

**ESCUELA COLOMBIANA DE INGENIERÍA**  
**PROGRAMACIÓN ORIENTADA A OBJETOS**  
**Parcial tercer tercio.**  
**S17: 2022-1**

**Nombre:** Daniel Antonio Santanilla Arias

**Nota esperada:** \_\_\_\_

**POOBChis**

**IMPORTANTE:**

**1) El desarrollo de los puntos del parcial debe quedar en el documento del parcial. (Sólo se revisará en el proyecto lo que relacionen en el documento).**

**2) Deben publicar la versión inicial y la versión final del proyecto.**

**I. REFACTORIZANDO: BDD-MDD XP**

Seleccione un comportamiento del proyecto que requiera refactorización.

Se va a refactorizar el comportamiento de la ficha cohete y aspiradora

1. Explique su selección, detallando las ventajas y desventajas de la implementación actual.

Ventajas:

Se asignará el comportamiento a la clase Cohete y Aspiradora según poder.

Se quita lógica de la vista y se pasará a donde corresponde.

Desventajas:

No hay desventajas dado que se asigna el comportamiento correcto a la clase que lo debe contener.

2. Incluya en el documento todos los componentes BDD-MDD del comportamiento seleccionado: requisitos, diseño (estructural y de comportamiento), código (presentación y aplicación) y pruebas (unidad y aceptación).
  - Diseño estructural  
Se refactoriza el método de la clase Ficha poderUsado()
  - Diseño de comportamiento  
Se asigna comportamiento en la clase Cohete y Aspiradora para poderUsado() en la refactorización.
  - Código presentación  
Se quitará código de comportamiento de la ficha Cohete y Aspiradora en actionChoose()
  - Código aplicación  
Se implementa el poderUsado() en las clases Cohete y Aspiradora.
3. ¿Cuál patrón podría aplicar para la refactorización? ¿por qué?  
El patrón que se aplica en la refactorización es de comportamiento dado que anteriormente esto no se tenía en cuenta a la hora de realizar los poderes con las fichas.
4. Refactorizar este comportamiento. Explique e incluya los componentes BDD-MDD que cambiaron.  
Ver código y astah clases Ficha, Cohete, Aspiradora y TableroGUI

## II. EXTENDIENDO. Nuevo elemento

Extienda el proyecto para poder jugar con un nuevo tipo de *Dado: Octaedro Mágico*. Este nuevo dado tiene los siguientes comportamientos:

- Posee 8 caras.
- Cada vez que se lanza genera un número aleatorio entre 1 y 8.
- Para las fichas *normales*, funciona de la siguiente manera:
  - Si saca 1, 2 o 3: se duplican los valores, es decir, el valor del dado es 2, 4 o 6.
- Para las fichas *saltarinas*, funciona de la siguiente manera:
  - Si saca 8: se divide el valor a la mitad, es decir, el valor del dado es 4.
- Para las fichas *ingenieras*, funciona de la siguiente manera:
  - Si saca número par: se puede mover una unidad, es decir, el valor del dado es 1.
- Para las fichas de tipo *cohetes, ventajosa o aspiradora*, no se alteran las reglas.

### Nota:

- Se le debe preguntar al usuario cuál tipo de dado usar [1. Normal ó 2. Octaedro Mágico]
- Debe ser visible el tipo de dado con el que se está jugando. (Ejemplos: cambiar imagen, cambiar color o mostrar un texto con el tipo de dado).
- El tipo de dado seleccionado aplica para ambos dados, es decir, se juega con 2 dados del mismo tipo.

1. Incluya en el documento los componentes BDD-MDD: requisitos, diseño (estructural y de comportamiento), código (presentación y aplicación) y pruebas (unidad y aceptación).

- ✓ Diseño estructural

Se agrega un atributo booleano a la clase Dado para saber si este es un dado octaedro mágico o no.

Se agrega un atributo "generacion" que por defecto es 6 y cuando el dado sea mágico este será 8

Se agregan el método en la ficha valorDeDadoAlterado() y sobre escriben las fichas Saltaria e Ingeniera

- ✓ Diseño de comportamiento

Se añaden los métodos agregados en la clase de fichas a los diagramas de secuencia.

- ✓ Código presentación

En configurationWindow en el método acciónIniciar() se pregunta si se quiere usar el dado mágico, de ser así se jugará con dicho dado

En tableroGUI antes de jugar se pregunta si el dado escogido por el jugador es mágico, de serlo, entonces utilizara el nuevo método de la ficha mencionado anteriormente

- ✓ Código aplicación

Se agrega un parámetro de construcción para el dado "esMagico" y un atributo "generacion" que indica el numero que genera los dados de 1 hasta n

Se agrega comportamiento al nuevo método valorDeDadoAlterado()

- ✓ Prueba aceptación

Dado que hay una partida iniciada y se desea comprobar el funcionamiento del dado mágico

se tiene:



Juanito lanza los dados y obtiene una representación visual de los dados en 5 y 7 respectivamente



Juanito sale con la ficha saltarina y luego juega 7 con el valor de dados



Juega juanita y no puede sacar ficha



Avanzados el juego Juanito obtiene 8 con un dado  
 Al jugar la ficha saltarina esta se moverá 4 dado que le dado mágico le ha cambiado el valor de los dados



2. ¿Cuáles patrones usa en su implementación?  
 Los patrones usados para esta implementación son de variación de extensión dado que el comportamiento del dado no cambia, además, se añaden comportamientos para representar lo mágico del dado en las fichas Normal Saltarina e Ingeniera

**NOTA: Las clases con color naranja fueron modificadas**

**COMPONENTES BDD-MDD**

<b>Requisitos</b>	<b>Descripción del comportamiento</b>
Diseño estructural	Zona(s) del diagrama de clase relevantes
Diseño de comportamiento	Diagrama(s) de secuencias relevantes
Código presentación	Código de los método(s) de la capa de presentación que atiende(n) el comportamiento
Código aplicación	Código de los método(s) de la capa de aplicación que atiende(n) el comportamiento
Pruebas de unidad	Código de las pruebas y evidencia de su ejecución
Prueba aceptación	Descripción y dos pantallas significativas