

Trabajo Primer Corte Arquitectura de Software

- Mateo Rendon
- Samuel Ibañez

Introducción:

El proyecto para este primer corte está inspirado en Naruto, un manga creado por Masashi Kishimoto el cual está ambientado en un mundo ficticio donde las aldeas ninjas entrenan guerreros capaces de manipular el chakra y ejecutar jutsus

En este contexto el Hokage de la Aldea Konoha ha solicitado el desarrollo de un sistema de software para gestionar ninjas, misiones, y entrenamientos, lo cual dicho sistema aplica Programación Orientada a Objetos y los patrones de diseño Factory, Builder, y Visitor, cada uno con su respectiva función en el código

Objetivos del Sistema:

El objetivo es implementar un sistema inspirado en Naruto para gestionar ninjas y misiones, también demostrar el uso de los patrones de diseño Factory, Builder y Visitor y también mantener un diseño fácil de usar y de mantener

Requerimientos Funcionales:

- Creación de ninjas y aldeas, cada ninja tiene nombre y rango el cual puede ser Genin, Chunin, Jonin y Kage, además de tener estadísticas de ataque, defensa y chakra y sus respectivos jutsus
- Gestión de misiones clasificadas por rango el cual puede ser rango D, C, B, A y S con su recompensa y requisito de rango
- Entrenamiento y Combate el cual permite entrenar ninjas para mejorar sus estadísticas y simular combates básicos
- Exportación de datos generando reportes en formato TXT y JSON

Desarrollo y Diseños Orientado a Objetos:

El sistema lo organismos en lo siguiente módulos o clases principales:

- **Ninja:** La clase Ninja es la base que modela un ninja, con sus respectivos atributos como nombre, rango, ataque, defensa, chakra y jutsus, y también sus métodos como combatir y entrenar
- **Mision:** La clase Mision representa una misión con su respectivo nombre, rango, recompensa y requisito
- **Factory:** Uno de los patrones utilizados para las fábricas concretas que crean los ninjas en su respectiva aldea, esta es utilizada en las clases llamadas KonohaFactory y SunoFactory
- **Builder:** Otro de los patrones utilizados para construir los ninjas paso a paso, este es utilizado en la clase NinjaBuilder donde permite insertar validaciones al hacer el build() y crear nuevos ninjas con sus respectivos atributos
- **Visitor:** Otro patrón de diseño utilizado en la clase ExportVisitor el cual visita las clases Ninja y Mision para hacer una respectiva lectura y extraer sus datos para que con los métodos exportar_txt() el cual escribe cada nombre y cada misión como texto y exportar_json() el cual convierte a los ninjas y las misiones en diccionarios y así tener un reporte o un archivo de su respectiva información
- **Main:** Y por último el archivo main el cual crea un menú interactivo el cual usa y integra todas las funcionalidades del sistema

Patrones de Diseño Implementados:

- Patrón Factory:

Se usa en las fábricas de cada aldea, KonohaFactory y SunoFactory, el cual encapsulan la creación de ninjas con sus atributos predefinidos

Ejemplo:

```
# Importa la clase base Ninja para crear instancias con los atributos requeridos
from ninja import Ninja

class KonohaFactory:
    # Método que crea un objeto Ninja con los atributos específicos de Naruto Uzumaki
    def crear_naruto(self):
        return Ninja(
            nombre="Naruto Uzumaki",      # Nombre del ninja
            rango="Genin",                 # Rango de Naruto
            ataque=75,                     # Nivel de ataque
            defensa=60,                    # Nivel de defensa
            chakra=100,                    # Nivel de chakra
            jutsus=["Rasengan", "Kage Bunshin no Jutsu"] # Lista de jutsus característicos
        )
```

```
===== MENÚ PRINCIPAL =====
1. Crear ninja (Factory)
2. Crear ninja personalizado (Builder)
3. Listar ninjas
4. Simular combate
5. Entrenar ninja
6. Exportar datos (TXT y JSON)
7. Salir
Seleccione una opción: 1
```

```
--- Crear Ninja con Factory ---
1. Naruto (Konoha)
2. Kakashi (Konoha)
3. Gaara (Suna)
4. Kankuro (Suna)
Seleccione una opción: 1
Ninja Naruto Uzumaki creado correctamente.
```

```
===== MENÚ PRINCIPAL =====
1. Crear ninja (Factory)
2. Crear ninja personalizado (Builder)
3. Listar ninjas
4. Simular combate
5. Entrenar ninja
6. Exportar datos (TXT y JSON)
7. Salir
Seleccione una opción: 3
```

```
--- Lista de Ninjas ---
1. Ninja: Naruto Uzumaki | Rango: Genin | Stats -> Ataque: 75, Defensa: 60, Chakra: 100 | Jutsus: Rasengan, Kage Bunshin no Jutsu
```

Vemos en el ejemplo como el sistema ha creado el ninja Naruto utilizando el patrón Factory

- Patrón Builder:

Se usa en NinjaBuilder el cual permite crear ninja personalizados de manera fácil y personalizando paso a paso sus atributos y jutsus

Ejemplo:

```
===== MENÚ PRINCIPAL =====
1. Crear ninja (Factory)
2. Crear ninja personalizado (Builder)
3. Listar ninjas
4. Simular combate
5. Entrenar ninja
6. Exportar datos (TXT y JSON)
7. Salir
Seleccione una opción: 2
```

```

--- Crear Ninja Personalizado ---
Nombre: Samuel
Rango (Genin, Chunin, Jonin): Jonin
Ataque: 100
Defensa: 100
Chakra: 100
Agregar jutsu (o 'fin' para terminar): Rasengan
Agregar jutsu (o 'fin' para terminar): Chidori
Agregar jutsu (o 'fin' para terminar): fin

```

```

4. Simular combate
5. Entrenar ninja
6. Exportar datos (TXT y JSON)
7. Salir
Seleccione una opción: 3

--- Lista de Ninjas ---
1. Ninja: Naruto Uzumaki | Rango: Genin | Stats -> Ataque: 75, Defensa: 60, Chakra: 100 | Jutsus: Rasengan, Kage Bunshin no Jutsu
2. Ninja: Samuel | Rango: Jonin | Stats -> Ataque: 100, Defensa: 100, Chakra: 100 | Jutsus: Rasengan, Chidori

```

Podemos ver como en el ejemplo cómo usamos el patrón de diseño builder para crear al ninja Samuel con sus respectivos atributos

- Patrón Visitor:

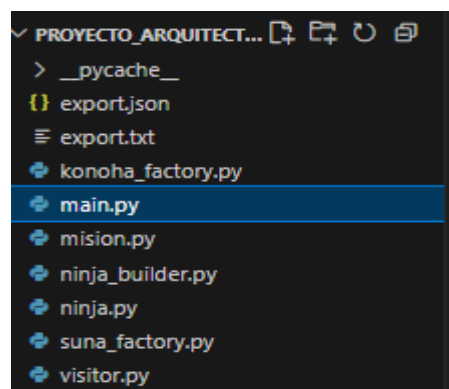
Se usa en el ExportVisitor el cual recorre la lista de ninjas y misiones, exportando los datos a distintos formatos sin modificar las clase base

Ejemplo:

```

===== MENÚ PRINCIPAL =====
1. Crear ninja (Factory)
2. Crear ninja personalizado (Builder)
3. Listar ninjas
4. Simular combate
5. Entrenar ninja
6. Exportar datos (TXT y JSON)
7. Salir
Seleccione una opción: 6

```



```

PROYECTO_ARQUITECT...
> __pycache__
{} export.json
≡ export.txt
konoha_factory.py
main.py
mision.py
ninja_builder.py
ninja.py
suna_factory.py
visitor.py

```

```

() export,json > ...
1  {
2    "ninjas": [
3      {
4        "nombre": "Naruto Uzumaki",
5        "rango": "Genin",
6        "ataque": 75,
7        "defensa": 60,
8        "chakra": 100,
9        "jutsus": [
10         "Rasengan",
11         "Kage Bunshin no Jutsu"
12       ]
13     },
14     {
15       "nombre": "Samuel",
16       "rango": "Jonin",
17       "ataque": 100,
18       "defensa": 100,
19       "chakra": 100,
20       "jutsus": [
21         "Rasengan",
22         "Chidori"
23       ]
24     }
25   ],
26   "misiones": [
27     {
28       "nombre": "Rescate",
29       "rango": "B",
30       "recompensa": 300,
31       "requisito_rango": "Chunin"
32     },
33     {
34       "nombre": "Defensa de tu pueblo",
35       "rango": "A",
36       "recompensa": 500,
37       "requisito_rango": "Jonin"
38     }
39   ]
40 }

```

```

≡ export.txt
1  === NINJAS ===
2  Ninja: Naruto Uzumaki | Rango: Genin | Stats -> Ataque: 75, Defensa: 60, Chakra: 100 | Jutsus: Rasengan, Kage Bunshin no Jutsu
3  Ninja: Samuel | Rango: Jonin | Stats -> Ataque: 100, Defensa: 100, Chakra: 100 | Jutsus: Rasengan, Chidori
4
5  === MISIONES ===
6  Misión: Rescate | Rango: B | Recompensa: 300
7  Misión: Defensa de tu pueblo | Rango: A | Recompensa: 500
8

```

Vemos en el ejemplo cómo usamos el patrón Visitor para extraer la información en formato TXT y en formato JSON

Ejemplo de ejecución

```
===== MENÚ PRINCIPAL =====
1. Crear ninja (Factory)
2. Crear ninja personalizado (Builder)
3. Listar ninjas
4. Simular combate
5. Entrenar ninja
6. Exportar datos (TXT y JSON)
7. Salir
Seleccione una opción: █
```

1:

```
--- Crear Ninja con Factory ---
1. Naruto (Konoha)
2. Kakashi (Konoha)
3. Gaara (Suna)
4. Kankuro (Suna)
Seleccione una opción: █
```

2:

```
Seleccione una opción: 2

--- Crear Ninja Personalizado ---
Nombre: Mateoreca
Rango (Genin, Chunin, Jonin): Genin
Ataque: 10000
Defensa: 10000
Chakra: 10000
Agregar jutsu (o 'fin' para terminar): Golpe Letal
Agregar jutsu (o 'fin' para terminar): fin
Ninja Mateoreca creado correctamente.
```

3:

```
Seleccione una opción: 3

--- Lista de Ninjas ---
1. Ninja: Naruto Uzumaki | Rango: Genin | Stats -> Ataque: 75, Defensa: 60, Chakra: 100 | Jutsus: Rasengan, Kage Bunshin no Jutsu
2. Ninja: Kakashi Hatake | Rango: Jonin | Stats -> Ataque: 85, Defensa: 70, Chakra: 90 | Jutsus: Chidori, Sharingan, Raikiri
3. Ninja: Mateoreca | Rango: Genin | Stats -> Ataque: 10000, Defensa: 10000, Chakra: 10000 | Jutsus: Golpe Letal
```

4:

```
Seleccione una opción: 4

--- Lista de Ninjas ---
1. Ninja: Naruto Uzumaki | Rango: Genin | Stats -> Ataque: 75, Defensa: 60, Chakra: 100 | Jutsus: Rasengan, Kage Bunshin no Jutsu
2. Ninja: Kakashi Hatake | Rango: Jonin | Stats -> Ataque: 85, Defensa: 70, Chakra: 90 | Jutsus: Chidori, Sharingan, Raikiri
3. Ninja: Mateoreca | Rango: Genin | Stats -> Ataque: 10000, Defensa: 10000, Chakra: 10000 | Jutsus: Golpe Letal
Seleccione el número del primer ninja: 1
Seleccione el número del segundo ninja: 3
Mateoreca gana el combate contra Naruto Uzumaki
```

5:

```

Seleccione una opción: 5

--- Lista de Ninjas ---
1. Ninja: Naruto Uzumaki | Rango: Genin | Stats -> Ataque: 75, Defensa: 60, Chakra: 100 | Jutsus: Rasengan, Kage Bunshin no Jutsu
2. Ninja: Kakashi Hatake | Rango: Jonin | Stats -> Ataque: 85, Defensa: 70, Chakra: 90 | Jutsus: Chidori, Sharingan, Raikiri
3. Ninja: Mateoreca | Rango: Genin | Stats -> Ataque: 10000, Defensa: 10000, Chakra: 10000 | Jutsus: Golpe Letal
Seleccione el número del ninja a entrenar: 1
Incremento de ataque: 100
Incremento de defensa: 200
Incremento de chakra: 1000
Naruto Uzumaki entrenado correctamente.

```

6:

```

Seleccione una opción: 6

=== EXPORTACIÓN COMPLETA ===
Archivos generados en:
C:\Users\mateo\OneDrive\Ing Sistemas\Codigos\export.txt
C:\Users\mateo\OneDrive\Ing Sistemas\Codigos\export.json

```

```

export.txt X
export.txt
1  === NINJAS ===
2  Ninja: Naruto Uzumaki | Rango: Genin | Stats -> Ataque: 175, Defensa: 260, Chakra: 1100 | Jutsus: Rasengan, Kage Bunshin no Jutsu
3  Ninja: Kakashi Hatake | Rango: Jonin | Stats -> Ataque: 85, Defensa: 70, Chakra: 90 | Jutsus: Chidori, Sharingan, Raikiri
4  Ninja: Mateoreca | Rango: Genin | Stats -> Ataque: 10000, Defensa: 10000, Chakra: 10000 | Jutsus: Golpe Letal
5
6  === MISIONES ===
7  Misión: Rescate | Rango: B | Recompensa: 300
8  Misión: Defensa de tu pueblo | Rango: A | Recompensa: 500
9

```

7:

```

Seleccione una opción: 7
Fin

```

- Conclusiones:

Podemos concluir que este Proyecto de primer corte refleja cómo aplicar Tres patrones de diseño esenciales dentro de un sistema inspirado en Naruto, podemos ver que el patrón Factory permite encapsular la creación de Ninjas según su aldea, el patrón Builder también ayudó a la creación de Ninjas personalizados y el patrón Visitor ayudó a exportar la información indicada del respectivo sistema, y para terminar vemos como los patrones de diseño nos ayudan a generar un sistema más ágil y flexible con sus respectivas funciones