**ESCUELA COLOMBIANA DE INGENIERÍA**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Parcial tercer tercio.**

**S17: 2021-1**

**Nombre: Natalia Orjuela**

**Nota esperada:3**

**I. REFACTORIZANDO: BBD-MDD XP Seleccione un comportamiento del proyecto que requiera refactorización.**

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

La refactorización que hice fue crear una clase llamada Elemento la cual tendrá atributos protected los cuales extenderán todos los elementos del juego como alimento, arociris etc; este cambio es una ventaja ya que así se generalizan los comportamientos de los elementos y al momento de querer modificar alguno de estos es más sencillo y optimo.

La desventaja que se tiene actualmente es que al no tenerlo general hay que modificar en todo el código estos atributos y eso nos podría generar errores en el código ya que se puede pasar por alto algún atributo.

**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). [NOTA: Consultar contenido al final del documento]**

**3. ¿Cuál patrón podría aplicar para la refactorización?¿por qué?**

**Cambiar los parámetros del proyecto**

**4. Refactorice este comportamiento. Explique e incluya los componentes BDD-MDD que cambiaron. [NOTA: Consultar contenido al final del documento]**

**II. EXTENDIENDO. Nuevo elemento**

**Extienda el proyecto para poder jugar con un nuevo elemento (sorpresa o alimento) con efecto de posición aleatoria:**

** Su posición cambia cada 3s hasta que alguna de las serpientes lo capture.**

Se modifico la clase run.

**public** **void** run() {

**int** cont = 0;

**while** (estado) {

snake.avanzar();

**if**(cont == 10) {

cont = 0;

snake.ali.generarComida();

}

cont +=1;

snake.repaint();

**try** {

Thread.*sleep*(150);

} **catch** (InterruptedException ex) {

Logger.*getLogger*(Caminante.**class**.getName()).log(Level.***SEVERE***, **null**, ex);

}

}

}

En la clase de snake se quitaron los atributos y se extendió la clase elemento.

Para aumentar las dos unidades **for** (**int** i=0;i<2;i++){snake.add(nuevo); }En clase snake