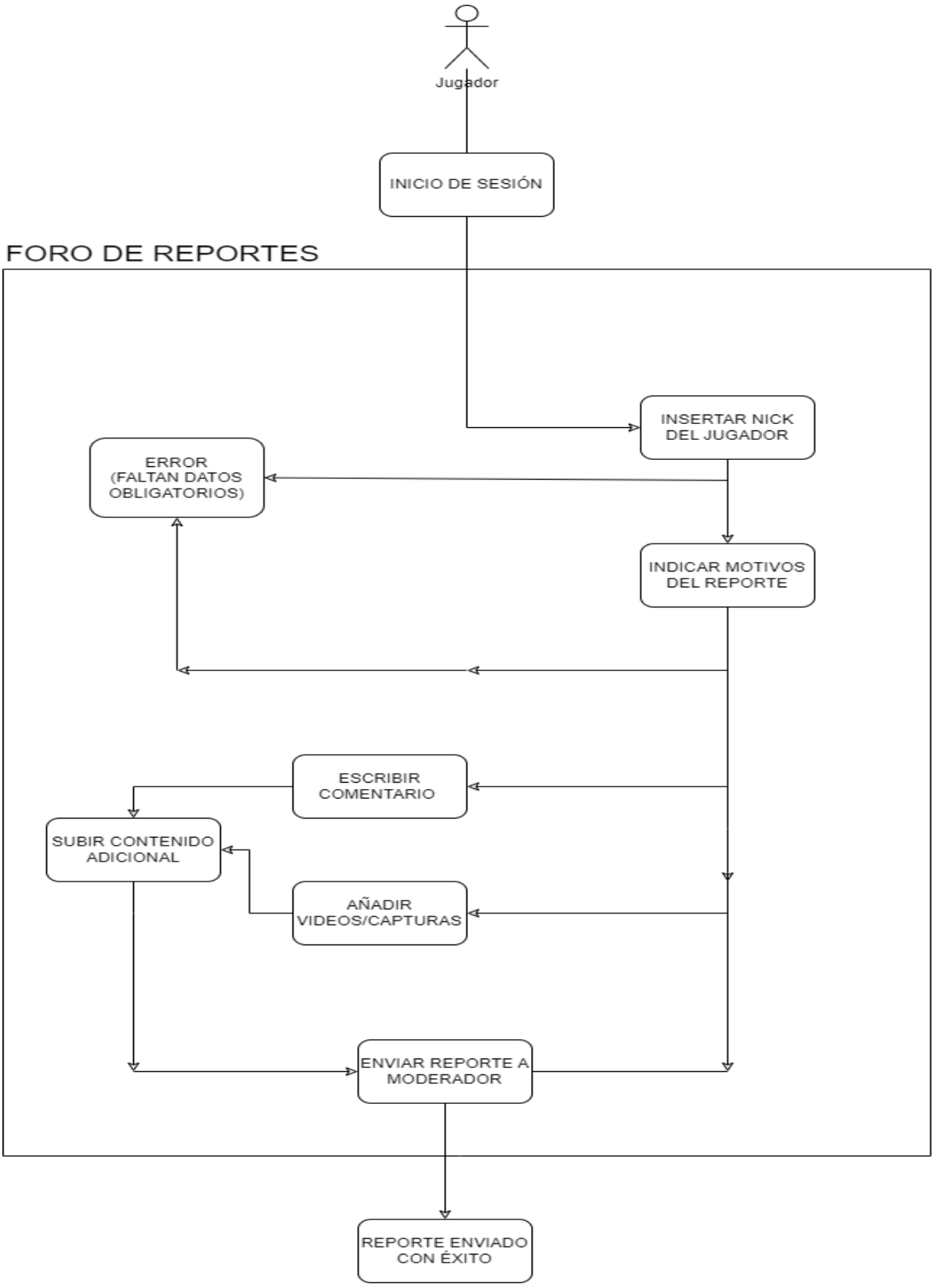
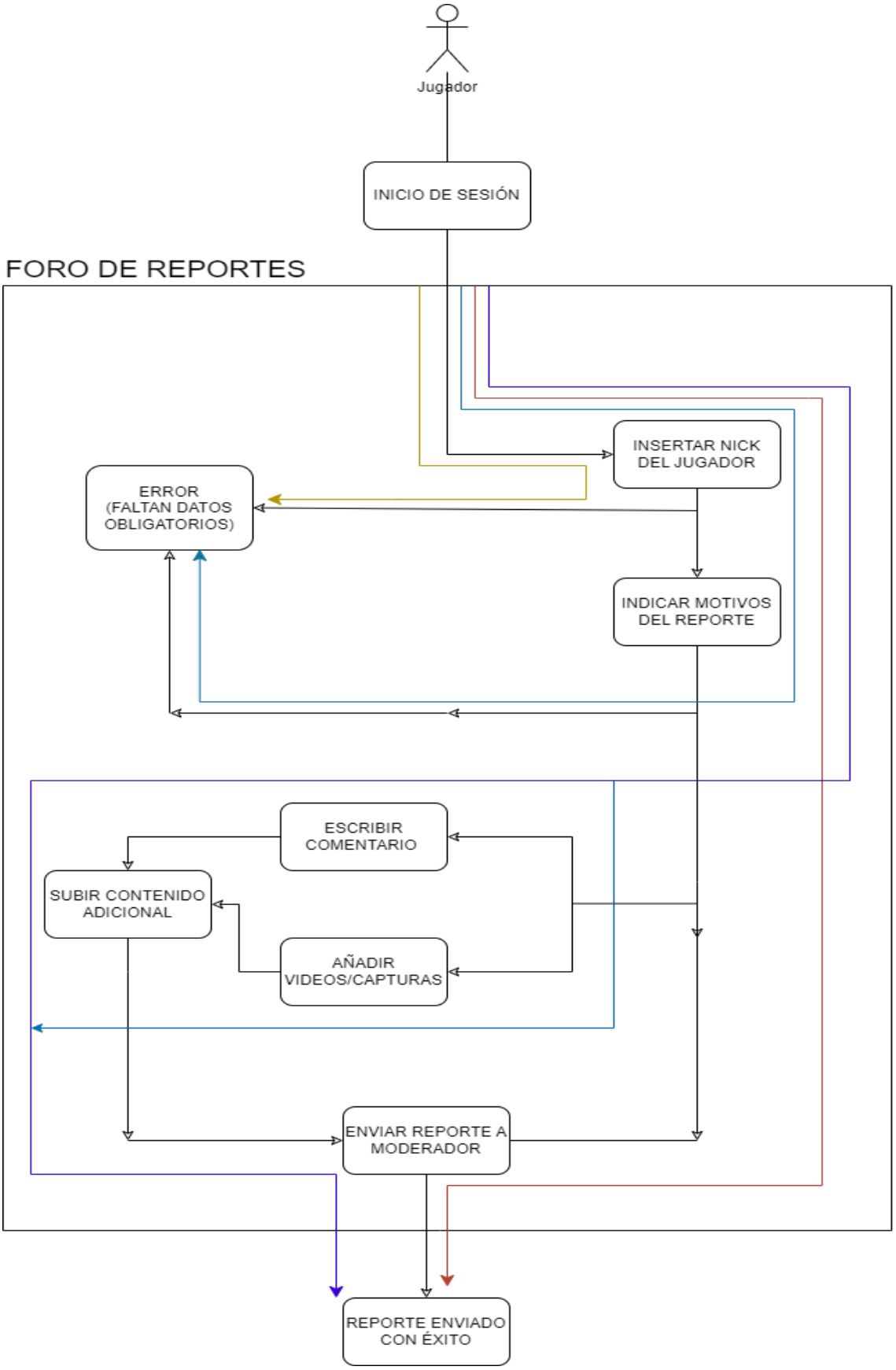
CAJA BLANCA:

GRAFO DE FLUJO:



CAMINOS CALCULADOS FUNCIONALES:



-. CAMINO AMARILLO: Insertar nick del jugador (no) -> Error, faltan datos obligatorios, se acaba el flujo.

-. CAMINO CIAN: Insertar nick del jugador (si) -> Indicar motivos del reporte (no) -> Error, faltan datos obligatorios, se acaba el flujo.

-. CAMINO ROJO: Insertar nick del jugador (si) -> Indicar motivos del reporte (si) -> Enviar reporte a moderador -> Reporte enviado con éxito, se acaba el flujo.

-. CAMINO MORADO: Insertar nick del jugador (si) -> Indicar motivos del reporte (si) -> Escribir comentario (si) -> -> Añadir videos/capturas(si) Subir contenido adicional -> Enviar reporte a moderador -> Reporte enviado con éxito, se acaba el flujo.

CASOS DE PRUEBA:

```java
public class ReporteTest {

    public ReporteTest() {
    }
@Test
    public void testFormularioValido() {

        Reporte Freporte = new Reporte();

        Freporte.setNickReportado("Player");

        Freporte.getMotivos().add("Palabras malsonantes");

        Freporte.getMotivos().add("Comentarios racistas");

        Freporte.setComentario("El jugador ha estado insultando durante toda la partida.");
        assertTrue(Freporte.formularioValido());
        System.out.println(Freporte.formularioValido());
```

```java
public class Reporte {
    /**
     * La clase reporte almacenará todos los datos recogidos del usuario que rellena el formulario de reportes.
     * Declaramos las variables necesarias para almacenar cada información proporcionada por el usuario:
     * Variable nickReportado de tipo String que almacenará el nick del usuario que va a ser reportado.
     * Creamos un ArrayList de tipo String que almacenará todos los motivos por los que el usuario considera que el otro jugador debe ser reportado
     * Variable comentario de tipo String que almacenara el comentario que puede proporcionar el usuario.
     * Creamos otro ArrayList para almacenar las fotos que puede aportar el usuario como pruebas.
     *
     */
    private String nickReportado;
    private ArrayList<String>  motivos = new ArrayList<String>();
    private String comentario;
    private ArrayList<BufferedImage> imagenes = new ArrayList<BufferedImage>();

    public String getNickReportado() {
        return nickReportado;
    }

    public void setNickReportado(String nickReportado) {
        this.nickReportado = nickReportado;
    }

    public ArrayList<String> getMotivos() {
        return motivos;
    }

    public void setMotivos(ArrayList<String> motivos) {
        this.motivos = motivos;
    }

    public String getComentario() {
        return comentario;
    }
```

```java
public void setComentario(String comentario) {
    this.comentario = comentario;
}

public ArrayList<BufferedImage> getImagenes() {
    return imagenes;
}

public void setImagenes(ArrayList<BufferedImage> imagenes) {
    this.imagenes = imagenes;
}

/**
 * Creamos el método formularioValido que dará por válido el formulario que tenga el nick del jugador reportado
 * y al menos un motivo para que así cuando el formulario llegue al moderador solo tenga que verificar que el nick exsiste
 * y que los motivos y las pruebas son suficientes para banear al jugador. Con este método todos los formularios que no contengan
 * nick o motivos para el ban seran excluidos automáticamente.
 * @return
 */
public boolean formularioValido(){
boolean valido = true;
if(nickReportado == null){
    valido = false;
}
if(motivos == null){
    valido = false;
}
return valido;


public static void main(String[] args){
    Reporte Freporte = new Reporte();
    System.out.println(Freporte.formularioValido());
```