



Instituto Politecnico Nacional

Escuela Superior de Cómputo



Bocanegra Heziquio Yestlanezi

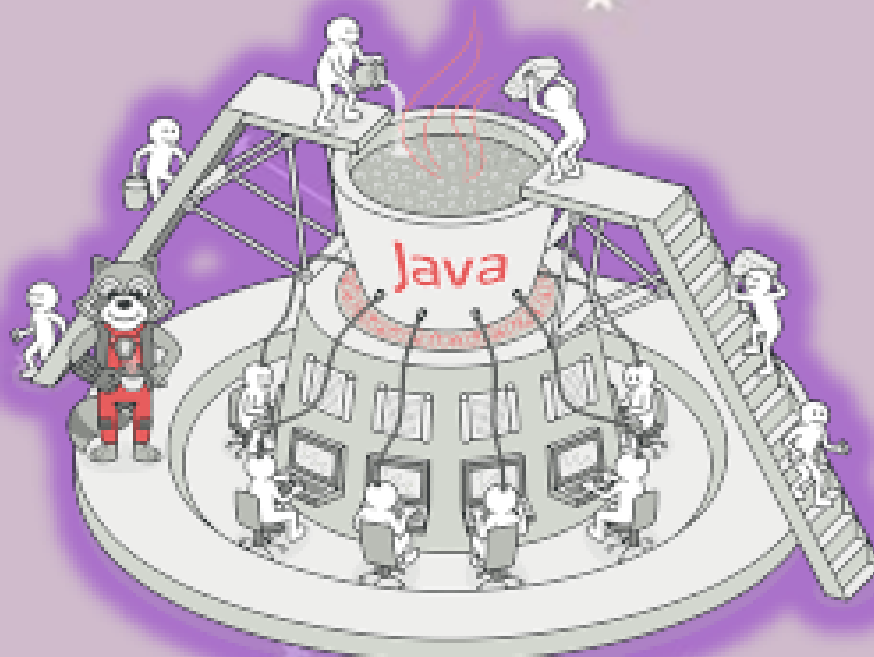
Tarea 3

Patrones de diseño

Fecha: 13 de junio de 2021

2CM13

Programación Orientada a Objetos



Contenido

Patrones de diseño	3
Problemática	3
Ejemplos de patrones de diseño de tipo 'Estructural'	3
Ejemplos de patrones de diseño de tipo 'Comportamiento'	4
★ Patrón Abstrac Factory	4
Ejemplo	4
Estructura	5
Ejemplo en java	6
Codigo	6
Patrón Builder	7
Diagrama estructura	7
Ejemplo con código	8
Patrón Factory Method	9
Diagrama de estructura	9
Ejemplo código	10
Patrón Prototype	11
Diagrama de estructura	11
Bibliografía	11

Patrones de diseño

Los patrones de construcción tienen la vocación de abstraer los mecanismos de creación de objetos. Un sistema que utilice estos patrones se vuelve independiente de la forma en que se crean los objetos, en particular, de los mecanismos ...

Aunque se pretende que los patrones de diseño sean independientes de los lenguajes de programación, es muy útil para mostrar reconocerlos y entenderlos usar implementaciones en diferentes idiomas.

Problemática

En la mayoría de lenguajes orientados a objetos, la creación de objetos se realiza gracias al mecanismo de instanciación, que consiste en crear un nuevo objeto mediante la llamada al operador new configurado para una clase (y eventualmente los argumentos del constructor de la clase cuyo objetivo es proporcionar a los atributos su valor inicial). Tal objeto es, por consiguiente, una instancia de esta clase.

Los lenguajes de programación más utilizados a día de hoy, como Java, C++ o C#, utilizan el mecanismo del operador new.

En Java, una instrucción de creación de un objeto puede escribirse de la siguiente manera:

```
objeto = new Clase();
```

En ciertos casos es necesario configurar la creación de objetos. Tomemos el ejemplo de un método construyeDoc que crea los documentos. Puede construir documentos PDF, RTF o HTML. Generalmente el tipo de documento a crear se pasa como parámetro al método mediante una cadena de caracteres, y se obtiene el código siguiente:

```
public Documento construyeDoc(String ...
```

Ejemplos de patrones de diseño de tipo 'Estructural'

Según Wikipedia, "los patrones de diseño estructural son los patrones de diseño que facilitan el diseño mediante la identificación de una forma sencilla de darse cuenta de las relaciones entre las entidades."

Aquí están los enlaces a los patrones de diseño de este tipo:

- * [Adapter Design Pattern](#)
- [Bridge Design Pattern](#)
- [Composite Design Pattern](#)
- * [Decorator Design Pattern](#)
- [Facade Design Pattern](#)
- [Flyweight Design Pattern](#)
- [Private Class Data](#)
- [Proxy Design Pattern](#)

Ejemplos de patrones de diseño de tipo 'Comportamiento'

Según Wikipedia, "los patrones de diseño de comportamiento son los patrones de diseño que identifican patrones comunes de comunicación entre los objetos y se dan cuenta de estos patrones. Al hacerlo, estos patrones de aumentar la flexibilidad en la realización de esta comunicación".

Los patrones de diseño de este tipo son:

- * Chain of Responsibility
- Command Design Pattern
- Interpreter Design Pattern
- Iterator Design Pattern
- Mediator Design Pattern
- Memento Design Pattern
- Null Object Design Pattern
- Observer Design Pattern
- State Design Pattern
- Strategy Design Pattern
- Template Method Design Pattern
- Visitor Design Pattern

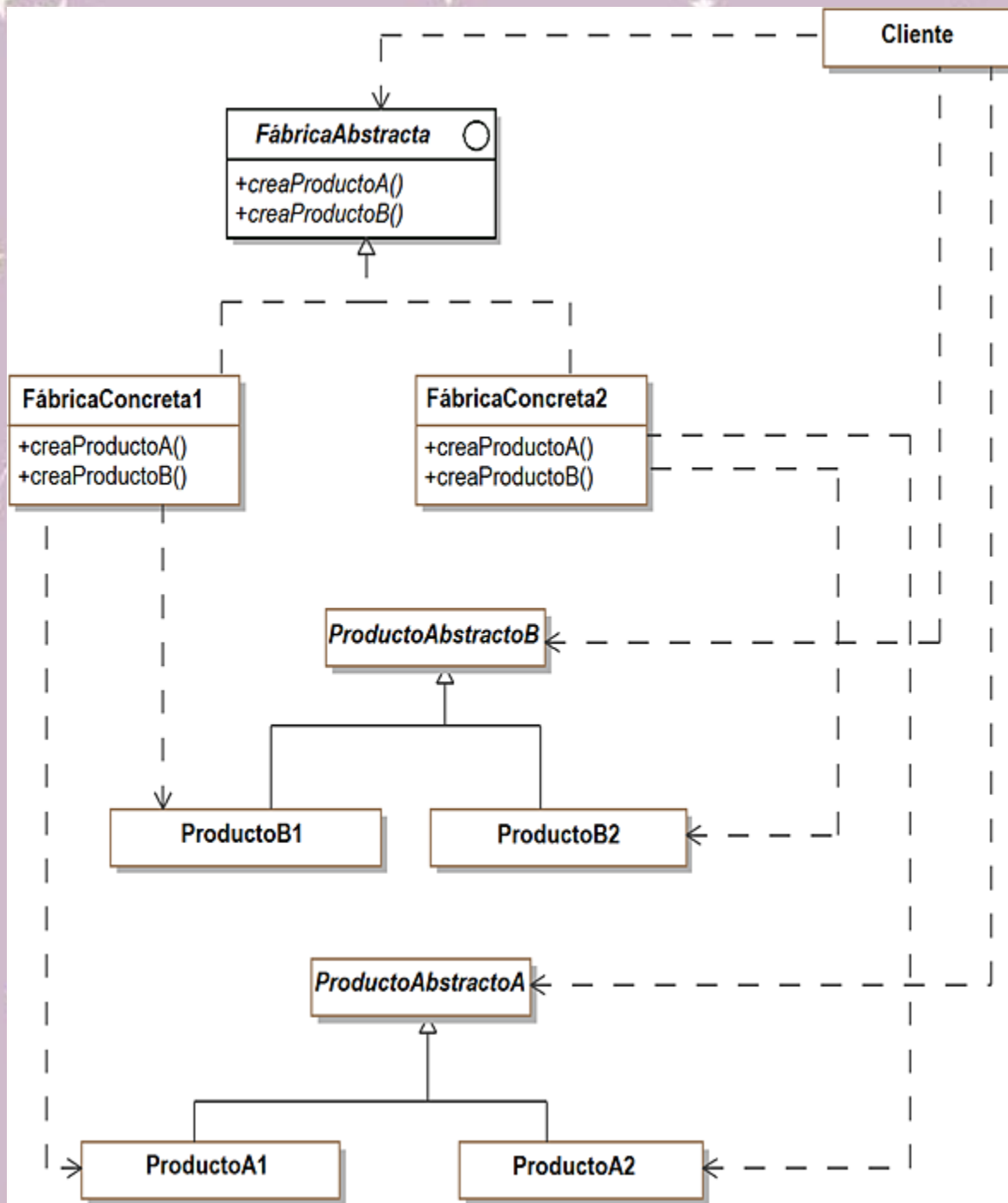
Patrón Abstrac Factory

Ejemplo

El sistema de venta de vehículos gestiona vehículos que funcionan con gasolina y vehículos eléctricos. Esta gestión está delegada en el objeto Catálogo encargado de crear tales objetos.

Para cada producto, disponemos de una clase abstracta, de una subclase concreta derivando una versión del producto que funciona con gasolina y de una subclase concreta derivando una versión del producto que funciona con electricidad. Por ejemplo, en la figura 4.1, para el objeto Scooter, existe una clase abstracta Scooter y dos subclases concretas ScooterElectricidad y ScooterGasolina.

Estructura



Ejemplo en java

Presentamos a continuación un pequeño ejemplo de uso del patrón escrito en Java. El código Java correspondiente a la clase abstracta Automovil y sus subclases aparece a continuación. Es muy sencillo, describe los cuatro atributos de los automóviles así como el método mostrarCaracteristicas que permite visualizarlas.

Codigo

```
public abstract class Automovil
{
    protected String modelo;
    protected String color;
    protected int potencia;
    protected double espacio;

    public Automovil(String modelo, String color, int
        potencia, double espacio)
    {
        this.modelo = modelo;
        this.color = color;
        this.potencia = potencia;
        this.espacio = espacio;
    }

    public abstract void mostrarCaracteristicas();
}

public class AutomovilElectricidad extends Automovil
{
    public AutomovilElectricidad(String modelo, String
        color, int potencia, double espacio)
    {
        super(modelo, color, potencia, espacio);
    }

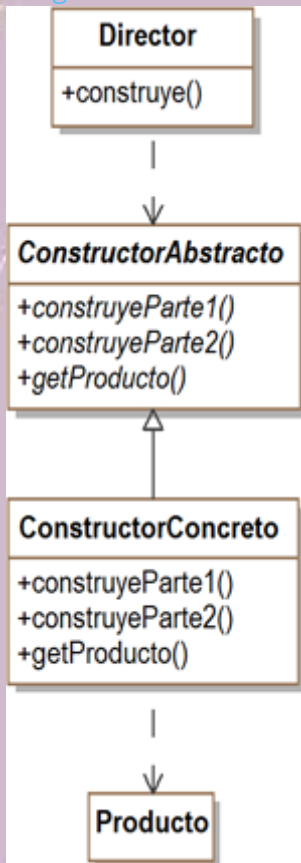
    public void mostrarCaracteristicas()
```

```
{  
    System.out.println(  
        "Automovil electrico de modelo: " ...
```

Patrón Builder

El objetivo del patrón Builder es abstraer la construcción de objetos

Diagrama estructura



Ejemplo con código

```
import java.util.*;

public abstract class Documentacion
{
    protected List<String> contenido =
        new ArrayList<String>();

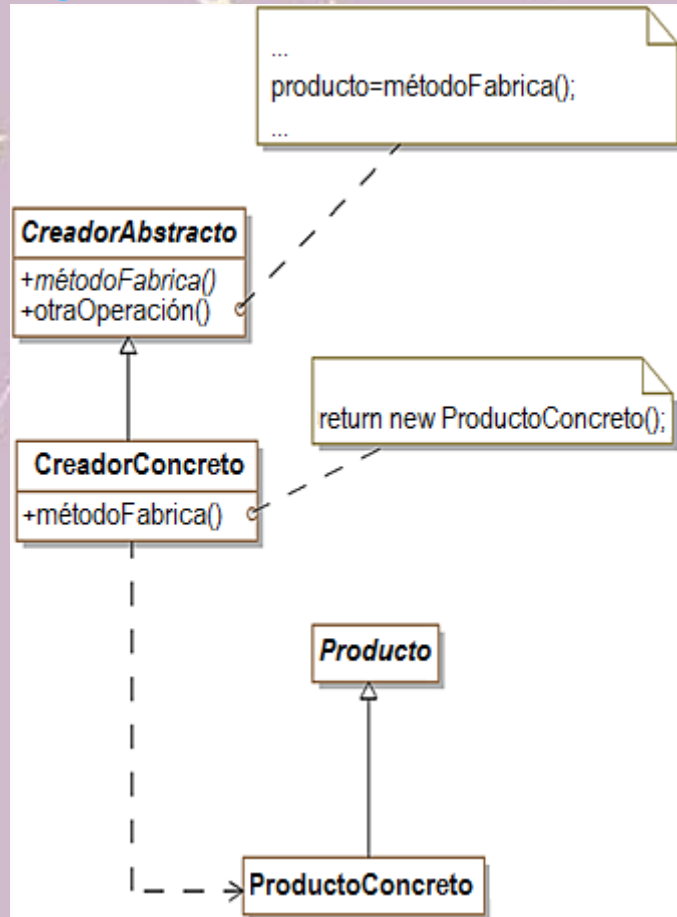
    public abstract void agregaDocumento(String documento);
    public abstract void imprime();
}

public class DocumentacionHtml extends Documentacion
{
    public void agregaDocumento(String documento)
    {
        if (documento.startsWith("<HTML>"))
            contenido.add(documento);
    }

    public void imprime()
    {
        System.out.println("Documentacion HTML");
        for (String s: contenido)
            System.out.println(s);
    }
    ...
}
```


Patrón Factory Method

Diagrama de estructura



Ejemplo código

```
public abstract class Pedido
{
    protected double importe;

    public Pedido(double importe)
    {
        this.importe = importe;
    }

    public abstract boolean valida();

    public abstract void paga();
}

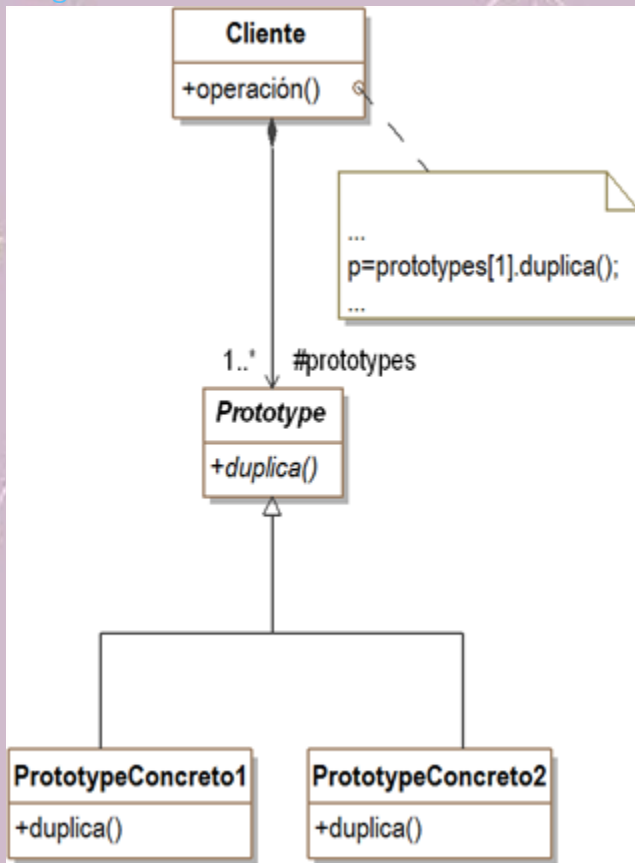
public class PedidoContado extends Pedido
{
    public PedidoContado(double importe)
    {
        super(importe);
    }

    public void paga()
    {
        System.out.println(
            "El pago del pedido por importe de: " +
            importe + " se ha realizado.");
    }

    public boolean valida()
    {
        return true;    ...
    }
}
```

Patrón Prototype

Diagrama de estructura



Bibliografía

- [1] J. R. Guerrero, «WordPress.com,» 26 agosto 2014. [En línea]. Available: <https://jesusramirezguerrero.com/2014/08/26/patrones-de-diseno-en-java/>. [Último acceso: 13 junio 2021].