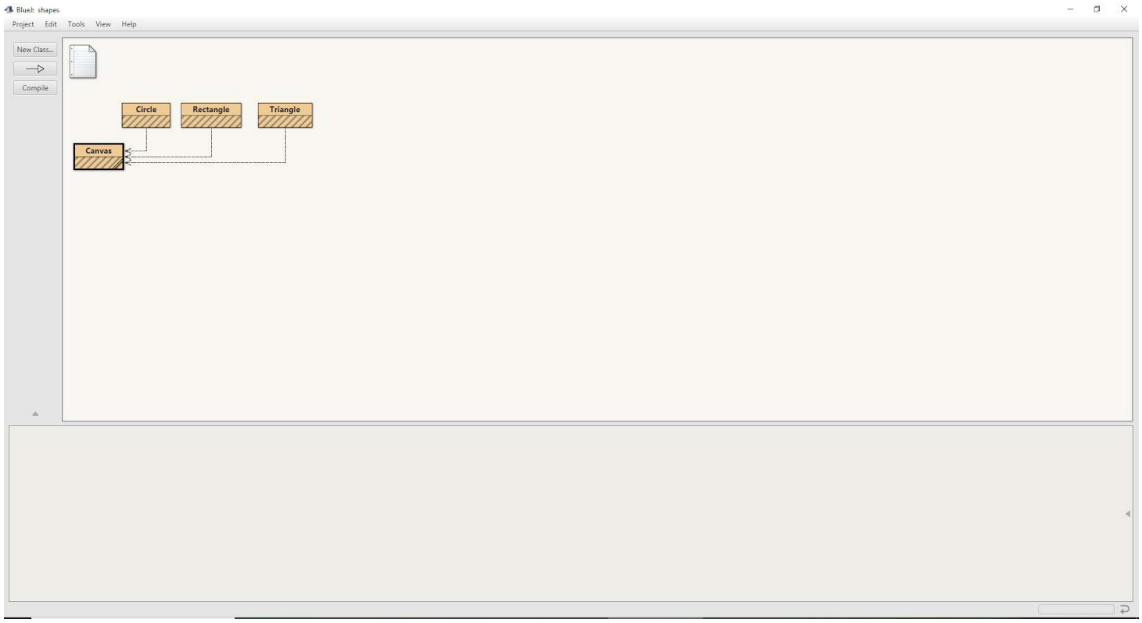
Laboratorio 1

Grupo: Miguel Angel Monroy Cardenas
Wilmer Arley Rodríguez Ropero

A Conociendo el proyecto Shapes

1. El proyecto “shapes” es una versión modificada de un recurso ofrecido por BlueJ. Para trabajar con él, bajen shapes.zip y ábranlo en BlueJ. Capturen la pantalla.

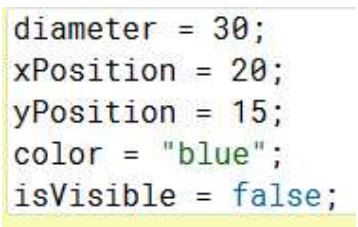


2. El diagrama de clases permite visualizar las clases de un artefacto software y las relaciones entre ellas. Considerando el diagrama de clases de “shapes” (a) ¿Qué clases ofrece? (b) ¿Qué relaciones existen entre ellas?
    a. Circle, Rectangle, Triangle, Canvas.
    b. Las clases Circle, Rectangle y Triangle hacen uso de la clase canvas.
3. La documentación presenta las clases del proyecto y, en este caso, la especificación de sus componentes públicos. De acuerdo con la documentación generada: (a) ¿Qué clases tiene el paquete shapes? (b) ¿Qué atributos ofrece la clase Circle? (c) ¿Cuántos métodos ofrece la clase Circle? (d) ¿Cuáles métodos ofrece la clase Circle para que la figura cambie (incluya sólo el nombre)?
    a. Triangle, Circle, Rectangle y Canvas.
    b. PI, diameter, xPosition, yPosition, color, isVisible
    c. 12 métodos ofrece la clase Circle
    d. changeColor, changeSize
4. En el código de cada clase está el detalle de la implementación. Revisen el código de la clase Circle. Con respecto a los atributos: (a) ¿Cuántos atributos realmente tiene? (b) ¿Cuáles atributos describen la forma de la figura?. Con respecto a los métodos: (c) ¿Cuántos métodos tiene en total? (d) ¿Quiénes usan los métodos privados?
    a. 6 atributos ofrece la clase
    b. PI, diameter, xPosition, yPosition, color, isVisible
    c. 14 métodos tiene la clase
    d. Los usa la clase Canvas

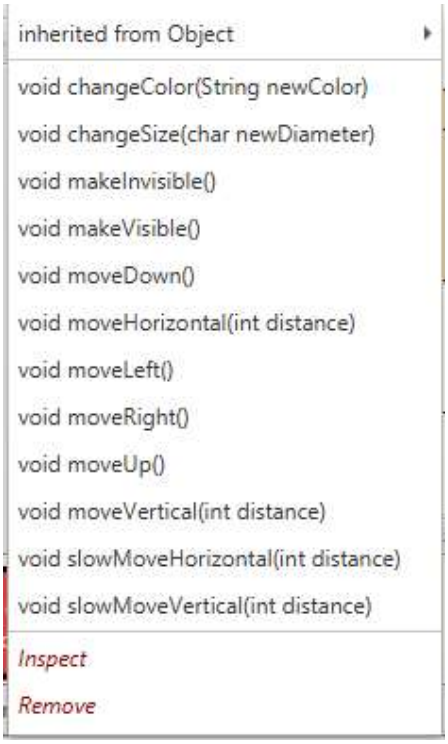
5. Comparando la documentación con el código (a) ¿Qué no se ve en la documentación? (b) ¿por qué debe ser así?
    a. No se ven los métodos y atributos privados
    b. Porque los modificadores de acceso limitan la accesibilidad a los métodos o atributos de la clase.
6. En el código de la clase Circle, revise el atributo PI (a) ¿Qué significa que sea public? (b) ¿Qué significa que sea static? (c) ¿Qué significa que sea final? (d) ¿De qué tipo de datos debería ser?¿Por qué? Actualicenlo.
    a. Que el atributo tiene accesibilidad para todas las clases
    b. Es un atributo que no se asocia al objeto, si no a la clase del objeto
    c. Significa el atributo no puede ser modificado, siendo una constante
    d. El tipo de dato es pequeño entonces decidimos cambiarlo a float, ya que double es muy grande
7. En el código de la clase Circle revisen el detalle del tipo del atributo diameter (a) ¿Qué se está indicando al decir que es int?. (b) Si sabemos todos circulos van a ser pequeños (diametro menor a 100), ¿de que tipo deberían ser este atributo? (c) Si son grandes, pero no tanto (diametro menor a 30000), ¿de qué tipo deberían ser este atributo? (d) ¿qué restricción adicional tendrían este atributo? Refactoricen el código considerando (c) y expliquen claramente sus respuestas.
    a. int es un tipo de entero con tamaño de 32 bits
    b. Consideramos que el tipo de dato que se puede poner en este caso sería byte o char
    c. Consideramos que el tipo de dato que se puede poner en este caso sería short
8. ¿Cuál dirían es el propósito del proyecto “shapes”?
    a. Hacer una breve introduccion al usuario en los conceptos como diagramas de clases, documentación y código con ayuda del IDE Bluej

B Manipulando objetos. Usando un objeto

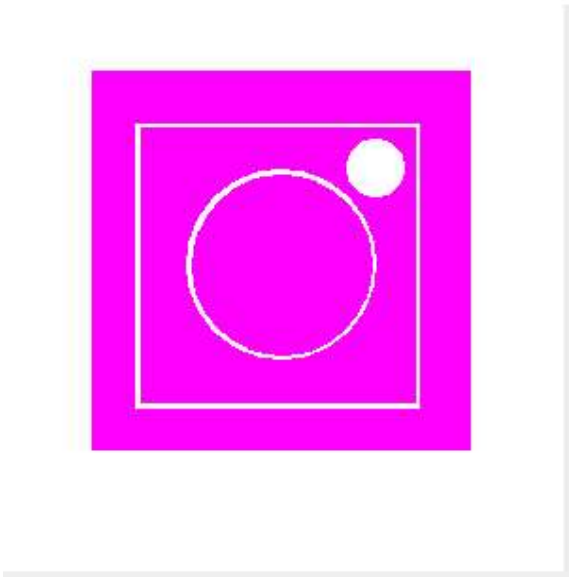
1. Creen un objeto de cada una de las clases que lo permitan. (a) ¿Cuántas clases hay? ¿Cuántos objetos crearon? (b) ¿Por qué?
    a. Hay un total de cuatro clases y tres de estas clases que nos permiten crear objetos; creamos tres objetos uno por cada clase
    b. Para probar como funciona cada clase, como el color y tamaño que sale cada objeto por defecto
2. Inspeccionen el estado del objeto *Circle*, ¿Cuáles son los valores de inicio de todos sus atributos? Capture la pantalla.



3. Inspeccionen el comportamiento que ofrece el objeto *Circle*. (a) Capturen la pantalla. (b) ¿Por qué no aparecen todos los que están en el código?
    a.



    b. No aparecen todos los métodos, ya que los métodos que faltan son privados
4. Construyan, con “shapes” sin escribir código, una propuesta de la imagen del logo de su red social favorita. (a) ¿Cuántas y cuáles clases se necesitan? (b) ¿Cuántos objetos se usan en total? (c) Capturen la pantalla. (d) Incluyan el logo original.
    a. Se necesitan 2 clases, que son circle y triangle
    b. Se utilizaron 6 objetos en total
    c.

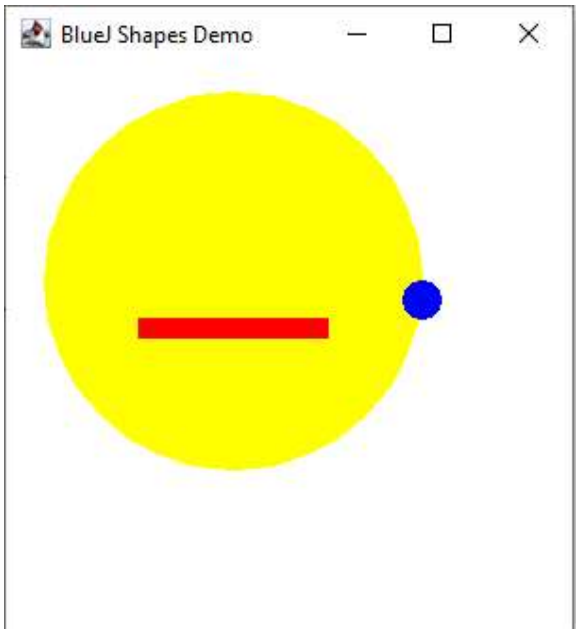


d.

C Manipulando objetos. Analizando y escribiendo código

1. Lean el código anterior. (a) ¿cuál es la figura resultante? (b) Píntenla.
    a. La figura resultante es una cara,y según el código está conformada por 4 objetos
    b.

2. Habiliten la ventana de código en línea , escriban el código. Para cada punto señalado indiquen: (a) ¿cuántas variables existen? (b) ¿cuántos objetos existen? (c) ¿qué color tiene cada uno de ellos? (d) ¿cuántos objetos se ven? Al final, (e) Expliquen sus respuestas. (f) Capturen la pantalla.
    a. Existen 4 variables: Tamaño, color, posición y visibilidad.
    b. 4 objetos que forman una cara.
    c. El color del objeto de la cara es amarilla, la boca es roja y los ojos son azules debido al color por defecto del constructor circle.
    d. Primero 3 y luego 4
    e. En un principio se veían 3 objetos debido a la forma en la cual se creaba el ojo derecho, al corregir ese error se consiguió que se vieran 4 objetos.
    f.

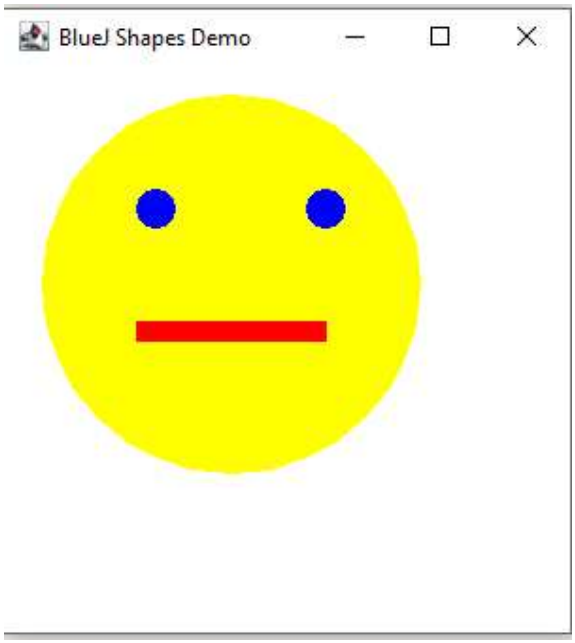


3.

a. Son bastante similares aunque hay algunas diferencias en el tamaño y posición de los objetos

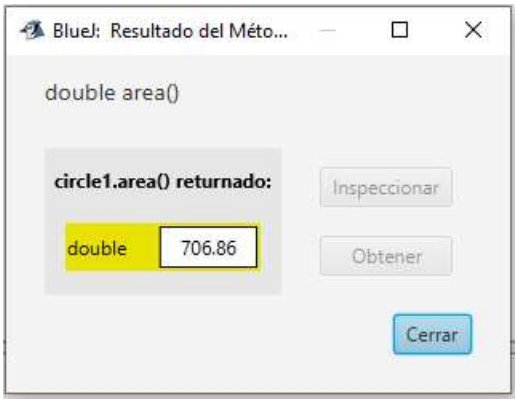
b. Debido a los cambios mencionados en el punto anterior.

c.

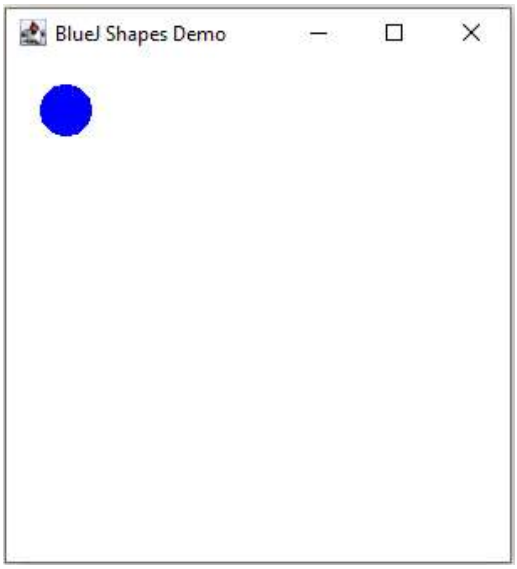


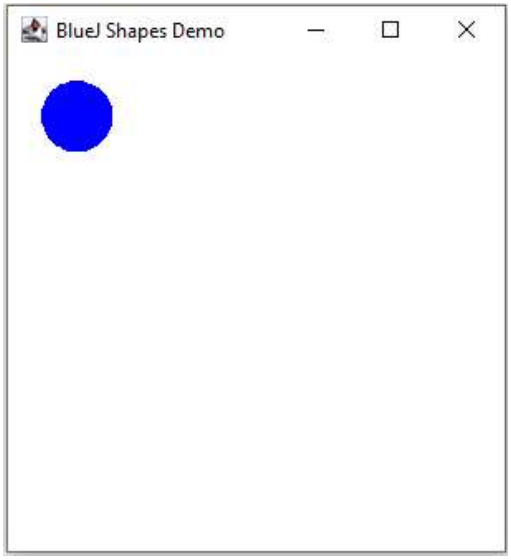
D. Extendiendo una clase. Circle

1. Desarrollen en Circle el método area(). ¡Pruébenlo! Capturen una pantalla

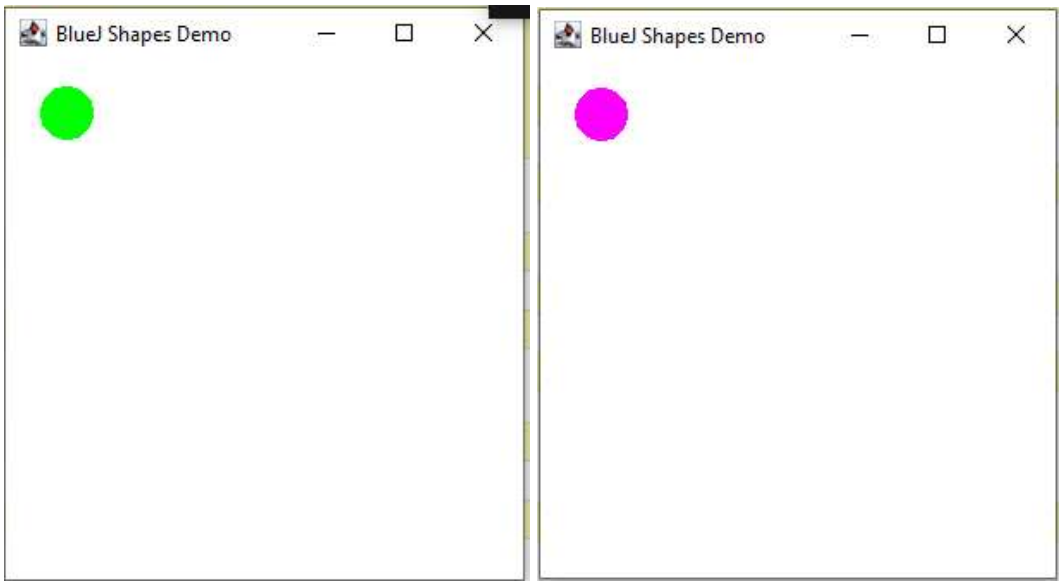


2. Desarrollen en Circle el método duplicate() (duplica su área) . ¡Pruébenlo! Capturen dos pantallas.





3. Desarrollen en Circle el método rainbow() (va cambiando de color siguiendo los colores del arco iris) . ¡Pruébenlo! Capturen dos pantallas.



4. Propongan un nuevo método para esta clase

```
/**
 * Exponentially change size and moves position
 */
public void changeSizePosition(){
    this.diameter += diameter;
    this.xPosition += 10;
    this.yPosition += 25;
    erase();
    draw();
}
```

5. Generen nuevamente la documentación y revise la información de estos nuevos métodos. Capturen la pantalla.

area [Show source in BlueJ]

```
public double area()
```

Calculate area

Returns:

int area

duplicate [Show source in BlueJ]

```
public void duplicate()
```

Duplicate circle area

rainbow [Show source in BlueJ]

```
public void rainbow()
```

Change the circle color with the rainbow colors

changeSizePosition [Show source in BlueJ]

```
public void changeSizePosition()
```

Exponentially change size and moves position

E. Codificando una nueva clase. Dice

1. Revisen el diseño y clasifiquen los métodos en: constructores, analizadores y modificadores.

   Constructor:

   _(digit)

   Analizador:

   get()

   Modificador:

   next()
   change()
   change(digit)

makeVisible()
makeInvisible()
moveTo(x,y)
changeColor()

2. Desarrollen la clase Dice considerando los 4 mini-ciclos. Al final de cada mini-ciclo realicen una prueba indicando su propósito. Capturen las pantallas relevantes.

```
public Dice(byte digit)
{
    x = digit;
    perimeter = new Rectangle();
    c1 = new Circle();
    c2 = new Circle();
    c3 = new Circle();
    c4 = new Circle();
    c5 = new Circle();
    c6 = new Circle();
    c7 = new Circle();

    array.add(c1);
    array.add(c2);
    array.add(c3);
    array.add(c4);
    array.add(c5);
    array.add(c6);
    array.add(c7);

    perimeter.changeSize(250, 250);
    perimeter.changeColor("black");

    c1.moveHorizontal(100);
    c1.moveVertical(30);

    c2.moveHorizontal(225);
    c2.moveVertical(30);

    c3.moveHorizontal(100);
    c3.moveVertical(110);

    c4.moveHorizontal(225);
    c4.moveVertical(110);

    c5.moveHorizontal(100);
    c5.moveVertical(185);

    c6.moveHorizontal(225);
    c6.moveVertical(185);
```

```
/**
 * Permite saber cual es el valor del dado en el momento
 * @returns Int x
 */
public byte get()
{
    return x;
}

/**
 * Cambia el valor del dado por el siguiente, si llega a 6 se devuelve a 1
 */
public void next()
{
    if(x < 6){
        x ++;
    }
    else{
        x = 1;
    }
    set();
}

/**
 * Cambia el valor almacenado a dado por uno ingresado por el usuario
 * @param byte digit
 */
public void change(byte digit){
    x = digit;
    set();
}

/**
 * Cambia el valor almacenado por el dado por uno aleatorio
 */
public void change(){
    int random = (int)(Math.random()*6+1);
    byte n = (byte) random;
    x = n;
    set();
}
```

```java
/**
 * Mueve el dado a las coordenadas dadas por el usuario
 * @param Int x
 * @param Int y
 */
public void moveTo(int x, int y){
    perimeter.moveHorizontal(x);
    perimeter.moveVertical(y);

    for(Circle i : array){
        i.moveHorizontal(x);
        i.moveVertical(y);
    }
}

/**
 * Cambia el color de los circulos en el dado por uno ingresado por el usuario
 * @param String color
 */
public void changeColor(String color){
    for(Circle i: array){
        i.changeColor(color);
        this.color = "green";
    }
}

public void makeVisible(){
    perimeter.makeVisible();
    this.isVisible = true;
    for(Circle i: array){
        i.makeVisible();
    }
}

public void makeInvisible(){
    perimeter.makeInvisible();
    this.isVisible = false;
    for(Circle i: array){
        i.makeInvisible();
    }
}
```

F. Diseñando y codificando una nueva clase. DiceTaken

1. Diseñen la clase DiceTaken, es decir, definan los métodos que debe ofrecer.

Diagrama.astah

2. Planifiquen la construcción definiendo algunos miniciclos. Recuerden que al final de cada miniciclo deben poder probar.

Para la construcción del DiceTaken planteamos los siguiente miniciclos

a.

i. _(n)
ii. rotar(x, y, z)

iii. rotar(x, y)

b.

i. deslizar(AcoorX, AcoorY, BcoorX, BcoorY)
ii. deslizarDerecha()
iii. deslizarIzquierda()
iv. deslizarArriba()
v. deslizarAbajo()

c.

i. minimo()
ii. verificar()

d.

i. ganar()
ii. reproducirSonido()
iii. parpadear()

3. Implementen la clase . Al final de cada miniciclo realicen una prueba de aceptación. Capturen las pantallas relevantes.
Retrospectiva
Ciclo a

```java
/**
 * Constructor for objects of class DiceTaken
 * @param byte n
 */
public DiceTaken(byte n){
    this.n = n;
    this.matrix = new Dice[n][n];
    for (int i=0;i<n;i++){
        for (int j = 0;j<n;j++){
            dice = new Dice(num);
            matrix[i][j] = dice;
            matrix[i][j].change();
        }
    }
    for (int i = 0; i<n; i++){
        for (int j=0;j<n;j++){
            matrix[i][j].moveTo(i*(280),j*(280));
        }
    }
    int randomi = (int)(Math.random()*n+0);
    byte i = (byte) randomi;
    int randomj = (int)(Math.random()*n+0);
    byte j = (byte) randomj;
    matrix[randomi][randomj].makeInvisible();
    matrix[randomi][randomj].change(pred);
    minimo();
    verificar();
}
```

```java
/**
 * Lanza el dado en la posicion dada junto con el digito que se quiere
 * @param int x
 * @param int y
 * @param int z
 */
public void rotar(int x,int y, byte z){
    this.matrix[x][y].change(z);
    verificar();
}

/**
 * Lanza el dado en la posicion dada
 * @param int x
 * @param int y
 */
public void rotar(int x,int y){
    this.matrix[x][y].change();
    minimo();
    verificar();
}
```

Ciclo B:

```java
/**
 * Cambia de posicion los dados
 * @param int AcoorX
 * @param int AcoorY
 * @param int BcoorX
 * @param int BcoorY
 */
private void deslizar(int AcoorX,int AcoorY,int BcoorX,int BcoorY){
    if(matrix[BcoorX][BcoorY].isVisible() == false){
        num = matrix[AcoorX][AcoorY].get();
        matrix[AcoorX][AcoorY].makeInvisible();
        matrix[AcoorX][AcoorY].change(pred);
        matrix[BcoorX][BcoorY].change(num);
        matrix[BcoorX][BcoorY].makeVisible();
    }
    verificar();
}
```

Ciclo C:

```java
/**
 * Verifica si los dados estan en orden en el tablero
 */
private void verificar(){
    if(matrix[0][0].get() == min){
        matrix[0][0].changeColor("green");
    }
    else{
        matrix[0][0].changeColor("red");
    }
    for (int i=0; i<n; i++){
        for (int j=0; j<n; j++){
            if(matrix[j][i].get() <= matrix[(j+1)%n][(i+((j+1)/n))%n].get() && matrix[j][i].getColor()=="green" && matrix[(j+1)%n][(i+((j+1)/
                matrix[(j+1)%n][(i+((j+1)/n))%n].changeColor("green");
            }
            else if((j+1)%n!=0 || (i+((j+1)/n))%n!=0){
                matrix[(j+1)%n][(i+((j+1)/n))%n].changeColor("red");
            }
        }
    }
    ganar();
}
```

Ciclo D:

```java
/**
 * Verifica si la condicion de victoria se logro
 */
private void ganar(){
    if(matrix[n-2][n-1].getColor() == "green"){
        parpadear();
        reproducirSonido("song.wav");
        JOptionPane.showMessageDialog(null,"Has ganado","Victoria", JOptionPane.INFORMATION_MESSAGE);
    }
}
```

4. Indique las extensiones necesarias para reutilizar el paquete shapes y la clase Dice. Explique.

Tuvimos que implementar en la clase de dado los métodos adicionales de getColor() y isVisible() para poder ejecutar métodos de la clase DiceTaken, en la clase de Circle no tocamos ningún método de los que ya estaban planteados.

getColor() lo usamos en el método de verificar si los dados están en orden para poder determinar si la secuencia estaba correcta.

isVisible() lo usamos en el método de deslizar para corroborar que el dado que se va a mover lo esté haciendo en donde no hay dado, y además en el constructor para determinar el espacio en blanco.

RETROSPECTIVA

1. Rodriguez / 12 horas
   Monroy / 12 horas

2. El laboratorio está completamente finalizado
3. Pair Programing fue muy útil a la horta de compartir ideas y encontrar errores en el código
4. El mayor logro fue completar todos los requisitos de la interfaz ya que había problemas para mostrar los elementos ordenados y reproducir un sonido de victoria
5. El mayor problema técnico fue representar los elementos ordenados, para resolverlo tuvimos que recorrer la matriz de distintas maneras y poner varias condiciones.
6. Como equipo tuvimos una buena comunicación y manejamos el tiempo de forma adecuada, algo que podríamos mejorar es la organización de prioridades.