

Informe Entrega Final TLA



2do cuatrimestre 2023

Integrantes:

- **Fernandez Dinardo, Juan Ignacio - 62466**
- **De Caro, Guido - 61590**

Introducción	3
Desarrollo del Proyecto	3
Imagen 1.1: Reglas de las propiedades.	3
Imagen 1.2: Creación de uniones.	4
Dificultades de desarrollo	4
Futuras modificaciones	4

Introducción

Durante el desarrollo de este trabajo, se buscó poder implementar un lenguaje de programación que al ser compilado generen mapas conceptuales y dar varias opciones de customización sobre el mismo siendo el resultado un archivo pdf que el usuario pueda enviar de ser necesario.

Se proveerán funciones para crear un concepto principal y varios subconceptos. Además el usuario podrá definir los colores que tendrán los cuadros de cada concepto. También se permitirá que las uniones tengan flechas que puedan ir en cualquier dirección, o en un loop si se desea. Las uniones podrán ser sólidas o cortadas .

Desarrollo del Proyecto

Durante el inicio del proyecto, se pensó una versión del lenguaje donde no era necesario hacer verificaciones de los nombres de los nodos, pero muy rápidamente nos dimos cuenta que esto era esencial para el lenguaje ya que sin eso podría haber varios errores en la creación de las uniones y las flechas. Además, esto podría resultar en la creación de nodos con el mismo nombre y así confundir al usuario. Al poder tener varias opciones de color tanto para el fondo del nodo como para el contorno se optó por tener las palabras clave “Border” y “Background” seguidas del color para poder diferenciar a donde se le quiere aplicar.

```
property: UNION OPEN_PARENTHESIS NAME CLOSE_PARENTHESIS LINE LINETYPE
| UNION OPEN_PARENTHESIS NAME CLOSE_PARENTHESIS
| TEXT OPEN_PARENTHESIS STRING CLOSE_PARENTHESIS
| BACKGROUND OPEN_PARENTHESIS COLOR CLOSE_PARENTHESIS
| BORDER COLOR OPEN_PARENTHESIS COLOR CLOSE_PARENTHESIS
;
```

Imagen 1.1: Reglas de las propiedades.

Como se puede observar en la imagen 1.1, todo lo que serían *datos* a procesar como el color, el string que sería el texto y los nombres de nodos van entre paréntesis. Esto es para que el usuario mantenga familiaridad con los lenguajes ya conocidos.

Por otro lado las conexiones entre nodos estan pensadas de modo que escribirlas sea tan natural como hacerlo en una hoja de papel o en un software. Esto se puede observar en la Imagen 1.2.

```

6
7 connect_nodes:  NAME ARROW NAME
8                 |  NAME DOUBLE_UNION NAME
9                 |  NAME LOOP
10                |  NAME ARROW NAME LINE LINETYPE
11                |  NAME DOUBLE_UNION NAME LINE LINETYPE
12                |  NAME LOOP LINE LINETYPE
13                ;

```

Imagen 1.2: Creación de uniones.

En conclusión, durante todos los momentos del desarrollo nuestra mayor prioridad fue generar un lenguaje lo más parecido a como se haría un cuadro conceptual en el día a día para que el uso del mismo sea intuitivo. Esto no fue sin sus problemas y limitaciones sobre las cuales hablaremos más adelante.

Dificultades de desarrollo

Como se mencionó anteriormente, el lenguaje fue pensado para que se asemeje lo más posible a como se hacen cuadros sinópticos en el día a día. Para esto debimos tomar en cuenta que el usuario podría querer utilizar varios colores y aquí fue donde surgió nuestro primer problema. Para poder llevar esto a cabo, en un inicio pensamos en agregar todas las reglas con los diferentes colores en flex hasta que nos pusimos a investigar y vimos que se podían agrupar todas en un array y denominarlo COLOR que es como nos quedó hoy en día. Esto se puede observar en la imagen 1.1.

Otra problemática que se presentó durante el desarrollo temprano del proyecto fue el hecho de como conectar los nodos. Tuvimos que tener en cuenta varios tipos de uniones que son las observadas en la imagen 1.2 y hacer ajustes al código acorde a esto para poder verificar y asegurar el correcto funcionamiento de todos los tipos de uniones.

Futuras modificaciones

1. Agrupar ideas. Esto fue propuesto originalmente como una devolución a la primera entrega pero no lo pudimos poner en práctica porque consideramos que para poder implementarlo correctamente necesitamos varias producciones más. No solo eso, sino que también hay varios casos a tener en cuenta como qué hacer si un nodo pertenece a varios grupos. Consideramos que es una idea que aporta mucho al lenguaje ya que permite visualizar mejor el flujo de ideas y hacer un mejor mapa conceptual pero por la alta dificultad decidimos no implementarlo.
2. Incluir imágenes en el cuadro. Para esto deberíamos averiguar como permitirle al usuario cargar imágenes y averiguar sobre como poder incluir las mismas dentro del cuadro cuando se genera el pdf. Por la falta de tiempo esta idea no se implementó, y

al no saber tanto sobre el tema no se puede dar un estimado de la dificultad de esta modificación.

3. Permitir al usuario agregar texto a las flechas. Se deberían idea nuevas producciones para esto. Consideramos que no es una modificación muy complicada pero si tiene un poco de complejidad por lo que sería una dificultad media.
4. Poder cambiar la forma de los nodos. Permitir que el usuario defina si el nodo será circular, cuadrado o de otra forma le otorga libertad a la hora de usar el lenguaje. Para esto habría que investigar un nuevo lenguaje al cual compilar el nuevo para poder hacer esto. Por eso definimos una dificultad media para esta modificación.