninjas y técnicas iniciales de los cuales uno:  e1 = “Uzumaki”  e2 = “Sasuke”.. y los demás | nameClan = “Uzumaki”  nameSamurai = “Kakachi”  nameNew = “Kakarato” | No se cambia el nombre de un ninja determinado |
| modifiqueNameInfluencer(String nameClan,String nameSamurai,String nameNew) | University | El escenario tiene unos clanes, ninjas y técnicas iniciales de los cuales uno:  e1 = “Uzumaki”  e2 = “Sasuke”.. y los demás | nameClan = “Uzumaki”  nameSamurai = “Sasuke”  nameNew = “Kakarato” | Se cambia el nombre de un ninja determinado |

Objetivo:Permitir cambiar el nombre de la técnica

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modifiqueName(String nameClan,String nameSamurai,String nameNew) | University | El escenario tiene unos clanes, ninjas y técnicas iniciales de los cuales uno:  e1 = “Uzumaki”  e2 = “Sasuke”.. y los demás | nameClan = “Uzumaki”  nameSamurai = “Sasuke”  nameNew = “Tobirama” | Se modifica el nombre del samurai |
| modifiqueName(String nameClan,String nameSamurai,String nameNew) | University | El escenario tiene unos clanes, ninjas y técnicas iniciales de los cuales uno:  e1 = “Uzumaki”  e2 = “Sasuke”.. y los demás | nameClan = “Uzumaki”  nameSamurai = “Sasuke2”  nameNew = “Tobirama” | No se modifica el nombre del samurai |

Objetivo: Permitir saber si se modifica el nombre del samurai

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| callDelatedTechnique(String nameClan,String nameSamurai,String nameTechnique) | University | El escenario tiene unos clanes, ninjas y técnicas iniciales de los cuales uno:  e1 = “Uzumaki”  e2 = “Sasuke”..  e3 = “Rasengan”  y los demás | nameClan = “Uzumaki”  nameSamurai = “Sasuke”  nameTechnique = “Rasengan” | Se elimina la técnica del objeto |
| callDelatedTechnique(String nameClan,String nameSamurai,String nameTechnique) | University | El escenario tiene unos clanes, ninjas y técnicas iniciales de los cuales uno:  e1 = “Uzumaki”  e2 = “Sasuke”..  e3=“Rasengan” y los demás | nameClan = “Uzumaki”  nameSamurai = “Sasuke1”  nameTechnique = “Rasengan” | No se elimina la técnica del objeto |

Objetivo: Permitir saber si se elimina una técnica.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| callDelatedSamurai(String nameClan,String nameSamurai) | University | El escenario tiene unos clanes, ninjas y técnicas iniciales de los cuales uno:  e1 = “Uzumaki”  e2 = “Sasuke”.. | nameClan = “Uzumaki”  nameSamurai = “Sasuke” | Se elimina el samurai |
| callDelatedSamurai(String nameClan,String nameSamurai) | University | El escenario tiene unos clanes, ninjas y técnicas iniciales de los cuales uno:  e1 = “Uzumaki”  e2 = “Sasuke”.. | nameClan = “Uzumaki”  nameSamurai = “Sasuke1” | No se elimina el samurai |

Objetivo: Saber si se elimina el samurai determinado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| callMethodSecuencialSearchInfluencer(String nameSamurai,int influencer) | Clan | El escenario tiene unos samurai y técnicas iniciales. e1 = “Sasuke”..,e2 = “Chidori”,1231 | nameSamurai = “Sasuke”  influencer = 1231 | Se llama el método de influencia secuencial |
| callMethodSecuencialSearchInfluencer(String nameSamurai,int influencer) | Clan | El escenario tiene unos samurai y técnicas iniciales. e1 = “Sasuke”..,e2 = “Chidori”,1231 | nameSamurai = “Sasuke”  influencer = 1231 | No se llama el método de influencia secuencial |

Objetivo: Permitir conocer si se llama el método de secuencia de la técnica

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| callMethodSecuencialSearchName(String nameSamurai,String nameTech) | Clan | El escenario tiene unos samurai y técnicas iniciales. e1 = “Sasuke”..,e2 = “Chidori” | nameSamurai = “Sasuke”  nameTech = “Chidori” | Se da el objeto de la búsqueda secuencial |
| callMethodSecuencialSearchName(String nameSamurai,String nameTech) | Clan | El escenario tiene unos samurai y técnicas iniciales. e1 = “Sasuke”..,e2 = “Chidori” | nameSamurai = “Sasuke”  nameTech = “Chidori1” | No se da el objeto de la búsqueda secuencial |

Objetivo: Permitir conocer si el hace la busqueda secuencial de la técnica.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| callMethodSelection(String nameSamurai) | Clan | El escenario tiene unos samurai y técnicas iniciales. e1 = “Sasuke”..,e2 = “Chidori” | nameSamurai = “Sasuke” | Se da el objeto de la búsqueda secuencial |
| callMethodSelection(String nameSamurai) | Clan | El escenario tiene unos samurai y técnicas iniciales. e1 = “Sasuke”..,e2 = “Chidori” | nameSamurai = “Sasuke1” | No se da el objeto de la búsqueda secuencial |

Objetivo: Permitir conocer si el hace la busqueda secuencial de la técnica.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| callMethodInsercion(String nameSamurai) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi” y los demás objetos | nameSamurai = “Itachi” | Se llama el metodo de insercion |
| callMethodInsercion(String nameSamurai) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi” y los demás objetos | nameSamurai = “Utachi” | No se puede llamar el metodo de inserción |

Objetivo: Permitir conocer si se obtiene el método de ordenar por inserción.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modificarInfluencerTechnique(String nameCharacter,String nameTechnique,int influencer) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”,1231 y los demás objetos | nameCharacter = “Itachi”  nameTechnique = “Chidori”  influencer = 123131 | Se modifica el influenciador de la tecnica |
| modificarInfluencerTechnique(String nameCharacter,String nameTechnique,int influencer) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”,1231 y los demás objetos | nameCharacter = “Itachi1”  nameTechnique = “Chidori1”  influencer = 123131 | No se modifica el influenciador de la tecnica |

Objetivo: Permitir saber si se modifica el influenciador de la tecnica

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modificarNameTechnique(String nameCharacter,String nameTechnique,String nameNewTechnique) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | nameCharacter = “Itachi”  nameTechnique=”Chidori”  nameNewTechnique = “Jhonja” | Se cambia el nombre de la técnica |
| modificarNameTechnique(String nameCharacter,String nameTechnique,String nameNewTechnique) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | nameCharacter = “Itachi1”  nameTechnique=”Chidori1”  nameNewTechnique = “Jhonja” | No se cambia el nombre de la técnica |

Objetivo: Permitir saber si se cambia el nombre de la técnica.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modificarPowerSamurai(String nameCharacter,int powerNew) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | nameCharacter = “Itachi”  nameTechnique=”Chidori”  powerNew = 12313 | Se cambia el poder de la técnica |
| modificarPowerSamurai(String nameCharacter,int powerNew) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | nameCharacter = “Itachi1”  nameTechnique=”Chidori1”  powerNew = 12313 | No se cambia el poder de la técnica |

Objetivo: Permitir saber si se modifica el poder del ninja determinado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modificarDateSamurai(String nameCharacter,String creationDate) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | nameCharacter = “Itachi”  creationDate = “2019/08/09” | Se cambia la fecha de creación del personaje |
| modificarDateSamurai(String nameCharacter,String creationDate) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | nameCharacter = “Itachi1”  creationDate = “2019/08/09” | No se cambia la fecha de creación del personaje |

Objetivo: Permitir saber si se modifica la fecha del samurai.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modificarNameInfluencer(String nameCharacter,String nameNewInfluencer) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | nameCharacter = “Itachi”  nameNewInfluencer = “Sabio” | Se cambia la personalidad del personaje |
| modificarNameInfluencer(String nameCharacter,String nameNewInfluencer) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | nameCharacter = “Itachi1”  nameNewInfluencer = “Sabio” | No se cambia la personalidad del personaje |

Objetivo: Saber si se cambia la personalidad del personaje.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modificarNameCharacter(String nameCharacter,String nameNew) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | nameCharacter = “Itachi”  nameNew = “Otsuchi” | Se cambia el nombre del personaje |
| modificarNameCharacter(String nameCharacter,String nameNew) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | nameCharacter = “Itachi1”  nameNew = “Otsuchi” | No se cambia el nombre del personaje |

Objetivo: Saber si se cambia el nombre del personaje determinado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| addCharacterFinal(Samurai e) | Clan | El escenario tiene una lista vacía | Samurai = “Itachi”... | Añade el samurai a la lista |
| addCharacterFinal(Samurai e) | Clan | El escenario tiene una lista con determinados objetos como: e1 = “Izuna”.. | Samurai = “Izuna”.. | No se añade en la lista |

Objetivo saber si se puede añadir un samurai a la lista de clanes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| delatedSamurai(String nameCharac) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | nameCharac = “Itachi” | Se elimina el personaje de la lista |
| delatedSamurai(String nameCharac) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | nameCharac = “Itachi1” | No se elimina el personaje de la lista |

Objetivo: Permitir eliminar a un samurai por su nombre

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| callMethodDelatedSamurai(String samuName,String nameTechnique) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | samuName = “Itachi”  nameTechnique = “Chidori” | Se elimina la técnica del personaje |
| callMethodDelatedSamurai(String samuName,String nameTechnique) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | samuName = “Itachi1”  nameTechnique = “Chidori1” | No se elimina la técnica del personaje |

Objetivo: Permitir saber si se elimina la técnica de un samurai determinado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| getSamuName(String samuName) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | samuName = “Itachi” | Se cambia el nombre de la tecnica |
| getSamuName(String samuName) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”y los demás objetos.además llama por parametro unos atributos | samuName = “Itachi1” | No se cambia el nombre de la tecnica |

Objetivo:Permitir saber si el metodo de busqueda secuencial retorna el personaje por su nombre.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| getSamuPower(int samuPower) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”,12313,los demás objetos.además llama por parametro unos atributos | samuPower = 12313 | Retorna el objeto de la búsqueda secuencial por el poder |
| getSamuPower(int samuPower) | Clan | El escenario tiene unos objetos iniciales de Samurai como uno : e1 = “Itachi”,e2 = “Chidori”,12313,los demás objetos.además llama por parametro unos atributos | samuPower = 1231 | No retorna el objeto de la búsqueda secuencial por el poder |

Objetivo: Permitir saber si el método de la búsqueda secuencial por el poder del ninja.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| ordenarPorBurbujaMejorada() | Clan | El escenario tiene unos objetos iniciales los cuales son: e2 =”Akamaru”,12313 y e1 = “Bonji”,3123 y los demás objetos de la lista de samurai | ------------------ | ordena de primero e2 y luego e1 |
| ordenarPorBurbujaMejorada() | Clan | El escenario tiene unos objetos iniciales los cuales son: e2 =”Akamaru”,12313 y e1 = “Aonjito”,3123 y los demás objetos de la lista de samurai | ------------------- | ordena primero e1 y luego e2 |

Objetivo: Permite saber si ordena los objetos de acuerdo a los nombres de los samurais.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| ordenarPorBurbujaMejorada1() | Clan | El escenario tiene unos objetos iniciales los cuales son: e2 =”Akamaru”,2313 y e1 = “Aonjito”,3123 y los demás objetos de la lista de samurai | -------------------- | ordena de primero e1 y luego e2 |
| ordenarPorBurbujaMejorada1() | Clan | El escenario tiene unos objetos iniciales los cuales son: e2 =”Akamaru”,4134 y e1 = “Aonjito”,3123 y los demás objetos de la lista de samurai | -------------------- | ordena de primero e2 y luego e1 |

Objetivo: Permite saber si ordena los objetos de acuerdo a los poderes del samurai

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| addInicio(Technique e) | Samurai | El escenario tiene unos objetos iniciales por parametro que son e1= “Aojito”,3123 y e2= “Akamaru”,12313 | e1 y e2 | Se agrega primero e1 y luego e2 |

Objetivo: Permitir saber si el método hace la función de añadir de primero.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| addTechniqueInfuencer(Technique e) | Samurai | El escenario tiene unos objetos iniciales por parametro que son e1= “Aojito”,3123 y e2= “Akamaru”,12313 | e1 y e2 | Se agrega primero e2 y luego e1 |

Objetivo: Permitir saber si se añade ordenadamente la lista.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| delatedTechniqueNameTechnique(String nameTecnique) | Samurai | El escenario tiene unos objetos iniciales de técnicas que unos son:e1 = “Rasengan”,12313 y e2 = “Chidori” 31333 | nameTechnique = “Rasengan” | Elimina la técnica de acuerdo a su nombre |
| delatedTechniqueNameTechnique(String nameTecnique) | Samurai | El escenario tiene unos objetos iniciales de técnicas que unos son:e1 = “Rasengan”,12313 y e2 = “Chidori” 31333 | nameTechnique = “Rasengan1” | No elimina la tecnica de acuerdo a su nombre |

Objetivo: Permitir saber si elimina a una técnica determinada.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modificarInfluencer(String nameTecnique,int numberTechnique) | Samurai | El escenario tiene unos objetos iniciales de técnicas que unos son:e1 = “Rasengan”,12313 y e2 = “Chidori” 31333 | nameTecnique = “Rasengan”  numberTechnique = 313333 | Se modifica el influenciador de la técnica |
| modificarInfluencer(String nameTecnique,int numberTechnique) | Samurai | El escenario tiene unos objetos iniciales de técnicas que unos son:e1 = “Rasengan”,12313 y e2 = “Chidori” 31333 | nameTecnique = “shidori”  numberTechnique = 313333 | No se modifica el influenciador de la técnica |

Objetivo: Permitir saber si el método modifica el influenciador

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| modificarInfoName(String nameTechnique,String otherName) | Samurai | El escenario tiene unos objetos iniciales de técnicas que unos son:e1 = “Rasengan”,12313 y e2 = “Chidori” 31333 | nameTechnique = “Rasengan”  otherName = “Mokton” | Se modifica el nombre de la técnica seleccionada |
| modificarInfoName(String nameTechnique,String otherName) | Samurai | El escenario tiene unos objetos iniciales de técnicas que unos son:e1 = “Rasengan”,12313 y e2 = “Chidori” 31333 | nameTechnique = “sasengan”  otherName = “Mokton” | No se modifica el nombre de la técnica seleccionada |

Objetivo: Permitir saber si el método cambia la información del nombre.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| ordenarPorSeleccionNombre() | Samurai | El escenario tiene unos objetos iniciales de técnicas que unos son:e1 = “Rasengan”,12313 y e2 = “Chidori” 31333 | ------------------- | Ordena primero el e2 y luego el e1 |

Objetivo: Permitir saber si ordena los samurai en orden ascendente de acuerdo al atributo de su nombre.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| ordenarPorInsercionInfluencer() | Samurai | El escenario tiene unos objetos iniciales de técnicas que unos son:e1 = “Rasengan”,12313 y e2 = “Chidori” 31333 | ------------------- | ordena primero el e2 y el e1 |

Objetivo: Permitir saber si ordena los samurai en orden ascendente de acuerdo al atributo de su influenciador

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| compare(Technique o1, Technique o2) | Technique | El escenario le entra dos objetos como parametro o1 = “Rasengan”1231, o2 = “Chidori”,3123 | o1 y o2 | el mayor seria el o2 |
| compare(Technique o1, Technique o2) | Technique | El escenario le entra dos objetos como parametro o1 = “Rasengan”123111, o2 = “Chidori”,3123 | o1 y o2 | el mayor seria el o1 |

Objetivo: Saber si los métodos de comparación están bien estructurados.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Método | Clase | Escenario | Valores de entrada | Resultado |
| compareTo(Technique o) | Technique | El escenario le entra dos objetos como parametro o1 = “Rasengan”123111, o2 = “Chidori”,3123 | o1 y o2 | el mayor seria o2 |
| compareTo(Technique o) | Technique | El escenario le entra dos objetos como parametro o1 = “Rasengan”123111, o2 = “Chidori”,3123 | o1 y o2 | el menor seria o1 |

Objetivo: Saber si los métodos de comparación están bien estructurados.