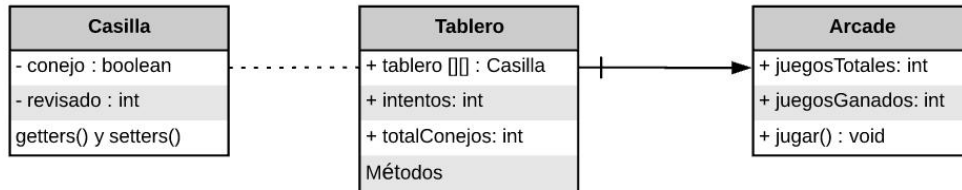


Desarrollo

Para poder desarrollar el Videojuego se hizo un analisis en el cual la mejor manera fue la del siguiente diagrama UML:



Donde:

- **Casilla:** Clase que nos ayudara a saber si hay conejo o no en dicha casilla, a su vez de saber si ya se reviso o no dicha casilla.
- **Tablero:** Clase que definira un Tablero de objeto casillas, las cuales pueden tener o no conejos. Aqui tambien definiremos la logica del Videojuego.
- **Arcade:** Clase que hereda de Tablero la cual nos permitira manejar las estadisticas y todas las partidas que desee jugar el usuario.

Clase Casilla

La clase mas sencilla de las tres, aqui solo se define que tiene como atributos un boolean llamado conejo, el cual si es false, no existira conejo en dicha casilla y existira si su valor es true, mientras que atributo es un int que valdra 1 si ha sido revisada la casilla o 0 en caso contrario.

Codigo Fuente

```
// Clase Casilla
public class Casilla
{
    // Variables privadas
    private Boolean conejo; // true = existe conejo, false = No
    private int revisado; // 1 = El jugador reviso esta casilla

    /* Constructor
     * Crea por defecto una casilla donde no existe un conejo
     * y no ha sido revisado por el jugador
     */
    public Casilla()
    {
        this.conejo = false;
        this.revisado = 0;
    }

    // Devuelve el valor de conejo
    public Boolean getConejo()
    {
        return this.conejo;
    }

    // Modifica el valor de conejo (recibe un booleano)
    public void setConejo(Boolean cone)
    {
        this.conejo = cone;
    }

    // Devuelve el valor de revisado
    public int getRevisado()
    {
        return this.revisado;
    }

    // Modifica el valor de revisado (recibe un entero)
    public void setRevisado(int revid)
    {
        this.revisado = revid;
    }
}
```

Clase Tablero

La clase que definio la logica del juego, esta se explicara por partes.

```
// Importando Paquetes
import java.util.Random; // Numero Aleatorios
import java.util.Scanner; // Scanner

public class Tablero
{
    // Atributos
    public Casilla tablero [][]; // Tablero de conejos
    public int intentos; // Numero de intentos
    public int totalConejos; // Contador de conejos totales no encontrados

    // Constructor que inicializa el tablero con 4 conejos aleatorios
    public Tablero()
    {
        this.tablero = new Casilla [5][5];
        this.intentos = 6;
        this.totalConejos = 0;
        this.inicializarTablero();
    }

    // Se crea en cada casilla del tablero un objeto de tipo Casilla
    // Donde se marca en cada una que no hay conejos y no han sido revisadas
    // Para finalmente llamar a insertaConejos()
    public void inicializarTablero()
    {
        for (int i = 0; i < 5; i++)
        {
            for (int j = 0; j < 5; j++)
            {
                this.tablero[i][j] = new Casilla();
            }
        }
        this.insertarConejos();
    }
}
```

Se define los contadores intentos y totalConejos, los cuales respectivamente nos marcaran el limite de intentos en cada partida y la cantidad de conejos a encontrar. Tambien se define una matriz de Casillas de 5x5 en la cual se le indicara posteriormente donde hay conejos ocultos y donde no.

Constructor(): Instancia el tablero e inicializa variables y el tablero.

inicializarTablero() : Se recorre el tablero para que en cada posicion se cree una casilla que no se haya revisado y no tenga conejo. Para finalmente llamar al metodo insertarConejos.

```

// Saca dos numeros aleatorios del 0 al 4 los cuales indicaran una posicion en el tablero
// y en esa posicion insertara un conejo si es que no hay
// Y en caso de haber saca otros dos numeros aleatorios para insertar el conejo en otro lado
// Finalmente llama a conejosTotales()
public void insertarConejos()
{
    Random r = new Random();
    for (int i = 0; i < 4; i++)
    {
        int ren = r.nextInt(5);
        int col = r.nextInt(5);
        if (this.tablero[ren][col].getConejo() == false) // Si no hay conejo
        {
            this.tablero[ren][col].setConejo(true); // Lo inserta
        }
        else // si no
        {
            this.tablero[r.nextInt(5)][r.nextInt(5)].setConejo(true); // Inserta conejo en otra posicion
        }
    }
    this.conejosTotales();
}

// Recorre el tablero cuenta la cantidad de conejos en el tablero
// Sirve para verificar si no hubo un problema a la hora de insertarlos
public void conejosTotales()
{
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            if (this.tablero[i][j].getConejo() == true )
            {
                this.totalConejos++;
            }
        }
    }
}

```

insertarConejos(): Es el encargado de ingresar a 4 casillas aleatorias un conejo, solamente cambiando el valor de conejo a true de las cuatro casillas. Para luego llamar al metodo conejosTotales.

conejosTotales(): Recorre todo el tablero (matriz) para contar cuantos conejos hay en el y poder asignar el numero a totalConejos.

```

// Recorre todo el tablero y dependiendo si hay conejos o no y si ya se reviso o no en cada casilla
// imprime un simbolo diferente para cada caso
// Ademas de imprimir el numero de intentos y conejos que faltan para encontrar
public void imprimirTablero()
{
    System.out.println("Numero de Intentos: " + this.intentos);
    System.out.println("Conejos Restantes: " + this.totalConejos);
    System.out.println(" ? -> Casilla no revisada\n * -> Casilla revisada sin Conejo encontrado\n X -> Casilla revisada con Conejo encontrado");
    System.out.println("-----");
    System.out.println(" / || 1 || 2 || 3 || 4 || 5 |");
    for (int i = 0; i < 5; i++)
    {
        System.out.println("-----");
        System.out.print(" " + (i+1) + " |");
        for (int j = 0; j < 5; j++)
        {
            if (this.tablero[i][j].getConejo() == false && this.tablero[i][j].getRevisado() == 0 )
                System.out.print(" ? |");
            else if (this.tablero[i][j].getConejo() == false && this.tablero[i][j].getRevisado() == 1 )
                System.out.print(" * |");
            else if (this.tablero[i][j].getConejo() == true && this.tablero[i][j].getRevisado() == 0 )
                System.out.print(" ? |");
            else if (this.tablero[i][j].getConejo() == true && this.tablero[i][j].getRevisado() == 1 )
                System.out.print(" X |");
        }
        System.out.println();
    }
    System.out.println("-----");
}

// Recorre todo el tablero y segun haya conejos o no te imprime algo diferente
// En resumen te imprime la posicion de cada conejo en el tablero
public void respuestaTablero()
{
    System.out.println("****JUEGO FINALIZADO****\nCONEJOS EN EL TABLERO");
    System.out.println("-----");
    System.out.println(" / || 1 || 2 || 3 || 4 || 5 |");
    for (int i = 0; i < 5; i++)
    {
        System.out.println("-----");
        System.out.print(" " + (i+1) + " |");
        for (int j = 0; j < 5; j++)
        {
            if (this.tablero[i][j].getConejo() == false )
                System.out.print(" ? |");
            else
                System.out.print(" X |");
        }
        System.out.println();
    }
    System.out.println("-----");
    System.out.println("Conejos Encontrados " + (4 - this.totalConejos) + " de 4");
}

```

imprimirTablero(): Con un formato donde muestra numero de intentos y conejos a encontrar, imprime el tablero en pantalla y en cada casilla mostrara algo diferente segun se haya hecho con ella:

- ? -> En caso de que haya o no conejo y no se haya revisado la casilla.
- * -> En caso de que no haya conejo y se haya revisado la casilla (No encontrado un conejo).
- X -> En caso de que haya conejo y se haya revisado la casilla (Encontrado un conejo)

respuestaTablero(): Imprime el tablero con la ubicacion de los conejos;

- ? -> En caso de que no haya conejo.
- X -> En caso de que haya conejo.


```

public int juego()
{
    Scanner sc = new Scanner(System.in);
    while(this.intentos != 0 && this.totalConejos != 0) // Mientras tengas intentos y no h
    {
        this.imprimirTablero();
        //this.respuestaTablero(); // Para verificar que se puede ganar descomentar esta li
        System.out.println("En que casilla esta el conejo?");
        System.out.print("Renglon: ");
        int ren = sc.nextInt();
        while(ren < 1 || ren > 5) // Si el numero del renglon no es valido
        {
            // se repetira hasta que ingrese uno valido
            System.out.println("Favor de poner un numero de renglon valido (1-5)");
            System.out.print("Renglon: ");
            ren = sc.nextInt();
        }
        System.out.print("Columna: ");
        int col = sc.nextInt();
        while(col < 1 || col > 5) // si el numero de la columna no es valido
        {
            // se repetira hasta que se ingrese uno valido
            System.out.println("Favor de poner un numero de columna valido (1-5)");
            System.out.print("Columna: ");
            col = sc.nextInt();
        }
        // Simula limpiar pantalla en la consola (En caso de no funcionar, quite las linea
        System.out.print("\033[H\033[2J");
        System.out.flush();
        // Simula limpiar pantalla en la consola
        if(this.tablero[ren-1][col-1].getConejo() == true) // Si hay conejo en la casilla
        {
            if (this.tablero[ren-1][col-1].getRevisado() == 1) // y si esta revisada
            {
                System.out.println("****CONEJO YA ENCONTRADO****"); // Se te notifica que
            }
            else // si no esta revisado
            {
                this.tablero[ren-1][col-1].setRevisado(1); // se checa como revisado
                this.totalConejos -= 1; // decrementa uno el total de conejos no encontrados
                System.out.println("****ENCONTRASTE UN CONEJO****");
            }
        }
        else // En caso de no haber conejo
        {
            this.tablero[ren-1][col-1].setRevisado(1); // Se marca como revisado
            this.intentos -= 1; // se decrementa uno los intentos
            System.out.println("****NO HABIA CONEJO****"); // Se notifica
        }
    }
    // Simula limpiar pantalla en la consola (En caso de no funcionar, quite las lineas)
    System.out.print("\033[H\033[2J");
    System.out.flush();
    // Simula limpiar pantalla en la consola
    this.respuestaTablero(); // Al final se muestra las ubicaciones de los conejos
    if(this.totalConejos == 0) // Dependiendo si los encontraste todos sin gastar todos lo
    {
        System.out.println("****FELICIDADES GANASTE :)****");
        return 1;
    }
    System.out.println("SUERTE PARA LA PROXIMA");
    return 0;
}

```

juego(): Su funcion es que mientras el usuario no gaste sus intentos o encuentre todos los conejos, le muestre el tablero con su situacion actual dentro del juego, le pida que inidique en que casilla revisar, verifica que la casilla sea valida, para luego ir a la casilla y si hay conejo decremente uno los conejos a buscar (totalConejos) y si no decrementa uno sus intentos. En caso de ganar te lo notifica y devuelve un 1 y en caso de perder devuleve un 0.

Clase Arcade

```
// Clase principal del proyecto
// Hereda de Tablero para poder manejar el juego de Conejos
public class Arcade extends Tablero
{
    // Contadores
    public int juegosTotales;
    public int juegosGanados;

    // Constructor
    public Arcade()
    {
        this.juegosTotales = 0;
        this.juegosGanados = 0;
    }

    // Metodo para jugar en la Arcade
    public void jugar()
    {
        int opc = 1; // Variable que define si se quiere volver a jugar
        Scanner sc = new Scanner(System.in); // Scanner

        System.out.println("***** Busca Conejos *****\n***** BIENVENIDO *****");
        while(opc != 1)
        {
            // Crea un Arcade con el Tablero de conejos
            Arcade juegoCone = new Arcade();
            int resultado = juegoCone.juego(); // Inicia el juego
            this.juegosTotales ++ 1;
            this.juegosGanados += resultado;
            System.out.print("\nQuieres volver a jugar?\nSelecciona el numero segun su respuesta:\n1.-Si\n2.-No\nRes:");
            opc = sc.nextInt();
        }

        // Simula limpiar pantalla en la consola (En caso de no funcionar, quite las lineas)
        System.out.print("\033[H\033[2J");
        System.out.flush();
        // Simula limpiar pantalla en la consola
        System.out.println("***** Busca Conejos *****");
        System.out.println("ESTADISTICAS");
        System.out.println("Partidas Totales Jugadas: " + this.juegosTotales);
        System.out.println("Partidas Ganadas: " + this.juegosGanados);
        System.out.println("Partidas Perdidas: " + (this.juegosTotales - this.juegosGanados));
        System.out.println("Porcentaje de Victorias (Efectividad): " + (double) ((double) this.juegosGanados / ((double) this.juegosTotales)) * 100 + "%");
        System.out.println("***** HASTA LUEGO <3 *****");
    }

    // Main del proyecto
    public static void main(String[] args)
    {
        // Se crea una Arcade y se juega el Busca Conejos
        Arcade arcade = new Arcade();
        arcade.jugar();
    }
}
```

Hereda todo lo de la clase Tablero, tiene dos contadores juegosTotales y juegosGanados que definiran los juegos totales y ganados que juegue el usuario respectivamente.

jugar(): Mientras el desee podra jugar toda las partidas que quiera y distintas ya que en cada partida se vuelve a crear un objeto de tipo Arcade, donde cada partida incrementara el contador juegosTotales y si se gana incrementara el contador juegosGanados con lo que mande el metodo juego. Y cuando el usuario salga se le muestre sus estadisticas de juego.

Resultados

Inicio del juego

```
***** Busca Conejos *****
***** BIENVENIDO *****
Numero de Intentos: 6
Conejos Restantes: 4
? -> Casilla no revisada
* -> Casilla revisada sin Conejo encontrado
X -> Casilla revisada con Conejo encontrado

-----
| / || 1 || 2 || 3 || 4 || 5 |
-----
| 1 || ? || ? || ? || ? || ? |
-----
| 2 || ? || ? || ? || ? || ? |
-----
| 3 || ? || ? || ? || ? || ? |
-----
| 4 || ? || ? || ? || ? || ? |
-----
| 5 || ? || ? || ? || ? || ? |
-----
En que casilla esta el conejo?
Renglon: 2
Columna: 3
```

Actualizacion del Tablero

```
****NO HABIA CONEJO****
Numero de Intentos: 2
Conejos Restantes: 3
? -> Casilla no revisada
* -> Casilla revisada sin Conejo encontrado
X -> Casilla revisada con Conejo encontrado

-----
| / || 1 || 2 || 3 || 4 || 5 |
-----
| 1 || X || ? || ? || ? || ? |
-----
| 2 || ? || * || ? || ? || ? |
-----
| 3 || ? || ? || * || ? || ? |
-----
| 4 || ? || ? || ? || * || ? |
-----
| 5 || ? || ? || ? || ? || * |
-----
En que casilla esta el conejo?
Renglon: 4
Columna: 5
```


Fin de una Partida

```
***JUEGO FINALIZADO***
CONEJOS EN EL TABLERO
-----
| / || 1 || 2 || 3 || 4 || 5 |
-----
| 1 || X || ? || ? || X || ? |
-----
| 2 || X || ? || ? || ? || ? |
-----
| 3 || ? || ? || ? || ? || ? |
-----
| 4 || ? || ? || ? || ? || ? |
-----
| 5 || ? || ? || ? || X || ? |
-----
Conejos Encontrados 2 de 4
SUERTE PARA LA PROXIMA
-----
4 || ? || ? || ? || * || ? |
-----
Quieres volver a jugar?
Selecciona el numero segun su respuesta:
1.-Si
2.-No
Res:█
```

Fin del Juego

```
***** Busca Conejos *****
ESTADISTICAS
Partidas Totales Jugadas: 2
Partidas Ganadas: 0
Partidas Perdidas: 2
Porcentaje de Victorias (Efectividad): 0.0%
***** HASTA LUEGO <3 *****
```

Conclusiones