El modelo–vista–controlador (MVC) es un patrón de [arquitectura de software](https://es.wikipedia.org/wiki/Arquitectura_de_software) que separa los [datos](https://es.wikipedia.org/wiki/Datos) y la [lógica de negocio](https://es.wikipedia.org/wiki/L%C3%B3gica_de_negocio) de una aplicación de la [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario) y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres [componentes](https://es.wikipedia.org/wiki/Componentes) distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.[[1]](https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador#cite_note-1) [[2]](https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador#cite_note-burbeck-2) Este patrón de [arquitectura de software](https://es.wikipedia.org/wiki/Arquitectura_de_software) se basa en las ideas de [reutilización de código](https://es.wikipedia.org/wiki/Reutilizaci%C3%B3n_de_c%C3%B3digo) y la [separación de conceptos](https://es.wikipedia.org/wiki/Separaci%C3%B3n_de_conceptos), características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Ejemplo de Patron de Diseño

Patron de Diseño Strategy

El patrón de diseño strategy define una familia de algoritmos los encapsula y los hace intercambiables entre si, independientemente del cliente que los use.

public abstract class Personaje {

private Integer vida;

public Integer getVida() {  
return vida;  
}

protected void setVida(Integer Vida){  
this. Vida = Vida;  
}

public abstract void atacar();

}

La clase del guerrero y del brujo serían algo así:

public class Guerrero extends Personaje{

public Guerrero(){}

public void atacar(){  
System.out.println("Soy un gerrero y te ataco con mi hacha");  
}  
}

public class Brujo extends Personaje{

public Brujo(){}

public void atacar(){  
System.out.println("Soy un brujo y te ataco con mi baston");  
}  
}

Ahora se viene el primer cambio en nuestra aplicación., nos dicen que como el juego esta teniendo exito se van a agregar más funcionalidades…