
PROYECTO 2 – ENSAMBLE DE PRODUCTOS – DIGITAL INTELLIGENCE S.A.

201701548 – Arnoldo Luis Antonio González Camey

Resumen

El proyecto consiste en la realización de un algoritmo para el desarrollo de una máquina capaz de ensamblar las partes de cualquier producto. La máquina consta de n líneas de ensamblaje y funciona de forma que se define un producto a ensamblar y se le proporciona un conjunto de instrucciones indicando la línea de producción y el componente a ensamblar.

El algoritmo debe ser capaz de ensamblar los productos en el menor tiempo posible; este debe ser planteado de forma lógica y aplicando los conceptos de la programación orientada a objetos, tipo de dato abstracto y la implementación de la memoria dinámica.

Para el desarrollo del algoritmo se hace uso de listas doblemente enlazadas donde se almacenan las clases con los atributos en nodos para seguidamente recorrerlas y mediante validaciones ensamblar los componentes requeridos tanto por medio de un archivo de simulación como por la interfaz gráfica. A su vez se realiza el reporte de la elaboración del producto en un archivo XML y una gráfica detallando el proceso y la matriz dispersa en Graphviz.

Palabras clave

Algoritmo – TDA – POO – XML – Nodo

Abstract

The project consists in the realization of an algorithm for the development of a machine capable of assembling the parts of any product. The machine consists of n assembly lines and works by defining a product to be assembled and provides a series of instructions indicating the production line and the component to be assembled.

The algorithm must be able to assemble the products in the shortest possible time; the algorithm must be logically proposed and applying the concepts of objects oriented programming, using the concept of abstract data types and the implementation of the dynamic memory.

A doubly linked list was used in which the classes containing the necessary attributes are stored in the nodes, then the nodes are traversed and by some validations assembly the required components, using a simulation file and the graphical user interface. Finally, a report of the elaboration of the product is made in an XML file and a graph detailing the process and the sparse matrix in Graphviz.

Keywords

Algorithm – ADT – OOP – XML – Node

Introducción

El siguiente proyecto consta de la realización de un algoritmo que solventa el problema planteado en el lenguaje de programación Python, implementando los conceptos de tipo de dato abstracto, programación orientada a objetos, implementación de una matriz dispersa, el uso de XML y la librería Graphviz. El algoritmo consiste en el desarrollo de una máquina capaz de ensamblar las partes de cualquier producto con n cantidad de líneas, indicándole la línea de producción y el componente a ensamblar.

Para ello se implementó una lista doblemente enlazada en donde se guarda una clase, que contiene los atributos de las líneas de producción y los componentes a ensamblar y estos se van almacenando en unos nodos con indicadores los cuales se recorren y luego de validaciones se obtiene el producto ensamblado en el menor tiempo posible.

Al finalizar se muestra los reportes solicitados tanto en XML como con la herramienta Graphviz.

Desarrollo del tema

La Empresa Digital Intelligence, S.A. ha desarrollado una máquina con la capacidad de ensamblar las partes de cualquier producto; para esto la máquina consta de “n” líneas de ensamblaje y un brazo robótico para cada línea. El brazo robótico demora 1 segundo en colocarse sobre el primer componente y 2 segundos sobre el segundo componente y así sucesivamente. La máquina funciona definiendo el producto a ensamblar y se le proporciona el conjunto de instrucciones indicando la línea de producción y el componente a ensamblar para fabricar el producto.

Al iniciar el programa se le muestra al usuario una interfaz gráfica en la cual debe de seleccionar la opción deseada entre cargar el archivo de la máquina, el archivo de simulación, analizar el producto y generar la gráfica de los componentes del producto, escribir el reporte en XML o la opción de ayuda.

a. Cargar archivo de Máquina

La información de las líneas de producción se encuentra en un archivo XML, en este se encuentra la cantidad de líneas de producción, el número, la cantidad de componentes y el tiempo de ensamblaje de cada una de las líneas de producción, así como el listado de productos con su respectivo nombre del producto y el proceso de elaboración de este. Si el usuario elige la opción de la interfaz cargar archivo máquina se procede a leer el archivo por medio de ElementTree el cual es una librería de Python que permite analizar y navegar por un documento XML, para ello se abre el archivo y mediante la sintaxis adecuada y un ciclo for se almacenan los datos en variables temporales para seguido almacenarlos en las respectivas clases y nodos correspondientes.

Para almacenar los datos en una lista doblemente enlazada se hace uso del siguiente algoritmo, primero que nada se crea un nuevo nodo y se guarda en la clase correspondiente, se aumenta en uno el tamaño de la lista y se valida si inicio en nulo, si es así se procede a guardar al nodo como el primero de la lista y a su vez el último de la lista, si esto no se cumple se guarda en una variable temporal la posición inicial y se procede a recorrer la lista por medio de un ciclo hasta que es sea nulo, en el momento que el nodo siguiente sea nulo acaba el ciclo y almacena al nuevo nodo como el siguiente de la posición actual y como el anterior del nuevo nodo al nodo actual y el final de

la lista se convierte en el nodo que se acaba de ingresar.

Para la lectura de la elaboración del producto se realiza mediante la siguiente una expresión regular ($L[0-9]+pC[0-9]+$) la cual primero acepta la letra “L” seguido de 1 o mas dígitos, luego de esto viene una literal de separación la cual es la letra “p”, luego la letra “C” seguido de 1 o más dígitos. Para el reconocimiento de los caracteres se procede a recorrer la cadena de caracteres y se va reconociendo cada uno de los tokens y agregándolos a una lista doblemente enlazada para luego ser almacenados en una clase para su posterior implementación en el programa.

b. Cargar archivo de Simulación

La información del archivo de los productos a simular simultáneamente se encuentra en un archivo XML, en este se encuentra el nombre de la simulación, y el listado de los nombres de los productos a simular. Si el usuario elige la opción del menú cargar archivo simulación se procede a leer el archivo por medio de ElementTree el cual es una librería de Python que permite analizar y navegar por un documento XML, para ello se abre el archivo y mediante la sintaxis adecuada y un ciclo for se almacenan los datos en variables temporales para seguido almacenarlos en las respectivas clases y nodos correspondientes.

Para almacenar los datos en una lista doblemente enlazada se hace uso del siguiente algoritmo, primero que nada se crea un nuevo nodo y se guarda en la clase correspondiente, se aumenta en uno el tamaño de la lista y se valida si inicio en nulo, si es así se procede a guardar al nodo como el primero de la lista y a su vez el último de la lista, si esto no se cumple se guarda en una variable temporal la posición inicial y

se procede a recorrer la lista por medio de un ciclo hasta que es sea nulo, en el momento que el nodo siguiente sea nulo acaba el ciclo y almacena al nuevo nodo como el siguiente de la posición actual y como el anterior del nuevo nodo al nodo actual y el final de la lista se convierte en el nodo que se acaba de ingresar.

b. Analizar Producto

Después que los datos de las líneas de producción y el listado de los productos se encuentran en los nodos correspondientes si el usuario desea ensamblar algún producto debe seleccionar de la lista el producto a procesar, luego mediante un ciclo while el cual compara el nombre seleccionado por el usuario con el nombre de los productos de la lista de elaboración se empieza a realizar la el movimiento de las líneas de ensamblaje si el brazo robótico se encuentra en el componente que debe de ensamblar se procede a guardar en una variable global el tiempo en segundos que va recorriendo dicha línea de producción. Cada vez que avanza a otro componente se almacena en la lista de líneas de producción el valor donde se encuentra actualmente el brazo robótico, se procede a sumar los movimientos y colocar en la tabla la producción del producto. Antes de ensamblar otro producto las máquinas regresan a su estado inicial.

Para insertar los datos en la matriz dispersa se debe verificar si ya existen encabezados en la matriz, si esto da nulo se procede a guardar en una variable los nuevos encabezados. Luego se procede a insertar una nueva fila mediante un ciclo while el cual recorre la totalidad de nodos internos, se evalúa si el siguiente nodo esta a la derecha o a la izquierda y se procede a guardar la posición en la matriz. Seguido de esto se procede a insertar una nueva columna mediante un

ciclo while el cual recorre la totalidad de nodos internos, se evalúa si el siguiente nodo está a la arriba o abajo del nodo actual y se procede a guardar la posición en la matriz.

Para insertar un nuevo encabezado se van insertando en orden para esto se evalúa si el primero es nulo si lo es guarda el encabezado nuevo, sino verifica si el nuevo encabezado es menor al primero si lo es lo guarda como el nuevo primer encabezado, si no lo es, verifica si es mayor al último si lo es lo guarda como el nuevo último de la lista y si no significa que el nuevo encabezado se encuentra entre el primero y ultimo encabezado, para la inserción se recorre la lista de encabezados y se evalúa si el nuevo es menor al siguiente si lo es se procede a agregar el encabezado si no lo es, pasa al siguiente encabezado y vuelve a evaluar hasta encontrar el indicado.

c. Reportes - Escribir archivo de salida

Si el usuario desea escribir el archivo de salida debe seleccionar la opción de la interfaz gráfica, al hacerlo se guarda la estructura básica de un archivo XML en variables temporales. Posteriormente se hace la instancia de un método encargado de recorrer el proceso de elaboración y guardar el nombre del producto en una variable que se retorna al finalizar dicho método. Al finalizar se unen las variables en una sola y se hace la instancia al método open y write los cuales son utilizadas para escribir un archivo.

d. Ayuda

En esta opción se crea una nueva ventana la cual cuenta con los datos del estudiante que realizó el proyecto y las instrucciones básicas de como operar el programa, para esto se hace la instancia de un

método que tiene guardada la información en una variable y retorna dicha variable.

e. Generar gráfica

Si se quiere generar la gráfica de la elaboración de un producto en particular y la gráfica de la matriz dispersa en una imagen, el usuario debe seleccionar en el programa la opción reportes, esta opción sirve para generar una imagen en formato png, para ello se hace uso de un método el cual consiste en crear la estructura básica de un archivo en Graphviz, con el tipo forma que se quiere que tengan los nodos, el color, el tipo de flecha. Primero para la matriz se van creando variables denominadas filas, tantas como filas existan en el terreno, lo mismo para las columnas, seguido de esto se enlaza la raíz previamente creada con la fila uno y la columna uno. Posteriormente se hace uso del método rank el cual consiste en alinear un nodo seguido de otro nodo, eso se realiza para la raíz y las columnas.

Luego con otro método que es el encargado de generar los nodos internos se comienza a recorrer la matriz dispersa. Se crean los nodos necesarios dependiendo las dimensiones de las líneas de producción del archivo de máquina cargado con anterioridad. al finalizar el método se retorna la variable que contiene todos los datos recopilados a lo largo del proceso. Luego se hace la instancia del método open y write los cuales sirven para escribir un archivo con extensión dot, seguido de esto se hace la instancia del método system el cual hace referencia a la consola y ejecuta un comando para convertir ese archivo dot a una imagen png, luego se abre la imagen por medio del método startfile. Para la creación de la imagen del proceso de elaboración del producto se realiza un proceso similar solo excluyendo las filas.

Conclusiones

- La utilización de listas doblemente enlazadas al principio puede ser algo confuso debido a que se debe tener muy presente en el nodo en que se encuentra, como moverse en la lista, pero son una herramienta muy útil en ciertos aspectos, ya que siempre tienes referencias al anterior y al siguiente.
- Las listas son dinámicas, no se establece cual es su dimensión al crearlo como se hace en un array, estás pueden ser de dos posiciones como de mil, esto sirve para no apartar espacio de memoria para datos que quizás nunca entren y tampoco te limitan en la cantidad de datos a ingresar
- La utilización de Graphviz como parte del reporte es una gran ayuda visual para representar la matriz dispersa, también sirve para comprender de una manera más evidente como están conectados los nodos entre sí.

Anexos

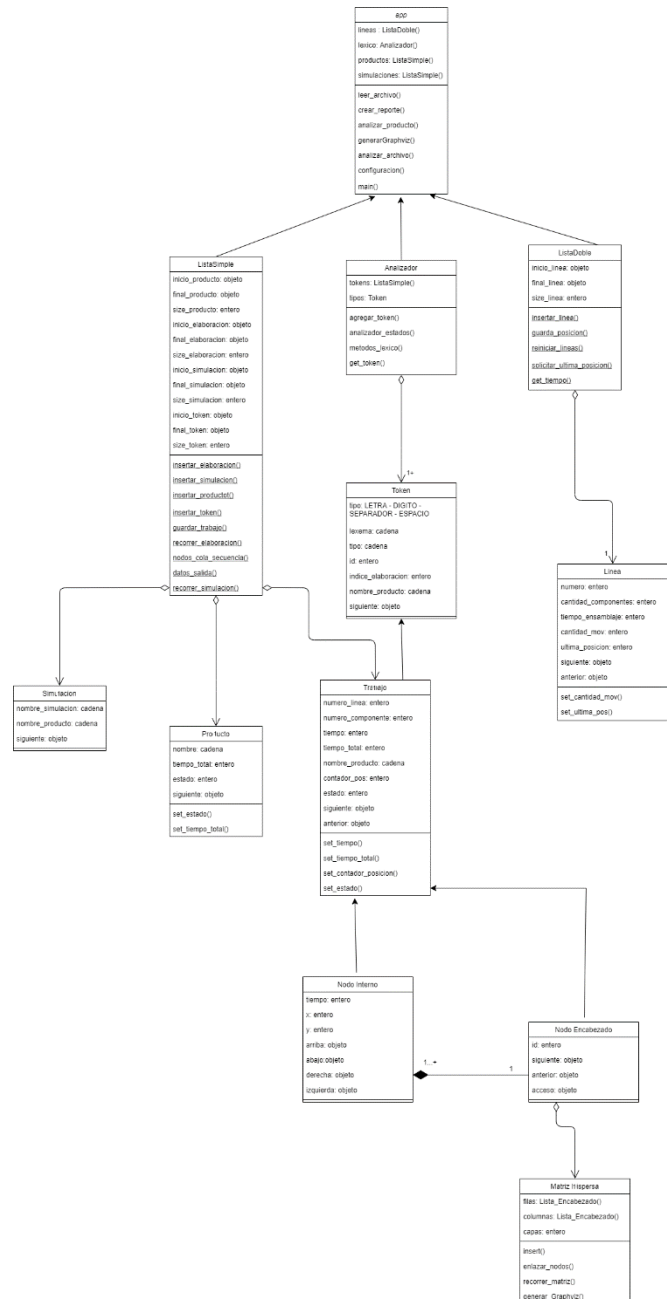


Figura 1. Diagrama de Clases

Fuente: elaboración propia



Figura 2. Interfaz gráfica

Fuente: elaboración propia

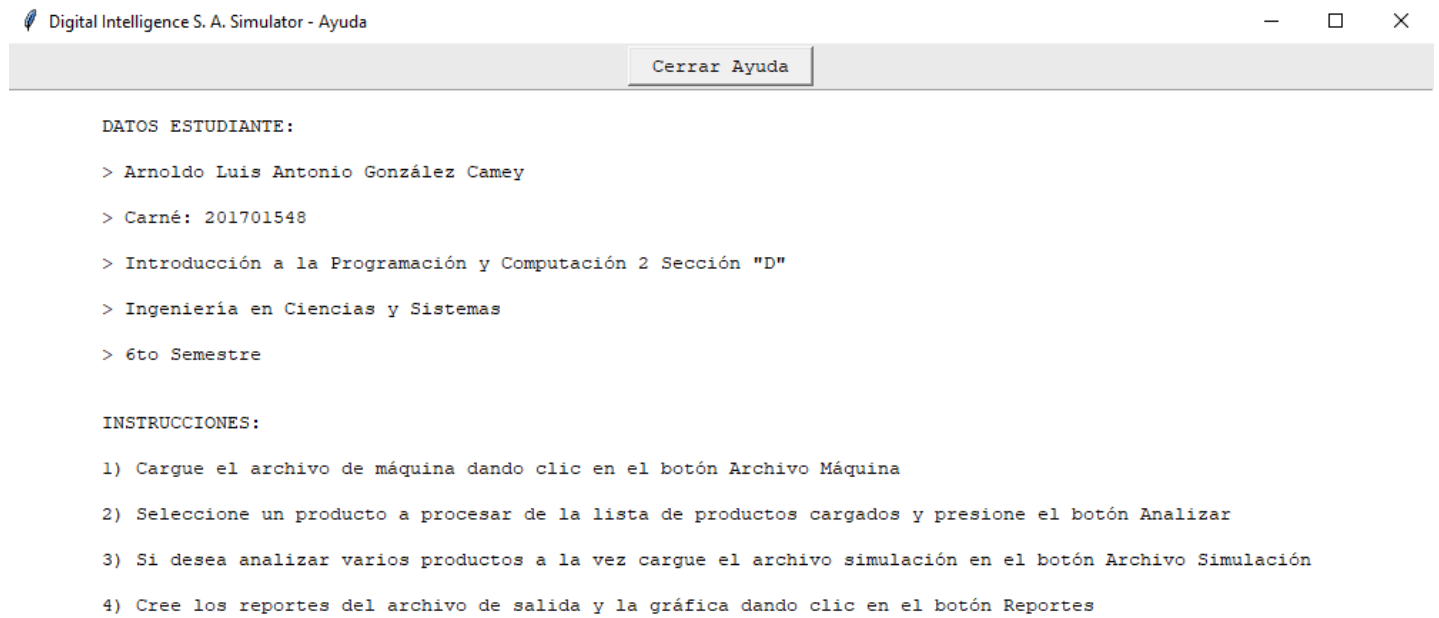


Figura 3. Ventana Ayuda

Fuente: elaboración propia

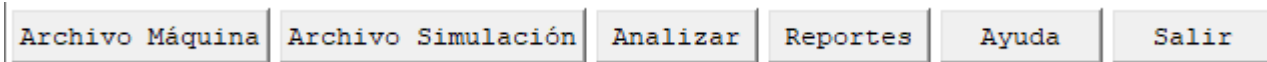


Figura 4. Botones Interfaz

Fuente: elaboración propia

```
<?xml version="1.0" encoding="UTF-8"?>
<Maquina>
  <CantidadLineasProduccion>
    13
  </CantidadLineasProduccion>
  <ListadoLineasProduccion>
    <LineaProduccion>
      <Numero>
        1
      </Numero>
      <CantidadComponentes>
        17
      </CantidadComponentes>
      <TiempoEnsamblaje>
        5
      </TiempoEnsamblaje>
    </LineaProduccion>
    <LineaProduccion>
      <Numero>
        2
      </Numero>
      <CantidadComponentes>
        24
      </CantidadComponentes>
      <TiempoEnsamblaje>
        1
      </TiempoEnsamblaje>
    </LineaProduccion>
    <LineaProduccion>
      <Numero>
        3
      </Numero>
```

Figura 5. Archivo de Entrada Máquina

Archivo XML con los datos de las líneas y productos

Fuente: elaboración propia

```
<Simulación>
  <Nombre>
    Simulacion_1
  </Nombre>
  <ListadoProductos>
    <Producto>
      Banda Galaxy Fit 2
    </Producto>
    <Producto>
      Barbie
    </Producto>
  </ListadoProductos>
</Simulación>
```

Figura 6. Archivo de Entrada Simulación

Archivo XML con los nombres de los productos a simular

Fuente: elaboración propia

```
<SalidaSimulacion>
  <Nombre>Maquina</Nombre>
  <ListadoProductos>
    <Producto>
      <Nombre>iPhone 6s</Nombre>
      <TiempoTotal>121</TiempoTotal>
      <ElaboracionOptima>
        <Tiempo NoSegundo = "1">
          <LineaEnsamblaje NoLinea = "2">
            Ensamblando
          </LineaEnsamblaje>
        </Tiempo>
        <Tiempo NoSegundo = "2">
          <LineaEnsamblaje NoLinea = "7">
            Ensamblando
          </LineaEnsamblaje>
        </Tiempo>
        <Tiempo NoSegundo = "11">
          <LineaEnsamblaje NoLinea = "9">
            Ensamblando
          </LineaEnsamblaje>
        </Tiempo>
        <Tiempo NoSegundo = "18">
          <LineaEnsamblaje NoLinea = "11">
            Ensamblando
          </LineaEnsamblaje>
        </Tiempo>
        <Tiempo NoSegundo = "24">
          <LineaEnsamblaje NoLinea = "10">
            Ensamblando
          </LineaEnsamblaje>
        </Tiempo>
      </ElaboracionOptima>
    </Producto>
  </ListadoProductos>
</SalidaSimulacion>
```

Figura 7. Archivo de Salida

Archivo XML con la elaboración del producto

Fuente: elaboración propia

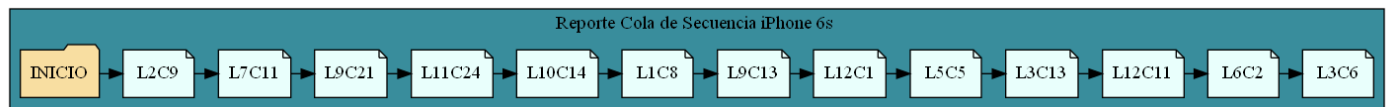


Figura 8. Gráfica Cola de Secuencia

Fuente: elaboración propia

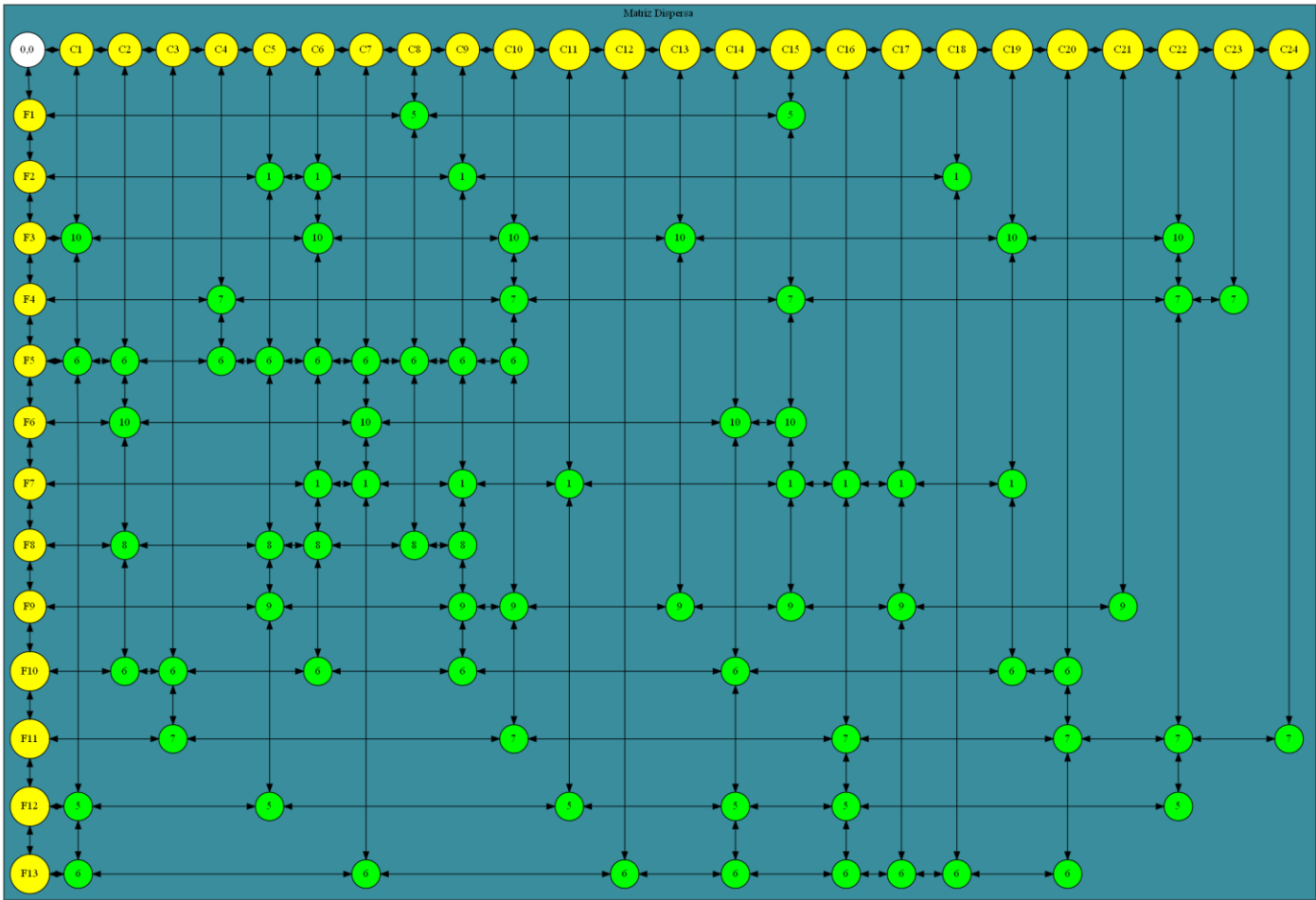


Figura 9. Gráfica Matriz Dispersa

Fuente: elaboración propia