



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

VISUALIZACIÓN PARA DOCUMENTOS EN PRODUCCIÓN

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL EN COMPUTACIÓN

IGNACIO BALTAZAR PÉREZ MESSINA

PROFESOR GUÍA:  
CLAUDIO GUTIÉRREZ GALLARDO

MIEMBROS DE LA COMISIÓN:  
JOCELYN SIMMONDS WAGEMANN  
MARÍA RIVARA ZÚÑIGA

SANTIAGO DE CHILE  
2017



RESUMEN DE LA MEMORIA PARA OPTAR  
AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN  
POR: IGNACIO BALTAZAR PÉREZ MESSINA  
FECHA: 06/09/2017  
PROF. GUÍA: CLAUDIO GUTIÉRREZ GALLARDO

## VISUALIZACIÓN PARA DOCUMENTOS EN PRODUCCIÓN

Gracias a los avances tecnológicos, hoy es posible preservar los diferentes borradores y etapas por los que transita el proceso de escritura de un documento. Un buen ejemplo de ello es el servicio de almacenamiento web Google Drive y su procesador de texto Docs, que permite almacenar diferentes datos del proceso de escritura que están siendo generados masivamente por los mismos usuarios mientras escriben. Un documento en Docs es más que un texto versionado, es un fino registro de cada cambio ocurrido, al cual se ha llamado “documento-en-producción” para distinguirlo del primero.

A pesar de que hoy contamos con ese material invaluable para la investigación del proceso de escritura, para los investigadores de la escritura eso no es suficiente pues exige avanzados conocimientos computacionales. Por ello aparece la necesidad de presentar esa valiosa información de manera más amigable para el usuario.

En esta Memoria se diseña una visualización para documentos-en-producción y se implementa un prototipo funcional para los datos que se pueden obtener de Google Drive. Fue necesario desarrollar una nueva estructura de datos para su preprocesamiento. La efectividad de la visualización se muestra por medio de casos de estudio de documentos en un dataset que fue obtenido del proceso de escritura de tareas de estudiantes de la Facultad de Ciencias Físicas y Matemáticas. El análisis de estos casos sugiere que la visualización se muestra adecuada para estudiar el proceso de escritura de textos expositivos, permitiendo observar la estructura jerárquica y/o tópica del texto final, y diferentes características de su proceso de producción. La visualización sigue un paradigma de diseño orgánico, contiene elementos de interactividad y basa su efectividad principalmente en las microdecisiones del usuario a nivel de movimientos del puntero.

Se generó un marco de evaluación teórico que arrojó luz sobre las funcionalidades faltantes y las tensiones en el diseño, que muestra, entre otras cosas, que aún no se ha explotado por completo la información disponible y señala el camino para nuevas visualizaciones y futuras exploraciones del documento-en-producción.



# Tabla de Contenido

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1.	Motivación	1
1.2.	Objetivos	3
1.2.1.	Objetivo general	3
1.2.2.	Objetivos específicos	3
1.3.	Metodología	3
1.4.	Resultados esperados	4
1.5.	Alcances	4
1.6.	Contenidos de la Memoria	4
<b>2</b>	<b>Marco conceptual</b>	<b>5</b>
2.1.	Conceptos	5
2.2.	Antecedentes	6
2.2.1.	Documentos Versionados.	6
2.2.2.	Documentos planos	10
2.2.3.	Documentos dinámicos	11
2.3.	Trabajos relacionados	12
2.3.1.	El Enfoque Ecológico	12
2.3.2.	Organic Information Design	13
2.3.3.	Progression analysis	14
2.4.	Tecnologías utilizadas	15
<b>3</b>	<b>Visualización</b>	<b>17</b>
3.1.	Evolución del diseño	18
3.2.	Diseño final	20
3.2.1.	Diagrama	20
3.2.2.	Glifo	20
3.2.3.	Color	21
3.2.4.	Variables globales	22
3.2.5.	Posicionamiento	23
3.2.6.	Interacción	24
<b>4</b>	<b>Estructura de Datos</b>	<b>25</b>
4.1.	Representación	25
4.2.	Estructura de datos abstracta	26
4.2.1.	Operaciones	27

4.3.	Implementación	29
4.3.1.	Clases	29
<b>5</b>	<b>Casos de estudio</b>	<b>31</b>
5.1.	Datos	31
5.1.1.	Descarga de los datos	31
5.1.2.	Preprocesamiento	32
5.1.3.	Dataset	32
5.2.	Metodología	33
5.3.	Casos de estudio	33
5.3.1.	Caso de estudio 1: “Resumen Charla”	34
5.3.2.	Caso de estudio 2: “Reflexión sobre elementos no verbales en la obra: ‘Los Invitados’”	36
5.3.3.	Caso de estudio 3: “Caso de la Presa Vajont”	38
5.3.4.	Caso de estudio 4: “Descripción proceso industrial División El Teniente”	40
5.3.5.	Caso de estudio 5: “El quehacer humano”	45
5.3.6.	Otros casos	45
<b>6</b>	<b>Modelo para la Visualización de Información</b>	<b>49</b>
6.1.	Capas	49
6.1.1.	Capa 1: La visualización como una función	50
6.1.2.	Capa 2: El espacio de la representación	51
6.1.3.	Capa 3: Percepción, legibilidad y tareas	52
<b>7</b>	<b>Evaluación</b>	<b>55</b>
7.1.	Evaluación por Capas del Prototipo	55
7.1.1.	Capa 1	55
7.1.2.	Capa 2	55
7.1.3.	Capa 3	57
<b>8</b>	<b>Conclusión</b>	<b>63</b>
8.1.	Cumplimiento de objetivos	63
8.2.	Implicaciones	64
8.3.	Limitaciones	66
8.4.	Trabajo futuro	67
	<b>Bibliografía</b>	<b>69</b>
	<b>Anexos</b>	<b>73</b>

# Índice de Tablas

Tabla 1. Funciones interactivas.	24
Tabla 2. Documentos del dataset analizados en esta sección y sus características.	36



# Índice de Ilustraciones

Figura 1. La historia de un documento visualizada con THOM.	1
Figura 2. History Flow (Viégas, Wattenberg & Dave, 2004).	7
Figura 3. Cumulative Revision Map (Kim, Dillon & Lebanon, 2012).	8
Figura 4. DocuViz (Wang et al., 2015a).	9
Figura 5. Dos textos visualizados en Caporossi & Leblay (2011).	12
Figura 6. Anemone (Fry, 2000).	13
Figura 7. Packed bubble chart (Fuente: <a href="http://bl.ocks.org/mbostock/4063269">bl.ocks.org/mbostock/4063269</a> ).	14
FIGURA 8. THOM de este capítulo en tres momentos de su desarrollo: (1) documentos vacío; (2) primera sesión; (3) tercera sesión.	17
Figura 9. Primer diseño. El tiempo corre hacia arriba y el documento completo es dibujado en cada fila. Los puntos negros que dan la sensación de caminos codifican los signos de puntuación y los más gruesos los lugares de operación.	18
Figura 10. Segundo diseño. Cada círculo codifica una operación: azul para inserciones y rojo para eliminaciones; su tamaño la cantidad de caracteres involucrados. El tiempo va de arriba a abajo y de izquierda a derecha, organizándose en filas.	19
Figura 11. Diversos diseños que precedieron al prototipo final.	19
Figura 12. Evolución del glifo. De izquierda a derecha: triangular; diente de león; paracaídas (actualmente en uso).	21
Figura 13. La gama de colores fue inspirada en las cuatro estaciones. Los colores parten en la primavera y se suceden en sentido antihorario.	22
Figura 14. Esquema del posicionamiento recursivo de glifos. $l$ es la distancia de inserción, $r$ el radio del glifo anterior relativo, $\Phi$ el ángulo con respecto a la vertical y $\alpha$ el ángulo de apertura.	23
Figura 15. Un árbol en crecimiento. ( <a href="https://drive.google.com/open?id=0B3CAAdHkf9F8iV1I1Tnp6RnlwRFk">https://drive.google.com/open?id=0B3CAAdHkf9F8iV1I1Tnp6RnlwRFk</a> )	24
Figura 16. Esquema progresivo de un documento. Los nodos de eliminación se representan con relleno negro y los de inserción con blanco, los lugares de operación como ejes que no van a ningún nodo. La dirección de los ejes siempre es “hacia arriba”. (1) Documento vacío. (2) Inserción de “A”. (3) Inserción de “BC” en la segunda posición, resultando en el string “ABC”. (4) Eliminación simultánea de “AB”. (5) Inserción de “D” antes de “C”, resultando en el string “DC”. (6) Eliminación de “C”. El documento resultante sólo contiene el string “D”.	26
Figura 17. A la izquierda, la altura de cada documento es su “índice de complejidad”, ordenados de menor a mayor. A la derecha los mismos documentos, su posición determinada por los factores del “índice de complejidad”. En ambas figuras, la línea punteada representa el plano de corte, desde el cual los documentos con mayor complejidad pasan a formar parte del dataset.	32
Figura 18. Captura del documento “Resumen Charla” y su visualización.	34

Figura 19. Aunque la primera impresión del documento no permite distinguir bien su estructura (izquierda), al ocultar un nodo podemos distinguir que este corresponde a la primera rama principal, dejando a la vista la segunda (derecha).	35
Figura 20. La visualización (izquierda) y el documento (derecha) con las áreas correspondientes resaltadas del mismo color.	36
Figura 21. Captura del documento “Reflexión sobre elementos no verbales en la obra: ‘Los invitados’” y su THOM final.	37
Figura 22. El documento en un momento intermedio de su producción (izquierda) y en su versión final (derecha). Se observa que todo el texto original (de color morado) fue borrado en una sesión posterior.	37
Figura 23. El documento y su visualización en su versión final con las áreas correspondientes resaltadas.	38
Figura 24. Captura del documento “Caso de la Presa Vajont: Análisis Axiológico y Reflexión Personal”.	39
Figura 25. Visualización del documento “Caso de la Presa Vajont: Análisis Axiológico y Reflexión Personal” (izquierda) y con ramas coloreadas (derecha).	40
Figura 26. Captura del documento con colores correspondientes a la Figura 25.	41
Figura 27. Captura del documento “Descripción proceso industrial División el Teniente”.	42
Figura 28. El árbol de “Descripción proceso industrial División el Teniente“ (arriba) y el mismo árbol con sus principales ramas coloreadas artificialmente (abajo). Los colores (amarillo, rojo, naranja, morado, verde claro, verde oscuro y azul) indican la correspondencia de los nodos con las partes del texto resaltadas con los mismo colores en la Figura 29.	43
Figura 29. Captura del documento visualizado en 1a. Los textos resaltados están contenidos en los nodos marcados con el mismo color en 1a.	44
Figura 30. Visualización del documento “El quehacer humano”. Se distinguen los siguientes características: (a) nodos copiados de otro documento; (b) principales ramas; (c) ediciones posteriores.	46
Figura 31. Visualizaciones de casos considerados ilegibles.	47
Figura 32. Tipología multi-nivel de tareas de visualización abstractas (Brehmer & Munzner, 2013).	53

# 1 Introducción

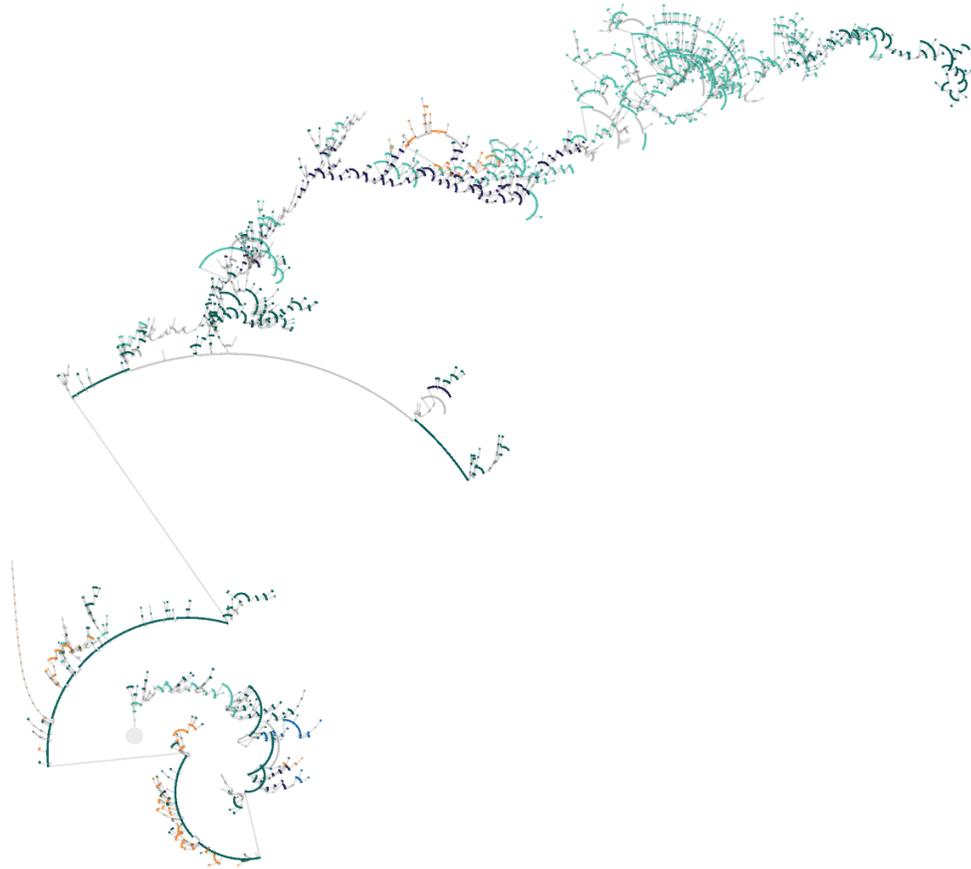


Figura 1. La historia de un documento visualizada con THOM.

En esta Memoria se presenta una nueva visualización de información, apodada THOM, diseñada para hacer visible el proceso de producción detrás de un documento, usando los datos de granularidad fina que registra Google Docs (Figura 1). Este capítulo provee una Motivación, Objetivos y Metodología para el trabajo, seguido de Alcances y un sumario de Contenidos.

## 1.1. MOTIVACIÓN

El computador es una forma de relacionarse con el lenguaje escrito: escribimos en nuestro computador y leemos textos que fueron escritos en otros computadores. En la mayoría de los casos, tal como en un libro, las marcas físicas del proceso de escritura quedan fuera del producto final. “Lo escrito (el producto) busca superar la escritura (el proceso)” (Grésillon y Perrin, 2014), es decir, escribir es como caminar devolviéndonos constantemente a borrar nuestros pasos. Para poder dilucidar este proceso, los investigadores se han visto en la necesidad de utilizar herramientas especiales, costosas e intrusivas, que muchas veces entregan información demasiado gruesa, como grabar la pantalla (Latif, 2008). Actualmente, sin embargo, los mismos programas usados para escribir con

el computador, los procesadores de texto, están empezando a producir un registro cada vez más fino de la actividad del usuario, bajo una idea que se ha extendido desde la misma computación: el versionamiento.

Google Drive es un servicio de almacenamiento web introducida por Google en 2012. Provee una suite de aplicaciones tipo Microsoft Office, entre las cuales cuenta con un editor de textos conocido como Docs. Docs posee un sistema de versionamiento que permite la colaboración a través de internet y donde cada documento es un historial de los cambios ocurridos en él registrados con una precisión de milésimas de segundo y prácticamente sin agregación alguna. Basta esto para afirmar que datos de escritura están siendo producidos masivamente por los mismos usuarios y sin intrusión alguna. Por esto, Docs aparece como una herramienta idónea para la investigación lingüística y la experimentación para el desarrollo de aplicaciones de apoyo en la escritura, en particular tratándose del proceso de producción de texto de forma individual y colectiva en la era digital. Sin embargo, la disponibilidad de información no es suficiente: esta también tiene que ser visibilizada.

La visualización, técnica que de a poco se ha ido conformando en su propio campo dentro de la ciencia, es la forma que tenemos de acercar a nuestros sentidos las grandes cantidades de datos y hacerlos perceptibles de la mejor manera, i.e, tratando de aprovechar las capacidades de procesamiento de nuestro sistema visual. Visualizaciones se desarrollan de manera independiente cada vez que se encuentra una fuente de datos que prometa ser capitalizable (y hoy en día, cualquier fuente de datos lo es potencialmente). Con la aparición de Internet, las redes sociales y el movimiento de la economía hacia una de la información y el *knowledge worker*, la visualización se ve cada vez más demandada y, en el área de visualización de texto en particular, hay una tendencia creciente a la producción de estas visualizaciones (Kucher & Kerren, 2015).

Drive no sólo representa una importante fuente de datos, nunca antes en producción masiva, sino un nuevo desafío para la visualización. Los datos que nos provee Drive sobre un documento no es lo mismo que el texto versionado (como el que podemos encontrar en un sistema de control de versiones): en términos informáticos, el primero, al que de ahora en adelante llamaremos “documento-en-producción”, es una serie de *operaciones* (inserción, eliminación) local y temporalmente determinadas, vale decir, cada elemento de esta serie corresponde a un “delta” de texto necesario para llegar al documento en su estado final. Un archivo versionado común, por el contrario, es una colección de estados particulares (en vez de una secuencia de cambios atómicos), que incluso pueden ser excluyentes, o sea, mantener más de un estado final para un mismo archivo de origen. Por esta razón, las visualizaciones existentes para texto versionado no son de utilidad para documentos-en-producción. Es el propósito de esta Memoria el desarrollar una nueva visualización para este tipo de dato, que visibilice esta información y sirva como herramienta para el estudio de la escritura y la producción escrita.

## 1.2. OBJETIVOS

En esta sección son detallados los objetivos general y específicos del proyecto.

### 1.2.1. Objetivo general

Diseñar y desarrollar una visualización para documentos-en-producción que permita observar el proceso de producción de un documento, considerando su irreversibilidad. Esta debe ser utilizable aunque no necesariamente más que un prototipo. Debe ser validada a través de casos de estudio que ilustren su uso.

### 1.2.2. Objetivos específicos

1. Producir un dataset sobre el cual se trabajará en esta Memoria. (Puntos 1 y 2 en Metodología; no existe hoy un conjunto de fuentes estándar para ser usado).
2. Hacer una revisión sistemática de la literatura.
3. Diseñar e implementar las estructuras de datos necesarias para el preprocesamiento de los datos.
4. Diseñar y desarrollar una visualización de información para documentos-en-producción.
5. Establecer indicadores y métricas preliminares para evaluar la soluciones, que a la vez permitan compararlas con otras (futuras) visualizaciones.
6. Probar la efectividad de las visualizaciones realizadas aplicándolas sobre el dataset del punto 1.

## 1.3. METODOLOGÍA

A continuación se describe la metodología seguida en este trabajo. Éste ha sido dividido en cinco ejes: (1) *Generación del dataset*; (2) *Revisión de la literatura*; (3) *Desarrollo de la visualización*; (4) *Experimentación*; y (5) *Evaluación*.

1. *Generación del dataset*. Obtener los datos sobre los que trabajar (documentos escritos en Docs), filtrarlos y hacer un esquema de preprocesamiento.
2. *Revisión de la literatura*. Llevar a cabo una revisión sistemática de la literatura y los desarrollos asociados a la visualización de textos.
3. *Desarrollo de la visualización*. Diseñar e implementar la visualización.
4. *Experimentación*. Estudiar el dataset usando la visualización y generar casos de estudio que demuestren su efectividad y apoyen su evaluación.
5. *Evaluación*. Generar un marco teórico para la evaluación de la visualización y usarlo para encontrar sus fortalezas y debilidades, y así dejar un camino para futuras iteraciones.

## **1.4. RESULTADOS ESPERADOS**

El resultado esperado de esta Memoria es un prototipo de visualización para documentos-en-producción, junto con una evaluación teórica que la respalde y casos de estudio que demuestren su utilidad. Esta visualización debe ser diferente a las actuales soluciones para texto versionado o documentos-en-producción que se encuentran en la literatura, ya que se considera que éstas están lejos de poder abarcar el fenómeno que el documento-en-producción presenta.

## **1.5. ALCANCES**

Debido a las limitaciones de tiempo impuestas por la Memoria, sólo se considera la producción de un prototipo funcional de la visualización que permita su uso con los datos que se obtengan durante la producción del dataset. Tampoco se hará un análisis a fondo del dataset, puesto que no es una investigación sobre la escritura, sino el desarrollo de una herramienta para ella. Por último, una evaluación de la solución con usuarios está fuera del alcance de esta Memoria.

## **1.6. CONTENIDOS DE LA MEMORIA**

Esta Memoria se estructura de la siguiente forma: en el Capítulo 2 se sondea el estado del arte en visualización de texto y se presentan los trabajos relacionados para rápidamente pasar, en el Capítulo 3, al diseño de la visualización y prototipado de la visualización. El Capítulo 4 se describe el diseño y la implementación de la estructura de datos abstracta sobre la que funciona la visualización. En el Capítulo 5 se presentan varios casos de estudio que muestran la visualización y su uso, con algunos de los datos obtenidos, presentándose además algunas conclusiones preliminares sobre la solución alcanzada. El Capítulo 6 desarrolla un modelo basado en diferentes autores que nos permitirá en el Capítulo 7 evaluarla y guiar su desarrollo futuro. Por último, el Capítulo 8 contiene la discusión del trabajo futuro, limitaciones y conclusión de este trabajo.

## 2 Marco conceptual

En este capítulo se da el contexto teórico en el que se desarrolla este trabajo, incluyendo visualizaciones y trabajos en áreas relacionadas al lenguaje natural que inspiran y aportan a esta Memoria en diferentes niveles, conceptos y las tecnologías a utilizar. El capítulo se organiza en cuatro secciones: (1) una aclaración de los conceptos utilizados durante la Memoria; (2) un sondeo del estado del arte en visualización tradicional de documentos, orientado principalmente a entender las limitaciones de esas soluciones que las hacen inutilizables para visualizar documentos-en-producción; (3) presentación del “enfoque ecológico”, el diseño de información orgánico y el *progression analysis*, los cuales inspiraron la visualización aquí desarrollada; y por último, (4) una breve descripción de los lenguajes involucrados en el desarrollo de esta Memoria.

### 2.1. CONCEPTOS

Hay esencialmente dos conceptos que son muy relevante en el desarrollo de esta Memoria: las nociones de diagrama y de glifo.

**Diagrama.** En este trabajo se llama “diagrama” a la relación organizativa de los elementos visuales discretos con el “fondo” de la visualización. La noción esta inspirada en el uso que hace Deleuze de este concepto para estudiar la pintura (Deleuze, 2007). Se habla de “diagrama lineal” cuando el espacio está organizado de acuerdo a coordenadas cartesianas ortogonales (en la dimensión que sea), y equivalentemente se puede hablar de “diagrama logarítmico” e “hiperbólico”. Estos describen casi todas las visualizaciones que se utilizan comúnmente. Los grafos escapan a este diagrama, ya que el fondo no cumplen muchas veces ninguna función organizativa de los elementos. Mucho debate existe sobre las formas de dibujar grafos, siendo una de las más aceptadas el *force-directed layout*, en el que el diagrama funciona como un simulador que se adapta a las fuerzas que los elementos ejercen entre sí. Este caso podría ser llamado de diagrama “descentralizado adaptativo” ya que no hay un orden preestablecido de para los elementos, sino que estos se reordenan en la medida que van entrando nuevos datos.

**Glifo.** En visualización de información, un glifo se define como una representación visual de un pedazo de información, donde los atributos de una entidad gráfica están dictados por uno o más atributos de un dato (Ward, 2008). Para dilucidar esta definición, un tanto oscura, podemos retrotraernos al anterior uso técnico de la palabra, que tiene su origen en la tipografía. En tipografía, dentro del contexto de la escritura digital, *glifo* se define como una versión conceptual del signo abstracto *carácter* (Bringhurst, 2014). Así, un mismo carácter puede tener varios glifos asociados, glifos que a su vez pueden tener diferentes representaciones en diferentes fuentes (e.g., z y z). Un glifo en este caso particular sería el equivalente a un valor Unicode. Lo mismo se puede decir de un carácter que de un dato: puede ser conceptualizado a través de diferentes glifos, que a su vez

pueden ser representados en diferentes tamaños y estilos. Por ejemplo, los puntos en un *scatter-plot* pueden ser conceptualizados con diferentes glifos: cuadrados, círculos, triángulos, estrellas, etc. Cómo sean dibujados puede depender de otros datos, como un tercer atributo cuantitativo o categórico, de la misma manera que un carácter varía su tamaño o fuente cuando viene con esta información.

## **2.2. ANTECEDENTES**

En esta sección se presenta el resultado de una revisión sistemática de la literatura sobre visualización de texto más cercana a este trabajo. Como se puede observar, ningún trabajo expuesto provee de una solución para el problema que se ha propuesto esta Memoria, el documento-en-producción, y la revisión sirve fundamentalmente dos fines: constatar esta ausencia de soluciones y exponer el paradigma del cual se desprende su diseño.

La visualización de texto ha sido un área de investigación desde hace tiempo, debido a la complejidad intrínseca del contenido textual. Se ha desarrollado un considerable número de visualizaciones (en TextVis (Kucher & Kerren, 2015) se registran 375 entre 1976 y 2017, con 58 publicadas sólo durante 2016) para cumplir diferentes propósitos, que van desde permitir al usuario ver el desarrollo de los tópicos a través de una colección de textos hasta ver las zonas más afectadas por el versionamiento en el código fuente. La mayoría de los esfuerzos han sido dedicados a visualizar grandes cantidades de documentos, corpora, usando y perfeccionando técnicas del procesamiento de lenguaje natural. Para una descripción extensiva del estado del arte en visualización de texto, ver Šilić (2010), Alencar (2012) y Cao & Cui (2016).

Esta sección está dividida en tres subsecciones, de acuerdo al tipo de dato: (1) documentos versionados, (2) documentos estáticos y (3) documentos dinámicos.

### **2.2.1. Documentos Versionados.**

Sólo recientemente ha comenzado a ser un tema de estudio el versionamiento de documentos, sobre todo por la expansión de la producción colaborativa (Wikipedia, Drive, entre tantos otros servicios web), y ya se ha presentado como un área llena de interesantes patrones por revelar. Las visualizaciones aquí descritas representan lo más cercano que se ha hecho con respecto a este trabajo, en términos de datos o pretensiones. Sin embargo, ninguna está diseñada para abarcar el nivel de detalle que pretendemos visualizar en el presente trabajo. La mayoría están enfocadas en distinguir patrones de colaboración en documentos colectivos, pero nos enseñan sobre el esquema imperante en este tipo de visualización. La visión que plantean del texto versionado está más cerca del texto estático que de un texto en evolución, lo cual se debe en parte a la granularidad de los datos a la que apuntan (revisiones en vez de diferenciales). Otro punto débil, aunque relacionado a la granularidad, tiene que ver con la eliminación de texto. Ésta operación, al igual que insertar, ocurre en todos los niveles del proceso de escritura: sirve para darle un ritmo a la producción, un tiempo propio, y nos indica qué partes del texto pueden haber resultado más difíciles, o albergar mayor

complejidad. Las visualizaciones que aquí son descritas reproducen la lógica que le da su función al borrar, el poder *ocultar las trazas del mismo proceso*. Más aún, aunque no podemos saber a partir de los datos dónde el autor está *releyendo* lo que escribió, una borradura siempre es la marca de una lectura, por lo que portan la información de dónde se estaba poniendo atención en un momento determinado, de manera similar a como ocurre en Perrin & Wildi (2010).

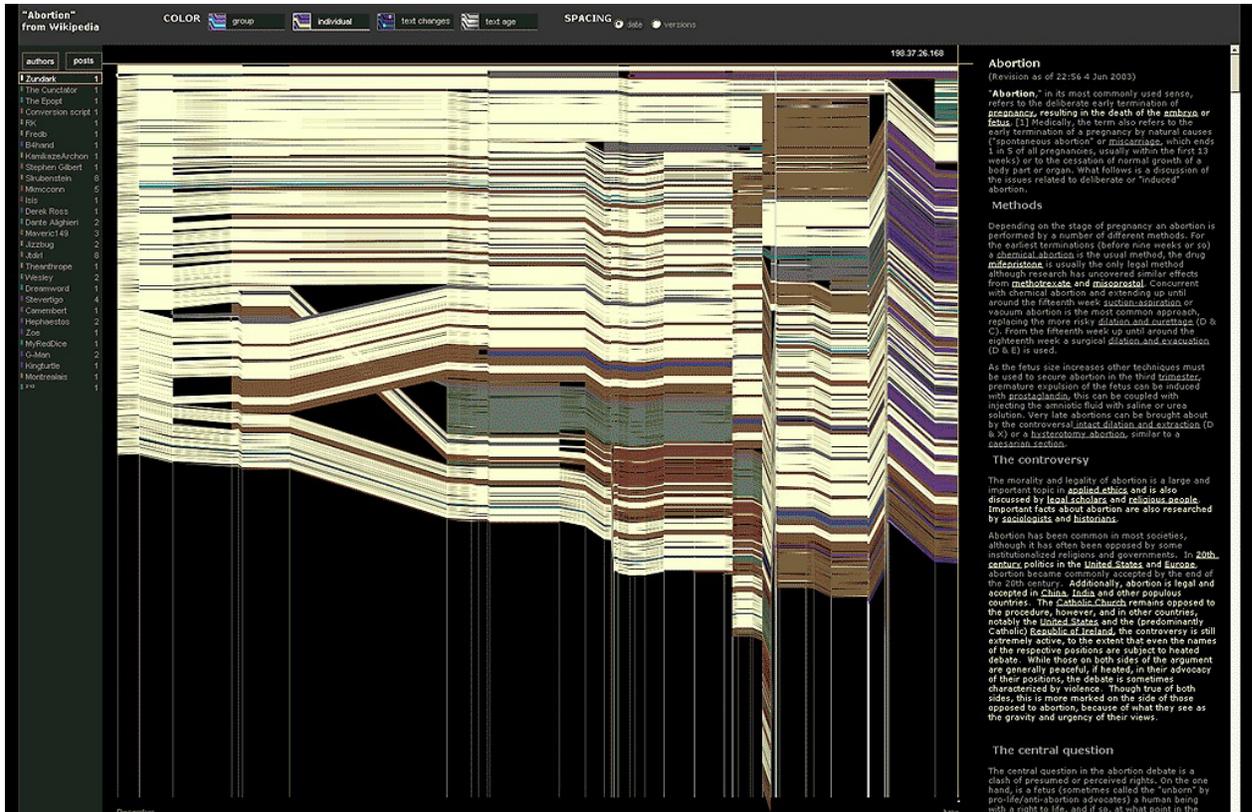


Figura 2. History Flow (Viégas, Wattenberg & Dave, 2004).

**History Flow.** History Flow (Viégas, Wattenberg & Dave, 2004) es una visualización desarrollada por IBM para observar los patrones de edición y colaboración entre usuarios a través del historial de versiones de un artículo de Wikipedia. En estas visualizaciones, la historia del texto es representada como una serie de líneas verticales dispuestas sobre un eje de tiempo horizontal. Las franjas representan el texto (el alto de la franja es el largo del string), cuándo es añadido o suprimido es codificado por el comienzo y final de la franja, y el color el usuario que agregó el contenido (ver Figura 2). Esto permite identificar ciertos comportamientos que se dan en documentos colaborativos, como las llamadas *edit wars*, donde recurrentemente es borrado y reinsertado el contenido propuesto por un usuario. Se pueden ver además patrones en que las franjas se entrecruzan, significando movimientos de un texto dentro del mismo documento.

Esta visualización marca la pauta para el resto de los trabajos citados en esta sección, lo que es prueba de su influencia. La visualización desarrollada busca proponer un esquema diferente, que se aleje de la visión que History Flow presenta del documento versionado como una secuencia de documentos estáticos.

**Cumulative Revision Map.** Cumulative Revision Map (Kim, Dillon & Lebanon, 2012) se mantiene en la línea de History Flow, con la diferencia de que en ésta, lo que se visualiza en cada nueva revisión no es el estado completo del texto sino la diferencia entre uno y otro. Esta visualización propone la idea de visualizar el documento a través de las diferencias sucesivas de sus versiones, con el fin de ahorrar espacio visual y mostrar información más fina, sin embargo su linealidad, heredada de History Flow, le impide hacer mejor uso del espacio, como se ve en la Figura 3. El tiempo en este caso va hacia abajo y el largo del documento horizontalmente. Una franja representa un texto agregado en cierto momento, pero el color de esta nos dice si fue borrado posteriormente (rojo) o si está aún vigente (azul). Las líneas verticales forman el camino para recuperar el texto actual, para lo cual hay que ir de izquierda a derecha, partiendo en la primera franja azul y siguiendo estas líneas hasta el otro extremo de la visualización. Los rectángulos más oscuros ayudan a diferenciar secciones del texto (cómo lidia esta visualización con cambios en la estructura del documento es desconocido). La franja superior muestra una escala de color que señala las zonas del documento más editadas.

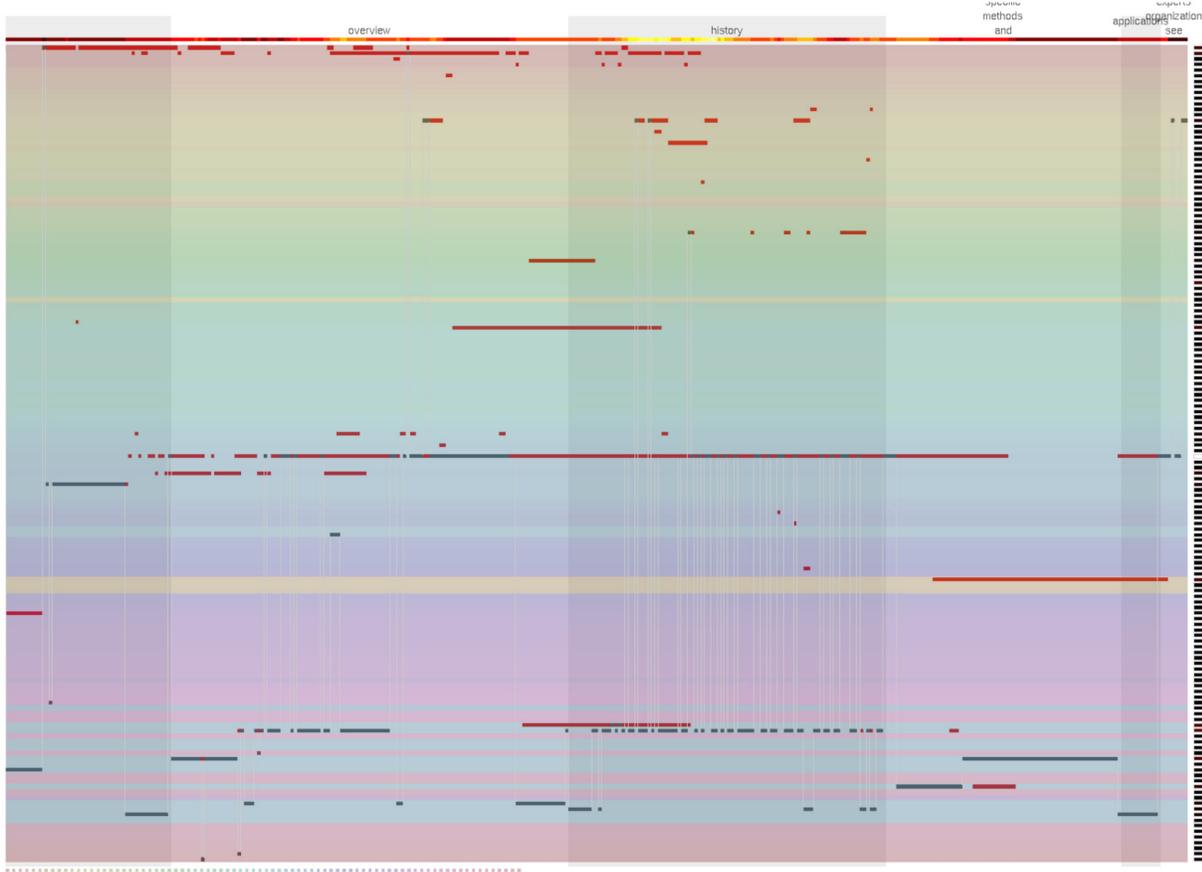


Figura 3. Cumulative Revision Map (Kim, Dillon & Lebanon, 2012).

La idea de formar la historia del texto a partir de las diferencias es recogida en esta Memoria, sin embargo, es aún más importante la idea de que hay que apartarse de la ortogonalidad tiempo-espacial si es que se quiere optimizar el espacio visual.

**DocuViz.** DocuViz (Figura 4) aparece en 2015 desarrollado por un grupo de la University of California aplicando el concepto de History Flow en documentos de Google Docs, debido a que la herramienta desarrollada por IBM no es de uso libre y sólo funciona con páginas de Wikipedia. DocuViz “visualiza cómo evoluciona un documento en Google Docs, mostrando quién hizo qué y cuándo. Es una imagen de la historia de revisiones en el tiempo” (Wang et al., 2015a). De hecho, se realizó una experiencia en la que estudiantes universitarios analizaron sus patrones de colaboración en sus tareas hechas en esta plataforma (Wang et al., 2015b). Similar a History Flow, una versión del documento está representada por una columna, con franjas que unen las partes de texto correspondientes entre una versión y otra. Los colores nuevamente codifican al usuario que agrega el texto. El largo de la columna es el largo del documento y el espacio entre ellas puede ser mostrado en una relación lineal con el tiempo real de edición o de forma equidistante.

DocuViz hace una primera aproximación a la idea de estudiar la producción de texto en alumnos a través de la visualización, descubriendo en Wang et al. (2015b) varios patrones observables, poniendo así una marca que este trabajo pretende expandir (ver Capítulo Experimentación), aunque no igualar, ya que no se cuenta con la información de usuario.

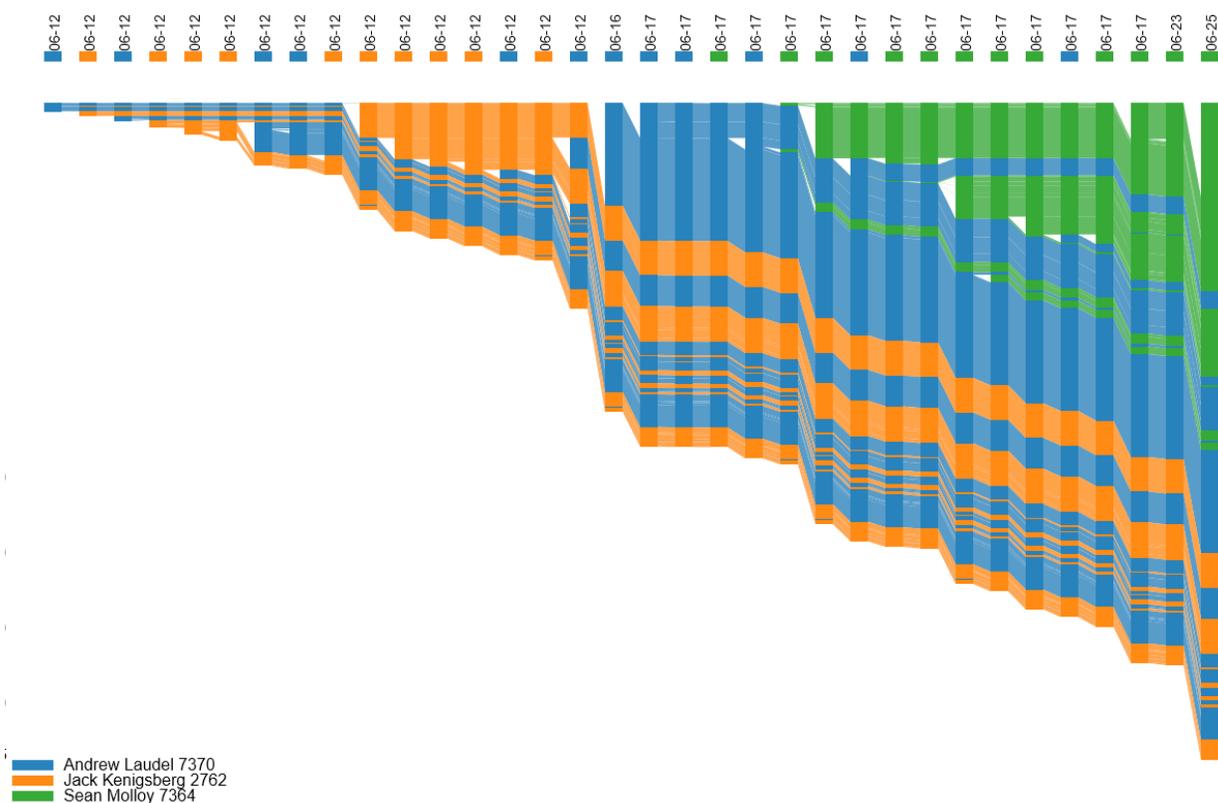


Figura 4. DocuViz (Wang et al., 2015a).

**Historial de Google Docs.** Google Docs (actualmente en su versión 1.7) es el programa de edición de textos que Google ofrece a través de su servicio web gratuito Google Drive. En cualquier documento del que tengamos permisos de edición en Docs, podemos acceder a su historial de revisión. Aquí se nos presenta una vista del documento similar a la que tenemos normalmente en el modo

edición, con un menú lateral que nos permite navegar entre las “revisiones” del documento desde su origen. Una revisión es la acumulación de cambios pequeños ocurridos dentro de una sesión de Google Docs (una sesión se “cierra” después de una pausa de inactividad mayor a 10 minutos). También se nos da la opción de “mostrar cambios”, en donde el texto añadido desde la revisión anterior aparece con un color correspondiente al usuario que lo introdujo y el texto borrado hace su aparición tachado (en el lugar del documento donde correspondía).

Las visualizaciones que se detallan más adelante trabajan sobre este nivel de granularidad, sin embargo, Docs permite además ver revisiones más detalladas, función que expande las revisiones y nos expone su desglose: las versiones del documento con los pequeños cambios ocurridos en el orden de segundos que acumulados durante una sesión forman una nueva revisión. Si se quiere explorar esta información, hay que hacerlo navegando las versiones, cambio por cambio, lo cual en cualquier texto de pequeño-mediano tamaño ya no tiene sentido. Es por eso que esta visualización agrupa primero los cambios en revisiones, las cuales son la diferencia entre una sesión y otra, una acumulación de cambios infinitesimales, pero al costo de perder la información temporal de los cambios (no podemos saber qué tuvo lugar antes que otra cosa) y perder todo lo que fue añadido y a la vez eliminado (puesto que esto no tiene incidencia en la siguiente sesión). En este nivel de detalle la visualización que nos entrega la misma plataforma es inútil: de aquí proviene la necesidad de producir un nuevo diseño para esta información.

**InVite.** En Lamqaddam & Aerts (2016) utilizan un diagrama similar a History Flow para un sistema de apoyo a la escritura supervisada de documentos que pretende facilitarle el trabajo a investigadores y alumnos trabajando en documentos de tesis (vale decir, con retroalimentación unilateral) pero por ahora sólo muestra dos versiones a la vez. Esta visualización da un nuevo uso al diagrama ya conocido, introducido por History Flow, y significa para el presente trabajo un posible uso de apoyo a la cooperación que se podría dar a la visualización desarrollada.

### **2.2.2. Documentos planos**

Los documentos planos y los conjuntos de documentos planos (corpora) han sido los que más atención han recibido hasta ahora, probablemente por su temprana disponibilidad y por la simpleza de su formato. Estos datos no consideran información sobre su historia, por lo que no son adecuados para investigar la escritura. No se tratan aquí visualizaciones de corpora de texto ya que el presente trabajo apunta sólo a visualizar documentos individuales. Las visualizaciones que se mencionan a continuación se abocan a la tarea de la segmentación tópica de un texto.

Particionar un texto en segmentos temáticamente cohesionados (tópicos) ha sido objeto de estudio desde los años 90, considerándose una herramienta importante para producir información automática a partir de documentos. El trabajo de Hearst prácticamente funda esta área al introducir el algoritmo TextTiling (Hearst, 1997). Este enfoque particiona el texto en bloques contiguos (segmentos subtópicos) según su coherencia local, que se calcula a partir de la variación léxica (movimiento subtópico) entre frases (Hearst, 1994; Richmond et al., 1997). Es una forma relativamente simple e

independiente del lenguaje, pero la partición conseguida depende bastante en la forma de computar puntajes para los bloques. En Dias et al. (2007) se mejoran los resultados de este algoritmo y se lo hace más estable utilizando TF-IDF (*term frequency inverse document frequency*) sobre un corpus de texto.

En Lebanon et al., (2007) se busca visualizar el movimiento subtópico de manera continua, introduciendo un nueva abstracción, el *locally-weighted bag of words (lowbow)*, que transforma un texto en una curva multidimensional. Esta es visualizada al ser proyectada en el espacio dos-dimensional por sus componentes principales. A diferencia del enfoque anterior, donde el texto es accesado a través de una ventana de cierta amplitud que se va desplazando por el texto, el *lowbow* es una distribución de probabilidad de las palabras a lo largo del texto, es decir, mantiene la concepción secuencial del texto pero tomándolo como un campo unificado en vez de como una colección de elementos discretos. La velocidad de la curva (su primera derivada con respecto al tiempo) es la velocidad del movimiento subtópico, por lo que al encontrar los puntos del texto la velocidad alcanza un máximo local se encuentran los “puntos de quiebre” o divisiones donde cambia la temática. Esta técnica trae consigo la novedad de que variando el umbral de velocidad de la curva, podemos obtener mayor o menor granularidad en la partición, obteniéndose así una partición con una estructura jerárquica por la subdivisión recursiva del texto (Mao et al., 2007). En Kim & Lebanon (2010) esta idea es extendida a documentos versionados con la tentativa de predecir futuras operaciones, pero la lógica sigue siendo la misma que para documentos individuales.

Todos los trabajos mencionados trabajan sobre documentos planos, y hasta donde se tiene conocimiento no hay algoritmos que utilicen la información de la escritura de un texto, por lo que la visualización desarrollada tendría que aportar en esta área al mostrar una manera de ocupar el historial para la segmentación tópica, como se ve en el capítulo de Experimentación.

### 2.2.3. Documentos dinámicos

El documento dinámico es lo que más se acerca al proceso de escritura personal por su nivel de granularidad, dato al cual se llama en este trabajo documento-en-producción. Los textos acá citados carecen de mayor consideración por el preprocesamiento o la estructura de datos y se limitan a mostrar información de bajo nivel.

**Basada en grafos.** En Caporossi & Leblay (2011) se usa el grafo como representación de los datos finos de escritura obtenidos con ScriptLog. Cada nodo es una operación y cada eje una relación entre operaciones que puede ser temporal o topológica. En la Figura 5 (página 12), se muestran dos grafos que representan el proceso de escritura de dos usuarios diferentes. Los nodos rojos mapean a la superficie del texto mientras que los nodos amarillos y azules son diferentes tipos de eliminación. Los ejes negros corresponden a la relación cronológica de los nodos, mientras que los otros a relaciones espaciales. Aunque esta visualización podría tener resultados similares a los obtenidos en esta Memoria, su lectura no es fácil ni intuitiva y la falta de más casos de estudio en Caporossi & Leblay (2011) hace difícil una comparación más profunda.

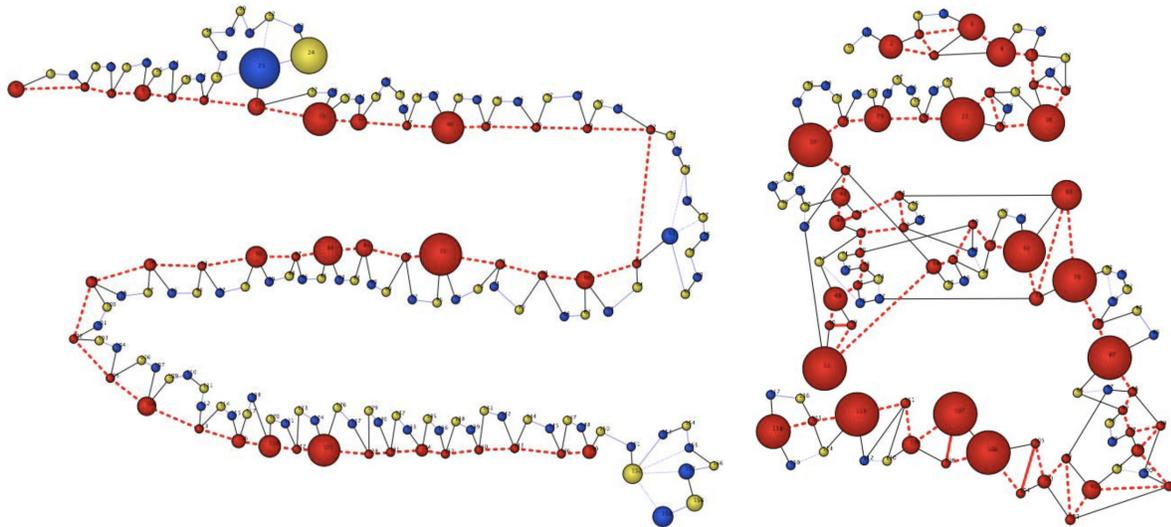


Figura 5. Dos textos visualizados en Caporossi & Leblay (2011).

**Basada en animación.** Otro tipo de visualización busca facilitar la revisión histórica del texto usando transiciones animadas de los cambios ocurridos entre versiones. Bajo esta forma, podemos encontrar a Diffamation (Chevalier et al., 2010), que se especializa en texto versionado, y Draft-back, que renderiza un video de nuestra escritura de un documento en Google Docs. Claramente, ninguna provee información de alto nivel, sino una forma más cómoda de revisión manual.

## 2.3. TRABAJOS RELACIONADOS

En esta sección se da cuenta de tres trabajos que ejercieron importante influencia y que prestaron sus ideas sobre visualización y análisis de texto para esta Memoria.

### 2.3.1. El Enfoque Ecológico

En Wise (1999) aparece la idea del “enfoque ecológico” en el diseño de visualizaciones. Esta plantea que nuestro sistema visual está mejor adaptado y se siente más cómodo al percibir las formas con las que ha tenido que interactuar desde su origen, es decir, las formas de la naturaleza, por lo que una visualización debe explotar esta sensibilidad imitando paisajes y organismos. Wise escribe,

A text visualization’s structure should then result from and reflect a process history, being intimately bound up with “how it got that way.” In the “ecological approach,” text visualization is not a process of “drawing pictures.” It is a result of transferring to the spatial realm the results of computational processes that are themselves analogs of the means by which physical forms are produced. (Wise, 1999)

En otras palabras, la lógica de la visualización debe reflejar la lógica de producción de los datos. En el caso de esta Memoria, esta afirmación significó tratar de entender el “funcionamiento” del documento en el procesador de texto, ya que éste no tiene nada natural sino que es más bien un juego de simples reglas, reflexión que culminó en la estructura de datos descrita en el Capítulo 4.

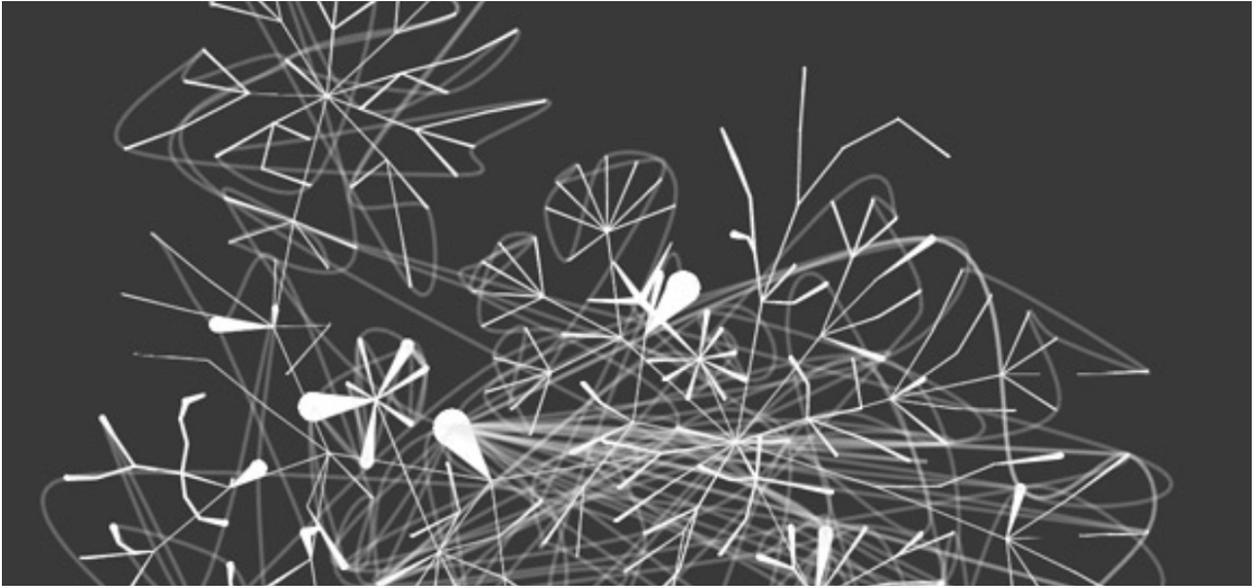


Figura 6. Anemone (Fry, 2000).

### 2.3.2. Organic Information Design

Benjamin Fry, cocreador de Processing (lenguaje en el que se desarrolla la parte de visualización de esta Memoria), plantea un paradigma en el que intenta dar vida a la información de sistemas dinámicos a través de visualizaciones orgánicas que recuerdan a organismos creciendo, floreciendo, reproduciéndose y exhibiendo toda clase de comportamientos (Fry, 2000). En estas visualizaciones, no existe una presuposición del espacio sobre el cual se organiza la información, como ocurre en las visualizaciones de grilla fija (gráficos de dispersión y todas sus derivaciones), y se acerca más a la representación de grafos, donde la posición particular de los elementos visuales no conlleva un significado sino las relaciones que existen entre ellos (conexión, distancia, parecido, etc.). En la Figura 6 se muestra una visualización desarrollada por Fry para visualizar el tráfico dentro de un sitio web. Cada nodo es una página y su tamaño la cantidad de visitas. Los ejes blancos que las unen representan la estructura del sitio, mientras que los semi-opacos los caminos tomados por los usuarios.

Otro tipo de visualización que nos hace pensar en los elementos visuales como entes “autónomos” es el *packed bubble chart* (Figura 7, siguiente página). Este simple diagrama es equivalente a un histograma de variables categóricas donde el valor está representado por el área de un círculo. Los círculos son luego mapeados en la pantalla de forma que no se superpongan y que ocupen el menor espacio. Esta visualización es en ciertos casos mejor que un histograma de barra clásico, ya que si las categorías no tienen una relación de orden entre ellas, la posición arbitraria de los círculos no fuerza ninguna sobre la percepción de ellos (ver Capítulo 6). El *packed bubble chart* puede ser entendida desde este paradigma como una visualización orgánica porque los elementos visuales nos recuerdan a células o, análogamente, a un corte transversal de una planta con varias ramas que ascienden verticalmente, porque captura el comportamiento de “crecer” cuando su valor aumenta.



## **2.4. TECNOLOGÍAS UTILIZADAS**

En esta sección se describen los lenguajes de programación en los que se implementó la solución y la justificación de su elección.

### ***Python***

Python es un lenguaje ampliamente usado por su sintaxis simple que preferencia la legibilidad del código, con un sistema de tipos dinámico y manejo de Memoria automático. Tiene una comunidad activa y una inmensa cantidad de librerías gratuitas son desarrolladas y mantenidas. Fue escogido para cumplir un rol menor en el preprocesamiento de los datos porque posee una librería para leer JSON que hace el trabajo de recibir el archivo que entrega Drive.

### ***Processing***

Processing es un lenguaje de programación y IDE construido para las artes digitales, mediáticas y el diseño visual. Creado por Ben Fry y Casey Reas, basado en Java y con una sintaxis muy similar, provee primitivas visuales que hacen que dibujar a través de la programación sea algo intuitivo. En este lenguaje se desarrollaron la estructura de datos y el prototipo presentados en los capítulos 3 y 4.



### 3 Visualización

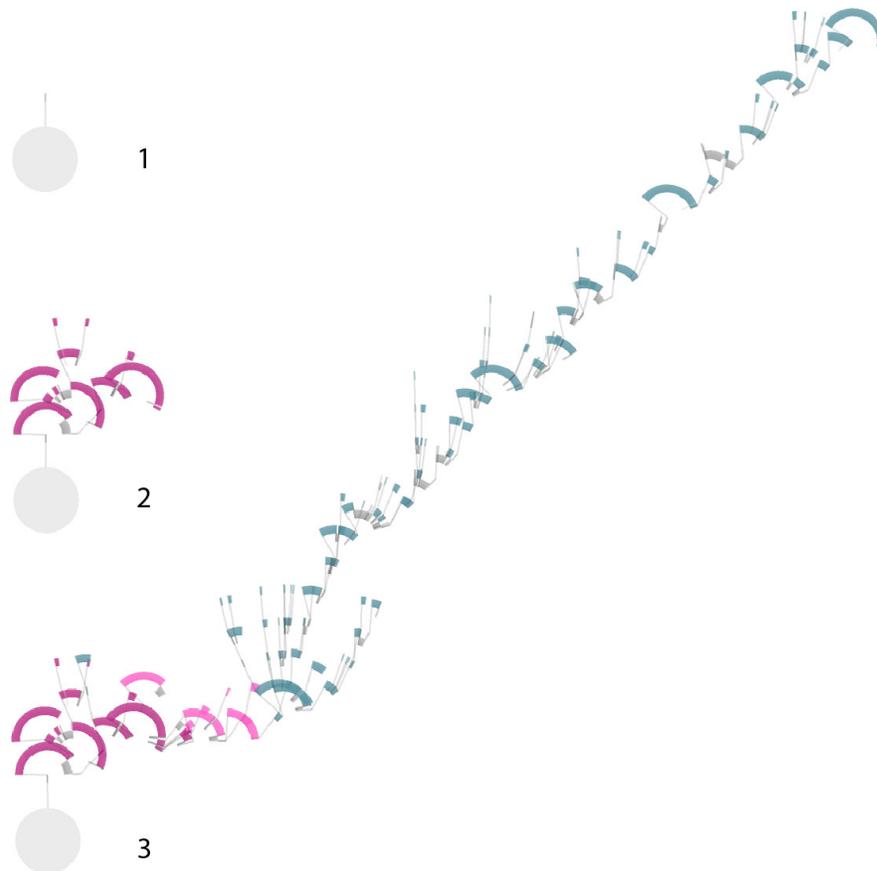


FIGURA 8. THOM de este capítulo en tres momentos de su desarrollo: (1) documentos vacío; (2) primera sesión; (3) tercera sesión.

Gracias a la noción de diagrama, que permitió reconocer las visualizaciones de texto versionado encontradas en literatura como soluciones lineales y distinguir un enfoque diferente, fractal, se llegó a una solución cuya estructura no está dada de antemano, sino que se genera a través de su propio glifo, el cual captura las microdecisiones del usuario, y se inspira en el paradigma orgánico del diseño de información. En esta, cada inserción es capturada en un glifo, el cual es la base para nuevas inserciones, y así recursivamente, formando un árbol, como se ilustra Figura 8. No es posible sacar un glifo de la estructura: al borrar su contenido este se oscurece, queda inactivo, mas no desaparece.

En esta sección se describe el diseño del prototipo desarrollado sobre la estructura de datos abstracta, al que se ha tomado la libertad de llamar “mapa de operaciones topológico-histórico” o THOM: *Topological-Historical Operation Map*. También se dan a conocer las ideas que precedieron al diseño final.

### 3.1. EVOLUCIÓN DEL DISEÑO

En el comienzo de la fase de diseño, no se tenía claro aún cómo serían satisfechos los requisitos impuestos en la propuesta, por lo que durante el diseño se dio inicio a un proceso de experimentación visual que culminó en la idea general que luego sería desarrollada durante la fase de implementación.

La primera visualización producida (Figura 9) tenía aún características muy lineales. Consistía básicamente en un gráfico de dispersión en que la posición en el texto está en el eje X y la revisión en el Y, dibujándose todo el estado del documento en cada fila. Cada carácter está codificado en un punto semi-opaco, mientras que los más oscuros representan las ocurrencias de signos de puntuación, produciendo la forma de caminos en el tiempo que señalan los lugares de inserción en sus bifurcaciones. Este prototipo tenía un serio problema con el uso del espacio y no parecía dar pie a interacciones interesantes, además que no dejaba claro ninguna estructura del proceso.

Se tomó luego un desvío hacia un nuevo prototipo que objetificaba la operación misma, representando cada una como un círculo semi-opaco cuya área indicaba la cantidad de caracteres involucrados, como se ve en la Figura 10. Los azules eran inserciones y los rojos eliminaciones. Estos se iban apilando hasta que aparecía una operación del tipo contrario, entonces se empezaba a hacer una nueva pila al lado de la otra. Esto da una cierta sensación del ritmo de la escritura, pero por supuesto el hecho de que no hubiera relación entre la posición de los círculos y el índice de la operación hacía que la información representada en esta visualización fuera demasiado escasa. Empezar a pensar el documento en términos de operaciones fue, sin embargo, crucial para llegar a la solución.

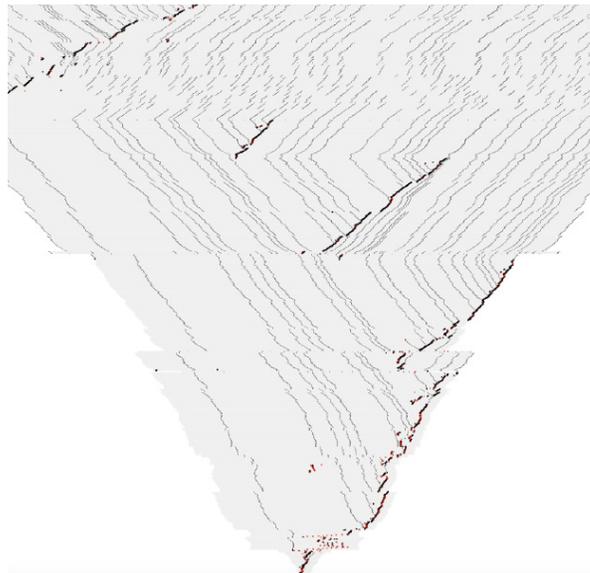


Figura 9. Primer diseño. El tiempo corre hacia arriba y el documento completo es dibujado en cada fila. Los puntos negros que dan la sensación de caminos codifican los signos de puntuación y los más gruesos los lugares de operación.

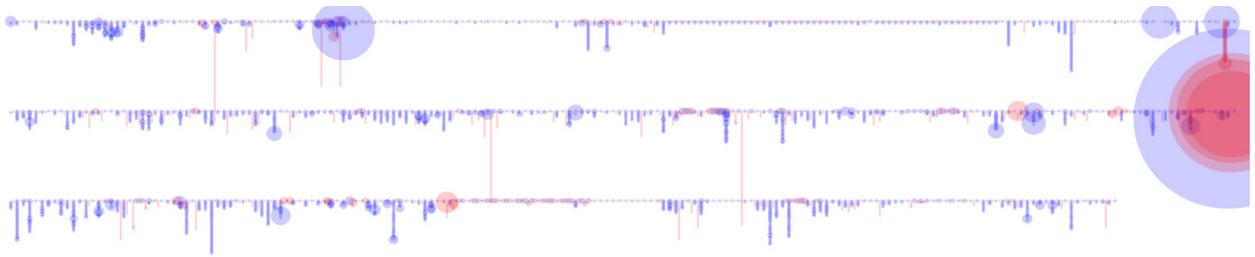


Figura 10. Segundo diseño. Cada círculo codifica una operación: azul para inserciones y rojo para eliminaciones; su tamaño la cantidad de caracteres involucrados. El tiempo va de arriba a abajo y de izquierda a derecha, organizándose en filas.

El diseño final es una mezcla entre estas primeras dos ideas, que combina la encapsulación de la operación alcanzada en el segundo prototipo con la idea de bifurcación o anidación del texto que se mostró de manera muy primitiva en el primero. En la Figura 11 se muestran diferentes experimentos que ya se aproximaban a lo que sería la solución. Fue aquí, al desligarse del diagrama lineal del tiempo y el espacio, en que se materializó la idea del texto que se va construyendo de una manera más bien fractal. También fue durante esta fase del prototipo que se comienza a desarrollar paralelamente la estructura de datos descrita en el capítulo anterior.

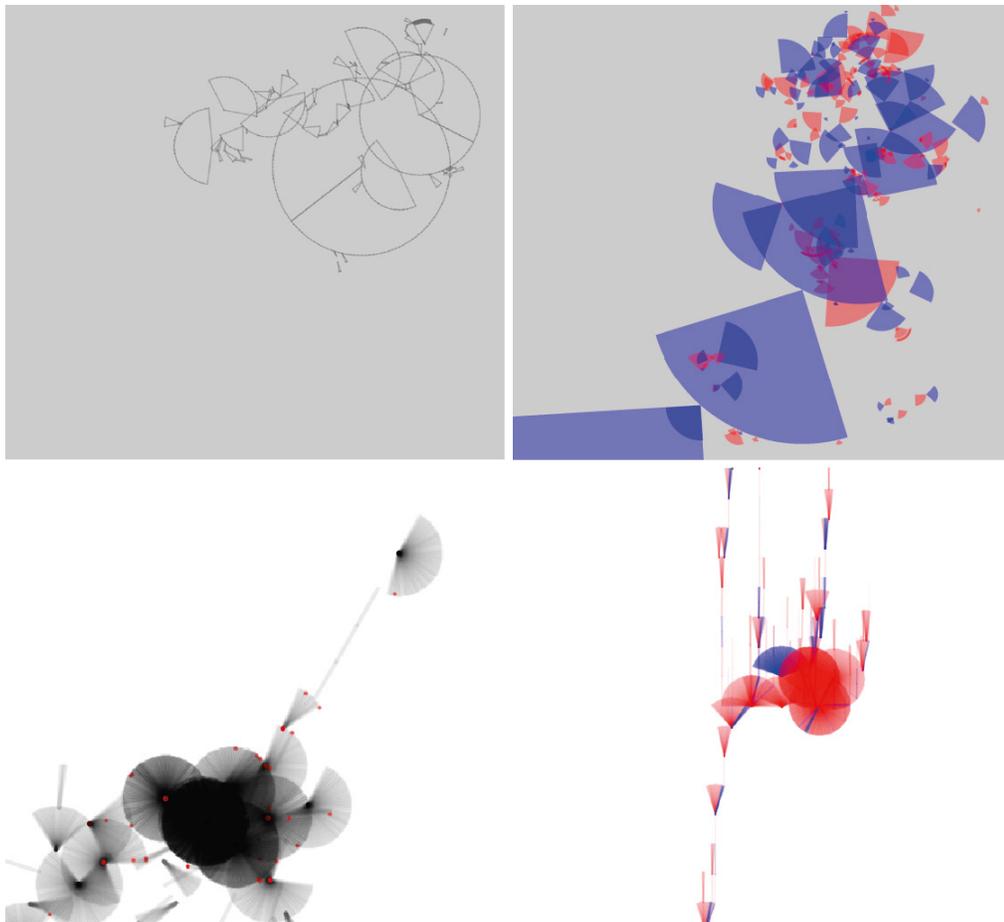


Figura 11. Diversos diseños que precedieron al prototipo final.

## 3.2. DISEÑO FINAL

El diseño final es una versión de los prototipos de la Figura 11. Utiliza un diagrama lineal, con un glifo en forma de paracaídas que representa la operación de inserción capturando el texto insertado, pero que anida las eliminaciones que se realizan sobre él.

### 3.2.1. Diagrama

Para el caso de este trabajo, era claro que un diagrama lineal no satisficaría las condiciones necesarias para siquiera poder mostrar efectivamente la cantidad de información que se tenía a mano, lo cual proponía un espacio de interesante desarrollo para la visualización. El diagrama descentralizado era ciertamente la solución, y se pensó usar un espacio adaptativo, con una simulación tipo cultivo de bacterias por ejemplo, pero esto no prosperó y la idea de que lo que se estaba buscando era una especie de fractalidad en la escritura del texto llevó a la noción de árbol, dejando claro que el diagrama debía ser uno recursivo, donde los elementos se posicionan siempre de acuerdo al elemento relativo anterior. Posteriormente se desarrolló la versión final de la estructura de datos, la cual resultaba ya no ser un árbol sino un DAG, pero el diagrama recursivo se mantuvo ya que estaba bien avanzado y aunque no captura toda la información de la estructura sería de todas maneras un primer paso para generar un prototipo.

Se decidió por trabajar en el espacio bidimensional plano en vez de usar uno tridimensional por la obvia comodidad visual que este espacio ofrece, siendo innecesario además de costoso en un principio agregar una tercera dimensión que sólo permitiría dar más holgura a los elementos, pero que trae consigo una inherente oclusión entre ellos debido a la perspectiva. El espacio bidimensional es más elegante, remite mejor a la idea de texto y es más fácil de imprimir como imagen, dando la posibilidad de tener una sola vista general de todos los datos cuando en el espacio tridimensional es necesario más de una.

### 3.2.2. Glifo

El glifo es, para el caso de esta visualización, la única marca que conlleva significado, por lo que, después de la estructura de datos (Capítulo 4), es lo más importante para la solución. Podríamos catalogar al tipo de glifo aquí descrito como un “glifo recursivo”, ya que parte de su diseño es irse componiendo a de otros glifos que a su vez siguen la misma lógica.

El diseño del glifo consideró la idea de “apertura del espacio” como su principal motivación, ya que cada glifo representa un cambio (localizado) en el documento. La información que codifica el glifo debía ser mínimamente: (1) largo del *string* insertado en la operación y (2) lugar de operación donde tuvo origen.

La primera forma que vino a satisfacer estas condiciones fue la del triángulo, el cual puede ser visto como un segmento bidimensional expandiéndose o contrayéndose (hasta desaparecer) de manera constante en el tiempo. El triángulo, sin embargo, planteaba la pregunta de qué parte de este representaba exactamente el texto contenido (¿el lado, el área, el borde?), además que en su crecimiento

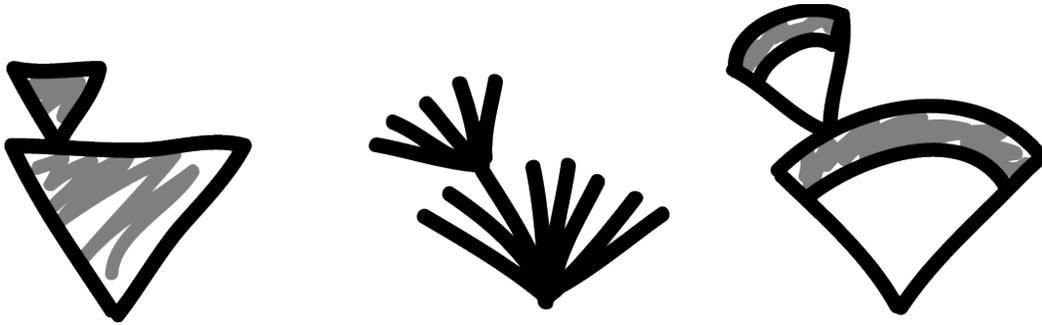


Figura 12. Evolución del glifo. De izquierda a derecha: triangular; diente de león; paracaídas (actualmente en uso).

lineal (en una sola dirección) no hace un uso eficiente del espacio. Para resolver el primer problema se pasó a un diagrama radial, donde los ejes salen del centro en forma que recuerda a un diente de león. En este diseño, el largo de *string* está mapeado al ángulo de apertura, y presenta la característica de crecer no linealmente, es decir, de hacer un uso más distribuido del espacio. Sin embargo, también presenta el problema de que si el ángulo entre ejes es constante puede llegar a cerrar la circunferencia, lo cual produce una “indefinición” de la representación. La solución a esto la dio usar arcos cuyo largo representa el tamaño del *string*, con unas líneas, en forma de paracaídas, que van desde los extremos del arco y convergen en el punto de inserción. Este glifo mantiene lo beneficios de un diagrama radial, pero también permite que jugando con el radio se puedan alcanzar cualquier largo sin superar un cierto ángulo de apertura. En la Figura 12 se ilustran los glifos mencionados.

El tratamiento que se le dio a la eliminación es el siguiente: cuando una parte o la totalidad del *string* contenido en un nodo es eliminada por una operación posterior, el segmento del arco que corresponde a esa parte se separa del resto y se “hunde” un poco hacia el origen del nodo, además de colorearse de manera diferente, como se ve más abajo, para que sea más fácilmente identificable. Como un nodo de eliminación (en la estructura de datos) también tiene un lugar de operación, un nuevo glifo puede aparecer al comienzo de un segmento eliminado, en caso de que algo sea insertado en esa posición.

Se agregó además un círculo gris claro para señalar la raíz del árbol, a manera únicamente de ayuda visual.

### 3.2.3. Color

El color como codificación de algún atributo de los datos se pensó exclusivamente para aplicarse en el glifo, ya que el fondo no tiene mayor sentido en esta visualización. Como el tiempo estaba quedando fuera de la representación “topológica” del documento, se decidió utilizar el color para codificar un orden temporal, aunque parcial, en las operaciones.

Las opciones fueron: (1) mapear el atributo temporal de manera continua a una escala de color, o (2) usar una gama de colores discreta para distinguir operaciones ocurridas en diferentes sesiones. La primera opción parecía atractiva pero tiene algunas complicaciones, a saber, el tiempo de

edición en principio no tiene límite y los intervalos entre operaciones van del rango de segundos a días, por lo que una interpolación lineal entre dos colores no daría buenos resultados. Por esto, se decidió por generar una gama de colores para ser usados en secuencia a través de intervalos de tiempo (sesiones) y que determinan un orden parcial entre las operaciones. Un cambio de sesión se produce (sólo para efectos de determinar el color) cuando ocurre un intervalo de al menos 12 horas entre una operación y la siguiente.

Diferentes gamas de color fueron probadas y por esta razón puede que en las figuras en las que se muestra la visualización no siempre estén en uso los mismos colores. Finalmente se optó por una gama circular de ocho colores inspirada por las estaciones del año. Cada estación (primavera, verano, invierno y otoño) tiene dos colores asociados que se siguen y dan paso a la próxima estación. Se apuntó a que hubieran suficientes colores para que no se repitieran con demasiada frecuencia pero que a la vez fueran suficientemente diferenciables uno de otro. Los textos comienzan a escribirse en primavera. La Figura 13 ilustra la gama de colores. Esta es una solución simple para el problema del tiempo, el cual *a priori* es indefinido (no se sabe en un comienzo cuánto tiempo abarca la

producción de un documento), ya que le da una forma a la vez cíclica e intuitiva para cualquier posible usuario. Hay que notar que puede que la gama elegida no sea amigable para daltónicos, pero constatar eso requiere de un estudio especializado que se ha dejado fuera del alcance de esta memoria.

Se usa transparencia en los colores para que se haga más evidente cuando un arco está sobrepuesto a otro. Para el color de fondo se decidió usar blanco para que la transparencia en los colores transluciera mejor y porque así es más apto para ser impresa en páginas blancas.



Figura 13. La gama de colores fue inspirada en las cuatro estaciones. Los colores parten en la primavera y se suceden en sentido antihorario.

### 3.2.4. Variables globales

Existen dos parámetros del árbol que decidieron dejarse como variables globales que pueden ser afectadas dinámicamente: los factores de apertura y de fototropismo. Manejando estos dos parámetros (ver Sec. 3.2.6) se puede reducir la oclusión local y globalmente en un árbol, quedando a discreción del usuario ya que ninguna combinación de valores es la óptima para todos los datos.

### ***Factor de apertura***

La distancia visual que se establece entre un lugar de operación y otro en el mismo nodo es una medida arbitraria que define cuán largo será un arco de acuerdo a su cantidad de caracteres. Al afectarla se consigue que el árbol se “cierre” o se “abra” lo que puede producir una mejor vista general, dependiendo de la disposición de los nodos.

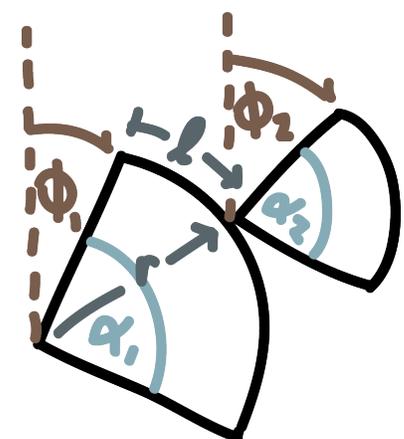
### ***Factor de fototropismo***

Fototropismo describe el comportamiento de las plantas de crecer u orientarse hacia el sol u otra fuente de luz. El factor de fototropismo fue introducido en un momento del diseño en que dos modelos estéticos entraron en conflicto. En uno, los nodos se orientaban siempre hacia arriba, tal como en el glifo triangular de la Figura 5.5. En el otro, los glifos se orientaban en un ángulo que dependía de todos sus relativos anteriores. Ninguno de estos modelos extremos resultaba ser muy adecuado ya que en el primero comprimía el árbol a un espacio vertical donde todas las ramas son paralelas, produciendo mucha oclusión y árboles demasiado altos, es decir, una mala relación dimensional o razón de aspecto (*aspect ratio*), mientras que en el segundo las ramas se dan vuelta sobre sí mismas de forma elíptica produciendo nuevamente oclusión e ilegibilidad. La solución fue entonces hacer un parámetro que pudiera variar continuamente entre un extremo y otro, es decir, un factor que tomara en cuenta la desviación de la vertical de cada nodo para darle a cada una una pequeña orientación hacia arriba. En un valor intermedio esta solución reduce la oclusión y produce una sensación orgánica.

### **3.2.5. Posicionamiento**

A diferencia de un *scatterplot*, donde los puntos (o, en general, glifos) son posicionados de manera independiente en el espacio de acuerdo a sus coordenadas, las cuales codifican parte de su información, o a un *force-directed layout*, donde la posición es definida por una simulación de fuerzas donde se hace “interactuar” a los nodos pero no codifica información, el diagrama que se propone ubica a los glifos de manera que un nuevo nodo siempre es relativo a su nodo padre (anterior relativo). La posición de un glifo queda definida de manera absoluta por: (1) el radio y ángulo de su anterior relativo y (2) su posición (índice de inserción) dentro de su anterior relativo. La Figura 14 ilustra el posicionamiento. Notar que los ángulos de apertura y con respecto a la vertical son controlados por el factor de apertura y de fototropismo respectivamente. Notar que la distancia de inserción del nuevo nodo también depende del factor de apertura.

Figura 14. Esquema del posicionamiento recursivo de glifos.  $l$  es la distancia de inserción,  $r$  el radio del glifo anterior relativo,  $\phi$  el ángulo con respecto a la vertical y  $\alpha$  el ángulo de apertura.



### 3.2.6. Interacción

Las componentes interactivas de la visualización fueron siendo añadidos en la medida que fueron requeridas, por lo que responden a las necesidades de uso de la visualización. La más importante es la capacidad de ocultar nodos: esto viene de la necesidad de filtrar datos para poder inspeccionar sólo aquellos de interés. Cada nodo guarda un estado binario el cual dice si está oculto o mostrándose. Por defecto todos los nodos están mostrándose, pero ocultarse uno todos sus hijos cambian de estado con él, permitiendo despejar la visión y discernir ramas fácilmente. La Tabla 1 contiene las funciones con su comando asignado y descripción. El prototipo no considera botones gráficos, por lo que las funciones están asignados al teclado o al ratón.

Función	Comando	Descripción
<b>Panear</b>	Arrastrar con botón izquierdo del ratón	Mover la cámara de forma paralela al plano de visualización.
<b>Zoom</b>	Girar rueda del ratón	Acercar o alejar la cámara.
<b>Restablecer cámara</b>	Doble click	Devuelve la cámara a su posición y nivel de <i>zoom</i> original.
<b>Inspeccionar</b>	Click sobre un nodo	Imprime en consola el contenido del nodo seleccionado y de todos sus hijos no ocultos.
<b>Pausar/Continuar</b>	Espacio	Mientras se está leyendo el <i>changelog</i> del documento, detiene y reanuda el crecimiento del árbol.
<b>Ocultar/Mostrar</b>	Click derecho sobre un nodo	Oculto este y sus hijos. Al quedar oculto, aparece un círculo rojo en su lugar, que al ser vuelto a clickear trae la rama de vuelta.
<b>Aumentar/Disminuir apertura</b>	Flechas derecha/ izquierda	Aumenta o disminuye de manera continua el factor de apertura.
<b>Aumentar/Disminuir fototropismo</b>	Flechas arriba/ abajo	Aumenta o disminuye de manera continua el factor de fototropismo.
<b>Exportar</b>	P	Guardar vista actual en formato PNG.

Tabla 1. Funciones interactivas.

Se puede acceder a un video en Internet de un árbol en crecimiento escaneando el código QR de la Figura 15.

Figura 15. Un árbol en crecimiento.  
<https://drive.google.com/open?id=0B3CAAdHkf9F8iV111Tnp6RnlwRFk>



## 4 Estructura de Datos

Central para este trabajo fue encontrar una representación de los datos adecuada para su visualización. Esto significó generar una estructura de datos que mediara entre el historial ya parseado y la forma de la visualización. La estructura que se encontró más adecuada es la de un grafo dirigido acíclico, cuya operación, restricciones e implementación se detallan en este capítulo. Este se divide en *Representación*, donde se presentan los requisitos que el diseño de la estructura de datos debía considerar, *Estructura de datos abstracta*, donde también se explica el proceso para llegar a esta solución, e *Implementación*, donde se explica la forma en que esto se llevó a cabo.

### 4.1. REPRESENTACIÓN

El objetivo de buscar una nueva representación para el historial de un documento viene de la necesidad de generar un nuevo entendimiento (entendimiento entendido, al menos, como una forma de articular la información) de los datos a los que se enfrenta este trabajo. En este sentido, no difiere en lo absoluto del objetivo de la visualización misma, y puede considerarse que la estructura conlleva de manera casi directa su manera de mostrarse. Por lo menos en este trabajo, visualización y estructura de datos fueron desarrollados de manera paralela, el progreso de cada uno retroalimentando el del otro. La representación buscada debía cumplir con tres requerimientos, dan pie a la estructura de datos abstracta que se define en la siguiente sección:

1. Modelar un documento como una estructura dinámica, en oposición a la visión clásica del texto como algo estático, finalizado (de tal manera que fuese posible representar estados intermedios).
2. Modular esta estructura de acuerdo a los ritmos y decisiones del usuario que escribe.
3. Articular en el espacio las nociones anteriores produciendo una idea topológica del documento.

Las soluciones adoptadas en respuesta a estos requerimientos serán enunciadas como una serie de principios correspondientes:

1. El documento se divide en el tiempo en una serie de operaciones, no estados.
2. Las operaciones pueden agruparse de acuerdo a diferentes criterios como linealidad, intervalo temporal, etc.
3. Las operaciones o grupos de operaciones se anidan dentro de las previas inserciones de texto, que se puede interpretar como que “abren” espacio en la página para nuevas operaciones, produciendo una jerarquía, mientras que las eliminaciones lo “cierran”, encapsulando de manera permanente un fragmento del espacio y destruyendo parcialmente la organización jerárquica.

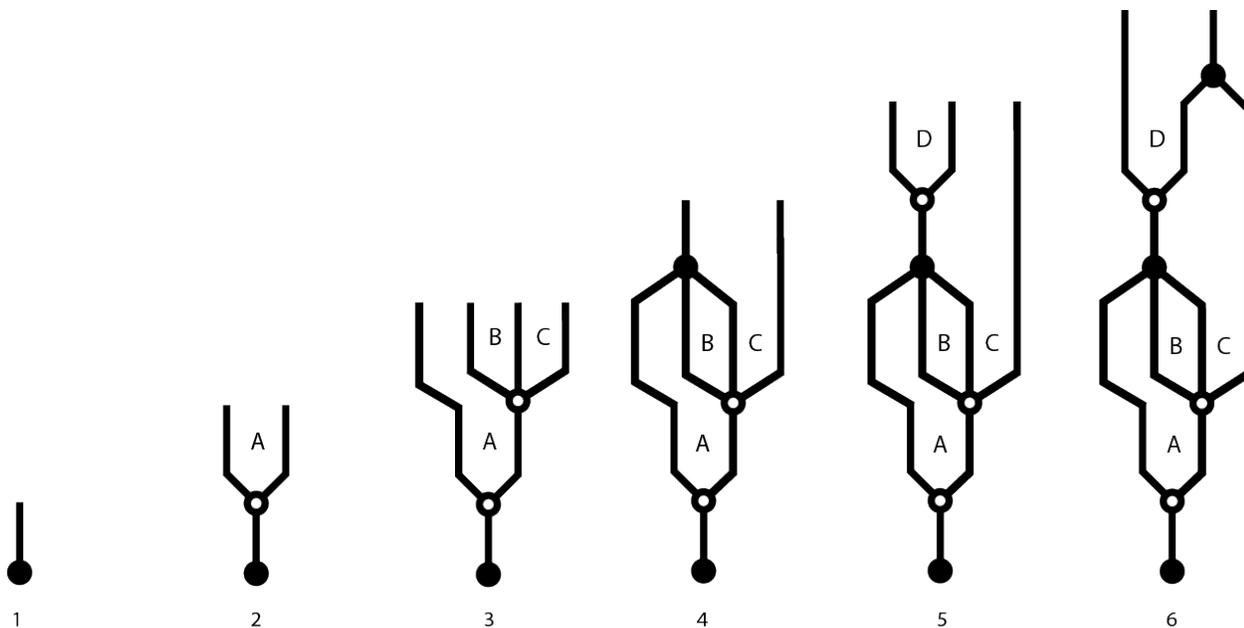


Figura 16. Esquema progresivo de un documento. Los nodos de eliminación se representan con relleno negro y los de inserción con blanco, los lugares de operación como ejes que no van a ningún nodo. La dirección de los ejes siempre es “hacia arriba”. (1) Documento vacío. (2) Inserción de “A”. (3) Inserción de “BC” en la segunda posición, resultando en el string “ABC”. (4) Eliminación simultánea de “AB”. (5) Inserción de “D” antes de “C”, resultando en el string “DC”. (6) Eliminación de “C”. El documento resultante sólo contiene el string “D”.

## 4.2. ESTRUCTURA DE DATOS ABSTRACTA

Aunque en un comienzo se pensó que la estructura dinámica del documento se asemejaba a la de un árbol, en la que el texto se veía crecer hacia diferentes ramas donde cada nodo representaba una cadena de caracteres y sus hijos las cadenas que le siguieron topológicamente, se llegó a la conclusión de que este modelo no era adecuado cuando se incluía el caso general de la operación de eliminación, la cual no respeta la estructura jerárquica, pudiendo actuar en diferentes nodos al mismo tiempo. El error de modelar el comportamiento del documento con árboles fue inducido por la idea intuitiva pero equivocada de que son los caracteres los que producen la apertura del espacio al teclear. En ese caso, ¿qué es lo que produce el consecuente cierre del espacio al borrar? Consideremos qué es el documento antes de que haya nada escrito en él: no es sólo una página en blanco, ésta considera la posibilidad (única) de una apertura, materializada en la forma del puntero. *El puntero es el origen y la condición estructural de un documento, pues es el lugar de operación.* A su vez, cada operación produce una modificación en la disposición de espacios de operación: una inserción es la división o apertura del espacio donde tuvo origen. Análogamente, la borradura es la unión o cierre de dos o más lugares de operación en uno. Los lugares de operación en un documento son siempre igual a  $n + 1$ , donde  $n$  es el número de caracteres. Con esto se puede observar que si nos restringimos al caso de un texto formado exclusivamente de inserciones, es decir, en el que no haya tenido lugar una operación de eliminación, la estructura consecuente tendrá la forma de un árbol, donde cada nodo representa una inserción y es hijo del nodo del cual se abrió el lugar de operación donde se insertó. La eliminación puede, naturalmente, hacer desaparecer esta estructura:

borrar todo el texto, por ejemplo, significa tomar todos los lugares de operación de todos los nodos y “atarlos” de vuelta en un único lugar de operación, que reproduce el estado inicial de la página en blanco.

La estructura que se nos presenta entonces es la de un grafo dirigido acíclico (DAG por sus siglas en inglés) en el que cada nodo representa exactamente una operación. El grafo es dirigido porque hay una sola dirección (hacia “arriba” en la representación bidimensional mostrada en la figura más abajo) del tiempo y acíclico porque de haber ciclos significaría que se podría pasar un número cualquiera de veces sobre un mismo set operaciones, lo cual no es cierto para el caso general. No es coincidencia que los DAG sean utilizados para representar relaciones de causalidad (red bayesiana) y que sean el modelo de los Sistemas de Control de Versiones como Git. Git es en cierto sentido un caso muy generalizado de la estructura que acá se describe, con la diferencia que en este sistema cada nodo guarda una versión estática del documento en vez de las operaciones realizadas sobre este, las cuales no pueden ser recuperadas mediante el diff entre dos versiones (basta imaginar el caso de algo que es insertado y borrado entre una versión y otra). Esta diferencia, sin embargo, produce que las operaciones sobre estas dos estructuras sean completamente diferentes. Basta considerar la operación de lectura: en Git, cada nodo es un documento completo, por lo que para recuperar la versión actual, por ejemplo, sólo es necesario acceder a uno de estos, no habiendo mayor sentido en recorrer toda la historia más que para obtener todas las versiones, las cuales son, en realidad, independientes. Mientras, en el esquema que se está proponiendo, para recuperar la versión actual del documento es necesario recorrer todo el grafo ya que toda su información es necesaria, es decir, los nodos no tienen sentido de forma independiente sino dentro de la estructura total. Esto hace que lo único que compartan estas dos estructuras es el diagrama en forma de DAG, pero que su operación e interpretación sean dos cosas distintas.

La Figura 16 muestra un esquema de la estructura de un documento sometido a algunas operaciones. En ella se observa que existen ejes que no llevan a ningún nodo. En realidad, estos están puestos únicamente con fines ilustrativos, ya que no existen más que como posibilidad antes de que ocurra una operación en su lugar. Otra particularidad del grafo es que un caracter está “borrado” cuando los dos ejes que lo rodean se juntan en un mismo nodo de eliminación. De lo contrario, es decir, mientras el espacio en el que está se mantenga “abierto”, el caracter se encuentra presente en el documento. Por eso, en este trabajo se llama *superficie* a la parte del documento que no ha sido aún borrada. También se observa que que los ejes, en el espacio entre que son creados o eliminados, siempre se mantienen paralelos unos a otros. Esta restricción se rompería si, por ejemplo, se empezara a considerar el “mover” texto como una operación posible.

#### **4.2.1. Operaciones**

A continuación se describen las operaciones que se pueden realizar sobre esta estructura de datos. Como su principal función es guardar un documento con su historia, las operaciones básicas son inserción, eliminación, lectura en superficie y lectura en profundidad.

*Inserción.* La inserción ocurre en un sólo lugar de operación determinado, aunque puede ser multicaracter. Insertar requiere de buscar el nodo en el grafo en el cual ocurre la inserción, que va a ser el padre del nuevo nodo de inserción. Como no se puede insertar *entre* nodos, la operación de inserción es relativamente simple.

*Eliminación.* La eliminación ocurre en un rango de lugares de operación, por lo que puede afectar a más de un nodo. El nuevo nodo, es hijo de los nodos a los que afecta, lo cual no significa necesariamente que borre algo dentro de ellos: como el eliminar produce una cerradura sobre los lugares de operación puede que, por ejemplo, cierre el espacio justo entre dos nodos hermanos, caso en el que el contenido borrado sólo pertenece al padre, pero el rango de lugares de operación abarca los nodos hijos por lo que el nuevo nodo es hijo de los tres recién mencionados. Más aún, el espacio entre estos nodos converge a un sólo lugar de operación, es decir, el espacio entre los nodos hermanos desapareció y dos lugares de operación que pertenecían a dos nodos diferentes se fusionaron en uno sólo que ya no pertenece a ninguno de los nodos originales sino al nuevo nodo donde se produjo la eliminación.

*Lectura en superficie.* Es producir el estado actual del documento, en otras palabras, el string tal como se vería en el editor de texto.

*Lectura en profundidad.* Es producir un string que represente sin ambigüedad la estructura y el contenido del grafo. En un árbol esto es simple de lograr usando paréntesis que abren y cierran en cada nodo y conteniendo a su vez los nodos hijos de este. Para el caso de la eliminación, se añadió otro tipo de paréntesis que encierran el contenido borrado por un nodo de eliminación, y que seguidamente abren un nuevo paréntesis que señala el lugar de operación del nodo de eliminación. Para el caso ilustrado en la Figura 3.1, por ejemplo, la codificación para cada instante, usando llaves para los nodos y corchetes para las eliminaciones, sería (añadiendo el identificador correspondiente a cada operación después de la apertura de su paréntesis):

```
{ 1 : }
{ 1 : { 2 : A } }
{ 1 : { 2 : A { 3 : BC } } }
{ 1 : { 2 : [ A { 3 : B } { 4 : } C ] } }
{ 1 : { 2 : [ A { 3 : B } { 4 : { 5 : D } } C ] } }
{ 1 : { 2 : [ A { 3 : B } { 4 : { 5 : D [ ] } C ] { 6 : } } } }
```

La efectividad de esta codificación queda más clara si se pone un # en cada lugar de operación vacante:

```
{ 1 : # }
{ 1 : { 2 : # A # } }
{ 1 : { 2 : # A { 3 : # B # C # } } }
{ 1 : { 2 : [ A { 3 : B } { 4 : # } C # ] } }
{ 1 : { 2 : [ A { 3 : B } { 4 : { 5 : # D # } } C # ] } }
{ 1 : { 2 : [ A { 3 : B } { 4 : { 5 : # D [ ] } C ] { 6 : # } } } }
```

## 4.3. IMPLEMENTACIÓN

La estructura de datos fue implementada en *Processing*, con un patrón de diseño orientado a objetos. Las primitivas visuales de este lenguaje hacen posible que cada objeto pueda tener un método para mostrarse a sí mismo, haciendo de la visualización de la estructura más directa y a la mano.

### 4.3.1. Clases

El sistema se dividió en las clases *Changelog*, *Document* y *Node*. El primero funciona como el *wrapper* del archivo que contiene el log, mientras que *Document* y *Node* forman la estructura de datos que soporta la visualización.

#### *Changelog*

El archivo que se obtiene de *Google Drive* es procesado por un script en *Python* que devuelve una tabla con cada registro de operación en una fila. Para interactuar con este archivo se creó una clase *Changelog*, que lee el archivo completo cuando es instanciada.

Su función es pasar los registros de a uno a *Document*, aunque puede preprocesarlos para entregarlos de una forma más comprimida a la que se llamó *burst*. Un *burst* es una serie de varias operaciones sobre el documentos que ocurren secuencialmente, y que son entregadas por *Changelog* como un solo registro, con el fin de compactar la información que la estructura de datos tendrá que manejar, particularmente el número de nodos. Dónde empieza y cuándo termina un *burst* puede ser implementado de diferentes maneras, dependiendo de qué sentido se le quiera dar, por ejemplo, considerando el tiempo entre operaciones. Para el caso de este trabajo se consideró que un *burst* es una secuencia de operaciones *lineales*, es decir, que ocurrieron sin que hubiera habido un movimiento del puntero hecho por el usuario. Por supuesto, esto lleva a pérdidas de información al pasar del archivo original a la estructura, de la misma forma que se pierde información cuando sólo se guardan las versiones de un documento, pero lo que se pierde en granularidad de los datos se gana en inteligibilidad de la visualización posterior, ya que lo primero que desaparece al pasar de logs atómicos a *bursts* son los errores grafomotores. Queda pendiente hacer una estructura dinámica, que pueda aumentar o disminuir su granularidad según las necesidades del usuario.

#### *Document*

*Document* es la fachada de la estructura de datos. Tiene los métodos para operar sobre la estructura, visualizarla e imprimirla. Contiene una lista con los nodos ordenados temporalmente. Cuando es instanciado crea un primer nodo que representa la creación del documento y que es la raíz única del grafo. Cada vez que ocurre una operación, crea un nuevo nodo cuya referencia añade al final de la lista y envía un mensaje a la raíz con el nuevo nodo para que se inserte de manera recursiva en el grafo.

#### *Node*

Esta clase forma la estructura de datos a través de un patrón compuesto. En sus campos el nodo

mantiene una referencia a otros nodos que son sus hijos a través de ejes representados por la clase `Link`. Cada nodo representa una decisión del usuario y se inicializa con el registro de esa actividad, cuya información usa para posicionarse en el grafo. Sin embargo, una vez posicionado, la única información que no queda capturada en la estructura del grafo en sí es el timestamp y los caracteres particulares insertados. Estos son guardados en los campos del nodo para hacer el texto recuperable. Los nodos, una vez creados y fijados en el grafo, permanecen inmutables. Cada nodo guarda un arreglo de nodos, los cuales son sus hijos. Se usa un arreglo porque la cantidad de posibles links a otros nodos es conocida y es igual a la cantidad de caracteres insertados en la operación más uno. La clase `Nodo` sirve para representar inserciones y eliminaciones.

### ***Link***

Esta clase representa los ejes que unen los nodos en el grafo. Más específicamente, representa un lugar de operación no vacío, es decir, donde ocurrió una actividad por parte del usuario; un lugar, por lo tanto, que lleva a otro nodo del grafo. Un link empalma entonces un nodo con otro en una cierta dirección, porque se trata de un grafo dirigido, y en una cierta posición particular, porque al ser los ejes paralelos en dos dimensiones están sometidos a un orden. Esta clase se creó justamente para poder mantener ese orden: si los nodos sólo tuvieran referencias a otros nodos, no sabrían dónde comienza o termina una eliminación, porque esa información puede estar dividida entre diferentes nodos y como la comunicación sólo va hacia abajo en la jerarquía (arriba en el árbol) sólo se conocería hasta llegar al nodo de eliminación, el cual tendría que devolver la llamada o hacer algo no recursivo para los nodos de más atrás pudieran tener una visión global del asunto. `Link` viene a encapsular esa información y la pone a disposición para mantener la dinámica recursiva en el manejo del DAG. Tiene en sus campos el nodo al que lleva y su posición entre los otros links que llevan al mismo nodo (recordar que esto es útil cuando se trata de un nodo de eliminación, que tiene más de un padre), además actúa como un proxy para este, atendiendo a las llamadas recursivas que hace el nodo padre, tomando alguna decisión usando su información y pasándola luego a su nodo. Su rol es de particular importancia al hacer que el grafo se muestre, ya sea en la visualización o para imprimirlo en forma lineal en consola.

## 5 Casos de estudio

En este capítulo se hace un pequeño estudio de varios casos (documentos) usando el prototipo descrito en el Capítulo 3 y 4. Primero se habla del *dataset* con el que se trabajó, luego de la metodología que se estableció para analizar los casos y luego se detalla cada caso de estudio.

Se analizaron variados textos de carácter expositivo, que van en extensión de una a nueve carillas, han sido escritos por uno o dos autores y en diferentes plazos de tiempo. Fueron encontrados diversos patrones de escritura (lineal, borrador, estratégico) y en todos estos se pudo advertir la relación entre las diferentes partes del texto, las cuales a su vez estaban bien delimitadas por el elemento visual emergente “rama”. Hubieron también casos que se consideraron demasiado “poco legibles” y que sirven para mostrar las falencias actuales de la visualización y hacia dónde podría seguir su camino para mejorar. Los resultados obtenidos con este análisis son principalmente un éxito para la estructura de datos subyacente, la cual aún contiene más información de la que la visualización está siendo capaz de mostrar hasta el momento. Lo que revela este análisis, es que hay meta-información sobre el texto que se puede extraer sólo mirando las posiciones de operación del proceso de producción de un documento, en otras palabras, que hay una estructura mayor a la sintáctica que se refleja en la combinación de movimientos que producen los usuarios al escribir un texto. Esta estructura emerge al traspasar el documento-en-producción a la estructura de datos, de lo cual la visualización es un reflejo.

### 5.1. DATOS

En esta sección se describe cómo pueden ser obtenidos los datos de un documento en Drive, el procesamiento que se hace para poderlos visualizar y por último de la generación y composición del dataset.

#### 5.1.1. Descarga de los datos

El historial de un documento creado en Google Drive puede ser descargado usando el API REST de Google Docs, para lo cual es necesario conocer dos cosas: el ID del documento y el valor correspondiente a la última revisión. El ID lo podemos obtener de la URL del documento. Obtener el último número de revisión no es tan simple, de hecho, la única manera de encontrarlo es por prueba y error, es decir, por búsqueda binaria en el mejor de los casos. Por suerte, una extensión del browser de Google, Chrome, hace por nosotros este trabajo: Draftback, al tenerla instalada, aparece como un botón en la vista de edición del documento en Docs y calcula el número de revisiones automáticamente. Con esta información, sigue ingresar el siguiente URL reemplazando en la query el ID del documento y el número de revisiones:

```
https://docs.google.com/document/d/{docid}/revisions/  
load?id={docid}&start=1&end={revisions}
```

Por supuesto, es necesario tener acceso al documento para poder descargar su historial, es decir, éste debe estar en nuestra unidad de Drive. Así, se descarga el archivo JSON portador del “changelog”: la información de cada cambio hecho sobre el documento (inserciones, eliminaciones y cambios de formato).

### 5.1.2. Preprocesamiento

El archivo Json que entrega Drive contiene mucha información relacionada a formato del texto y otras cosas que no serían relevantes en primera instancia para este trabajo. Además, la estructura anidada de este formato no era adecuada para su visualización, por lo que se creó un script en Python que extrae las modificaciones al texto (inserciones y eliminaciones) y las guarda en un archivo CSV en forma de lista, cada una como una fila con las columnas *timestamp*, tipo, índice de inicio y contenido textual o índice de término en el caso de las eliminaciones. Este archivo es leído por el programa de visualización en Processing e ingresado línea por línea a la estructura de datos, tal como si fuera un *streaming* del documento siendo producido.

### 5.1.3. Dataset

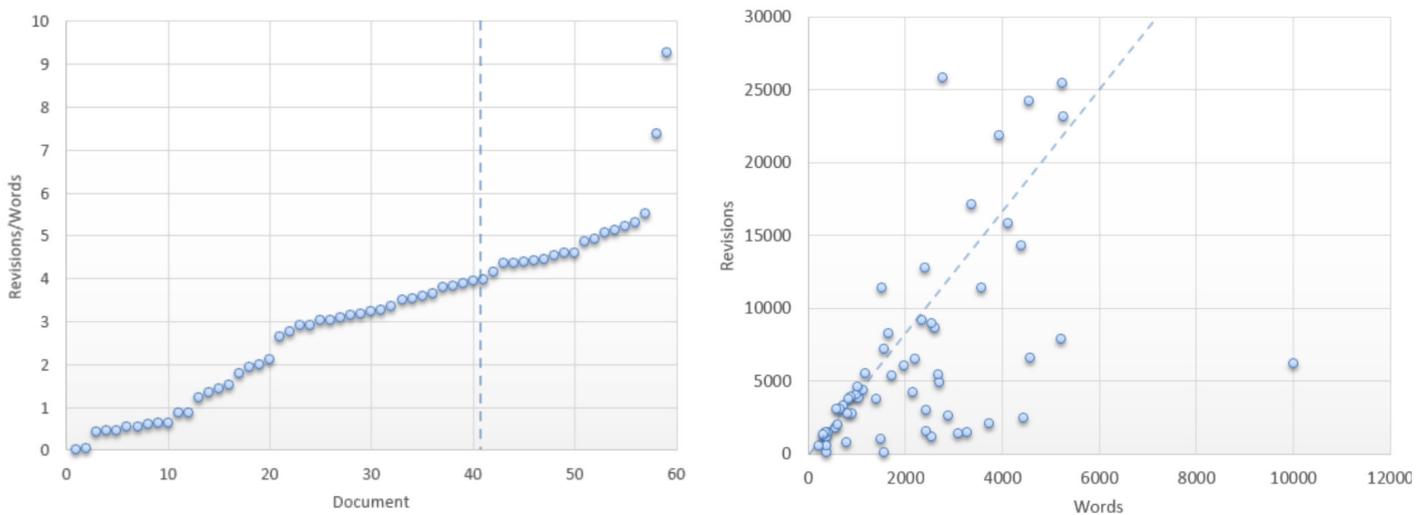


Figura 17. A la izquierda, la altura de cada documento es su “índice de complejidad”, ordenados de menor a mayor. A la derecha los mismos documentos, su posición determinada por los factores del “índice de complejidad”. En ambas figuras, la línea punteada representa el plano de corte, desde el cual los documentos con mayor complejidad pasan a formar parte del dataset.

Los datos explorados en esta sección fueron obtenidos a través de una publicación en U-Cursos en la que se solicitaba a los estudiantes de la Facultad compartir sus documentos hechos con fines académicos en Google Docs. Alrededor de 60 documentos fueron recibidos de diversas personas y especialidades. Muchos de estos, sin embargo, estaban compuestos de imágenes, fórmulas, tablas, o eran demasiado breves como para ser interesantes para un análisis de su proceso de producción. Aún así, todos fueron descargados y se creó en Python un script que calcula el “índice de complejidad” de cada documento, el cual definimos como la razón entre la cantidad de palabras en la versión final y su cantidad de revisiones, en un intento por capturar aquellos en los que se había

invertido más trabajo en proporción a su extensión (y así conseguir documentos de diferentes extensiones). Se tomaron los 20 documentos con mayor índice de complejidad y con ellos se formó el dataset (ver Figura 17).

## **5.2. METODOLOGÍA**

Al ver la forma de las primeras visualizaciones, se hizo evidente que la característica emergente de estas eran las “ramas” que se podían observar en su estructura arbórea. El paso lógico fue entonces pesquisar el contenido textual efectivo de estas ramas. En ciertos casos las ramas no eran tan evidentes en un principio y era necesario usar las capacidades interactivas (reorganizar, ocultar/mostrar) para encontrarlas y aislarlas.

La metodología seguida entonces fue entonces para cada caso:

1. Observar la forma de crecimiento del árbol, para tener una visión de su comportamiento en el tiempo.
2. Con el árbol ya formado, reorganizarlo de forma que las ramas se vean lo más claramente posible, tratando de minimizar la oclusión. Hacer una descripción de las características gruesas que se pueden observar del proceso.
3. Ocultando y desocultando nodos, definir las ramas principales. Luego se inspeccionan y correlacionan con el documento final en Drive, obteniendo un mapeo del árbol al documento.

Siguiendo estos pasos para cada caso se fueron tomando imágenes y evidencia para poder llegar a conclusiones. Para el paso 3 hay que notar que se utilizó Drive para colorear los documentos y capturarlos y Adobe Illustrator para colorear los árboles (lo cual podría estar indicando funcionalidades faltantes del prototipo, como se ve en Evaluación).

## **5.3. CASOS DE ESTUDIO**

En esta sección se analiza caso por caso algunos datos extraídos del dataset (ver Tabla 2 en la siguiente página). No se analizan todos por algunas razones problemáticas: algunos aún eran inadecuados por la cantidad de elementos no textuales que poseían, otros simplemente daban resultados ilegibles al visualizarlos (se muestran al final de esta sección) y unos pocos presentaban inconsistencias al llevarlos a la estructura de datos por errores presumiblemente presentes en el parseo.

Título	Departamento	Curso	Usuarios	Palabras	Cambios
Resumen Charla	DCC	Introducción a la Investigación en Cs. de la Computación	1	312	1307
Reflexión sobre elementos no verbales en la obra: “Los Invitados”	DII	Taller de Oratoria	1	657	3015
Caso de la Presa Vajont: Análisis Axiológico y Reflexión Personal	EH	Ética Profesional de la Ingeniería	1	1567	7136
Descripción proceso industrial División El Teniente	DIQBT	Reactores II	2	1135	4350
El quehacer humano	EH	Filosofía e Historia de la Ciencia: Sociedad y Poder	1	5242	15411

Tabla 2. Documentos del dataset analizados en esta sección y sus características.

### 5.3.1. Caso de estudio 1: “Resumen Charla”

Este caso consta de una breve tarea individual que consistía en resumir una charla dictada por un profesor del Departamento en el contexto del ramo. La Figura 18 muestra el texto y su THOM, previo a cualquier interacción. El texto consta de un título y dos párrafos, cada uno hablando de una parte de la charla (al parecer la charla estaba dividida en dos parte bien delimitadas).

#### Resumen Charla

La primera parte de la charla trató sobre los diferentes niveles de información segura que nos puede dar un lenguaje estáticamente sobre su comportamiento en runtime. Por ejemplo, cuando se declara una función, cuánto es posible especificar sobre ésta que me revele y me asegure su comportamiento dinámico, es decir, tipos de parámetros, tipo de retorno, excepciones que pueda lanzar, input/output, etc. Lenguajes como Python (no tipificados) tienen un sistema de especificación muy pobre; Java se encuentra un poco más arriba (tipificado); lenguajes como Haskell ya nos permiten asegurar más cosas al definir funciones ya que su sintaxis se empieza a parecer más a las definiciones matemáticas y otros lenguajes llevan la especificación al extremo y son capaces de verificar teoremas (esto significa que no pueden existir comportamientos errados). Aún existen unos lenguajes más extraños que admiten diferentes niveles de especificación para mayor flexibilidad durante el desarrollo.

La segunda parte se concentra en la ética de la investigación en ciencias de la computación. Sobre todo en consideraciones que hay que tener a la hora de enviar papers a conferencias o editoriales. Mucho de esto se resume en que hay tener cuidado cuando se cita verbatim a otro autor o se re-usa material de una publicación propia: en el primer caso, es necesario usar comillas para denotar la cita textual, y en el segundo hay que dejar muy claro que se está haciendo una autocita; o si se está postulando a un congreso con un artículo que es una versión un poco actualizada de otra que fue enviada a otra revisión, también hay que especificar que éste es el caso. De lo contrario el artículo será rechazado y podrían haber consecuencias más severas. Hay que leer los manuales de ética de cada organización (como ACM) ya que cada uno tiene sus propias consideraciones en estos aspectos.

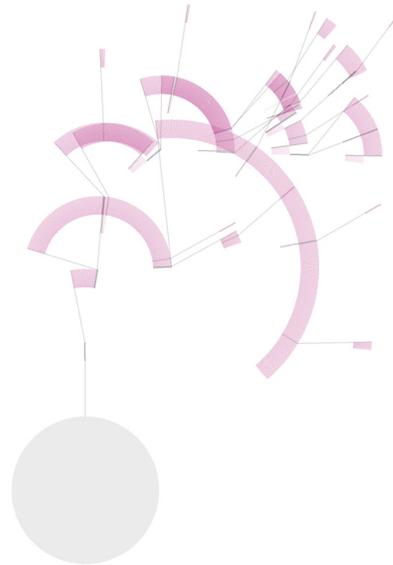


Figura 18. Captura del documento “Resumen Charla” y su visualización.

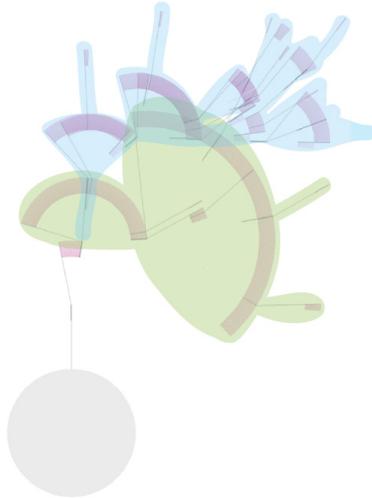


Figura 19. Aunque la primera impresión del documento no permite distinguir bien su estructura (izquierda), al ocultar un nodo podemos distinguir que este corresponde a la primera rama principal, dejando a la vista la segunda (derecha).

Para entender la visualización, se usó la interactividad para ocultar las ramas y así poder distinguir las, como se muestra en la Figura 19. Esta simple operación deja a la vista que el texto tiene dos ramas principales que se desprenden del primer nodo. Por medio de la inspección se observó que el primer nodo contiene el título. No se observan casi eliminaciones (fragmentos grises, desplazados hacia el centro del nodo) y el color revela que todos los cambios ocurrieron en una misma sesión. Probablemente fue escrito durante la misma charla o rápidamente a partir de apuntes tomados durante esta.

Una vez identificadas las ramas principales se pasa a la inspección de éstas para comparar su contenido con el del documento. La Figura 20 (página siguiente) muestra el resultado de esta tarea. En ella se observa que cada una de las ramas identificadas corresponde a un párrafo. Esto permite concluir que el primer párrafo, a pesar de tener una extensión similar al segundo, fue más editado que el segundo ya que tiene una mayor cantidad de nodos, e incluso se pueden observar subramas que podrían ser consecuentemente inspeccionadas para encontrar segmentos subpárrafo. El segundo párrafo, por el contrario fue escrito de manera casi lineal, ya que sólo tiene dos grandes nodos con unas mínimas adiciones posteriores.

## Resumen Charla



La primera parte de la charla trató sobre los diferentes niveles de información segura que nos puede dar un lenguaje estáticamente sobre su comportamiento en runtime. Por ejemplo, cuando se declara una función, cuánto es posible especificar sobre ésta que me revele y me asegure su comportamiento dinámico, es decir, tipos de parámetros, tipo de retorno, excepciones que pueda lanzar, input/output, etc. Lenguajes como Python (no tipificados) tienen un sistema de especificación muy pobre; Java se encuentra un poco más arriba (tipificado); lenguajes como Haskell ya nos permiten asegurar más cosas al definir funciones ya que su sintaxis se empieza a parecer más a las definiciones matemáticas y otros lenguajes llevan la especificación al extremo y son capaces de verificar teoremas (esto significa que no pueden existir comportamientos errados). Aún existen unos lenguajes más extraños que admiten diferentes niveles de especificación para mayor flexibilidad durante el desarrollo.

La segunda parte se concentra en la ética de la investigación en ciencias de la computación. Sobre todo en consideraciones que hay que tener a la hora de enviar papers a conferencias o editoriales. Mucho de esto se resume en que hay tener cuidado cuando se cita verbatim a otro autor o se re-usa material de una publicación propia: en el primer caso, es necesario usar comillas para denotar la cita textual, y en el segundo hay que dejar muy claro que se está haciendo una autocita; o si se está postulando a un congreso con un artículo que es una versión un poco actualizada de otra que fue enviada a otra revisión, también hay que especificar que éste es el caso. De lo contrario el artículo será rechazado y podrían haber consecuencias más severas. Hay que leer los manuales de ética de cada organización (como ACM) ya que cada uno tiene sus propias consideraciones en estos aspectos.

Figura 20. La visualización (izquierda) y el documento (derecha) con las áreas correspondientes resaltadas del mismo color.

### 5.3.2. Caso de estudio 2: “Reflexión sobre elementos no verbales en la obra: ‘Los Invitados’”

Este caso es un documento escrito para un curso de Estudios Humanísticos, donde se analizan elementos de una obra vista por los alumnos. El documento fue escrito por un solo usuario. Su estructura jerárquica consta de título y una sola sección, la cual está dividida en siete párrafos. La Figura 21 muestra el documento y la visualización de su proceso. Se puede observar al visualizarlo una rama que fue totalmente borrada (en un rosado claro) pero de la que salen dos ramas “vivas”. Esto sugirió una observación más cautelosa del crecimiento del árbol.

En la Figura 22 se muestra la visualización del documento en dos momentos diferentes de su evolución, observándose que todo lo que fue escrito durante la primera sesión (de color morado) fue borrado en una sesión posterior, en la que se añadió todo el contenido que finalmente quedó. Se puede observar en la última versión que las ramas principales del documento se desprenden de esta zona borrada. Al inspeccionar la rama eliminada, queda claro que ese texto había sido escrito en forma de “borrador”, sin mucha sintaxis o hilamiento entre frases, para ser luego reescrito en forma legible. Se procedió a inspeccionar el contenido de las ramas vivas.

En la obra se pudo apreciar que los actores no sólo comunicaban con lo que decían, sino que en incluso mayor medida con sus gestos, movimientos, tonos, todo lo que abarca el lenguaje no verbal.

Empezaré por destacar los focos, a lo largo de la presentación los personajes tenían distintos focos: primero los que iban directo a la audiencia, los cuales casi siempre iban acompañados de una interpelación, una invitación, los que hacían que nos sintiéramos parte de esta, como al comienzo, cuando sale el primer personaje, lo primero que hace es mirarnos. Segundo los focos entre ellos, cuando interactúan y se miran mutuamente, hacia uno o más de sus compañeros, por ejemplo cuando Nicolás presenta a su polola "la cosa" el resto de la familia se la queda mirando fijamente, además esta escena va acompañada de expresiones en sus rostros de desagrado lo cual nos comunicó la mala impresión que le dio la apariencia de "la cosa" a la familia de nico. Por último el foco a objetos, en un momento hacen un foco al cielo cuando hablan de las luces navideñas colgadas en el techo y también ponen su vista hacia los regalos en el momento de la entrega.

Pasando a otro elemento hablaré de los gestos, las caras de los actores siempre estaban expresando sus sentimientos, de forma más exagerada que en el diario vivir, cada uno expresando de forma particular la personalidad de su personaje, por ejemplo la joven rubia, su cara siempre mantenía los gestos de tonta, ingenua.

Otros ejemplos de este elemento en la obra fue la escena donde uno de ellos hablaba al público mientras los demás se quedaban congelados en medio de la conversación, en ese momento lo que estaban sentados en la mesa mantenían su expresión, entretanto él que hablaba decía sus verdaderos pensamientos. También el capítulo que hablan del KFC y de la tragedia, todo esto iba acompañado con gestos de dolor. Otro instante donde los gestos comunican mucho fue cuando estaban sentados en la mesa y todos dicen tallas con respecto países y la invitada tira una talla "desubicada" lo cual se logra apreciar porque los demás no se rien y se forma un silencio el cual es acompañado por caras de incomodidad.

Durante la obra los actores se desplazaban alrededor de todo el escenario, desde el piano al frente, a la mesa que estaba al fondo, hacia los regalos al frente a la derecha y así. Las escenas donde más se notó el desplazamiento fueron por ejemplo cuando la sirvienta se pelea con la dueña de casa y se forma el ring de combate, además de gritarse insultos, se tiraban la una a la otra de un lado a otro, limpiaban el piso con la otra, se zamarreaban el pelo, etc. También todas las escenas de baile donde se movían libremente por el escenario.

Todo lo que se ha hablado hasta ahora iba acompañado con los movimientos de brazos y manos, por ejemplo la parte insigne de esto es cuando sentados en la mesa "juegan" a decirse a ellos mismos que hicieron mal o que no los enorgullece con lo que terminaban todos golpeándose el pecho con una mano.

Para finalizar una parte de la obra que para mí recoge todos estos elementos, es el momento en que todos están sentados en el piso y reparten los chalecos navideños, aquí uno de los personajes se pone a llorar porque santa no iba a venir, con todos los gestos de llanto de niño, mientras otros dos se estaban molestando como cabros chicos inquietos y cuando los retaban, se encogían y alejaban con gestos de amurrado, para después volver a molestar. Después todos se paran y empiezan a cantar y dar vueltas alrededor mientras van siguiendo las instrucciones de la "adulta" guía, excepto la invitada que no canta y esta con expresión de extrañeza.

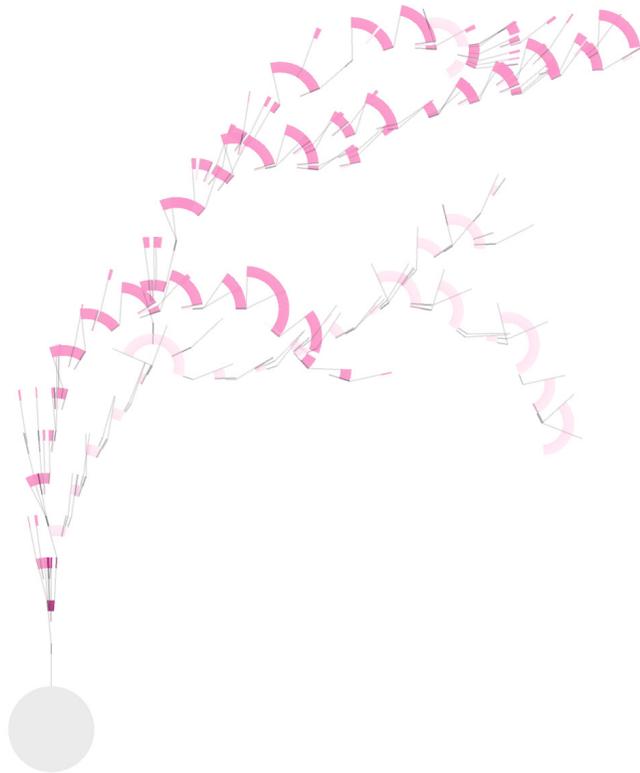


Figura 21. Captura del documento "Reflexión sobre elementos no verbales en la obra: 'Los invitados'" y su THOM final.

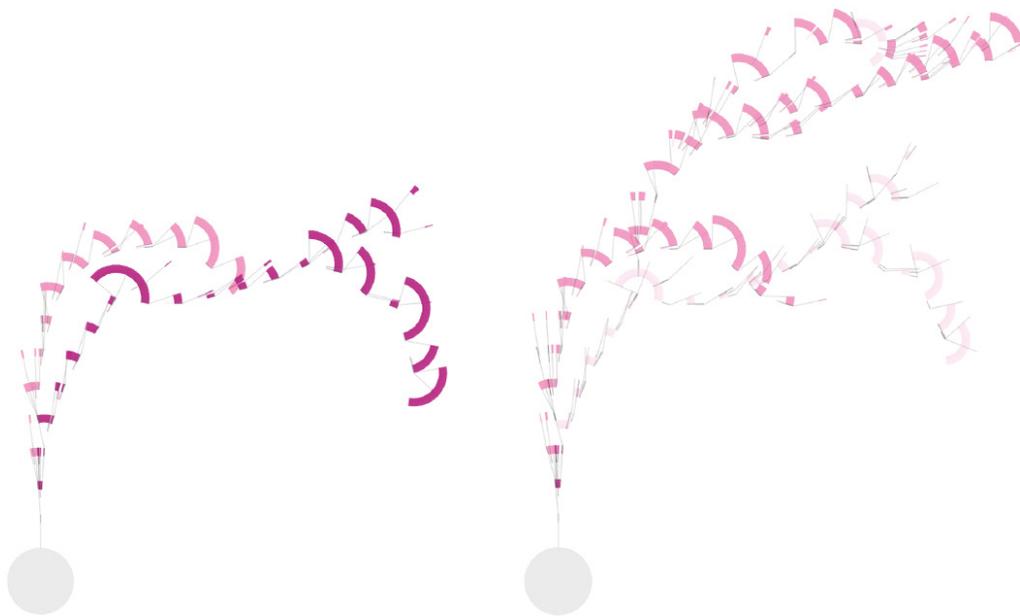
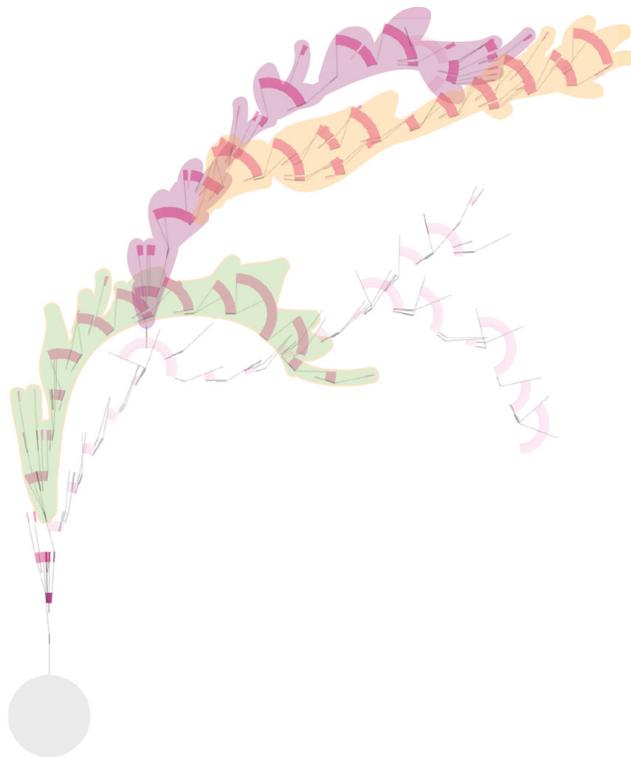


Figura 22. El documento en un momento intermedio de su producción (izquierda) y en su versión final (derecha). Se observa que todo el texto original (de color morado) fue borrado en una sesión posterior.



Reflexión sobre elementos no verbales en la obra: "Los Invitados" Por: [REDACTED]

En la obra se pudo apreciar que los actores no sólo comunicaban con lo que decían, sino que en incluso mayor medida con sus gestos, movimientos, tonos, todo lo que abarca el lenguaje no verbal.

Empezaré por destacar los focos, a lo largo de la presentación los personajes tenían distintos focos: primero los que iban directo a la audiencia, los cuales casi siempre iban acompañados de una interpelación, una invitación, los que hacían que nos sintiéramos parte de esta, como al comienzo, cuando sale el primer personaje, lo primero que hace es mirarnos. Segundo los focos entre ellos, cuando interactúan y se miran mutuamente, hacia uno o más de sus compañeros, por ejemplo cuando Nicolás presenta a su polola "la cosa" el resto de la familia se la queda mirando fijamente, además esta escena va acompañada de expresiones en sus rostros de desagrado lo cual nos comunicó la mala impresión que le dio la apariencia de "la cosa" a la familia de nico. Por último el foco a objetos, en un momento hacen un foco al cielo cuando hablan de las luces navideñas colgadas en el techo y también ponen su vista hacia los regalos en el momento de la entrega.

Pasando a otro elemento hablaré de los gestos, las caras de los actores siempre estaban expresando sus sentimientos, de forma más exagerada que en el diario vivir, cada uno expresando de forma particular la personalidad de su personaje, por ejemplo la joven rubia, su cara siempre mantenía los gestos de tonta, Ingenua.

Otros ejemplos de este elemento en la obra fue la escena donde uno de ellos hablaba al público mientras los demás se quedaban congelados en medio de la conversación, en ese momento lo que estaban sentados en la mesa mantenían su expresión, entretanto él que hablaba decía sus verdaderos pensamientos. También el capítulo que hablan del KFC y de la tragedia, todo esto iba acompañado con gestos de dolor. Otro instante donde los gestos comunican mucho fue cuando estaban sentados en la mesa y todos dicen tallas con respecto países y la invitada tira una talla "desubicada" lo cual se logra apreciar porque los demás no se ríen y se forma un silencio el cual es acompañado por caras de incomodidad.

Durante la obra los actores se desplazaban alrededor de todo el escenario, desde el piano al frente, a la mesa que estaba al fondo, hacia los regalos al frente a la derecha y así. Las escenas donde más se notó el desplazamiento fueron por ejemplo cuando la sirvienta se pelea con la dueña de casa y se forma el ring de combate, además de gritarse insultos, se tiraban la una a la otra de un lado a otro, limpiaban el piso con la otra, se zamarreaban el pelo, etc. También todas las escenas de baile donde se movían libremente por el escenario.

Todo lo que se ha hablado hasta ahora iba acompañado con los movimientos de brazos y manos, por ejemplo la parte insigne de esto es cuando sentados en la mesa "juegan" a decirse a ellos mismos que hicieron mal o que no los enorgullece con lo que terminaban todos golpeándose el pecho con una mano.

Para finalizar una parte de la obra que para mí recoge todos estos elementos, es el momento en que todos están sentados en el piso y reparten los chalecos navideños, aquí uno de los personajes se pone a llorar porque santa no iba a venir, con todos los gestos de llanto de niño, mientras otros dos se estaban molestando como cabros chicos inquietos y cuando los retaban, se encogían y alejaban con gestos de amurrado, para después volver a molestar. Después todos se paran y empiezan a cantar y dar vueltas alrededor mientras van siguiendo las instrucciones de la "adulta" guía, excepto la invitada que no canta y esta con expresión de extrañeza.

Figura 23. El documento y su visualización en su versión final con las áreas correspondientes resaltadas.

Como se observa en la Figura 23, la rama más cercana a la raíz (en verde) corresponde a los primeros dos párrafos, que tratan principalmente sobre la iluminación en la obra, y la segunda rama (en rojo), de la que a su vez sale una tercera rama (en naranja), se concentra en la gestualidad de los actores y luego en el lenguaje kinésico. Aunque no se puede argüir con total seguridad dónde se encuentran las divisiones tópicas del texto, se puede observar que al menos las ramas identificadas de la visualización proveen una aproximación razonable.

### 5.3.3. Caso de estudio 3: "Caso de la Presa Vajont"

Este documento, un informe hecho para un curso de Ética, por un usuario, está estructurado en una portada, seguida de tres secciones y una sección bibliográfica (Figura 24).

En la visualización de este documento (Figura 25, página 40) se distinguen tres ramas principales, las cuales corresponden a las secciones del documento (Figura 26, página 41). Un poco más difícil de distinguir es la pequeña rama que sale del primer "brote" y que corresponde a la Bibliografía. La relación entre estas ramas indica la estructura jerárquica general del documento: la Portada lleva por un lado a la Bibliografía y por otro a las tres secciones, las cuales se encuentran al mismo nivel. El color muestra que en la primera sesión fueron introducidos la portada y la bibliografía, y en la siguiente las tres secciones. Se observan además varios fragmentos borrados a lo largo del texto y ramificaciones pequeñas que dan cuenta del trabajo puesto en su escritura.

## Caso de la Presa Vajont: Análisis Axiológico y Reflexión Personal

Nombre: [REDACTED]

Profesor: [REDACTED]

Fecha de entrega: 23/12 /2015

### b) Análisis del caso a la luz de las variables que propone Echeverría.

Se realizará un análisis de las distintas responsabilidades de los agentes miradas desde 7 valores:

Mirando la tragedia desde los valores básicos se puede notar que hay una grave falta al valor de la vida y el de seguridad, los encargados de llevar el proyecto no tomaron en consideración de manera prioritaria la población que estaba aledaña a la presa y las consecuencias que esta podría tener en caso del derrumbe de tierras del que ya tenían conocimiento y no se previene con una evacuación.

Desde el punto de vista de los valores morales, no hubo honestidad por parte de la compañía y el gobierno, ya que los habitantes de los pueblos nunca fueron informados claramente de lo que ocurría, es más, el gobierno trató de ocultar información entregada por periodistas al resto de la nación. Por otro lado aunque la presa tenía por finalidad generar electricidad, no se hizo el bien, al contrario terminó con muchas vidas y pueblos destruidos. Pasando a la responsabilidad económica, SADE tuvo prisa en entregar el proyecto antes de la nacionalización de las empresas eléctricas porque si no las ganancias previstas serían menores, pero al minimizar los riesgos y al ocurrir el desastre los costos de reparación e indemnizaciones fueron muchos más altos.

De los valores ecológicos, se observa que al construir una represa de esas dimensiones, en una zona de montes frágiles, y además al probar vaciando y llenando esto altera el equilibrio natural y en vez de conservar la biodiversidad, es destruida.

Viendo los valores epistémicos, hay varias faltas, partiendo con los geólogos y los ingenieros a cargo que no fueron precisos en sus estudios y no buscaron la verdad, con un análisis más profundo hubieran encontrado la arquilla, también tendrían que haber puesto en duda la solución de Müller de elevar y disminuir el nivel del agua.

Dentro de los valores técnicos falla la calidad de la simulación que pronostica una ola 10 veces menor a la resultante ya que no contaban con la tecnología computacional de hoy. Por último los valores jurídicos, el juicio de la catástrofe fue realizado en el tribunal de L'Aquila, lejos de la zona que habría correspondido, lo que hizo difícil la asistencia de los supervivientes, se declaró "desastre culpable múltiple premeditado", sin embargo tras sucesivos recursos y apelaciones, las penas se fueron suavizando e incluso se exculpó a algunos de los imputados. Finalmente hubieron tres culpables oficiales y solo uno de ellos cumplió pena en cárcel algo más de una año, lo que es una falta a la justicia, legitimidad y no hubo grandes garantías a las víctimas.

### c) Evaluación del caso (reflexión personal, toma de postura y propuesta de acción al respecto).

Para mí, el caso de la presa de Vajont es muy grave, es un caso donde hubieron muchas negligencias: de parte de la compañía que buscó su bienestar por sobre el de las personas, del equipo de geólogos encabezados por Müller que no realizaron investigaciones precisas de los terrenos, por parte del ingeniero en jefe que debía ser más exhaustivo en el tema de buscar más opiniones, expertos hacer más estudios para garantizar la seguridad de los pobladores y contemplar aspectos como la lluvia en sus cálculos. Uno de los principales errores fue la sobrevaloración de las soluciones realizadas, lo que los hizo creer que tenían el control de la situación cuando no era así, por lo que no alertaron a la población. En un acontecimiento así, evacuar el lugar hubiera salvado muchas vidas. Mi postura frente al caso es de que no fue un desastre natural como fue divulgado al comienzo, sino que fue producto de muchos errores de las compañías y personas a cargo.

### a) Descripción del caso (antecedentes)

La presa de Vajont, estructura de 263m de altura, ubicada en el valle del río Vajont, bajo el monte Toc, a 80 Km al norte de Venecia, Italia, creaba un embalse de 150 Hm<sup>3</sup> que tenía como fin la producción de energía eléctrica. Su construcción inició en 1957 y finalizó en 1959 a cargo de la organización SADE (Sociedad Adriática de Electricidad), teniendo como ingeniero en jefe a Carlos Semenza famoso constructor de presas, siendo este su proyecto final. Junto a sus orillas se encontraban las poblaciones de Erto y Casso que fueron altamente expropiadas y frente a la presa a 1 Km, la ciudad de Longarone.

Teniendo que se repita lo ocurrido en Portesesi, antes del empleo de la presa, la compañía manda a realizar estudios geológicos al monte a manos del profesor Leopold Müller, quien emplea a Eduardo Semenza, hijo del jefe, recién titulado de geólogo, este descubre antiguos derrumbes, por lo que la roca no sería firme y podrían volver a ocurrir.

Carlos manda a hacer un estudio más detallado, se realizan tres agujeros de 170 m de profundidad en la zona de peligro, donde los geólogos buscan signos de desmoronamientos previos, pero no hallan nada. Müller asegura que se trata de roca caliza sólida. 1960 se autoriza a llenar la cuenca. Los granjeros descubren una grieta en forma de m de alrededor de 1700 m de largo la que se agranda 3 cm por día, se observan desprendimientos de tierra y movimientos menores pero el gobierno hace caso omiso y demanda a los periodistas que debían a conocer esto por "socavar el orden público".

El 4 de noviembre de 1960 ocurre un derrumbe de una porción de tierra proporcional a 6 canchas de fútbol lo que provoca una ola de 2 m de altura que no causa mayores problemas, pero alerta a la compañía que por el estudio de Eduardo, temen que el terreno inestable tenga un volumen equivalente a 90 manzanas con edificios de 30 pisos. Müller propone que si al subir el nivel de agua produce un aceleramiento en la caída del terreno, reducirlo la frenaría, por lo que planean controlar la velocidad el desprendimiento, para que este ocurra sin provocar daños. Carlos manda a realizar una simulación física del tsunami en el peor de los casos en el que se desprendiera la ladera. Aquí se encuentra que la mayor ola sería de 20 m por lo que se mantendría el nivel del agua bajo 25 m del muro para que la presa contuviera la ola.

En 1962, ya sin Semenza a la cabeza, continuando las pruebas de llenado y vaciado, se generalizan los terremotos (sismicidad inducida) los cuales son alertados a SADE, pero esta minimizó su importancia y en este mismo año es autorizada a completar el llenado hasta el máximo, 240 m de altura, 10 m más que lo estipulado por Semenza.

El 15 de septiembre de 1963, todo al de la montaña se vino abajo 22 cm. El 26 de septiembre se decide vaciar lentamente la cuenca hasta el nivel seguro propuesto por Semenza. Para el 9 de octubre la ladera se había movido 30 cm por día, pero ya se había alcanzado el nivel seguro por lo que los trabajadores fueron a esperar la creación de la ola y la población no es alertada de la situación, no existen registros de ningún aviso u orden de desplazamiento. A las 10:39 pm de ese día, la combinación del vaciado de la presa y las recientes lluvias torrenales provocaron un derrumbe de 260 millones de metros cúbicos de bosques y tierra que se desplazó hasta 110 Km por hora lo que provocó un tsunami de más de 200 m por sobre la represa, que destruyó los pueblos de Longarone, Pírago, Rivaíta, Villanova, acabando con la vida de unas 2000 personas, además de los pequeños pueblos de Erto y Casso que fueron destruidos en gran parte. Los geólogos no desubrieron que debajo la caliza había una capa de arcilla lo que produjo que el derrumbe alcanzara esa velocidad.

que tuvieron intereses de por medio con el gobierno y buscaron sus propias ganancias, se hizo caso omiso a lo alertado por la gente del lugar y se subestimaron las consecuencias, el proyecto debió ser frenado cuando los primeros estudios geológicos alertaron la posibilidad de derrumbes y si hubieran continuado con el embalse al ver la grieta formada y los continuos desprendimientos y temblores hubieran cesado su funcionamiento y ya en última instancia cuando se tenía el conocimiento del gran desprendimiento, hubieran alertado a la gente y ubicado en una zona segura.

Las acciones posteriores a la tragedia tampoco fueron realizadas de la mejor manera, en un caso así, los tribunales deberían ser más estrictos y otorgar altas penas a los culpables y su imposibilidad de seguir ejerciendo. También brindar un mayor apoyo a los sobrevivientes, que como en los casos presentado en el documental de Russell como en el ensayo de Mejón, eran niños que perdieron a toda su familia, por lo que debieron entregarles ayuda tanto económica como psicológica y dejarlos reubicados en algún hogar.

### Bibliografía

[http://www.aguardesypueblos.org/wp-content/uploads/2010/12/11\\_VAJONT\\_UNA\\_TRAGEDIA\\_ANUNCIADA.pdf](http://www.aguardesypueblos.org/wp-content/uploads/2010/12/11_VAJONT_UNA_TRAGEDIA_ANUNCIADA.pdf) "VAJONT, UNA TRAGEDIA ANUNCIADA"  
Autor: Marisacho Menjón

<http://marcianomx.com/desastre-de-la-presa-vajont/>

SEGUNDOS CATASTROFICOS: LA PRESA VAJONT (SECONDS FROM DISASTER: MOUNTAIN TSUNAMI) Director: Russell Eatough

El principio de responsabilidad: Ensayo de una axiología para la tecnología  
Autor: Javier Echeverría

Figura 24. Captura del documento "Caso de la Presa Vajont: Análisis Axiológico y Reflexión Personal".



Figura 25. Visualización del documento “Caso de la Presa Vajont: Análisis Axiológico y Reflexión Personal” (izquierda) y con ramas coloreadas (derecha).

#### 5.3.4. Caso de estudio 4: “Descripción proceso industrial División El Teniente”

Este caso, un reporte en el que se describe el proceso productivo en una planta de cobre, consta de una introducción seguida de una descripción de las etapas del proceso en forma de lista, más una sección de anexos y bibliografía (Figura 27, página 42). Fue escrito en el lapso de dos días y editado por ambos usuarios (información extraída de Google Drive).

En la Figura 28 (página 43) se muestra la visualización producida por el documento. Se hizo más difícil distinguir todas las ramas en esta visualización, debido a su mayor envergadura y posiblemente a que fue editado por dos usuarios, quedando algunas ramas fuera de la inspección (no coloreadas). Aún así, se alcanzó a extraer una conclusión interesante: la jerarquía de las secciones documento se ve reflejada también en la estructura del árbol (ver Figura 29, página 44): del título (el primer triángulo desde la raíz) se desprenden dos ramas, la que conforma la Introducción (amarilla) y la que va a Anexos (roja). Esto tiene sentido porque Anexos no es parte de ninguna otra sección, más que del documento en sí. La Introducción se separa en cinco ramas prácticamente en un mismo punto: esto sugiere que están al mismo nivel en la jerarquía del documento, y efectivamente, corresponden a las etapas del proceso productivo, presentadas en forma de lista.

La sensación de “planta joven” que da la visualización (ramas alargadas, sin abundancia de sub-ramas) indica que fue el reporte fue escrito de forma expedita, no habiendo mayores ediciones al texto una vez escrito.

Caso de la Presa Vajont:  
Análisis Axiológico y Reflexión Personal

Nombre: [Redacted]  
Profesor: [Redacted]  
Fecha de entrega: 23/12 /2015

b) Análisis del caso a la luz de las variables que propone Echeverría.

Se realizará un análisis de las distintas responsabilidades de los agentes miradas desde 7 valores:

Mirando la tragedia desde los valores básicos se puede notar que hay una grave falta al valor de la vida y el de seguridad, los encargados de llevar el proyecto no tomaron en consideración de manera prioritaria la población que estaba aledaña a la presa y las consecuencias que esta podría tener en caso del derrumbe de tierras del que ya tenían conocimiento y no se previene con una evacuación.

Desde el punto de vista de los valores morales, no hubo honestidad por parte de la compañía y el gobierno, ya que los habitantes de los pueblos nunca fueron informados claramente de lo que ocurría, es más, el gobierno trató de ocultar información entregada por periodistas al resto de la nación. Por otro lado aunque la presa tenía por finalidad generar electricidad, no se hizo el bien, el contrario terminó con muchas vidas y pueblos destruidos. Pasando a la responsabilidad económica, SADE tuvo prisa en entregar el proyecto antes de la nacionalización de las empresas eléctricas porque si no las ganancias previstas serían menores, pero al minimizar los riesgos y al ocurrir el desastre los costos de reparación e indemnizaciones fueron muchos más altos.

De los valores ecológicos, se observa que al construir una represa de esas dimensiones, en una zona de montes frágiles, y además al probar vaciando y llenando esto altera el equilibrio natural y en vez de conservar la biodiversidad, es destruida.

Viendo los valores epistémicos, hay varias faltas, partiendo con los geólogos y los ingenieros a cargo que no fueron precisos en sus estudios y no buscaron la verdad, con un análisis más profundo hubieran encontrado la arcilla, también tendrían que haber puesto en duda la solución de Müller de elevar y disminuir el nivel del agua.

Dentro de los valores técnicos falla la calidad de la simulación que pronostica una ola 10 veces menor a la resultante ya que no contaban con la tecnología computacional de hoy.

Por último los valores jurídicos, el juicio de la catástrofe fue realizado en el tribunal de L'Aquila, lejos de la zona que habría correspondido, lo que hizo difícil la asistencia de los supervivientes, se declaró "desastre culpable múltiple premeditado", sin embargo tras sucesivos recursos y apelaciones, las penas se fueron suavizando e incluso se exculpa a algunos de los imputados, finalmente hubieron tres culpables oficiales y solo uno de ellos cumplió pena en cárcel algo más de un año, lo que es una falta a la justicia, legitimidad y no hubo grandes garantías a las víctimas.

c) Evaluación del caso (reflexión personal, toma de postura y propuesta de acción al respecto).

Para mí, el caso de la presa de Vajont es muy grave, es un caso donde hubieron muchas negligencias: de parte de la compañía que buscó su bienestar por sobre el de las personas, del equipo de geólogos encabezados por Müller que no realizaron investigaciones precisas de los terrenos, por parte del ingeniero en jefe que debía ser más exhaustivo en el tema de buscar más opiniones, expertos hacer más estudios para garantizar la seguridad de los pobladores y contemplar aspectos como la lluvia en sus cálculos. Uno de los principales errores fue la sobrevaloración de las soluciones realizadas, lo que los hizo creer que tenían el control de la situación cuando no era así, por lo que no alertaron a la población. En un acontecimiento así, evacuar el lugar hubiera salvado muchas vidas. Mi postura frente al caso es de que no fue un desastre natural como fue divulgado al comienzo, sino que fue producto de muchos errores de las compañías y personas a cargo,

a) Descripción del caso (antecedentes)

La presa de Vajont, estructura de 263m de altura, ubicada en el valle del río Vajont, bajo el monte Toc, a 80 Km al norte de Venecia, Italia, creaba un embalse de 150 Hm<sup>3</sup> que tenía como fin la producción de energía eléctrica. Su construcción inició en 1957 y finalizó en 1959 a cargo de la organización SADE (Sociedad Adriática de Electricidad), teniendo como ingeniero en jefe a Carlos Semenza famoso constructor de presas, siendo este su proyecto final. Junto a sus orillas se encontraban las poblaciones de Erto y Casso que fueron altamente expropiadas y frente a la presa a 1 Km, la ciudad de Longarone.

Teniendo que se repita lo ocurrido en Pontesei, antes del empleo de la presa la compañía manda a realizar estudios geológicos al monte a manos del profesor Leopold Müller, quien emplea a Eduardo Semenza, hijo del jefe, recién titulado de geólogo, este descubre antiguos derrumbes, por lo que la roca no sería firme y podrían volver a ocurrir.

Carlos manda a hacer un estudio más detallado, se realizan tres agujeros de 170 m de profundidad en la zona de peligro, donde los geólogos buscan signos de desmoronamientos previos, pero no hallan nada. Müller asegura que se trata de roca caliza sólida. 1960 se autoriza a llenar la cuenca. Los granjeros descubren una grieta en forma de m de alrededor de 1700 m de largo la que se agranda 3 cm por día, se observan desprendimientos de tierra y movimientos menores pero el gobierno hace caso omiso y demanda a los periodistas que daban a conocer esto por "socavar el orden público".

El 4 de noviembre de 1960 ocurre un derrumbe de una porción de tierra proporcional a 6 canchas de fútbol lo que provoca una ola de 2 m de altura que no causa mayores problemas, pero alerta a la compañía que por el estudio de Eduardo, temen que el terreno inestable tenga un volumen equivalente a 90 manzanas con edificios de 30 pisos. Müller propone que si al subir el nivel de agua produce un acostamiento en la caída del terreno, reducirlo la frenaría, por lo que planean controlar la velocidad del desprendimiento, para que este ocurra sin provocar daños. Carlos manda a realizar una simulación física del tsunami en el peor de los casos en el que se desprendiera la ladera. Aquí se encuentra que la mayor ola sería de 20 m por lo que se mantendría el nivel del agua bajo 25 m del muro para que la presa contuviera la ola.

En 1962, ya sin Semenza a la cabeza, continuando las pruebas de llenado y vaciado, se generalizan los terremotos (sismicidad inducida) los cuales son alertados a SADE, pero esta minimizó su importancia y en este mismo año es autorizada a completar el llenado hasta el máximo, 240 m de altura, 10 m más que lo estipulado por Semenza.

El 15 de septiembre de 1963, todo el día la montaña se vino abajo 22 cm. El 26 de septiembre se decide vaciar lentamente la cuenca hasta el nivel seguro propuesto por Semenza. Para el 9 de octubre la ladera se había movido 30 cm por día, pero ya se había alcanzado el nivel seguro por lo que los trabajadores fueron a esperar la creación de la ola y la población no es alertada de la situación, no existen registros de ningún aviso u orden de desplazamiento. A las 10:39 pm de ese día, la combinación del vaciado de la presa y las recientes lluvias torrenciales provocaron un derrumbe de 260 millones cúbicos de bosques y tierra que se desplazó hasta 110 Km por hora lo que provocó un tsunami de más de 200 m por sobre la represa, que destruyó los pueblos de Longarone, Pirago, Rivalta, Vilanova, acabando con la vida de unas 2000 personas, además de los pequeños pueblos de Erto y Casso que fueron destruidos en gran parte.

Los geólogos no descubrieron que debajo la caliza había una capa de arcilla lo que produjo que el derrumbe alcanzara esa velocidad.

que tuvieron intereses de por medio con el gobierno y buscaron sus propias ganancias, se hizo caso omiso a lo alertado por la gente del lugar y se subestimaron las consecuencias, el proyecto debió ser frenado cuando los primeros estudios geológicos alertaron la posibilidad de derrumbes y si hubieran continuado con el embalse al ver la grieta formada y los continuos desprendimientos y temblores hubieran cesado su funcionamiento y ya en última instancia cuando se tenía el conocimiento del gran desprendimiento, hubieran alertado a la gente y ubicado en una zona segura.

Las acciones posteriores a la tragedia tampoco fueron realizadas de la mejor manera, en un caso así, los tribunales deberían ser más estrictos y otorgar altas penas a los culpables y su imposibilidad de seguir ejerciendo. También brindar un mayor apoyo a los sobrevivientes, que como en los casos presentado en el documental de Russell como en el ensayo de Mejón, eran niños que perdieron a toda su familia, por lo que debieron entregarles ayuda tanto económica como psicológica y dejarlos reubicados en algún hogar.

Bibliografía

[http://www.aguayosypueblos.org/wp-content/uploads/2010/12/11\\_VAJONT\\_UNA\\_TRAGEDIA\\_ANUNCIADA.pdf](http://www.aguayosypueblos.org/wp-content/uploads/2010/12/11_VAJONT_UNA_TRAGEDIA_ANUNCIADA.pdf)

Autor: Marislancho Menjón

<http://marcianomx.com/desastre-de-la-presa-vajont/>

SEGUNDOS CATASTROFICOS: LA PRESA VAJONT (SECONDS FROM DISASTER: MOUNTAIN TSUNAMI) Director: Russell Eatough

El principio de responsabilidad: Ensayo de una axiología para la tecnociencia

Autor: Javier Echeverría

Figura 26. Captura del documento con colores correspondientes a la Figura 25.

## Descripción proceso industrial

### División el Teniente

Grupo 8:

El proceso de obtención de cobre metálico, puede ser llevado a cabo de dos formas, dependiendo de la composición del mineral extraído. Para el caso de un mineral sulfurado se utilizan procesos pirometalúrgicos, mientras que para el mineral oxidado un proceso hidrometalúrgico. Este último se describe a continuación.

El proceso hidrometalúrgico se divide en seis etapas:

1. Extracción mina, en el Teniente se emplea el método de extracción por hundimiento, que consiste en explotar la superficie del yacimiento subterráneo.
2. Chancado y molienda, el objetivo de esta etapa es efectuar las reducciones de tamaño necesarias de mineral extraído, hasta obtener un producto de una granulometría deseada. Para el caso de lixiviación en pilas, el mineral se suele triturar a tamaños entre 100 y 250 mm. Estos equipos se encuentran ubicados a la distancia mínima entre el yacimiento y la planta, ya que el transporte del material es costoso.
3. Aglomeración, esta etapa tiene como objetivo preparar el material mineralizado para la lixiviación, garantizando un buen coeficiente de permeabilidad de la solución. Este coeficiente es un factor crítico que depende de las características físicas del material en cuanto a proporción de poros, la que depende de la proporción entre lamas (granulometrías inferiores a 5 – 10 micrones) finas (granulometrías inferiores a 100-150 micrones) y gruesas. Una proporción inadecuada puede influir en un aumento innecesario del tiempo de lixiviación y con ello el consumo de reactivos. Para evitar estos inconvenientes es recomendable la eliminación de finos, realizando una aglomeración, que permite la unión de varias partículas finas a otras de mayor tamaño. El aglomerado puede ser realizado con algún adherente, que típicamente es el mismo lixivante ácido y es necesario disponer de una serie de equipos, que pueden formar parte de la propia planta. [Ver anexo A].
4. Lixiviación (LX), es el proceso más importante en que ocurre la extracción sólido-líquido, donde el mineral aglomerado es distribuido en pilas, espacio impermeabilizado [Ver anexo B], en el que se riega el mineral con una solución preparada de agua con ácido. Luego de que la solución a percolado a través de la pila se recolecta el líquido extractor para ser llevado a la etapa de extracción, donde el mineral es recuperado. Para el desarrollo de la lixiviación se deben considerar una serie de factores para asegurar la permeabilidad del mineral, como la proveniencia del mineral, volumen del material y distribución de tamaños, área expuesta, superficie específica, tamaño de partícula, porosidad, presión capilar y rugosidad de la superficie. Además se debe establecer previamente el ángulo de reposo del material en la pila, la pendiente lateral para el drenaje y la inclinación en sentido longitudinal para la evacuación de las soluciones. Cuando la solución lixivante se encuentra en la superficie de la pila comienza la etapa de difusión de los reactivos,

luego la adsorción del material y la reacción química, donde ocurre la disolución de óxidos con ácido sulfúrico luego de la oxidación de sulfuros como se observa en las siguientes reacciones:

\*\*\*

A continuación ocurre la desorción de los productos de la reacción en la superficie del mineral, para finalmente difundir los productos solubles en la solución. De este proceso se espera obtener soluciones de sulfato de cobre. El objetivo será modelar el reactor de lixiviación ácida.

5. Extracción por solvente (SX), corresponde a un proceso efectivo y económico para separar el cobre de la solución lixiviada. Es una operación de transferencia de masa de un sistema de dos fases líquidas, donde una de las fases es orgánica, generalmente petróleo o kerosene inmiscible en agua y la otra fase corresponde a la solución lixiviada. La separación se produce debido a que estos compuestos orgánicos tienen un alto grado de afinidad selectiva con algunos metales como el cobre y casi ninguna afinidad con iones contaminantes como hierro, aluminio, etc. Las fases se mezclan por agitación, en donde el cobre en la solución lixivante se desplaza químicamente a la fase orgánica hasta alcanzar el equilibrio químico. Luego la mezcla resultante es llevada a un sedimentador, generando una capa correspondiente a la fase orgánica (superior, debido a su bajo peso específico) donde se encuentra retenido el ión cobre, y que será llevada a la etapa siguiente de electroobtención y la fase inferior que será la solución lixivante con la menor concentración de cobre. Los parámetros de operación en este proceso son la relación flujo orgánico/ flujo lixivado, tiempo de residencia en el mezclador y flujo total en cada equipo (mezclador y sedimentador).

6. Electro-obtención (EW), este proceso permite recuperar en forma pura y selectiva el cobre en solución. Para ello se hace circular a través de la solución electrolito, una corriente eléctrica continua de baja intensidad entre un ánodo y un cátodo, así los iones cobre son atraídos por el cátodo depositando en él, y las impurezas quedan disueltas en el electrolito.

Para realizar esta operación se requiere de instalaciones llamadas celdas electrolíticas

equipadas con sistemas de circuitos eléctricos para hacer circular una corriente eléctrica continua de baja intensidad.

Las semireacciones que ocurren en el cátodo y ánodo son las siguientes:

### Anexos

#### Anexo A: Procesos de aglomeración

En términos generales, para una buena aglomeración es recomendable que:

- Adicionar la cantidad de líquido o de solución que sea necesaria para alcanzar la humedad óptima de aglomeración.
- La adición del o los aglomerantes se realice sobre el mineral relativamente seco. Por ejemplo, llevaría a cabo en la trituración o chancado si es que está incluida en el esquema de tratamiento.
- La mezcla entre mineral y aglomerantes sea lo más íntima posible.
- Se considere un cierto tiempo de reposo o curado, antes de proceder a la lixiviación. A mayor cantidad de finos, se necesitarán más aglomerantes y más tiempo de curado. Como mínimo, debe existir un período de reposo de unas 24 horas.



Tambores aglomeradores

#### Anexo B: Formación de la pila

El lugar donde se formará la pila requiere una instalación previa de una membrana impermeable, generalmente de plástico, sobre la cual se dispone de un sistema de drenes (tuberías ranuradas) que permitan recoger la solución infiltrada. El material aglomerado es descargado por un equipo esparcidor en el lugar donde se formará la pila, que lo va depositando ordenadamente formando un continuo de 6 a 8 metros de altura (pila de lixiviación). Sobre esta pila se instala un sistema de riego por goteo y aspersores que cubren toda la superficie de contacto. El riego de las pilas se mantiene por 45 a 80 días, hasta que se ha agotado casi completamente la cantidad de cobre lixivado.

### Bibliografía

[1] CODELCO, División El Teniente. [en línea]. <[https://www.codelco.com/nosotros/prontus\\_codelco/2011-08-02/195321.html](https://www.codelco.com/nosotros/prontus_codelco/2011-08-02/195321.html)>. [Consulta: 27 marzo 2016].

[2] EL TENIENTE, MINERÍA DEL FUTURO. [en línea]. <[https://www.codelco.com/prontus\\_codelco/site/artic/20140520/asocfile/20140520183026/el\\_teniente\\_mineria\\_de\\_futuro.pdf](https://www.codelco.com/prontus_codelco/site/artic/20140520/asocfile/20140520183026/el_teniente_mineria_de_futuro.pdf)>. [Consulta: 28 marzo 2016].

Figura 27. Captura del documento “Descripción proceso industrial División el Teniente”.

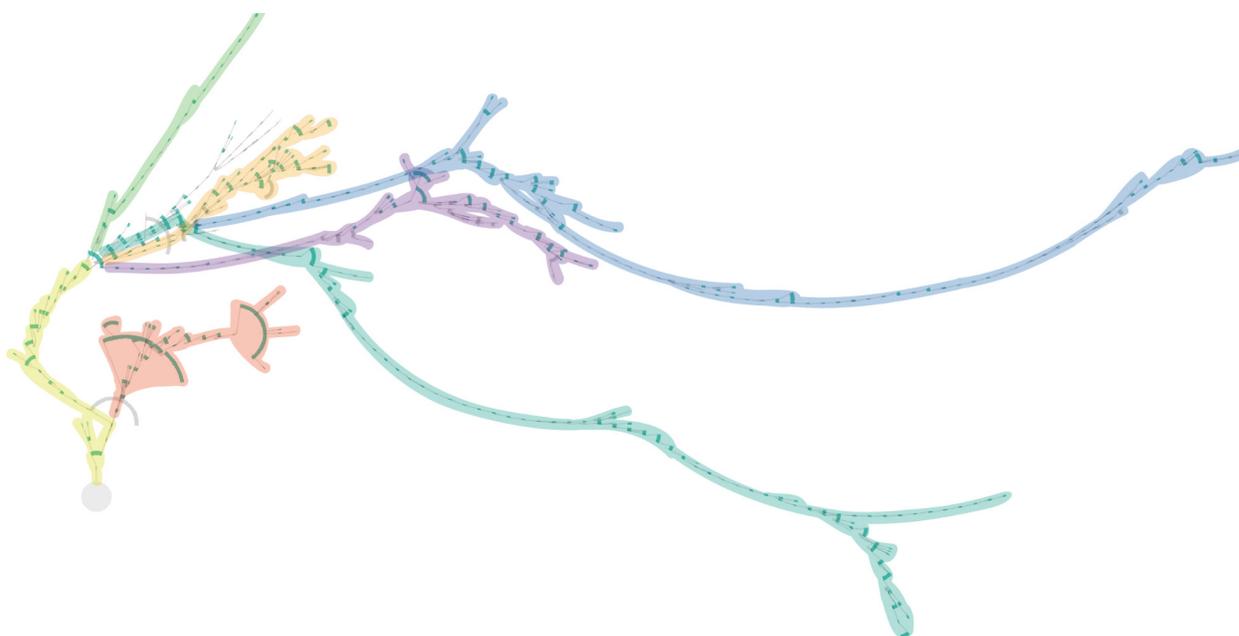


Figura 28. El árbol de “Descripción proceso industrial División el Teniente” (arriba) y el mismo árbol con sus principales ramas coloreadas artificialmente (abajo). Los colores (amarillo, rojo, naranja, morado, verde claro, verde oscuro y azul) indican la correspondencia de los nodos con las partes del texto resaltadas con los mismo colores en la Figura 29.

## Descripción proceso industrial

### División el Teniente

Grupo 8:

El proceso de obtención de cobre metálico, puede ser llevado a cabo de dos formas, dependiendo de la composición del mineral extraído. Para el caso de un mineral sulfurado se utilizan procesos pirometalúrgicos, mientras que para el mineral oxidado un proceso hidrometalúrgico. Este último se describe a continuación.

El proceso hidrometalúrgico se divide en seis etapas:

1. Extracción mina, en el Teniente se emplea el método de extracción por hundimiento, que consiste en explotar la superficie del yacimiento subterráneo.
2. Chancado y molienda, el objetivo de esta etapa es efectuar las reducciones de tamaño necesarias de mineral extraído, hasta obtener un producto de una granulometría deseada. Para el caso de lixiviación en pilas, el mineral se suele triturar a tamaños entre 100 y 250 mm. Estos equipos se encuentran ubicados a la distancia mínima entre el yacimiento y la planta, ya que el transporte del material es costoso.
3. Aglomeración, esta etapa tiene como objetivo preparar el material mineralizado para la lixiviación, garantizando un buen coeficiente de permeabilidad de la solución. Este coeficiente es un factor crítico que depende de las características físicas del material en cuanto a proporción de poros, la que depende de la proporción entre laminas (granulometrías inferiores a 5 – 10 micrones) finas (granulometrías inferiores a 100-150 micrones) y gruesas. Una proporción inadecuada puede influir en un aumento innecesario del tiempo de lixiviación y con ello el consumo de reactivos. Para evitar estos inconvenientes es recomendable la eliminación de finos, realizando una aglomeración, que permite la unión de varias partículas finas a otras de mayor tamaño. El aglomerado puede ser realizado con algún adherente, que típicamente es el mismo lixivante ácido y es necesario disponer de una serie de equipos, que pueden formar parte de la propia planta. [Ver anexo A].
4. Lixiviación (LX), es el proceso más importante en que ocurre la extracción sólido-líquido, donde el mineral aglomerado es distribuido en pilas, espacio impermeabilizado [Ver anexo B], en el que se reaga el mineral con una solución preparada de agua con ácido. Luego de que la solución a percolado a través de la pila se recolecta el líquido extractor para ser llevado a la etapa de extracción, donde el mineral es recuperado. Para el desarrollo de la lixiviación se deben considerar una serie de factores para asegurar la permeabilidad del mineral, como la proveniencia del mineral, volumen del material y distribución de tamaños, área expuesta, superficie específica, tamaño de partícula, porosidad, presión capilar y rugosidad de la superficie. Además se debe establecer previamente el ángulo de reposo del material en la pila, la pendiente lateral para el drenaje y la inclinación en sentido

#### Anexos

##### Anexo A: Procesos de aglomeración

En términos generales, para una buena aglomeración es recomendable que:

- Adicionar la cantidad de líquido o de solución que sea necesaria para alcanzar la humedad óptima de aglomeración.
- La adición del o los aglomerantes se realice sobre el mineral relativamente seco. Por ejemplo, llevaría a cabo en la trituración o chancado si es que está incluida en el esquema de tratamiento.
- La mezcla entre mineral y aglomerantes sea lo más íntima posible.
- Se considere un cierto tiempo de reposo o curado, antes de proceder a la lixiviación. A mayor cantidad de finos, se necesitarán más aglomerantes y más tiempo de curado. Como mínimo, debe existir un período de reposo de unas 24 horas.



Ilustración aglomeración

##### Anexo B: Formación de la pila

El lugar donde se formará la pila requiere una instalación previa de una membrana impermeable, generalmente de plástico, sobre la cual se dispone de un sistema de drenes (tuberías ranuradas) que permitan recoger la solución infiltrada. El material aglomerado es descargado por un equipo esparcidor en el lugar donde se formará la pila, que lo va depositando ordenadamente formando un continuo de 6 a 8 metros de altura (pila de lixiviación). Sobre esta pila se instala un sistema de riego por goteo y aspersores que cubren toda la superficie de contacto. El riego de las pilas se mantiene por 45 a 60 días, hasta que se ha agotado casi completamente la cantidad de cobre lixiviado.

longitudinal para la evacuación de las soluciones. Cuando la solución lixivante se encuentra en la superficie de la pila comienza la etapa de difusión de los reactivos, luego la adsorción del material y la reacción química, donde ocurre la disolución de óxidos con ácido sulfúrico luego de la oxidación de sulfuros como se observa en las siguientes reacciones:

...

A continuación ocurre la desorción de los productos de la reacción en la superficie del mineral, para finalmente difundir los productos solubles en la solución. De este proceso se espera obtener soluciones de sulfato de cobre. El objetivo será modelar el reactor de lixiviación ácida.

5. Extracción por solvente (SX), corresponde a un proceso efectivo y económico para separar el cobre de la solución lixiviada. Es una operación de transferencia de masa de un sistema de dos fases líquidas, donde una de las fases es orgánica, generalmente petróleo o kerosene Inmiscible en agua y la otra fase corresponde a la solución lixiviada. La separación se produce debido a que estos compuestos orgánicos tienen un alto grado de afinidad selectiva con algunos metales como el cobre y casi ninguna afinidad con iones contaminantes como hierro, aluminio, etc. Las fases se mezclan por agitación, en donde el cobre en la solución lixivante se desplaza químicamente a la fase orgánica hasta alcanzar el equilibrio químico. Luego la mezcla resultante es llevada a un sedimentador, generando una capa correspondiente a la fase orgánica (superior, debido a su bajo peso específico) donde se encuentra retenido el ión cobre, y que será llevada a la etapa siguiente de electroobtención y la fase inferior que será la solución lixivante con la menor concentración de cobre. Los parámetros de operación en este proceso son la relación flujo orgánico/ flujo lixiviado, tiempo de residencia en el mezclador y flujo total en cada equipo (mezclador y sedimentador).
6. Electro-obtención (EW), este proceso permite recuperar en forma pura y selectiva el cobre en solución. Para ello se hace circular a través de la solución electrolito, una corriente eléctrica continua de baja intensidad entre un ánodo y un cátodo, así los iones cobre son atraídos por el cátodo depositando en él, y las impurezas quedan disueltas en el electrolito. Para realizar esta operación se requiere de instalaciones llamadas celdas electrolíticas equipadas con sistemas de circuitos eléctricos para hacer circular una corriente eléctrica continua de baja intensidad. Las semirreacciones que ocurren en el cátodo y ánodo son las siguientes:

#### Bibliografía

- [1] CODELCO, División El Teniente, [en línea]. <[https://www.codelco.com/nosotros/prontus\\_codeico/2011-08-02/195321.html](https://www.codelco.com/nosotros/prontus_codeico/2011-08-02/195321.html)>. [Consulta: 27 marzo 2016].
- [2] EL TENIENTE, MINERIA DEL FUTURO, [en línea]. <[https://www.codelco.com/prontus\\_codeico/site/artic/20140520/asocfile/20140520183026/el\\_teniente\\_mineria\\_de\\_futuro.pdf](https://www.codelco.com/prontus_codeico/site/artic/20140520/asocfile/20140520183026/el_teniente_mineria_de_futuro.pdf)>. [Consulta: 28 marzo 2016].

Figura 29. Captura del documento visualizado en 1a. Los textos resaltados están contenidos en los nodos marcados con el mismo color en 1a.

### 5.3.5. Caso de estudio 5: “El quehacer humano”

Este documento corresponde a un ensayo de nueve páginas, sin otra estructura más que las divisiones de párrafos. La Figura 30 (siguiente página) muestra su visualización. Los grandes arcos de color rosado (a) son inequívocamente fragmentos extraídos de un texto anterior, la mayor parte del cual se mantuvo, con añadiduras en varios puntos. Sin embargo, el mayor trabajo en la producción de este documento se puede identificar como dos tareas o “fases” diferentes del proceso: la (b) reescritura de las partes borradas en el arco más largo, ya que se observa que la rama más grande y la segunda más grande tienen su raíz en dos grandes eliminaciones de este arco, y (c) el trabajo de edición que se puede apreciar por todo el documento, en la proliferación de pequeñas ramificaciones de color diferente al de su anterior relativo (i.e., que fueron insertadas en una sesión posterior). Esta última característica produce una sensación de “madurez” del árbol, de que ha visto pasar el tiempo (a diferencia de los otros casos donde las ramas se nuevas, sin intervenciones posteriores), que hace manifiesto inmediatamente el hecho de que el texto ha sido releído varias veces. La estructura de este árbol es básicamente de una sola gran rama, lo que se condice con la falta de secciones en el documento original. Se puede decir que el patrón de producción que presenta este caso es el de ampliación y reescritura de un texto anterior.

### 5.3.6. Otros casos

Otros documentos resultaron en visualizaciones muy difíciles de leer, por lo que fueron eximidos de un mayor análisis. Esta ilegibilidad se debe en la mayor parte de los casos a alguno de los siguientes factores:

- una extensión superior a las 5000 palabras;
- muchas ediciones concentradas en reducidos espacios (en general debido a edición de tablas);
- dos usuarios escribiendo al mismo tiempo, lo cual produce la aparición de un artefacto en la visualización en el cual las ramas se generan extremadamente finas y alargadas, ya que el crecimiento de un nodo es interrumpido por la aparición de otro nodo en otra parte del documento y así sucesivamente.

Esto no quiere decir que su análisis sea imposible, sólo que requiere de más tiempo y que sería más propicio si se hicieran algunas mejoras a la visualización. En la Figura 31 (página 47) se ven ejemplos de estos documentos visualizados.

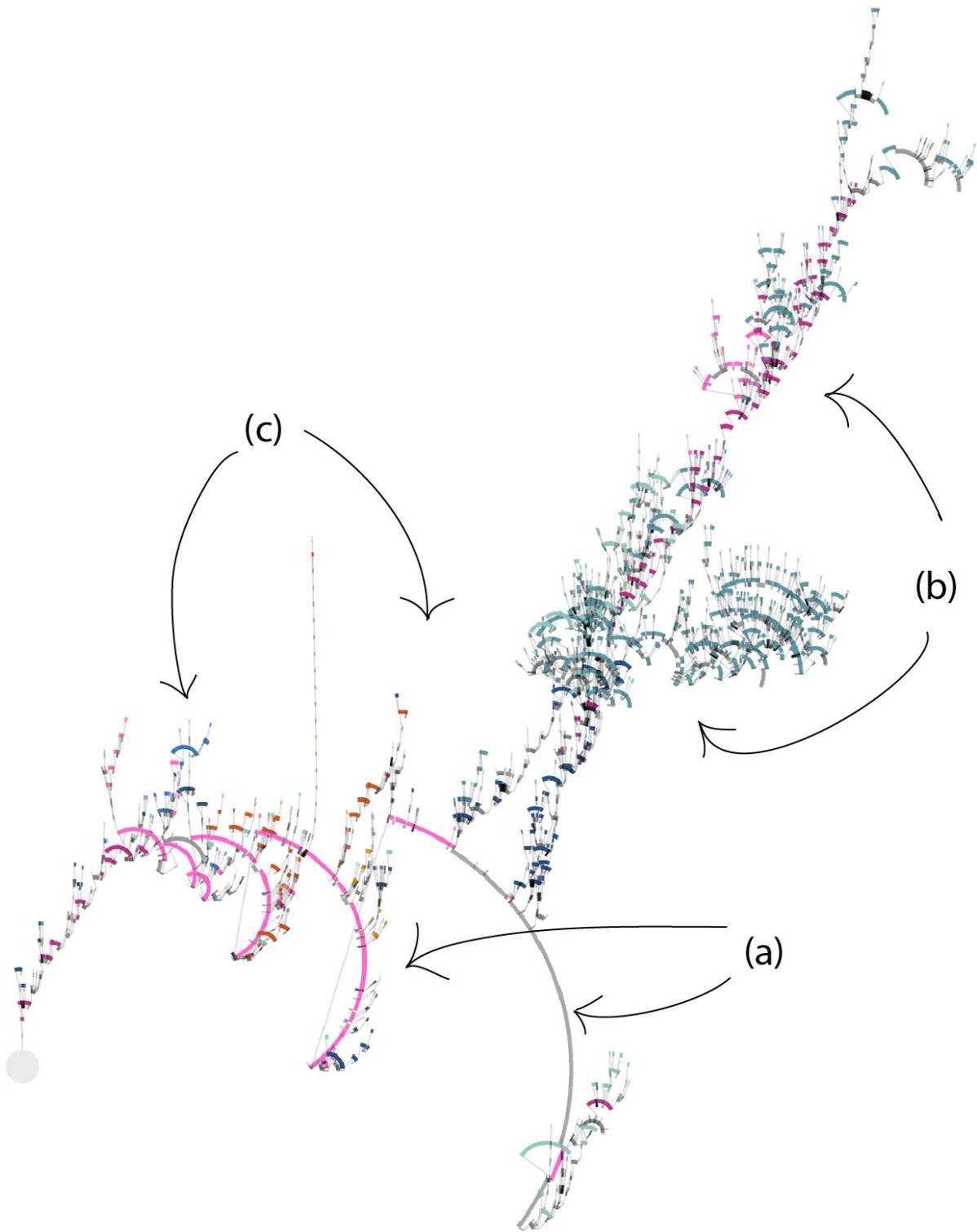


Figura 30. Visualización del documento "El quehacer humano". Se distinguen los siguientes características: (a) nodos copiados de otro documento; (b) principales ramas; (c) ediciones posteriores.

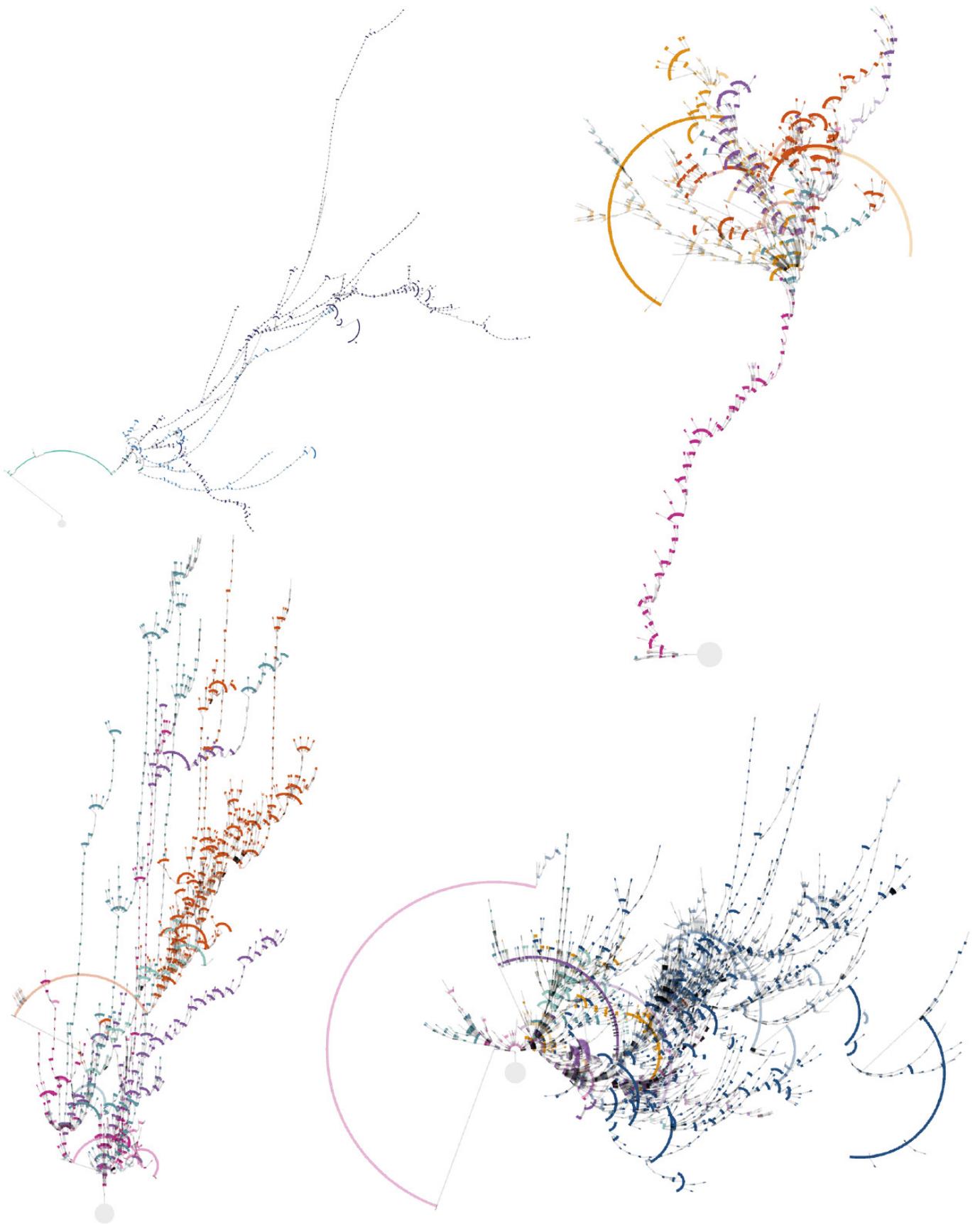


Figura 31. Visualizaciones de casos considerados ilegibles.



# 6 Modelo para la Visualización de Información

Evaluar una visualización es, en cualquier caso, un tema complicado. En parte porque siempre se tiene la esperanza de que esta muestre algo en los datos que no estaba previsto en su diseño, que exhiba una propiedad emergente, y en general, el éxito de las visualizaciones desarrolladas con fines de investigación se mide en su sacar a relucir patrones en los datos que no habían sido calculados en un comienzo pero que “emergen” a través de una configuración significativa de los elementos visuales (caso de las *edit wars* en History Flow). Sin un estudio de usabilidad, incluyendo sistemáticas pruebas de usuario, es imposible tener una muestra empírica de la efectividad de una visualización y aún así, el valor de esta clase de validación es seguramente debatible. Son raras las visualizaciones que llegan a entrar en uso común, entre las cuales podríamos destacar el treemap (Johnson & Shneiderman, 1991).

En esta sección se busca proponer un marco teórico para la evaluación de la visualización desarrollada. Para ello primero se desarrolló un modelo basado en la literatura actual que intenta formalizar la idea de visualización de información en tres pasos: definirla y diferenciarla de otras formas de visualización, describir las reglas que rigen la relación entre el espacio de los datos y el espacio visual, y llegar a una expresión de efectividad por medio de su relación al usuario. Con este análisis se produjo una crítica al diseño y se generó una lista de tareas básicas que son evaluadas en el prototipo alcanzado.

Generar una pauta de evaluación *in vitro* para una visualización no tiene mucho sentido si no es para ayudar y retroalimentar las iteraciones en el diseño de la misma. Por ello, se condensó el modelo antes descrito como un manual de apoyo al diseño desde la base (qué es una visualización de información), pasando por los principios de la representación (cómo (no) deben mostrarse los datos) para finalmente llegar al objetivo: el usuario. La evaluación entonces, y por la falta de un estudio con usuarios reales, debe considerar las tres capas de este modelo genérico y satisfacer cada una en su manera particular. La evaluación propiamente tal se lleva a cabo en la sección Evaluación de la solución.

## 6.1. CAPAS

Aquí se formula un modelo para entender la visualización de información, qué es y cómo funciona. El modelo tiene tres capas: primero, una definición formal de lo que es la visualización, luego una capa “lógica” de cómo el espacio de los datos interactúa con el visual, y por último una interfaz en la que se conecta al usuario y sus motivaciones.

La literatura sobre evaluación de visualizaciones es dispersa, no existiendo ningún estándar ni teoría unificada, pero se reconoce la creciente necesidad de desarrollar esta área para el futuro de la visualización. Las métricas encontradas son variadas y en general su utilidad depende del propósito, los datos o la representación que se utilice, por lo que en muchos casos fueron reinterpretadas y reestructuradas para obtener una pauta consistente y lo más integrada posible. Donde ocurrieron estas intervenciones siempre se cita al trabajo original.

### **6.1.1. Capa 1: La visualización como una función**

Según Ziemkiewicz & Kosara (2009) podemos definir la visualización de información y distinguirla de áreas similares (visualización científica, infographics, visualización con intención artística o comercial, etc.), a la vez proveer los medios para su clasificación y generar un nuevo entendimiento del campo, como una función con las siguientes características:

1. *Data-driven*: Que su dominio sean el dominio de los datos y su codominio el de los elementos visuales.
2. Biyectividad: Que sea un mapeo biyectivo.
3. Interactividad: Que su dominio no esté restringido.
4. Notacionalidad: Que defina en su codominio un sistema notacional sintáctico.

En otros términos, para que algo califique estrictamente como visualización de información (1) debe originarse de un conjunto de datos, (2) debe ser posible recuperar este conjunto de datos desde su visualización además de que no hayan “artefactos” que no provengan de los datos, (3) debe permitir la interacción no-trivial del usuario, es decir, que pueda activamente buscar información usando un conjunto arbitrario de datos como input y (4) debe mapear los datos a símbolos que formen clases de equivalencia (i.e., que puedan ser intercambiados sin efecto sintáctico) y que sean finitamente diferenciables.

El último punto quiere decir que la representación debe conferir significado a través de una especie de gramática, un lenguaje visual articulado. Esto deja fuera de la visualización de información, por ejemplo, a las visualizaciones de fractales, a las visualizaciones de funciones en general y en parte a la visualización científica.

Por supuesto, que una visualización cumpla estas características no nos habla de su efectividad, sólo que estamos frente a una visualización de información, y aún así, muchas visualizaciones no conforman al cien por ciento con estas restricciones. Además, este modelo se abstrae de todo tipo de datos y de la materialidad del proceso. Veremos que diferentes relajaciones de las ideas de biyectividad y notacionalidad nos pueden proveer de herramientas para la evaluación cualitativa de una visualización de información.

## 6.1.2. Capa 2: El espacio de la representación

En Kindlman & Scheidegger (2014) se introduce un mapeo intermedio entre los datos y la visualización. La Representación es, en este caso, todas las posibles formas y estructuras intermedias que pueden tomar los datos antes de su visualización. Luego se definen tres principios sobre la interacción entre el espacio de Datos, Representación y Visualización que una visualización puede cumplir (idealmente) o, en caso contrario, fallar. Estos principios son: el principio de Invarianza (una visualización debe ser invariante respecto a cambios en la Representación), el principio de No-ambigüedad (no pueden haber dos conjuntos de datos que lleven a la misma visualización) y el principio de Correspondencia (cambios significativos en los datos deben corresponder con cambios significativos en la visualización). A continuación damos ejemplo de cada uno de estos principios en acción.

1. El principio de Invarianza estipula que la Representación de los datos no puede tener incidencia en el resultado final. Un ejemplo de esto son los gráficos con información categórica (barras, barras compuestas, pie charts, matriz de adyacencia, entre otros). La información categórica por lo general se representa como una tabla, en que el orden de los valores es arbitrario (el orden alfabético es tan arbitrario como el alfabético invertido o el random). En todos los gráficos mencionados, un cierto orden queda establecido entre las categorías al ser visualizadas: orden, sin embargo, que viene del espacio de la Representación y no es intrínseco a los datos. Una visualización que falla el Principio de Invarianza, en este caso, puede mostrarnos patrones que tienen su origen en la representación de los datos, y no en los datos mismos. Se dice entonces que la visualización posee un *alucinador* (*hallucinator*).
2. El principio de No-ambigüedad refiere a casos degenerados en que dos datasets distintos no pueden ser diferenciados en su visualización. En el treemap clásico, por ejemplo, cuando todas las hojas poseen el mismo peso, se pierde la estructura jerárquica al visualizarlos. Se dice de visualizaciones que no cumplen esta propiedad que poseen *confusores* (*confusers*). En un sentido más amplio, es muy similar a la idea de *faithfulness* en grafos (Nguyen et al., 2013), y en cualquier caso está muy ligado a la idea de biyectividad definida anteriormente.
3. El principio de Correspondencia, el más irregular de los tres, especifica que cambios significativos en los datos o en la visualización se deben corresponder con cambios significativos en su contraparte. Es típico, por ejemplo, que en un grafo dibujado mediante un *force-directed layout*, la posición de los nodos cambia radicalmente cada vez que uno nuevo es añadido, distorsionando la percepción del espacio y destruyendo el mapa mental del sujeto, o al comparar dos sets de datos que sólo difieren en unos pocos nodos (en Lee et al., 2006, se pone algo de trabajo en esto). Esto viola el principio de Correspondencia, donde un cambio pequeño en los datos no puede equivaler a un cambio de la visualización que se preste para otras interpretaciones. En este caso, se dice que la visualización es *engañososa* (*misleading*). Otros ejemplos de visualizaciones engañosas son clusterización por PCA (un movimiento importante en la visualización no necesariamente se corresponde con uno en los datos, ya que pueden ocurrir rotaciones en espa-

cios de mayor dimensión) y en general cualquier visualización donde una escala de color mal elegida nos lleva a pensar que existe una simetría interna a los datos (e.g., permutabilidad, positivo-negativo) que sólo es un efecto de nuestra percepción del color. El caso simétricamente opuesto a un *engaño* es la *mezcolanza (jumble)*, que es cuando una simetría propia de los datos no se ve reflejada por los elementos visuales presentes. Nuevamente, una escala de color que no tenga simetría puede producir efectos extraños en la visualización frente a una inversión de los datos (Borland y Taylor (2007) presentan argumentos similares sobre las desventajas de la escala arcoiris). Un ejemplo de esto aplicado a la geometría de la representación supercuádrlica de tensores se puede encontrar en Kindlman (2004).

Estos tres principios ofrecen buenas fórmulas para saber dónde mirar para encontrar posibles errores en una visualización. Podemos ver, por ejemplo, que una visualización que presente problemas de oclusión podría fallar el principio de No-ambigüedad. Sin embargo, el cumplirlos tampoco asegura que una visualización sea efectiva y, de hecho, muchas visualizaciones hacen uso de abstracciones que pueden contrariarse con algunos de los principios y aún son efectivas, pero pueden degenerar en casos en que dejan de serlo, dependiendo los datos.

### 6.1.3. Capa 3: Percepción, legibilidad y tareas

En Nguyen et al. (2013) consideran que la efectividad de una visualización no es algo inmanente a sus condiciones, sino que dependen en su interacción con un usuario que a su vez está sometido a un contexto. Por esto consideran en su modelo de la visualización una función de percepción, la cual es modulada en función de una tarea. Esta función de percepción humana es la que busca ser optimizada. Legibilidad (*readability*) sería la propiedad de una visualización que la hace ser entendible por el usuario, o en otras palabras, cuánta información o conocimiento es capaz de transferir a su percepción. Es decir, es una medida directa de su eficacia. Se hace la diferencia entre información-legibilidad, tarea-legibilidad y cambio-legibilidad, para explicitar que de acuerdo a una cierta tarea la visualización puede exhibir mayor o menor legibilidad que para otra, y que la visualización mantiene su legibilidad entre cambios (de forma similar a la idea de mapa mental (Lee et al., 2006)). Es decir, tarea-legibilidad es la legibilidad respecto a una tarea específica, y cambio-legibilidad atañe a su legibilidad en el tiempo, incluyendo el aspecto dinámico de una visualización. Aunque la legibilidad se suele asociar a una serie de factores mayoritariamente estéticos (e.g., en el caso de grafos: la cantidad de cruces entre aristas, la cantidad de curvas, etc.) hemos visto ya la cantidad de errores que pueden afectar la legibilidad de una visualización bajo ciertas condiciones.

Una breve pero útil taxonomía de tareas que puede ofrecer una visualización se encuentra en Shneiderman (1996). Estas son:

- *Overview*: obtener una vista de todos los elementos.
- *Zoom*: hacer zoom sobre elementos de interés.
- *Filter*: filtrar elementos sin interés.

- *Details-on-demand*: seleccionar un ítem o grupo y obtener detalles.
- *Relate*: ver relaciones entre elementos.
- *History*: mantener una historia de acciones ejercidas sobre los datos para soportar *undo*, *redo* y un refinamiento progresivo de este.
- *Extract*: exportación a otros formatos del dataset refinado.

Esta lista da una idea de lo que una visualización correcta debe permitir de forma genérica y amplia. Brehmer & Munzner (2013) propone una tipología más refinada de tareas de visualización abstractas. En esta, cada tarea ocurre en tres niveles diferentes a la vez: *qué* (qué datos toma como *input* y devuelve como *output*), *cómo* (qué métodos utiliza para producir el resultado) y *por qué* (cuál es el propósito de la operación). Así, es posible generar tareas complejas (alto-nivel) a partir de tareas atómicas (bajo-nivel) a la vez que incluir las motivaciones del usuario en el *por qué* de la tarea. No es necesario el describirla por completo (para más detalle, ver Figura 32), siendo su utilidad entregar un espacio sintáctico en que describir tareas complejas (que requieren varias operaciones), y para efectos de evaluación proveer una lista más actualizada de tareas atómicas (*how*) para descubrir posibilidades de interacción que podrían estar presentes en el prototipo o en el futuro de este. Por último, introduce en este modelo el aspecto interactivo de la visualización de información a la vez que lo liga al usuario y sus motivos (*present, discover, enjoy*), cerrando conceptualmente el “ambiente” en el que se mueve una visualización.

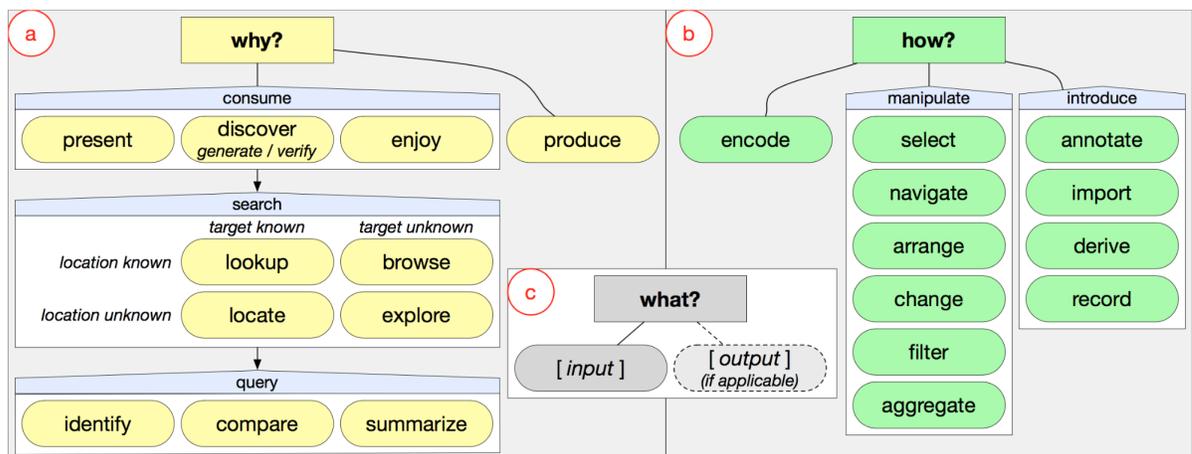


Figura 32. Tipología multi-nivel de tareas de visualización abstractas (Brehmer & Munzner, 2013).



## 7 Evaluación

En este capítulo se lleva a cabo una evaluación de la solución de acuerdo al modelo planteado en el Capítulo 6. La evaluación arrojó que el prototipo corresponde efectivamente a una visualización de información, tiene algunas problemáticas en el diseño y que una serie de funcionalidades que deberían ser implementadas para proveer una más completa experiencia de al usuario.

### 7.1. EVALUACIÓN POR CAPAS DEL PROTOTIPO

En esta parte se va a evaluar cómo el prototipo desarrollado satisface las capas del modelo propuesto en el Capítulo 6.

#### 7.1.1. Capa 1

Aunque parezca trivial, vale la pena asegurarse de que lo que estamos evaluando es precisamente una visualización de información y no otro tipo de visualización. Para esto hay que revisar las cuatro condiciones que plantean en Ziemkiewicz & Kosara (2009):

1. El prototipo es *data-driven*, porque toma como *input* un conjunto de datos.
2. El prototipo es biyectivo, ya que mapea cada atributo de los datos a una variable visual. Se está considerando que el texto mismo, aunque no está presente en la vista general de la visualización, puede ser mostrado *on-demand* usando la interactividad. Hay dos excepciones a la biyectividad, referentes a la información temporal de las operaciones y a la compresión de los datos, que se desarrollan en la siguiente capa.
3. El prototipo es interactivo, dado que puede tomar cualquier valor perteneciente al dominio de los documentos de Google Docs como *input*.
4. El prototipo es notacional, porque todos los elementos significantes de los que hace uso son discretos y forman una clase de equivalencia (Glifo).

Así se concluye de que el prototipo corresponde en sus rasgos a una visualización de información, aunque con una biyectividad parcial pero teóricamente alcanzable en su totalidad, sobre lo cual se expande a continuación.

#### 7.1.2. Capa 2

Utilizando los principios de Kindlman & Scheidegger (2014) podemos echar una luz desde la teoría a la calidad de una visualización, en la forma en que lo visual interactúa con los datos y la incidencia del espacio que la Representación tiene en esta interacción. Claramente, la Representación en el caso de este prototipo se refiere a la estructura de datos sobre la que la visualización está

construida, mientras que los Datos corresponden al archivo JSON que entrega Drive. La evaluación de la visualización a través de los tres Principios nos permitió encontrar tres problemáticas en ella, algunas de las cuales tienen una solución clara y otras no tanto: (1) la compresión de los datos a través del *burstlog*, (2) la organización visual producida por la eliminación y (3) la pérdida de información temporal en la estructura.

A continuación se detallan estas problemáticas, seguidas de las posibles soluciones:

### ***Problemáticas***

1. Como se especifica en el capítulo Estructura de Datos, antes de agregar un cambio a la estructura, las operaciones se van agrupando en bloques llamados *bursts*, con el fin de no sobrecargar el espacio visual. Esta suerte de compresión de los datos fuerza una pérdida de resolución de la información, la cual es una violación en diferentes niveles de los tres principios, pero sobre todo el de Invarianza, ya que la Representación está afectando el resultado visual.
2. La eliminación es adecuadamente capturada en la estructura de datos a través del DAG, pero no así en la visualización que mantiene la estructura de árbol. Este tratamiento de la eliminación no significa una pérdida de información (es decir, el árbol sigue siendo equivalente en un sentido visual-estructural con el DAG), pero obliga a tomar la decisión arbitraria de si una nueva inserción que ocurre en el lugar de operación de una eliminación aparece a la izquierda, a la derecha o en alguna otra posición, frente a lo cual se decidió posicionar el glifo a la izquierda porque parecía lo más intuitivo y daba buenos resultados. Esta decisión cambia efectivamente la forma del árbol, por lo que estaría violando el principio de Correspondencia.
3. El prototipo sería ambiguo o contendría confusores si dos datasets distintos dieran el mismo resultado al visualizarlos. La visualización, en su forma estática, puede de hecho prestarse a confusores, en lo que se refiere al orden de ocurrencia de nodos no relativos. Dos documentos distintos podrían haber sido escritos con operaciones similares pero en distinto orden, y mientras esto ocurra en ramas paralelas y dentro de la misma sesión serían indiscernibles uno del otro. De hecho casi cualquier árbol tiene un cierto grado de ambigüedad debido a esta cuota de información temporal perdida.

### ***Soluciones***

1. Una posible solución sería hacer de la agrupación un parámetro que pueda ser controlado dinámicamente, es decir, darle a la estructura de datos la capacidad de comprimirse y descomprimirse, para obtener árboles de resolución variable (otra hubiera sido nunca agrupar los datos en *bursts*, alternativa con la que no se experimentó).
2. Esta problemática es la que tiene una solución menos clara, ya que al parecer, todo el espacio de posibilidades de cómo posicionar los nodos sobre las inserciones responde sólo a una decisión arbitraria, es decir, la estructura de la visualización actual no soporta la eliminación de una manera definida, tal como ocurre en la estructura de datos. Esto es sin duda por la discrepancia

que existe entre la estructura de datos y la visualización (DAG vs. árbol) y apunta a una falta fundamental de la última. Para superarla, habría seguramente que cambiar el glifo e incluso el diagrama recursivo de la visualización, lo que es lo mismo que decir que habría que diseñar otra desde cero, lo que sin duda queda como tema para investigaciones futuras. Esto nos revela una profunda crítica al diseño propuesto: este en realidad, y al igual que la primera estructura de datos, exclusivamente jerárquica, funciona sobre un modelo en el cual sólo las inserciones cumplen un rol estructural, es decir, la eliminación no fue plenamente considerada como una manera de dar forma al proceso de escritura. Si es que llevar esto a la visualización es siquiera posible es algo que no se puede saber hasta que ocurra.

3. El usar diferentes colores en distintas sesiones remedia esto al punto de hacerlo tolerable, pero una solución total al problema de la ambigüedad necesitaría de una escala de color continua. De todas maneras, al ver el árbol en su fase de crecimiento las ambigüedades respecto al orden temporal de las operaciones desaparecen. Sería interesante también poder controlar el tiempo con un *slider*. Por otro lado, la gama de colores funciona de manera continua en un patrón circular, lo que introduce una simetría que no está presente en los datos, en que el tiempo es lineal, por lo que continuar usándola produciría un *jumbler*, es decir, una violación del principio de Correspondencia. En otras palabras, si un texto fue escrito en más de ocho sesiones, los colores se repiten haciendo que se mezclen datos de dos tiempos diferentes dentro de la misma ventana de tiempo.

Estos, en la opinión del autor, son los puntos más importantes que hay para considerar en lo que respecta al futuro desarrollo de la visualización. Poder visualizar de una manera significativa la eliminación en la historia del documento es una parte central del este trabajo, sin embargo, aunque fue bien definida en la estructura de datos, no pudo ser tan satisfactoriamente integrada en la visualización. La siguiente capa investiga los aspectos interactivos del prototipo y su relación con el usuario.

### **7.1.3. Capa 3**

Siguiendo la taxonomía de Shneiderman (1996), se describen a continuación las características de la visualización que satisfacen las tareas de la taxonomía o en su defecto cómo podrían ser satisfechas. La evaluación arroja que de las siete tareas, cuatro son cumplidas a cabalidad y para dos (Historial y Extraer) no se ofrece en el prototipo actual ninguna funcionalidad.

#### ***Vista general***

La visualización es en sí una panorámica de los datos por lo que esta tarea es trivialmente satisfecha.

#### ***Zoom***

La función de acercar o alejar los elementos está cubierta por la visión de cámara interactiva.

### ***Filtrar***

El filtrado de elementos es provisto por la función Ocultar/Mostrar de cada nodo.

### ***Detalles-en-demanda***

Se pueden obtener los detalles (el texto contenido) manualmente haciendo click en un nodo. Sería útil expandir esta función para que entregue más información de los nodos, como el *timestamp*, o para que se pudiera seleccionar un nodo individual y obtener sus detalles sin tener que primero filtrar sus hijos.

### ***Relacionar***

Esta tarea es más la ambigua ya que depende mucho del tipo de datos con los que se está trabajando. Funcionalidades para relacionar en este caso podrían ser: seleccionar un nodo y que se resalten todos sus hijos; seleccionar un nodo y que se resalten todos los nodos que ocurrieron después de él; seleccionar un nodo y que se resalten todas las partes de la estructura que comparten contenido (útil para encontrar desplazamientos de texto). Ninguna de estas funciones es provista por el prototipo actualmente, pero no hay nada que impida su implementación futura. Una forma simple y efectiva de añadir estas interacciones al prototipo actual sería usando un mapeado cada una a una combinación de [tecla]+click. Otro tipo diferente de relacionar sería ingresar un *string* y que se resalten todas las partes de la estructura donde se encuentra, como la función “buscar” de cualquier programa.

### ***Historial***

No se proveen funciones de historial. Esto sería útil sin duda para ofrecer un *undo/redo* de las acciones de ocultamiento, lo cual facilitaría el análisis. Se podría implementar manteniendo una pila de los nodos que son accionados, y que al presionar ctrl+z (cmdz) se volviera a accionar el nodo del tope (lo cual deshalaría la acción anterior, tratándose de un *switch* binario) y luego se extrajera de la pila.

### ***Exportar***

La visualización permite exportar a PNG la vista actual, pero no permite exportar la estructura de datos del documento actual, ya que tampoco se ha hecho un programa que lea esta estructura de datos directamente.

Esta taxonomía es bastante limitada, por eso expandimos el modelo con la tipología de Brehmer & Munzner (2013) que nos permite agregar nuevas tareas, las cuales pueden ser articuladas para producir tareas más complejas. A continuación expandimos el conjunto de tareas anterior con conceptos extraídos de esta nueva tipología.

### ***Seleccionar***

Similar a Relacionar.

### ***Navegar***

La funcionalidad de navegación está dada por el movimiento que se puede hacer de la vista arrastrándola con el mouse.

### ***Organizar***

Los elementos (glifos) pueden ser reorganizados modificando los factores de apertura y fototropismo con las teclas de movimiento. Algo muy útil podría ser una combinación de Seleccionar y Organizar que permitiera arrastrar un nodo para ubicarlo en una posición más favorable.

### ***Cambiar***

Se refiere a cambiar ciertas codificaciones visuales, por ejemplo, el color. En este momento no hay “variedades” de la visualización, pero podría ofrecerse en un futuro otro estilo de glifo u otras escalas de color.

### ***Agregar***

Agregar los datos es una funcionalidad importante en cualquier visualización, y en este caso significaría implementar lo que se mencionaba en la Capa 2 de la estructura compresible/descompresible. Esto podría ser un parámetro global, pero también podría usarse una combinación de Seleccionar y Agregar para agregar localmente zonas sin interés para dejar más espacio a las que se quieran estudiar.

### ***Anotar***

Hacer anotaciones (dejar comentarios gráficos o textuales sobre los objetos) es algo que sin duda es una comodidad para la investigación y la presentación de resultados, pero no es algo que verdaderamente agregue algo a la visualización, además de ser algo costoso de implementar, pero algo como una forma de *tagging* (Selección y Anotación) para poder colorear partes del árbol a voluntad, como se hizo en Experimentación con un programa externo, sería de buena utilidad.

### ***Importar***

Importar (introducir nuevos datos a la visualización) podría significar en este caso importar documentos en su forma documento para poder relacionar las partes del texto con los nodos correspondiente dentro de la misma visualización.

### ***Derivar***

Derivar consiste en producir nuevos datos *persistentes* (lo que lo diferencia de Agregar) a partir de operar sobre los ya presentes. No está claro qué funcionalidad podría tener esto con los datos que usamos en este trabajo.

### ***Filtrar***

Lo mismo que en Derivar.

## ***Codificar***

Según los autores de esta tipología, la mayoría de las tareas de visualización dependen en cómo la data es inicialmente codificada en una representación visual. Esto se contradice con lo que establece el Principio de Invarianza, que dicta que una visualización debe ser invariante frente a la representación de los datos. Se ha adherido en una capa previa a esta visión de la visualización, por lo que esta categoría simplemente ya no tiene sentido.

Siguiendo en esta línea, podríamos tratar de generar las tarea complejas que se realizaron según la metodología llevada a cabo en Experimentación, sin embargo, basta darse cuenta de que parte de ella involucra un recurso externo (el documento mismo en Docs) para ver que la visualización, para ser verdaderamente comprensiva, debe poder incluir dentro de su interfaz otra clase de archivos, para poder realizar una operación de Importar y Relacionar por ejemplo, y que automáticamente sean resaltados en la vista del documento las zonas correspondientes al árbol seleccionado. Una forma simplificada de lograr esto, sería no importando el documento, pero imprimiendo dentro de la misma imagen el texto plano y con un color diferente la parte seleccionada. Esto no es difícil, pero requiere considerar dónde y cómo mostrar el texto, que a veces puede ser muy extenso, y se perdería la información de formato, que es importante para distinguir rápidamente la estructura y a veces incluso el significado del texto.

En lo que respecta a los motivos del usuario, en la tipología encontramos tres principales: *enjoy* (usar la visualización por ocio u otros fines poco formales), *discover* (estudiar características de los datos activamente) y *present* (usar la visualización para mostrar los datos en un documento o presentación).

## ***Enjoy***

El prototipo tiene un claro atractivo estético que puede llevar a usuarios sin un interés particular a producir imágenes de sus documentos, e incluso arreglarlas y exportarlas, es decir, a realizar tareas compuestas de Organizar y Extraer. El prototipo soporta estas operaciones por lo que está bien adaptado para usos con este motivo.

## ***Discover***

Este es el motivo esencial por el que se utiliza una visualización de información como esta. La mayoría de las funciones descubiertas por esta evaluación son para mejorar este aspecto, incluyendo las más formales de la Capa 2, pero pocas añaden algo que realmente diferente a lo que se hizo en Experimentación, por lo que se puede evaluar bien a la visualización para estos fines aunque no perfectamente, ya que el prototipo no permitía descubrir patrones en, por ejemplo, textos muy extensos.

## ***Present***

Presentar información utilizando la visualización directamente es algo que se está practicando cada vez. Hay software de visualización especializado en esto, como Tableau, pero cualquier visualiza-

ción con una buena interfaz gráfica puede prestarse para este fin. En este prototipo no se desarrolló particularmente este aspecto ya que se consideró el menos relevante para esta etapa del desarrollo. La adición de botones, la integración de la vista de un documento con formato y más comodidad para realizar tareas son los factores clave que harían de esta visualización una mejor herramienta de presentación en vivo..



## 8 Conclusión

En este trabajo se ha diseñado una visualización para documentos-en-producción y desarrollado un prototipo que funciona con los datos que se pueden obtener de Google Drive, basados en una nueva forma de representación del texto versionado. Además, se estudiaron con este prototipo casos particulares de documentos en un dataset generado con tareas de estudiantes de la Facultad. El análisis de estos casos sugiere que una adecuada visualización puede permitir identificar secciones del texto que se corresponden con su estructura jerárquica o en su defecto tónica. Esto a su vez parece demostrar que los datos de posición de inserción en la producción de un documento contienen información de alto-nivel, como la estructura organizativa del texto y la segmentación por secciones. Sin duda la conclusión más relevante de este trabajo es haber mostrado las posibilidades que puede tener la visualización para documentos-en-producción para ayudar a entender su proceso productivo. Por último, la evaluación teórica del prototipo arrojó luz sobre las funcionalidades faltantes y las tensiones en el diseño, que muestran que aún no se ha explotado por completo esta estructura de datos y marcan el camino para futuras exploraciones del documento-en-producción.

En este capítulo se revisa el cumplimiento de los objetivos planteados en la sección 1.2, se hace un análisis de las implicancias de este trabajo, sus limitaciones y, por último, se vislumbra el trabajo futuro y las posibles áreas de expansión que podría tener un próximo prototipo.

### 8.1. CUMPLIMIENTO DE OBJETIVOS

El sistema THOM fue desarrollado para visualizar documentos-en-producción, lo cual se probó a través de casos de estudio obtenidos de un *dataset* generado por los mismos estudiantes de la Facultad. Estos casos demostraron la usabilidad del programa y qué información podía ser extraída a través de él. Por último, se realizó una evaluación exhaustiva de la visualización por medio del modelo teórico propuesto. Se concluye entonces que el objetivo general de este trabajo, definido en la sección 1.2.1, se cumple. Se analiza a continuación el cumplimiento de los objetivos específicos, definidos en la sección 1.2.2.

1. *Producir un dataset sobre el cual se trabajará en esta memoria:* Se generó un *dataset* con documentos compartidos por los mismos estudiantes de la Facultad, del cual se extrajeron los casos de estudio, tal como es descrito en la sección 5.1.
2. *Hacer una revisión sistemática de la literatura:* En el Capítulo 2 se revisa el trabajo relacionado que existe en visualización de textos y los enfoques al diseño de información que inspiran el diseño propuesto. Así se puede concluir que las soluciones actuales son insuficientes para el nivel de granularidad de la información disponible y que una nueva visualización, basada en los enfoques descritos, se hace necesaria.

3. *Diseñar e implementar las estructuras de datos necesarias para el preprocesamiento de los datos*: Se diseñó e implementó una particular estructura de datos que cumplía con los fines de preprocesamiento y representación de los datos requeridos por el esquema de visualización diseñado, la cual es descrita en el Capítulo 4.
4. *Diseñar y desarrollar una visualización de información para documentos-en-producción*: A través de un proceso iterativo de diseño, se llegó a una solución que cumplía teóricamente el requerimiento de que capturara el proceso de escritura en su irreversibilidad. Se implementó un prototipo de la visualización diseñada, con funciones de interactividad que permiten realizar tareas de alto nivel sobre los datos y así extraer la información buscada. Esta es descrita en Capítulo 3.
5. *Establecer indicadores y métricas preliminares para evaluar la soluciones, que a la vez permitan compararlas con otras (futuras) visualizaciones*: Se generó un modelo para la visualización de información en el Capítulo 6, a partir de una literatura diversa, que permitió en el Capítulo 7 evaluar el diseño generado y mostrar sus debilidades, superficiales y estructurales, que marcan un camino para seguir desarrollando el sistema.
6. *Probar la efectividad de las visualizaciones realizadas aplicándolas sobre el dataset del punto 1*: En el Capítulo 5 se analizan diferentes documentos y se demuestra la capacidad del sistema, también encontrándose sus limitaciones. Se demuestra que se pueden realizar tareas de alto nivel y extraer conclusiones no triviales usando la visualización.

## 8.2. IMPLICACIONES

Teniendo una forma de observar no sólo la superficie de un documento sino todo su proceso, se puede saber mucho más de éste y de su o sus autores. Propone una forma casi infalible de detectar copias textuales, patrones de “escritura perezosa” y “a la rápida”. Se podría exigir que proyectos, reportes y documentos oficiales dentro de una institución fueran creados y compartidos en Google Drive, para que su honestidad y cantidad de trabajo invertidos pudieran ser verificadas. Esta visualización no provee por ahora los medios para decir que un texto es “bueno”, pero al ser una visualización de un proceso más que de un producto puede mostrar diferentes manifestaciones de éste. Por esto mismo, podría ser utilizada para estudiar las estrategias de escritura de diferentes personas o grupos con fines científicos. Estos estudios irían acortando la distancia que existe entre la escritura y el resultado, permitiendo eventualmente decir cuáles estrategias llevan a mejores soluciones.

Ludwig Wittgenstein escribe su *Tractatus Logico-Philosophicus* en 1921 siguiendo una estricta estructura jerárquica. En ella, cada párrafo se encuentra a una cierta altura en el árbol, y da paso a otros párrafos que son el desglose del enunciado padre y que a su vez llevan a otros párrafos (el nodo raíz sería, naturalmente, el título del mismo libro. Una visualización web del libro se encuentra como “Tractatus’ Tree” en la Bibliografía de esta memoria). Esto sugiere dos posibles formas de leer formalmente la obra: *depth-first* y *breadth-first*. La primera es equivalente a la forma tradi-

cional, lineal, de leer un libro, acabando cada idea antes de continuar a la siguiente. La siguiente permite ir desarrollando todas las ideas de manera paralela. Esta estructura recursiva, fractal, que va profundizando en sí misma sin nunca repetirse, es de una manera extrema la misma que se adopta en la producción de texto científico, e.g., artículos académicos, en que cada paso que se da hacia abajo en el árbol aumenta la “resolución” de un tópico particular, tal como al hacer *zoom* para ver el detalle de una imagen. A diferencia del *Tractatus* de Wittgenstein, que está escrito de una manera completamente minimalista, la modalidad textual expositiva es de tipo redundante, por así decirlo, porque cada nodo debe ofrecer una panorámica de sus nodos subordinados. Ambos, sin embargo, se rigen por una estructura lógica que determina que desde los hechos (las hojas del árbol) se van obteniendo conclusiones, construcciones más generales o simplemente un resumen. Esta estructura a su vez reproduce la manera tradicional de organizar el conocimiento en la Ciencia y sugiere también una nueva estrategia de lectura, a saber, *bottom-up*, en la que partimos de los datos y terminamos leyendo el *abstract*. Seguir este camino no es nada trivial ya que lo que corresponde a las hojas en la estructura subyacente al texto está dividido y disperso por todas las secciones del documento.

A pesar de que buena parte del proceso de producción de un *paper* se debería seguir de reglas lógicas, es decir, de un protocolo, sigue siendo una tarea costosa para el humano académico promedio y actualmente existe investigación acerca de cómo alivianar este trabajo con asistencia computacional (di Sciascio et al., 2017). Sin embargo, poco conocimiento hay de cómo la evolución de un documento, su particular proceso de producción, puede informar al entendimiento de la escritura y a sistemas de asistencia o incluso escritura automatizada. La estrategia *bottom-up* puede parecer ineficiente y poco razonable para la lectura, pero ¿podría ser una buena estrategia de producción? Es algo que la visualización desarrollada podría ayudar a contestar, y en general para cualquier estrategia.

Más aún, usando la estructura de datos para codificar documentos parece hacer emerger información de alto-nivel a partir del ruidoso comportamiento de la escritura. Podría imaginarse eventualmente entrenar una red neuronal con estos documentos-en-producción para que aprenda algo más que a reproducir patrones estadísticos sin ninguna coherencia local o global, es decir, mejorar el entendimiento del lenguaje natural por parte de las máquinas, o al menos sus habilidades de comunicación técnica (si es que nos limitamos al texto organizado). Este trabajo cree haber dado un pequeño pero sólido paso en esa dirección.

Diferentes características de un texto más allá de su contenido, como la distribución y el uso de signos de puntuación, n-gramas de palabras, n-gramas de partes-del-discurso, conectores, morfología general de las palabras y desviaciones del uso formal del texto han demostrado ser muy efectivas, utilizadas en conjunto, para atribuir autoría de artículos de prensa, libros de literatura y blogs (Reicher et al., 2010) debido a que al no estar directamente relacionadas con el significado del texto, su uso es más bien inconsciente. Ésto sugiere que características identificables del texto-en-producción como distribución de errores grafomotores, eliminaciones, espaciamiento entre tpeos, al ser independientes del significado, podrían ser una buena fuente de información para la atribución de

autoría automática utilizando aprendizaje de máquina. Ésto quiere decir que al visualizar la dinámica del proceso de escritura como un mapa (*i.e.* una proyección bi-dimensional estática) estaríamos frente a las huellas estadísticas del autor, de una forma quizás más clara incluso que en el clásico *literature fingerprinting* (Keim & Oelke, 2007) donde se hace un análisis secuencial de distintas características estadísticas de libros.

### 8.3. LIMITACIONES

Processing es un lenguaje que permite programar aplicaciones visuales de manera rápida e intuitiva (de hecho, en la jerga de Processing, los programas se llaman *sketches*). Sin embargo, tiene sus limitaciones en cuanto a lo que es fácil hacer y lo que requiere mucho trabajo y esfuerzo computacional. Algunas características que hubieran sido útiles para este proyecto: caracteres vectorizados, *pop-ups* y objetos visuales “autoconscientes” de la región que ocupan en el espacio. Esto, por ejemplo, podría permitir que al pasar el mouse sobre un glifo, se dibuje un *pop-up* que muestre el contenido textual del nodo. Estas características serían más propicias de implementarse en Javascript utilizando D3, por lo que una siguiente versión podría considerar mudarse a este lenguaje que además está hecho para funcionar en los browsers modernos y trae la librería de visualización web por antonomasia. Processing no está hecho para funcionar en navegadores, sino para exportarse como aplicaciones ejecutables para Windows o Linux.

Una limitación técnica que rápidamente se vuelve teórica es qué ocurre con los textos que partieron en un documento y luego se copiaron en otro. Al copiar un fragmento este pierde la información de su proceso de escritura. “Anidar” documentos es, por lo tanto (y como siempre ha sido), imposible. Pero si se pudiese, abriría una posibilidad que vale la pena considerar: ¿puede un documento tener más de un origen? La estructura de datos propuesta sólo permite una raíz. Ésto es una especie de afirmación óptica sobre el documento, y puede que funcione en lo que se refiere a textos cuya estructura siga una jerarquía arbórea, pero quizás no para otros tipos de escritura, que responda a estructuras creativas.

Sobre la evaluación de la solución: Sin un cliente que ponga requerimientos o usuarios de prueba que actúen de agentes objetivos, el proceso de diseño corre el riesgo de estancarse en funcionalidades que poco aportan a la usabilidad del sistema e ignorar aspectos que podrían ser importantes para su éxito. Finalmente, el éxito de una visualización está dado por su adopción y uso en la sociedad, y por lo mismo es sorprendente la poca cantidad de publicaciones en esta área que incluyen estudios *in the wild* que puedan demostrar, sobre todo, que el sistema es utilizable y tiene algún valor de uso para algún tipo de usuario. Más aún, las visualizaciones presentadas muy rara vez se encuentran disponibles comercial o abiertamente, limitando severamente su valor académico.

Como Plaisant (2004) sugiere, hay que conciliar el deseo de innovar con el de producir visualizaciones exitosas, y para eso deben poder insertarse en contextos sociales siendo facilitadores de soluciones a problemas reales. La visualización aquí desarrollada tiene la ventaja de utilizar datos

de uso y acceso común que no han recibido mucha visibilidad hasta ahora. Un objetivo en el futuro próximo es hacer este sistema disponible al público a través de una página web donde la visualización esté anidada en el código HTML. Para conseguir esto debe estar programada en Javascript, pero este obstáculo puede ser superado sin reprogramar todo gracias a Processing.js.

## **8.4. TRABAJO FUTURO**

En el capítulo Evaluación se presentan los puntos más importantes que deben mejorarse en próximas versiones del sistema. La evaluación revela varias funcionalidades que deberían ser implementadas, llegando incluso a la estructura de datos, la cual debería tener la capacidad de cambiar de estado dinámicamente para su visualización, o de otra manera, la visualización debería poder derivar nuevos estados “comprimidos” a partir de una estructura de datos que guarda la información más fina, probablemente usando algo como estructuras virtuales.

Más radical aún es la idea de darle a la eliminación un rol estructural en la visualización, lo cual se cree que significa replantear su diseño desde el glifo. Esto sería lo correcto de querer satisfacer cabalmente las pretensiones que se fueron planteadas respecto al documento-en-producción y es algo que podría ser muy fructífero como desarrollo tanto para el campo de la visualización de texto como para el estudio de la escritura.

Otro camino que puede dar a investigación es empezar a automatizar el análisis sobre la estructura de datos para segmentar texto u otras aplicaciones que se le puedan dar usando teoría de grafos. Esto podría llevar a nuevas técnicas de segmentación tópica y al estudio de otras características que se puedan extraer de esta nueva estructura.

Aparte de todas estas consideraciones, se espera poder subir esta visualización a Internet para que sea utilizada por usuarios en el mundo, ofreciendo la posibilidad de crear imágenes de sus documentos y estudiarlos.



# Bibliografía

Alencar, A. B., de Oliveira, M. C. F., & Paulovich, F. V. (2012). Seeing beyond reading: a survey on visual text analytics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6), 476-492.

Borland, D., & Li, R. M. T. (2007). Rainbow color map (still) considered harmful. *IEEE computer graphics and applications*, 27(2).

Brehmer, M., & Munzner, T. (2013). A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12), 2376-2385.

Bringhurst, R. 2014. Los elementos del estilo tipográfico. 2ª ed. México, FCE. 471p. (Colección Libros Sobre Libros).

Cao N., Cui W. (2016) Overview of Text Visualization Techniques. In: Introduction to Text Visualization. Atlantis Briefs in Artificial Intelligence, vol 1. Atlantis Press, Paris.

Caporossi, G., & Leblay, C. (2011, October). Online writing data representation: a graph theory approach. In *International Symposium on Intelligent Data Analysis* (pp. 80-89). Springer, Berlin, Heidelberg.

Chevalier, F., Dragicevic, P., Bezerianos, A., & Fekete, J. D. (2010, April). Using text animated transitions to support navigation in document histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 683-692). ACM.

Deleuze, G. 2007. Pintura: el concepto de diagrama. 1ª ed. Buenos Aires, Cactus. 292p. (Serie Clases)

di Sciascio, C., Strohmaier, D., Errecalde, M., & Veas, E. (2017, March). WikiLyzer: Interactive Information Quality Assessment in Wikipedia. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*(pp. 377-388). ACM.

Dias, G., Alves, E., & Lopes, J. G. P. (2007, July). Topic segmentation algorithms for text summarization and passage retrieval: An exhaustive evaluation. In *AAAI* (Vol. 7, pp. 1334-1340).

Fry, B. J. (2000). *Organic information design* (Doctoral dissertation, Massachusetts Institute of Technology).

Grésillon, A., & Perrin, D. (2014). Methodology. From speaking about writing to tracking text production. *Handbook of writing and text production*, 10, 79-111.

- Hearst, M. A. (1994, June). Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* (pp. 9-16). Association for Computational Linguistics.
- Hearst, M. A. (1997). TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1), 33-64.
- Johnson, B., & Shneiderman, B. (1991, October). Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd conference on Visualization '91* (pp. 284-291). IEEE Computer Society Press.
- Keim, D. A., & Oelke, D. (2007, October). Literature fingerprinting: A new method for visual literary analysis. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on* (pp. 115-122). IEEE.
- Kim, S. (2011). *Modeling and visualization of version-controlled documents* (Doctoral dissertation, Georgia Institute of Technology).
- Kim, S., & Lebanon, G. (2010, August). Local space-time smoothing for version controlled documents. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (pp. 552-560). Association for Computational Linguistics.
- Kim, S., Dillon, J. V., & Lebanon, G. (2012). Cumulative Revision Map. *arXiv preprint arXiv:1205.3205*.
- Kindlmann, G. (2004, May). Superquadric tensor glyphs. In *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization* (pp. 147-154). Eurographics Association.
- Kindlmann, G., & Scheidegger, C. (2014). An algebraic process for visualization design. *IEEE transactions on visualization and computer graphics*, 20(12), 2181-2190.
- Kucher, K., & Kerren, A. (2015, April). Text visualization techniques: Taxonomy, visual survey, and community insights. In *Visualization Symposium (PacificVis), 2015 IEEE Pacific* (pp. 117-121). IEEE.
- Lamqaddam, H., & Aerts, J. 2016. InVITe: towards intuitive visualization of iterations over text. In *Proceedings of the Eurographics / IEEE VGTC Conference on Visualization: Posters* (EuroVis '16). Eurographics Association, Goslar Germany, Germany, 57-59.
- Latif, M. M. A. (2008). A state-of-the-art review of the real-time computer-aided study of the writing process. *International Journal of English Studies*, 8(1), 29-50.
- Lebanon, G., Mao, Y., & Dillon, J. (2007). The locally weighted bag of words framework for document representation. *Journal of Machine Learning Research*, 8(Oct), 2405-2441.

- Lee, Y. Y., Lin, C. C., & Yen, H. C. (2006, January). Mental map preserving graph drawing using simulated annealing. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualization-Volume 60* (pp. 179-188). Australian Computer Society, Inc..
- Mao, Y., Dillon, J., & Lebanon, G. (2007). Sequential document visualization. *IEEE transactions on visualization and computer graphics*, 13(6).
- Nguyen, Q., Eades, P., & Hong, S. H. (2013, February). On the faithfulness of graph visualizations. In *Visualization Symposium (PacificVis), 2013 IEEE Pacific*(pp. 209-216). IEEE.
- Perrin, D. and Wildi, M. 2010. Statistical modeling of writing processes, in *Traditions of Writing Research*. C. BAZERMAN, et al., Eds.: Routledge, 378-393.
- Plaisant, C. (2004, May). The challenge of information visualization evaluation. In *Proceedings of the working conference on Advanced visual interfaces* (pp. 109-116). ACM.
- Reicher, T., Krišto, I., Belša, I., & Šilić, A. (2010). Automatic authorship attribution for texts in croatian language using combinations of features. *Knowledge-Based and Intelligent Information and Engineering Systems*, 21-30.
- Richmond, K., Smith, A. J., & Amitay, E. (1997). Detecting subject boundaries within text: A language-independent statistical approach.
- Shneiderman, B. (1996, September). The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on* (pp. 336-343). IEEE.
- Šilić, A., & Bašić, B. (2010). Visualization of text streams: A survey. *Knowledge-based and intelligent information and engineering systems*, 31-43.
- Viégas, F. B., Wattenberg, M., & Dave, K. (2004, April). Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 575-582). ACM.
- Wang, D., Olson, J. S., Zhang, J., Nguyen, T., & Olson, G. M. (2015, April). DocuViz: visualizing collaborative writing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 1865-1874). ACM.
- Wang, D., Olson, J. S., Zhang, J., Nguyen, T., & Olson, G. M. (2015). How Students Collaboratively Write using Google Docs. *iConference 2015 Proceedings*.
- Ward, M. O. (2008). Multivariate data glyphs: Principles and practice. *Handbook of data visualization*, 179-198.

Wise, J. A. (1999). The ecological approach to text visualization. *Journal of the Association for Information Science and Technology*, 50(13), 1224.

Ziemkiewicz, C., & Kosara, R. (2009). Embedding information visualization within visual representation. In *Advances in Information and Intelligent Systems*(pp. 307-326). Springer Berlin Heidelberg.

### ***Páginas web***

Draftback home page, <https://draftback.com/> (visitado octubre 2016)

TextVis home page, <http://textvis.lnu.se/> (visitado octubre 2016)

Processing.js home page, <http://processingjs.org> (visitado julio 2017)

Tractatus' Tree, <https://pbellon.github.io/tractatus-tree/#/> (visitado julio 2017)

# Anexos

## Instrucciones de uso

Todo el código desarrollado para este trabajo se puede encontrar en el repositorio

<https://github.com/kidonkey/Memoria-Irreversible>

el cual hay que clonar y usar el archivo Indels.py para convertir el archivo json.txt que entrega Drive en un CSV para entregarlo a thom1.pde, que permite visualizarlo e interactuar con él.

Es necesario tener Python y Processing 3 con la librería peascam instalada.