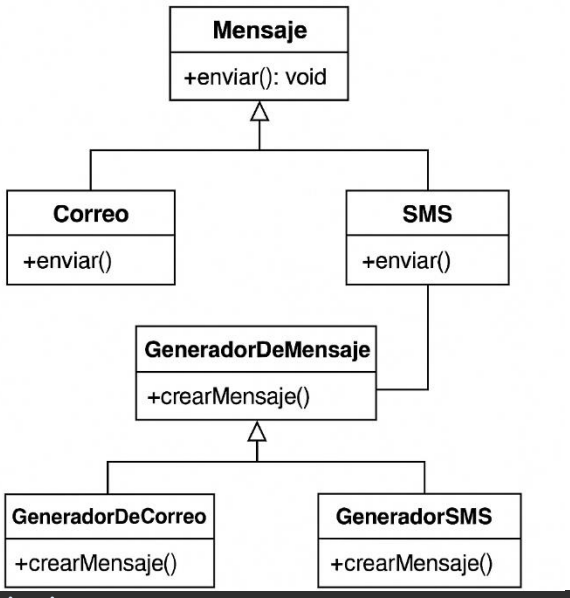
PRACTICA 12

INF-121

Nombre: Mariam Portillo Ariscurinaga

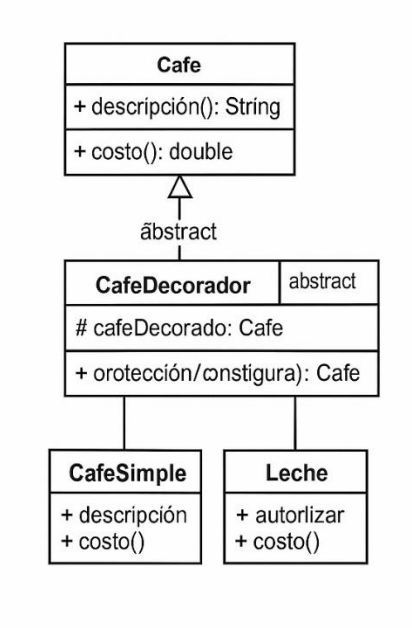
PATRÓN CREACIONAL: FACTORY METHOD




```java
package SistemaDeMensajeria;

public interface Mensaje {
	void enviar();

}
public class Correo implements Mensaje{
	public void enviar() {
		System.out.println("Enviando correo electronico....");
	}

}
public class SMS implements Mensaje {
	public void enviar() {
		System.out.println("Enviando SMS...");
	}

}
public abstract class GeneradorDeMensaje {
	public abstract Mensaje crearMensaje();

}
public class GeneradorDeCorreo extends GeneradorDeMensaje {
    public Mensaje crearMensaje() {
        return new Correo();
```

```
    }
}
public class GeneradorSMS extends GeneradorDeMensaje {
    @Override
    public Mensaje crearMensaje() {
        return new SMS();
    }
}
package SistemaDeMensajeria;

public class Main {
    public static void main(String[] args) {
        GeneradorDeMensaje generadorCorreo = new GeneradorDeCorreo();
        Mensaje correo = generadorCorreo.crearMensaje();
        correo.enviar();

        GeneradorDeMensaje generadorSMS = new GeneradorSMS();
        Mensaje sms = generadorSMS.crearMensaje();
        sms.enviar();
    }
}
```

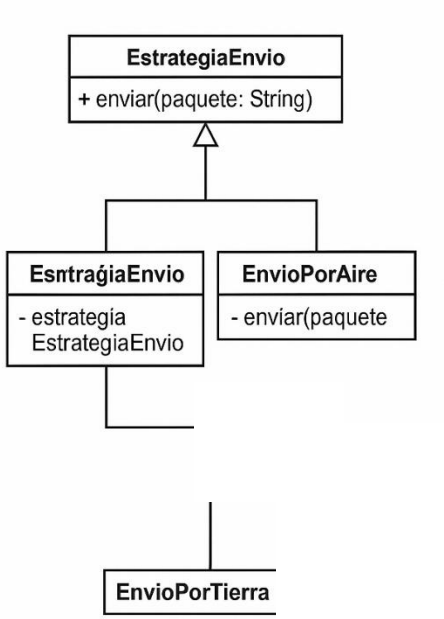
PATRÓN ESTRUCTURAL




```
package SistemaDePedidosDeCafe;

public interface Cafe {
    String descripcion();
    double costo();
}

public abstract class CafeDecorador implements Cafe {
    protected Cafe cafeDecorado;
```

```java
    public CafeDecorador(Cafe cafe) {
        this.cafeDecorado = cafe;
    }

    public String descripcion() {
        return cafeDecorado.descripcion();
    }
    public double costo() {
        return cafeDecorado.costo();
    }
}

public class CafeSimple implements Cafe {
    public String descripcion() {
        return "Café simple";
    }
    public double costo() {
        return 5.0;
    }
}

public class Leche extends CafeDecorador {
    public Leche(Cafe cafe) {
        super(cafe);
    }

    public String descripcion() {
        return super.descripcion() + ", con leche";
    }

    public double costo() {
        return super.costo() + 1.5;
    }
}
public class Main {
    public static void main(String[] args) {
        Cafe miCafe = new CafeSimple();
        miCafe = new Leche(miCafe);

        System.out.println(miCafe.descripcion()); // Café simple, con leche
        System.out.println("Costo: " + miCafe.costo()); // 6.5
    }
}
```

PATRÓN DE COMPORTAMIENTO




```java
package SistemaDeEnvio;

public interface EstrategiaEnvio {
    void enviar(String paquete);
}
public class ServicioEnvio {
    private EstrategiaEnvio estrategia;

    public ServicioEnvio(EstrategiaEnvio estrategia) {
        this.estrategia = estrategia;
    }

    public void setEstrategia(EstrategiaEnvio estrategia) {
        this.estrategia = estrategia;
    }

    public void enviarPaquete(String paquete) {
        estrategia.enviar(paquete);
    }
}
public class EnvioPorAire implements EstrategiaEnvio {
    public void enviar(String paquete) {
        System.out.println("Enviando " + paquete + " por aire.");
    }
}
public class EnvioPorTierra implements EstrategiaEnvio {
    public void enviar(String paquete) {
        System.out.println("Enviando " + paquete + " por tierra.");
    }
}
public class Main {
```

```
    public static void main(String[] args) {
        ServicioEnvio envio = new ServicioEnvio(new EnvioPorTierra());
        envio.enviarPaquete("Paquete1");

        envio.setEstrategia(new EnvioPorAire());
        envio.enviarPaquete("Paquete2");
    }
}
```