**TRABAJO FINAL CORTE 1 – SISTEMA NARUTO**  
Grupo Tomates: Natalia Osejo y Cristian Reyes

**Uso de patrones de diseño en el proyecto** (Cómo se utilizó cada patrón de diseño)

* Factory

El patron factory se utilizó para centralizar la creación de los diferentes ninjas que maneja el código, en específico se trabajó con el patrón AbstractFactory con el fin de permitir manejar con fabricas concretas por cada una de las diferentes aldeas y de esta forma los ninjas se crean con una configuración por defecto según la aldea.

Se implemento por medio de una interfaz llamada NinjaFactory, de la cual las otras fábricas concretas extienden su método crearNinja() e implementan la aldea y la lista de jutsus característicos, sin necesidad que el usuario final se preocupe por su creación.

* Builder:

El patrón builder se utilizó para la creación y personalización de instancias de Ninjas paso a paso, con atributos más complejos y personalizables, que van más allá de los atributos básicos que son asignados a un ninja al ser creado por la fábrica de una aldea y no se necesita pasar por un constructor de múltiples parámetros.

La clase NinjaBuilder funciona llamando a métodos que se pueden encadenar (setNombre, setRango, etc), lo que permite al usuario establecer los atributos que quiera y facilita la construcción de ninjas con muchas combinaciones posibles de sus diferentes atributos.

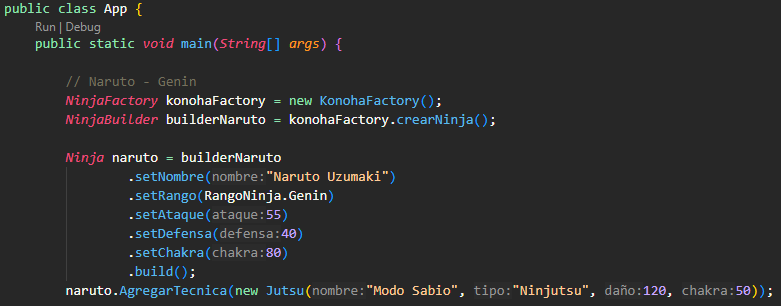
* Visitor:

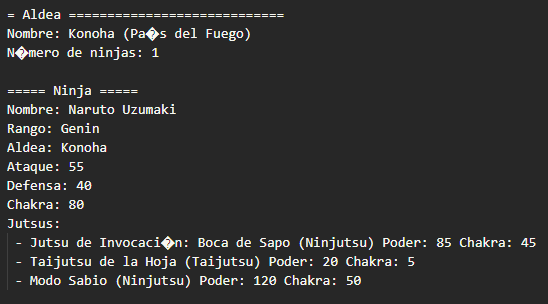
El patrón visitor se utilizó para recorrer y realizar la exportación de los datos de las clases Ninja y Mision sin la necesidad de modificar las clases bases para agregar métodos dedicados para exportar a formatos como XML y JSON.

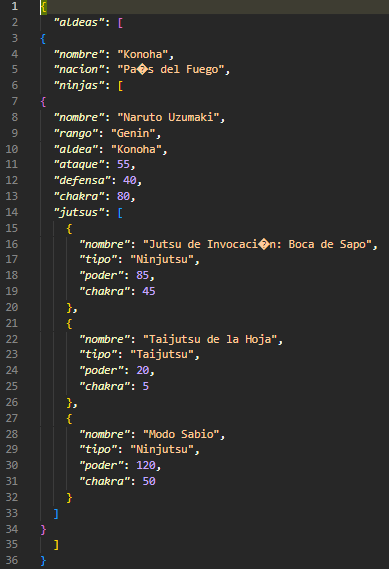
Se implemento por medio de una interfaz Visitor de la cual los otros visitantes concretos implementan, y visitan las clases que tienen la característica de ser visitables con el fin de obtener la información necesaria para exportar a un nuevo formato especificado por la clase como lo puede ser una salida de texto, JSON o XML.

**Ejemplos de ejecución**

* **Creación de un Ninja**

Construir el Ninja desde el Main:  


Informe de texto:   


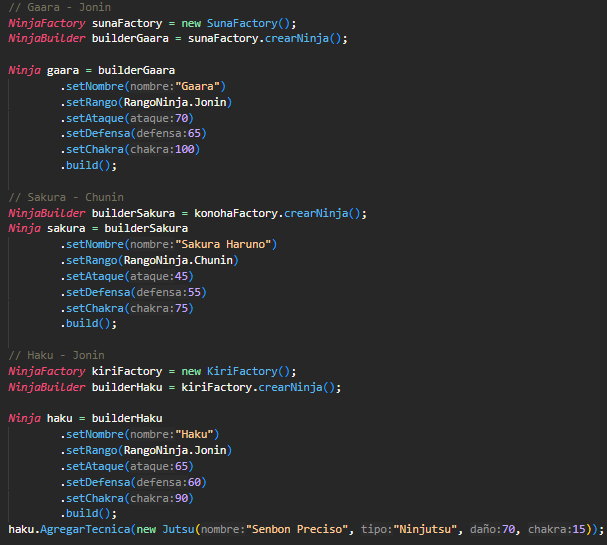
Informe JSON:  


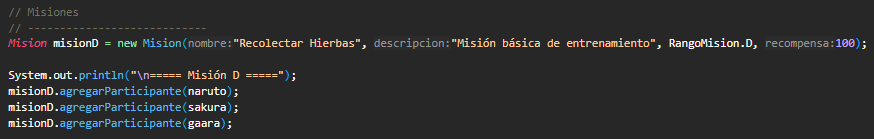
Informe XML:



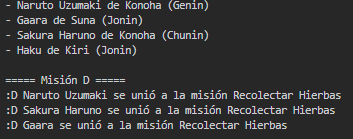
* **Asignación de una misión a varios ninjas**

Creación de otros ninjas:

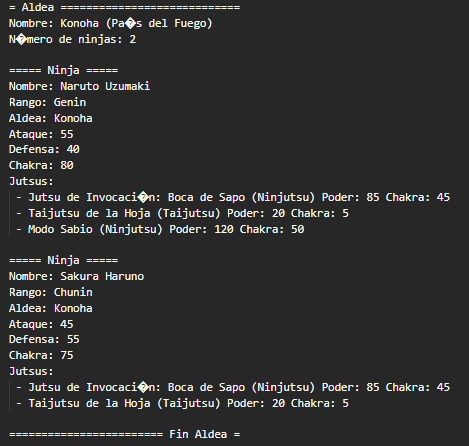
  
Creación y asignación de una misión de rango D:

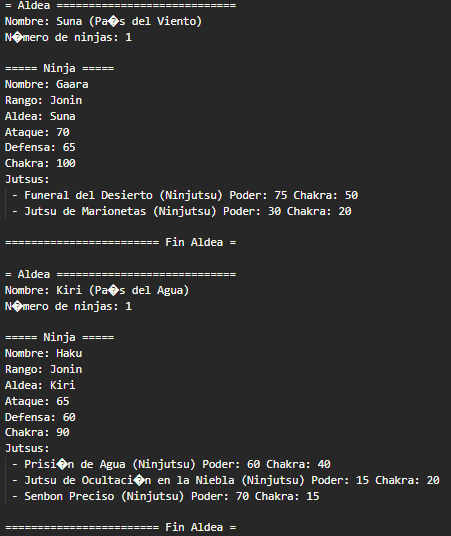
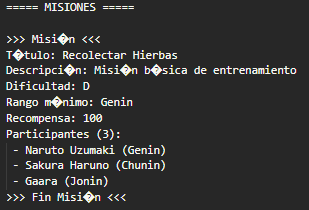


Output de la terminal:



Informe de Texto:



Informe JSON:

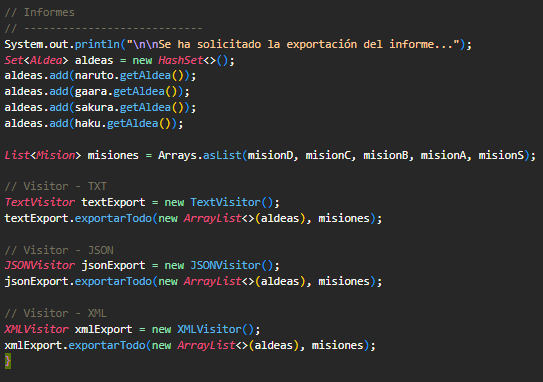


Informe XML:



* **Generación de informes:**

Codigo en el main:



Output en la terminal:  
