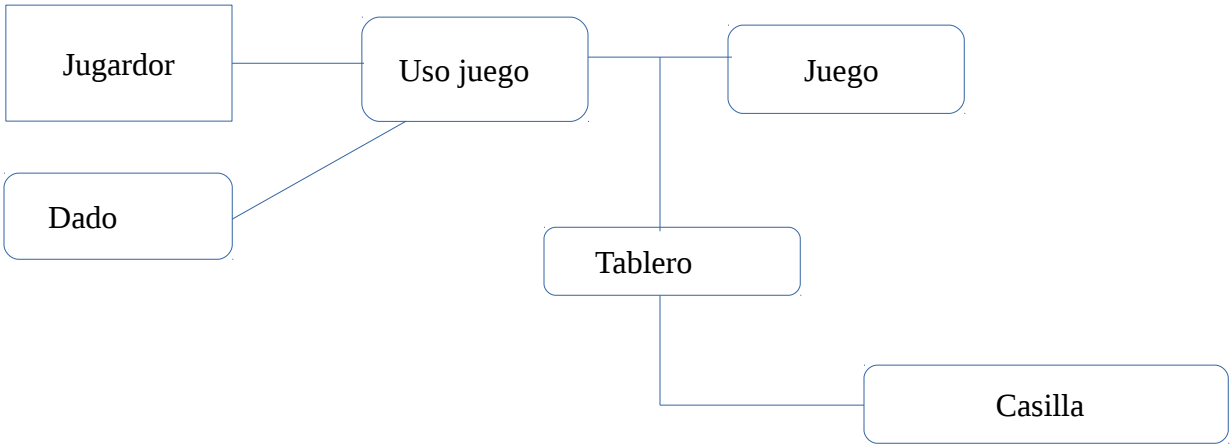
Proyecto.



Para la clase ***jugador*** tendremos que tener en cuenta el nombre y dinero del jugador, en este caso es importante la ficha para poder distinguirlo del otro jugador, pero por conveniencia es mejor ponerlo en tablero, de esta forma tendríamos algo así:

```
public class Jugador{
private String nombre;
Pivate int dinero;
Private String ficha;

Public jugador servira para contruir un jugador

public Jugador (String nombre, String ficha, int dinero)
this.nombre = nombre;
this.ficha = ficha;
this.dinero = dinero;

Obtendremos el nombre de un jugador
public String getNombre() {
return nombre;
}
Obtendremos la ficha de dicho jugador
public String getFicha() {
return ficha;
 }

Obtendremos el dinero que tiene el jugador, el cual al principio sera 0 ya que todo estara apostado
public int getDinero() {
return dinero;
```

Para la clase ***Dado*** modela un objeto del tipo dado “ balanceado” para el tablero del juego patea y corre podemos ocupar el dado de la practica 05

```
public class Dado {
Definimos tributos o variables que utilizaremos, u visibilidad es privada para evitar cambios
   private int numCaras;
   private boolean balanceado;

Nos servira para construir un dado de 6 caras
   public Dado() {

Construye el dado con el numero de caras dado y color
   public Dado(int numCaras, int color) {
            }

 Servira para modificar el numero de caras del dado
   setNumCaras(int numCaras) {
         this.numCaras = numCaras;
   }

 Obtenemos cuantas caras tiene el dado
   public int getNumCaras() {
         return numCaras;
   }

  Podremos saber si el dado usado es balanceado o no en otro caso
   public boolean esBalanceado() {
         return false;
   }
  Obtendremos el color del dado
   public int getColor() {
   }
  Modifica el color del dado
    public void setColor() {
    this.color = color;
   }
  Lanzamos el dado y ontendremos el valor de dicho dado
   public int lanzaDado() {
         return 0;
   }
  Permite saber cuantas veces se lanzara el dado
    public int[] lanzaNVeces(int lanzamientos) {
```

```
    }
  Obtendremo el numero de cuantos lanzamientos hacemos
    public int getUltimoLanzamiento() {
    }
```

Para la clase ***Juego*** tendremos en cuenta la logica, asi como casos excepcionales,

Public class Juego {

Robar nos servira para el momento en que el jugador este en la situción de poder robar parte de la apuesta

```
  public void robar(){
 }
```

Nos permitira saber cuando un jugador le pega a otro la partida termina, en el primer lanzamiento no aplica

```
  public void pega(){
 }
```

Nos permitira saber cuando le pegamos al primer jugador en el primer turno

```
 public void primeroypega(){
 }
```

Nos permite decidir si un jugador elige entre avanzar, robar de la apuesta o subir a la casilla neutral del lado del tablero en que se encuentra solamente cuando obtenga un doble 6.

```
public void doble6(){
}
```

Un jugador en su primer lanzamiento obtiene 6 en cada dado y ademas pega este permanece en la casilla neutral y roba doble, no se detiene el juego ya que es el primer lanzamiento.

```
public void dobleypega() {
}
```

Permite al jugador avanzar en el circuito pricipal en numero obtenido de tirar los dado

```
public void avanza(){
}
```

Si con el numero obtenido de tirar los dados llegamos donde se encuentra la casilla neutral y nos queda 1 moviento podremos elegir entre subir a la casilla neutral ó avanzar en el circuito principal por lo tanto avanzaysube se ejecutara cuando tengamos la opcion de subir a una casilla neutral

```
public void avanzaysube(){
}
```

La clase ***Usojuego*** permite a un usuario utilizar los métodos de la clase Juego para

llevar a cabo la partida, ya sea contruir dado y lnzarlos por ejemplo, el uso de doble6, pega, roba, doble y pega. Además si el juego termina en tablero debe avisar a Usojuego nosotros sabremos cuando acaba pero el juego seguira ejecutandose

La clase ***Casilla*** Clase que modela un objeto de tipo casilla especial para el tablero del juego Patea y corre,
public class Casilla {

private int ocupante; Nos permite saber si esta ocupada por el jugador 1 o el jugador 2 o si esta vacia
private Casilla cNeutral; Permite saber si la casilla es neutral (null)
private int indice; para poder hacer diferencia entre casillas

private int lado; Saber de que lado del tablero nos encontramos es util para saber si podemos acceder a la casilla neutral de dicho lado

Creamos la casilla del circuito
public Casilla(Casilla cNeutral, int indice, int lado) {

```
        ocupante = 0;
        this.cNeutral = cNeutral;
        this.indice = indice;
        this.lado = lado;
    }
```

Creamos ahora si la casilla neutral la cual nos ayudara en los casos excepcionales

```
    public Casilla(int indice, int lado) {
        ocupante = 0;
        cNeutral = null;
        this.indice = indice;
        this.lado = lado;
    }
```

estaOcupada nos dice si la casilla esta ocupada por algún jugador, sera indispensable para el caso en el que dos jugadores esten en la misma el juego habrá terminado

```
    public boolean estaOcupada() {
        return (ocupante != 0)? true : false;
    }
```

int gerOcupante nos dice quién está en la casilla para que al siguiente turno no se reinicie

```
    public int getOcupante() {
        return ocupante;
    }
```

esNeutral nos dice si la casilla es neutral y nos regresa un boolean

```
  public boolean esNeutral() {
        return (cNeutral == null)? true : false;
  }
```

setCasilla modifica el ocupante de la casilla se hara cada que un jugador se mueva de una casilla

```
  public void setCasilla(int ocupante) {
        this.ocupante = ocupante;
  }
```

getNeutral Permite saber la casilla neutral nos encontramos y que lado del tablero estaremos ocupando

```
  public Casilla getNeutral() {
        return cNeutral;
  }
```

getIndice permite saber la posición que ocupa la casilla en un tablero 1,2,3,4,5,6 etc al igual servira para saber en que casilla del tablero se encuentra el jugador

```
  public int getIndice() {
        return indice;
  }
```

Permite saber el lado en el que se encuentra dentro de un tablero sea 1, 2 , 3 ,4.

```
  public int getLado() {
        return lado;
  }
}
```

La clase ***tablero*** ocuparemos todo lo que tenemos hasta ahora, la logica del juego (Juego), Dado, Casilla, Jugador, Harémos uso de los dados y de java.util.Random para generar los numeros, cuando se de el momento el uso de casos excepcionales, asi como el saber en que casilla se encuentra cada jugador, los movimientos que puede realizar con base al numero obtenido de tirar los dados, por ejemplo, el primer jugador obtiene un 3 y un 4 en el segundo dado por ende avanzara a la casilla con el indice 7, no le queda otra que solo “avanzar” no puede avanzar y subir para evitar que le pegue el segundo jugador, en cambio si obtiene un 8 podra “avanzar y subir” y así estar a salvo en la casilla neutral, se desarrollaran desiciones para el jugador, al igual cuando este obtenga 6 en ambos dados eligira entre avanzar, robar de la apuesta o estar asalvo en una casilla neutral, el juego terminara si cualquiera de los dos jugadores le pega a otro despues del primer turno, al igual haremos uso de las fichas para poder diferencia a un jugador del otro, por el uso de casilla sabremos el indice de esta, sí esta ocupada o si no lo esta, tambien sabremos de que lado del tablero se encuentra cada jugador y si es el caso poder hacer uso de las casilla “neutrales” y saber si es una casilla neutral, el uso de casilla determinara cuando un jugador le pega a otro (después del primer turno) y asi el juego habra terminado.