

Proyecto Corte I

Camilo Andrés Castillo Guardia

Dylan Sebastián Gutiérrez Ramírez

Santiago Andrés Daza Solarte

Universidad Jorge Tadeo Lozano

Arquitectura de Software

Docente Felipe Esteban Hernández Baquero

26 de Agosto de 2025

Tabla de Contenido

USO DE PATRONES DE DISEÑO.....	3
EJEMPLOS DE EJECUCIÓN.....	5
CONCLUSIONES	13

USO DE PATRONES DE DISEÑO

En el desarrollo del sistema se implementaron tres patrones de diseño solicitados: Factory, Builder y Visitor. A continuación, se describe cómo se aplicó cada uno dentro del proyecto.

1. Patrón Factory

Objetivo del patrón: Centralizar y simplificar la creación de objetos, delegando a una fábrica la responsabilidad de instanciarlos según un contexto.

Implementación en el proyecto:

1. Se definió la interfaz NinjaFactory con el método createNinja().
2. Cada aldea concreta implementa su propia fábrica (KonohaFactory, SunaFactory, KiriFactory), encargada de crear ninjas con características iniciales específicas (ataque, defensa, chakra y jutsus base).
3. La clase Village utiliza estas fábricas para reclutar ninjas, desacoplando la creación de personajes de la lógica del programa.

Beneficio obtenido: Permite que nuevas aldeas con ninjas personalizados puedan añadirse sin modificar el resto del sistema.

En resumidas cuentas: Cada aldea tiene su propia fábrica de ninjas. Cuando el sistema necesita un nuevo ninja, no lo crea directamente, sino que le pide a la fábrica de esa aldea que lo haga.

2. Patrón Builder

Objetivo del patrón: Facilitar la construcción paso a paso de objetos complejos, sin necesidad de múltiples constructores sobrecargados.

Implementación en el proyecto:

- La clase NinjaBuilder permite crear ninjas personalizando atributos como nombre, rango, ataque, defensa, chakra y jutsus.

- En el menú (ConsoleApp), el usuario puede ingresar manualmente los valores y, mediante el builder, se genera un objeto Ninja a medida, como lo desee el usuario.

Beneficio obtenido: Flexibilidad en la creación de ninjas personalizados y claridad en el código al evitar constructores largos.

En resumidas cuentas: Sirve para crear ninjas personalizados. En lugar de tener que llenar todos los datos de golpe, el builder permite ir armando el ninja paso a paso (nombre, rango, ataque, defensa, jutsus, etc.).

3. Patrón Visitor

Objetivo del patrón: Definir operaciones adicionales sobre una estructura de objetos sin modificar sus clases.

Implementación en el proyecto:

1. Se creó la interfaz `NarutoVisitor` con métodos `visit(Ninja)` y `visit(Mission)`.
2. La clase `ExportVisitor` implementa la exportación de información en distintos formatos (JSON, XML y TXT).
3. Los objetos `Ninja` y `Mission` son visitables a través de la interfaz `Acceptable`, lo que permite que el exportador recorra y procese los datos sin alterar la lógica de cada clase.
4. En el menú, la opción de Exportar Datos usa este visitante para generar reportes automáticos, dentro de la carpeta `reports`

Beneficio obtenido: Se pueden añadir fácilmente nuevos tipos de exportación sin modificar las clases principales de `Ninja` o `Mission`.

En resumidas cuentas: Se empleó para exportar información de ninjas y misiones en distintos formatos (JSON, XML, TXT) sin tener que cambiar esas clases.

EJEMPLOS DE EJECUCIÓN

Opción 1:

```
--- Sistema Naruto ---  
1. Crear ninja (manual con Builder)  
2. Crear ninja por aldea (Factory)  
3. Listar ninjas  
4. Entrenar ninja  
5. Crear misión  
6. Listar misiones  
7. Asignar misión a ninja  
8. Ver estado de misiones  
9. Simular combate  
10. Exportar datos  
0. Salir  
Seleccione opción: 1
```

Al iniciar el programa se nos despliega un menú en el cual podemos elegir entre varias opciones, empezaremos por la de crear un ninja con Builder dado como opción el 1.

Posteriormente nos trae otro menú en el que podemos asignarle sus atributos como aldea, nombre, rango, nivel de ataque, defensa y chakra, jutsu, poder del jutsu, nombre del jutsu:

```
--- Crear Ninja (Manual con Builder) ---  
Seleccione aldea:  
1. Suna  
2. Konoha  
3. Kiri  
Opción: 2  
Nombre: Naruto Uzumaki  
Rango (GENIN, CHUNIN, JONIN): GENIN  
Ataque: 70  
Defensa: 50  
Chakra: 100  
Nombre Jutsu (deje vacío para terminar): Rasengan  
Poder del Jutsu: 120  
Nombre Jutsu (deje vacío para terminar):  
Ninja creado: Naruto Uzumaki [GENIN] (Konoha) ATK:70 DEF:50 CH:100 Jutsus:[Rasengan(p=120)]
```

Finalmente queda el ninja creado como se puede ver en la última línea de código

Al finalizar con la opción 1 nos retornara al menú principal

Opción 2:

```
--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: 2
```

En esta opción no tenemos la misma cantidad de atributos a asignar, pero si podemos decidir el nombre del ninja, su aldea y rango

```
--- Crear Ninja por Aldea (Factory) ---
Seleccione aldea:
1. Suna
2. Konoha
3. Kiri
Opción: 1
Nombre: Gaara
Rango (GENIN, CHUNIN, JONIN): JONIN
Ninja creado con Factory:
Gaara [JONIN] (Suna) ATK:9 DEF:10 CH:10 Jutsus:[Manipulación de Arena(p=18)]
```

El Builder se encargará de darnos los atributos faltantes.

Igual como en el ejemplo anterior el ninja queda creado al salir de la configuración de la opción número 2

Opción 3:

Podemos observar los ninjas creados anteriormente con sus atributos.

```

--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: 3

--- Listado de Ninjas ---
Aldea: Suna
  Gaara [JONIN] (Suna) ATK:9 DEF:10 CH:10 Jutsus:[Manipulación de Arena(p=18)]
Aldea: Konoha
  Naruto Uzumaki [GENIN] (Konoha) ATK:70 DEF:50 CH:100 Jutsus:[Rasengan(p=120)]
Aldea: Kiri
  No hay ninjas.

```

Opción 4:

Aquí podemos entrenar alguno de los ninjas que se hayan creado

```

--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: 4

--- Entrenar Ninja ---
1. Gaara (Suna, JONIN)
2. Naruto Uzumaki (Konoha, GENIN)
Seleccione ninja: 2
Entrenamiento completado: +50 a ataque, defensa y chakra.
Ninja entrenado: Naruto Uzumaki [GENIN] (Konoha) ATK:120 DEF:100 CH:150 Jutsus:[Rasengan(p=120)]

```

Al seleccionar uno de los dos ninjas este al entrenar gana atributos

Opción 5:

Aquí pasaremos a la creación de misiones en donde elegiremos el rango de la misión y su respectiva recompensa con el rango mínimo requerido para poder hacer la misión para finalmente crearse y tener un ID de misión

```
--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: 5

--- Crear Misión ---
Rango de misión (D, C, B, A, S): C
Recompensa: 500
Rango mínimo requerido (GENIN, CHUNIN, JONIN): GENIN
Misión creada con ID automática: 1
```

Opción 6:

En este podemos ver que misiones hay, el rango mínimo requerido, su estado, la recompensa y a quien este asignado


```

--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: 6

--- Listado de Misiones ---
1 (C) reward:500 min:GENIN estado:PENDIENTE ninja:N/A

```

Opción 7:

Pasando a la asignación de misión se nos da a elegir el ninja que va a realizar la misión

```

--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: 7

--- Asignar Misión ---
1. 1 (C) reward:500 min:GENIN estado:PENDIENTE ninja:N/A
Seleccione misión: 1
1. Gaara (Suna, JONIN)
2. Naruto Uzumaki (Konoha, GENIN)
Seleccione ninja: 2
Misión asignada: 1 a Naruto Uzumaki

```

Opción 8:

Aquí sabremos el estado de la misión asignada, como se puede ver, la misión esta en progreso

```
--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: 8

--- Estado de Misiones ---
Misión 1: Estado: EN_PROGRESO, Ninja asignado: Naruto Uzumaki
```

O posteriormente ver si logro completar la o fracaso

```
--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: El ninja Naruto Uzumaki fracasó en la ejecución de la misión 1
```

Opción 9:

En esta opción tendremos la simulación de combate entre los ninjas que estén creados

Se nos da a elegir quienes van a pelear y finalmente se nos da el resultado

```

--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: 9

--- Simular Combate ---
1. Gaara (Suna, JONIN)
2. Naruto Uzumaki (Konoha, GENIN)
Seleccione ninja: 1
1. Gaara (Suna, JONIN)
2. Naruto Uzumaki (Konoha, GENIN)
Seleccione ninja: 2
Gaara pwr=47 vs Naruto Uzumaki pwr=490
¡Ganador: Naruto Uzumaki!

```

Opción 10:

Para finalizar tenemos la opción de exportación de datos la cual nos creara el reporte o entrega de datos.

```

--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: 10

--- Exportar Datos ---
Formato (TEXT, JSON, XML): JSON
Informe exportado a: C:\Users\Santiago Daza\Downloads\naruto-main\src\naruto\reports\reporte_2025-08-17_12-25-36.json

```

```

--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: 10

--- Exportar Datos ---
Formato (TEXT, JSON, XML): TEXT
Informe exportado a: C:\Users\Santiago Daza\Downloads\naruto-main\src\naruto\reports\reporte_2025-08-17_12-27-45.txt




```

```

--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: 10

--- Exportar Datos ---
Formato (TEXT, JSON, XML): XML
Informe exportado a: C:\Users\Santiago Daza\Downloads\naruto-main\src\naruto\reports\reporte_2025-08-17_12-27-51.xml

```

Nombre	Fecha de modificación	Tipo	Tamaño
▼ hoy			
 reporte_2025-08-17_12-27-51	17/08/2025 12:27 p. m.	Microsoft Edge HTML Doc...	1 KB
 reporte_2025-08-17_12-27-45	17/08/2025 12:27 p. m.	Documento de texto	1 KB
 reporte_2025-08-17_12-25-36	17/08/2025 12:25 p. m.	Archivo de origen JSON	1 KB

Opción 11:

Finalmente ponemos el 0 y salimos del programa

```

--- Sistema Naruto ---
1. Crear ninja (manual con Builder)
2. Crear ninja por aldea (Factory)
3. Listar ninjas
4. Entrenar ninja
5. Crear misión
6. Listar misiones
7. Asignar misión a ninja
8. Ver estado de misiones
9. Simular combate
10. Exportar datos
0. Salir
Seleccione opción: 0

Saliendo...
Proyecto creado por Dylan Gutierrez, Camilo Castillo y Santiago Daza

```

CONCLUSIONES

- El uso combinado de Factory, Builder y Visitor permitió construir un sistema modular que interpreta y muestra el mundo de Naruto, gestionando así aldeas, ninjas y misiones de forma clara y extensible.
- Con los ejemplos de creación automática de Factory se demuestra que el sistema puede generar ninjas con características base propias de cada aldea, esto facilita la creación de nuevas aldeas y ninjas sin afectar el resto del sistema.
- La creación manual con Builder evidencia que este facilita al usuario construir objetos complejos paso a paso dando así la flexibilidad y evitando la sobrecarga al código con los múltiples constructores.
- Con las exportaciones de datos realizadas se puede ver que el patrón Visitor permitió agregar una funcionalidad de reportes sin modificar las clases principales de ninjas o misión, así garantiza nuevas formas de exportación de datos.
- Los ejemplos de ejecución confirman que los patrones no solo se aplicaron de la manera apropiada, también aportaron valor práctico al sistema, haciendo que este sea más ordenado, flexible y fácil de mantener.