

```
public class Ficha {  
    private String color; // Puede ser "Blanco" o "Negro"  
  
    // Constructor para inicializar la ficha  
    public Ficha(String color) {  
        this.color = color;  
    }  
  
    // Getter para obtener el color de la ficha  
    public String getColor() {  
        return color;  
    }  
  
    @Override  
    public String toString() {  
        return color.equals("Blanco") ? "B" : "N";  
    }  
}
```

```

import java.util.ArrayList;
import java.util.List;

public class Tablero {
    private List<List<Ficha>> puntos; // 24 puntos en el tablero, cada uno tiene una lista
    de fichas
    private List<Ficha> barajaBlancas; // Fichas que aún están en la "baraja" para mover
    private List<Ficha> barajaNegras;

    // Constructor para inicializar el tablero
    public Tablero() {
        puntos = new ArrayList<>();
        barajaBlancas = new ArrayList<>();
        barajaNegras = new ArrayList<>();

        // Inicializamos los puntos del tablero (24 puntos)
        for (int i = 0; i < 24; i++) {
            puntos.add(new ArrayList<>()); // Cada punto está vacío al principio
        }

        // Inicializamos las barajas de fichas
        for (int i = 0; i < 15; i++) {
            barajaBlancas.add(new Ficha("Blanco"));
            barajaNegras.add(new Ficha("Negro"));
        }

        // Colocamos las fichas iniciales en el tablero (2 para cada color en 24 y 13)
        for (int i = 0; i < 2; i++) {
            puntos.get(0).add(new Ficha("Blanco"));
            puntos.get(23).add(new Ficha("Negro"));
        }

        for (int i = 0; i < 3; i++) {
            puntos.get(5).add(new Ficha("Blanco"));
            puntos.get(18).add(new Ficha("Negro"));
        }

        for (int i = 0; i < 5; i++) {
            puntos.get(7).add(new Ficha("Blanco"));
            puntos.get(11).add(new Ficha("Negro"));
        }

        for (int i = 0; i < 5; i++) {
            puntos.get(13).add(new Ficha("Blanco"));
        }
    }

    // Mostrar el estado del tablero (simplificado)

```

```

public void mostrarTablero() {
    for (int i = 0; i < 24; i++) {
        System.out.print("Punto " + (i + 1) + ": ");
        if (puntos.get(i).size() > 0) {
            for (Ficha ficha : puntos.get(i)) {
                System.out.print(ficha.toString() + " ");
            }
        }
        System.out.println();
    }
}

```

```

// Mover ficha de un punto a otro
public boolean moverFicha(int desde, int hasta, String color) {
    if (desde < 0 || hasta < 0 || desde >= 24 || hasta >= 24) {
        System.out.println("Movimiento no válido: puntos fuera de rango.");
        return false;
    }
}

```

```

List<Ficha> puntoDesde = puntos.get(desde);
List<Ficha> puntoHasta = puntos.get(hasta);

```

```

if (color.equals("Blanco")) {
    if (barajaBlancas.isEmpty()) {
        System.out.println("No hay más fichas blancas en la baraja.");
        return false;
    }
    if (puntoDesde.isEmpty() || puntoDesde.get(0).getColor().equals("Negro")) {
        System.out.println("No puedes mover una ficha negra.");
        return false;
    }
    puntoHasta.add(new Ficha("Blanco"));
    barajaBlancas.remove(barajaBlancas.size() - 1);
    puntoDesde.remove(0); // Se mueve la ficha del punto de inicio
} else {
    if (barajaNegras.isEmpty()) {
        System.out.println("No hay más fichas negras en la baraja.");
        return false;
    }
    if (puntoDesde.isEmpty() || puntoDesde.get(0).getColor().equals("Blanco")) {
        System.out.println("No puedes mover una ficha blanca.");
        return false;
    }
    puntoHasta.add(new Ficha("Negro"));
    barajaNegras.remove(barajaNegras.size() - 1);
    puntoDesde.remove(0);
}
return true;

```

```
}
```

```
// Comprobar si el jugador ha ganado (por ejemplo, si ha llegado con todas las  
piezas a la casilla 25)
```

```
public boolean haGanado(String color) {  
    if (color.equals("Blanco")) {  
        return barajaBlancas.isEmpty(); // Si no hay
```

```

import java.util.Scanner;

public class JuegoBackgammon {
    private Tablero tablero;
    private String jugadorActual;

    // Constructor para iniciar el juego
    public JuegoBackgammon() {
        tablero = new Tablero();
        jugadorActual = "Blanco"; // Comienza el jugador blanco
    }

    // Método para cambiar el turno
    public void cambiarTurno() {
        if (jugadorActual.equals("Blanco")) {
            jugadorActual = "Negro";
        } else {
            jugadorActual = "Blanco";
        }
    }

    // Método para iniciar el juego
    public void jugar() {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("Turno del jugador: " + jugadorActual);
            tablero.mostrarTablero();

            // Pedir movimiento
            System.out.print("Introduce el punto de salida: ");
            int desde = scanner.nextInt() - 1;
            System.out.print("Introduce el punto de destino: ");
            int hasta = scanner.nextInt() - 1;

            if (tablero.moverFicha(desde, hasta, jugadorActual)) {
                if (tablero.haGanado(jugadorActual)) {
                    System.out.println("¡El jugador " + jugadorActual + " ha ganado!");
                    break;
                }
                cambiarTurno();
            }
        }
        scanner.close();
    }

    public static void main(String[] args) {
        JuegoBackgammon juego = new JuegoBackgammon();
    }
}

```

```
        System.out.println("¡Bienvenido al Backgammon!");  
        juego.jugar();  
    }  
}
```