Command Pattern

Usuario.java

```
package uam.patrones.command.clases;

public class Usuario{

  private String login;
  private String password;

  public Usuario(String u, String p) {
    login = u;
    password = p;
  }



  public String getLogin() {
    return login;
  }

  public void setLogin( String login ){
    this.login = login;
  }


  public String getPassword(){
    return password;
  }


  public void setPassword( String password ){
    this.password = password;
  }



}
```

Comando.java

```
package uam.patrones.command.operaciones;


public abstract class Comando{

  private String usuario;
  private String password;

  public abstract boolean execute();

  public void setUsuario( String usuario ) {
    this.usuario = usuario;
  }
```

```
  public void setPassword( String password ) {
    this.password = password;
  }

  public String getUsuario() {
    return usuario;
  }

  public String getPassword(){
    return password;
  }

}
```

Invocador.java

```
package uam.patrones.command.operaciones;

public class Invocador {

  Comando comando;

  public void asignarComando(Comando c) {
    comando = c;
  }

  public void ejecutarComando() {
    comando.execute();
  }

}
```

RegistrarLog.java

```
package uam.patrones.command.operaciones;

import uam.patrones.command.clases.Usuario;

public class RegistrarLog {

  public void registrarUsuario(Usuario usuario) {
    System.out.println("Registrando en el log al usuario " + usuario.getLogin());
  }

}
```

ValidarUsuario.java

```java
package uam.patrones.command.operaciones;

import uam.patrones.command.clases.Usuario;

public class ValidarUsuario{

  private Usuario usuario;

  public Usuario validarUsuarioBD(String usr, String pwd) {
    System.out.println("Validando al usuario en BD");
    usuario = new Usuario(usr,pwd);
    return usuario;
  }

  public Usuario validarUsuarioSW(String usr, String pwd) {
    System.out.println("Validando al usuario en un Servicio Web");
    usuario = new Usuario(usr,pwd);
    return usuario;
  }

}
```

ValidarUsuarioBD.java

```java
package uam.patrones.command.operaciones;

import uam.patrones.command.clases.Usuario;

public class ValidarUsuarioBD extends Comando{

  private ValidarUsuario validacion;
  private RegistrarLog registro;



  @Override
  public boolean execute(){
    validacion = new ValidarUsuario();
    registro = new RegistrarLog();
    Usuario usr = validacion.validarUsuarioBD(getUsuario(),getPassword());
    registro.registrarUsuario( usr );
    return true;
  }




}
```

ValidarUsuarioSW.java

```java
package uam.patrones.command.operaciones;

import uam.patrones.command.clases.Usuario;


public class ValidarUsuarioSW extends Comando{

  private ValidarUsuario validacion;
  private RegistrarLog registro;


  @Override
  public boolean execute(){
    validacion = new ValidarUsuario();
    registro = new RegistrarLog();
    Usuario usr = validacion.validarUsuarioSW( getUsuario(),getPassword() );
    registro.registrarUsuario( usr );
    return true;
  }

}
```

Principal.java

```java
package uam.patornes.command.principal;

import uam.patrones.command.operaciones.*;

public class Principal{

  public static void main( String[] args ){

    Invocador invoca = new Invocador();
    ValidarUsuarioBD comandoBD = new ValidarUsuarioBD();
    ValidarUsuarioSW comandoSW = new ValidarUsuarioSW();

    comandoBD.setPassword( "12345" );
    comandoBD.setUsuario( "jfg" );

    comandoSW.setUsuario( "jfg" );
    comandoSW.setPassword( "12345" );

    invoca.asignarComando( comandoBD );
    invoca.ejecutarComando();

    invoca.asignarComando( comandoSW );
    invoca.ejecutarComando();

  }

}
```