



“Informe de ejecución de la implementación de los patrones de diseño Composite, Visitor y Singleton”

Programación Avanzada



Ing. David Santiago Velásquez Cifuentes.¹
Maestría en Ingeniería. Facultad de Ingeniería y Ciencias Básicas.
Fundación Universitaria Los Libertadores.

1. Introducción

En este informe, se presenta una detallada ejecución de la implementación de tres patrones de diseño clave en el desarrollo de software: el Patrón Composite, el Patrón Visitor y el Patrón Singleton. Cada uno de estos patrones desempeña un papel crucial en la arquitectura de software, y su correcta implementación es esencial para lograr sistemas eficientes y mantenibles.

El objetivo principal de este informe es demostrar la correcta implementación de estos patrones de diseño y su funcionamiento eficiente a través de capturas de pantalla. Estas capturas de pantalla ilustrarán la ejecución exitosa de los códigos correspondientes a cada patrón, validando su correcta funcionalidad. Además de la ejecución de los códigos, se prestará especial atención al proceso de carga de los mismos en un repositorio de GitHub. A través de capturas de pantalla, se documentará cada paso necesario para añadir, confirmar y cargar los archivos relacionados con la implementación de estos patrones en el repositorio GitHub.

2. Patrón Composite

Se utiliza para componer objetos en estructuras de árbol para representar jerarquías de objetos. Esto permite a los clientes tratar tanto a los objetos individuales como a las composiciones de objetos de manera uniforme. En otras palabras, se puede trabajar con elementos individuales y colecciones de elementos de la misma manera.

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22621.2428]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Acer>cd C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Composite

C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Composite>javac ComponenteVehiculo.java

C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Composite>javac VehiculoIndividual.java

C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Composite>javac GrupoVehiculos.java

C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Composite>javac Principal.java

C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Composite>java Principal
Grupo de Vehículos: Vehículos
Grupo de Vehículos: Deportivos
Vehículo: Automóvil Deportivo
Vehículo: Motocicleta
Vehículo: Automóvil Familiar

C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Composite>
```

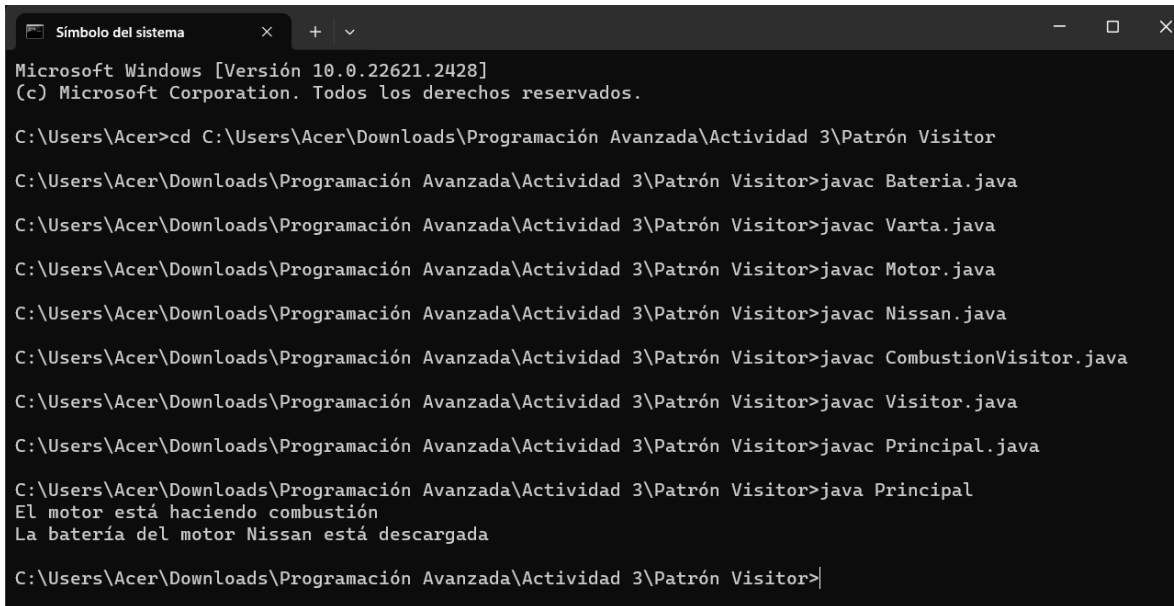
Imagen 1 – Compilación y ejecución del patrón Composite en CMD

3. Patrón Visitor

Se emplea para separar los algoritmos de la estructura de los objetos a los que se aplican. Permite definir una nueva operación sin cambiar las clases de los elementos sobre los que opera. Esto es particularmente útil cuando se tienen muchas clases diferentes con operaciones comunes.

¹ Email: dsvelasquezc@ulibertadores.edu.co

Actividad 3 – “Informe de ejecución de la implementación de los patrones de diseño Composite, Visitor y Singleton”



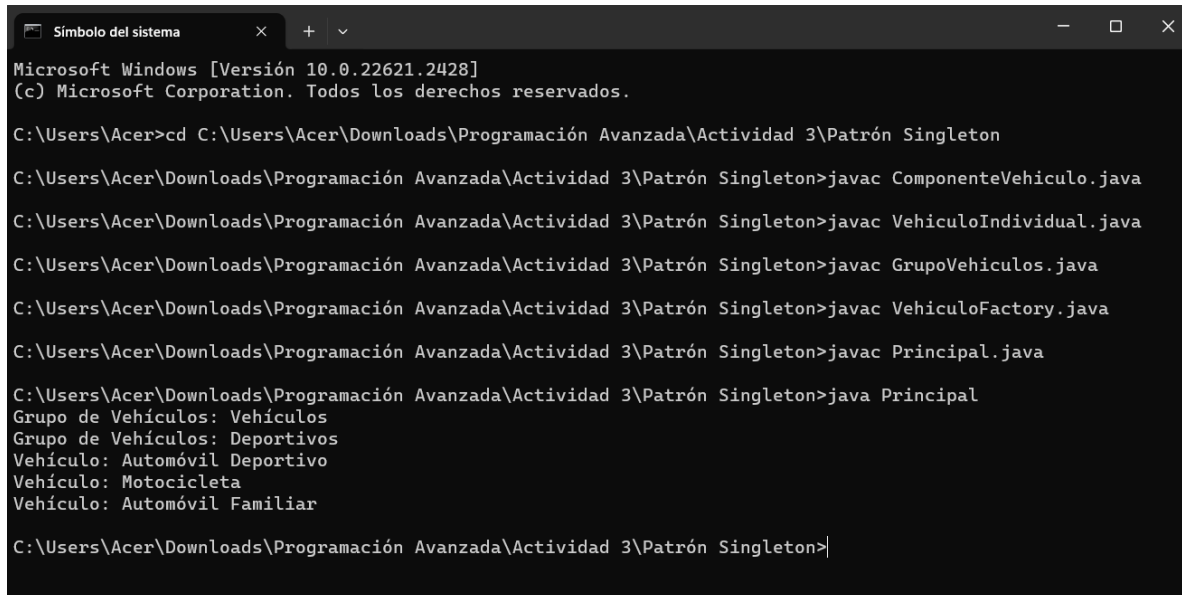
```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22621.2428]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Acer>cd C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Visitor
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Visitor>javac Bateria.java
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Visitor>javac Varta.java
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Visitor>javac Motor.java
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Visitor>javac Nissan.java
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Visitor>javac CombustionVisitor.java
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Visitor>javac Visitor.java
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Visitor>javac Principal.java
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Visitor>java Principal
El motor está haciendo combustión
La batería del motor Nissan está descargada
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Visitor>
```

Imagen 2 - Compilación y ejecución del patrón Visitor en CMD

4. Patrón Singleton

Garantiza que una clase tenga una única instancia y proporciona un punto de acceso global a esa instancia. Esto es útil cuando se necesita compartir una sola instancia de una clase en todo el sistema, como un registro central de eventos o una conexión a una base de datos.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22621.2428]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Acer>cd C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Singleton
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Singleton>javac ComponenteVehiculo.java
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Singleton>javac VehiculoIndividual.java
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Singleton>javac GrupoVehiculos.java
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Singleton>javac VehiculoFactory.java
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Singleton>javac Principal.java
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Singleton>java Principal
Grupo de Vehículos: Vehículos
Grupo de Vehículos: Deportivos
Vehículo: Automóvil Deportivo
Vehículo: Motocicleta
Vehículo: Automóvil Familiar
C:\Users\Acer\Downloads\Programación Avanzada\Actividad 3\Patrón Singleton>
```

Imagen 3 - Compilación y ejecución del patrón Singleton en CMD

5. Carga de la actividad al repositorio en GITHUB

Los archivos de esta actividad han sido añadidos al repositorio personal (dsvelasquezc) y están disponibles en: https://github.com/dsvelasquezc/Actividad_3_Patrones.git

Actividad 3 – “Informe de ejecución de la implementación de los patrones de diseño Composite, Visitor y Singleton”

```
MINGW64:/c/Users/Acer/Downloads/Programación Avanzada/Actividad 3
Acer@DavidSantiagoVC MINGW64 ~/Downloads/Programación Avanzada/Actividad 3 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
  (commit or discard the untracked or modified content in submodules)
        modified:   "Patr\303\263n Visitor" (modified content)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        "Patr\303\263n Composite/"
        "Patr\303\263n Singleton/"

no changes added to commit (use "git add" and/or "git commit -a")

Acer@DavidSantiagoVC MINGW64 ~/Downloads/Programación Avanzada/Actividad 3 (master)
$ git add .

Acer@DavidSantiagoVC MINGW64 ~/Downloads/Programación Avanzada/Actividad 3 (master)
$ git commit -m "Agregar actividad 3 patrones"
[master 931335d] Agregar actividad 3 patrones
18 files changed, 139 insertions(+)
create mode 100644 "Patr\303\263n Composite/ComponenteVehiculo.class"
create mode 100644 "Patr\303\263n Composite/ComponenteVehiculo.java"
create mode 100644 "Patr\303\263n Composite/GrupoVehiculos.class"
create mode 100644 "Patr\303\263n Composite/GrupoVehiculos.java"
create mode 100644 "Patr\303\263n Composite/Principal.class"
create mode 100644 "Patr\303\263n Composite/Principal.java"
create mode 100644 "Patr\303\263n Composite/VehiculoIndividual.class"
create mode 100644 "Patr\303\263n Composite/VehiculoIndividual.java"
create mode 100644 "Patr\303\263n Singleton/ComponenteVehiculo.class"
create mode 100644 "Patr\303\263n Singleton/ComponenteVehiculo.java"
create mode 100644 "Patr\303\263n Singleton/GrupoVehiculos.class"
create mode 100644 "Patr\303\263n Singleton/GrupoVehiculos.java"
create mode 100644 "Patr\303\263n Singleton/Principal.class"
create mode 100644 "Patr\303\263n Singleton/Principal.java"
create mode 100644 "Patr\303\263n Singleton/VehiculoFactory.class"
create mode 100644 "Patr\303\263n Singleton/VehiculoFactory.java"
create mode 100644 "Patr\303\263n Singleton/VehiculoIndividual.class"
create mode 100644 "Patr\303\263n Singleton/VehiculoIndividual.java"

Acer@DavidSantiagoVC MINGW64 ~/Downloads/Programación Avanzada/Actividad 3 (master)
$ git branch -M main

Acer@DavidSantiagoVC MINGW64 ~/Downloads/Programación Avanzada/Actividad 3 (main)
$ git remote add origin https://github.com/dsvelasquezc/Actividad_3_Patrones.git

Acer@DavidSantiagoVC MINGW64 ~/Downloads/Programación Avanzada/Actividad 3 (main)
$ git push -u origin main
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 8 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (18/18), 4.66 KiB | 1.16 MiB/s, done.
Total 18 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/dsvelasquezc/Actividad_3_Patrones.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Acer@DavidSantiagoVC MINGW64 ~/Downloads/Programación Avanzada/Actividad 3 (main)
$
```

Imagen 4– Proceso de carga al repositorio mediante líneas de código

Actividad 3 – “Informe de ejecución de la implementación de los patrones de diseño Composite, Visitor y Singleton”

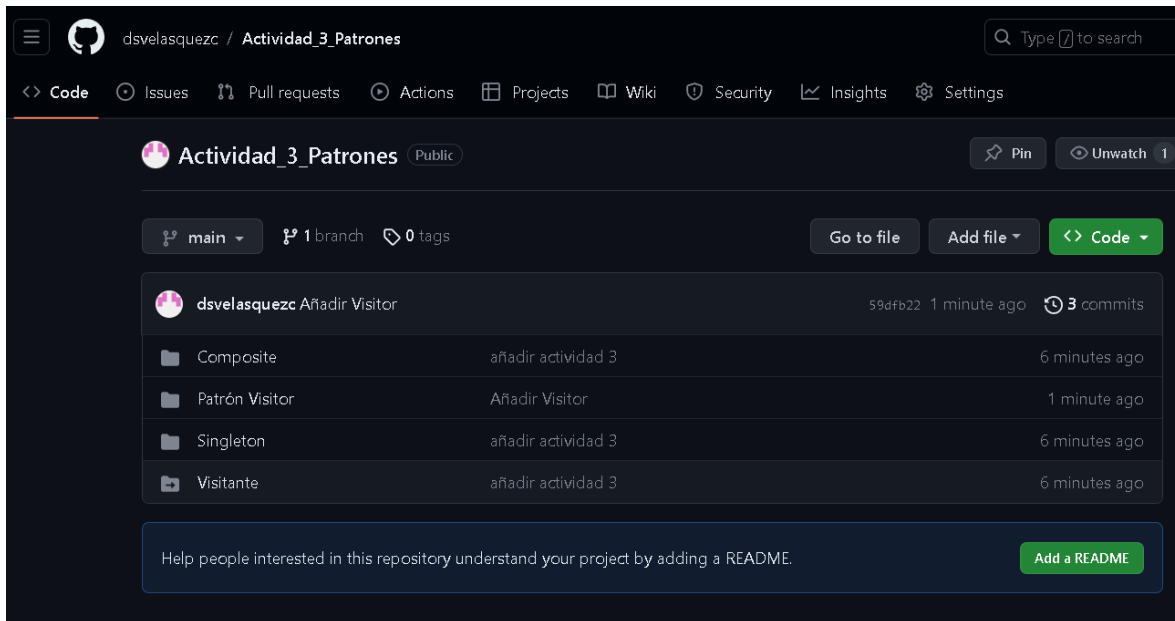


Imagen 5 – Confirmación de la creación del repositorio y la carga exitosa de los documentos