

# Identificación del comportamiento no observable de un DES emulado, a partir de un generador de lenguaje.

ANEXOS



Rubén Darío Muñoz Chávez

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Electrónica, Instrumentación y Control.

Ingeniería en Automática Industrial

Popayán, Junio de 2017



# Identificación del comportamiento no observable de un DES emulado, a partir de un generador de lenguaje.

ANEXOS



Rubén Darío Muñoz Chávez

Director: PhD. Mariela Muñoz Añasco

Anexos parte de la monografía presentada como requisito parcial para optar por el título de Ingeniero en Automática Industrial

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Electrónica, Instrumentación y Control.

Ingeniería en Automática Industrial

Popayán, Junio de 2017



# Índice general

<b>1 Anexo A - Códigos de las etapas del proceso</b>	<b>7</b>
1.1 Códigos generados del SED . . . . .	8
1.1.1 Subsistema suministro de materias primas . . . . .	8
1.1.2 Subsistema molienda de material . . . . .	14
1.1.3 Subsistema atomizado y secado . . . . .	19
1.1.4 Subsistema prensado . . . . .	24
1.1.5 Subsistema girado . . . . .	30
1.1.6 Subsistema secado por luz UV . . . . .	36
1.1.7 Subsistema impresión . . . . .	40
1.1.8 Subsistema horneado . . . . .	44
<b>2 Anexo B - Integración herramienta Graphviz</b>	<b>49</b>
2.1 Comprobar Integración CMD comando dot . . . . .	49
2.2 Integración CMD en Windows 10, 8 y 7 . . . . .	50
<b>3 Anexo C - Modelos Identificados para el caso de estudio</b>	<b>55</b>
3.1 Autómatas Con base al método de Klein: . . . . .	55
3.1.1 Subsistema suministro de materias primas . . . . .	56
3.1.2 Subsistema molienda de material . . . . .	60
3.1.3 Subsistema atomizado y secado . . . . .	62
3.1.4 Subsistema prensado . . . . .	65
3.1.5 Subsistema girado . . . . .	70
3.1.6 Subsistema secado por luz UV . . . . .	74
3.1.7 Subsistema impresión . . . . .	76
3.1.8 Subsistema horneado . . . . .	78
3.2 IPNs Con base al método de Estrada - Vargas : . . . . .	80
3.2.1 Subsistema suministro de materias primas . . . . .	81
3.2.2 Subsistema molienda de material . . . . .	87
3.2.3 Subsistema atomizado y secado . . . . .	90
3.2.4 Subsistema prensado . . . . .	95
3.2.5 Subsistema girado . . . . .	104
3.2.6 Subsistema secado por luz UV . . . . .	111
3.2.7 Subsistema impresión . . . . .	115
3.2.8 Subsistema horneado . . . . .	119



# Índice de figuras

1.1. Código Banda transportadora de suministro . . . . .	8
1.2. Código Compuerta de suministro . . . . .	8
1.3. Código Tolva de suministro . . . . .	9
1.4. Código Planta suministro . . . . .	10
1.5. Código Controlador Suministro Parte 1 . . . . .	11
1.6. Código Controlador Suministro Parte 2 . . . . .	12
1.7. Código Controlador Suministro Parte 3 . . . . .	13
1.8. Código Tanque de molienda de material . . . . .	14
1.9. Código Tanque grande de molienda de material . . . . .	15
1.10. Código Planta molienda de material . . . . .	16
1.11. Código Controlador molienda de material Parte 1 . . . . .	17
1.12. Código Controlador molienda de material Parte 2 . . . . .	18
1.13. Código Tanque de atomizado y secado . . . . .	19
1.14. Código Banda2 de atomizado y secado . . . . .	19
1.15. Código Banda1 de atomizado y secado . . . . .	20
1.16. Código Temperatura de atomizado y secado . . . . .	20
1.17. Código Planta atomizado y secado . . . . .	21
1.18. Código Controlador atomizado y secado Parte 1 . . . . .	22
1.19. Código Controlador atomizado y secado Parte 2 . . . . .	23
1.20. Código Paleta de prensado . . . . .	24
1.21. Código Prensa de prensado . . . . .	25
1.22. Código Planta prensado . . . . .	26
1.23. Código Controlador prensado Parte 1 . . . . .	27
1.24. Código Controlador prensado Parte 2 . . . . .	28
1.25. Código Controlador prensado Parte 3 . . . . .	29
1.26. Código Mecanismo de girado . . . . .	30
1.27. Código Banda1 de girado . . . . .	30
1.28. Código Banda2 de girado . . . . .	31
1.29. Código Planta girado . . . . .	32
1.30. Código Controlador girado Parte 1 . . . . .	33
1.31. Código Controlador girado Parte 2 . . . . .	34
1.32. Código Controlador girado Parte 3 . . . . .	35
1.33. Código Banda de secado por luz UV . . . . .	36
1.34. Código Planta secado por luz UV . . . . .	37

1.35. Código Controlador secado por luz UV Parte 1 . . . . .	38
1.36. Código Controlador secado por luz UV Parte 2 . . . . .	39
1.37. Código Banda de impresión . . . . .	40
1.38. Código Planta impresión . . . . .	41
1.39. Código Controlador impresión Parte 1 . . . . .	42
1.40. Código Controlador impresión Parte 2 . . . . .	43
1.41. Código Banda de horneado . . . . .	44
1.42. Código Temperatura de horneado . . . . .	45
1.43. Código Planta horneado . . . . .	46
1.44. Código Controlador horneado Parte 1 . . . . .	47
1.45. Código Controlador horneado Parte 2 . . . . .	48
2.1. Ventana CMD comando dot . . . . .	49
2.2. Abrir Propiedades del Sistema . . . . .	50
2.3. Ventana Propiedades del Sistema . . . . .	51
2.4. Ventana Opciones Avanzadas . . . . .	52
2.5. Ventana Variables de Entorno . . . . .	53
2.6. Ventana Editar Variables de Entorno . . . . .	54
3.1. Autómata identificado subsistema suministro Parte 1 . . . . .	56
3.2. Autómata identificado subsistema suministro Parte 2 . . . . .	57
3.3. Autómata identificado subsistema molino . . . . .	60
3.4. Autómata identificado subsistema atomizado y secado Parte 1 . . . . .	62
3.5. Autómata identificado subsistema atomizado y secado Parte 2 . . . . .	63
3.6. Autómata identificado subsistema prensado Parte 1 . . . . .	65
3.7. Autómata identificado subsistema prensado Parte 2 . . . . .	66
3.8. Autómata identificado subsistema prensado Parte 3 . . . . .	67
3.9. Autómata identificado subsistema girado Parte 1 . . . . .	70
3.10. Autómata identificado subsistema girado Parte 2 . . . . .	71
3.11. Autómata identificado subsistema secado por luz UV . . . . .	74
3.12. Autómata identificado subsistema impresión . . . . .	76
3.13. Autómata identificado subsistema horneado . . . . .	78
3.14. PN identificada subsistema suministro Parte 1 . . . . .	81
3.15. PN identificada subsistema suministro Parte 2 . . . . .	82
3.16. PN identificada subsistema molino . . . . .	87
3.17. PN identificada subsistema atomizado y secado Parte 1 . . . . .	90
3.18. PN identificada subsistema atomizado y secado Parte 2 . . . . .	91
3.19. PN identificada subsistema prensado Parte 1 . . . . .	95
3.20. PN identificada subsistema prensado Parte 2 . . . . .	96
3.21. PN identificada subsistema prensado Parte 3 . . . . .	97
3.22. PN identificada subsistema girado Parte 1 . . . . .	104
3.23. PN identificada subsistema girado Parte 2 . . . . .	105



3.24. PN identificada subsistema girado Parte 3 . . . . .	106
3.25. PN identificada subsistema secado por luz UV Parte 1 . . . . .	111
3.26. PN identificada subsistema secado por luz UV Parte 2 . . . . .	112
3.27. PN identificada subsistema impresión Parte 1 . . . . .	115
3.28. PN identificada subsistema impresión Parte 2 . . . . .	116
3.29. PN identificada subsistema horneado Parte 1 . . . . .	119
3.30. PN identificada subsistema horneado Parte 2 . . . . .	120



# Índice de tablas

3.1. Código Automata subsistema suministro . . . . .	58
3.2. Código Automata subsistema molino . . . . .	61
3.3. Código Automata subsistema atomizado y secado . . . . .	64
3.4. Código Automata subsistema prensado Parte 1 . . . . .	68
3.5. Código Automata subsistema prensado Parte 2 . . . . .	69
3.6. Código Automata subsistema girado Parte 1 . . . . .	72
3.7. Código Automata subsistema girado Parte 2 . . . . .	73
3.8. Código Automata subsistema secado por luz UV . . . . .	75
3.9. Código Automata subsistema impresión . . . . .	77
3.10. Código Automata subsistema horneado . . . . .	79
3.11. PN subsistema de suministro Parte 1 . . . . .	83
3.12. PN subsistema de suministro Parte 2 . . . . .	84
3.13. PN subsistema de suministro Parte 3 . . . . .	85
3.14. PN subsistema de suministro Parte 4 . . . . .	86
3.15. Código PN subsistema molino Parte 1 . . . . .	88
3.16. Código PN subsistema molino Parte 2 . . . . .	89
3.17. Código PN subsistema atomizado y secado Parte 1 . . . . .	92
3.18. Código PN subsistema atomizado y secado Parte 2 . . . . .	93
3.19. Código PN subsistema atomizado y secado Parte 3 . . . . .	94
3.20. Código PN subsistema prensado Parte 1 . . . . .	98
3.21. Código PN subsistema prensado Parte 2 . . . . .	99
3.22. Código PN subsistema prensado Parte 3 . . . . .	100
3.23. Código PN subsistema prensado Parte 4 . . . . .	101
3.24. Código PN subsistema prensado Parte 5 . . . . .	102
3.25. Código PN subsistema prensado Parte 6 . . . . .	103
3.26. Código PN subsistema girado Parte 1 . . . . .	107
3.27. Código PN subsistema girado Parte 2 . . . . .	108
3.28. Código PN subsistema girado Parte 3 . . . . .	109
3.29. Código PN subsistema girado Parte 4 . . . . .	110
3.30. Código PN subsistema secado por luz UV Parte 1 . . . . .	113
3.31. Código PN subsistema secado por luz UV Parte 2 . . . . .	114
3.32. Código PN subsistema impresión Parte 1 . . . . .	117
3.33. Código PN subsistema impresión Parte 2 . . . . .	118
3.34. Código PN subsistema horneado Parte 1 . . . . .	121

3.35. Código PN subsistema horneado Parte 2 . . . . .	122
---	-----

# 1 Anexo A - Códigos de las etapas del proceso

Los códigos generados a partir de los procedimientos planteados para la integración de planta controlador resultan ser demasiado numerosos generando un gran número de ficheros independientes, por ello en esta subsección solo se describirá brevemente los códigos generados en cada subsistema, además el código correspondiente a la planta de cada subsistema, se estableció con el mismo tiempo de actualización por ello da exactamente igual el realizar los segmentos de código de planta por separado como también el unirlos todos en un único archivo de planta, esto solo aplica para los códigos de planta.

En algunas figuras relacionadas con bandas transportadoras aparecen 2 franjas rojas, estas indican que se encoje la imagen debido a lo largo de algunas líneas que impiden la visualización del código.



```

//Suministro_Tolva

boolean Material_Suministro_Tolva = true; // Usada para representar la tarea de adición de material en la Tolva
// por parte del operario

int16_t LLenado_Suministro_Tolva = 0; //Indica el nivel de LLenado en Unidades Volumetricas

#define LLenado_Suministro_Tolva_Alto 300 // 300 Unidades Volumetricas
#define LLenado_Suministro_Tolva_Medio 100 // 100 Unidades Volumetricas
#define LLenado_Suministro_Tolva_Bajo 0 // 0 Unidades Volumetricas

#define LLenado_Suministro_Tolva_Razon 4 // Velocidad de LLenado [Unidades Volumetricas / Delta de Tiempo]
#define Vaciado_Suministro_Tolva_Razon 2 // Velocidad de Vaciado [Unidades Volumetricas / Delta de Tiempo]

void Suministro_Tolva ()
{
  if (LLenado_Suministro_Tolva >= LLenado_Suministro_Tolva_Alto)
  {
    digitalWrite(S3, HIGH);
    //LLenado_Suministro_Tolva = LLenado_Suministro_Tolva_Alto;
  }
  else
  {
    digitalWrite(S3, LOW );
  }
  if (LLenado_Suministro_Tolva >= LLenado_Suministro_Tolva_Medio)
  {
    digitalWrite(S2, HIGH);
  }
  else
  {
    digitalWrite(S2, LOW );
  }
  if (LLenado_Suministro_Tolva > LLenado_Suministro_Tolva_Bajo)
  {
    digitalWrite(S1, HIGH);
  }
  else
  {
    digitalWrite(S1, LOW );
    LLenado_Suministro_Tolva = LLenado_Suministro_Tolva_Bajo;
  }
  ////////////////////////////////////////////////////
}

```

Figura 1.3: Código Tolva de suministro

Para el comportamiento de planta se obtuvo la Figura 1.4

```

//Planta

boolean Activado_Planta = true;
#define Tiempo_Planta 200
uint32_t tiempo_Actual_Planta;
int64_t tiempo_Total_Planta;

void Planta ()
{
  if (Activado_Planta)
  {
    tiempo_Actual_Planta = millis();
    Activado_Planta = false;
  }
  tiempo_Total_Planta = millis() - tiempo_Actual_Planta;
  if (tiempo_Total_Planta >= Tiempo_Planta)
  {
    //////////////////////////////////////
    if (digitalRead(Compuerta_Der))
    {
      Apertura_Suministro_Compuerta = Apertura_Suministro_Compuerta + Apertura_Suministro_Compuerta_Razon;
    }
    if (digitalRead(Compuerta_Izq))
    {
      Apertura_Suministro_Compuerta = Apertura_Suministro_Compuerta - Apertura_Suministro_Compuerta_Razon;
    }
    //////////////////////////////////////
    if ((digitalRead(Luz_Material) == 1) && Material_Suministro_Tolva)
    {
      LLenado_Suministro_Tolva = LLenado_Suministro_Tolva + LLenado_Suministro_Tolva_Razon;
    }
    //////////////////////////////////////
    if (digitalRead(Banda1))
    {
      for (int i = 1; i <= Bits_Suministro_Banda; i++)
      {
        Cadena_Suministro_Banda[Bits_Suministro_Banda - i] = Cadena_Suministro_Banda[Bits_Suministro_Banda - 1 - i];
      }
      if ((Apertura_Suministro_Compuerta > Apertura_Suministro_Compuerta_Medio)
          && (LLenado_Suministro_Tolva > LLenado_Suministro_Tolva_Bajo))
      {
        Cadena_Suministro_Banda[0] = 1;
        LLenado_Suministro_Tolva = LLenado_Suministro_Tolva - Vaciado_Suministro_Tolva_Razon;
      }
      else
      {
        Cadena_Suministro_Banda[0] = 0;
      }
    }
    //////////////////////////////////////
    tiempo_Actual_Planta = millis();
  }
}

```

**Figura 1.4:** Código Planta suministro

Por último se genera la Figura 1.5, Figura 1.6, Figura 1.7 que corresponden al comportamiento del controlador y corresponde fielmente al diseñado en el modelo de máquina de estado.



```

//Suministro_Controlador
int EA_Suministro = 1; //Estado Actual

void Suministro_Salidas (unsigned char a, unsigned char b, unsigned char c, unsigned char d)
{
    digitalWrite(Banda1, a);
    digitalWrite(Compuerta_Izq, b);
    digitalWrite(Compuerta_Der, c);
    digitalWrite(Luz_Material, d);
}

boolean Suministro_Estado_Entradas (unsigned char a, unsigned char b, unsigned char c,
    unsigned char d, unsigned char e, unsigned char f, unsigned char g, unsigned char h, unsigned char i)
{
    boolean resultado = true;
    if (Activado_Estado(a))
    {
        resultado = resultado && (digitalRead(S0) == a);
    }
    if (Activado_Estado(b))
    {
        resultado = resultado && (digitalRead(S1) == b);
    }
    if (Activado_Estado(c))
    {
        resultado = resultado && (digitalRead(S2) == c);
    }
    if (Activado_Estado(d))
    {
        resultado = resultado && (digitalRead(S3) == d);
    }
    if (Activado_Estado(e))
    {
        resultado = resultado && (digitalRead(S4) == e);
    }
    if (Activado_Estado(f))
    {
        resultado = resultado && (digitalRead(S5) == f);
    }
    if (Activado_Estado(g))
    {
        resultado = resultado && (digitalRead(S6) == g);
    }
    if (Activado_Estado(h))
    {
        resultado = resultado && (digitalRead(S9) == h);
    }
    if (Activado_Estado(i))
    {
        resultado = resultado && (digitalRead(S10) == i);
    }
    return resultado;
}

```

Figura 1.5: Código Controlador Suministro Parte 1

```

void Suministro_Controlador ()
{
    switch (EA_Suministro)
    {
        case 1:
            if (Suministro_Estado_Entradas(1, x, x, x, 1, 0, 0, 0, 0))
            {
                Suministro_Salidas(0, 0, 0, 1);
                EA_Suministro = 2;
            }
            break;
        case 2:
            if (Suministro_Estado_Entradas(1, 1, 1, 1, 1, 0, 0, x, x))
            {
                Suministro_Salidas(1, 0, 1, 0);
                EA_Suministro = 3;
            }
            else if (Suministro_Estado_Entradas(0, x, x, x, 1, 0, 0, x, x))
            {
                Suministro_Salidas(0, 0, 0, 0);
                EA_Suministro = 1;
            }
            break;
        case 3:
            if (Suministro_Estado_Entradas(1, 1, x, x, 0, 1, x, x, x))
            {
                Suministro_Salidas(1, 0, 0, 0);
                EA_Suministro = 4;
            }
            else if (Suministro_Estado_Entradas(0, x, x, x, x, x, x, x, x))
            {
                Suministro_Salidas(1, 1, 0, 0);
                EA_Suministro = 6;
            }
            break;
        case 4:
            if (Suministro_Estado_Entradas(1, x, x, x, 0, 1, x, 1, 1))
            {
                Suministro_Salidas(1, 1, 0, 0);
                EA_Suministro = 6;
            }
            else if (Suministro_Estado_Entradas(1, 1, 0, 0, 0, 1, 1, x, x))
            {
                Suministro_Salidas(1, 0, 0, 1);
                EA_Suministro = 5;
            }
            else if (Suministro_Estado_Entradas(1, 0, 0, 0, 1, 0, 0, x, x))
            {
                Suministro_Salidas(0, 0, 0, 0);
                EA_Suministro = 1;
            }
            else if (Suministro_Estado_Entradas(1, x, x, x, 1, 0, 0, x, 1))
            {
                Suministro_Salidas(0, 0, 0, 0);
                EA_Suministro = 1;
            }
    }
}

```

Figura 1.6: Código Controlador Suministro Parte 2

```

else if (Suministro_Estado_Entradas(0, x, x, x, 0, 1, x, x, x))
{
    Suministro_Salidas(1, 1, 0, 0);
    EA_Suministro = 6;
}
else if (Suministro_Estado_Entradas(0, x, x, x, 1, 0, 0, x, x))
{
    Suministro_Salidas(0, 0, 0, 0);
    EA_Suministro = 1;
}
break;
case 5:
if (Suministro_Estado_Entradas(1, 0, 0, 0, 0, 1, x, x, x))
{
    Suministro_Salidas(1, 1, 0, 0);
    EA_Suministro = 6;
}
else if (Suministro_Estado_Entradas(1, x, x, x, 0, 1, x, 1, 1))
{
    Suministro_Salidas(1, 1, 0, 0);
    EA_Suministro = 6;
}
else if (Suministro_Estado_Entradas(1, 1, 1, 1, 0, 1, 1, x, x))
{
    Suministro_Salidas(1, 0, 0, 0);
    EA_Suministro = 4;
}
else if (Suministro_Estado_Entradas(0, x, x, x, 0, 1, x, x, x))
{
    Suministro_Salidas(1, 1, 0, 0);
    EA_Suministro = 6;
}
break;
case 6:
if (Suministro_Estado_Entradas(x, x, x, x, 1, 0, x, x, x))
{
    Suministro_Salidas(1, 0, 0, 0);
    EA_Suministro = 4;
}
break;
}
}

```

Figura 1.7: Código Controlador Suministro Parte 3

### 1.1.2. Subsistema molienda de material

Para los elementos de planta correspondientes al subsistema se generaron las Figura 1.8, Figura 1.9:

```
//Molino_Tanque

int16_t LLenado_Molino_Tanque = 0;

#define LLenado_Molino_Tanque_Alto 600
#define LLenado_Molino_Tanque_Bajo 0

//[Unidades volumetricas]/[Delta Tiempo]
#define LLenado_Molino_Tanque_Valvula 1 // Agua
#define LLenado_Molino_Tanque_Material 2 // Material de entrada
#define Vaciado_Molino_Tanque_Bomba 5 // Asociado con la Bomba

void Molino_Tanque ()
{
    if (LLenado_Molino_Tanque >= LLenado_Molino_Tanque_Alto)
    {
        digitalWrite(S7, HIGH);
        //LLenado_Molino_Tanque = LLenado_Molino_Tanque_Alto;
    }
    else
    {
        digitalWrite(S7, LOW );
    }
    if (LLenado_Molino_Tanque > LLenado_Molino_Tanque_Bajo)
    {
        digitalWrite(S8, HIGH);
    }
    else
    {
        digitalWrite(S8, LOW );
        LLenado_Molino_Tanque = LLenado_Molino_Tanque_Bajo;
    }
    //////////////////////////////////////
}
}
```

Figura 1.8: Código Tanque de molienda de material

```

//Molino_Tanque_grande

int16_t LLenado_Molino_Tanque_Grande = 0;

#define LLenado_Molino_Tanque_Grande_Alto 5000
#define LLenado_Molino_Tanque_Grande_Medio 900
#define LLenado_Molino_Tanque_Grande_Bajo 0

// #define LLenado_Molino_Tanque_Grande_Razon 4
#define Vaciado_Molino_Tanque_Grande_Razon 2

void Molino_Tanque_Grande ()
{
  if (LLenado_Molino_Tanque_Grande >= LLenado_Molino_Tanque_Grande_Alto)
  {
    digitalWrite( S9, HIGH);
    //LLenado_Molino_Tanque_Grande = LLenado_Molino_Tanque_Grande_Alto;
  }
  else
  {
    digitalWrite( S9, LOW );
  }
  if (LLenado_Molino_Tanque_Grande >= LLenado_Molino_Tanque_Grande_Medio)
  {
    digitalWrite(S10, HIGH);
  }
  else
  {
    digitalWrite(S10, LOW );
  }
  if (LLenado_Molino_Tanque_Grande > LLenado_Molino_Tanque_Grande_Bajo)
  {
    digitalWrite(S11, HIGH);
  }
  else
  {
    digitalWrite(S11, LOW );
    LLenado_Molino_Tanque_Grande = LLenado_Molino_Tanque_Grande_Bajo;
  }
  //////////////////////////////////////
}

```

Figura 1.9: Código Tanque grande de molienda de material

Para el comportamiento de planta se obtuvo la Figura 1.10

```
//Planta

boolean Activado_Planta = true;
#define Tiempo_Planta 200
uint32_t tiempo_Actual_Planta;
int64_t tiempo_Total_Planta;

void Planta ()
{
  if (Activado_Planta)
  {
    tiempo_Actual_Planta = millis();
    Activado_Planta = false;
  }
  tiempo_Total_Planta = millis() - tiempo_Actual_Planta;
  if (tiempo_Total_Planta >= Tiempo_Planta)
  {
    //////////////////////////////////////
    //////////////////////////////////////
    if (digitalRead(Molino))
    {
      LLenado_Molino_Tanque = LLenado_Molino_Tanque + LLenado_Molino_Tanque_Material;
    }
    //////////////////////////////////////
    if (digitalRead(Valvula1))
    {
      LLenado_Molino_Tanque = LLenado_Molino_Tanque + LLenado_Molino_Tanque_Valvula;
    }
    //////////////////////////////////////
    if (digitalRead(Bomba1))
    {
      LLenado_Molino_Tanque = LLenado_Molino_Tanque - Vaciado_Molino_Tanque_Bomba;
      LLenado_Molino_Tanque_Grande = LLenado_Molino_Tanque_Grande + Vaciado_Molino_Tanque_Bomba;
    }
    //////////////////////////////////////
    //////////////////////////////////////
    tiempo_Actual_Planta = millis();
  }
}
```

Figura 1.10: Código Planta molienda de material

Por último se genera la Figura 1.11, Figura 1.12 que corresponden al comportamiento del controlador y corresponde fielmente al diseñado en el modelo de máquina de estado.

```

//Molino_Controlador
int EA_Molino = 1; //Estado Actual

void Molino_Salidas (unsigned char a, unsigned char b, unsigned char c)
{
    digitalWrite(Molino,  a );
    digitalWrite(Bombal,  b );
    digitalWrite(Valvula1, c );
}

boolean Molino_Estado_Entradas (unsigned char a, unsigned char b,
                                unsigned char c, unsigned char d)
{
    boolean resultado = true;
    if (Activado_Estado(a))
    {
        resultado = resultado && (digitalRead(S0) == a);
    }
    if (Activado_Estado(b))
    {
        resultado = resultado && (digitalRead(S6) == b);
    }
    if (Activado_Estado(c))
    {
        resultado = resultado && (digitalRead(S7) == c);
    }
    if (Activado_Estado(d))
    {
        resultado = resultado && (digitalRead(S8) == d);
    }
    return resultado;
}

```

**Figura 1.11:** Código Controlador molienda de material Parte 1

```
void Molino_Controlador ()
{
  switch (EA_Molino)
  {
    case 1:
      if (Molino_Estado_Entradas(1, 1, 0, 0))
      {
        Molino_Salidas(1, 0, 1);
        EA_Molino = 2;
      }
      break;
    case 2:
      if (Molino_Estado_Entradas(x, 1, 1, 1))
      {
        Molino_Salidas(1, 1, 1);
        EA_Molino = 3;
      }
      else if (Molino_Estado_Entradas(x, 0, x, x))
      {
        Molino_Salidas(1, 1, 0);
        EA_Molino = 4;
      }
      break;
    case 3:
      if (Molino_Estado_Entradas(x, 1, 0, 0))
      {
        Molino_Salidas(1, 0, 1);
        EA_Molino = 2;
      }
      else if (Molino_Estado_Entradas(x, 0, x, x))
      {
        Molino_Salidas(1, 1, 0);
        EA_Molino = 4;
      }
      break;
    case 4:
      if (Molino_Estado_Entradas(x, 1, x, x))
      {
        Molino_Salidas(1, 1, 1);
        EA_Molino = 3;
      }
      else if (Molino_Estado_Entradas(x, 0, 0, 0))
      {
        Molino_Salidas(0, 0, 0);
        EA_Molino = 1;
      }
      break;
  }
}
```

Figura 1.12: Código Controlador molienda de material Parte 2





```

//Secado1_Banda

#define Bits_Secado1_Banda 30
// 30 bits * 200 ms = 6 seg
//Cadena_... Usada para representar el desplazamiento de objetos en la Banda
int Cadena_Secado1_Banda[Bits_Secado1_Banda] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

void Secado1_Banda()
{
    if (Cadena_Secado1_Banda[Bits_Secado1_Banda - 1])
    {
        digitalWrite(S12, HIGH);
    }
    else
    {
        digitalWrite(S12, LOW );
    }
    ////////////////////////////////////////
}

```

Figura 1.15: Código Banda1 de atomizado y secado

```

//Secado1_Temperatura

int16_t Valor_Secado1_Temperatura = 25;

#define Maxima_Secado1_Temperatura 310
#define Minima_Secado1_Temperatura 300
#define Ambiente_Secado1_Temperatura 25

#define Razon_Secado1_Temperatura_Incremento 5
#define Razon_Secado1_Temperatura_Decremento 3

void Secado1_Temperatura ()
{
    if (Valor_Secado1_Temperatura >= Maxima_Secado1_Temperatura)
    {
        Valor_Secado1_Temperatura = Maxima_Secado1_Temperatura;
    }
    if (Valor_Secado1_Temperatura >= Minima_Secado1_Temperatura)
    {
        digitalWrite(S13, HIGH );
    }
    else
    {
        digitalWrite(S13, LOW );
    }
    if (Valor_Secado1_Temperatura <= Ambiente_Secado1_Temperatura)
    {
        Valor_Secado1_Temperatura = Ambiente_Secado1_Temperatura;
    }
}

```

Figura 1.16: Código Temperatura de atomizado y secado

Para el comportamiento de planta se obtuvo la Figura 1.17

```

//Planta
boolean Activado_Planta = true;
#define Tiempo_Planta 200
uint32_t tiempo_Actual_Planta;
int64_t tiempo_Total_Planta;

void Planta ()
{
  if (Activado_Planta)
  {
    tiempo_Actual_Planta = millis();
    Activado_Planta = false;
  }
  tiempo_Total_Planta = millis() - tiempo_Actual_Planta;
  if (tiempo_Total_Planta >= Tiempo_Planta)
  {
    //////////////////////////////////////
    //////////////////////////////////////
    if (digitalRead(Quemador1))
    {
      Valor_Secado1_Temperatura = Valor_Secado1_Temperatura + Razon_Secado1_Temperatura_Incremento;
    }
    else
    {
      Valor_Secado1_Temperatura = Valor_Secado1_Temperatura - Razon_Secado1_Temperatura_Decremento;
    }
    //////////////////////////////////////
    if (digitalRead(Banda3))
    {
      if (Cadena_Secado1_Banda2[Bits_Secado1_Banda2 - 1])
      {
        LLenado_Secado1_Tanque = LLenado_Secado1_Tanque + LLenado_Secado1_Tanque_Razon;
      }
      for (int i = 1; i <= Bits_Secado1_Banda2; i++)
      {
        Cadena_Secado1_Banda2[Bits_Secado1_Banda2 - i] = Cadena_Secado1_Banda2[Bits_Secado1_Banda2 - 1 - i];
      }
      Cadena_Secado1_Banda2[0] = 0;
    }
    //////////////////////////////////////
    if (digitalRead(Banda2))
    {
      if (Cadena_Secado1_Banda[Bits_Secado1_Banda - 1])
      {
        Cadena_Secado1_Banda2[0] = 1;
      }
      for (int i = 1; i <= Bits_Secado1_Banda; i++)
      {
        Cadena_Secado1_Banda[Bits_Secado1_Banda - i] = Cadena_Secado1_Banda[Bits_Secado1_Banda - 1 - i];
      }
      //Cadena_Secado1_Banda[0] = 0;
    }
    //////////////////////////////////////
    if (digitalRead(Bomba2))
    {
      Cadena_Secado1_Banda[0] = 1;
      LLenado_Molino_Tanque_Grande = LLenado_Molino_Tanque_Grande - Vaciado_Molino_Tanque_Grande_Razon;
    }
    //////////////////////////////////////
    tiempo_Actual_Planta = millis();
  }
}

```

Figura 1.17: Código Planta atomizado y secado

Por último se genera la Figura 1.18, Figura 1.19 que corresponden al comportamiento del controlador y corresponde fielmente al diseñado en el modelo de máquina de estado.

```

//Definicion estado inicial del subprocesso
int EA_Secado1 = 1; //Estado Actual

void Secado1_Salidas(unsigned char a, unsigned char b, unsigned char c,
                    unsigned char d, unsigned char e)
{
    digitalWrite(Quemador1, a );
    digitalWrite(Ventilador, b );
    digitalWrite(Banda2,    c );
    digitalWrite(Bomba2,    d );
    digitalWrite(Banda3,    e );
}

boolean Secado1_Estado_Entradas (unsigned char a, unsigned char b, unsigned char c,
                                unsigned char d, unsigned char e, unsigned char f)
{
    boolean resultado = true;
    if (Activado_Estado(a))
    {
        resultado = resultado && (digitalRead(S0) == a);
    }
    if (Activado_Estado(b))
    {
        resultado = resultado && (digitalRead(S11) == b);
    }
    if (Activado_Estado(c))
    {
        resultado = resultado && (digitalRead(S12) == c);
    }
    if (Activado_Estado(d))
    {
        resultado = resultado && (digitalRead(S13) == d);
    }
    if (Activado_Estado(e))
    {
        resultado = resultado && (digitalRead(S15) == e);
    }
    if (Activado_Estado(f))
    {
        resultado = resultado && (digitalRead(S16) == f);
    }
    return resultado;
}

```

Figura 1.18: Código Controlador atomizado y secado Parte 1

```

void Secado1_Controlador ()
{
    switch (EA_Secado1)
    {
        case 1:
            if (Secado1_Estado_Entradas(1, 1, x, x, 0, 0))
            {
                Secado1_Salidas(1, 1, 0, 0, 0);
                EA_Secado1 = 2;
            }
            break;
        case 2:
            if (Secado1_Estado_Entradas(1, 1, 0, 1, 0, 0))
            {
                Secado1_Salidas(1, 1, 1, 1, 0);
                EA_Secado1 = 3;
            }
            else if (Secado1_Estado_Entradas(0, 1, x, x, 0, 0))
            {
                Secado1_Salidas(0, 0, 0, 0, 0);
                EA_Secado1 = 1;
            }
            break;
        case 3:
            if (Secado1_Estado_Entradas(1, 1, 1, 1, 0, 0))
            {
                Secado1_Salidas(1, 1, 1, 1, 1);
                EA_Secado1 = 4;
            }
            else if (Secado1_Estado_Entradas(0, 1, 0, 1, 0, 0))
            {
                Secado1_Salidas(0, 0, 0, 0, 0);
                EA_Secado1 = 1;
            }
            break;
        case 4:
            if (Secado1_Estado_Entradas(1, 1, 1, 1, 1, 1))
            {
                Secado1_Salidas(0, 0, 1, 0, 1);
                EA_Secado1 = 5;
            }
            else if (Secado1_Estado_Entradas(1, 0, 1, 1, x, x))
            {
                Secado1_Salidas(0, 0, 1, 0, 1);
                EA_Secado1 = 5;
            }
            else if (Secado1_Estado_Entradas(0, x, 1, 1, x, x))
            {
                Secado1_Salidas(0, 0, 1, 0, 1);
                EA_Secado1 = 5;
            }
            break;
        case 5:
            if (Secado1_Estado_Entradas(x, x, 0, x, x, x))
            {
                Secado1_Salidas(0, 0, 0, 0, 0);
                EA_Secado1 = 1;
            }
            break;
    }
}

```

Figura 1.19: Código Controlador atomizado y secado Parte 2

### 1.1.4. Subsistema prensado

Para los elementos de planta correspondientes al subsistema se generaron las Figura 1.20, Figura 1.21:

```
//Prensado_Paleta

uint8_t Posicion_Prensado_Paleta = 0;

#define Posicion_Prensado_Paleta_Fin 100 //100%
// X% para añadir nueva pieza en banda de girado
#define Posicion_Prensado_Paleta_Min 90
#define Posicion_Prensado_Paleta_Inicio 0 // 0%

#define Posicion_Prensado_Paleta_Razon 10 // [ X% / Intervalo de tiempo]
void Prensado_Paleta ()
{
  if (Posicion_Prensado_Paleta >= Posicion_Prensado_Paleta_Fin)
  {
    digitalWrite(S20, HIGH);
    digitalWrite(S19, LOW );
    Posicion_Prensado_Paleta = Posicion_Prensado_Paleta_Fin;
  }
  else if (Posicion_Prensado_Paleta > Posicion_Prensado_Paleta_Inicio)
  {
    digitalWrite(S20, LOW );
    digitalWrite(S19, LOW );
  }
  else
  {
    digitalWrite(S20, LOW );
    digitalWrite(S19, HIGH);
    Posicion_Prensado_Paleta = Posicion_Prensado_Paleta_Inicio;
  }
  if(Posicion_Prensado_Paleta == Posicion_Prensado_Paleta_Min)
  {
    digitalWrite(S21, HIGH);
  }
  //////////////////////////////////////
}
```

Figura 1.20: Código Paleta de prensado

```

//Prensado_Prensa

uint8_t Posicion_Prensado_Prensa = 0;

#define Posicion_Prensado_Prensa_Fin    100    //100%
#define Posicion_Prensado_Prensa_Min    90    // X%
#define Posicion_Prensado_Prensa_Inicio  0    // 0%

#define Posicion_Prensado_Prensa_Razon 10 // [ X% / Intervalo de tiempo]

void Prensado_Prensa ()
{
  if (Posicion_Prensado_Prensa >= Posicion_Prensado_Prensa_Fin)
  {
    digitalWrite(S18, HIGH);
    digitalWrite(S17, LOW );
    Posicion_Prensado_Prensa = Posicion_Prensado_Prensa_Fin;
  }
  else if (Posicion_Prensado_Prensa > Posicion_Prensado_Prensa_Inicio)
  {
    digitalWrite(S18, LOW );
    digitalWrite(S17, LOW );
  }
  else
  {
    digitalWrite(S18, LOW );
    digitalWrite(S17, HIGH);
    Posicion_Prensado_Prensa = Posicion_Prensado_Prensa_Inicio;
  }
  if(Posicion_Prensado_Prensa == Posicion_Prensado_Prensa_Min)
  {
    digitalWrite(S21, LOW );
  }
  //////////////////////////////////////
}

```

**Figura 1.21:** Código Prensa de prensado

Para el comportamiento de planta se obtuvo la Figura 1.22

```

//Planta

boolean Pieza_Prensado = false;
boolean Activado_Planta = true;
#define Tiempo_Planta 200
uint32_t tiempo_Actual_Planta;
int64_t tiempo_Total_Planta;

void Planta ()
{
  if (Activado_Planta)
  {
    tiempo_Actual_Planta = millis();
    Activado_Planta = false;
  }
  tiempo_Total_Planta = millis() - tiempo_Actual_Planta;
  if (tiempo_Total_Planta >= Tiempo_Planta)
  {
    //////////////////////////////////////
    //////////////////////////////////////
    if (digitalRead(Prensa_Arriba))
    {
      Posicion_Prensado_Prensa = Posicion_Prensado_Prensa - Posicion_Prensado_Prensa_Razon;
    }
    if (digitalRead(Prensa_Abajo))
    {
      Posicion_Prensado_Prensa = Posicion_Prensado_Prensa + Posicion_Prensado_Prensa_Razon;
    }
    //////////////////////////////////////
    if (digitalRead(Paleta_Fin))
    {
      Posicion_Prensado_Paleta = Posicion_Prensado_Paleta + Posicion_Prensado_Paleta_Razon;
      if (Pieza_Prensado)
      {
        if (Posicion_Prensado_Paleta == Posicion_Prensado_Paleta_Min)
        {
          //Añadir bits a la banda de ingreso para girado
          Cadena_Gira_Banda[0] = 1;
          Cadena_Gira_Banda[1] = 1;
          Cadena_Gira_Banda[2] = 1;
        }
      }
    }
    if (digitalRead(paleta_Inicio))
    {
      Pieza_Prensado = true;
      Posicion_Prensado_Paleta = Posicion_Prensado_Paleta - Posicion_Prensado_Paleta_Razon;
    }
    //////////////////////////////////////
    if (digitalRead(Aspiradora) && digitalRead(Valvula2))
    { // 1 pieza ceramica = 10 Unidades volumetricas
      LLenado_Secado1_Tanque = LLenado_Secado1_Tanque - Vaceado_Secado1_Tanque_Razon;
    }
    //////////////////////////////////////
    //////////////////////////////////////
    tiempo_Actual_Planta = millis();
  }
}

```

Figura 1.22: Código Planta prensado

Por último se genera la Figura 1.23, Figura 1.24, Figura 1.25 que corresponden al comportamiento del controlador y corresponde fielmente al diseñado en el modelo de



máquina de estado.

```
//Definicion estado inicial del subproceso
int EA_Prensado = 1; //Estado Actual

void Prensado_Salidas (unsigned char a, unsigned char b, unsigned char c, unsigned char d,
                      unsigned char e, unsigned char f)
{
    digitalWrite(Prensa_Arriba, a);
    digitalWrite(Prensa_Abajo, b);
    digitalWrite(paleta_Inicio, c);
    digitalWrite(Paleta_Fin, d);
    digitalWrite(Valvula2, e);
    digitalWrite(Aspiradora, f);
}

boolean Prensado_Estado_Entradas (unsigned char a, unsigned char b, unsigned char c, unsigned char d,
                                  unsigned char e, unsigned char f, unsigned char g, unsigned char h)
{
    boolean resultado = true;
    if (Activado_Estado(a))
    {
        resultado = resultado && (digitalRead(S0) == a);
    }
    if (Activado_Estado(b))
    {
        resultado = resultado && (digitalRead(S14) == b);
    }
    if (Activado_Estado(c))
    {
        resultado = resultado && (digitalRead(S17) == c);
    }
    if (Activado_Estado(d))
    {
        resultado = resultado && (digitalRead(S18) == d);
    }
    if (Activado_Estado(e))
    {
        resultado = resultado && (digitalRead(S19) == e);
    }
    if (Activado_Estado(f))
    {
        resultado = resultado && (digitalRead(S20) == f);
    }
    if (Activado_Estado(g))
    {
        resultado = resultado && (digitalRead(S21) == g);
    }
    if (Activado_Estado(h))
    {
        resultado = resultado && (digitalRead(S30) == h);
    }
    return resultado;
}
```

Figura 1.23: Código Controlador prensado Parte 1

```
void Prensado_Controlador ()
{
  switch (EA_Prensado)
  {
    case 1:
      if (Prensado_Estado_Entradas(1, 1, 1, 0, 1, 0, 0, 1))
      {
        Prensado_Salidas (0, 0, 0, 1, 1, 1);
        EA_Prensado = 4;
      }
      else if (Prensado_Estado_Entradas(1, 1, 1, 0, 0, 0, 0, x, 1))
      {
        Prensado_Salidas (0, 0, 0, 1, 1, 1);
        EA_Prensado = 4;
      }
      else if (Prensado_Estado_Entradas(1, x, 1, 0, 1, 0, 1, 1))
      {
        Prensado_Salidas (0, 1, 0, 0, 0, 0);
        EA_Prensado = 5;
      }
      else if (Prensado_Estado_Entradas(1, x, 0, 1, 1, 0, 0, 1))
      {
        Prensado_Salidas (1, 0, 0, 0, 0, 0);
        EA_Prensado = 2;
      }
      break;
    case 2:
      if (Prensado_Estado_Entradas(1, 1, 1, 0, 1, 0, 0, x))
      {
        Prensado_Salidas (0, 0, 0, 1, 1, 1);
        EA_Prensado = 4;
      }
      else if (Prensado_Estado_Entradas(1, 0, 1, 0, 1, 0, 0, x))
      {
        Prensado_Salidas (0, 0, 0, 0, 0, 0);
        EA_Prensado = 1;
      }
      else if (Prensado_Estado_Entradas(0, x, 1, 0, 1, 0, 0, x))
      {
        Prensado_Salidas (0, 0, 0, 0, 0, 0);
        EA_Prensado = 1;
      }
      break;
  }
}
```

Figura 1.24: Código Controlador prensado Parte 2

```

case 3:
if (Prensado_Estado_Entradas(1, x, 1, 0, 1, 0, 1, x))
{
Prensado_Salidas (0, 1, 0, 0, 0, 0);
EA_Prensado = 5;
}
else if (Prensado_Estado_Entradas(0, x, 1, 0, 1, 0, 1, x))
{
Prensado_Salidas (0, 0, 0, 0, 0, 0);
EA_Prensado = 1;
}
break;
case 4:
if (Prensado_Estado_Entradas(1, x, 1, 0, 0, 1, 1, x))
{
Prensado_Salidas (0, 0, 1, 0, 0, 0);
EA_Prensado = 3;
}
else if (Prensado_Estado_Entradas(1, 0, 1, 0, 0, 0, x, x))
{
Prensado_Salidas (0, 0, 0, 0, 0, 0);
EA_Prensado = 1;
}
else if (Prensado_Estado_Entradas(0, 1, 1, 0, 0, 0, 0, x))
{
Prensado_Salidas (0, 0, 0, 0, 0, 0);
EA_Prensado = 1;
}
break;
case 5:
if (Prensado_Estado_Entradas(1, x, 0, 1, 1, 0, 0, x))
{
Prensado_Salidas (1, 0, 0, 0, 0, 0);
EA_Prensado = 2;
}
else if (Prensado_Estado_Entradas(0, x, 0, 1, 1, 0, 0, x))
{
Prensado_Salidas (0, 0, 0, 0, 0, 0);
EA_Prensado = 1;
}
break;
}
}

```

Figura 1.25: Código Controlador prensado Parte 3

### 1.1.5. Subsistema girado

Para los elementos de planta correspondientes al subsistema se generaron las Figura 1.26, Figura 1.27, Figura 1.16:

```
//Gira_Mecanismo

uint8_t Posicion_Gira_Mecanismo = 0;

#define Posicion_Gira_Mecanismo_Fin 180 //180° de giro
#define Posicion_Gira_Mecanismo_Inicio 0 // 0° de giro
// [ X° / Intervalo de tiempo] 36--> 1 seg en girar
#define Posicion_Gira_Mecanismo_Razon 36

void Gira_Mecanismo ()
{
  if (Posicion_Gira_Mecanismo >= Posicion_Gira_Mecanismo_Fin)
  {
    digitalWrite(S23, HIGH);
    digitalWrite(S22, LOW );
    Posicion_Gira_Mecanismo = Posicion_Gira_Mecanismo_Fin;
  }
  else if (Posicion_Gira_Mecanismo > Posicion_Gira_Mecanismo_Inicio)
  {
    digitalWrite(S23, LOW );
    digitalWrite(S22, LOW );
  }
  else
  {
    digitalWrite(S23, LOW );
    digitalWrite(S22, HIGH);
    Posicion_Gira_Mecanismo = Posicion_Gira_Mecanismo_Inicio;
  }
  //////////////////////////////////////
}
}
```

Figura 1.26: Código Mecanismo de girado

```
//Gira_Banda

#define Bits_Gira_Banda 15
// 15 bits * 200 ms = 3 seg
//Cadena... Usada para representar el desplazamiento de objetos en la Banda
int Cadena_Gira_Banda[Bits_Gira_Banda] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
```

Figura 1.27: Código Banda1 de girado

```
//Gira_Banda2

#define Bits_Gira_Banda2 5
// 5 bits * 200 ms = 1 seg
//Cadena... Usada para representar el desplazamiento de objetos en la Banda
int Cadena_Gira_Banda2[Bits_Gira_Banda2] = {0, 0, 0, 0, 0};

void Gira_Banda2()
{
  if (Cadena_Gira_Banda2[3])
  {
    digitalWrite(S24, HIGH);
  }
  else
  {
    digitalWrite(S24, LOW );
  }
  //////////////////////////////////////,
}
```

Figura 1.28: Código Banda2 de girado

Para el comportamiento de planta se obtuvo la Figura 1.29

```

boolean Activado_Planta = true;
#define Tiempo_Planta 200
uint32_t tiempo_Actual_Planta;
int64_t tiempo_Total_Planta;

void Planta ()
{
  if (Activado_Planta)
  {
    tiempo_Actual_Planta = millis();
    Activado_Planta = false;
  }
  tiempo_Total_Planta = millis() - tiempo_Actual_Planta;
  if (tiempo_Total_Planta >= Tiempo_Planta)
  {
    //////////////////////////////////////
    //////////////////////////////////////
    if (digitalRead(Gira_Izq))
    {
      Posicion_Gira_Mecanismo = Posicion_Gira_Mecanismo + Posicion_Gira_Mecanismo_Razon;
    }
    if (digitalRead(Gira_Der))
    {
      Posicion_Gira_Mecanismo = Posicion_Gira_Mecanismo - Posicion_Gira_Mecanismo_Razon;
    }
    //////////////////////////////////////
    if (digitalRead(Banda4_Der))
    {
      for (int i = 1; i <= Bits_Gira_Banda2; i++)
      {
        Cadena_Gira_Banda2[Bits_Gira_Banda2 - i] = Cadena_Gira_Banda2[Bits_Gira_Banda2 - 1 - i];
      }
      Cadena_Gira_Banda2[0] = 0;
    }
    if (digitalRead(Banda4_Izq))
    {
      if (Cadena_Gira_Banda2[0])
      {
        Cadena_Secado2_Banda[0] = 1;
      }
      for (int i = 1; i < Bits_Gira_Banda2; i++)
      {
        Cadena_Gira_Banda2[i - 1] = Cadena_Gira_Banda2[i];
      }
      Cadena_Gira_Banda2[4] = 0;
    }
    //////////////////////////////////////
    if (digitalRead(Banda5))
    {
      if (Cadena_Gira_Banda[Bits_Gira_Banda - 1])
      {
        Cadena_Gira_Banda2[0] = 1;
      }
      for (int i = 1; i <= Bits_Gira_Banda; i++)
      {
        Cadena_Gira_Banda[Bits_Gira_Banda - i] = Cadena_Gira_Banda[Bits_Gira_Banda - 1 - i];
      }
      Cadena_Gira_Banda[0] = 0;
    }
    //////////////////////////////////////
    //////////////////////////////////////
    tiempo_Actual_Planta = millis();
  }
}

```

Figura 1.29: Código Planta girado

Por último se genera la Figura 1.30, Figura 1.31, Figura 1.32 que corresponden al comportamiento del controlador y corresponde fielmente al diseñado en el modelo de máquina de estado.

```
//Definicion estado inicial del subproceso
int EA_Gira = 1; //Estado Actual

void Gira_Salidas(unsigned char a, unsigned char b, unsigned char c, unsigned char d,
                 unsigned char e)
{
    digitalWrite(Banda5, a);
    digitalWrite(Gira_Izq, b);
    digitalWrite(Gira_Der, c);
    digitalWrite(Banda4_Iqz, d);
    digitalWrite(Banda4_Der, e);
}

boolean Gira_Estado_Entradas (unsigned char a, unsigned char b, unsigned char c,
                              unsigned char d, unsigned char e, unsigned char f)
{
    boolean resultado = true;
    if (Activado_Estado(a))
    {
        resultado = resultado && (digitalRead(S0) == a);
    }
    if (Activado_Estado(b))
    {
        resultado = resultado && (digitalRead(S22) == b);
    }
    if (Activado_Estado(c))
    {
        resultado = resultado && (digitalRead(S23) == c);
    }
    if (Activado_Estado(d))
    {
        resultado = resultado && (digitalRead(S24) == d);
    }
    if (Activado_Estado(e))
    {
        resultado = resultado && (digitalRead(S25) == e);
    }
    if (Activado_Estado(f))
    {
        resultado = resultado && (digitalRead(S30) == f);
    }
    return resultado;
}
```

Figura 1.30: Código Controlador girado Parte 1

```
void Gira_Controlador ()
{
    switch (EA_Gira)
    {
        case 1:
            if (Gira_Estado_Entradas(1, 1, 0, 0, 0, 1))
            {
                Gira_Salidas(1, 0, 0, 0, 1);
                EA_Gira = 2;
            }
            else if (Gira_Estado_Entradas(1, 0, 0, 1, 0, 1))
            {
                Gira_Salidas(1, 1, 0, 0, 0);
                EA_Gira = 3;
            }
            else if (Gira_Estado_Entradas(1, 0, 1, x, x, 1))
            {
                Gira_Salidas(1, 0, 0, 1, 0);
                EA_Gira = 4;
            }
            else if (Gira_Estado_Entradas(1, 0, 0, 0, 1, 1))
            {
                Gira_Salidas(1, 0, 1, 0, 0);
                EA_Gira = 5;
            }
            break;
        case 2:
            if (Gira_Estado_Entradas(1, 1, 0, 1, 0, x))
            {
                Gira_Salidas(1, 1, 0, 0, 0);
                EA_Gira = 3;
            }
            else if (Gira_Estado_Entradas(0, 1, 0, 0, 0, x))
            {
                Gira_Salidas(0, 0, 0, 0, 0);
                EA_Gira = 1;
            }
            break;
    }
}
```

Figura 1.31: Código Controlador girado Parte 2



```

case 3:
    if (Gira_Estado_Entradas(1, 0, 1, 1, 0, x))
    {
        Gira_Salidas(1, 0, 0, 1, 0);
        EA_Gira = 4;
    }
    else if (Gira_Estado_Entradas(0, 0, 0, 1, 0, x))
    {
        Gira_Salidas(0, 0, 0, 0, 0);
        EA_Gira = 1;
    }
    break;
case 4:
    if (Gira_Estado_Entradas(1, 0, 1, 0, 1, x))
    {
        Gira_Salidas(1, 0, 1, 0, 0);
        EA_Gira = 5;
    }
    else if (Gira_Estado_Entradas(0, 0, 1, x, x, x))
    {
        Gira_Salidas(0, 0, 0, 0, 0);
        EA_Gira = 1;
    }
    break;
case 5:
    if (Gira_Estado_Entradas(1, 1, 0, 0, 0, x))
    {
        Gira_Salidas(1, 0, 0, 0, 1);
        EA_Gira = 2;
    }
    else if (Gira_Estado_Entradas(0, 0, 0, 0, 1, x))
    {
        Gira_Salidas(0, 0, 0, 0, 0);
        EA_Gira = 1;
    }
    break;
}
}

```

**Figura 1.32:** Código Controlador girado Parte 3



```

//Planta

boolean Activado_Planta = true;
#define Tiempo_Planta 200
uint32_t tiempo_Actual_Planta;
int64_t tiempo_Total_Planta;

void Planta ()
{
  if (Activado_Planta)
  {
    tiempo_Actual_Planta = millis();
    Activado_Planta = false;
  }
  tiempo_Total_Planta = millis() - tiempo_Actual_Planta;
  if (tiempo_Total_Planta >= Tiempo_Planta)
  {
    //////////////////////////////////////
    //////////////////////////////////////
    if (digitalRead(Banda6))
    {
      if (Cadena_Secado2_Banda[Bits_Secado2_Banda - 1])
      {
        Cadena_Impresion_Banda[0] = 1;
      }
      for (int i = 1; i <= Bits_Secado2_Banda; i++)
      {
        Cadena_Secado2_Banda[Bits_Secado2_Banda - i] = Cadena_Secado2_Banda[Bits_Secado2_Banda - 1 - i];
      }
      Cadena_Secado2_Banda[0] = 0;
    }
    //////////////////////////////////////
    //////////////////////////////////////
    tiempo_Actual_Planta = millis();
  }
}

```

Figura 1.34: Código Planta secado por luz UV

Por último se genera la Figura 1.35, Figura 1.36 que corresponden al comportamiento del controlador y corresponde fielmente al diseñado en el modelo de máquina de estado.

```
//Definicion estado inicial del subproceso
int EA_Secado2 = 1; //Estado Actual

void Secado2_Salidas(unsigned char a, unsigned char b, unsigned char c,
                    unsigned char d)
{
    digitalWrite(Lampara, a);
    digitalWrite(Banda6, b);
    digitalWrite(Bomba3, c);
    digitalWrite(Valvula3, d);
}

boolean Secado2_Estado_Entradas (unsigned char a, unsigned char b,
                                unsigned char c, unsigned char d)
{
    boolean resultado = true;
    if (Activado_Estado(a))
    {
        resultado = resultado && (digitalRead(S0) == a);
    }
    if (Activado_Estado(b))
    {
        resultado = resultado && (digitalRead(S25) == b);
    }
    if (Activado_Estado(c))
    {
        resultado = resultado && (digitalRead(S26) == c);
    }
    if (Activado_Estado(d))
    {
        resultado = resultado && (digitalRead(S30) == d);
    }
    return resultado;
}
```

Figura 1.35: Código Controlador secado por luz UV Parte 1

```

void Secado2_Controlador ()
{
  switch (EA_Secado2)
  {
    case 1:
      if (Secado2_Estado_Entradas(1, 0, 0, 1))
      {
        Secado2_Salidas(0, 1, 1, 1);
        EA_Secado2 = 2;
      }
      else if (Secado2_Estado_Entradas(1, 0, 1, 1))
      {
        Secado2_Salidas(0, 1, 1, 1);
        EA_Secado2 = 2;
      }
      else if (Secado2_Estado_Entradas(1, 1, 0, 1))
      {
        Secado2_Salidas(1, 1, 1, 1);
        EA_Secado2 = 3;
      }
      else if (Secado2_Estado_Entradas(1, 1, 1, 1))
      {
        Secado2_Salidas(1, 1, 1, 1);
        EA_Secado2 = 3;
      }
      break;
    case 2:
      if (Secado2_Estado_Entradas(1, 1, 0, 1))
      {
        Secado2_Salidas(1, 1, 1, 1);
        EA_Secado2 = 3;
      }
      else if (Secado2_Estado_Entradas(0, 0, 0, 1))
      {
        Secado2_Salidas(0, 0, 0, 0);
        EA_Secado2 = 1;
      }
      break;
    case 3:
      if (Secado2_Estado_Entradas(1, 0, 1, 1))
      {
        Secado2_Salidas(0, 1, 1, 1);
        EA_Secado2 = 2;
      }
      else if (Secado2_Estado_Entradas(0, 1, 0, 1))
      {
        Secado2_Salidas(0, 0, 0, 0);
        EA_Secado2 = 1;
      }
      else if (Secado2_Estado_Entradas(0, 1, 1, 1))
      {
        Secado2_Salidas(0, 0, 0, 0);
        EA_Secado2 = 1;
      }
      else if (Secado2_Estado_Entradas(0, 0, 1, 1))
      {
        Secado2_Salidas(0, 0, 0, 0);
        EA_Secado2 = 1;
      }
      break;
  }
}

```

Figura 1.36: Código Controlador secado por luz UV Parte 2



```

//Planta

boolean Activado_Planta = true;
#define Tiempo_Planta 200
uint32_t tiempo_Actual_Planta;
int64_t tiempo_Total_Planta;

void Planta ()
{
  if (Activado_Planta)
  {
    tiempo_Actual_Planta = millis();
    Activado_Planta = false;
  }
  tiempo_Total_Planta = millis() - tiempo_Actual_Planta;
  if (tiempo_Total_Planta >= Tiempo_Planta)
  {
    //////////////////////////////////////
    if (digitalRead(Banda7))
    {
      if (Cadena_Impresion_Banda[Bits_Impresion_Banda - 1])
      {
        Cadena_Horneado_Banda[0] = 1;
      }
      for (int i = 1; i <= Bits_Impresion_Banda; i++)
      {
        Cadena_Impresion_Banda[Bits_Impresion_Banda - i] = Cadena_Impresion_Banda[Bits_Impresion_Banda - 1 - i];
      }
      Cadena_Impresion_Banda[0] = 0;
    }
    //////////////////////////////////////
    tiempo_Actual_Planta = millis();
  }
}

```

Figura 1.38: Código Planta impresión

Por último se genera la Figura 1.39, Figura 1.40 que corresponden al comportamiento del controlador y corresponde fielmente al diseñado en el modelo de máquina de estado.

```
//Definicion estado inicial del subprocesso
int EA_Impresion = 1; //Estado Actual

void Impresion_Salidas (unsigned char a, unsigned char b, unsigned char c, unsigned char d)
{
    digitalWrite(Banda7, a);
    digitalWrite(Impresora, b);
    digitalWrite(Valvula4, c);
    digitalWrite(Bomba4, d);
}

boolean Impresion_Estado_Entradas (unsigned char a, unsigned char b, unsigned char c, unsigned char d)
{
    boolean resultado = true;
    if (Activado_Estado(a))
    {
        resultado = resultado && (digitalRead(S0) == a);
    }
    if (Activado_Estado(b))
    {
        resultado = resultado && (digitalRead(S27) == b);
    }
    if (Activado_Estado(c))
    {
        resultado = resultado && (digitalRead(S28) == c);
    }
    if (Activado_Estado(d))
    {
        resultado = resultado && (digitalRead(S30) == d);
    }
    return resultado;
}
```

Figura 1.39: Código Controlador impresión Parte 1



```

void Impresion_Controlador ()
{
    switch (EA_Impresion)
    {
        case 1:
            if (Impresion_Estado_Entradas(1, 0, 0, 1))
            {
                Impresion_Salidas(1, 0, 1, 1);
                EA_Impresion = 2;
            }
            else if (Impresion_Estado_Entradas(1, 0, 1, 1))
            {
                Impresion_Salidas(1, 0, 1, 1);
                EA_Impresion = 2;
            }
            else if (Impresion_Estado_Entradas(1, 1, 0, 1))
            {
                Impresion_Salidas(1, 1, 1, 1);
                EA_Impresion = 3;
            }
            else if (Impresion_Estado_Entradas(1, 1, 1, 1))
            {
                Impresion_Salidas(1, 1, 1, 1);
                EA_Impresion = 3;
            }
            break;
        case 2:
            if (Impresion_Estado_Entradas(1, 1, 0, 1))
            {
                Impresion_Salidas(1, 1, 1, 1);
                EA_Impresion = 3;
            }
            else if (Impresion_Estado_Entradas(0, 0, 0, 1))
            {
                Impresion_Salidas(0, 0, 0, 0);
                EA_Impresion = 1;
            }
            break;
        case 3:
            if (Impresion_Estado_Entradas(1, 0, 1, 1))
            {
                Impresion_Salidas(1, 0, 1, 1);
                EA_Impresion = 2;
            }
            else if (Impresion_Estado_Entradas(0, 1, 0, 1))
            {
                Impresion_Salidas(0, 0, 0, 0);
                EA_Impresion = 1;
            }
            else if (Impresion_Estado_Entradas(0, 1, 1, 1))
            {
                Impresion_Salidas(0, 0, 0, 0);
                EA_Impresion = 1;
            }
            else if (Impresion_Estado_Entradas(0, 0, 1, 1))
            {
                Impresion_Salidas(0, 0, 0, 0);
                EA_Impresion = 1;
            }
            break;
    }
}

```

Figura 1.40: Código Controlador impresión Parte 2



```
//Horneado_Temperatura

int16_t Valor_Horneado_Temperatura = 25;

#define Maxima_Horneado_Temperatura 1200
#define Minima_Horneado_Temperatura 1150
#define Ambiente_Horneado_Temperatura 25

#define Razon_Horneado_Temperatura_Incremento 50
#define Razon_Horneado_Temperatura_Decremento 15

void Horneado_Temperatura ()
{
  if (Valor_Horneado_Temperatura >= Maxima_Horneado_Temperatura)
  {
    Valor_Horneado_Temperatura = Maxima_Horneado_Temperatura;
  }
  if (Valor_Horneado_Temperatura >= Minima_Horneado_Temperatura)
  {
    digitalWrite(S30, HIGH );
  }
  else
  {
    digitalWrite(S30, LOW );
  }
  if (Valor_Horneado_Temperatura <= Ambiente_Horneado_Temperatura)
  {
    Valor_Horneado_Temperatura = Ambiente_Horneado_Temperatura;
  }
}
```

Figura 1.42: Código Temperatura de horneado

Para el comportamiento de planta se obtuvo la Figura 1.43

```

//Planta

boolean Activado_Planta = true;
#define Tiempo_Planta 200
uint32_t tiempo_Actual_Planta;
int64_t tiempo_Total_Planta;

void Planta ()
{
  if (Activado_Planta)
  {
    tiempo_Actual_Planta = millis();
    Activado_Planta = false;
  }
  tiempo_Total_Planta = millis() - tiempo_Actual_Planta;
  if (tiempo_Total_Planta >= Tiempo_Planta)
  {
    //////////////////////////////////////
    //////////////////////////////////////
    if (digitalRead(Quemador2))
    {
      Valor_Horneado_Temperatura = Valor_Horneado_Temperatura + Razon_Horneado_Temperatura_Incremento;
    }
    else
    {
      Valor_Horneado_Temperatura = Valor_Horneado_Temperatura - Razon_Horneado_Temperatura_Decremento;
    }
    if (digitalRead(Quemador3))
    {
      Valor_Horneado_Temperatura = Valor_Horneado_Temperatura + Razon_Horneado_Temperatura_Incremento;
    }
    //////////////////////////////////////
    if (digitalRead(Banda8))
    {
      for (int i = 1; i <= Bits_Horneado_Banda; i++)
      {
        Cadena_Horneado_Banda[Bits_Horneado_Banda - i] = Cadena_Horneado_Banda[Bits_Horneado_Banda - 1 - i];
      }
      Cadena_Horneado_Banda[0] = 0;
    }
    //////////////////////////////////////
    //////////////////////////////////////
    tiempo_Actual_Planta = millis();
  }
}

```

Figura 1.43: Código Planta horneado

Por último se genera la Figura 1.44, Figura 1.45 que corresponden al comportamiento del controlador y corresponde fielmente al diseñado en el modelo de máquina de estado.

```

//Definicion estado incial del subprocesso
int EA_Horneado = 1; //Estado Actual

void Horneado_Salidas(unsigned char a, unsigned char b, unsigned char c)
{
    digitalWrite(Banda8, a);
    digitalWrite(Quemador2, b);
    digitalWrite(Quemador3, c);
}

boolean Horneado_Estado_Entradas (unsigned char a, unsigned char b, unsigned char c)
{
    boolean resultado = true;
    if (Activado_Estado(a))
    {
        resultado = resultado && (digitalRead(S0) == a);
    }
    if (Activado_Estado(b))
    {
        resultado = resultado && (digitalRead(S29) == b);
    }
    if (Activado_Estado(c))
    {
        resultado = resultado && (digitalRead(S30) == c);
    }
    return resultado;
}

```

Figura 1.44: Código Controlador horneado Parte 1

```
void Horneado_Controlador()
{
    switch (EA_Horneado)
    {
        case 1:
            if (Horneado_Estado_Entradas(1, 0, 0))
            {
                Horneado_Salidas(0, 1, 0);
                EA_Horneado = 4;
            }
            else if (Horneado_Estado_Entradas(1, 1, 0))
            {
                Horneado_Salidas(0, 1, 1);
                EA_Horneado = 2;
            }
            break;
        case 2:
            if (Horneado_Estado_Entradas(1, x, 1))
            {
                Horneado_Salidas(1, 1, 0);
                EA_Horneado = 3;
            }
            else if (Horneado_Estado_Entradas(0, x, x))
            {
                Horneado_Salidas(0, 0, 0);
                EA_Horneado = 1;
            }
            break;
        case 3:
            if (Horneado_Estado_Entradas(0, x, x))
            {
                Horneado_Salidas(0, 0, 0);
                EA_Horneado = 1;
            }
            break;
        case 4:
            if (Horneado_Estado_Entradas(1, x, 1))
            {
                Horneado_Salidas(1, 1, 0);
                EA_Horneado = 3;
            }
            else if (Horneado_Estado_Entradas(0, x, x))
            {
                Horneado_Salidas(0, 0, 0);
                EA_Horneado = 1;
            }
            break;
    }
}
```

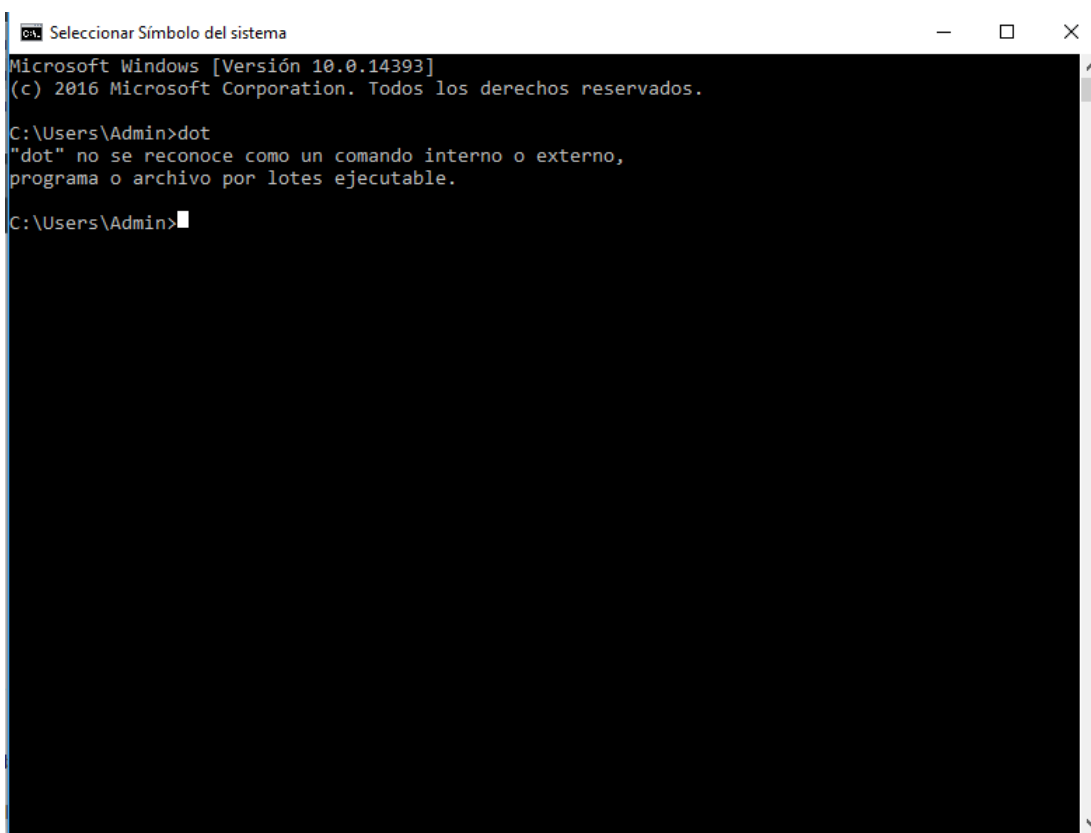
Figura 1.45: Código Controlador horneado Parte 2

## 2 Anexo B - Integración herramienta Graphviz

### 2.1. Comprobar Integración CMD comando dot

Oprimir las teclas combinadas «Windows + E» se abre una ventana de «Ejecutar» escriben «cmd» y da click en aceptar.

Se abre una ventana de Símbolos del sistema «CMD» en esta ventana escribir el comando «dot».



```
Selecionar Símbolo del sistema
Microsoft Windows [Versión 10.0.14393]
(c) 2016 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Admin>dot
"dot" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\Admin>
```

Figura 2.1: Ventana CMD comando dot

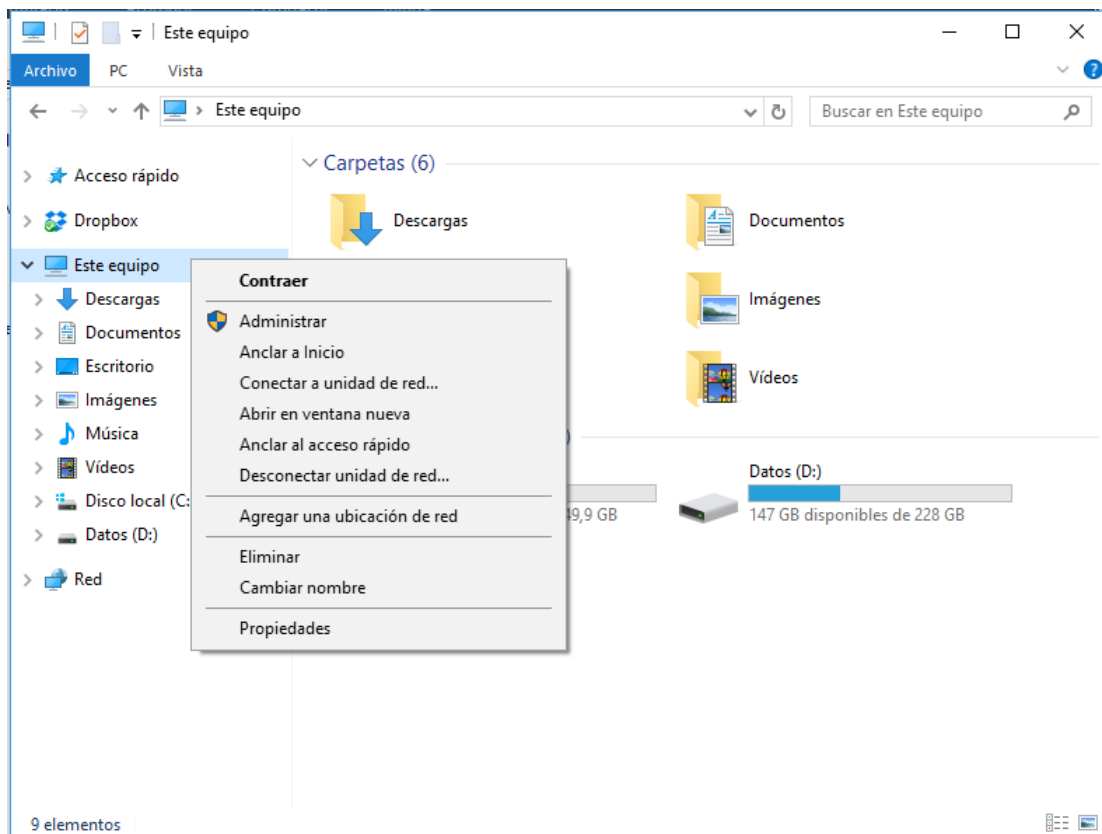
Si sale el mensaje diciendo que «dot» no se reconoce como un comando ..., entonces se debe realizar en proceso de integración CMD.

En caso de no dar dicho mensaje, ya sea que se quede pensando o que diga ingrese parámetro ... etc, significa que durante la instalación si se integro al Símbolo del sistema y debe ser ejecutado la combinación de teclas « Ctrl + C» que indica terminación de proceso, cerrando el proceso dot abierto.

Cerrar la ventana «Símbolo del sistema CMD» y no realizar el proceso de Integración.

## 2.2. Integración CMD en Windows 10, 8 y 7

Abrir cualquier Carpeta o directorio, hacer click derecho sobre «Este Equipo» y desplegar el menú de opciones haciendo click en la opción «Propiedades».



**Figura 2.2:** Abrir Propiedades del Sistema

En la ventana de propiedades del sistema dirigirse al menú de opciones avanzadas encontrado a la izquierda de la ventana y dar click en donde dice «Configuración



avanzada del sistema».

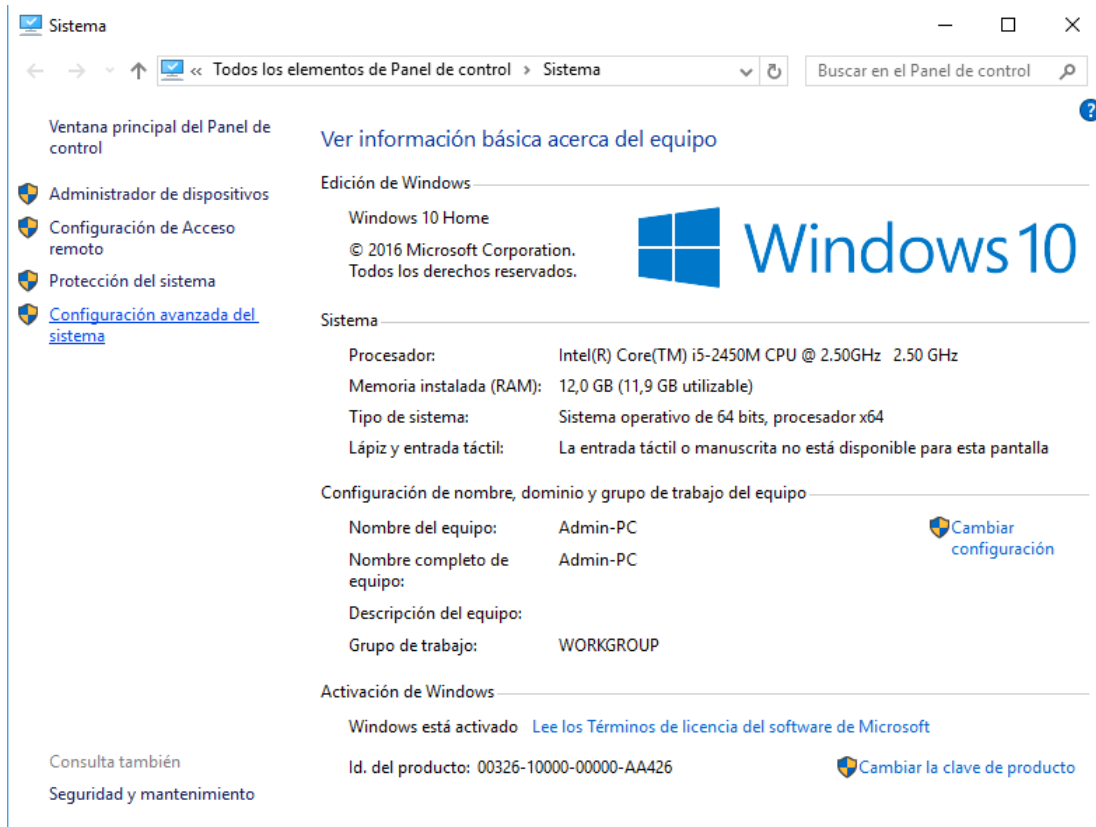
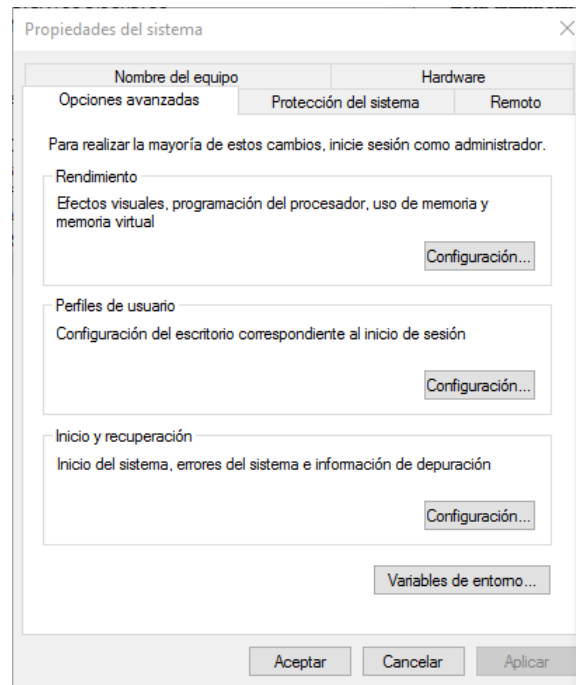


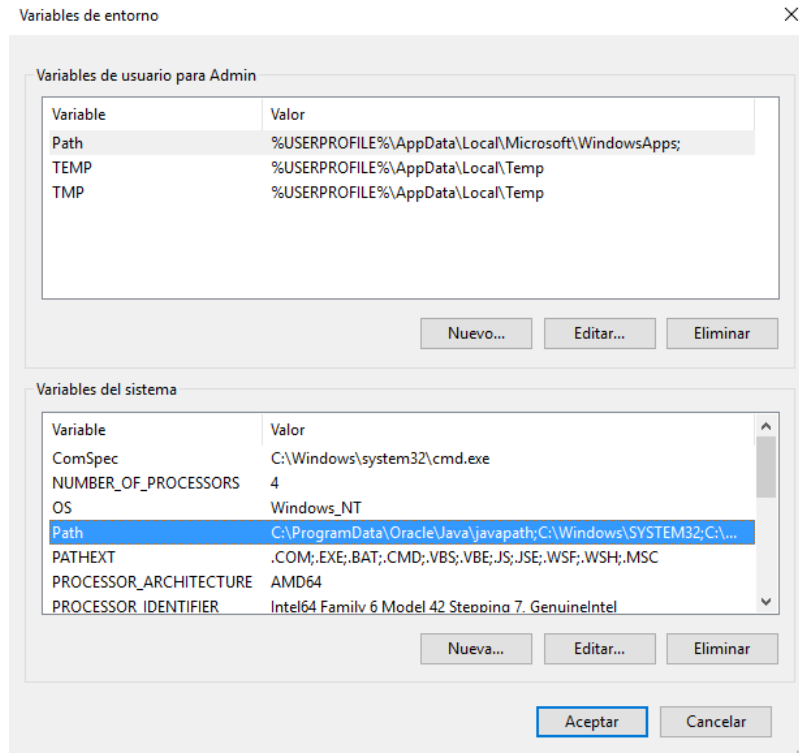
Figura 2.3: Ventana Propiedades del Sistema

En la ventana que se abre ubicarse en la pestaña «Opciones avanzadas» y dar click en el botón «Variables de entorno».



**Figura 2.4:** Ventana Opciones Avanzadas

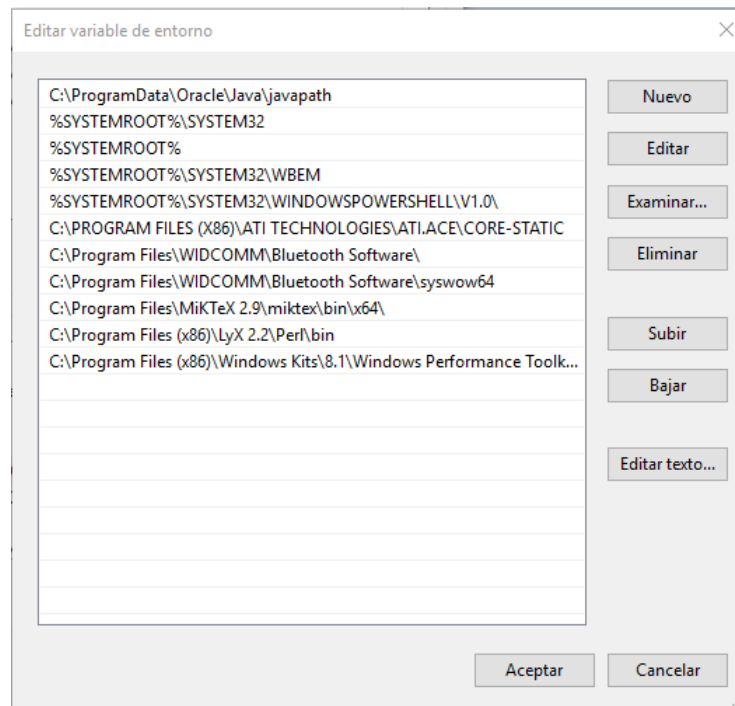
Ubicarse en el cuadro «Variables del sistema», buscar un elemento que diga «PATH» seleccionarlo y dar click en el botón «Editar».



**Figura 2.5:** Ventana Variables de Entorno

Obtener la ruta de sistema donde se ha instalado el programa Graphviz para el caso es «C:\Program Files (x86)\Graphviz2.38\bin».

En la ventana Abierta hacer click en «Nuevo» y se coloca la ruta de instalacion del programa Graphviz.



**Figura 2.6:** Ventana Editar Variables de Entorno

Le da en aceptar a todas ventanas abiertas hasta cerrar «Propiedades del sistema»

Ya esta lista la integración con símbolo del sistema CMD

## **3 Anexo C - Modelos Identificados para el caso de estudio**

### **3.1. Autómatas Con base al método de Klein:**

Los autómatas obtenidos mediante el método de Klein, están constituidos en dos partes, una primera son los conjuntos de estados y transiciones identificados, en el que se listan por cada línea de texto el nombre de un estado seguido del marcado asociado y este seguido de los nombres de los estados posteriores a los que se conecta mediante transiciones.

### 3.1.1. Subsistema suministro de materias primas

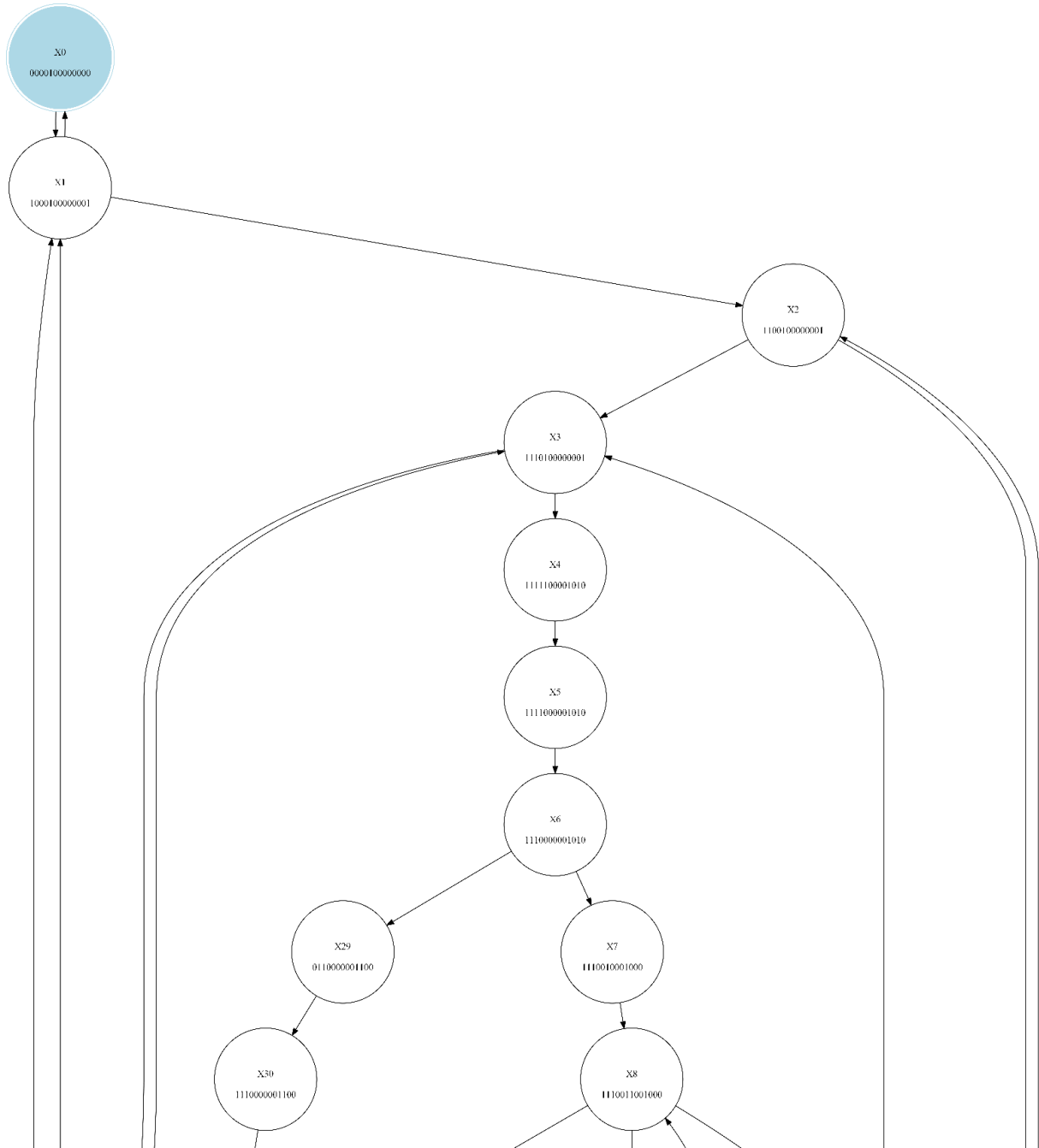
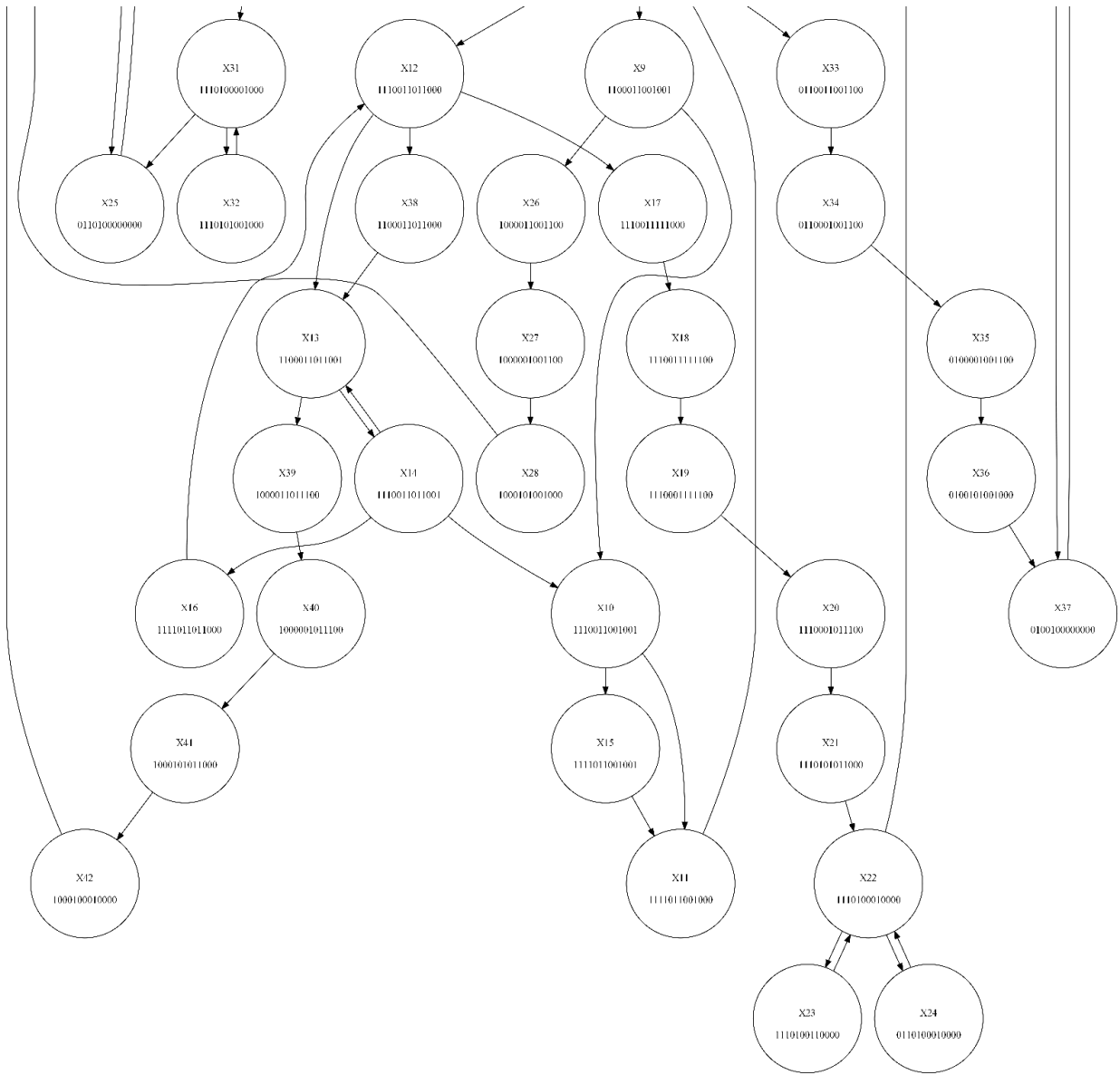


Figura 3.1: Autómata identificado subsistema suministro Parte 1



**Figura 3.2:** Autómata identificado subsistema suministro Parte 2

En la Figura 3.1, Figura 3.2 se aprecia el modelo del autómata identificado del subsistema suministro de materia prima, el cual describe una estructura simple de estados donde se representa solamente los ciclos encontrados en el sistema tomando como estado inicial el estado marcado en color azul y el marcado de los estados equivale solo al vector sin procesar de las observaciones únicas encontradas. La identificación se realizó en tres partes, la primera parte de generación de secuencias válidas encontrando 3 secuencias tardando 19,0129 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 6,5045 mili-segundos, obteniendo 43 estados.

```

X0,0000100000000,X1
X1,1000100000001,X2,X0
X2,1100100000001,X3,X37
X3,1110100000001,X4,X25
X4,1111100001010,X5
X5,1111000001010,X6
X6,1110000001010,X7,X29
X7,1110010001000,X8
X8,1110011001000,X9,X12,X33
X9,1100011001001,X10,X26
X10,1110011001001,X11,X15
X11,1111011001000,X8
X12,1110011011000,X13,X17,X38
X13,1100011011001,X14,X39
X14,1110011011001,X10,X16,X13
X15,1111011001001,X11
X16,1111011011000,X12
X17,1110011111000,X18
X18,1110011111100,X19
X19,1110001111100,X20
X20,1110001011100,X21
X21,1110101011000,X22
X22,1110100010000,X23,X24,X3
X23,1110100110000,X22
X24,0110100010000,X22
X25,0110100000000,X3
X26,1000011001100,X27
X27,1000001001100,X28
X28,1000101001000,X1
X29,0110000001100,X30
X30,1110000001100,X31
X31,1110100001000,X32,X25
X32,1110101001000,X31
X33,0110011001100,X34
X34,0110001001100,X35
X35,0100001001100,X36
X36,0100101001000,X37
X37,0100100000000,X2
X38,1100011011000,X13
X39,1000011011100,X40
X40,1000001011100,X41
X41,1000101011000,X42
X42,1000100010000,X1

```

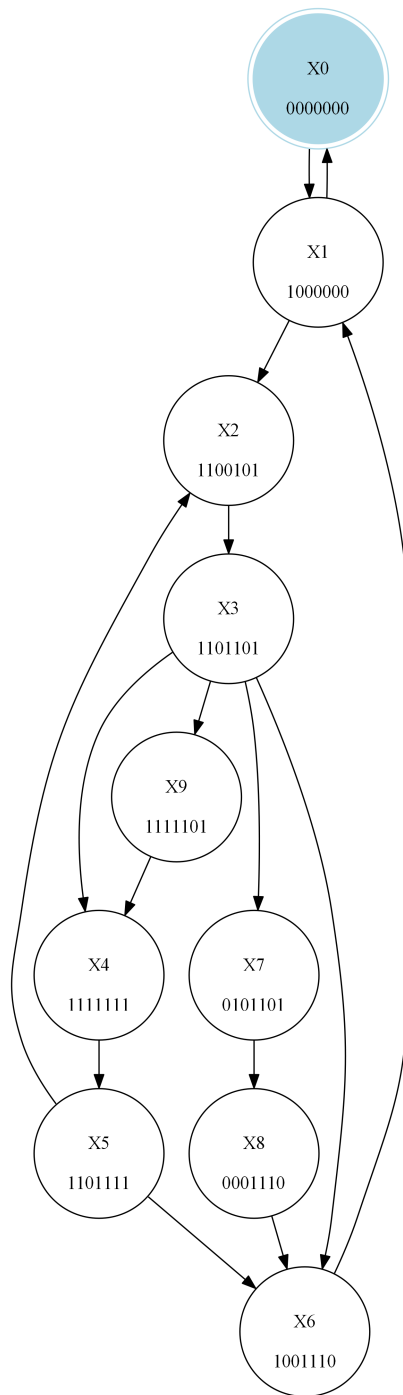
**Tabla 3.1:** Código Autómata subsistema suministro



---

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_1\_Suministro.aut» observable en la Tabla 3.1 en este se guardan los nombres de cada estado seguidos del marcado asociado que es la observación sin repetición seguidas del conjunto de nombres de los estados con los que se conecta mediante transiciones. El guardado de la identificación se tardó 24,5157 mili-segundos, también en esta parte se genera la gráfica automática del autómata identificado, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz.

**3.1.2. Subsistema molienda de material**



**Figura 3.3:** Autómata identificado subsistema molino

---

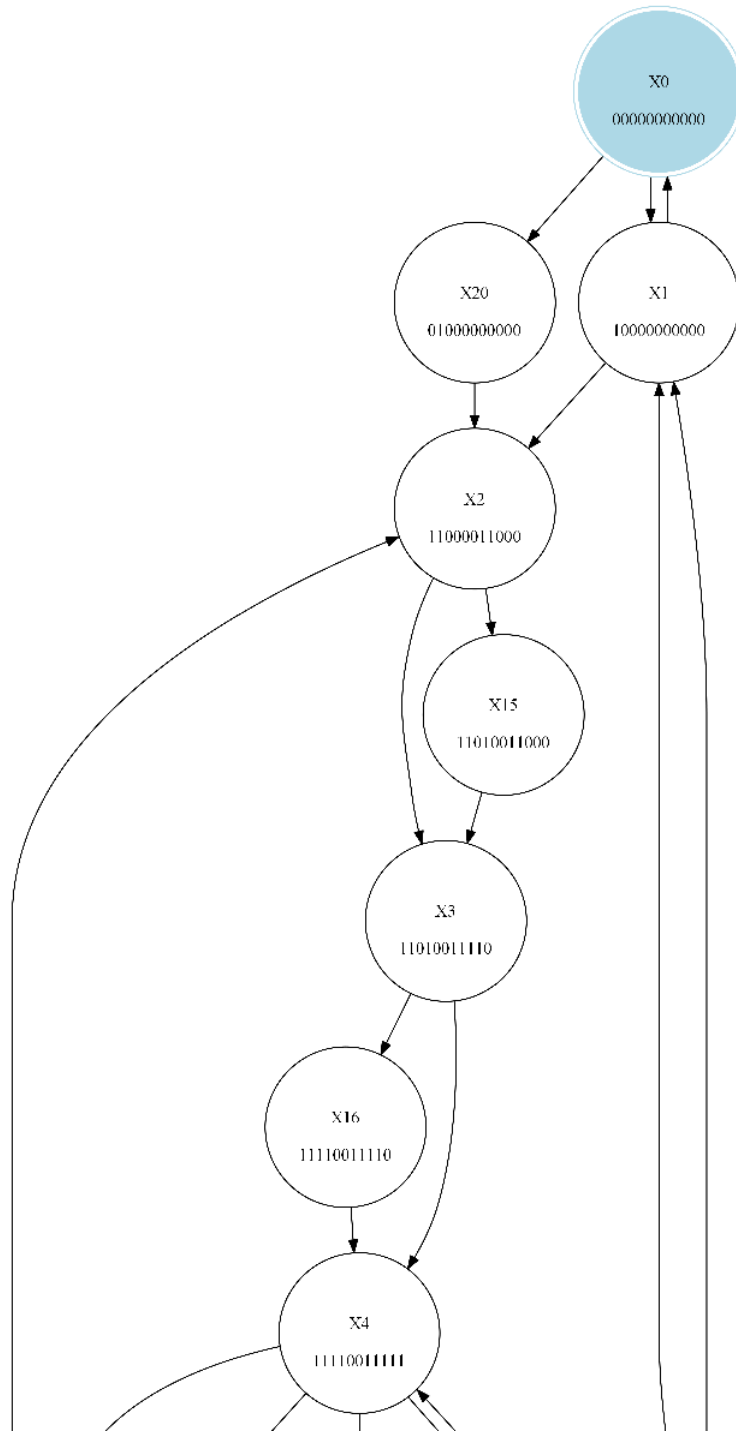
En la Figura 3.3 se aprecia el modelo del autómata identificado del subsistema molino, el cual describe una estructura simple de estados donde se representa solamente los ciclos encontrados en el sistema tomando como estado inicial el estado marcado en color azul y el marcado de los estados equivale solo al vector sin procesar de las observaciones únicas encontradas. La identificación se realizó en tres partes, la primera parte de generación de secuencias válidas encontrando 28 secuencias tardando 35,5126 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 5,0024 mili-segundos, obteniendo 10 estados.

X0,0000000,X1
X1,1000000,X2,X0
X2,1100101,X3
X3,1101101,X4,X6,X7,X9
X4,1111111,X5
X5,1101111,X2,X6
X6,1001110,X1
X7,0101101,X8
X8,0001110,X6
X9,1111101,X4

**Tabla 3.2:** Código Autómata subsistema molino

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_2\_Molino.aut» observable en la Tabla 3.2 en este se guardan los nombres de cada estado seguidos del marcado asociado que es la observación sin repetición seguidas del conjunto de nombres de los estados con los que se conecta mediante transiciones. El guardado de la identificación se tardó 27,0183 mili-segundos, también en esta parte se genera la gráfica automática del autómata identificado, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz .

**3.1.3. Subsistema atomizado y secado**



**Figura 3.4:** Autómata identificado subsistema atomizado y secado Parte 1

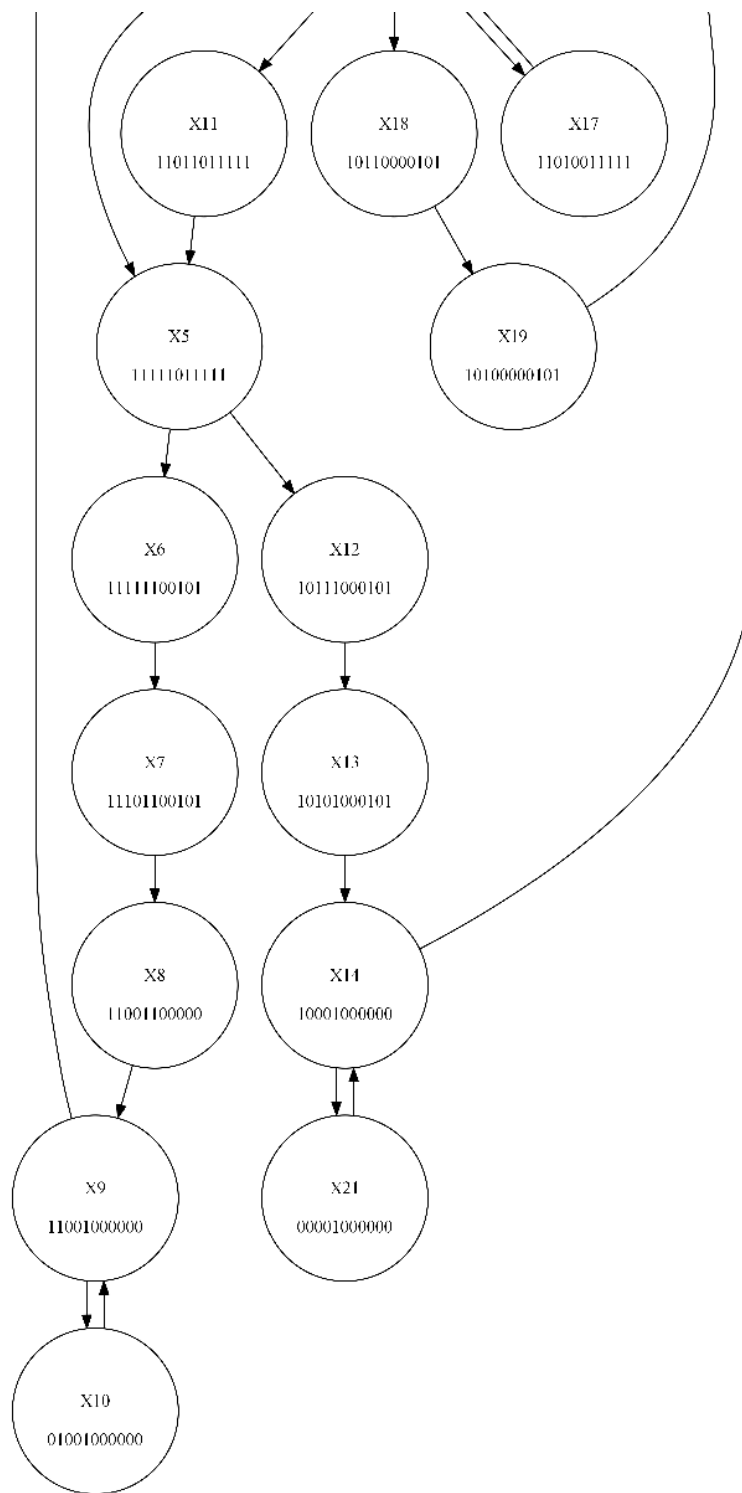


Figura 3.5: Autómata identificado subsistema atomizado y secado Parte 2

En la Figura 3.4, Figura 3.5 se aprecia el modelo del autómata identificado del subsistema atomizado y secado de materia prima, el cual describe una estructura simple de estados donde se representa solamente los ciclos encontrados en el sistema tomando como estado inicial el estado marcado en color azul y el marcado de los estados equivale solo al vector sin procesar de las observaciones únicas encontradas. La identificación se realizó en tres partes, la primera parte de generación de secuencias válidas encontrando 21 secuencias tardando 32,0206 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 7,0053 mili-segundos, obteniendo 22 estados.

```
X0,00000000000,X1,X20
X1,10000000000,X2,X0
X2,11000011000,X3,X15
X3,11010011110,X4,X16
X4,11110011111,X5,X11,X17,X18
X5,11111011111,X6,X12
X6,11111100101,X7
X7,11101100101,X8
X8,11001100000,X9
X9,11001000000,X10,X2
X10,01001000000,X9
X11,11011011111,X5
X12,10111000101,X13
X13,10101000101,X14
X14,10001000000,X1,X21
X15,11010011000,X3
X16,11110011110,X4
X17,11010011111,X4
X18,10110000101,X19
X19,10100000101,X1
X20,01000000000,X2
X21,00001000000,X14
```

**Tabla 3.3:** Código Autómata subsistema atomizado y secado

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_3\_Secado1.aut» observable en la Tabla 3.3 en este se guardan los nombres de cada estado seguidos del marcado asociado que es la observación sin repetición seguidas del conjunto de nombres de los estados con los que se conecta mediante transiciones. El guardado de la identificación se tardó 21,0134 mili-segundos, también en esta parte se genera la gráfica automática del autómata identificado, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz.

### 3.1.4. Subsistema prensado

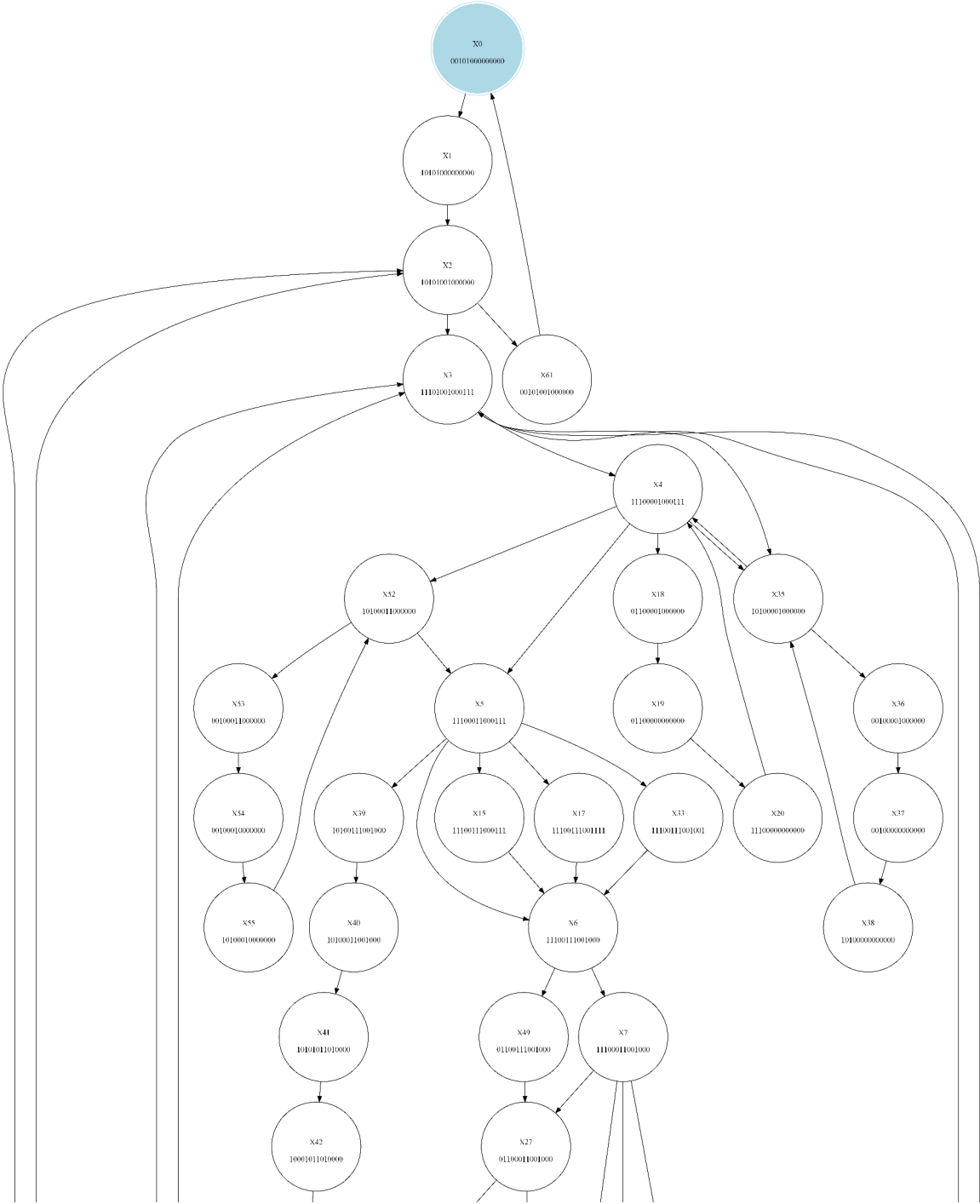
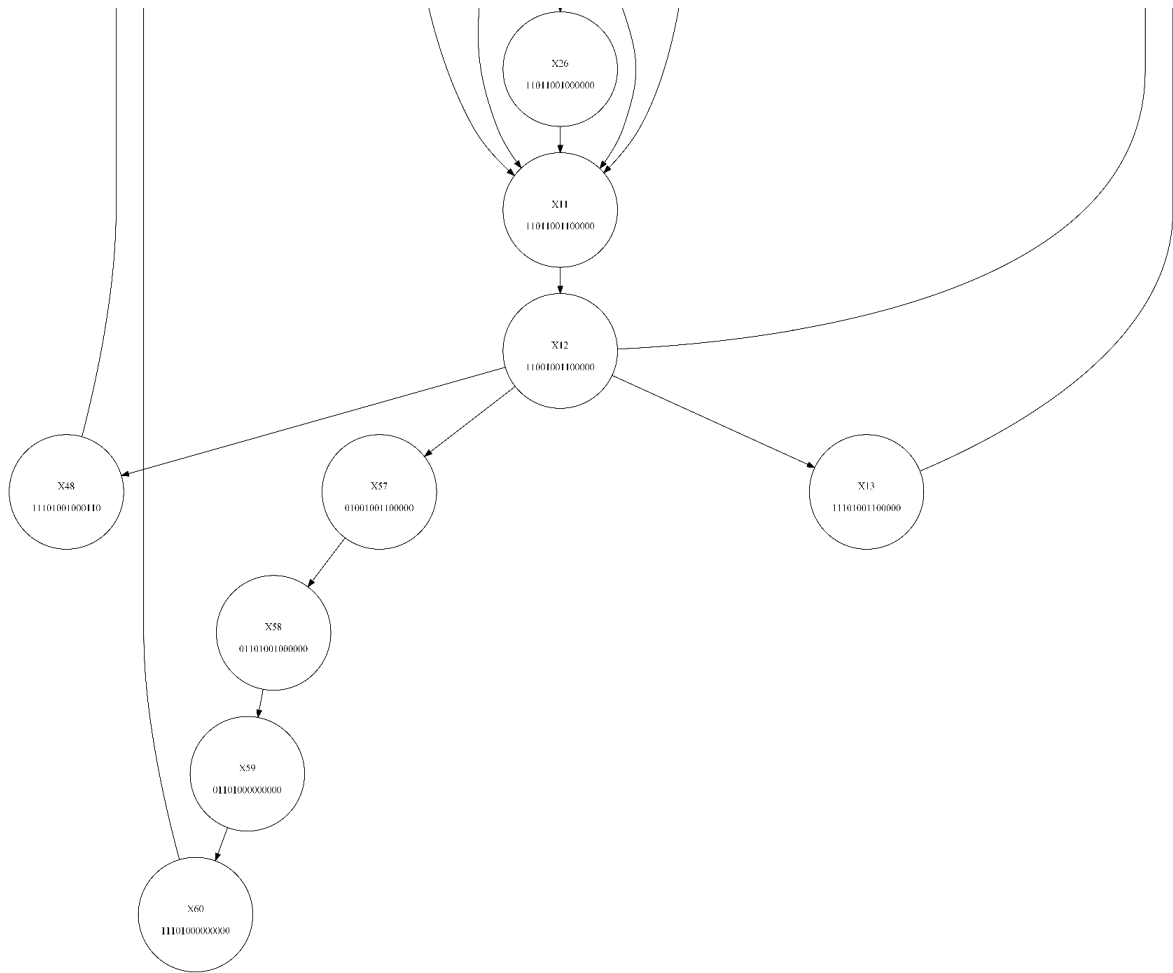


Figura 3.6: Autómata identificado subsistema prensado Parte 1







**Figura 3.8:** Autómata identificado subsistema prensado Parte 3

En la Figura 3.6, Figura 3.7, Figura 3.8 se aprecia el modelo del autómata identificado del subsistema prensado, el cual describe una estructura simple de estados donde se representa solamente los ciclos encontrados en el sistema tomando como estado inicial el estado marcado en color azul y el marcado de los estados equivale solo al vector sin procesar de las observaciones únicas encontradas. La identificación se realizó en tres partes, la primera parte de generación de secuencias válidas encontrando 1 secuencias tardando 15,6273 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 31,2328 mili-segundos, obteniendo 62 estados.

```

X0,00101000000000,X1
X1,10101000000000,X2
X2,10101001000000,X3,X61
X3,11101001000111,X4,X35
X4,11100001000111,X5,X18,X35,X52
X5,11100011000111,X6,X15,X17,X33,X39
X6,11100111001000,X7,X49
X7,11100011001000,X8,X14,X27,X34
X8,11101011010000,X9
X9,11001011010000,X10,X21
X10,11001001010000,X11,X16,X50,X56
X11,11011001100000,X12
X12,11001001100000,X3,X13,X48,X57
X13,11101001100000,X3
X14,11101011001000,X8
X15,11100111000111,X6
X16,11011001010000,X11
X17,11100111001111,X6
X18,01100001000000,X19
X19,01100000000000,X20
X20,11100000000000,X4
X21,01001011010000,X22
X22,01001010010000,X23
X23,01001000010000,X24
X24,01011000000000,X25
X25,11011000000000,X26,X11
X26,11011001000000,X11
X27,01100011001000,X28,X32
X28,01100010001000,X29
X29,01101010000000,X30
X30,11101010000000,X31,X8
X31,11101011000000,X8
X32,01101011000000,X29
X33,11100111001001,X6
X34,11101011011000,X8
X35,10100001000000,X36,X4
X36,00100001000000,X37
X37,00100000000000,X38
X38,10100000000000,X35
X39,10100111001000,X40
X40,10100011001000,X41
X41,10101011010000,X42
X42,10001011010000,X43

```

**Tabla 3.4:** Código Autómata subsistema prensado Parte 1

X43,10001001010000,X44,X45
X44,10011001010000,X45
X45,10011001100000,X46
X46,10001001100000,X47,X2
X47,10101001100000,X2
X48,11101001000110,X3
X49,01100111001000,X27
X50,01001001010000,X51
X51,01011001000000,X24
X52,10100011000000,X53,X5
X53,00100011000000,X54
X54,00100010000000,X55
X55,10100010000000,X52
X56,11011001110000,X11
X57,01001001100000,X58
X58,01101001000000,X59
X59,01101000000000,X60
X60,11101000000000,X3
X61,00101001000000,X0

**Tabla 3.5:** Código Autómata subsistema prensado Parte 2

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_4\_Prensado.aut» observable en la Tabla 3.4,Tabla 3.5 en este se guardan los nombres de cada estado seguidos del marcado asociado que es la observación sin repetición seguidas del conjunto de nombres de los estados con los que se conecta mediante transiciones. El guardado de la identificación se tardó 93,7394 mili-segundos, también en esta parte se genera la gráfica automática del autómata identificado, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz.

### 3.1.5. Subsistema girado

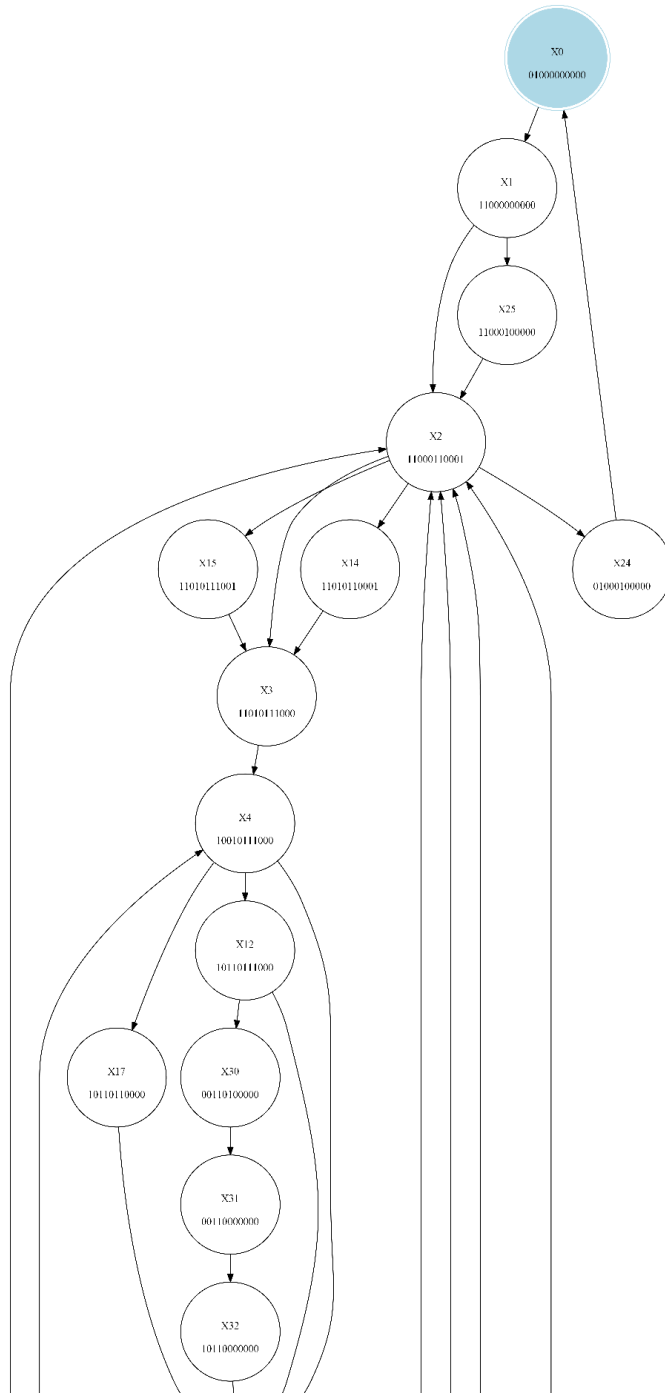


Figura 3.9: Autómata identificado subsistema girado Parte 1

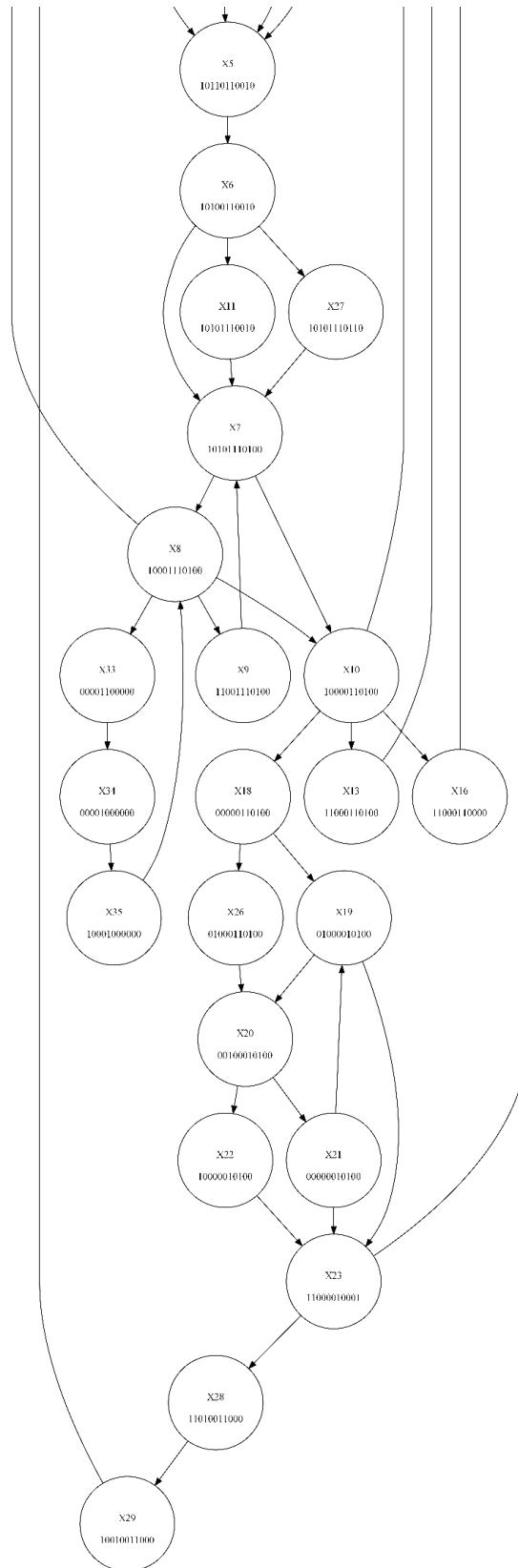


Figura 3.10: Autómata identificado subsistema girado Parte 2

En la Figura 3.9, Figura 3.10 se aprecia el modelo del autómata identificado del subsistema girado, el cual describe una estructura simple de estados donde se representa solamente los ciclos encontrados en el sistema tomando como estado inicial el estado marcado en color azul y el marcado de los estados equivale solo al vector sin procesar de las observaciones únicas encontradas. La identificación se realizó en tres partes, la primera parte de generación de secuencias válidas encontrando 23 secuencias tardando 43,0327 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 27,0186 mili-segundos, obteniendo 36 estados.

```

X0,01000000000,X1
X1,11000000000,X2,X25
X2,11000110001,X3,X14,X15,X24
X3,11010111000,X4
X4,10010111000,X5,X12,X17
X5,10110110010,X6
X6,10100110010,X7,X11,X27
X7,10101110100,X8,X10
X8,10001110100,X9,X10,X2,X33
X9,11001110100,X7
X10,10000110100,X2,X13,X16,X18
X11,10101110010,X7
X12,10110111000,X5,X30
X13,11000110100,X2
X14,11010110001,X3
X15,11010111001,X3
X16,11000110000,X2
X17,10110110000,X5
X18,00000110100,X19,X26
X19,01000010100,X20,X23
X20,00100010100,X21,X22
X21,00000010100,X19,X23
X22,10000010100,X23
X23,11000010001,X2,X28
X24,01000100000,X0
X25,11000100000,X2
X26,01000110100,X20
X27,10101110110,X7
X28,11010011000,X29

```

**Tabla 3.6:** Código Autómata subsistema girado Parte 1

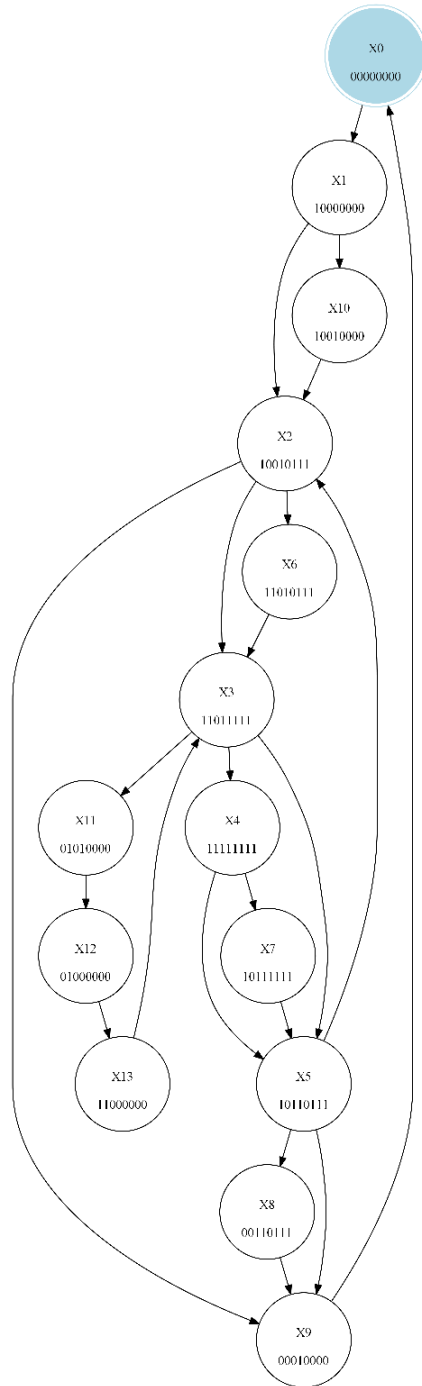
---

X29,10010011000,X4
X30,00110100000,X31
X31,00110000000,X32
X32,10110000000,X5
X33,00001100000,X34
X34,00001000000,X35
X35,10001000000,X8

**Tabla 3.7:** Código Autómata subsistema girado Parte 2

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_5\_Gira.aut» observable en la Tabla 3.6, Tabla 3.7 en este se guardan los nombres de cada estado seguidos del marcado asociado que es la observación sin repetición seguidas del conjunto de nombres de los estados con los que se conecta mediante transiciones. El guardado de la identificación se tardó 24,0132 mili-segundos, también en esta parte se genera la gráfica automática del autómata identificado, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz.

**3.1.6. Subsistema secado por luz UV**



**Figura 3.11:** Autómata identificado subsistema secado por luz UV



En la Figura 3.11 se aprecia el modelo del autómata identificado del subsistema secado por luz UV, el cual describe una estructura simple de estados donde se representa solamente los ciclos encontrados en el sistema tomando como estado inicial el estado marcado en color azul y el marcado de los estados equivale solo al vector sin procesar de las observaciones únicas encontradas. La identificación se realizó en tres partes, la primera parte de generación de secuencias válidas encontrando 28 secuencias tardando 63,5437 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 12,5118 mili-segundos, obteniendo 14 estados.

X0,00000000,X1
X1,10000000,X2,X10
X2,10010111,X3,X6,X9
X3,11011111,X4,X5,X11
X4,11111111,X5,X7
X5,10110111,X2,X8,X9
X6,11010111,X3
X7,10111111,X5
X8,00110111,X9
X9,00010000,X0
X10,10010000,X2
X11,01010000,X12
X12,01000000,X13
X13,11000000,X3

**Tabla 3.8:** Código Autómata subsistema secado por luz UV

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_6\_Secado2.aut» observable en la Tabla 3.8 en este se guardan los nombres de cada estado seguidos del marcado asociado que es la observación sin repetición seguidas del conjunto de nombres de los estados con los que se conecta mediante transiciones. El guardado de la identificación se tardó 21,0126 mili-segundos, también en esta parte se genera la gráfica automática del autómata identificado, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz .

### 3.1.7. Subsistema impresión

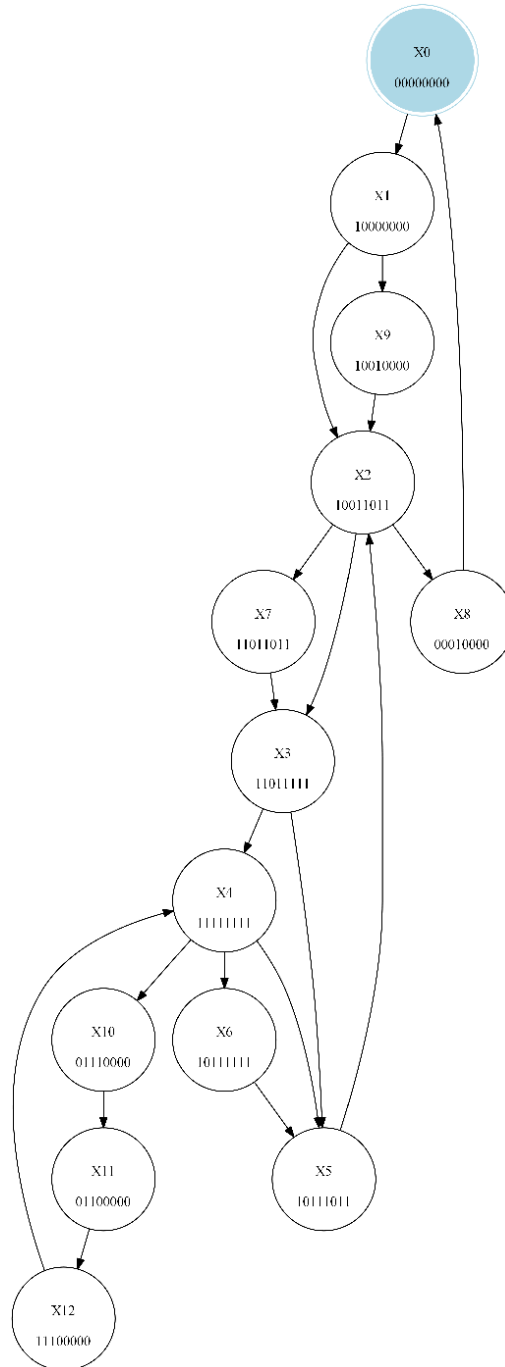


Figura 3.12: Autómata identificado subsistema impresión

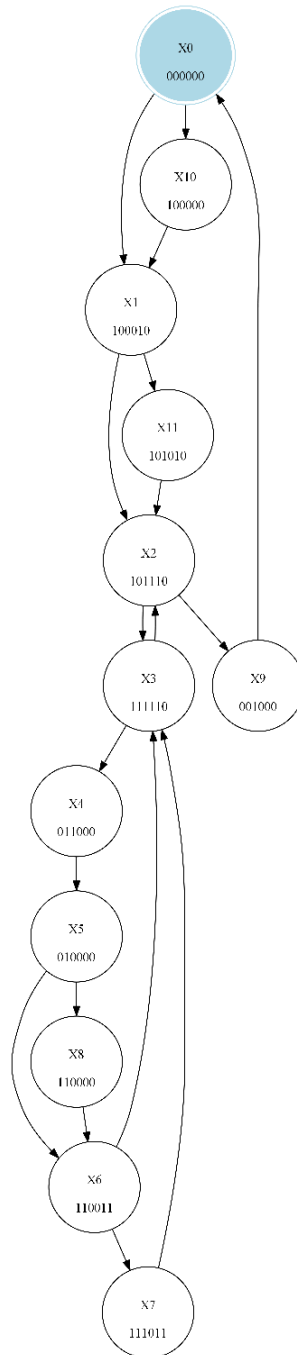
En la Figura 3.12 se aprecia el modelo del autómata identificado del subsistema impresión, el cual describe una estructura simple de estados donde se representa solamente los ciclos encontrados en el sistema tomando como estado inicial el estado marcado en color azul y el marcado de los estados equivale solo al vector sin procesar de las observaciones únicas encontradas. La identificación se realizó en tres partes, la primera parte de generación de secuencias válidas encontrando 28 secuencias tardando 62,0403 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 15,5132 mili-segundos, obteniendo 13 estados.

X0,00000000,X1
X1,10000000,X2,X9
X2,10011011,X3,X7,X8
X3,11011111,X4,X5
X4,11111111,X5,X6,X10
X5,10111011,X2
X6,10111111,X5
X7,11011011,X3
X8,00010000,X0
X9,10010000,X2
X10,01110000,X11
X11,01100000,X12
X12,11100000,X4

**Tabla 3.9:** Código Autómata subsistema impresión

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_7\_Impresion.aut» observable en la Tabla 3.9 en este se guardan los nombres de cada estado seguidos del marcado asociado que es la observación sin repetición seguidas del conjunto de nombres de los estados con los que se conecta mediante transiciones. El guardado de la identificación se tardó 34,0236 mili-segundos, también en esta parte se genera la gráfica automática del autómata identificado, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz .

**3.1.8. Subsistema horneado**



**Figura 3.13:** Autómata identificado subsistema horneado

En la Figura 3.13 se aprecia el modelo del autómata identificado del subsistema horneado, el cual describe una estructura simple de estados donde se representa solamente los ciclos encontrados en el sistema tomando como estado inicial el estado marcado en color azul y el marcado de los estados equivale solo al vector sin procesar de las observaciones únicas encontradas. La identificación se realizó en tres partes, la primera parte de generación de secuencias válidas encontrando 18 secuencias tardando 30,0213 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 7,5041 mili-segundos, obteniendo 12 estados.

X0,000000,X1,X10
X1,100010,X2,X11
X2,101110,X3,X9
X3,111110,X4,X2
X4,011000,X5
X5,010000,X6,X8
X6,110011,X3,X7
X7,111011,X3
X8,110000,X6
X9,001000,X0
X10,100000,X1
X11,101010,X2

**Tabla 3.10:** Código Autómata subsistema horneado

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_8\_Horneado.aut» observable en la Tabla 3.10 en este se guardan los nombres de cada estado seguidos del marcado asociado que es la observación sin repetición seguidas del conjunto de nombres de los estados con los que se conecta mediante transiciones. El guardado de la identificación se tardó 26,5171 mili-segundos, también en esta parte se genera la gráfica automática del autómata identificado, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz .

## 3.2. IPNs Con base al método de Estrada - Vargas :

Las redes de Petri obtenidas mediante el método de Estrada - Vargas, están constituidas en múltiples partes, una primera es los conjuntos de eventos únicos observados seguidos de los conjuntos de lugares y transiciones, cada elemento está especificado por cada línea de texto, donde los elementos están organizados primero el nombre del evento seguido del evento observado, luego el conjunto de símbolos de entrada y por último el conjunto de símbolos de salida.

El conjunto de lugares está conformado por el nombre del lugar, el vector de marcado asociado, el conjunto de nombres de transiciones previas y por último del conjunto de nombres de transiciones posteriores. De igual manera las transiciones se conforma por el nombre de la transición, el nombre del evento asociado, el nombre del lugar previo, el nombre del lugar posteriores y finalmente el conjunto de símbolos de entrada asociados al evento de la transición.

### 3.2.1. Subsistema suministro de materias primas

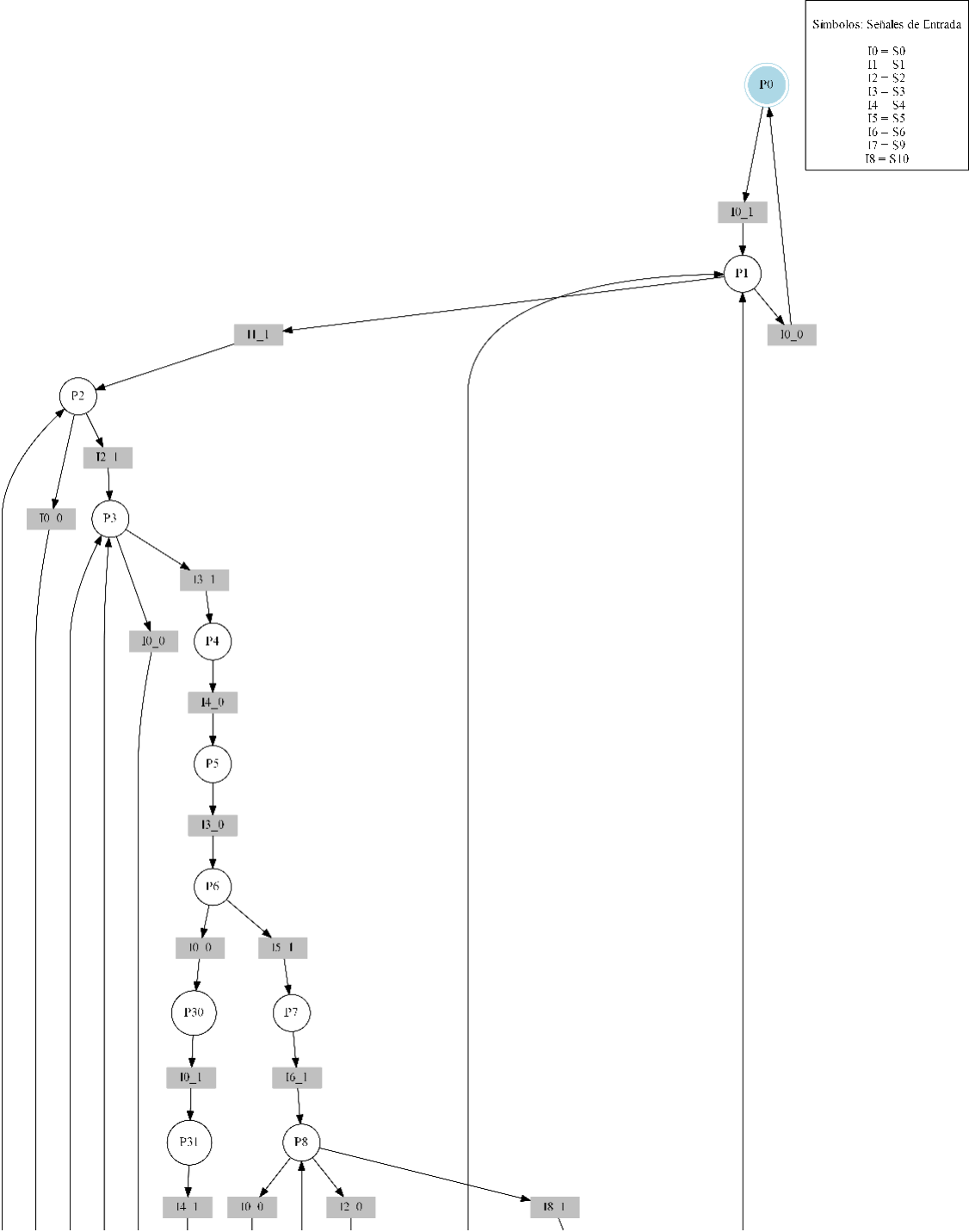


Figura 3.14: PN identificada subsistema suministro Parte 1





En la Figura 3.14, Figura 3.15 se aprecia el modelo de la PN identificada del subsistema suministro de materias primas, el cual describe una estructura mas elaborada de lugares y transiciones donde se representa la dinámica de comportamiento logrando observar el sistema de manera mas completa ya que en este se procesan todas las observaciones. La identificación se realizó en tres partes, la primera parte de generación de eventos con un total de 283 observaciones se tardó 168,7143 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 23,0152 mili-segundos, reduciendo el número de eventos a 32, obteniendo 46 lugares y 59 transiciones.

<p>S0,S1,S2,S3,S4,S5,S6,S9,S10,Banda1,Compuerta_Izq,Compuerta_Der,Luz_Material,  I0,I1,I2,I3,I4,I5,I6,I7,I8,O0,O1,O2,O3,O4  M0,P0  E0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,I0_1   O3_1  E1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,I1_1   ε  E2,0,0,1,0,0,0,0,0,0,0,0,0,0,0,I2_1   ε  E3,0,0,0,1,0,0,0,0,0,1,0,1,-1,I3_1   O0_1,O2_1,O3_0  E4,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,I4_0   ε  E5,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,I3_0   ε  E6,0,0,0,0,1,0,0,0,0,0,-1,0,I5_1   O2_0  E7,0,0,0,0,0,0,1,0,0,0,0,0,0,0,I6_1   ε  E8,0,0,-1,0,0,0,0,0,0,0,0,0,1,I2_0   O3_1  E9,0,0,0,1,0,0,0,0,0,0,0,0,-1,I3_1   O3_0  E10,0,0,0,0,0,0,0,0,1,0,0,0,0,I8_1   ε  E11,0,0,0,0,0,0,0,0,-1,0,0,0,0,I8_0   ε  E12,0,0,0,1,0,0,0,0,0,0,0,0,0,I3_1   ε  E13,0,0,0,0,0,0,0,0,0,0,0,0,-1,ε   O3_0  E14,0,0,0,0,0,0,0,1,0,0,0,0,0,I7_1   ε  E15,0,0,0,0,0,0,0,0,0,0,1,0,0,ε   O1_1  E16,0,0,0,0,0,-1,0,0,0,0,0,0,0,I5_0   ε  E17,0,0,0,0,0,0,0,-1,0,0,0,0,0,I7_0   ε  E18,0,0,0,0,1,0,0,0,0,0,-1,0,0,I4_1   O1_0  E19,0,0,0,0,0,0,-1,0,0,-1,0,0,0,I6_0   O0_0  E20,-1,0,0,0,0,0,0,0,0,0,0,0,0,I0_0   ε  E21,1,0,0,0,0,0,0,0,0,0,0,0,0,I0_1   ε  E22,0,0,0,0,0,0,0,0,-1,0,0,0,1,I8_0   O3_1  E23,-1,0,0,0,0,0,0,0,0,0,0,0,-1,I0_0   O3_0  E24,0,-1,0,0,0,0,0,0,0,1,0,-1,I1_0   O1_1,O3_0  E25,0,0,0,0,0,-1,0,0,-1,0,0,1,I6_0   O0_0,O3_1  E26,-1,0,0,0,0,0,0,0,0,1,-1,0,I0_0   O1_1,O2_0</p>
---

**Tabla 3.11:** PN subsistema de suministro Parte 1

E27,0,0,0,0,0,0,-1,0,0,0,0,0,I6_0  ε
E28,-1,0,0,0,0,0,0,0,-1,0,0,0,I0_0  O0_0
E29,-1,0,0,0,0,0,0,0,0,1,0,0,I0_0  O1_1
E30,0,0,-1,0,0,0,0,0,0,0,0,I2_0  ε
E31,0,0,0,0,0,0,0,0,0,0,1,ε  O3_1
P0,000010000000,T37  T0
P1,100010000001,T0,T36,T58  T1,T37
P2,110010000001,T1,T49  T2,T50
P3,111010000001,T2,T30,T32  T3,T31
P4,1111100001010,T3  T4
P5,1111000001010,T4  T5
P6,1110000001010,T5  T6,T38
P7,1110010001000,T6  T7
P8,1110011001000,T7,T11  T8,T12,T44
P9,1100011001001,T8  T9,T33
P10,1110011001001,T9,T15  T10,T16
P11,1111011001000,T10,T17  T11
P12,1110011011000,T12,T19  T13,T20,T51
P13,1100011011001,T13,T52  T14
P14,1110011011001,T14  T15,T18,T53
P15,1111011001001,T16  T17
P16,1111011011000,T18  T19
P17,1110011111000,T20  T21
P18,1110011111100,T21  T22
P19,1110001111100,T22  T23
P20,1110001011100,T23  T24
P21,1110101011000,T24  T25
P22,1110100010000,T25  T26
P23,1110100110000,T26  T27
P24,1110100010000,T27,T29  T28,T30
P25,0110100010000,T28  T29
P26,0110100000000,T31,T43  T32
P27,1000011001100,T33  T34
P28,1000001001100,T34  T35
P29,1000101001000,T35  T36
P30,0110000001100,T38  T39
P31,1110000001100,T39  T40
P32,1110100001000,T40  T41
P33,1110101001000,T41  T42
P34,1110100001000,T42  T43
P35,0110011001100,T44  T45
P36,0110001001100,T45  T46
P37,0100001001100,T46  T47

**Tabla 3.12:** PN subsistema de suministro Parte 2

P38,0100101001000,T47|||T48  
 P39,0100100000000,T48,T50|||T49  
 P40,1100011011000,T51|||T52  
 P41,1100011011001,T53|||T54  
 P42,1000011011100,T54|||T55  
 P43,1000001011100,T55|||T56  
 P44,1000101011000,T56|||T57  
 P45,1000100010000,T57|||T58  
 T0,E0,P0,P1,I0\_1  
 T1,E1,P1,P2,I1\_1  
 T2,E2,P2,P3,I2\_1  
 T3,E3,P3,P4,I3\_1  
 T4,E4,P4,P5,I4\_0  
 T5,E5,P5,P6,I3\_0  
 T6,E6,P6,P7,I5\_1  
 T7,E7,P7,P8,I6\_1  
 T8,E8,P8,P9,I2\_0  
 T9,E2,P9,P10,I2\_1  
 T10,E9,P10,P11,I3\_1  
 T11,E5,P11,P8,I3\_0  
 T12,E10,P8,P12,I8\_1  
 T13,E8,P12,P13,I2\_0  
 T14,E2,P13,P14,I2\_1  
 T15,E11,P14,P10,I8\_0  
 T16,E12,P10,P15,I3\_1  
 T17,E13,P15,P11, $\epsilon$   
 T18,E9,P14,P16,I3\_1  
 T19,E5,P16,P12,I3\_0  
 T20,E14,P12,P17,I7\_1  
 T21,E15,P17,P18, $\epsilon$   
 T22,E16,P18,P19,I5\_0  
 T23,E17,P19,P20,I7\_0  
 T24,E18,P20,P21,I4\_1  
 T25,E19,P21,P22,I6\_0  
 T26,E14,P22,P23,I7\_1  
 T27,E17,P23,P24,I7\_0  
 T28,E20,P24,P25,I0\_0  
 T29,E21,P25,P24,I0\_1  
 T30,E22,P24,P3,I8\_0  
 T31,E23,P3,P26,I0\_0  
 T32,E0,P26,P3,I0\_1  
 T33,E24,P9,P27,I1\_0

**Tabla 3.13:** PN subsistema de suministro Parte 3

T34,E16,P27,P28,I5_0
T35,E18,P28,P29,I4_1
T36,E25,P29,P1,I6_0
T37,E23,P1,P0,I0_0
T38,E26,P6,P30,I0_0
T39,E21,P30,P31,I0_1
T40,E18,P31,P32,I4_1
T41,E7,P32,P33,I6_1
T42,E27,P33,P34,I6_0
T43,E28,P34,P26,I0_0
T44,E29,P8,P35,I0_0
T45,E16,P35,P36,I5_0
T46,E30,P36,P37,I2_0
T47,E18,P37,P38,I4_1
T48,E19,P38,P39,I6_0
T49,E0,P39,P2,I0_1
T50,E23,P2,P39,I0_0
T51,E30,P12,P40,I2_0
T52,E31,P40,P13, $\epsilon$
T53,E30,P14,P41,I2_0
T54,E24,P41,P42,I1_0
T55,E16,P42,P43,I5_0
T56,E18,P43,P44,I4_1
T57,E19,P44,P45,I6_0
T58,E22,P45,P1,I8_0

**Tabla 3.14:** PN subsistema de suministro Parte 4

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_1\_Suministro.ipn» observable en la Tabla 3.11, Tabla 3.12, Tabla 3.13, Tabla 3.14 en este se guardan los nombres de cada señal, la señal abreviada, el conjunto de eventos únicos observados, el conjunto de lugares y las transiciones tardando 30,0205 mili-segundos, también en esta parte es donde se genera la gráfica automática de la PN identificada, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz , en el modelo graficado el lugar inicial esta determinado por color azul.

### 3.2.2. Subsistema molienda de material

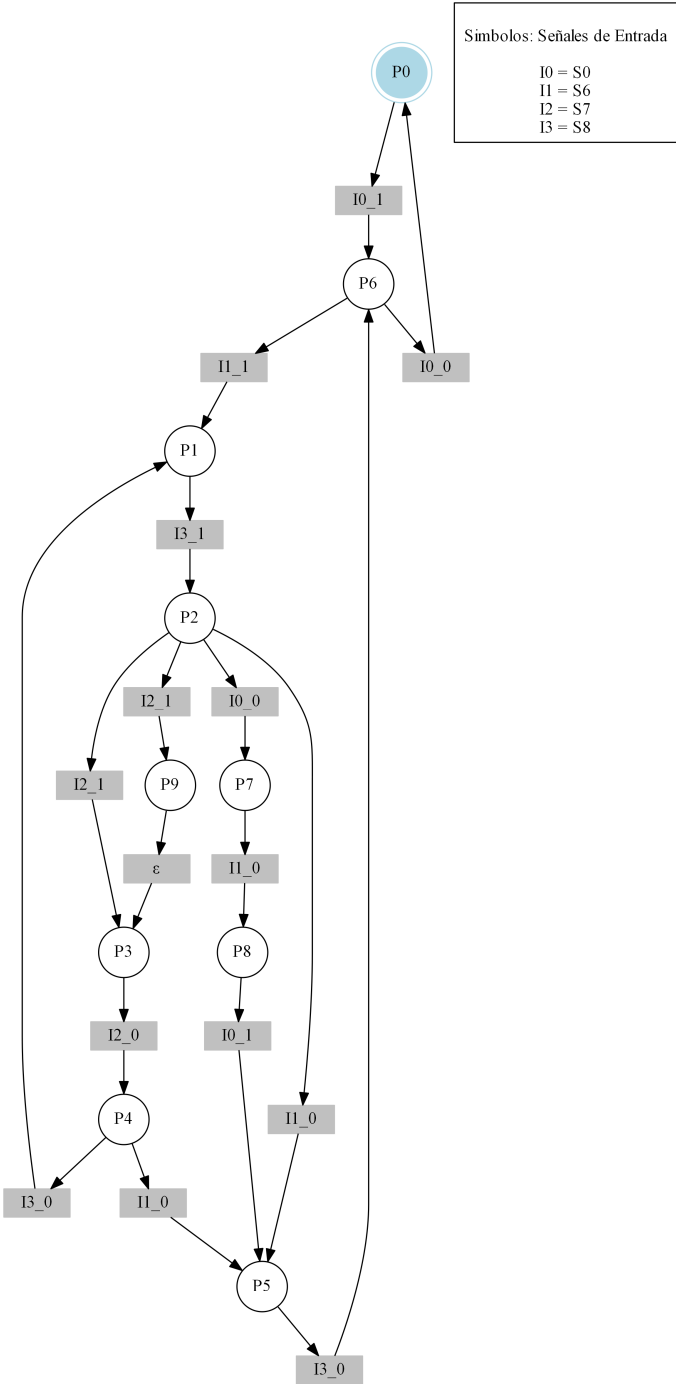


Figura 3.16: PN identificada subsistema molino

En la Figura 3.16 se aprecia el modelo de la PN identificada del subsistema molino, el cual describe una estructura mas elaborada de lugares y transiciones donde se representa la dinámica de comportamiento logrando observar el sistema de manera mas completa ya que en este se procesan todas las observaciones. La identificación se realizó en tres partes, la primera parte de generación de eventos con un total de 145 observaciones se tardó 96,061 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 26,8157 mili-segundos, reduciendo el número de eventos a 12, obteniendo 10 lugares y 15 transiciones.

```

S0,S6,S7,S8,Molino,Bomba1,Valvula1,
I0,I1,I2,I3,O0,O1,O2,O3
M0,P0
E0,1,0,0,0,0,0,0,I0_1||ε
E1,0,1,0,0,1,0,1,I1_1||O0_1,O2_1
E2,0,0,0,1,0,0,0,I3_1||ε
E3,0,0,1,0,0,1,0,I2_1||O1_1
E4,0,0,-1,0,0,0,0,I2_0||ε
E5,0,0,0,-1,0,-1,0,I3_0||O1_0
E6,0,-1,0,0,0,1,-1,I1_0||O1_1,O2_0
E7,0,0,0,-1,-1,-1,0,I3_0||O0_0,O1_0
E8,-1,0,0,0,0,0,0,I0_0||ε
E9,0,-1,0,0,0,0,-1,I1_0||O2_0
E10,0,0,1,0,0,0,0,I2_1||ε
E11,0,0,0,0,0,1,0,ε||O1_1
P0,0000000,T8||T0
P1,1100101,T1,T5||T2
P2,1101101,T2||T3,T6,T10,T13
P3,1111111,T3,T14||T4
P4,1101111,T4||T5,T9
P5,1001110,T6,T9,T12||T7
P6,1000000,T7,T0||T8,T1
P7,0101101,T10||T11
P8,0001110,T11||T12
P9,1111101,T13||T14
T0,E0,P0,P6,I0_1
T1,E1,P6,P1,I1_1
T2,E2,P1,P2,I3_1
T3,E3,P2,P3,I2_1
T4,E4,P3,P4,I2_0

```

**Tabla 3.15:** Código PN subsistema molino Parte 1

---

T5,E5,P4,P1,I3_0
T6,E6,P2,P5,I1_0
T7,E7,P5,P6,I3_0
T8,E8,P6,P0,I0_0
T9,E9,P4,P5,I1_0
T10,E8,P2,P7,I0_0
T11,E6,P7,P8,I1_0
T12,E0,P8,P5,I0_1
T13,E10,P2,P9,I2_1
T14,E11,P9,P3, $\epsilon$

**Tabla 3.16:** Código PN subsistema molino Parte 2

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_2\_Molino.ipn» observable en la Tabla 3.15, Tabla 3.16 en este se guardan los nombres de cada señal, la señal abreviada, el conjunto de eventos únicos observados, el conjunto de lugares y las transiciones tardando 30,0189 mili-segundos, también en esta parte es donde se genera la gráfica automática de la PN identificada, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz , en el modelo graficado el lugar inicial esta determinado por color azul.

### 3.2.3. Subsistema atomizado y secado

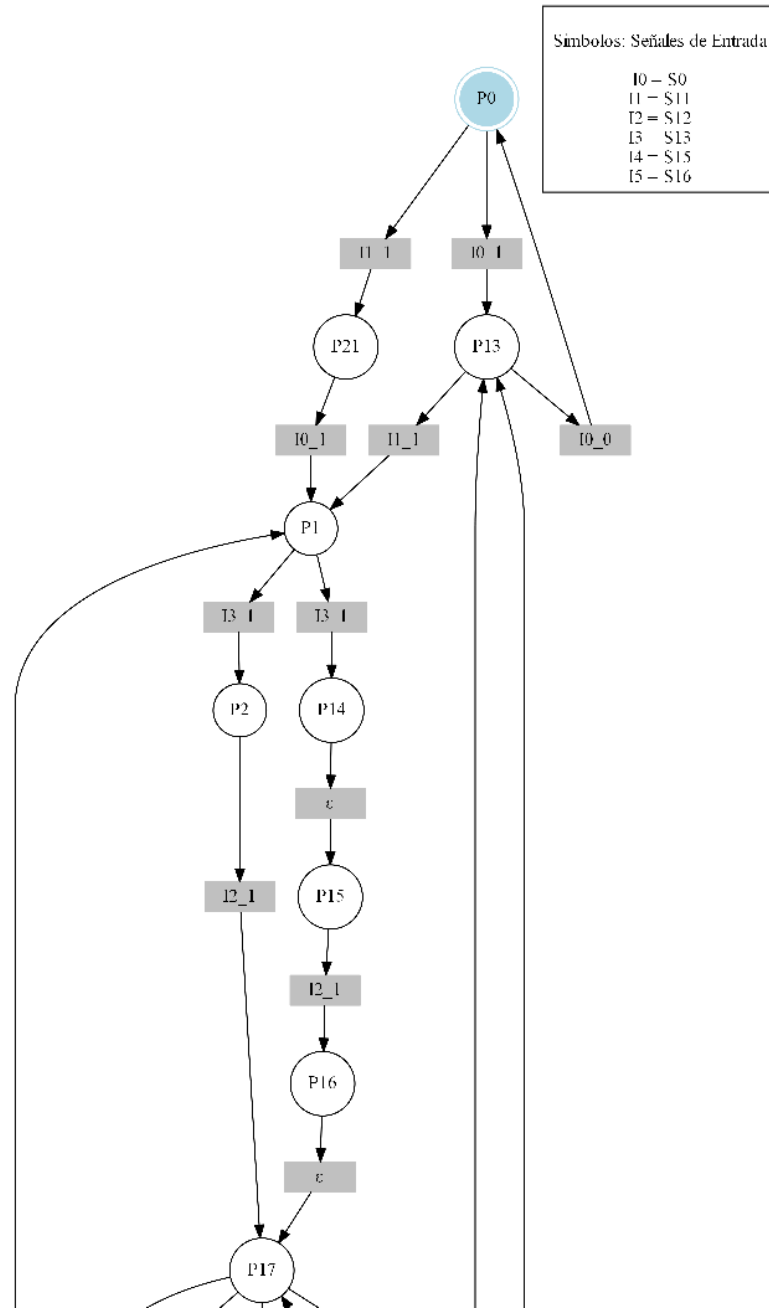


Figura 3.17: PN identificada subsistema atomizado y secado Parte 1



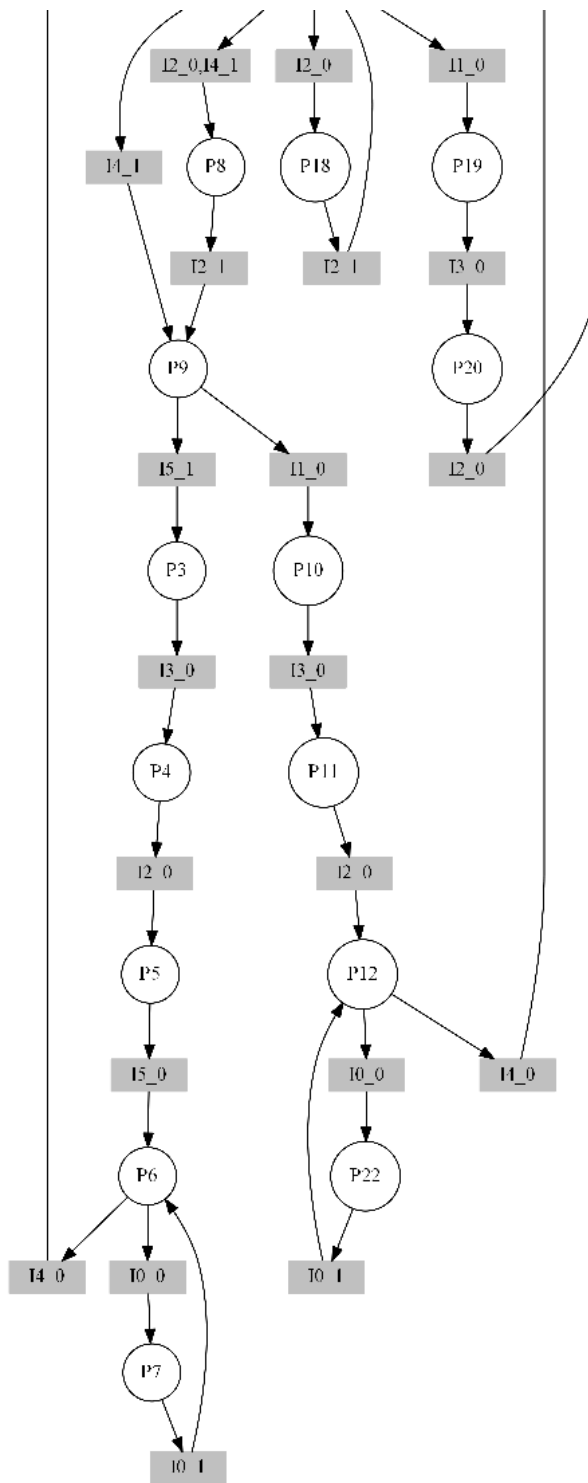


Figura 3.18: PN identificada subsistema atomizado y secado Parte 2

En la Figura 3.17, Figura 3.18 se aprecia el modelo de la PN identificada del subsistema atomizado y secado, el cual describe una estructura mas elaborada de lugares y transiciones donde se representa la dinámica de comportamiento logrando observar el sistema de manera mas completa ya que en este se procesan todas las observaciones. La identificación se realizó en tres partes, la primera parte de generación de eventos con un total de 119 observaciones se tardó 46,8572 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 15,6068 mili-segundos, reduciendo el número de eventos a 21, obteniendo 23 lugares y 32 transiciones.

<p>S0,S11,S12,S13,S15,S16,Quemador1,Ventilador,Banda2,Bomba2,Banda3,  I0,I1,I2,I3,I4,I5,O0,O1,O2,O3,O4,O5  M0,P0  E0,1,0,0,0,0,0,0,0,0,0,I0_1  ε  E1,0,1,0,0,0,0,1,1,0,0,0,I1_1  O0_1,O1_1  E2,0,0,0,1,0,0,0,0,1,1,0,I3_1  O2_1,O3_1  E3,0,0,1,0,0,0,0,0,0,1,I2_1  O4_1  E4,0,0,0,0,1,0,0,0,0,0,I4_1  ε  E5,0,0,0,0,1,-1,-1,0,-1,0,I5_1  O0_0,O1_0,O3_0  E6,0,0,0,-1,0,0,0,0,0,0,I3_0  ε  E7,0,0,-1,0,0,0,0,0,-1,0,-1,I2_0  O2_0,O4_0  E8,0,0,0,0,0,-1,0,0,0,0,I5_0  ε  E9,-1,0,0,0,0,0,0,0,0,0,I0_0  ε  E10,0,0,0,0,-1,0,1,1,0,0,0,I4_0  O0_1,O1_1  E11,0,0,-1,0,1,0,0,0,0,0,I2_0,I4_1  ε  E12,0,0,1,0,0,0,0,0,0,0,I2_1  ε  E13,0,-1,0,0,0,0,-1,-1,0,-1,0,I1_0  O0_0,O1_0,O3_0  E14,0,0,0,0,-1,0,0,0,0,0,I4_0  ε  E15,0,0,0,1,0,0,0,0,0,0,I3_1  ε  E16,0,0,0,0,0,0,0,0,1,1,0,ε  O2_1,O3_1  E17,0,0,0,0,0,0,0,0,0,1,ε  O4_1  E18,0,0,-1,0,0,0,0,0,0,0,I2_0  ε  E19,0,1,0,0,0,0,0,0,0,0,I1_1  ε  E20,1,0,0,0,0,0,1,1,0,0,0,I0_1  O0_1,O1_1  P0,0000000000,T18  T0,T28  P1,11000011000,T1,T11,T29  T2,T19  P2,11010011110,T2  T3  P3,11111100101,T5  T6  P4,11101100101,T6  T7  P5,11001100000,T7  T8</p>
---

**Tabla 3.17:** Código PN subsistema atomizado y secado Parte 1

P6,11001000000,T8,T10|||T9,T11  
 P7,01001000000,T9|||T10  
 P8,11011011111,T12|||T13  
 P9,11111011111,T13,T4|||T14,T5  
 P10,10111000101,T14|||T15  
 P11,10101000101,T15|||T16  
 P12,10001000000,T16,T31|||T17,T30  
 P13,10000000000,T17,T0,T27|||T18,T1  
 P14,11010011000,T19|||T20  
 P15,11010011110,T20|||T21  
 P16,11110011110,T21|||T22  
 P17,11110011111,T22,T3,T24|||T23,T4,T12,T25  
 P18,11010011111,T23|||T24  
 P19,10110000101,T25|||T26  
 P20,10100000101,T26|||T27  
 P21,01000000000,T28|||T29  
 P22,00001000000,T30|||T31  
 T0,E0,P0,P13,I0\_1  
 T1,E1,P13,P1,I1\_1  
 T2,E2,P1,P2,I3\_1  
 T3,E3,P2,P17,I2\_1  
 T4,E4,P17,P9,I4\_1  
 T5,E5,P9,P3,I5\_1  
 T6,E6,P3,P4,I3\_0  
 T7,E7,P4,P5,I2\_0  
 T8,E8,P5,P6,I5\_0  
 T9,E9,P6,P7,I0\_0  
 T10,E0,P7,P6,I0\_1  
 T11,E10,P6,P1,I4\_0  
 T12,E11,P17,P8,I2\_0,I4\_1  
 T13,E12,P8,P9,I2\_1  
 T14,E13,P9,P10,I1\_0  
 T15,E6,P10,P11,I3\_0  
 T16,E7,P11,P12,I2\_0  
 T17,E14,P12,P13,I4\_0  
 T18,E9,P13,P0,I0\_0  
 T19,E15,P1,P14,I3\_1  
 T20,E16,P14,P15, $\epsilon$   
 T21,E12,P15,P16,I2\_1  
 T22,E17,P16,P17, $\epsilon$   
 T23,E18,P17,P18,I2\_0

**Tabla 3.18:** Código PN subsistema atomizado y secado Parte 2

T24,E12,P18,P17,I2_1
T25,E13,P17,P19,I1_0
T26,E6,P19,P20,I3_0
T27,E7,P20,P13,I2_0
T28,E19,P0,P21,I1_1
T29,E20,P21,P1,I0_1
T30,E9,P12,P22,I0_0
T31,E0,P22,P12,I0_1

**Tabla 3.19:** Código PN subsistema atomizado y secado Parte 3

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_3\_Secado1.ipn» observable en la Tabla 3.17, Tabla 3.18, Tabla 3.19 en este se guardan los nombres de cada señal, la señal abreviada, el conjunto de eventos únicos observados, el conjunto de lugares y las transiciones tardando 93,732 mili-segundos, también en esta parte es donde se genera la gráfica automática de la PN identificada, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz , en el modelo graficado el lugar inicial esta determinado por color azul.



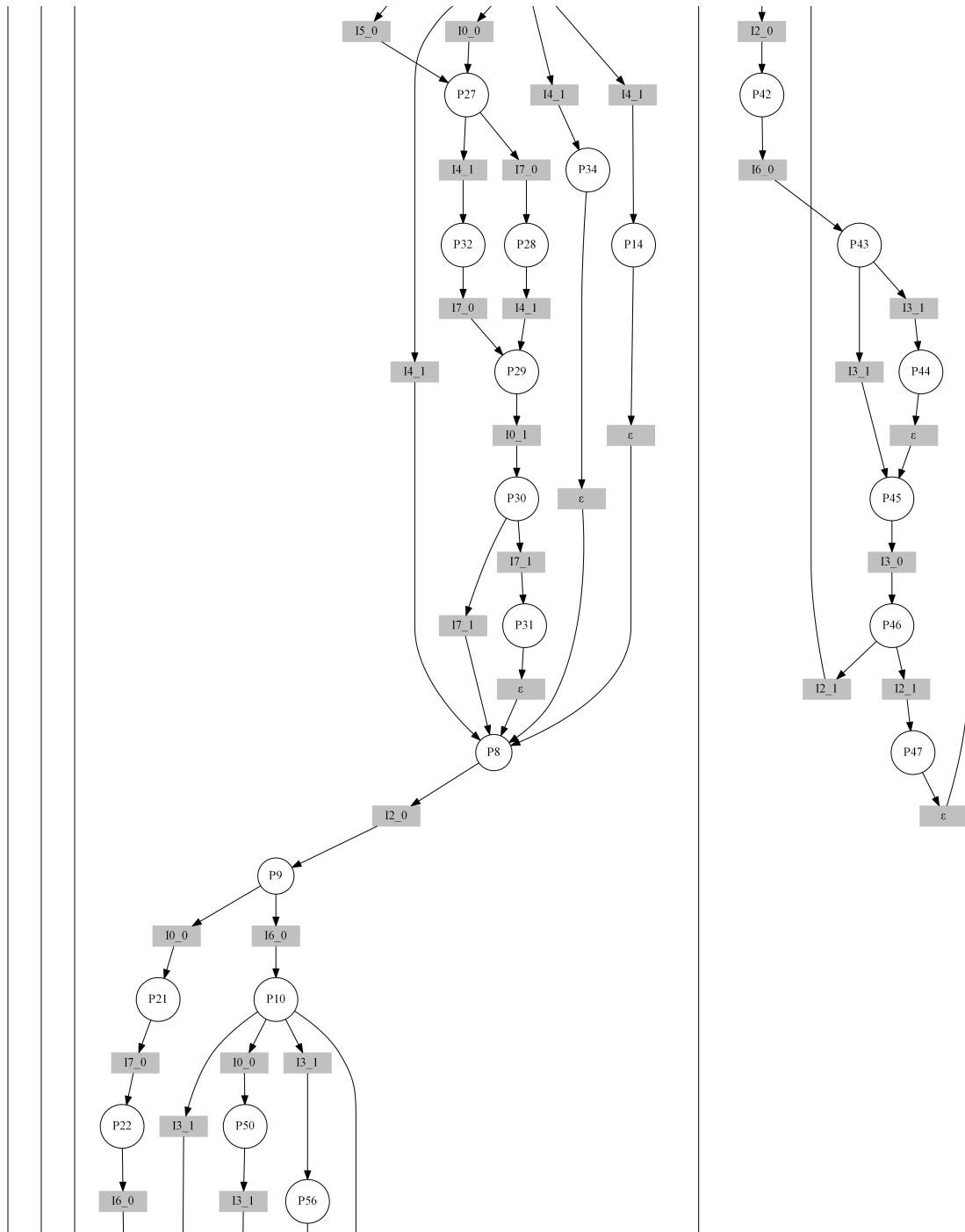


Figura 3.20: PN identificada subsistema prensado Parte 2

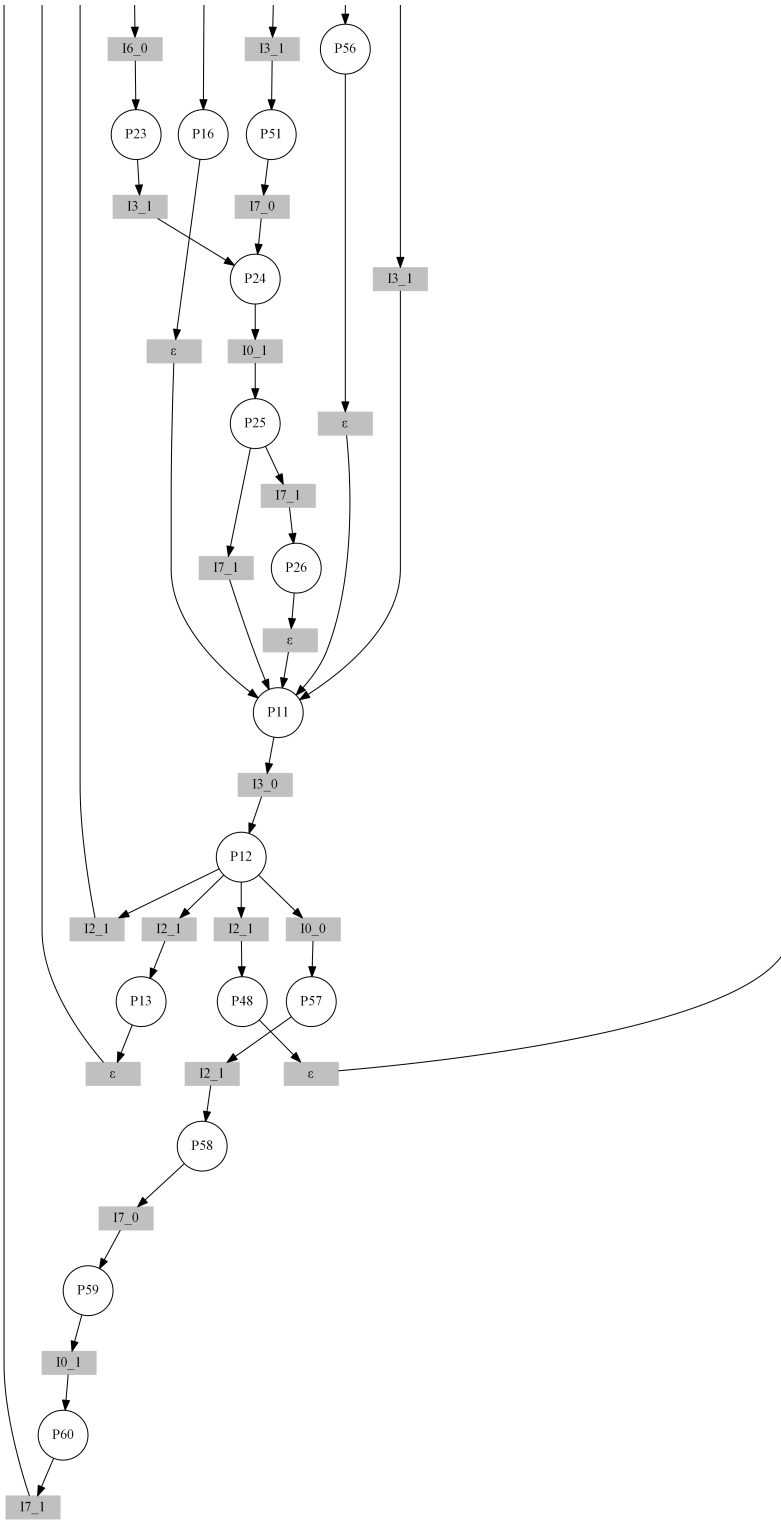


Figura 3.21: PN identificada subsistema prensado Parte 3

En la Figura 3.19, Figura 3.20, Figura 3.21 se aprecia el modelo de la PN identificada del subsistema prensado, el cual describe una estructura mas elaborada de lugares y transiciones donde se representa la dinámica de comportamiento logrando observar el sistema de manera mas completa ya que en este se procesan todas las observaciones. La identificación se realizó en tres partes, la primera parte de generación de eventos con un total de 13218 observaciones se tardó 7240,1619 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 65,0479 mili-segundos, reduciendo el número de eventos a 47, obteniendo 63 lugares y 89 transiciones.

<p>S0,S14,S17,S18,S19,S20,S21,S30,Prensa_Arriba,Prensa_Abajo,paleta_Inicio,</p> <p style="text-align: right;">Paleta_Fin,Valvula2,Aspiradora,</p> <p>I0,I1,I2,I3,I4,I5,I6,I7,O0,O1,O2,O3,O4,O5,O6</p> <p>M0,P0</p> <p>E0,1,0,0,0,0,0,0,0,0,0,0,0,0,I0_1   ε</p> <p>E1,0,0,0,0,0,0,0,1,0,0,0,0,0,I7_1   ε</p> <p>E2,0,1,0,0,0,0,0,0,0,0,1,1,1,I1_1   O3_1,O4_1,O5_1</p> <p>E3,0,0,0,0,-1,0,0,0,0,0,0,0,0,I4_0   ε</p> <p>E4,0,0,0,0,0,1,0,0,0,0,0,0,0,I6_1   ε</p> <p>E5,0,0,0,0,0,1,0,0,0,0,1,-1,-1,-1,I5_1   O2_1,O3_0,O4_0,O5_0</p> <p>E6,0,0,0,0,0,-1,0,0,0,0,0,0,0,I5_0   ε</p> <p>E7,0,0,0,0,1,0,0,0,0,1,-1,0,0,0,I4_1   O1_1,O2_0</p> <p>E8,0,0,-1,0,0,0,0,0,0,0,0,0,0,I2_0   ε</p> <p>E9,0,0,0,0,0,-1,0,0,0,0,0,0,0,I6_0   ε</p> <p>E10,0,0,0,1,0,0,0,0,1,-1,0,0,0,I3_1   O0_1,O1_0</p> <p>E11,0,0,0,-1,0,0,0,0,0,0,0,0,0,I3_0   ε</p> <p>E12,0,0,1,0,0,0,0,0,-1,0,0,1,1,1,I2_1   O0_0,O3_1,O4_1,O5_1</p> <p>E13,0,0,1,0,0,0,0,0,0,0,0,0,0,I2_1   ε</p> <p>E14,0,0,0,0,0,0,0,0,-1,0,0,1,1,1,ε   O0_0,O3_1,O4_1,O5_1</p> <p>E15,0,0,0,0,1,0,0,0,0,0,0,0,0,I4_1   ε</p> <p>E16,0,0,0,0,0,0,0,0,0,1,-1,0,0,0,ε   O1_1,O2_0</p> <p>E17,0,0,0,0,0,1,0,0,0,0,0,0,0,I5_1   ε</p> <p>E18,0,0,0,0,0,0,0,0,0,1,-1,-1,-1,ε   O2_1,O3_0,O4_0,O5_0</p> <p>E19,0,0,0,1,0,0,0,0,0,0,0,0,0,I3_1   ε</p> <p>E20,0,0,0,0,0,0,0,0,1,-1,0,0,0,0,ε   O0_1,O1_0</p> <p>E21,0,0,0,0,0,1,0,0,0,0,1,0,0,0,I5_1   O2_1</p> <p>E22,0,0,0,0,0,0,0,0,0,0,-1,-1,-1,ε   O3_0,O4_0,O5_0</p> <p>E23,-1,0,0,0,0,0,0,0,0,0,-1,-1,-1,I0_0   O3_0,O4_0,O5_0</p> <p>E24,0,0,0,0,0,0,0,-1,0,0,0,0,0,I7_0   ε</p>
--

**Tabla 3.20:** Código PN subsistema prensado Parte 1



E25,0,0,0,0,0,0,0,1,0,0,0,1,1,1,I7_1   O3_1,O4_1,O5_1
E26,-1,0,0,0,0,0,0,0,0,0,0,0,0,I0_0   ε
E27,0,0,0,1,0,0,0,0,0,-1,0,0,0,0,I3_1   O1_0
E28,0,0,0,0,0,0,0,0,1,0,0,0,0,ε   O0_1
E29,0,0,0,0,1,0,0,0,0,0,-1,0,0,0,I4_1   O2_0
E30,0,0,0,0,0,0,0,0,0,1,0,0,0,ε   O1_1
E31,0,0,0,0,0,0,0,1,0,1,0,0,0,0,I7_1   O1_1
E32,0,0,0,0,0,0,0,1,1,0,0,0,0,0,I7_1   O0_1
E33,0,0,0,0,1,0,0,0,0,1,-1,-1,0,I5_1   O2_1,O3_0,O4_0
E34,0,0,0,0,0,0,0,0,0,0,0,0,-1,ε   O5_0
E35,0,0,0,0,1,0,0,0,0,1,0,0,0,0,I4_1   O1_1
E36,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,ε   O2_0
E37,0,-1,0,0,0,0,0,0,0,0,-1,-1,-1,I1_0   O3_0,O4_0,O5_0
E38,0,-1,0,0,0,1,0,0,0,0,1,-1,-1,-1,I1_0,I5_1   O2_1,O3_0,O4_0,O5_0
E39,0,0,0,0,0,0,0,0,-1,0,0,0,0,ε   O0_0
E40,0,0,1,0,0,0,0,0,-1,0,0,1,1,0,I2_1   O0_0,O3_1,O4_1
E41,0,0,0,0,0,0,0,0,0,0,0,0,0,1,ε   O5_1
E42,0,0,1,0,0,0,0,0,-1,0,0,0,0,0,I2_1   O0_0
E43,0,-1,0,0,0,0,1,0,0,0,0,-1,-1,-1,I1_0,I6_1   O3_0,O4_0,O5_0
E44,0,-1,0,0,-1,0,0,0,0,0,0,-1,-1,-1,I1_0,I4_0   O3_0,O4_0,O5_0
E45,0,0,0,1,0,0,0,0,1,0,0,0,0,0,I3_1   O0_1
E46,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,ε   O1_0
P0,00101000000000,T   T0
P1,10101000000000,T0   T1
P2,10101001000000,T1,T63,T72   T2,T87
P3,11101001000111,T2,T12,T14,T65,T86   T3,T79
P4,11100001000111,T3,T26,T53   T4,T23,T48,T73
P5,11100011000111,T4,T78   T5,T17,T21,T44,T54
P6,11100111001000,T5,T18,T22,T45   T6,T66
P7,11100011001000,T6   T7,T15,T34,T46
P8,11101011010000,T7,T16,T39,T42,T47   T8
P9,11001011010000,T8   T9,T27
P10,11001001010000,T9   T10,T19,T68,T80
P11,11011001100000,T10,T20,T33,T43,T81   T11
P12,11001001100000,T11   T12,T13,T64,T82
P13,11101001100000,T13   T14
P14,11101011001000,T15   T16
P15,11100111000111,T17   T18
P16,11011001010000,T19   T20
P17,11100111001111,T21   T22
P18,01100001000000,T23   T24
P19,01100000000000,T24   T25

**Tabla 3.21:** Código PN subsistema prensado Parte 2

P20,11100000000000,T25   T26
P21,01001011010000,T27   T28
P22,01001010010000,T28   T29
P23,01001000010000,T29   T30
P24,01011000000000,T30,T70   T31
P25,11011000000000,T31   T32,T43
P26,11011001000000,T32   T33
P27,01100011001000,T34,T67   T35,T40
P28,01100010001000,T35   T36
P29,01101010000000,T36,T41   T37
P30,11101010000000,T37   T38,T42
P31,11101011000000,T38   T39
P32,01101011000000,T40   T41
P33,11100111001001,T44   T45
P34,11101011011000,T46   T47
P35,10100001000000,T48,T52,T79   T49,T53
P36,00100001000000,T49   T50
P37,00100000000000,T50   T51
P38,10100000000000,T51   T52
P39,10100111001000,T54   T55
P40,10100011001000,T55   T56
P41,10101011010000,T56   T57
P42,10001011010000,T57   T58
P43,10001001010000,T58   T59,T71
P44,10011001010000,T59   T60
P45,10011001100000,T60,T71   T61
P46,10001001100000,T61   T62,T72
P47,10101001100000,T62   T63
P48,11101001000110,T64   T65
P49,01100111001000,T66   T67
P50,01001001010000,T68   T69
P51,01011001000000,T69   T70
P52,10100011000000,T73,T77   T74,T78
P53,00100011000000,T74   T75
P54,00100010000000,T75   T76
P55,10100010000000,T76   T77
P56,11011001110000,T80   T81
P57,01001001100000,T82   T83
P58,01101001000000,T83   T84
P59,01101000000000,T84   T85

**Tabla 3.22:** Código PN subsistema prensado Parte 3

P60,11101000000000,T85|||T86  
 P61,00101001000000,T87|||T88  
 P62,00101000000000,T88|||ε  
 T0,E0,P0,P1,I0\_1  
 T1,E1,P1,P2,I7\_1  
 T2,E2,P2,P3,I1\_1  
 T3,E3,P3,P4,I4\_0  
 T4,E4,P4,P5,I6\_1  
 T5,E5,P5,P6,I5\_1  
 T6,E6,P6,P7,I5\_0  
 T7,E7,P7,P8,I4\_1  
 T8,E8,P8,P9,I2\_0  
 T9,E9,P9,P10,I6\_0  
 T10,E10,P10,P11,I3\_1  
 T11,E11,P11,P12,I3\_0  
 T12,E12,P12,P3,I2\_1  
 T13,E13,P12,P13,I2\_1  
 T14,E14,P13,P3,ε  
 T15,E15,P7,P14,I4\_1  
 T16,E16,P14,P8,ε  
 T17,E17,P5,P15,I5\_1  
 T18,E18,P15,P6,ε  
 T19,E19,P10,P16,I3\_1  
 T20,E20,P16,P11,ε  
 T21,E21,P5,P17,I5\_1  
 T22,E22,P17,P6,ε  
 T23,E23,P4,P18,I0\_0  
 T24,E24,P18,P19,I7\_0  
 T25,E0,P19,P20,I0\_1  
 T26,E25,P20,P4,I7\_1  
 T27,E26,P9,P21,I0\_0  
 T28,E24,P21,P22,I7\_0  
 T29,E9,P22,P23,I6\_0  
 T30,E27,P23,P24,I3\_1  
 T31,E0,P24,P25,I0\_1  
 T32,E1,P25,P26,I7\_1  
 T33,E28,P26,P11,ε  
 T34,E26,P7,P27,I0\_0  
 T35,E24,P27,P28,I7\_0  
 T36,E29,P28,P29,I4\_1

**Tabla 3.23:** Código PN subsistema prensado Parte 4

T37,E0,P29,P30,I0_1
T38,E1,P30,P31,I7_1
T39,E30,P31,P8,ε
T40,E29,P27,P32,I4_1
T41,E24,P32,P29,I7_0
T42,E31,P30,P8,I7_1
T43,E32,P25,P11,I7_1
T44,E33,P5,P33,I5_1
T45,E34,P33,P6,ε
T46,E35,P7,P34,I4_1
T47,E36,P34,P8,ε
T48,E37,P4,P35,I1_0
T49,E26,P35,P36,I0_0
T50,E24,P36,P37,I7_0
T51,E0,P37,P38,I0_1
T52,E1,P38,P35,I7_1
T53,E2,P35,P4,I1_1
T54,E38,P5,P39,I1_0,I5_1
T55,E6,P39,P40,I5_0
T56,E7,P40,P41,I4_1
T57,E8,P41,P42,I2_0
T58,E9,P42,P43,I6_0
T59,E19,P43,P44,I3_1
T60,E20,P44,P45,ε
T61,E11,P45,P46,I3_0
T62,E13,P46,P47,I2_1
T63,E39,P47,P2,ε
T64,E40,P12,P48,I2_1
T65,E41,P48,P3,ε
T66,E26,P6,P49,I0_0
T67,E6,P49,P27,I5_0
T68,E26,P10,P50,I0_0
T69,E27,P50,P51,I3_1
T70,E24,P51,P24,I7_0
T71,E10,P43,P45,I3_1
T72,E42,P46,P2,I2_1
T73,E43,P4,P52,I1_0,I6_1
T74,E26,P52,P53,I0_0
T75,E24,P53,P54,I7_0
T76,E0,P54,P55,I0_1
T77,E1,P55,P52,I7_1

**Tabla 3.24:** Código PN subsistema prensado Parte 5

---

T78,E2,P52,P5,I1_1
T79,E44,P3,P35,I1_0,I4_0
T80,E45,P10,P56,I3_1
T81,E46,P56,P11, $\epsilon$
T82,E26,P12,P57,I0_0
T83,E42,P57,P58,I2_1
T84,E24,P58,P59,I7_0
T85,E0,P59,P60,I0_1
T86,E25,P60,P3,I7_1
T87,E26,P2,P61,I0_0
T88,E24,P61,P62,I7_0

**Tabla 3.25:** Código PN subsistema prensado Parte 6

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_4\_Prensado.ipn» observable en la Tabla 3.20, Tabla 3.21, Tabla 3.22, Tabla 3.23, Tabla 3.24, Tabla 3.25 en este se guardan los nombres de cada señal, la señal abreviada, el conjunto de eventos únicos observados, el conjunto de lugares y las transiciones tardando 56,5387 mili-segundos, también en esta parte es donde se genera la gráfica automática de la PN identificada, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz , en el modelo graficado el lugar inicial esta determinado por color azul.

3.2.5. Subsistema girado

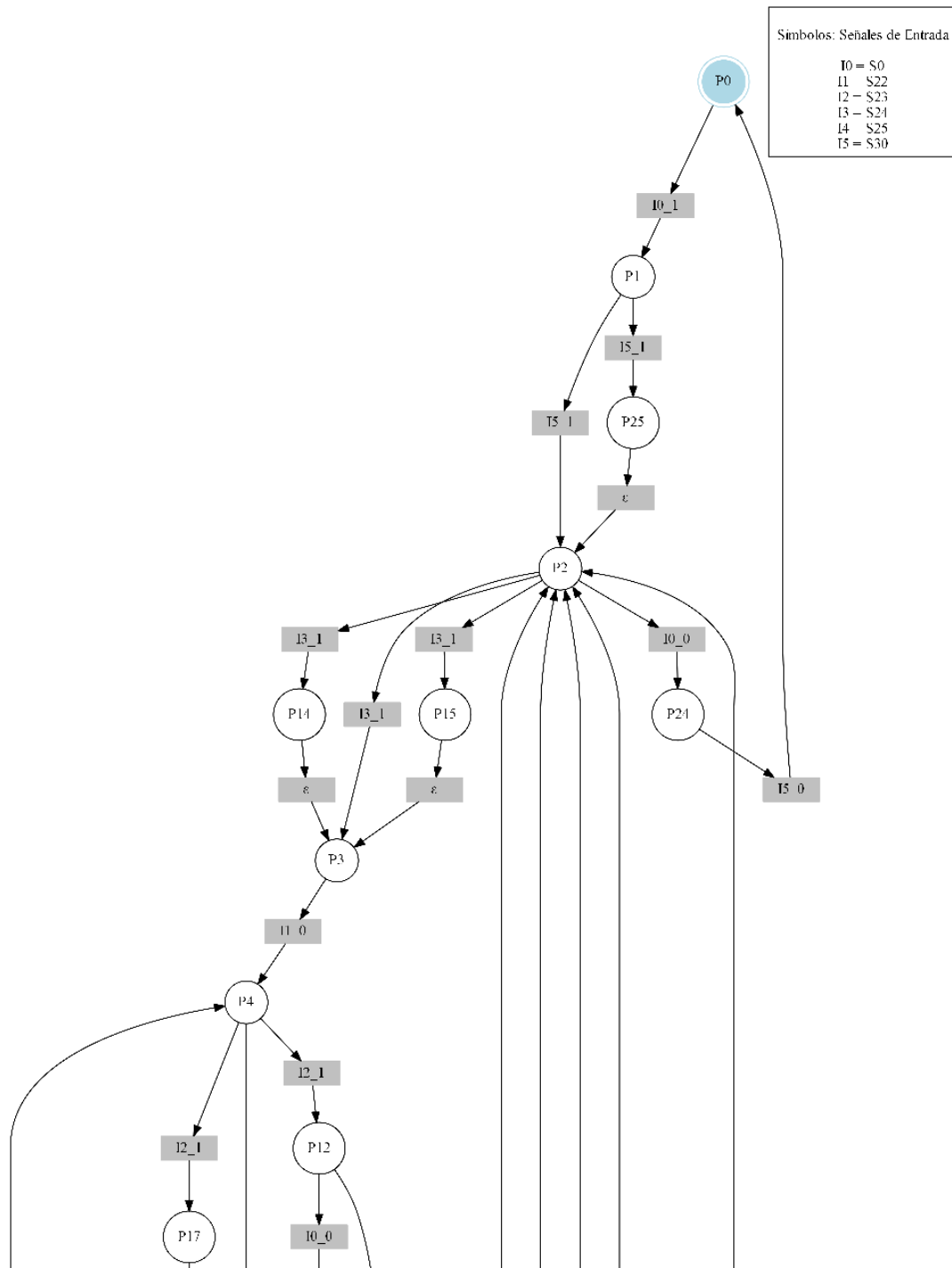


Figura 3.22: PN identificada subsistema girado Parte 1

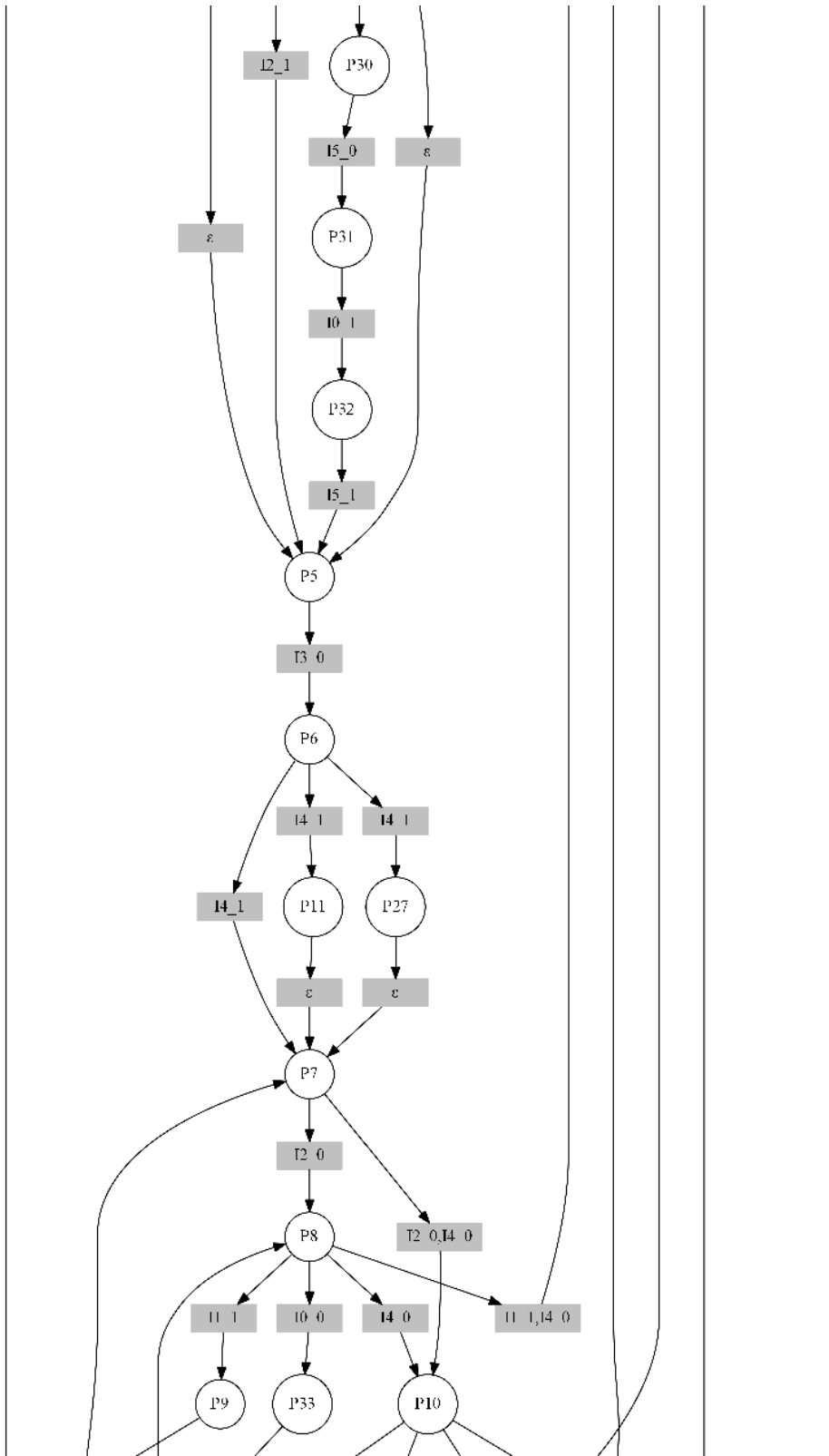


Figura 3.23: PN identificada subsistema girado Parte 2

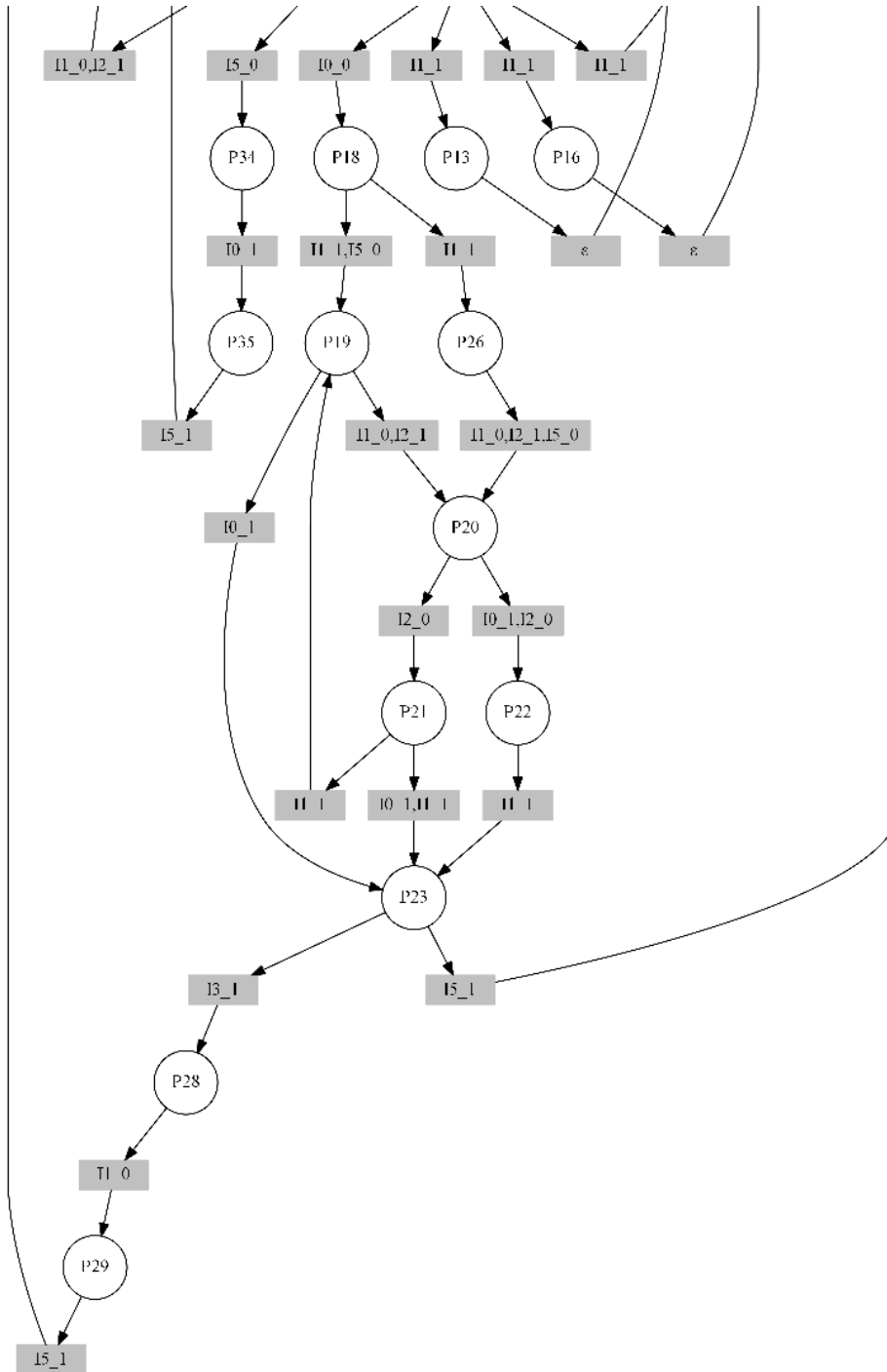


Figura 3.24: PN identificada subsistema girado Parte 3



En la Figura 3.22, Figura 3.23, Figura 3.24 se aprecia el modelo de la PN identificada del subsistema girado, el cual describe una estructura mas elaborada de lugares y transiciones donde se representa la dinámica de comportamiento logrando observar el sistema de manera mas completa ya que en este se procesan todas las observaciones. La identificación se realizó en tres partes, la primera parte de generación de eventos con un total de 13352 observaciones se tardó 7654,3947 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 62,5075 mili-segundos, reduciendo el número de eventos a 43, obteniendo 36 lugares y 57 transiciones.

S0,S22,S23,S24,S25,S30,Banda5,Gira_Izq,Gira_Der,Banda4_Iqz,Banda4_Der, I0,I1,I2,I3,I4,I5,O0,O1,O2,O3,O4,O5 M0,P0 E0,1,0,0,0,0,0,0,0,0,0,0,0,I0_1   ε E1,0,0,0,0,0,1,1,0,0,0,1,I5_1   O0_1,O4_1 E2,0,0,0,1,0,0,0,1,0,0,-1,I3_1   O1_1,O4_0 E3,0,-1,0,0,0,0,0,0,0,0,0,I1_0   ε E4,0,0,1,0,0,0,0,-1,0,1,0,I2_1   O1_0,O3_1 E5,0,0,0,-1,0,0,0,0,0,0,0,I3_0   ε E6,0,0,0,0,1,0,0,0,1,-1,0,I4_1   O2_1,O3_0 E7,0,0,-1,0,0,0,0,0,0,0,0,I2_0   ε E8,0,1,0,0,0,0,0,0,0,0,0,I1_1   ε E9,0,-1,1,0,0,0,0,0,0,0,0,I1_0,I2_1   ε E10,0,0,-1,0,-1,0,0,0,0,0,0,I2_0,I4_0   ε E11,0,1,0,0,0,0,0,0,-1,0,1,I1_1   O2_0,O4_1 E12,0,0,0,0,1,0,0,0,0,0,0,I4_1   ε E13,0,0,0,0,0,0,0,0,1,-1,0,ε   O2_1,O3_0 E14,0,0,1,0,0,0,0,0,0,0,0,I2_1   ε E15,0,0,0,0,0,0,0,-1,0,1,0,ε   O1_0,O3_1 E16,0,0,0,0,0,0,0,0,-1,0,1,ε   O2_0,O4_1 E17,0,0,0,1,0,0,0,0,0,0,0,I3_1   ε E18,0,0,0,0,0,0,0,1,0,0,-1,ε   O1_1,O4_0 E19,0,0,0,1,0,0,0,1,0,0,0,I3_1   O1_1 E20,0,0,0,0,0,0,0,0,0,0,-1,ε   O4_0 E21,0,0,0,0,-1,0,0,0,0,0,0,I4_0   ε E22,0,1,0,0,0,0,0,0,-1,0,0,I1_1   O2_0 E23,0,0,0,0,0,0,0,0,0,0,1,ε   O4_1 E24,0,0,1,0,0,0,0,-1,0,0,0,I2_1   O1_0 E25,0,0,0,0,0,0,0,0,0,1,0,ε   O3_1
---

**Tabla 3.26:** Código PN subsistema girado Parte 1

E26,-1,0,0,0,0,0,0,0,0,0,I0_0   ε
E27,0,1,0,0,0,-1,0,0,0,0,I1_1,I5_0   ε
E28,1,0,-1,0,0,0,0,0,0,0,I0_1,I2_0   ε
E29,0,0,0,0,1,0,0,0,0,I5_1   ε
E30,-1,0,0,0,0,-1,0,0,0,-1,I0_0   O0_0,O4_0
E31,0,0,0,0,-1,0,0,0,0,I5_0   ε
E32,0,0,0,0,1,0,0,0,1,ε   O0_1,O4_1
E33,0,-1,1,0,0,-1,0,0,0,0,I1_0,I2_1,I5_0   ε
E34,1,0,0,0,0,0,0,-1,0,1,I0_1   O2_0,O4_1
E35,1,1,0,0,0,0,0,-1,0,1,I0_1,I1_1   O2_0,O4_1
E36,0,0,0,1,0,0,0,1,0,0,I4_1   O2_1
E37,0,0,0,0,0,0,0,-1,0,ε   O3_0
E38,0,1,0,0,-1,0,0,0,-1,0,1,I1_1,I4_0   O2_0,O4_1
E39,-1,0,0,0,0,-1,-1,0,0,0,I0_0   O0_0,O1_0
E40,0,0,0,0,1,1,0,0,1,0,I5_1   O0_1,O3_1
E41,-1,0,0,0,0,-1,0,-1,0,0,I0_0   O0_0,O2_0
E42,0,0,0,0,1,1,0,1,0,0,I5_1   O0_1,O2_1
P0,0100000000,T36   T0
P1,1100000000,T0   T1,T37
P2,11000110001,T1,T11,T17,T24,T34,T38,T48   T2,T18,T20,T35
P3,11010111000,T2,T19,T21   T3
P4,10010111000,T3,T47   T4,T14,T25
P5,10110110010,T4,T15,T26,T52   T5
P6,10100110010,T5   T6,T12,T43
P7,10101110100,T6,T13,T9,T44   T7,T10
P8,10001110100,T7,T56   T8,T22,T48,T53
P9,11001110100,T8   T9
P10,10000110100,T10,T22   T11,T16,T23,T27
P11,10101110010,T12   T13
P12,10110111000,T14   T15,T49
P13,11000110100,T16   T17
P14,11010110001,T18   T19
P15,11010111001,T20   T21
P16,11000110000,T23   T24
P17,10110110000,T25   T26
P18,00000110100,T27   T28,T39
P19,01000010100,T28,T31   T29,T41
P20,00100010100,T29,T40   T30,T32
P21,00000010100,T30   T31,T42
P22,10000010100,T32   T33
P23,11000010001,T33,T41,T42   T34,T45
P24,01000100000,T35   T36

**Tabla 3.27:** Código PN subsistema girado Parte 2

P25,11000100000,T37		T38
P26,01000110100,T39		T40
P27,10101110110,T43		T44
P28,11010011000,T45		T46
P29,10010011000,T46		T47
P30,00110100000,T49		T50
P31,00110000000,T50		T51
P32,10110000000,T51		T52
P33,00001100000,T53		T54
P34,00001000000,T54		T55
P35,10001000000,T55		T56
T0,E0,P0,P1,I0_1		
T1,E1,P1,P2,I5_1		
T2,E2,P2,P3,I3_1		
T3,E3,P3,P4,I1_0		
T4,E4,P4,P5,I2_1		
T5,E5,P5,P6,I3_0		
T6,E6,P6,P7,I4_1		
T7,E7,P7,P8,I2_0		
T8,E8,P8,P9,I1_1		
T9,E9,P9,P7,I1_0,I2_1		
T10,E10,P7,P10,I2_0,I4_0		
T11,E11,P10,P2,I1_1		
T12,E12,P6,P11,I4_1		
T13,E13,P11,P7, $\epsilon$		
T14,E14,P4,P12,I2_1		
T15,E15,P12,P5, $\epsilon$		
T16,E8,P10,P13,I1_1		
T17,E16,P13,P2, $\epsilon$		
T18,E17,P2,P14,I3_1		
T19,E18,P14,P3, $\epsilon$		
T20,E19,P2,P15,I3_1		
T21,E20,P15,P3, $\epsilon$		
T22,E21,P8,P10,I4_0		
T23,E22,P10,P16,I1_1		
T24,E23,P16,P2, $\epsilon$		
T25,E24,P4,P17,I2_1		
T26,E25,P17,P5, $\epsilon$		
T27,E26,P10,P18,I0_0		
T28,E27,P18,P19,I1_1,I5_0		

**Tabla 3.28:** Código PN subsistema girado Parte 3

T29,E9,P19,P20,I1_0,I2_1
T30,E7,P20,P21,I2_0
T31,E8,P21,P19,I1_1
T32,E28,P20,P22,I0_1,I2_0
T33,E11,P22,P23,I1_1
T34,E29,P23,P2,I5_1
T35,E30,P2,P24,I0_0
T36,E31,P24,P0,I5_0
T37,E29,P1,P25,I5_1
T38,E32,P25,P2, $\epsilon$
T39,E8,P18,P26,I1_1
T40,E33,P26,P20,I1_0,I2_1,I5_0
T41,E34,P19,P23,I0_1
T42,E35,P21,P23,I0_1,I1_1
T43,E36,P6,P27,I4_1
T44,E37,P27,P7, $\epsilon$
T45,E2,P23,P28,I3_1
T46,E3,P28,P29,I1_0
T47,E29,P29,P4,I5_1
T48,E38,P8,P2,I1_1,I4_0
T49,E39,P12,P30,I0_0
T50,E31,P30,P31,I5_0
T51,E0,P31,P32,I0_1
T52,E40,P32,P5,I5_1
T53,E41,P8,P33,I0_0
T54,E31,P33,P34,I5_0
T55,E0,P34,P35,I0_1
T56,E42,P35,P8,I5_1

**Tabla 3.29:** Código PN subsistema girado Parte 4

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_5\_Gira.ipn» observable en la Tabla 3.26, Tabla 3.27, Tabla 3.28, Tabla 3.29 en este se guardan los nombres de cada señal, la señal abreviada, el conjunto de eventos únicos observados, el conjunto de lugares y las transiciones tardando 28,0257 mili-segundos, también en esta parte es donde se genera la gráfica automática de la PN identificada, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz , en el modelo graficado el lugar inicial esta determinado por color azul.

### 3.2.6. Subsistema secado por luz UV

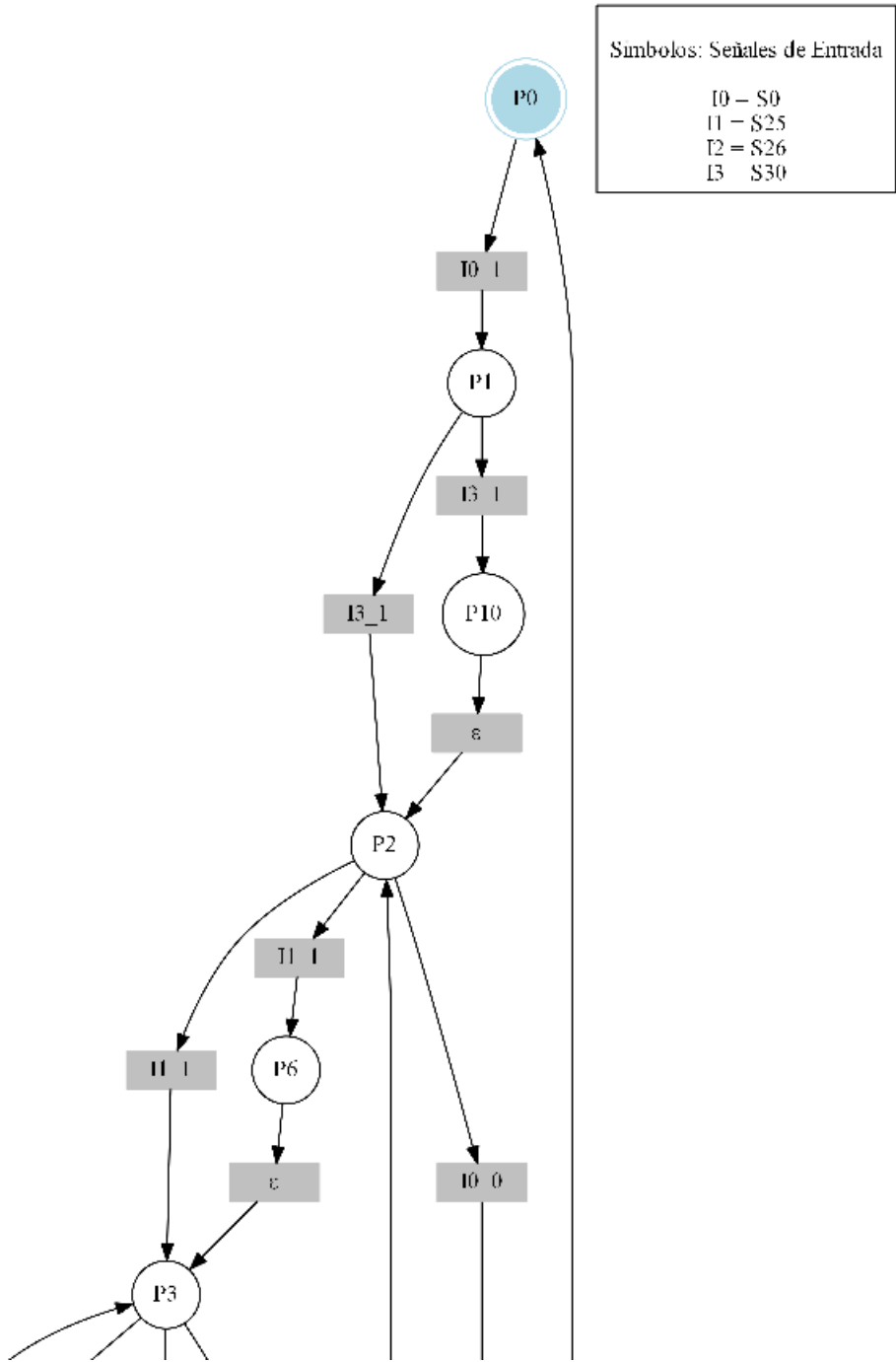


Figura 3.25: PN identificada subsistema secado por luz UV Parte 1

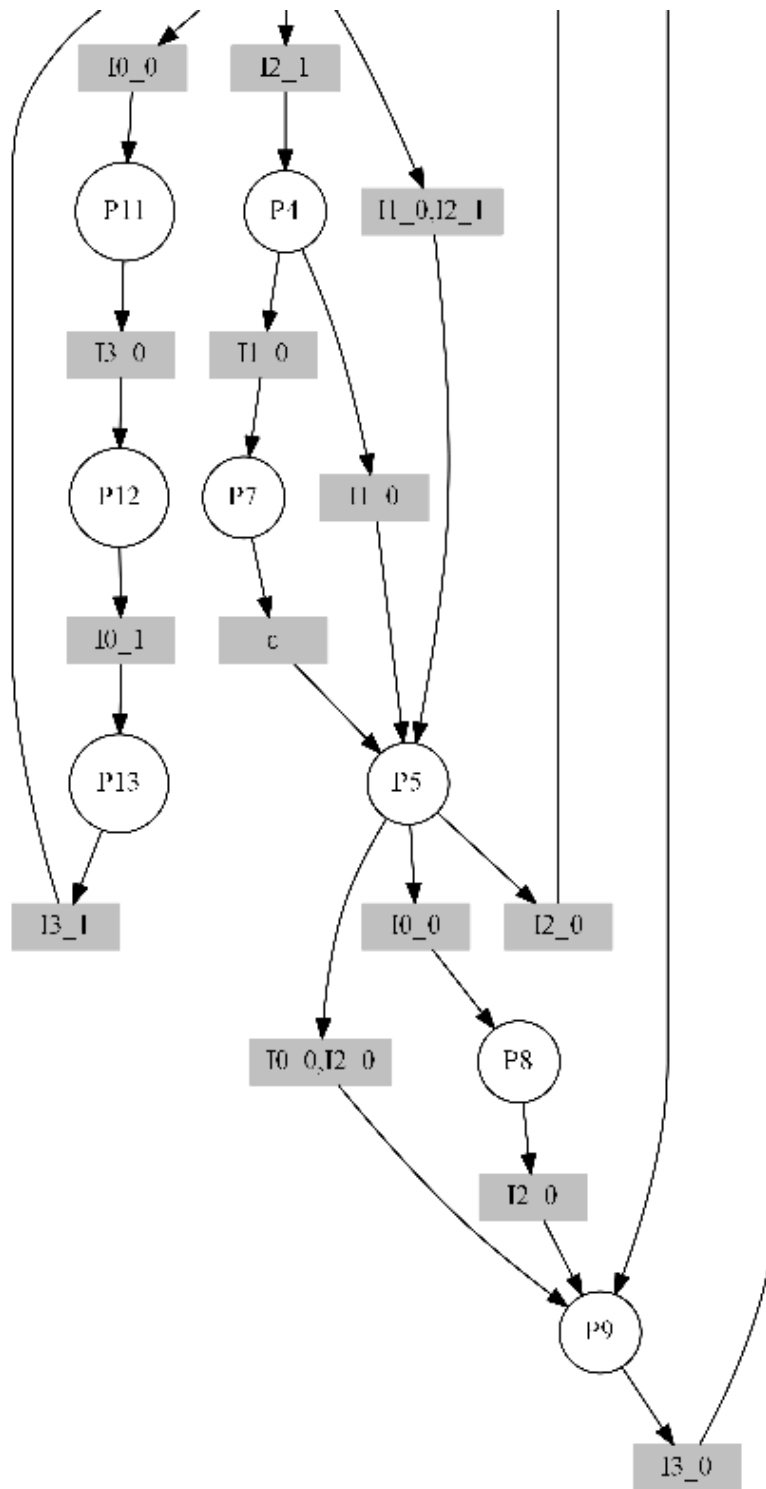


Figura 3.26: PN identificada subsistema secado por luz UV Parte 2

En la Figura 3.25, Figura 3.26 se aprecia el modelo de la PN identificada del subsistema secado por luz UV, el cual describe una estructura mas elaborada de lugares y transiciones donde se representa la dinámica de comportamiento logrando observar el sistema de manera mas completa ya que en este se procesan todas las observaciones. La identificación se realizó en tres partes, la primera parte de generación de eventos con un total de 5379 observaciones se tardó 3999,2052 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 51,5348 mili-segundos, reduciendo el número de eventos a 20, obteniendo 14 lugares y 22 transiciones.

S0,S25,S26,S30,Lampara,Banda6,Bomba3,Valvula3, I0,I1,I2,I3,O0,O1,O2,O3,O4 M0,P0 E0,1,0,0,0,0,0,0,I0_1   ε E1,0,0,0,1,0,1,1,I3_1   O1_1,O2_1,O3_1 E2,0,1,0,0,1,0,0,I1_1   O0_1 E3,0,0,1,0,0,0,0,I2_1   ε E4,0,-1,0,0,-1,0,0,I1_0   O0_0 E5,0,0,-1,0,0,0,0,I2_0   ε E6,0,1,0,0,0,0,0,I1_1   ε E7,0,0,0,0,1,0,0,ε   O0_1 E8,0,-1,0,0,0,0,0,I1_0   ε E9,0,0,0,0,-1,0,0,ε   O0_0 E10,-1,0,0,0,0,0,0,I0_0   ε E11,0,0,-1,0,0,-1,-1,-1,I2_0   O1_0,O2_0,O3_0 E12,0,0,0,-1,0,0,0,I3_0   ε E13,-1,0,0,0,-1,-1,-1,I0_0   O1_0,O2_0,O3_0 E14,0,0,0,1,0,0,0,I3_1   ε E15,0,0,0,0,1,1,1,ε   O1_1,O2_1,O3_1 E16,-1,0,-1,0,0,-1,-1,-1,I0_0,I2_0   O1_0,O2_0,O3_0 E17,0,-1,1,0,-1,0,0,I1_0,I2_1   O0_0 E18,-1,0,0,0,-1,-1,-1,-1,I0_0   O0_0,O1_0,O2_0,O3_0 E19,0,0,0,1,1,1,1,I3_1   O0_1,O1_1,O2_1,O3_1 P0,00000000,T12   T0 P1,10000000,T0   T1,T14 P2,10010111,T1,T5,T15   T2,T6,T13 P3,11011111,T2,T7,T21   T3,T17,T18 P4,11111111,T3   T4,T8 P5,10110111,T4,T9,T17   T5,T10,T16
---

**Tabla 3.30:** Código PN subsistema secado por luz UV Parte 1

```

P6,11010111,T6|||T7
P7,10111111,T8|||T9
P8,00110111,T10|||T11
P9,00010000,T11,T13,T16|||T12
P10,10010000,T14|||T15
P11,01010000,T18|||T19
P12,01000000,T19|||T20
P13,11000000,T20|||T21
T0,E0,P0,P1,I0_1
T1,E1,P1,P2,I3_1
T2,E2,P2,P3,I1_1
T3,E3,P3,P4,I2_1
T4,E4,P4,P5,I1_0
T5,E5,P5,P2,I2_0
T6,E6,P2,P6,I1_1
T7,E7,P6,P3, $\varepsilon$ 
T8,E8,P4,P7,I1_0
T9,E9,P7,P5, $\varepsilon$ 
T10,E10,P5,P8,I0_0
T11,E11,P8,P9,I2_0
T12,E12,P9,P0,I3_0
T13,E13,P2,P9,I0_0
T14,E14,P1,P10,I3_1
T15,E15,P10,P2, $\varepsilon$ 
T16,E16,P5,P9,I0_0,I2_0
T17,E17,P3,P5,I1_0,I2_1
T18,E18,P3,P11,I0_0
T19,E12,P11,P12,I3_0
T20,E0,P12,P13,I0_1
T21,E19,P13,P3,I3_1

```

**Tabla 3.31:** Código PN subsistema secado por luz UV Parte 2

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_6\_Secado2.ipn» observable en la Tabla 3.30, Tabla 3.31 en este se guardan los nombres de cada señal, la señal abreviada, el conjunto de eventos únicos observados, el conjunto de lugares y las transiciones tardando 27,5191 mili-segundos, también en esta parte es donde se genera la gráfica automática de la PN identificada, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz , en el modelo graficado el lugar inicial esta determinado por color azul.



### 3.2.7. Subsistema impresión

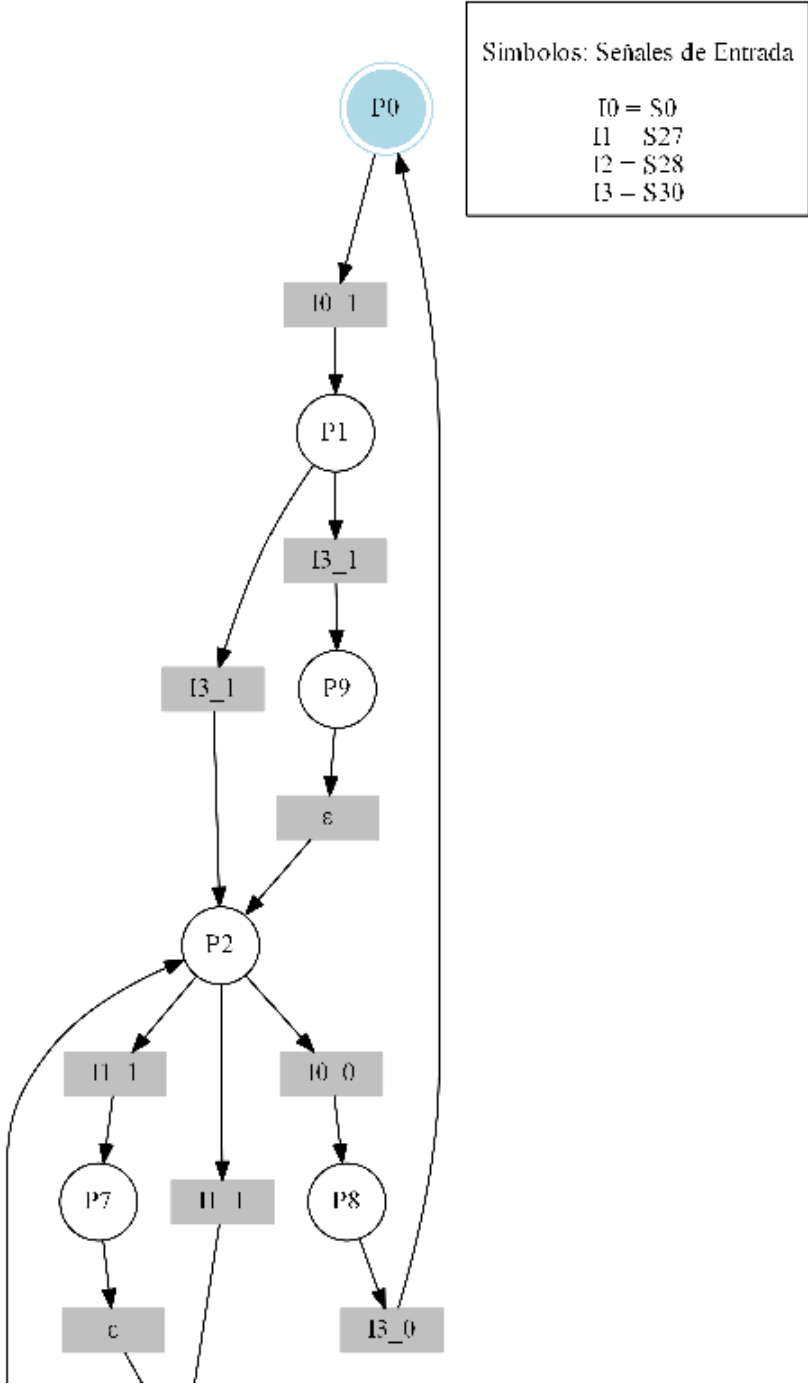


Figura 3.27: PN identificada subsistema impresión Parte 1

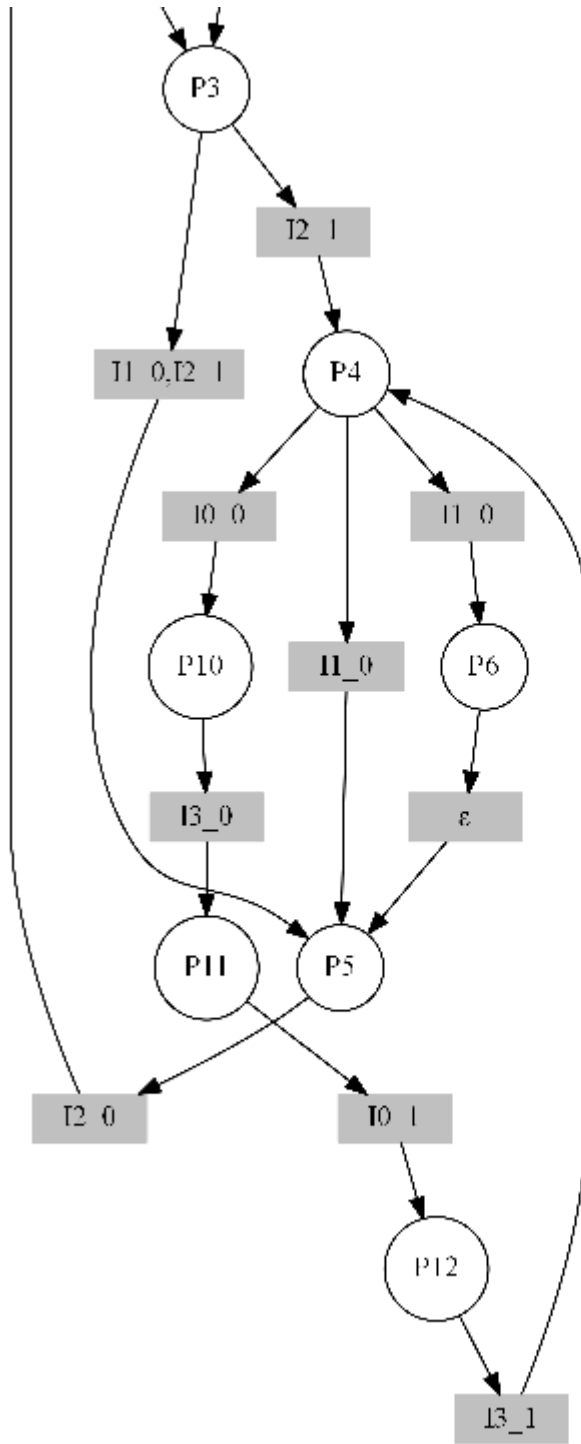


Figura 3.28: PN identificada subsistema impresión Parte 2

En la Figura 3.27, Figura 3.28 se aprecia el modelo de la PN identificada del subsistema impresión, el cual describe una estructura mas elaborada de lugares y transiciones donde se representa la dinámica de comportamiento logrando observar el sistema de manera mas completa ya que en este se procesan todas las observaciones. La identificación se realizó en tres partes, la primera parte de generación de eventos con un total de 5363 observaciones se tardó 3995,4427 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 38,5275 mili-segundos, reduciendo el número de eventos a 17, obteniendo 13 lugares y 19 transiciones.

S0,S27,S28,S30,Banda7,Impresora,Valvula4,Bomba4, I0,I1,I2,I3,O0,O1,O2,O3,O4 M0,P0 E0,1,0,0,0,0,0,0,I0_1  ε E1,0,0,0,1,1,0,1,I3_1  O0_1,O2_1,O3_1 E2,0,1,0,0,0,1,0,I1_1  O1_1 E3,0,0,1,0,0,0,0,I2_1  ε E4,0,-1,0,0,0,-1,0,I1_0  O1_0 E5,0,0,-1,0,0,0,0,I2_0  ε E6,0,-1,0,0,0,0,0,I1_0  ε E7,0,0,0,0,-1,0,0,ε  O1_0 E8,0,1,0,0,0,0,0,I1_1  ε E9,0,0,0,0,1,0,0,ε  O1_1 E10,-1,0,0,-1,0,-1,-1,I0_0  O0_0,O2_0,O3_0 E11,0,0,0,-1,0,0,0,I3_0  ε E12,0,0,0,1,0,0,0,I3_1  ε E13,0,0,0,1,0,1,1,ε  O0_1,O2_1,O3_1 E14,-1,0,0,-1,-1,-1,-1,I0_0  O0_0,O1_0,O2_0,O3_0 E15,0,0,0,1,1,1,1,I3_1  O0_1,O1_1,O2_1,O3_1 E16,0,-1,1,0,0,-1,0,I1_0,I2_1  O1_0 P0,0000000,T11  T0 P1,10000000,T0  T1,T12 P2,10011011,T1,T5,T13  T2,T8,T10 P3,11011111,T2,T9  T3,T18 P4,11111111,T3,T17  T4,T6,T14 P5,10111011,T4,T7,T18  T5 P6,10111111,T6  T7 P7,11011011,T8  T9 P8,00010000,T10  T11
--

**Tabla 3.32:** Código PN subsistema impresión Parte 1

```

P9,10010000,T12|||T13
P10,01110000,T14|||T15
P11,01100000,T15|||T16
P12,11100000,T16|||T17
T0,E0,P0,P1,I0_1
T1,E1,P1,P2,I3_1
T2,E2,P2,P3,I1_1
T3,E3,P3,P4,I2_1
T4,E4,P4,P5,I1_0
T5,E5,P5,P2,I2_0
T6,E6,P4,P6,I1_0
T7,E7,P6,P5, $\varepsilon$ 
T8,E8,P2,P7,I1_1
T9,E9,P7,P3, $\varepsilon$ 
T10,E10,P2,P8,I0_0
T11,E11,P8,P0,I3_0
T12,E12,P1,P9,I3_1
T13,E13,P9,P2, $\varepsilon$ 
T14,E14,P4,P10,I0_0
T15,E11,P10,P11,I3_0
T16,E0,P11,P12,I0_1
T17,E15,P12,P4,I3_1
T18,E16,P3,P5,I1_0,I2_1

```

**Tabla 3.33:** Código PN subsistema impresión Parte 2

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_7\_Impresion.ipn» observable en la Tabla 3.32, Tabla 3.33 en este se guardan los nombres de cada señal, la señal abreviada, el conjunto de eventos únicos observados, el conjunto de lugares y las transiciones tardando 29,0205 mili-segundos, también en esta parte es donde se genera la gráfica automática de la PN identificada, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz, en el modelo graficado el lugar inicial esta determinado por color azul.

3.2.8. Subsistema horneado

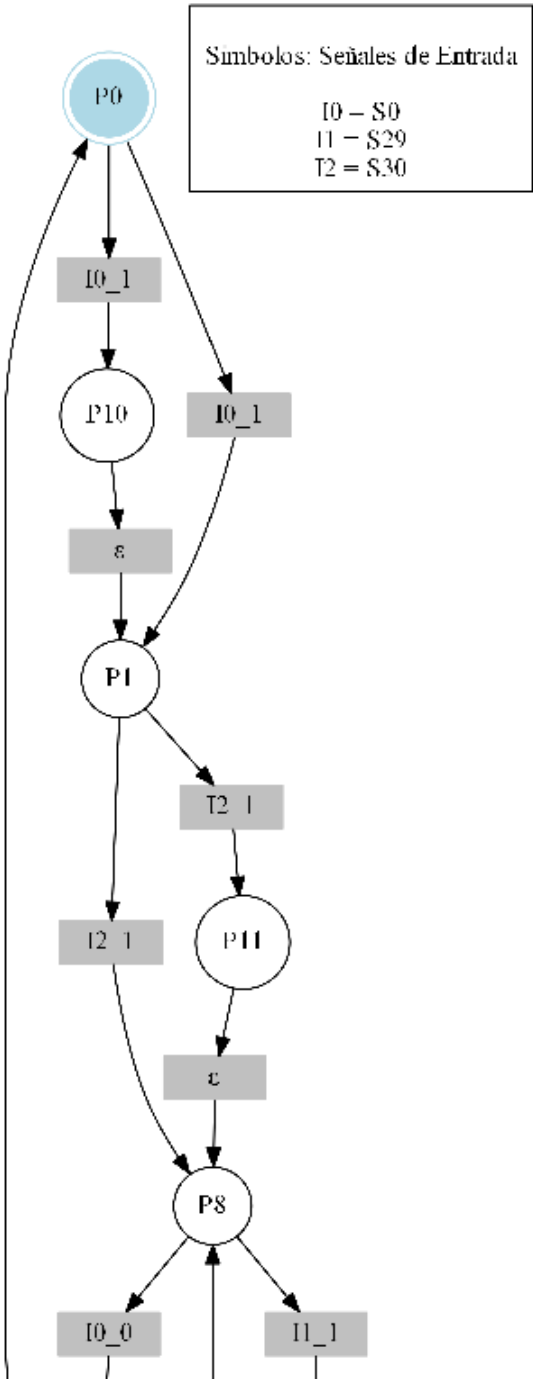


Figura 3.29: PN identificada subsistema horneado Parte 1

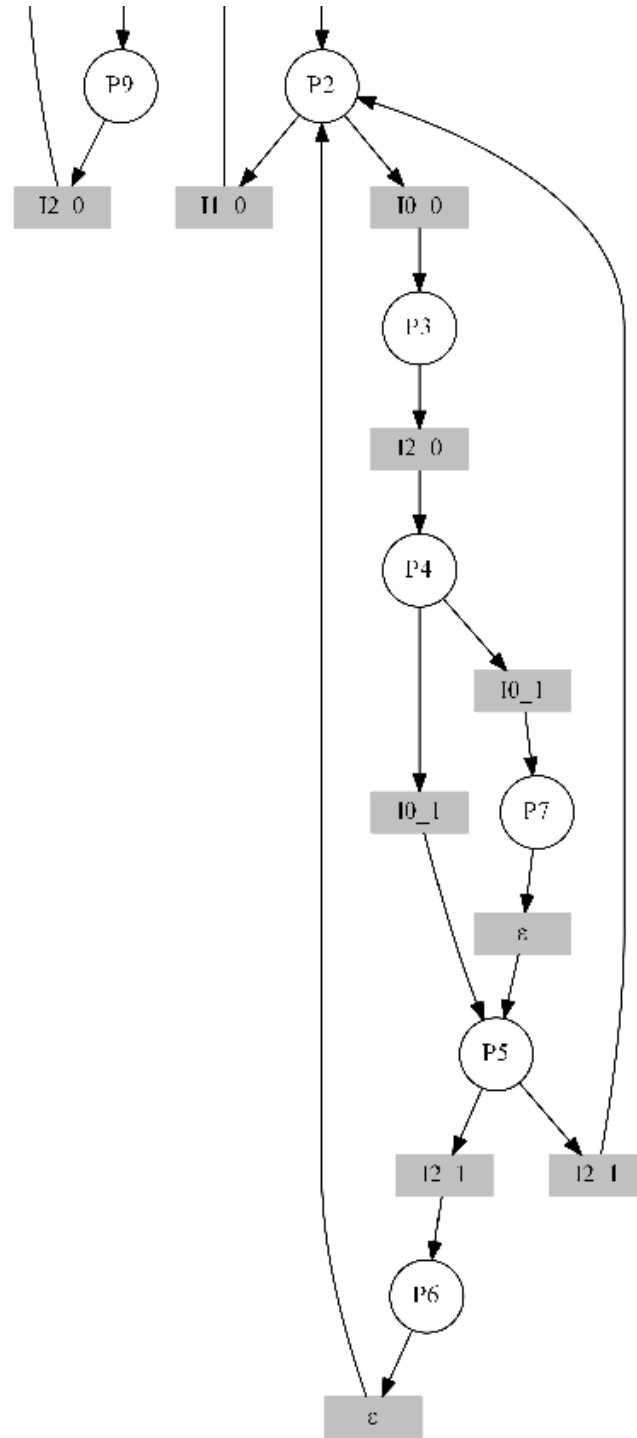


Figura 3.30: PN identificada subsistema horneado Parte 2

En la Figura 3.29, Figura 3.30 se aprecia el modelo de la PN identificada del subsistema horneado, el cual describe una estructura mas elaborada de lugares y transiciones donde se representa la dinámica de comportamiento logrando observar el sistema de manera mas completa ya que en este se procesan todas las observaciones. La identificación se realizó en tres partes, la primera parte de generación de eventos con un total de 134 observaciones se tardó 78,1109 mili-segundos, la segunda parte que consiste en la identificación del subsistema se tardó 31,2525 mili-segundos, reduciendo el número de eventos a 14, obteniendo 12 lugares y 18 transiciones.

S0,S29,S30,Banda8,Quemador2,Quemador3,
I0,I1,I2,O0,O1,O2,O3
M0,P0
E0,1,0,0,0,1,0,I0_1   O1_1
E1,0,0,1,1,0,0,I2_1   O0_1
E2,0,1,0,0,0,0,I1_1   ε
E3,-1,0,0,-1,-1,0,I0_0   O0_0,O1_0
E4,0,0,-1,0,0,0,I2_0   ε
E5,1,0,0,0,1,1,I0_1   O1_1,O2_1
E6,0,0,1,1,0,-1,I2_1   O0_1,O2_0
E7,0,0,1,0,0,0,I2_1   ε
E8,0,0,0,1,0,-1,ε   O0_1,O2_0
E9,1,0,0,0,0,0,I0_1   ε
E10,0,0,0,0,1,1,ε   O1_1,O2_1
E11,0,-1,0,0,0,0,I1_0   ε
E12,0,0,0,0,1,0,ε   O1_1
E13,0,0,0,1,0,0,ε   O0_1
P0,000000,T13   T0,T14
P1,100010,T0,T15   T1,T16
P2,111110,T2,T6,T8   T3,T11
P3,011000,T3   T4
P4,010000,T4   T5,T9
P5,110011,T5,T10   T6,T7
P6,111011,T7   T8
P7,110000,T9   T10
P8,101110,T11,T1,T17   T12,T2
P9,001000,T12   T13
P10,100000,T14   T15
P11,101010,T16   T17

**Tabla 3.34:** Código PN subsistema horneado Parte 1

T0,E0,P0,P1,I0_1
T1,E1,P1,P8,I2_1
T2,E2,P8,P2,I1_1
T3,E3,P2,P3,I0_0
T4,E4,P3,P4,I2_0
T5,E5,P4,P5,I0_1
T6,E6,P5,P2,I2_1
T7,E7,P5,P6,I2_1
T8,E8,P6,P2, $\varepsilon$
T9,E9,P4,P7,I0_1
T10,E10,P7,P5, $\varepsilon$
T11,E11,P2,P8,I1_0
T12,E3,P8,P9,I0_0
T13,E4,P9,P0,I2_0
T14,E9,P0,P10,I0_1
T15,E12,P10,P1, $\varepsilon$
T16,E7,P1,P11,I2_1
T17,E13,P11,P8, $\varepsilon$

**Tabla 3.35:** Código PN subsistema horneado Parte 2

La última parte del proceso es mediante el sistema identificado en memoria el cual es guardado en un fichero bajo el nombre «\_8\_Horneado.ipn» observable en la Tabla 3.34, Tabla 3.35 en este se guardan los nombres de cada señal, la señal abreviada, el conjunto de eventos únicos observados, el conjunto de lugares y las transiciones tardando 15,6252 mili-segundos, también en esta parte es donde se genera la gráfica automática de la PN identificada, esto suele tardar algunos segundos debido a que depende del procesamiento externo de la aplicación Graphviz, en el modelo graficado el lugar inicial esta determinado por color azul.