



UNIVERSIDAD DE GRANADA

TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

Aplicación de gestión de matrices Harris en arqueología

HAMMON: Harris Matrix Manager on desktop

Autor

Carlos Martín Sánchez

Tutor

Daniel Sánchez Fernández



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

Granada 15 de junio de 2025





Aplicación de gestión de matrices Harris en arqueología

HAMMON: Harris Matrix Manager on desktop

Autor

Carlos Martín Sánchez

Tutor

Daniel Sánchez Fernández

Aplicación de gestión de matrices Harris en arqueología: HAMMON: Harris Matrix Manager on desktop

Carlos Martín Sánchez

Palabras clave: arqueología, unidad estratigráfica, matriz de Harris, diagrama de Hasse, Python, Tkinter

Resumen

Se ha desarrollado una aplicación que permite la creación y edición de matrices de Harris para la organización de estratos de una excavación arqueológica. Está pensada para usuarios con poca experiencia usando aplicaciones o que están empezando a aprender y hacer sus primeras matrices de Harris. Aunque ya existan alternativas como *ArchEd*, *Stratify* y *Harris Matrix Composer*, la mayoría tienen mucha antigüedad o aspectos que podrían ser mejorables. Se realizó un análisis de requisitos para tener una idea más concreta de las características que debía presentar la aplicación. Cabe mencionar que se contó con la ayuda de un arqueólogo para fijar los requisitos.

Como los arqueólogos suelen trabajar en lugares con baja o nula conexión a internet, se decidió diseñarla como aplicación de escritorio. Se desarrolló una interfaz diseñada especialmente para usuarios nuevos o poco experimentados. El lenguaje de programación utilizado es *Python* junto a la biblioteca *Tkinter*, que contiene módulos para el desarrollo de GUI. Para cargar información de matrices de Harris, se usan ficheros con formato *.csv* que deben mantener una serie de reglas cuyo cumplimiento es comprobado por parte de la aplicación. Se decidió utilizar *.csv* para minimizar el tamaño del fichero y facilitar su edición desde aplicaciones como *Microsoft Excel* o similares. Otro motivo para usar ficheros *.csv* fue para evitar escribir manualmente la matriz en la aplicación.

El objetivo de esta memoria es informar sobre el desarrollo del proyecto y sus fortalezas y debilidades, y facilitar la comprensión sobre el uso de la aplicación mediante un manual de usuario.

El resultado final fue una aplicación completamente funcional que cumple con los objetivos marcados inicialmente dentro del marco de tiempo que se planteó.

Harris Matrix Management App in archaeology: HAMMON: Harris Matrix Manager on desktop

Carlos Martín Sánchez

archaeology, stratigraphic unit, Harris matrix, Hasse diagram, Python, Tkinter

Abstract

In this project an application that allows the creation and edition of Harris matrices for strata organization of an archaeological excavation has been developed. It's thought for users with little experience using applications or starting to learn about and composing their firsts Harris matrices. Although alternatives such as *ArchEd*, *Stratify* and *Harris Matrix Composer* exists, many if not all are several years old or have room for improvement. In order to have a more specific idea of the characteristics the application should present, a requirement analysis was made. It's worth noting that the analysis was developed with the help of an archaeologist to specify the requirements.

Archaeologist usually work in places with low or null Internet connection, it was decided to develop a desktop application. The interface was designed specially for new or less experimented users. The programming language used is *Python* along the *Tkinter* library, which contains modules for GUI development. Files with *.csv* extension are used to upload Harris matrices data, which must maintain certain rules, that are checked by the application. The decision to use *.csv* files was fundamented on several reasons. The first one was to minimize file size. The second reason was to make its edition easier using applications such as *Microsoft Excel*, for example. The third reason was to avoid writing the entire matrix manually in the application.

The goal of this report is to inform about the project development, its strong and weak points and allow an easier understanding of the use of the application with an user manual.

In the end, the final result was a completely functional application that fulfills with the goals initially planned in the time frame.

Yo, **Carlos Martín Sánchez**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77153684T, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Para saber las licencias que contienen el código fuente y la memoria, véase Apéndice A: Licencias.

Fdo: Carlos Martín Sánchez

Granada a 15 de junio de 2025.

D. **Daniel Sánchez Fernández**, Profesor del Área de Ciencias de la Computación de Inteligencia Artificial del Departamento de Ciencias de la Computación de Inteligencia Artificial de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado *Aplicación de gestión de matrices Harris en arqueología, HAMMON: Harris Matrix Manager on desktop*, ha sido realizado bajo su supervisión por **Carlos Martín Sánchez**, y autoriza la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada a 15 de junio de 2025.

El tutor:



Daniel Sánchez Fernández

Agradecimientos

A mi familia y especialmente a mis padres y a mi hermana, por su apoyo constante a lo largo de toda la carrera; y a mis abuelos, por su cariño incondicional.

A mis amigos del Colegio Mayor, por su amistad y por haber hecho de mi estancia en el Colegio una experiencia inolvidable.

A todos los profesores que en mayor o menor medida me las dieron herramientas para llegar hasta donde estoy ahora.

Al Colegio Mayor Cardenal Cisneros por haber sido mi segunda casa durante 5 años.

A Daniel, por haber decidido ser mi tutor y por su guía durante el desarrollo de este trabajo.

A Manuel, el arqueólogo que me aconsejó durante la realización del trabajo.

En memoria de mi abuelo Paco.

Índice general

Índice general	13
Índice de figuras	17
Índice de tablas	19
1. Introducción	21
1.1. Motivación	21
1.2. Contexto	22
1.3. Estructura	26
2. Objetivos	29
2.1. Alcance de los objetivos	29
3. Planificación	31
3.1. Fases	31
3.2. Costes	34
4. Estado del arte	35
4.1. Trabajos previos	35
4.1.1. ArchEd	35
4.1.2. Stratify	37
4.1.3. Harris Matrix Composer (HMC)	40
5. Especificación de requisitos	41
5.1. Requisitos funcionales	41
5.2. Requisitos no funcionales	44
5.3. Requisitos de información	44
5.4. Casos de uso	46
5.4.1. Descripción de los actores	46
5.4.2. Descripción de Caso de Uso	46
5.4.3. Diagramas de Modelo de Casos de Uso	81
5.4.4. Diagramas de Actividad	86

6. Diseño	95
6.1. Plataforma	95
6.1.1. Conclusión	97
6.2. Diseño de la interfaz	97
6.2.1. Ventana de las figuras	99
6.2.2. Nodos	100
6.2.3. Información de la matriz	101
6.2.4. Estética	108
6.2.5. Filtrado	108
6.3. Prevención de errores	109
6.4. Redundancia y reducción transitiva	109
6.5. Icono	110
7. Implementación	111
7.1. Elección del lenguaje	111
7.1.1. C++	111
7.1.2. Python	112
7.1.3. Conclusión	113
7.2. Formato de los archivos	113
7.2.1. Aplicación	113
7.2.2. Ficheros de matrices	113
7.3. Comprobación de errores	115
7.3.1. Errores en los ficheros	115
7.3.2. Añadir relaciones	117
7.3.3. Añadir equivalencia	118
7.3.4. Añadir agrupación	118
7.3.5. Añadir o editar una unidad estratigráfica	119
7.3.6. Añadir o editar un hecho	119
7.4. Diagrama de clases	119
8. Conclusiones y trabajos futuros	121
8.1. Conclusiones	121
8.2. Trabajos futuros	123
A. Licencias	124
A.1. Licencia de la memoria	124
A.2. Licencia del código fuente	124
B. Glosario	125
C. Matriz	127

D. Manual de Usuario	131
D.1. Inicio	131
D.2. Barra de herramientas	132
D.3. Ventana de figuras	133
D.4. Barra de herramientas de vista	133
D.5. Unidades	135
D.5.1. Añadir unidades	135
D.5.2. Eliminar unidades	136
D.5.3. Editar unidad	136
D.6. Relaciones	136
D.6.1. Añadir relación	137
D.6.2. Eliminar relación	137
D.7. Equivalencias	138
D.8. Hechos	138
D.8.1. Información interna	138
D.8.2. Agrupamiento	139
D.8.3. Aplicar hecho	140
D.9. Estética	140
D.9.1. Cambiar color de las aristas	141
D.9.2. Cambiar la forma de la línea de la arista	142
D.9.3. Cambiar la forma de la flecha de la arista	142
D.9.4. Cambiar anchura de la arista	142
D.9.5. Cambiar el color de una fase	143
D.10. Filtrado	143
D.10.1. Filtrar	143
D.10.2. Zoom	143
D.10.3. Opciones	144
D.10.4. Selección de columnas	144
D.10.5. Guardar	145
D.10.6. Bibliotecas	145
Bibliografía	147

Índice de figuras

1.1. Matriz de Harris de Nawarla Gabarnmang. Extraída de [HG17].	22
1.2. La relación $A \rightarrow B$ se muestra en verde en el grafo.	23
1.3. La equivalencia $B = C$ se muestra en rojo en el grafo.	24
1.4. El hecho D agrupa a los nodos B y C.	25
3.1. Metodología de cascada para desarrollo software. Tomada de [Sen21]	31
3.2. Diagrama de Gantt.	33
4.1. Interfaz de <i>ArchEd</i> . Tomada de [Hun+98].	36
4.2. Matriz de Harris en <i>ArchEd</i> . Tomada de [Hun+98].	37
4.3. Interfaz de <i>Stratify</i>	38
4.4. Matriz de Harris en <i>Stratify</i>	39
4.5. Interfaz de <i>Harris Matrix Composer</i> con la matriz por defecto. Tomada desde mi ordenador.	40
5.1. Diagrama de Casos de Uso. Barra de herramientas	82
5.2. Diagrama de Casos de Uso. Vista de figuras	83
5.3. Diagrama de Casos de Uso. Edición de listas	84
5.4. Diagrama de Casos de Uso. Estilización	85
5.5. Diagrama de Casos de Uso. Filtrado	86
5.6. Diagrama de Actividad. Vista completa.	87
5.7. Diagrama de Actividad. Detalle A.	88
5.8. Diagrama de Actividad. Detalle B.	89
5.9. Diagrama de Actividad. Detalle C.	90
5.10. Diagrama de Actividad. Detalle D.	91
5.11. Diagrama de Actividad. Detalle E.	92
5.12. Diagrama de Actividad. Detalle F.	93
6.1. Boceto inicial de la interfaz de la aplicación.	98
6.2. Detalle: Barra de herramientas para la vista	98
6.3. HAMMON: Aplicación con la pestaña Grafo abierta.	99
6.4. HAMMON: Aplicación con la pestaña Matriz abierta.	100

6.5. En la ventana de HAMMON se muestra un grafo de nodos con los tres tipos de nodos. En el sentido contrario de las agujas del reloj: A es una unidad positiva (cuadrado), B es una unidad negativa (círculo) mientras que D es un hecho (hexágono).	101
6.6. HAMMON: Unidades.	102
6.7. HAMMON: Relaciones.	103
6.8. HAMMON: Equivalencias.	104
6.9. HAMMON: Hechos - Información interna.	105
6.10. HAMMON: Hechos - Agrupamiento.	106
6.11. HAMMON: Edición estética.	107
6.12. HAMMON: Filtrado	108
6.13. Icono de la aplicación	110
7.1. Fichero CSV con formato correcto.	114
7.2. Fichero csv tres columnas <i>Nombre</i> .	116
7.3. Columnas con nombre duplicado en la aplicación.	116
7.4. Diagrama de las clases de HAMMON.	120
C.1. Extracto de la matriz de ejemplo proporcionada por el arqueólogo 1.	127
C.2. Submatriz 1. Proporcionada por el arqueólogo.	128
C.3. Extracto de la matriz de ejemplo proporcionada por el arqueólogo 2.	129
C.4. Submatriz 2. Proporcionada por el arqueólogo.	130
D.1. Interfaz de <i>HAMMON</i> al iniciar la aplicación.	131
D.2. Barra de herramientas.	132
D.3. Ventana con archivos del sistema.	133
D.4. Barra de herramientas de vista.	133
D.5. HAMMON: Matriz con leyenda.	134
D.6. Pestaña de unidades.	135
D.7. Ventana para añadir unidad.	136
D.8. Pestaña de relaciones.	137
D.9. Pestaña de equivalencias.	138
D.10. Pestaña de hechos: Información interna.	139
D.11. Pestaña de hechos: Agrupamiento.	140
D.12. Pestaña de hechos: Edición estética.	141
D.13. Ventana de selección de color.	142
D.14. Sección de filtrado.	143

Índice de tablas

5.1. Descripción del actor	46
5.2. Atributos del actor	46
5.3. Caso de uso Nueva plantilla	47
5.4. Caso de uso Cargar matriz.	48
5.5. Caso de uso Guardar.	49
5.6. Caso de uso Guardar como.	50
5.7. Caso de uso Deshacer.	51
5.8. Caso de uso Rehacer.	52
5.9. Caso de uso Redundancia.	53
5.10. Caso de uso Mover la figura.	54
5.11. Caso de uso Zoom.	55
5.12. Caso de uso Deshacer vista	56
5.13. Caso de uso Rehacer vista	57
5.14. Caso de uso Vista original.	58
5.15. Caso de uso Guardar figura.	59
5.16. Caso de uso Mostrar/ocultar leyenda.	60
5.17. Caso de uso Añadir nodo.	61
5.18. Caso de uso Eliminar nodo.	61
5.19. Caso de uso Editar nodo.	62
5.20. Caso de uso Añadir relación.	63
5.21. Caso de uso Eliminar relación.	64
5.22. Caso de uso Añadir equivalencia.	65
5.23. Caso de uso Eliminar equivalencia.	65
5.24. Caso de uso Añadir hecho.	66
5.25. Caso de uso Eliminar hecho.	67
5.26. Caso de uso Aplicar hecho.	67
5.27. Caso de uso Cambiar color de las relaciones.	68
5.28. Caso de uso Cambiar la forma de la línea (relación).	69
5.29. Caso de uso Cambiar la forma de la flecha (relación).	70
5.30. Caso de uso Cambiar la anchura (relación).	71
5.31. Caso de uso Cambiar color de las equivalencias.	72
5.32. Caso de uso Cambiar la forma de la línea (equivalencia).	73
5.33. Caso de uso Cambiar la forma de la flecha (equivalencia).	74

5.34. Caso de uso Cambiar la anchura (equivalencia).	75
5.35. Caso de uso Cambiar el color (Fase).	76
5.36. Caso de uso Filtrar nodos.	77
5.37. Caso de uso Guardar filtrado.	78
5.38. Caso de uso Zoom (filtrado).	79
5.39. Caso de uso Cambiar opciones de filtrado.	80
5.40. Caso de uso Salir.	81
6.1. Comparación entre Web y Escritorio	96

Capítulo 1

Introducción

En la introducción de la memoria primero se explicará la motivación de realizar este proyecto, seguido del contexto que fundamenta por qué el objetivo de este trabajo es importante, para finalmente concluir con la estructura que sigue esta memoria.

1.1. Motivación

El latinismo “*Verba volant, scripta manent*” se le atribuye al senador romano Tito Cayo, y significa “las palabras vuelan, lo escrito queda”. Si bien hay dudas sobre quién lo dijo primero, en la actualidad esta frase se suele utilizar para remarcar que es la palabra escrita la que pasa la prueba del tiempo. Preservar la información es esencial para que las futuras generaciones puedan aprender de los errores de sus antepasados, de sus costumbres o de sus ideas. Ese es, en parte, el propósito principal de la arqueología, la ciencia que estudia las civilizaciones antiguas a través de sus restos materiales.

Durante una excavación arqueológica, se mantiene un registro de todos los estratos y en qué orden aparecen. Al remover la tierra durante una excavación, el orden original de los estratos se pierde, siendo importante registrar la organización original que permita tener más información de cara a futuras excavaciones.

Este trabajo, además de ser útil para la arqueología y la preservación de la historia, también me parece interesante de manera personal. Como interesado en la historia que soy, creo que es importante preservar y difundir la información no solo de pueblos antiguos o la prehistoria, sino también información del presente, porque aunque ahora tengamos mucha información disponible, nada nos asegura que en el futuro lejano esta información llegue a las generaciones futuras, siendo más difícil comprender sucesos históricos o la vida de la gente común en el pasado.

1.2. Contexto

En una excavación arqueológica la organización de los estratos es información fundamental a la hora de interpretar y contextualizar correctamente los hallazgos realizados sobre la misma. El problema es que a la hora de extraer artefactos se altera la estratificación del yacimiento.

Para evitar que la información sobre la estratificación se perdiese, había que desarrollar algún método para guardarla. Dicho método llegaría en 1973 de la mano del Dr. Edward C. Harris, la mente detrás de las **matrices de Harris**. La matriz de Harris es un diagrama secuencial que define relaciones entre distintas **unidades estratigráficas**. Las unidades estratigráficas son volúmenes de roca que se clasifican en función de sus características particulares [TNI08; CHA75; Hed58]. En la figura 1.1 se puede ver una matriz de Harris con unidades estratigráficas representadas como cuadrados y las relaciones como flechas.

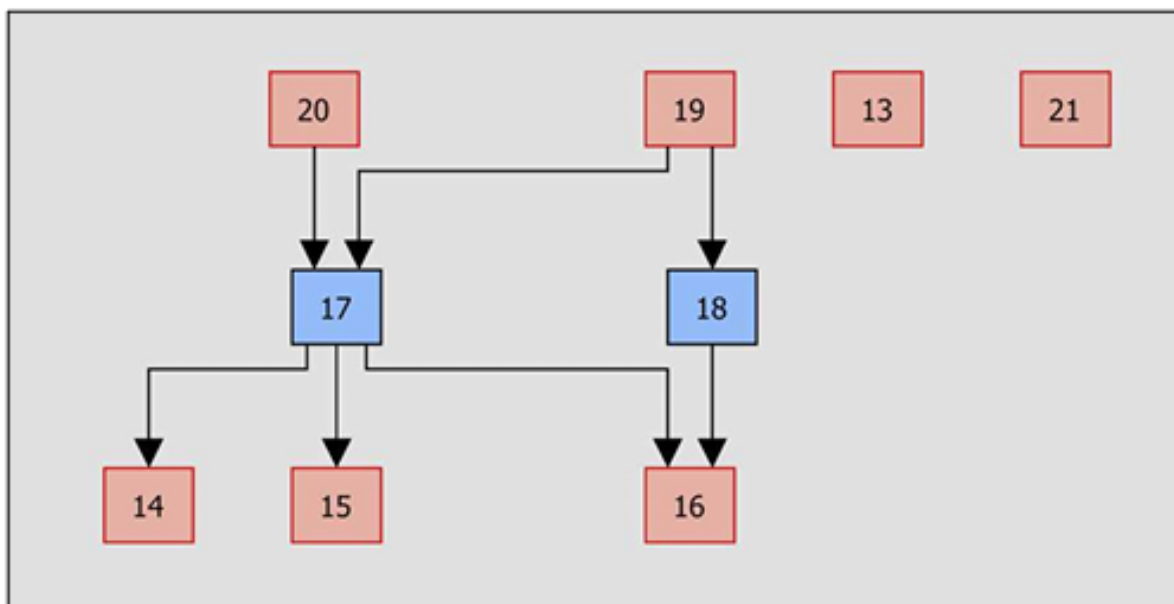


Figura 1.1: Matriz de Harris de Nawarla Gabarnmang. Extraída de [HG17].

Las matrices de Harris son fundamentales en la arqueología, hasta el punto en que es de obligado cumplimiento incluirlas siempre en todos los informes arqueológicos. Con la llegada de las nuevas tecnologías, se han desarrollado herramientas para facilitar la creación y edición de matrices de Harris, no solo por su importancia sino por la complejidad que puede llegar a alcanzar si en una excavación hay un gran número de estratos. Algunas aplicaciones de gestión de matrices de Harris a mencionar son *ArchEd*, *Stratify* y *Harris Matrix Composer*.

La matriz de Harris se puede expresar como un grafo dirigido de nodos y, aunque sea poco común, también como una matriz tradicional. Normalmente cuando se hable de matriz de Harris se estará refiriendo al grafo. En una matriz de Harris se encuentran los siguientes elementos:

- **Relaciones:** Se refiere a las relaciones padre-hijo entre dos unidades. La unidad padre está por encima de la unidad hijo. En la matriz se representa con un flecha o una arista.

La figura 1.2 muestra una relación entre dos unidades positivas A y B, donde A se encuentra encima de B. A la izquierda de la imagen se puede ver el grafo de nodos y a la derecha una tabla con *Unidad 1* y *Unidad 2*. Para cada fila la *Unidad 1* está por encima de la *Unidad 2*.

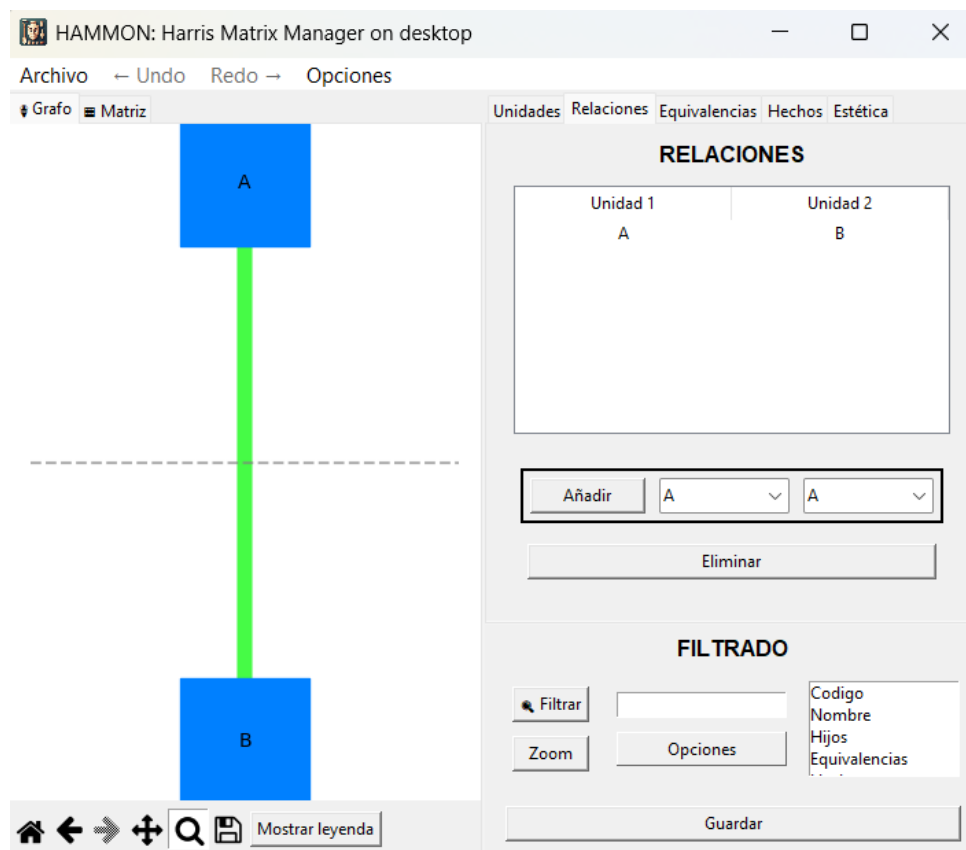


Figura 1.2: La relación $A \rightarrow B$ se muestra en verde en el grafo.

- **Equivalencias:** Si a dos o más estratos se les ha asignado distinto número o nombre para posteriormente darse cuenta que en realidad eran el mismo, se pueden indicar que son el mismo mediante una equivalencia.

La figura 1.3 muestra una equivalencia entre las unidades B y C, que se representa como una línea roja horizontal. En este contexto, para cada fila de la tabla las columnas *Unidad 1* y *Unidad 2* señalan las unidades que son equivalentes.

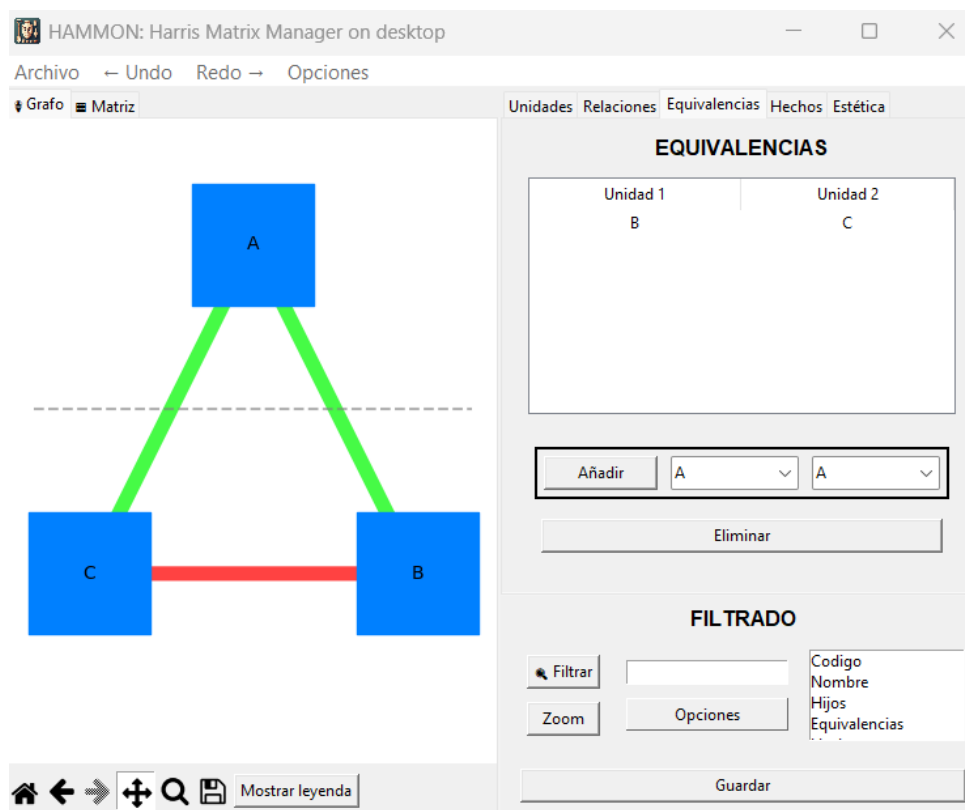


Figura 1.3: La equivalencia $B = C$ se muestra en rojo en el grafo.

- **Hechos:** Son agrupaciones de varias unidades estratigráficas. Se usan para que la matriz sea más legible. En el contexto del proyecto, se va a distinguir el hecho como el nodo que agrupa unidades y otros hechos, del agrupamiento que indica qué unidades o hechos están agrupados dentro del hecho agrupador.

En la figura 1.4 el nodo con forma de hexágono representa una agrupación de nodos. En la tabla a la derecha la columna *Hecho* indica los hechos que agrupan y la columna *Unidades* indica qué unidad o hecho está siendo agrupado.

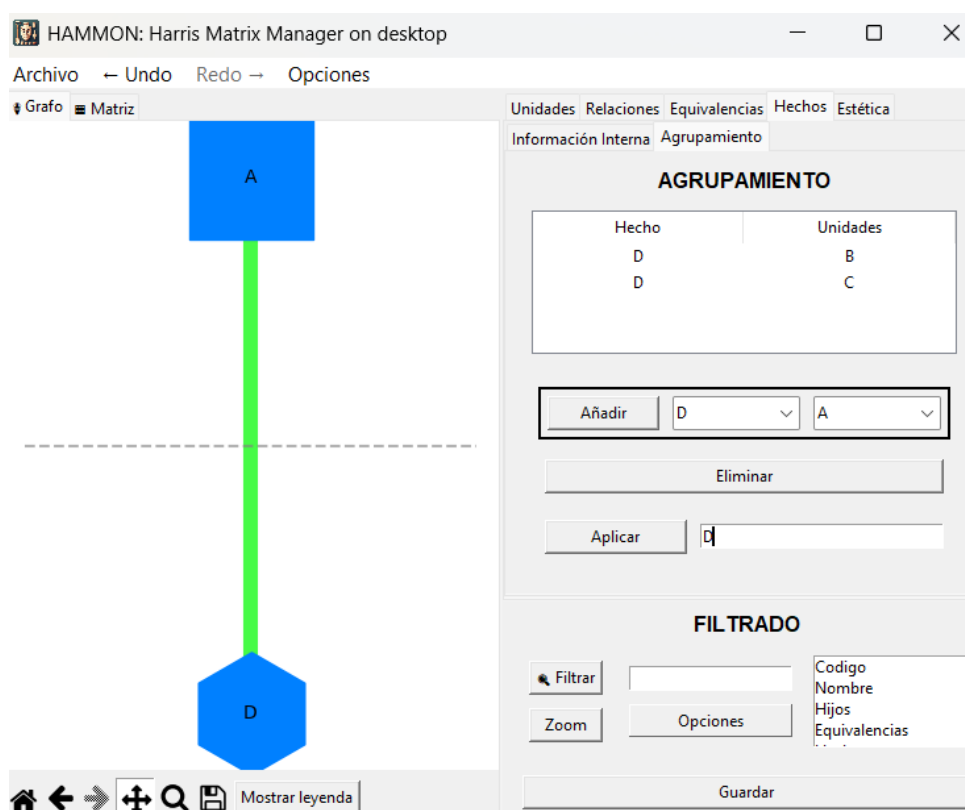


Figura 1.4: El hecho D agrupa a los nodos B y C.

Se pueden ver las definiciones de estos conceptos en Apéndice B: Glosario.

Tras analizar las características de la matriz de Harris, se llegó a la conclusión de que se puede representar como un **diagrama de Hasse**. Los diagramas de Hasse son un tipo de grafo dirigido acíclico (no tienen ciclos) a los que se les aplica la reducción transitiva (véase Sección 6.4: Redundancia y reducción transitiva), y se usan para ordenar conjuntos parcialmente ordenados [ES12].

El objetivo final de este trabajo es crear una aplicación de gestión de matrices de Harris completamente funcional que no solo permita cambiar la información que contiene la matriz sino

que dichos cambios también se expresen en la representación visual de la matriz. Precisamente es del objetivo final de donde se ha tomado la idea del nombre con el que se bautizará a la aplicación. Su función es la gestión de matrices de Harris, y como se explicará posteriormente, será una aplicación de escritorio. Desde aquí surgió el nombre *Harris Matrix Manager on desktop*. Como acrónimo se decidió *HAMMON* (**H**arris **M**atrix **M**anager **o**n **d**esktop), que además coincide con el nombre de **Baal Hammon**, un dios fenicio asociado con el dios griego Cronos [Cag91]. Cabe recalcar que a los arqueólogos se les proporcionará una versión completamente compilada de la aplicación.

1.3. Estructura

Esta memoria es bastante extensa, por lo que se dará una breve explicación sobre el contenido de cada capítulo.

- El Capítulo 1: Introducción empieza explicando algunas definiciones y conceptos que serán relevantes a lo largo de la memoria y ayudará a tener un mayor contexto sobre algunas de las ideas que se hablen. Posteriormente se expresan las razones por las que se decidió desarrollar este proyecto. Finalmente, se introduce la estructura que tendrá el documento.
- En el Capítulo 2: Objetivos se expresan los distintos objetivos que se pretenden alcanzar con la realización de este trabajo y el alcance que pretenden tener estos.
- El Capítulo 3: Planificación comienza explicando las distintas fases de desarrollo del trabajo, junto a la duración estimada de cada una. También se enseña un diagrama de Gantt que representa la planificación de una manera más visual. También se mencionan los costes económicos estimados del proyecto.
- El Capítulo 4: Estado del arte habla acerca de distintas aplicaciones para crear matrices de Harris, y se mencionan brevemente algunas de sus características principales y su funcionamiento.
- En el Capítulo 5: Especificación de requisitos se enumeran los distintos requisitos que debe cumplir el programa. Posteriormente se describe al actor de los casos de uso (usuario) y a los casos de uso del programa. Una vez explicados, se describen los casos de uso y se muestran los modelos de caso de uso y los diagramas de actividad.
- El Capítulo 6: Diseño menciona las distintas cuestiones de diseño que surgieron sobre la creación de la aplicación, por ejemplo, si debía ser web o de escritorio, o el formato de la interfaz.
- El Capítulo 7: Implementación explica cuestiones relacionadas con la programación de la aplicación, más específicamente el lenguaje de programación usado, el formato de los archivos que se cargan y la comprobación de errores en la información de la matriz. También incluye el diagrama de clases de la implementación de la aplicación.

- El Capítulo 8: Conclusiones y trabajos futuros es el último capítulo de la memoria, que cierra con las conclusiones extraídas tras finalizar el trabajo. También habla acerca de posibles mejoras que se podrían hacer a futuro.

Esta memoria también cuenta con varios apéndices relevantes:

- El Apéndice A: Licencias muestra la licencia que tiene el programa y el documento de la memoria.
- El Apéndice B: Glosario contiene las definiciones fundamentales que se han utilizado en este documento.
- El Apéndice C: Matriz contiene dos submatrices de Harris desarrolladas en HAMMON. Ambas submatrices forman parte de una matriz más grande proporcionada por el arqueólogo, por lo que se ha usado HAMMON con datos reales.
- En el Apéndice D: Manual de Usuario se encuentran las instrucciones de uso de la aplicación.

Al término del documento se encuentran las bibliografías usadas.

Capítulo 2

Objetivos

El principal objetivo de este TFG es la creación de una aplicación que permita a los arqueólogos generar y editar matrices de Harris de manera fácil y sencilla. Los objetivos más concretos se han establecido siguiendo las indicaciones del marco de trabajo **SMART** [Dor81]. Del objetivo principal parten los siguientes objetivos concretos:

- **OBJ-1.** Investigar las distintas alternativas que ya existen y analizar sus ventajas y desventajas.
- **OBJ-2.** Estudiar las distintas alternativas de diseño que se pueden tomar para desarrollar la aplicación.
- **OBJ-3.** Otorgar a la aplicación una interfaz lo más sencilla e intuitiva posible para los usuarios con menos experiencia usando aplicaciones.
- **OBJ-4.** Desarrollar el proyecto, tanto el software como la memoria como código abierto u *open source*, dando más transparencia al trabajo y permitiendo a otras personas usar su contenido libremente.

2.1. Alcance de los objetivos

El principal objetivo de este trabajo es proporcionar a los arqueólogos una alternativa a la hora de disponer de un programa apropiado para crear y editar matrices de Harris. El programa estará destinado principalmente a personas con poca experiencia usando ordenadores o que están empezando en el desarrollo de matrices de Harris (como estudiantes de arqueología, por ejemplo). Esta memoria tiene como fin ayudar al lector a tener una mejor comprensión sobre el desarrollo del trabajo y cómo funciona el programa. Además, tanto el programa como este informe se liberarán bajo una licencia libre para que cualquier persona pueda utilizarlos.

Capítulo 3

Planificación

En este capítulo se expondrá la planificación propuesta para desarrollar este proyecto, esto es, una breve explicación de las distintas fases y el tiempo estimado para cada una. Finalmente, se especificarán los costes que hubiera tenido este proyecto si se hubiese encargado profesionalmente.

3.1. Fases

La organización temporal del proyecto se basó en la **metodología de cascada**, también conocida como modelo cíclico de vida secuencial linear. La metodología de cascada cuenta con 5 fases donde una fase debe ser terminada antes de pasar a la siguiente, de ahí lo de *linear*. Las fases en cuestión son: análisis, diseño, implementación, pruebas y mantenimiento.

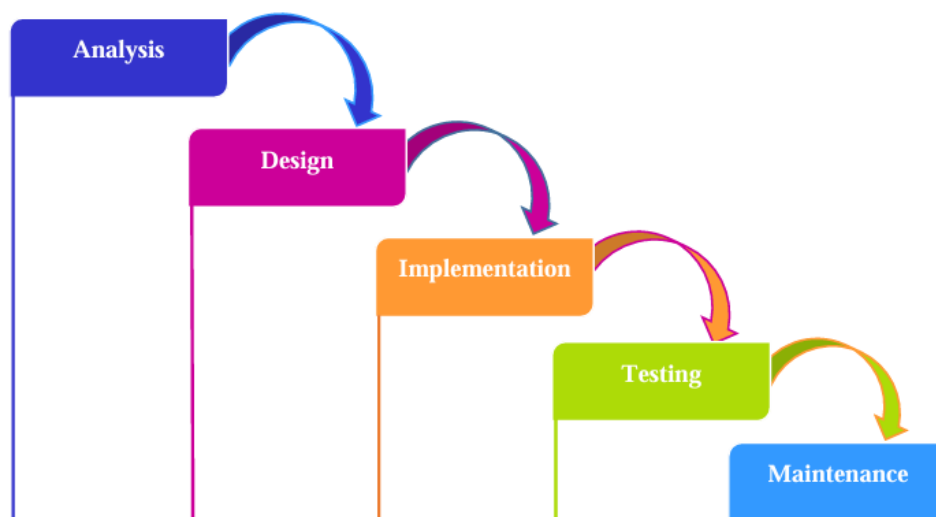


Figura 3.1: Metodología de cascada para desarrollo software. Tomada de [Sen21]

La figura 3.1 muestra cómo funciona visualmente la metodología de cascada. Para la planificación se eliminó la fase de mantenimiento y se añadieron las de búsqueda de la información y escritura de la memoria, que se explican en detalle más adelante [Sen21]. Se puede ver el diagrama de Gantt de la planificación en 3.2.

A continuación se explican las fases: cómo se han llamado, qué se ha trabajado en cada una y el tiempo que se ha tardado en cada una se muestra visualmente en la figura 3.2.

- **Fase 1: Planificación.** Organización de los tiempos para cada fase, es decir, hasta cuándo va a durar la realización de cada fase. Cabe mencionar que se trata de una estimación de lo que se tardará en terminar cada fase.
- **Fase 2: Búsqueda de información.** Búsqueda de información que ayudará en el desarrollo del proyecto, ya sea sobre otras aplicaciones de diseño de matrices de Harris, documentación para la programación de la aplicación o bibliografía para la escritura de la memoria. Esta fase es la que abarca más tiempo pues es complementaria al resto de fases que le siguen ya que en todas habrá que recoger y aplicar información.
- **Fase 3: Análisis y especificación de requisitos:** Se definen y especifican todos los requisitos funcionales, no funcionales y de información de la aplicación junto con el arquitecto. Aquí también se incluirían las descripciones de actores y casos de uso y los diagramas de modelo de caso de uso y diagramas de actividad del sistema.
- **Fase 4: Diseño.** Para cada cuestión sobre el diseño de la aplicación se barajan distintas propuestas, se debaten sus ventajas e inconvenientes y se decide cuál se utilizará para el desarrollo de la aplicación.
- **Fase 5: Implementación.** Codificación de la aplicación. Se escribe el código siguiendo el diseño especificado, se resuelven errores detectados y se depura el código.
- **Fase 6: Pruebas.** Comprobaciones que el código funciona correctamente y no ocurren errores inesperados. Se buscan errores con el código terminado y completamente funcional en caso de que haya un problema más general o haya algún error de código de último minuto que no se ha detectado anteriormente.
- **Fase 7: Memoria.** Se escribe la documentación del proyecto, donde se explica el estado del arte, las decisiones de diseño, la implementación, entre otros.

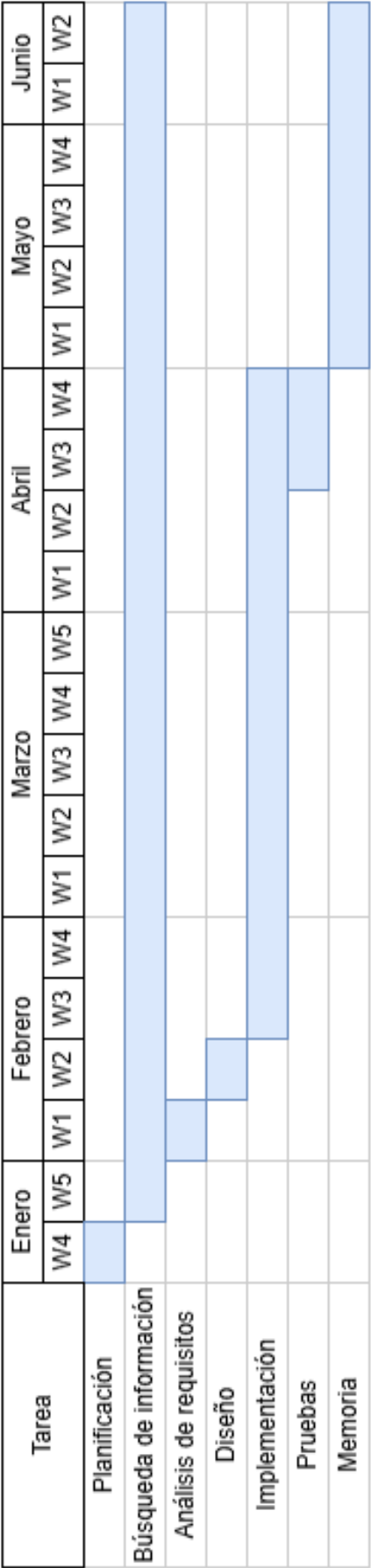


Figura 3.2: Diagrama de Gantt.

3.2. Costes

Antes de exponer los costes, hay que tener en cuenta que esto sería una estimación aproximada del presupuesto necesario para realizarla. Esto aplica tanto para el presupuesto de recursos humanos como para el material. Todos los datos aportados aquí se tomaron el 13 de mayo de 2025, importante mencionarlo puesto que en futuras lecturas probablemente habrá que tener en cuenta la inflación.

El ordenador utilizado es un HP ENVY Laptop 15-ep0xxx con procesador Intel(R) Core(TM) i7-10750H CPU con 16,0 GB de RAM. fue comprado hace 5 años con un precio de alrededor de 1200€.

Todas las herramientas software utilizadas en este proyecto como *Visual Studio Code*, *Github* o *Latex Overleaf* son gratuitas o tienen versiones gratuitas, por lo que no las tendremos en cuenta para el cómputo final. Tampoco tendremos en cuenta las versiones de pago de algunos de los programas o aplicaciones usados que tengan versiones gratis.

El sueldo bruto medio mensual de un desarrollador de software es de alrededor de 2625€. Dado que la realización del proyecto duró alrededor de 5 meses, eso daría un total de $2625 \times 5 = 13125$ € aproximadamente.

En conclusión, el coste final estimado sería de alrededor de $1200 + 13125 = 14325$ €.

Capítulo 4

Estado del arte

En este capítulo se pretende dar a conocer programas de composición de matrices de Harris ya existentes, qué diferencias existen entre estos y sus principales fortalezas y debilidades.

4.1. Trabajos previos

Analizaremos algunas aplicaciones que ya existen para crear matrices de Harris. Cabe mencionar que ha sido difícil encontrar programas específicos de composición de matrices de Harris, y los ya existentes tienen una antigüedad considerable:

- *ArchEd*: 1998
- *Stratify*: 2003
- *Harris Matrix Composer*: 2008

Todos los programas mencionados arriba son aplicaciones de escritorio. Solo he podido probar *Harris Matrix Composer*, el resto eran demasiado antiguos para poder ser instalados.

4.1.1. ArchEd

ArchEd es el más antiguo de todos los programas. Cabe mencionar que no hay mucha información sobre él, lo que es entendible dada la antigüedad de este.

ArchEd permite la edición de matrices de Harris mediante dos métodos distintos:

- Completamente manual, usando la barra de menú.
- Manual combinado con el área de diseño.

Usaremos la adición de unidades estratigráficas para poner un ejemplo:

- **Completamente manual:** La manera más simple, solo hay que pulsar a un botón en la barra de herramientas y se añade la unidad.
- **Manual+Área de diseño:** Seleccionas en la barra del menú *Layout*, seleccionas *STRATUM* y finalmente seleccionas *ADD*. Esta opción permite introducir unidades en la posición indicada por el usuario. Solo tiene que colocar la flecha del ratón sobre la posición en el área y hacer click. Se creará una unidad en la posición. El usuario puede salir de este estado haciendo click sobre la barra de herramientas o presionando el botón ESC del ordenador.

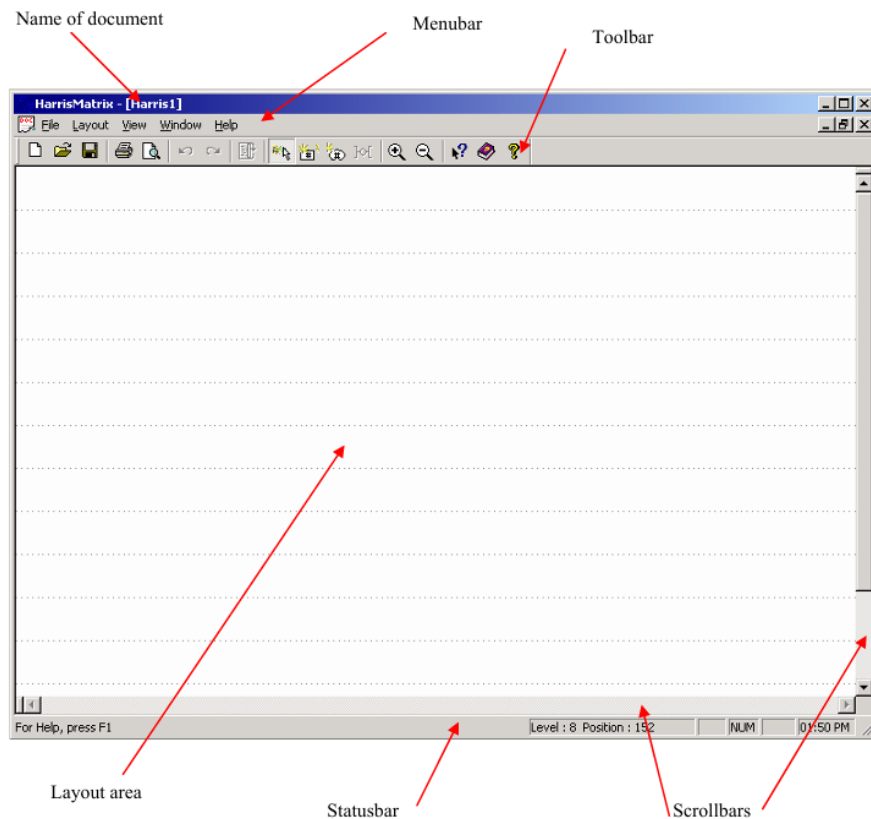


Figura 4.1: Interfaz de *ArchEd*. Tomada de [Hun+98].

Uno de los puntos más fuertes de *ArchEd* es permitir al usuario elegir la forma en la que quiere interactuar con el programa. Esto se extiende a la adición de relaciones, eliminación de unidades y relaciones, entre otras. También cuenta con otras funciones, como cambiar el color de las unidades y las relaciones, elegir las fuentes que se usarán en el diagrama o poder hacer

zoom in o zoom out en el área de diseño [Hun+98].

La figura 4.1 muestra la interfaz de *ArchEd* y señala sus elementos significativos. La figura 4.2 incluye una imagen de una matriz diseñada con *ArchEd*.

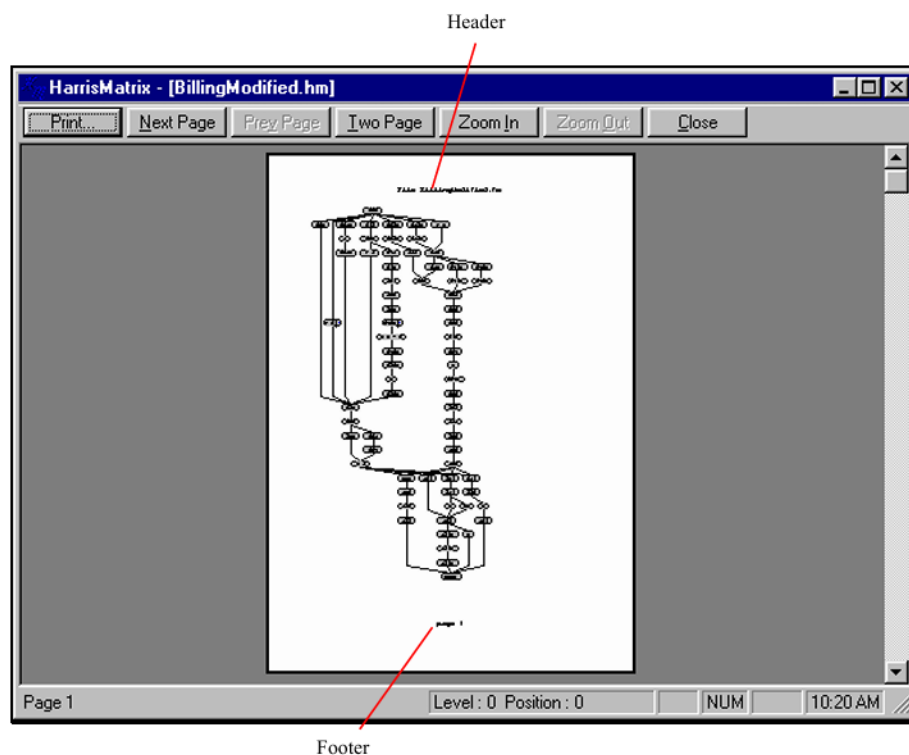


Figura 4.2: Matriz de Harris en *ArchEd*. Tomada de [Hun+98].

4.1.2. Stratify

La versión 1.0, la versión base, de *Stratify* salió el 27 de julio de 2003, mientras que la versión 1.5 fue la última en salir el 23 de agosto de 2008.

Usaremos la adición de unidades para dar un ejemplo de cómo funciona *Stratify*. Simplemente se selecciona *Units/New* y se rellenan los datos que se indican. También hay otras maneras de añadir una nueva unidad, como copiar una ya existente o dividir una unidad ya existente. Si se necesita crear una cadena grande de unidades unidas por relaciones, *Stratify* cuenta con una opción en *Relations/Chain* para conectar varias unidades seguidas ¹².

¹http://www.stratify.org/Download/Stratify_Manual.pdf

²http://www.stratify.org/Download/Stratify_tutorial_v1_5.pdf

Stratify dispone de funcionalidades adicionales, algunas de las cuales son:

- Imprimir informes (cortos o largos, tablas HTML) y el diagrama de Harris.
- Importar y exportar ficheros en diferentes formatos (CSV, DBase, Paradox)
- Enlazar fotos e imágenes a la base de datos de las unidades.

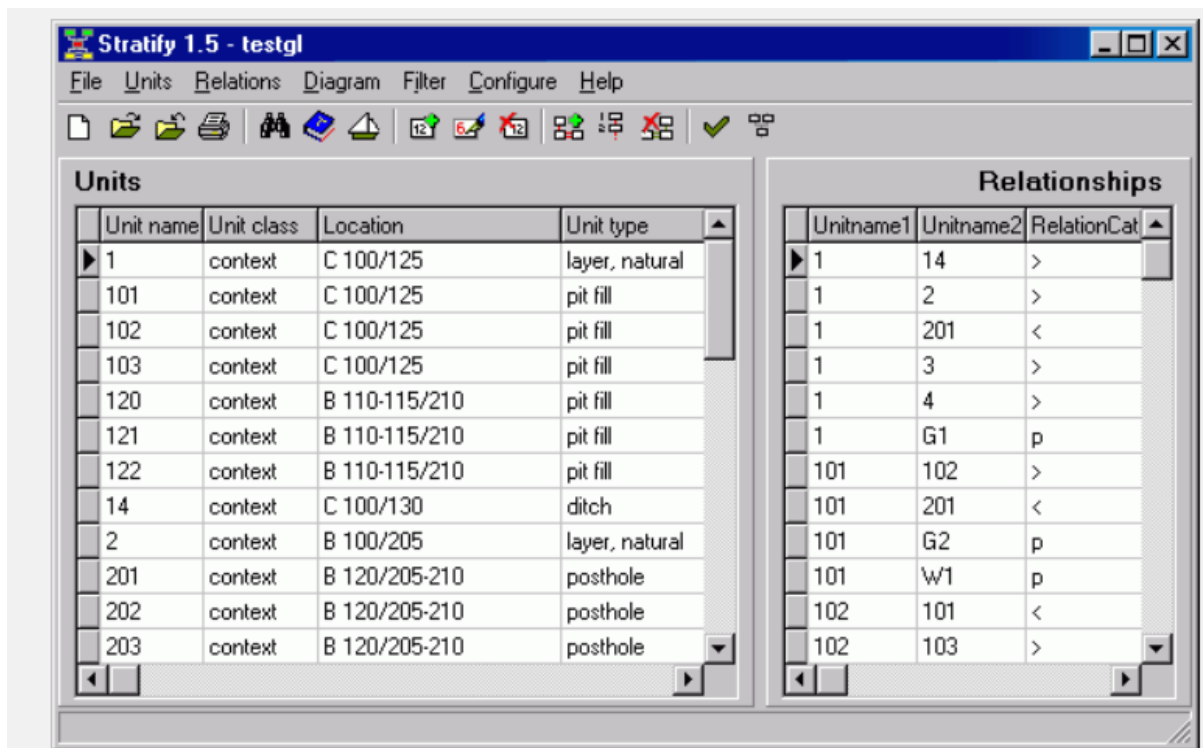


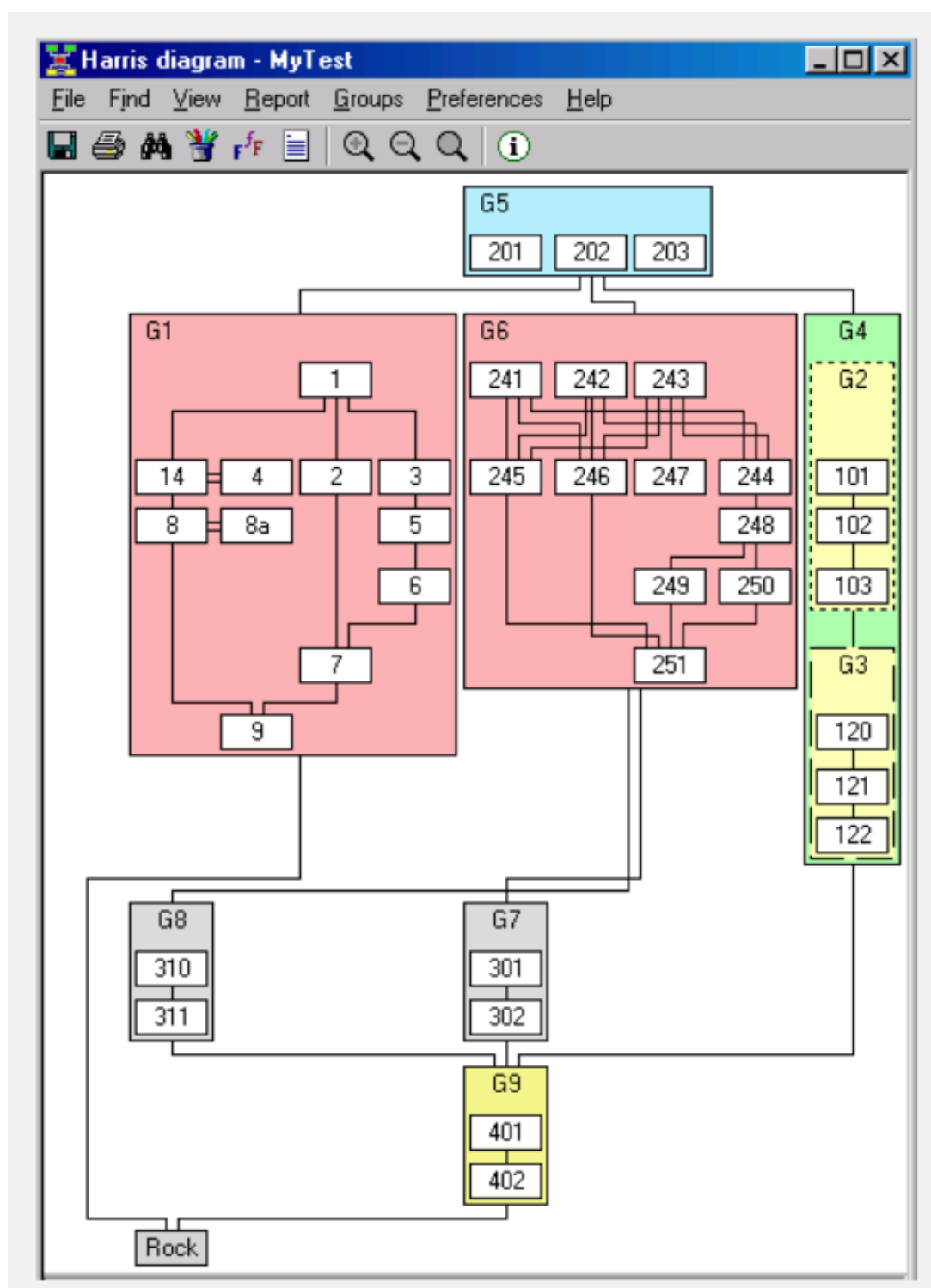
Figura 4.3: Interfaz de *Stratify*.

Las figuras 4.3 y 4.4 contienen capturas tomadas desde este enlace ³ en la sección de *Screenshots*.

En la figura 4.3 aparece la lista de unidades estratigráficas a la izquierda y la lista de relaciones a la derecha.

En la figura 4.4 se expone una matriz de Harris diseñada con *Stratify* donde las agrupaciones se representan como cajas o cuadrados de colores.

³<http://www.stratify.org/index.htm>

Figura 4.4: Matriz de Harris en *Stratify*.

4.1.3. Harris Matrix Composer (HMC)

Desde 2008 existe *Harris Matrix Composer*, o HMC, que permite la creación de grafos de matrices de Harris. Se utilizó Java para su codificación.

Cuenta con una ventana de edición directa que permite la inserción de nodos y unirlos manualmente, haciéndolo más intuitivo y menos tedioso que escoger en un menú. Cuando entras ya hay un pequeño grafo con los nodos **T**, **Unexcavated** y **G** que representan la superficie, la estratificación arqueológica no excavada y la base geológica respectivamente. Si hay una relación incorrecta el nodo que produce el error será de color rojo. Cuenta también con la posibilidad de agrupar nodos en una ventana que se puede abrir para mostrarlo o cerrarla para mayor legibilidad

La versión gratuita de este programa permite grafos de hasta 50 unidades estratigráficas. La versión de pago amplía este límite [TNI08].

Esta fue la única aplicación que pude probar personalmente. Se puede ver en la figura 4.5 una captura con la interfaz de *Harris Matrix Composer* al iniciar el programa.

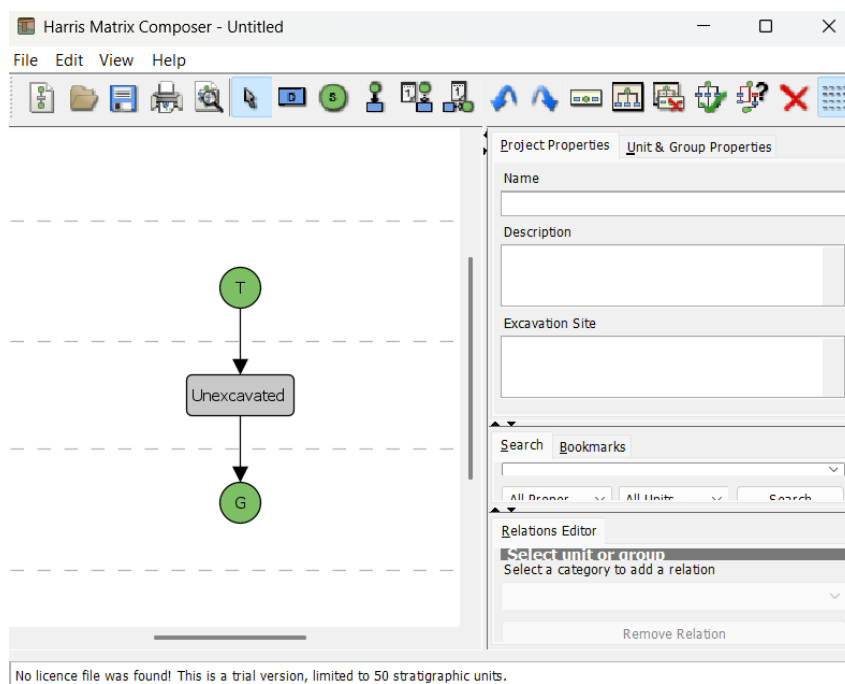


Figura 4.5: Interfaz de *Harris Matrix Composer* con la matriz por defecto. Tomada desde mi ordenador.

Capítulo 5

Especificación de requisitos

En este capítulo se describirán los distintos requisitos funcionales y no funcionales del sistema. Para mayor facilidad, se usará un código con letras y números para identificar a cada requisito:

- **RF:** Requisito funcional
- **RNF** Requisito no funcional
- **RI:** Requisito de información

Cada tipo de requisito se explicará más en detalle en su sección correspondiente.

Para identificar cada requisito dentro de su tipo, utilizaremos un número para identificarlo. Por ejemplo, **RNF-2** se refiere al requisito no funcional número 2.

Por último, reiterar que se contó con la ayuda de un arqueólogo a la hora de establecer los requisitos.

5.1. Requisitos funcionales

Los **requisitos funcionales** describen la interacción sistema-entorno, indicando los servicios que ofrece el sistema e indicando su respuesta a cada posible estímulo que reciba. A continuación se enumeran los distintos requisitos funcionales que se extrajeron del análisis.

- **RF-1. Crear un grafo nuevo.** El sistema permite crear un grafo nuevo desde cero.
- **RF-2. Gestión de carga de matrices.** Se permitirá cargar al sistema una matriz de Harris en formato CSV.

- **RF-2.1. Asegurar extensión correcta.** El sistema se asegurará que solo se puedan cargar archivos en las extensiones permitidas (.csv).
- **RF-2.2. Asegurar formato correcto.** El sistema se asegurará que los valores del archivo cargado están en el lugar correcto y dentro de los valores permitidos.
- **RF-2.3. Gestión de errores al cargar.** El sistema mandará un mensaje de error si hay algún error dentro del archivo cargado y una sugerencia de cómo solucionarlo.
- **RF-3. Gestión de guardado de matrices.** El sistema permitirá guardar en el sistema la matriz con la que se está trabajando.
 - **RF-3.1. Guardar el archivo.** Se permite guardar en el archivo original los cambios realizados.
 - **RF-3.2. Guardar como un nuevo archivo.** Se permite guardar en un nuevo archivo los cambios realizados al original.
- **RF-4. Deshacer.** Se podrán deshacer los cambios realizados al archivo que contiene la información del grafo.
- **RF-5. Rehacer.** Se podrán rehacer los cambios deshechos con la opción Deshacer/Undo.
- **RF-6. Gestión de visualización.** El sistema permitirá elegir entre visualizar el grafo de nodos o la representación matricial.
 - **RF-6.1. Mover la figura.** El sistema permitirá mover la figura interactivamente.
 - **RF-6.2. Zoom.** El sistema permitirá hacer zoom tanto como para acercar un parte de la figura como para alejarla.
 - **RF-6.3. Opción de deshacer.** El sistema tendrá la opción de deshacer la última modificación a la vista y volver a la inmediatamente anterior.
 - **RF-6.4. Opción de rehacer.** El sistema permitirá volver a una vista deshecha perviamente.
 - **RF-6.5. Volver a la vista original.** Se permitirá volver a la vista inicial después de realizar varios cambios.
 - **RF-6.6. Guardar imagen de la figura.** Se podrá guardar una imagen de la vista actual.
 - **RF-6.7. Mostrar redundancias.** Se dará la opción entre mostrar el grafo o la matriz con redundancias o sin redundancias, es decir, si la lista de relaciones es $[A \rightarrow B, B \rightarrow C, A \rightarrow C]$, se pueda elegir entre representar la relación $A \rightarrow C$ o dejar que sea sobreescrita por $[A \rightarrow B, B \rightarrow C]$.
 - **RF-6.8. Mostrar leyenda.** Se podrá elegir si mostrar una leyenda que explique qué representa cada elemento en el grafo o la matriz.
- **RF-7. Tabla de nodos.** Lista de todos los nodos del grafo.
 - **RF-7.1. Visualizar la tabla.** Elegir entre mostrar la tabla o no.

- **RF-7.2. Añadir nodo.** No permitirá adiciones incorrectas, es decir, que el grafo que generen no sea correcto.
- **RF-7.3. Eliminar nodo.**
- **RF-7.4. Editar nodo.** No permitirá ediciones incorrectas que hagan que la aplicación no funcione correctamente.
- **RF-7.5. Redibujar figuras.** Los cambios que se hagan afectarán a la vista del grafo y la matriz.
- **RF-8. Tabla de relaciones.** Lista de todas las relaciones (padre-hijo) entre dos nodos.
 - **RF-8.1. Visualizar la tabla.** Elegir entre mostrar la tabla o no.
 - **RF-8.2. Añadir relación.** No permitirá adiciones incorrectas, es decir, que el grafo que generen no sea correcto.
 - **RF-8.3. Eliminar relación.**
 - **RF-8.4. Redibujar figuras.** Los cambios que se hagan afectarán a la vista del grafo y la matriz.
- **RF-9. Tabla de equivalencias.** Lista de todas las equivalencias entre dos nodos.
 - **RF-9.1. Visualizar la tabla.** Elegir entre mostrar la tabla o no.
 - **RF-9.2. Añadir equivalencia.** No permitirá adiciones incorrectas, es decir, que el grafo que generen no sea correcto.
 - **RF-9.3. Eliminar equivalencia.**
 - **RF-9.4. Redibujar figuras.** Los cambios que se hagan afectarán a la vista del grafo y la matriz.
- **RF-10. Tabla de hechos.** Lista de todos los nodos de hecho con los nodos que engloban.
 - **RF-10.1. Visualizar la tabla.** Elegir entre mostrar la tabla o no.
 - **RF-10.2. Añadir hecho.** No permitirá adiciones incorrectas, es decir, que el grafo que generen no sea correcto.
 - **RF-10.3. Eliminar hecho.**
 - **RF-10.4. Aplicar hecho.** Se podrá decidir qué hechos se aplican y cuáles no.
 - **RF-10.5. Redibujar figuras.** Los cambios que se hagan afectarán a la vista del grafo y la matriz.
- **RF-11. Edición estética.** Se podrán cambiar la forma en la que se muestran los nodos y las aristas del grafo.
 - **RF-11.1. Relaciones.** Se podrá cambiar el color, la forma de la línea, la forma de la flecha y la anchura de las aristas del grafo que representan las relaciones padre-hijo.
 - **RF-11.2. Equivalencias.** Se podrá cambiar el color, la forma de la línea, la forma de la flecha y la anchura de las aristas del grafo que representan las equivalencias.

- **RF-11.3. Fases.** Se podrá cambiar el color de una fase, cambiando el color de todos los nodos que contengan internamente dicha fase.
- **RF-12. Filtrado.** Se permitirá hacer un filtrado de los nodos para que se muestren solo aquellos que encajen con el filtro introducido. En consecuencia se dibujará el grafo y la matriz solo con los nodos que pasen el filtro y los que estén en relación/equivalencia/hecho con estos.
 - **RF-12.1. Introducción del filtro.** Se pueden introducir varias listas de caracteres como filtro.
 - **RF-12.2. Elegir columnas.** Se pueden elegir sobre qué columnas del archivo aplicar el filtrado. Si no se escoge ninguna, se aplican todas.
 - **RF-12.3. Guardar filtrado.** Se puede guardar un archivo con el filtrado utilizado.
 - **RF-12.4. Zoom.** Se puede usar el filtrado para que en vez de filtrar, se haga zoom sobre los nodos que encajen en el filtrado.
 - **RF-12.5 Opciones avanzadas.** El usuario podrá elegir si hacer coincidir o no mayúsculas y minúsculas, palabras completas o diacríticos.
- **RF-13. Buscar errores.** El sistema se encarga de que tanto al subir un archivo como al cambiar su información interna no haya errores que la vuelva incorrecta o contradictoria, de manera que el grafo y la matriz se dibujen correctamente.
- **RF-14. Salir de la aplicación.** El sistema permite al usuario salir de la aplicación.

5.2. Requisitos no funcionales

Los **requisitos no funcionales** describen las propiedades del propio sistema (rendimiento, fiabilidad, ...). Desde el análisis se extrajeron los siguientes requisitos:

- **RNF-1. Sencillez.** El sistema no debe tener funcionalidades que se consideren innecesarias o que hagan más complejo su uso.
- **RNF-2. Intuitivo.** El sistema debe ser lo suficientemente comprensible como para que una persona que utilice poco el ordenador lo pueda usar sin sentirse abrumada.

5.3. Requisitos de información

Los **requisitos de información** son aquellos que se refieren a las necesidades de almacenamiento de la información en el sistema. Para cada requisito se explicará su contenido (la información que contiene) y los requisitos que tiene asociados.

- **RI-1. Archivo.** Datos sobre el archivo subido por el usuario al programa.

- CONTENIDO: (En referencia a los nodos) nombre, hijos (relaciones), equivalencias, nodos que engloban (si son de tipo Hecho), tipo, fase, descripción, otras columnas que se deseen añadir.
 - REQUISITOS ASOCIADOS: RF-2, RF-4, RF-5, RF-7, RF-8, RF-9, RF-10, RF-13.
- **RI-2. Nodos.** Información sobre los hechos de la matriz.
- CONTENIDO: (En referencia a los nodos) código del nodo, tipo del nodo (positivo, negativo o hecho), fase del nodo, descripción del nodo, otros datos del nodo que el usuario haya podido introducir.
 - REQUISITOS ASOCIADOS: RF-10, RF-13.
- **RI-3. Relaciones.** Información sobre las relaciones de la matriz.
- CONTENIDO: Código del nodo padre, código del nodo hijo.
 - REQUISITOS ASOCIADOS: RF-7, RF-13.
- **RI-4. Equivalencias.** Información sobre las equivalencias de la matriz.
- CONTENIDO: Código del nodo A, código del nodo B.
 - REQUISITOS ASOCIADOS: RF-8, RF-13.
- **RI-5. Hechos.** Información sobre los hechos de la matriz.
- CONTENIDO: Código del nodo tipo hecho, código del nodo englobado por el nodo tipo hecho.
 - REQUISITOS ASOCIADOS: RF-9, RF-13.
- **RI-6. Edición estética.** Información sobre la edición estética del grafo.
- CONTENIDO: Color, forma de la línea, forma de la flecha y anchura de las relaciones en el grafo, color, forma de la línea, forma de la flecha y anchura de las equivalencias en el grafo, color de las fases en el grafo.
 - REQUISITOS ASOCIADOS: RF-6, RF-11, RNF-2.
- **RI-7. Filtrado.** Contiene información necesaria para almacenar y utilizar la función de filtrado.
- CONTENIDO: Filtro introducido por el usuario, las columnas sobre las que se aplica el filtrado.
 - REQUISITOS RELACIONADOS: RF-12, RNF-2.

5.4. Casos de uso

Esta sección se divide en 4 subsecciones:

- Descripción de los actores
- Descripción de casos de uso
- Diagramas de modelo de casos de uso
- Diagramas de actividad

El propósito de cada una se describe con más detalle en su respectiva subsección.

5.4.1. Descripción de los actores

Un **actor** desempeña un rol dentro de la interacción con el sistema siendo distinto a este último. Un actor está asociado a todos los casos de uso con los que interacciona [BS03]. Las tablas 5.1 y 5.2 presentan la descripción y los atributos del actor respectivamente.

Actor	Usuario	ACT_1
<i>Descripción</i>	Arqueólogo que utiliza la aplicación para organizar las unidades estratigráficas en un grafo.	
<i>Características</i>	Puede ser cualquier arqueólogo que esté trabajando en una excavación. Puede o no tener experiencia previa usando el programa.	
<i>Relaciones</i>	Se relaciona con el sistema.	

Tabla 5.1: Descripción del actor

Atributos		
Nombre	Descripción	Tipo
Archivo de la matriz	Archivo que contiene los datos de todos los nodos de la matriz (nombre, nodos hijos, equivalencias, tipo, etc...)	Archivo CSV

Tabla 5.2: Atributos del actor

5.4.2. Descripción de Caso de Uso

En esta subsección se describen los **casos de uso** del sistema. La definición de caso de uso que usaremos en el contexto de esta memoria se ha extraído de [BS03]: “*Los casos de uso,*

explicado de manera simple, permiten describir una secuencia de eventos que en conjunto hacen que el sistema haga algo útil.” Las tablas que van desde 5.3 a 5.40 incluyen la descripción de los casos de uso.

Caso de uso	Nueva plantilla	CU_00
Referencias	RF-1	
Precondición	Ninguna	
Postcondición	Toda la información de la matriz cargada en el programa se borra y se carga de nuevo la información de la matriz por defecto.	
Propósito	Permitir al usuario crear una matriz nueva desde cero.	
Resumen	El usuario informa al sistema que quiere crear una nueva matriz, el sistema le pide que lo vuelva a confirmar y, en caso positivo, borra toda la información de la matriz actual y carga la de la matriz por defecto.	

Curso Normal (básico) de eventos			
Actor		Sistema	
1	El usuario selecciona la opción de crear una matriz nueva	2	Informa que está a punto de crear una matriz nueva y borrar toda la información de la matriz actual, preguntando si desea continuar.
3	El usuario decide continuar con la creación de una matriz nueva	4	Borra todos los datos de la matriz actual y carga los datos de la matriz por defecto.

Curso Alterno (secundario) de eventos	
3.a	El usuario cancela la creación de una matriz nueva, por lo que el sistema no realiza la operación y finaliza el caso de uso.

Otros datos			
Frecuencia	600 veces al año	Rendimiento	No más de 10 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminada	Estabilidad	Alta

Tabla 5.3: Caso de uso Nueva plantilla

Caso de uso	Cargar matriz	CU_01
Referencias	RF-2	
Precondición	Tener un archivo con la información de una matriz.	
Postcondición	La información del archivo se carga y procesa en el sistema.	
Propósito	Cargar la información de un archivo dentro del programa.	
Resumen	Se selecciona un archivo dentro del programa y, si no hay errores en el archivo, la información se carga sustituyendo a la que había anteriormente.	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona la opción de cargar un archivo al programa.	2	El sistema pregunta cuál es el archivo que se va a cargar.
3	El usuario selecciona el archivo.	4	Se buscan posibles errores dentro de la extensión o el formato del archivo.
		5	Si no hay errores, la información se carga dentro del sistema.

Curso Alterno (secundario) de eventos	
3a	El usuario cancela la operación y finaliza el caso de uso.
5a	Si se detecta algún error, se manda un mensaje de error distinto dependiendo del tipo, y la información del archivo no se carga.
5b	Si se detectan unidades que no se dibujan, el sistema manda una mensaje advirtiendo de los nodos que no se dibujan y carga la matriz.

Otros datos			
Frecuencia	750 veces al año	Rendimiento	No más de 2 minutos.
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.4: Caso de uso Cargar matriz.

Caso de uso	Guardar	CU_02
Referencias	RF-3.1	
Precondición	Ya existe un archivo con la información de la matriz que se va a guardar.	
Postcondición	La información nueva se sube al archivo original.	
Propósito	Guardar la información editada.	
Resumen	El usuario informa al sistema que quiere guardar los cambios realizados sobre la matriz y el sistema los guarda sobre el archivo que guarda la matriz.	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona guardar la información de la matriz.	2	El sistema guarda la información en el archivo y el sistema lo indica mostrando un mensaje por pantalla.

Curso Alterno (secundario) de eventos	
2a	No existe un archivo previo, por lo que no hay un archivo sobre el que guardar. El sistema devuelve un mensaje de error y finaliza el caso de uso.

Otros datos			
Frecuencia	750 veces al año	Rendimiento	No más de 10 segundos.
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.5: Caso de uso Guardar.

Caso de uso	Guardar como	CU_03
Referencias	RF-3.2	
Precondición	Ninguna.	
Postcondición	Se crea un nuevo archivo con la información de la matriz en el programa.	
Propósito	Crear un nuevo archivo que contenga la información de la matriz.	
Resumen	El usuario informa al sistema que quiere crear un archivo nuevo con los datos de la matriz cargada en el sistema, el sistema le pide nombre y dirección y cuando el usuario los introduce, crea el archivo nuevo e informa al usuario.	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario solicita guardar la información en un archivo nuevo.	2	El sistema le pide el nombre y la dirección del nuevo archivo.
3	El usuario introduce la dirección y el nombre del archivo nuevo.	4	El sistema guarda el archivo nuevo y finaliza el caso de uso.

Curso Alterno (secundario) de eventos	
3a	El usuario introduce un nombre y un archivo en esa dirección ya lo tiene. El sistema se lo indica al usuario y le pregunta si quiere reemplazar el archivo.

Otros datos			
Frecuencia	650 veces al año	Rendimiento	No más de 10 segundos.
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.6: Caso de uso Guardar como.

Caso de uso	Deshacer	CU_04
Referencias	RF-4	CU_05
Precondición	Ya se ha realizado una acción que ha alterado la información interna de la matriz.	
Postcondición	La información vuelve a su estado previo y se habilita la opción de rehacer la acción deshecha.	
Propósito	Deshacer acciones que hayan alterado la información de la matriz.	
Resumen	El usuario indica al sistema que quiere deshacer el último cambio realizado a la información de la matriz y el sistema carga el estado anterior de la matriz.	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona la opción de deshacer la acción.	2	El programa carga el estado anterior de la información (que había sido guardado anteriormente) y añade la acción a una pila de acciones deshechas que se podrán rehacer con la opción de rehacer acción. Habilita la opción de rehacer acción si no estaba ya habilitada.

Otros datos			
Frecuencia	2500 veces al año	Rendimiento	No más de 30 segundos.
Importancia	Alta	Urgencia	Alta
Estado	Terminada	Estabilidad	Alta

Tabla 5.7: Caso de uso Deshacer.

Caso de uso	Rehacer	CU_05
Referencias	RF-5	CU_04
Precondición	Debe haberse deshecho una acción previamente.	
Postcondición	Se rehace la acción deshecha y la matriz vuelve a tener la información anterior a deshacer la acción.	
Propósito	Rehacer acciones previamente deshechas.	
Resumen	El usuario informa al sistema que quiere rehacer una acción que había deshecho previamente y el sistema carga la información de la matriz anterior a la acción de deshacer.	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona la opción de rehacer una acción.	2	El sistema guarda el estado actual de la información de la matriz en la pila de acciones deshechas y carga el estado de la información de la matriz previo a haber deshecho la acción.

Otros datos			
Frecuencia	2500 veces al año	Rendimiento	No más de 10 segundos.
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.8: Caso de uso Rehacer.

Caso de uso	Redundancia	CU_06
Referencias	RF-6.7	
Precondición	Ninguna.	
Postcondición	Se aplica redundancia si no se está aplicando o viceversa.	
Propósito	Activar o desactivar la aplicación de redundancia.	
Resumen	El usuario indica al sistema dibujar las figuras con relaciones redundantes si no las estaba dibujando antes, y viceversa.	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario informa al sistema que quiere cambiar el estado de la redundancia.	2	El sistema aplica la redundancia si no estaba aplicada o viceversa.

Otros datos			
Frecuencia	1000 veces al año	Rendimiento	No más de 10 segundos.
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.9: Caso de uso Redundancia.

Caso de uso	Mover la figura	CU_07
Referencias	RF-6.1	
Precondición	Existe una figura dibujada	
Postcondición	La ventana que muestra la figura se mueve y cambia la vista	
Propósito	Permitir moverse por la figura dibujada	
Resumen	El usuario selecciona un punto dentro de la ventana donde se muestra la figura y la arrastra hacia una dirección. El sistema recalcula la vista en consecuencia	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona en la barra de herramientas la opción de mover la ventana	2	El sistema habilita poder mover la ventana de la figura
3	El usuario hace clic en un punto de la ventana y arrastra la flecha del ratón hacia otra posición	4	El sistema calcula la nueva vista y finaliza el caso de uso

Curso Alternativo (secundario) de eventos	
3a	El usuario cancela la operación de mover la vista de la figura y finaliza el caso de uso

Otros datos			
Frecuencia	5000 veces al año	Rendimiento	Menos de 10 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.10: Caso de uso Mover la figura.

Caso de uso	Zoom	CU_08
Referencias	RF-6.2	
Precondición	Hay una figura dibujada.	
Postcondición	La vista de la figura cambia.	
Propósito	Hacer zoom in o zoom out sobre un área de la figura.	
Resumen	El usuario indica al sistema que quiere hacer zoom sobre un área de la ventana e indica la superficie y forma del área y si es zoom in o zoom out. El sistema calcula la nueva vista de la figura.	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario solicita al sistema poder utilizar el zoom sobre la vista.	2	El sistema habilita el zoom sobre la vista.
3	El usuario delimita un área sobre la que aplicar el zoom in o zoom out (dependiendo de si usa el botón izquierdo del ratón o el derecho, respectivamente).	4	El sistema calculará la nueva vista.

Curso Alternativo (secundario) de eventos	
3a	El usuario pide al sistema que deshabilite el zoom y finaliza el caso de uso.

Otros datos			
Frecuencia	7500 veces al año	Rendimiento	Menos de 10 segundos.
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.11: Caso de uso Zoom.

Caso de uso	Deshacer vista	CU_09
Referencias	RF-6.3	CU_10
Precondición	Hay una figura dibujada y se ha modificado la vista previamente.	
Postcondición	La vista actual cambia a una vista anterior disponible.	
Propósito	Permitir al usuario retroceder a una vista anterior de la figura.	
Resumen	El usuario indica al sistema que desea regresar a una vista anterior, y el sistema restaura esa vista y habilita la posibilidad de rehacer vistas.	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario solicita volver a una vista anterior.	2	El sistema cambia la vista actual a una vista anterior disponible, habilita la opción de rehacer y finaliza el caso de uso.

Curso Alterno (secundario) de eventos	
1.a	No hay vistas anteriores disponibles, por lo que el sistema no realiza ningún cambio y muestra un mensaje informativo al usuario.

Otros datos			
Frecuencia	7500 veces al año	Rendimiento	Menos de 10 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Stabilidad	Alta

Tabla 5.12: Caso de uso Deshacer vista

Caso de uso	Rehacer vista	CU_10
Referencias	RF-6.4	CU_09
Precondición	Hay una figura dibujada y se ha utilizado previamente la opción de deshacer una vista.	
Postcondición	La vista de la figura cambia a la que fue deshecha anteriormente.	
Propósito	Permitir al usuario rehacer una vista deshecha.	
Resumen	El usuario solicita rehacer una vista, y el sistema restaura la vista que había sido deshecha previamente.	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario solicita rehacer una vista deshecha.	2	El sistema carga la vista que existía antes de aplicar “Deshacer vista” y finaliza el caso de uso.

Curso Alterno (secundario) de eventos	
1.a	No hay una vista previa que rehacer, por lo que el sistema informa al usuario y no realiza ninguna acción.

Otros datos			
Frecuencia	7500 veces al año	Rendimiento	Menos de 10 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Stabilidad	Alta

Tabla 5.13: Caso de uso Rehacer vista

Caso de uso	Vista original	CU_11
Referencias	RF-6.5	
Precondición	Ninguna.	
Postcondición	La vista del sistema cambia a la original.	
Propósito	Restaurar la vista original de la figura, tal como estaba antes de aplicar cualquier cambio.	
Resumen	El usuario solicita volver a la vista original y el sistema carga dicha vista, eliminando los cambios visuales realizados previamente.	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario solicita volver a la vista original.	2	El sistema restaura la vista original de la figura y finaliza el caso de uso.

Curso Alterno (secundario) de eventos	
1.a	El usuario decide cancelar la restauración de la vista original, por lo que el sistema no realiza la acción y finaliza el caso de uso.

Otros datos			
Frecuencia	3000 veces al año	Rendimiento	Menos de 10 segundos
Importancia	Media-alta	Urgencia	Media-alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.14: Caso de uso Vista original.

Caso de uso	Guardar figura	CU_12
Referencias	RF-6.6	
Precondición	Hay una figura dibujada.	
Postcondición	Una imagen de la figura es guardada como un archivo.	
Propósito	Permitir al usuario guardar una imagen de la figura en su ordenador.	
Resumen	El usuario solicita guardar una imagen de la figura. El sistema solicita ancho, alto, formato, nombre y dirección de guardado. El usuario proporciona los datos y el sistema guarda el archivo.	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario solicita guardar una imagen de la figura.	2	El sistema solicita ingresar el ancho en píxeles (rango de 100 a 5000).
3	El usuario ingresa el ancho.	4	El sistema solicita ingresar el alto en píxeles (rango de 100 a 5000).
5	El usuario ingresa el alto.	6	El sistema solicita el formato, nombre y dirección para guardar el archivo.
7	El usuario indica el formato, nombre y dirección del archivo.	8	El sistema guarda el archivo y finaliza el caso de uso.

Curso Alterno (secundario) de eventos	
3a, 5a, 7a	El usuario cancela la operación y finaliza el caso de uso.
3b	El usuario introduce un ancho fuera del rango permitido. El sistema muestra un mensaje de error y solicita un nuevo valor.
5b	El usuario introduce un alto fuera del rango permitido. El sistema muestra un mensaje de error y solicita un nuevo valor.
7b	El sistema detecta que ya existe un archivo con el mismo nombre y formato. Solicita al usuario confirmar el reemplazo o modificar nombre, formato o dirección.

Otros datos			
Frecuencia	2000 veces al año	Rendimiento	No más de 30 segundos
Importancia	Media	Urgencia	Media
Estado	Terminado	Estabilidad	Alta

Tabla 5.15: Caso de uso Guardar figura.

Caso de uso	Mostrar/ocultar leyenda	CU_13
Referencias	RF-6.8	
Precondición	Hay una figura dibujada	
Postcondición	Muestra la leyenda si no se muestra y viceversa	
Propósito	Permitir mostrar u ocultar la figura	
Resumen	El usuario informa al sistema que quiere (o no) que se muestre la leyenda junto a la figura y el sistema redibuja la figura	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario solicita al sistema mostrar/ocultar la leyenda en la figura	2	El sistema cambia la vista para que se muestre/oculte la leyenda y termina el caso de uso

Otros datos			
Frecuencia	1000 veces al año	Rendimiento	No más de 10 segundos
Importancia	Media	Urgencia	Media
Estado	Terminada	Estabilidad	Alta

Tabla 5.16: Caso de uso Mostrar/ocultar leyenda.

Caso de uso	Añadir nodo	CU_14
Referencias	RF-7.2, RF-13	
Precondición	Hay una matriz cargada en el sistema	
Postcondición	Se añade la información de un nodo nuevo a la información de la matriz cargada	
Propósito	Permitir al usuario añadir nodos nuevos	
Resumen	El usuario solicita al sistema introducir un nodo nuevo, el sistema le pide que introduzca los datos del nodo nuevo y si no hay errores, los introduce en los datos de la matriz cuando el usuario solicita guardarlos	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario indica al sistema que quiere introducir un nodo nuevo	2	El sistema le pide todos los datos que puede tener un nodo en esa matriz, excluyendo los hijos, las equivalencias y los hechos
3	El usuario introduce los datos en el sistema	4	El sistema incluye los datos del nodo nuevo en la información de la matriz

Curso Alterno (secundario) de eventos	
4a	El usuario intenta introducir un nodo con una cadena vacía como nombre. El sistema muestra un mensaje indicando que todos los nodos deben tener un nombre y no permite al usuario añadir el nodo hasta que escriba un nombre
4b	El usuario introduce un nombre que ya tiene otro nodo. El sistema indica que no puede haber duplicados y no permite añadir el nodo.
4c	El usuario informa al sistema que no quiere continuar con la operación y finaliza el caso de uso

Otros datos			
Frecuencia	20000 veces al año	Rendimiento	No más de 10 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.17: Caso de uso Añadir nodo.

Caso de uso	Eliminar nodo	CU_15
Referencias	RF-7.3, RF-7.5, RF-13	
Precondición	Que la matriz cargada tenga al menos un nodo	
Postcondición	El nodo seleccionado es eliminado de la información de la matriz	
Propósito	Permitir al usuario eliminar nodos de la matriz cargada	
Resumen	El usuario informa al sistema del nodo que quiere eliminar, el sistema le pregunta si está seguro y en caso afirmativo lo elimina	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona el nodo que quiere eliminar	2	El sistema pregunta al usuario si está seguro de eliminar el nodo
3	El usuario confirma al sistema que está seguro de eliminar el nodo	4	El sistema elimina el nodo y su información de la matriz cargada y redibuja la figura

Curso Alterno (secundario) de eventos			
3a	El usuario informa al sistema que cancela la operación y finaliza el caso de uso		
Otros datos			
Frecuencia	10000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.18: Caso de uso Eliminar nodo.

Caso de uso	Editar nodo	CU_16
Referencias	RF-7.4, RF-7.5, RF-13	
Precondición	Hay al menos un nodo en la matriz cargada	
Postcondición	La información interna del nodo cambia	
Propósito	Permitir que el usuario cambie información interna del nodo	
Resumen	El usuario selecciona un nodo e informa al programa que quiere cambiar su información interna. El sistema habilita la edición y antes de guardar la información se asegura que el nombre del nodo no es una cadena vacía o comparte nombre con otro nodo. Finalmente, el sistema redibuja las figuras.	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona un nodo y solicita al sistema modificar su información interna	2	El sistema habilita una ventana con la información actual del nodo que está permitida editar
3	El usuario cambia la información del nodo	4	El sistema se asegura que el nombre del nodo no es un duplicado o una cadena vacía
5	El usuario solicita guardar la información cambiada	6	Si no hay problemas con el nombre, la información es añadida

Curso Alternativo (secundario) de eventos	
3a	El usuario informa al sistema que quiere cancelar la operación, el sistema aborta la edición del nodo y finaliza el caso de uso
4a	Si el nombre está duplicado o es una cadena vacía, el sistema le informa al usuario que el nombre del nodo no puede tener ese valor y no permite guardar el nodo

Otros datos			
Frecuencia	20000 veces al año	Rendimiento	No más de 20 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.19: Caso de uso Editar nodo.

Caso de uso	Añadir relación	CU_17
Referencias	RF-8.2, RF-8.4, RF-13	
Precondición	Hay una matriz cargada en el programa y suficientes nodos para crear una relación nueva	
Postcondición	Se añade la relación nueva y se vuelven a dibujar las figuras	
Propósito	Permitir al usuario añadir nuevas relaciones entre nodos	
Resumen	El usuario indica al sistema la nueva relación entre dos nodos que quiere añadir, el sistema se asegura que no haya errores con su introducción, en cuyo caso guarda la relación	

Curso Normal (básico) de eventos			
Actor		Sistema	
1	El usuario introduce una nueva relación entre dos nodos	2	El sistema se asegura que la nueva relación no produzca un error en la información actual de la matriz
		3	Si no hay errores, el sistema añade la nueva relación a la información de la matriz y dibuja de nuevo las figuras con la información modificada

Curso Alternativo (secundario) de eventos	
3a	El sistema detecta que la nueva relación produce un error, indicándolo al usuario, no incluyéndola en la información de la matriz y finalizando el caso de uso

Otros datos			
Frecuencia	Más de 10000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.20: Caso de uso Añadir relación.

Caso de uso	Eliminar relación	CU_18
Referencias	RF-8.3, RF-8.4, RF-13	
Precondición	Hay una matriz cargada en el programa y con al menos una relación	
Postcondición	Se elimina la relación y se vuelven a dibujar las figuras	
Propósito	Permitir al usuario eliminar relaciones entre nodos	
Resumen	El usuario selecciona una relación a eliminar y el sistema la elimina de la información de la matriz	

Curso Normal (básico) de eventos			
Actor		Sistema	
1	El usuario selecciona las relaciones que quiere eliminar	2	El sistema elimina las relaciones del sistema y redibuja las figuras

Otros datos			
Frecuencia	Más de 10000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.21: Caso de uso Eliminar relación.

Caso de uso	Añadir equivalencia	CU_19
Referencias	RF-9.2, RF-9.4, RF-13	
Precondición	Hay una matriz cargada en el programa y suficientes nodos para crear una equivalencia nueva	
Postcondición	Se añade la equivalencia nueva y se vuelven a dibujar las figuras	
Propósito	Permitir al usuario añadir nuevas equivalencias entre nodos	
Resumen	El usuario indica al sistema la nueva equivalencia entre dos nodos que quiere añadir, el sistema se asegura que no haya errores con su introducción, en cuyo caso la guarda	

Curso Normal (básico) de eventos			
Actor		Sistema	
1	El usuario introduce una nueva equivalencia entre dos nodos	2	El sistema se asegura que la nueva equivalencia no produzca un error en la información actual de la matriz
		3	Si no hay errores, el sistema añade la nueva equivalencia a la información de la matriz y dibuja de nuevo las figuras con la información modificada

Curso Alterno (secundario) de eventos			
3a	El sistema detecta que la nueva equivalencia produce un error, indicándolo al usuario, no incluyéndola en la información de la matriz y finalizando el caso de uso		
Otros datos			
Frecuencia	Más de 10000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.22: Caso de uso Añadir equivalencia.

Caso de uso	Eliminar equivalencia	CU_20
Referencias	RF-9.3, RF-9.4, RF-13	
Precondición	Hay una matriz cargada en el programa y con al menos una equivalencia	
Postcondición	Se elimina la equivalencia y se vuelven a dibujar las figuras	
Propósito	Permitir al usuario eliminar equivalencias entre nodos	
Resumen	El usuario indica al sistema que quiere borrar la equivalencia seleccionada y el sistema la elimina de la información de la matriz	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona las equivalencias que quiere eliminar	2	El sistema elimina las equivalencias del sistema y redibuja las figuras
Otros datos			
Frecuencia	Más de 10000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.23: Caso de uso Eliminar equivalencia.

Caso de uso	Añadir hecho	CU_21
Referencias	RF-10.2, RF-10.5, RF-13	
Precondición	Hay una matriz cargada en el programa y hay al menos un nodo de tipo hecho y un nodo de tipo positivo o negativo	
Postcondición	Se añade el hecho nuevo y se vuelven a dibujar las figuras	
Propósito	Permitir al usuario añadir nuevos hechos	
Resumen	El usuario introduce un nodo hecho y un nodo negativo o positivo e indica al sistema que quiere añadirlos como hecho. Si el sistema no detecta errores en su introducción, los añade a la información de la matriz	

Curso Normal (básico) de eventos			
Actor		Sistema	
1	El usuario introduce un nodo hecho y un nodo negativo o positivo	2	El sistema se asegura que el nuevo hecho no produzca un error en la información actual de la matriz
		3	Si no hay errores, el sistema añade el nuevo hecho a la información de la matriz y dibuja de nuevo las figuras con la información modificada

Curso Alternativo (secundario) de eventos	
3a	El sistema detecta que el nuevo hecho produce un error, indicándolo al usuario, no incluyéndolo en la información de la matriz y finalizando el caso de uso

Otros datos			
Frecuencia	Más de 10000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.24: Caso de uso Añadir hecho.

Caso de uso	Eliminar hecho	CU_22
Referencias	RF-10.3, RF-10.5, RF-13	
Precondición	Hay una matriz cargada en el programa y con al menos un hecho	
Postcondición	Se elimina el hecho y se vuelven a dibujar las figuras	
Propósito	Permitir al usuario eliminar hechos	
Resumen	El usuario selecciona un hecho y solicita al sistema eliminarlo. El sistema lo elimina de los datos de la matriz	

Curso Normal (básico) de eventos			
Actor		Sistema	
1	El usuario selecciona los hechos que quiere eliminar	2	El sistema elimina los hechos del sistema y redibuja las figuras

Otros datos			
Frecuencia	Más de 10000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.25: Caso de uso Eliminar hecho.

Caso de uso	Aplicar hecho	CU_23
Referencias	RF-10.4, RF-10.5	
Precondición	Hay una matriz cargada que tiene al menos un hecho	
Postcondición	Se redibuja la figura aplicando los hechos seleccionados	
Propósito	Permitir dibujar las figuras con los hechos seleccionados	
Resumen	El usuario introduce los hechos que quiere que se apliquen a las figuras y el sistema dibuja las figuras teniendo en cuenta los nodos tipo hecho introducidos por el usuario	

Curso Normal (básico) de eventos			
Actor		Sistema	
1	El usuario introduce los nodos de hecho que quiere que se apliquen al dibujo y solicita al sistema aplicar dichos nodos a las figuras	2	El sistema redibuja las figuras con los nodos tipo hecho introducidos por el usuario

Otros datos			
Frecuencia	10000 veces al año	Rendimiento	Alto
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.26: Caso de uso Aplicar hecho.

Caso de uso	Cambiar color de las relaciones	CU_24
Referencias	RF-11.1	
Precondición	Ninguna	
Postcondición	El color de las relaciones en el grafo de nodos cambia	
Propósito	Permitir al usuario cambiar el color de las relaciones	
Resumen	El usuario solicita cambiar el color actual de las relaciones, el sistema le pregunta por el color nuevo y el usuario introduce el color nuevo que el sistema aplicará redibujando el grafo	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario informa al sistema que quiere cambiar el color de las relaciones en el grafo de nodos	2	El sistema le pide al usuario que introduzca el color nuevo para las relaciones
3	El usuario introduce el color nuevo	4	El sistema cambia el color de las relaciones en el grafo de nodos con el color indicado por el usuario

Curso Alternativo (secundario) de eventos	
3a	El usuario cancela la operación y finaliza el caso de uso

Otros datos			
Frecuencia	1000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.27: Caso de uso Cambiar color de las relaciones.

Caso de uso	Cambiar la forma de la línea (relación)	CU_25
Referencias	RF-11.1	
Precondición	Ninguna	
Postcondición	La forma de la línea de la relación cambia en el dibujo del grafo de nodos	
Propósito	Permitir al usuario que decida cómo se dibuja la línea que representa las relaciones	
Resumen	El usuario selecciona un nuevo valor para la forma de las líneas de relación y el sistema redibuja el grafo de nodos para que se muestre con el nuevo formato de línea	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona una forma nueva de representar las líneas de relación del grafo de nodos	2	El sistema redibuja el grafo de nodos

Otros datos			
Frecuencia	1000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.28: Caso de uso Cambiar la forma de la línea (relación).

Caso de uso	Cambiar la forma de la flecha (relación)	CU_26
Referencias	RF-11.1	
Precondición	Ninguna	
Postcondición	La forma de la flecha de la relación cambia en el dibujo del grafo de nodos	
Propósito	Permitir al usuario que decida cómo se dibuja la flecha de las relaciones	
Resumen	El usuario selecciona un nuevo valor para la forma de las flechas de relación y el sistema redibuja el grafo de nodos para que se muestre con el nuevo formato de flecha	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona una forma nueva de representar las flechas de relación del grafo de nodos	2	El sistema redibuja el grafo de nodos

Otros datos			
Frecuencia	1000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.29: Caso de uso Cambiar la forma de la flecha (relación).

Caso de uso	Cambiar la anchura (relación)	CU_27
Referencias	RF-11.1	
Precondición	Ninguna	
Postcondición	La anchura de las líneas de relación cambia	
Propósito	Permitir al usuario cambiar la anchura de las líneas de relación en el grafo de nodos	
Resumen	El usuario selecciona una anchura para la línea de relación y el sistema redibuja el grafo de nodos usando dicha anchura	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona una nueva anchura para las líneas de relación en un rango entre 1 y 8, ambos incluidos	2	El sistema redibuja el grafo usando la anchura introducida por el usuario

Otros datos			
Frecuencia	2000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.30: Caso de uso Cambiar la anchura (relación).

Caso de uso	Cambiar color de las equivalencias	CU_28
Referencias	RF-11.2	
Precondición	Ninguna	
Postcondición	El color de las equivalencias en el grafo de nodos cambia	
Propósito	Permitir al usuario cambiar el color de las equivalencias	
Resumen	El usuario solicita cambiar el color actual de las equivalencias, el sistema le pregunta por el color nuevo y el usuario introduce el color nuevo que el sistema aplicará redibujando el grafo	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario informa al sistema que quiere cambiar el color de las equivalencias en el grafo de nodos	2	El sistema le pide al usuario que introduzca el color nuevo para las equivalencias
3	El usuario introduce el color nuevo	4	El sistema cambia el color de las equivalencias en el grafo de nodos con el color indicado por el usuario

Curso Alternativo (secundario) de eventos	
3a	El usuario cancela la operación y finaliza el caso de uso

Otros datos			
Frecuencia	1000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.31: Caso de uso Cambiar color de las equivalencias.

Caso de uso	Cambiar la forma de la línea (equiv.)	CU_29
Referencias	RF-11.2	
Precondición	Ninguna	
Postcondición	La forma de la línea de la equivalencia cambia en el dibujo del grafo de nodos	
Propósito	Permitir al usuario que decida cómo se dibuja la línea que representa las equivalencias	
Resumen	El usuario selecciona un nuevo valor para la forma de las líneas de equivalencia y el sistema redibuja el grafo de nodos para que se muestre con el nuevo formato de línea	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona una forma nueva de representar las líneas de equivalencia del grafo de nodos	2	El sistema redibuja el grafo de nodos

Otros datos			
Frecuencia	1000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.32: Caso de uso Cambiar la forma de la línea (equivalencia).

Caso de uso	Cambiar la forma de la flecha (equiv.)	CU_30
Referencias	RF-11.2	
Precondición	Ninguna	
Postcondición	La forma de la flecha de la equivalencia cambia en el dibujo del grafo de nodos	
Propósito	Permitir al usuario que decida cómo se dibuja la flecha de las equivalencias	
Resumen	El usuario selecciona un nuevo valor para la forma de las flechas de equivalencia y el sistema redibuja el grafo de nodos para que se muestre con el nuevo formato de flecha	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona una forma nueva de representar las flechas de equivalencia del grafo de nodos	2	El sistema redibuja el grafo de nodos

Otros datos			
Frecuencia	1000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.33: Caso de uso Cambiar la forma de la flecha (equivalencia).

Caso de uso	Cambiar la anchura (equivalencia)	CU_31
Referencias	RF-11.2	
Precondición	Ninguna	
Postcondición	La anchura de las líneas de equivalencia cambia	
Propósito	Permitir al usuario cambiar la anchura de las líneas de equivalencia en el grafo de nodos	
Resumen	El usuario selecciona una anchura de para la línea de equivalencia y el sistema redibuja el grafo de nodos usando dicha anchura	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona una nueva anchura para las líneas de equivalencia en un rango entre 1 y 8 ambos incluidos	2	El sistema redibuja el grafo usando la anchura introducida por el usuario

Otros datos			
Frecuencia	2000 veces al año	Rendimiento	No más de 15 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.34: Caso de uso Cambiar la anchura (equivalencia).

Caso de uso	Cambiar el color (Fase)	CU_32
Referencias	RF-11.3	
Precondición	Que al menos un nodo contenga una fase	
Postcondición	El color de todos los nodos con esa fase cambia	
Propósito	Permitir al usuario cambiar el color de todos los nodos que contengan una fase	
Resumen	El usuario selecciona una fase indicando que quiere cambiar su color, el sistema le muestra una selección de colores de la cual el usuario escoge uno que el sistema aplicará a todos los nodos con dicha fase	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario selecciona una fase indicando al sistema que quiere cambiar su color	2	El sistema muestra una selección de colores que se pueden aplicar
3	El usuario selecciona un color	4	El sistema aplica dicho color a todos los nodos de esa fase

Curso Alterno (secundario) de eventos	
3a	El usuario indica al sistema que cancele la operación y finaliza el caso de uso

Otros datos			
Frecuencia	5000 veces al año	Rendimiento	No más de 30 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.35: Caso de uso Cambiar el color (Fase).

Caso de uso	Filtrar nodos	CU_33
Referencias	RF-12, RF-12.1, RF-12.2	
Precondición	Ninguna	
Postcondición	Solo se muestran los nodos que cumplen el filtro y los que están relacionados con estos (relación, equivalencia o hechos). Solo se usarán estos nodos para dibujar las figuras	
Propósito	Permitir al usuario filtrar los nodos según el filtro introducido de manera que solo aparezcan estos y sus relaciones, equivalencias y hechos y que solo se dibujen estos en las figuras	
Resumen	El usuario introduce una cadena de caracteres a modo de filtro y las columnas donde aplicarlo y el sistema solo muestra y dibuja aquellos nodos que cumplan el filtro y los que estén dentro de la información interna de estos	

Curso Normal (básico) de eventos			
Actor		Sistema	
1	El usuario introduce una cadena de caracteres (como una lista de palabras) y las columnas donde quiere que se aplique el filtro y pide al sistema que aplique filtrado	2	El sistema utiliza la cadena y las columnas seleccionadas, borra la información que se ve en pantalla (no la de la matriz) y solo muestra la información de aquellos nodos que en alguna de las columnas seleccionadas su valor coincida con alguno de los valores en la cadena de caracteres. Estos nodos y los que estén dentro de la información interna de estos con los que usará el sistema para dibujar las figuras

Curso Alternativo (secundario) de eventos	
1a	El usuario no selecciona ninguna columna por lo que el sistema lo interpreta como que se usen todas las columnas

Otros datos			
Frecuencia	Más de 50000 veces al año	Rendimiento	No más de 30 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.36: Caso de uso Filtrar nodos.

Caso de uso	Guardar filtrado	CU_34
Referencias	RF-12.3	
Precondición	Ninguna	
Postcondición	El sistema guarda un archivo con la matriz filtrada	
Propósito	Permitir al usuario generar un nuevo archivo que guarde la información de la matriz con el filtrado	
Resumen	El usuario solicita al sistema guardar un archivo con la matriz filtrada, el sistema solicita dirección, nombre y, una vez introducidos, el sistema guarda el archivo y muestra un mensaje por pantalla indicando que el archivo se ha guardado exitosamente	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario solicita al sistema guardar la información de la matriz tras aplicar el filtro	2	El sistema trata la información de la matriz de manera que solo se guarda la información de los nodos filtrados y los que tienen algún tipo de conexión con estos
4	El usuario introduce la dirección y el nombre del archivo nuevo en el sistema	3	El sistema le pide la dirección y el nombre del archivo al usuario
		5	Si no hay problemas con el nombre o la dirección, el sistema genera el archivo, lo guarda con el nombre y dirección indicadas y le comunica al usuario mediante un mensaje que el archivo se ha guardado exitosamente

Curso Alterno (secundario) de eventos	
4a	El usuario solicita al sistema cancelar la operación y finaliza el caso de uso
5a	El usuario introduce un nombre y dirección que ya coincide con otro del mismo formato que ya existe en esa dirección y con ese nombre. El sistema le explica al usuario el problema y si desea reemplazar el archivo que ya existe por el nuevo o si quiere volver a introducir los datos

Otros datos			
Frecuencia	1000 veces al año	Rendimiento	No más de 30 segundos en alguno de sus pasos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.37: Caso de uso Guardar filtrado.

Caso de uso	Zoom (filtrado)	CU_35
Referencias	RF-12.4	
Precondición	Ninguna	
Postcondición	Se hace zoom sobre el grafo de nodos en base al contenido de la entrada del filtro	
Propósito	Permitir al usuario hacer zoom sobre los nodos que pasen el filtro introducido	
Resumen	El usuario solicita hacer zoom usando el filtro y el sistema hace zoom en el grafo de nodos sobre los nodos que pasen el filtro introducido	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario solicita al sistema que haga zoom teniendo en cuenta el filtro introducido (tanto la cadena de caracteres como la columna seleccionada)	2	El sistema cambia la vista del grafo de nodos para que se enfoque sobre los nodos indicados en el filtro

Otros datos			
Frecuencia	4000 veces al año	Rendimiento	No más de 30 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.38: Caso de uso Zoom (filtrado).

Caso de uso	Cambiar opciones de filtrado	CU_36
Referencias	RF-12.5	
Precondición	Ninguna	
Postcondición	El sistema filtra de nuevo con las nuevas opciones	
Propósito	Permitir al usuario un filtrado más complejo y concreto	
Resumen	El usuario solicita al sistema cambiar las opciones de filtrado. El sistema le muestra las distintas opciones que puede cambiar. El usuario selecciona una opción a cambiar y el sistema vuelve a realizar el filtrado con las nuevas opciones	

Curso Normal (básico) de eventos			
<i>Actor</i>		<i>Sistema</i>	
1	El usuario solicita al sistema cambiar las opciones de filtrado	2	El sistema le muestra las distintas opciones que puede cambiar
3	El usuario introduce un cambio en una opción	4	El sistema vuelve a realizar el filtrado con la opción nueva y termina el caso de uso

Curso Alternativo (secundario) de eventos	
3a	El usuario cancela la operación y termina el caso de uso

Otros datos			
Frecuencia	20000 veces al año	Rendimiento	No más de 30 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.39: Caso de uso Cambiar opciones de filtrado.

Caso de uso	Salir	CU_37
Referencias	RF-14	
Precondición	El programa está abierto	
Postcondición	El sistema cierra el programa	
Propósito	Permitir al usuario cerrar el programa	
Resumen	El usuario solicita al sistema cerrar el programa y el sistema lo cierra	

Curso Normal (básico) de eventos			
Actor		Sistema	
1	El usuario solicita al sistema cerrar el programa	2	El sistema cierra el programa y finaliza el caso de uso

Curso Alterno (secundario) de eventos	
2a	El sistema detecta que hay cambios sin guardar y pregunta al usuario si quiere guardar los cambios. El usuario pide al sistema que guarde los datos. El sistema guarda los cambios en el fichero de la matriz antes de cerrar.
2b	El sistema detecta que hay cambios sin guardar y pregunta al usuario si quiere guardar los cambios. El usuario pide al sistema que guarde los datos. El sistema, al detectar que los cambios no tiene fichero asociado, guarda los cambios en un nuevo fichero cuyo nombre y dirección debe ser introducido por el usuario antes de cerrar.
2c	El sistema detecta que hay cambios sin guardar y pregunta al usuario si quiere guardar los cambios. El usuario pide al sistema que no guarde los datos y el sistema cierra el programa.
2d	El sistema detecta que hay cambios sin guardar y pregunta al usuario si quiere guardar los cambios. El usuario cancela el cierre y finaliza el caso de uso.

Otros datos			
Frecuencia	1000 veces al año	Rendimiento	No más de 10 segundos
Importancia	Alta	Urgencia	Alta
Estado	Terminado	Estabilidad	Alta

Tabla 5.40: Caso de uso Salir.

5.4.3. Diagramas de Modelo de Casos de Uso

En un modelo de casos de uso se representan tanto a los actores como a los casos de uso, estos últimos como elipses. Las flechas indican quién inicia una interacción [BS03]. Las figuras que van desde 5.1 hasta 5.5 exponen los diagramas de caso de uso de la aplicación.

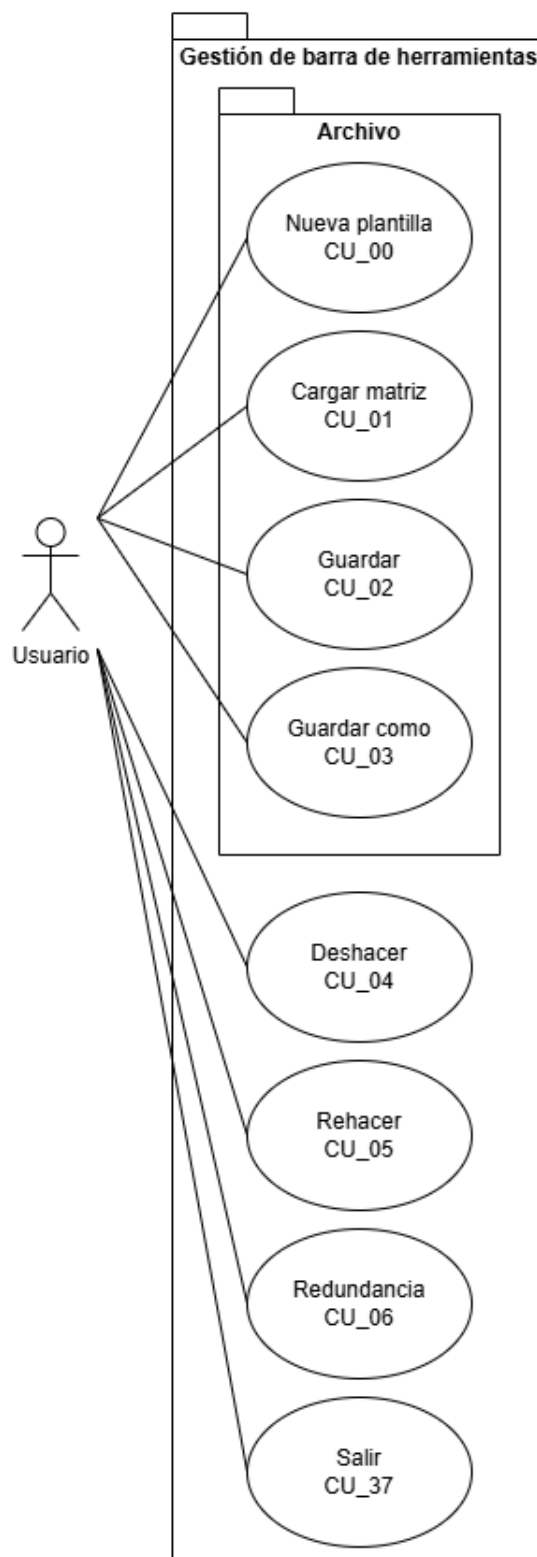


Figura 5.1: Diagrama de Casos de Uso. Barra de herramientas

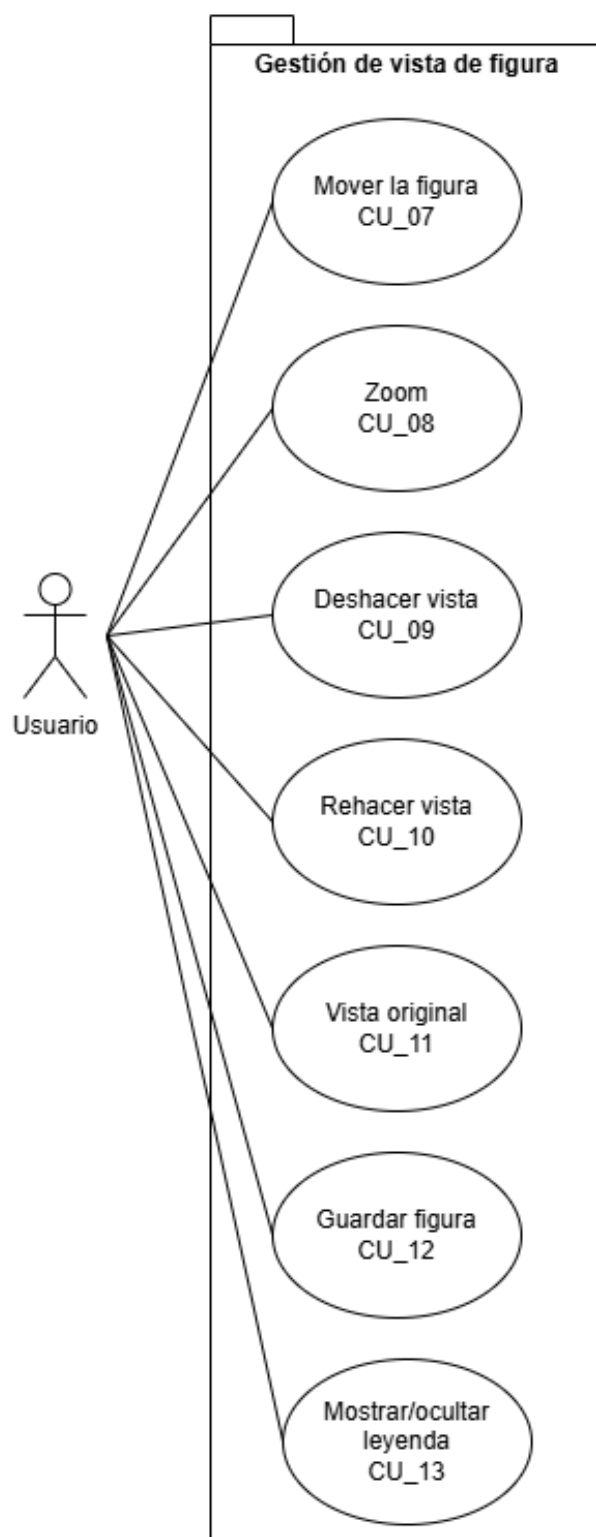


Figura 5.2: Diagrama de Casos de Uso. Vista de figuras

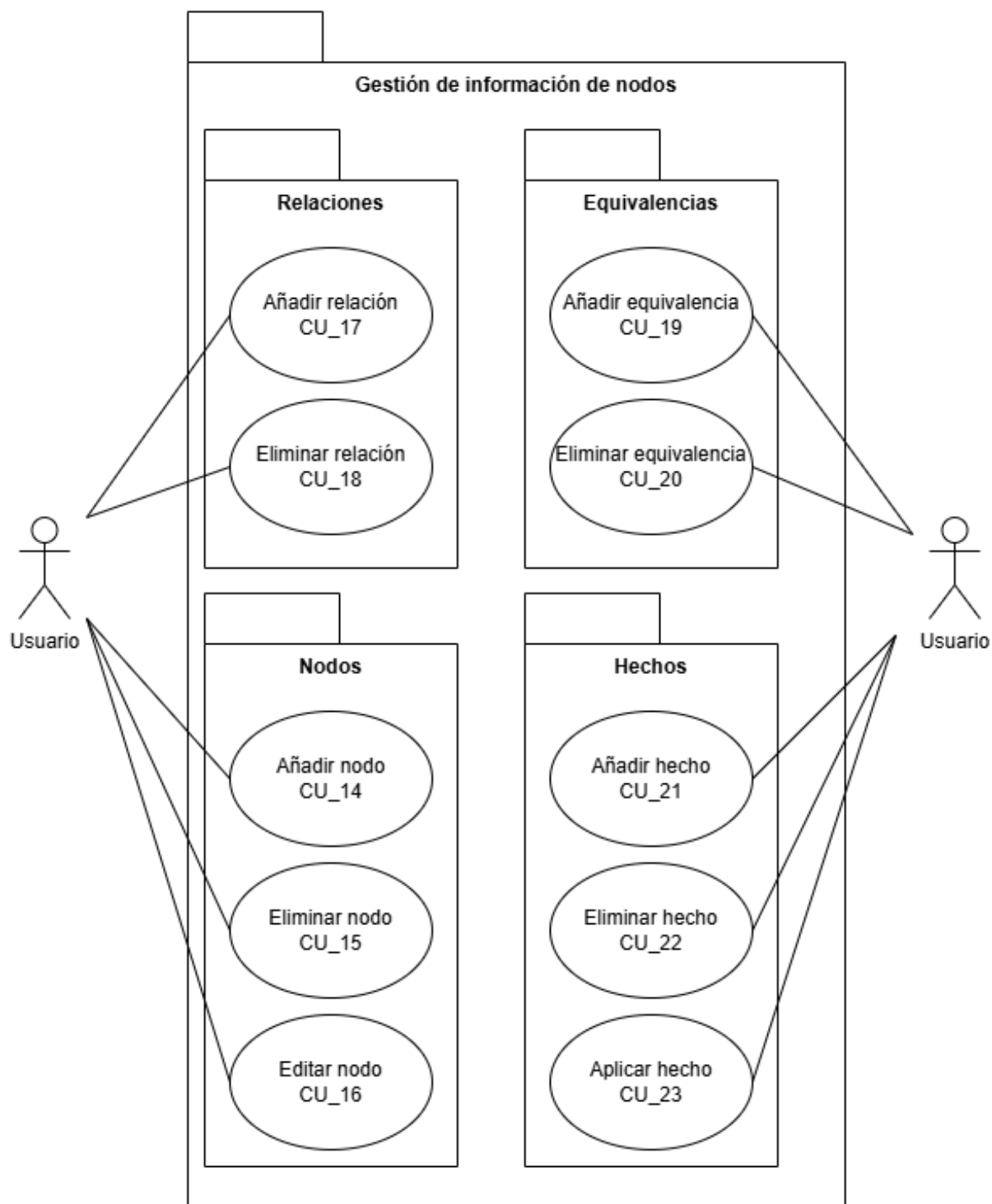


Figura 5.3: Diagrama de Casos de Uso. Edición de listas

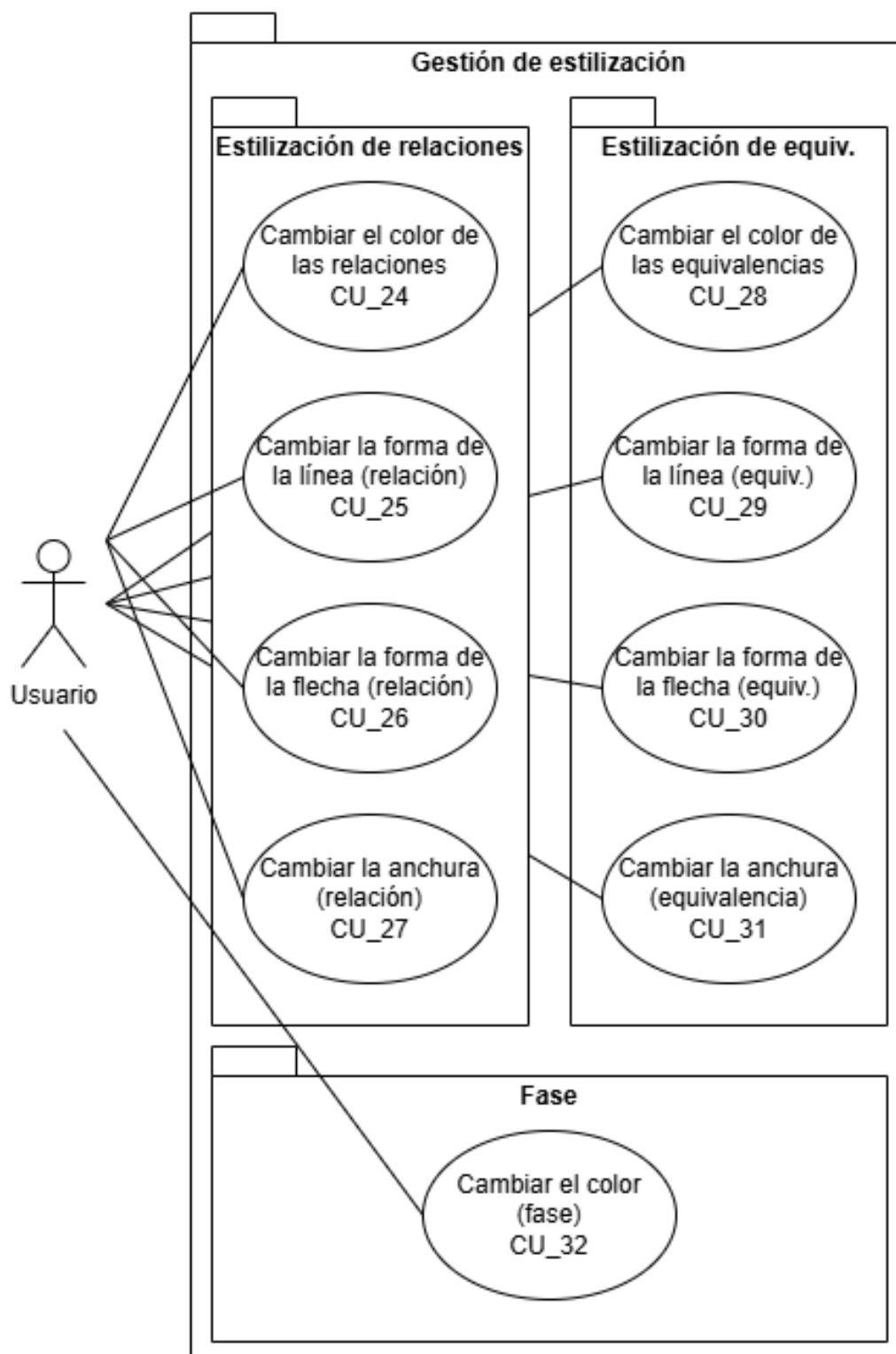


Figura 5.4: Diagrama de Casos de Uso. Estilización

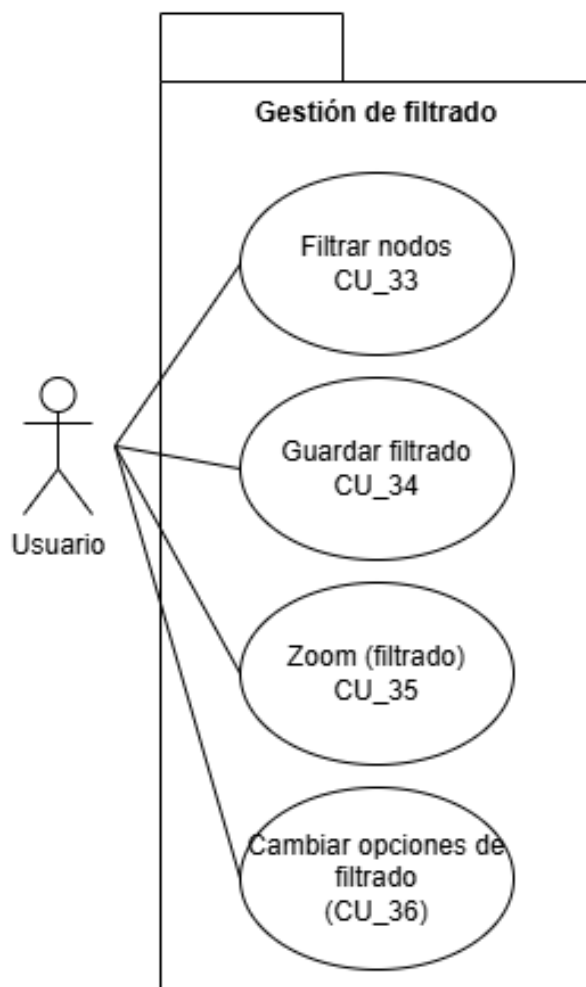


Figura 5.5: Diagrama de Casos de Uso. Filtrado

5.4.4. Diagramas de Actividad

Un diagrama de actividad describe el comportamiento o flujo de trabajo de un sistema o programa utilizando nodos y aristas. El nodo inicial es un punto negro grande que indica el inicio de las interacciones con el sistema mientras que el nodo final es un punto negro grande encapsulado por un círculo e indica el fin de las interacciones con el sistema [Ikr+15].

A continuación se muestra una vista completa del diagrama de actividad de la aplicación en la figura 5.6.

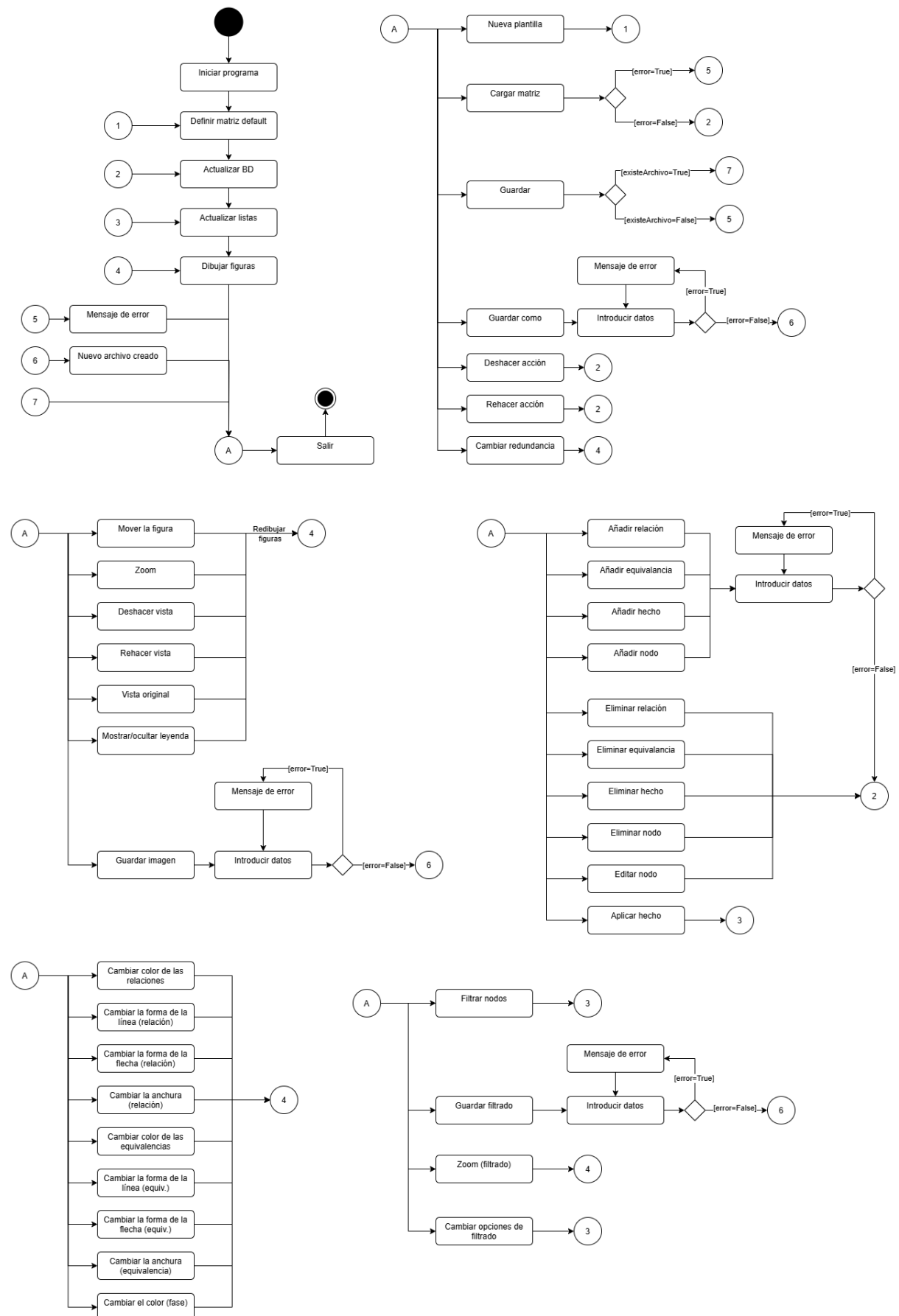


Figura 5.6: Diagrama de Actividad. Vista completa.

En la figura 5.7 se muestra el flujo de actividades al iniciar el programa. También se puede apreciar que hay varios conectores. Un **conector** conecta varios flujos para mayor legibilidad del diagrama.

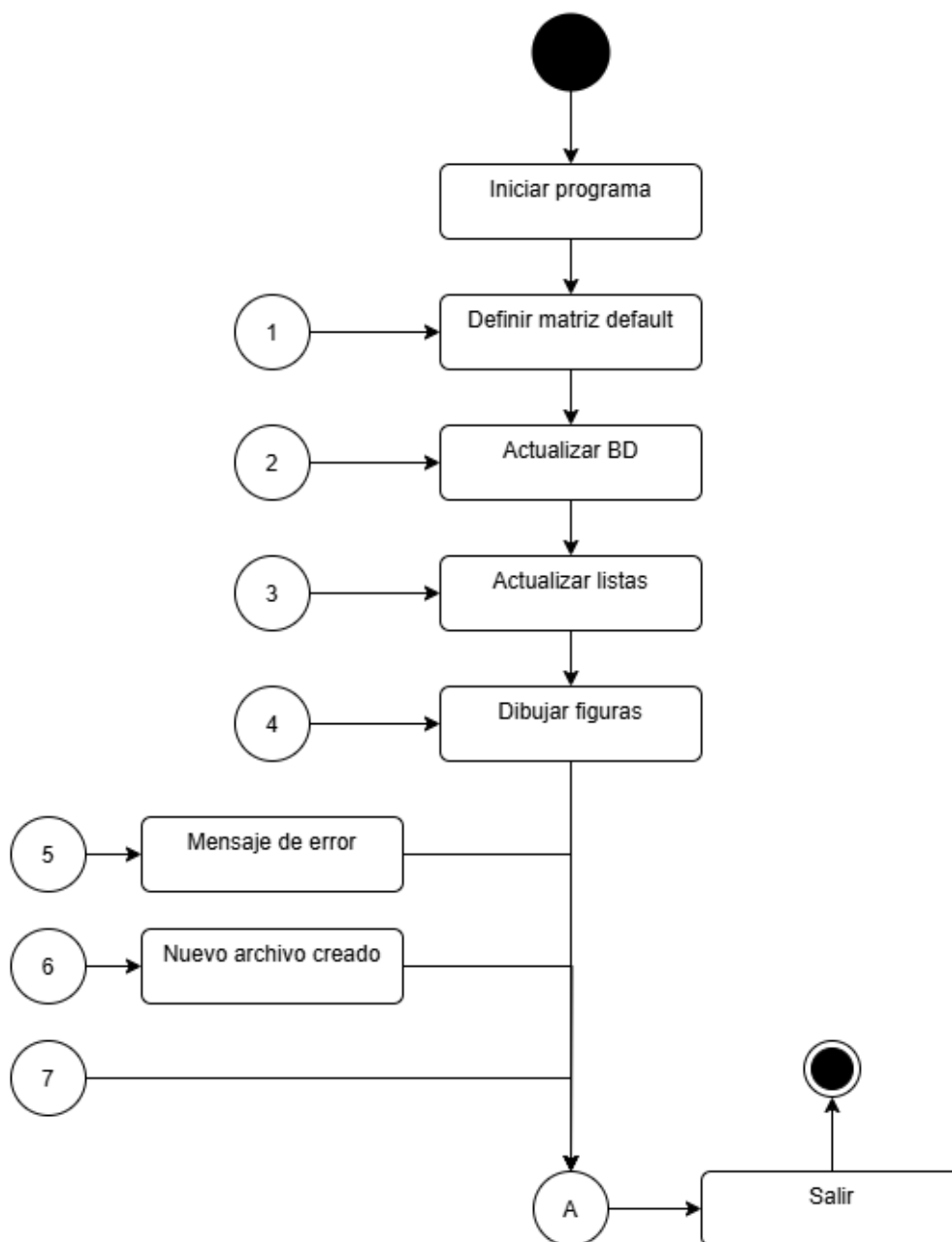


Figura 5.7: Diagrama de Actividad. Detalle A.

En la figura 5.8 se muestra el flujo de actividades para la barra de herramientas de la aplicación.

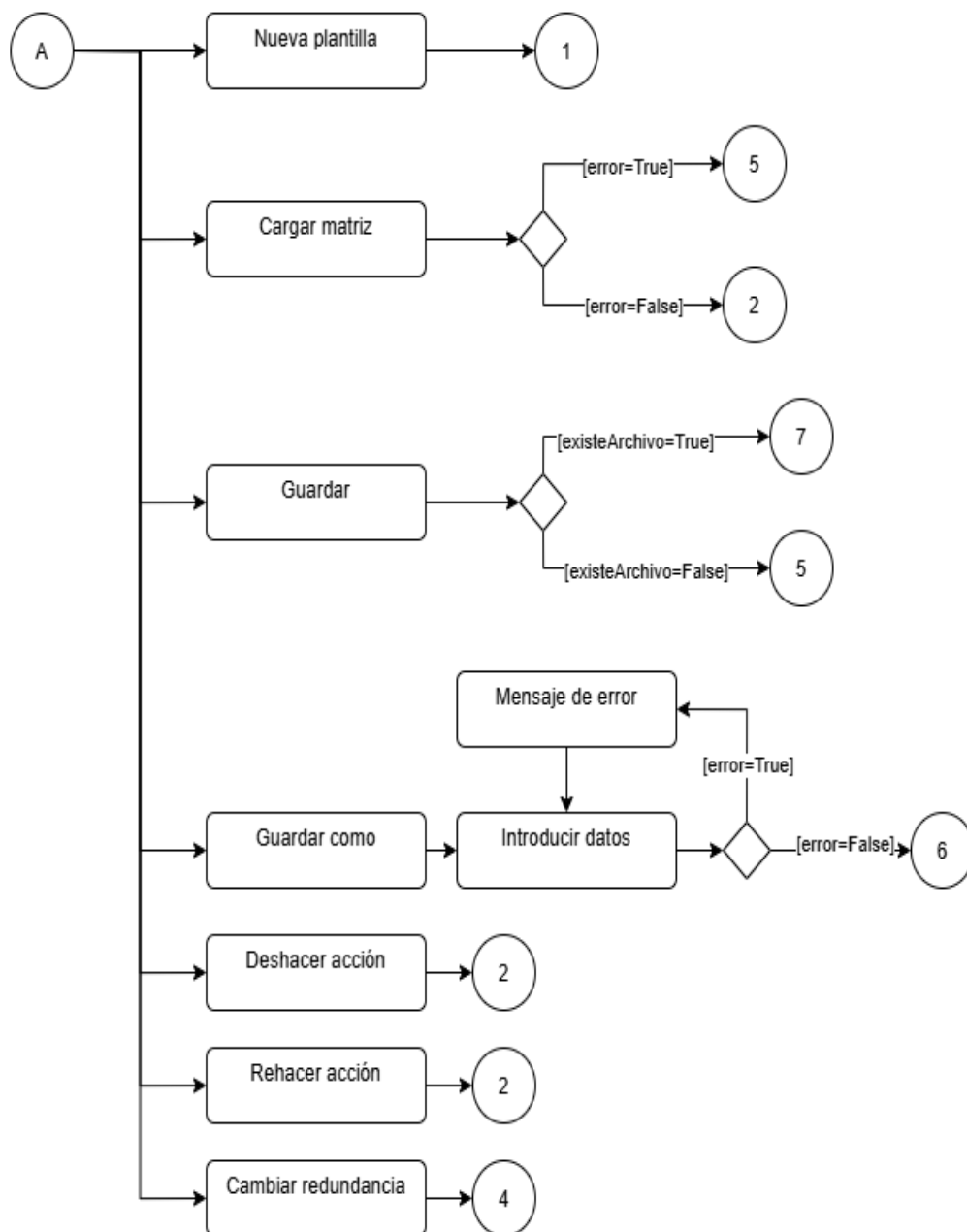


Figura 5.8: Diagrama de Actividad. Detalle B.

En la figura 5.9 se representa el flujo de actividades para la barra de herramientas de las figuras.

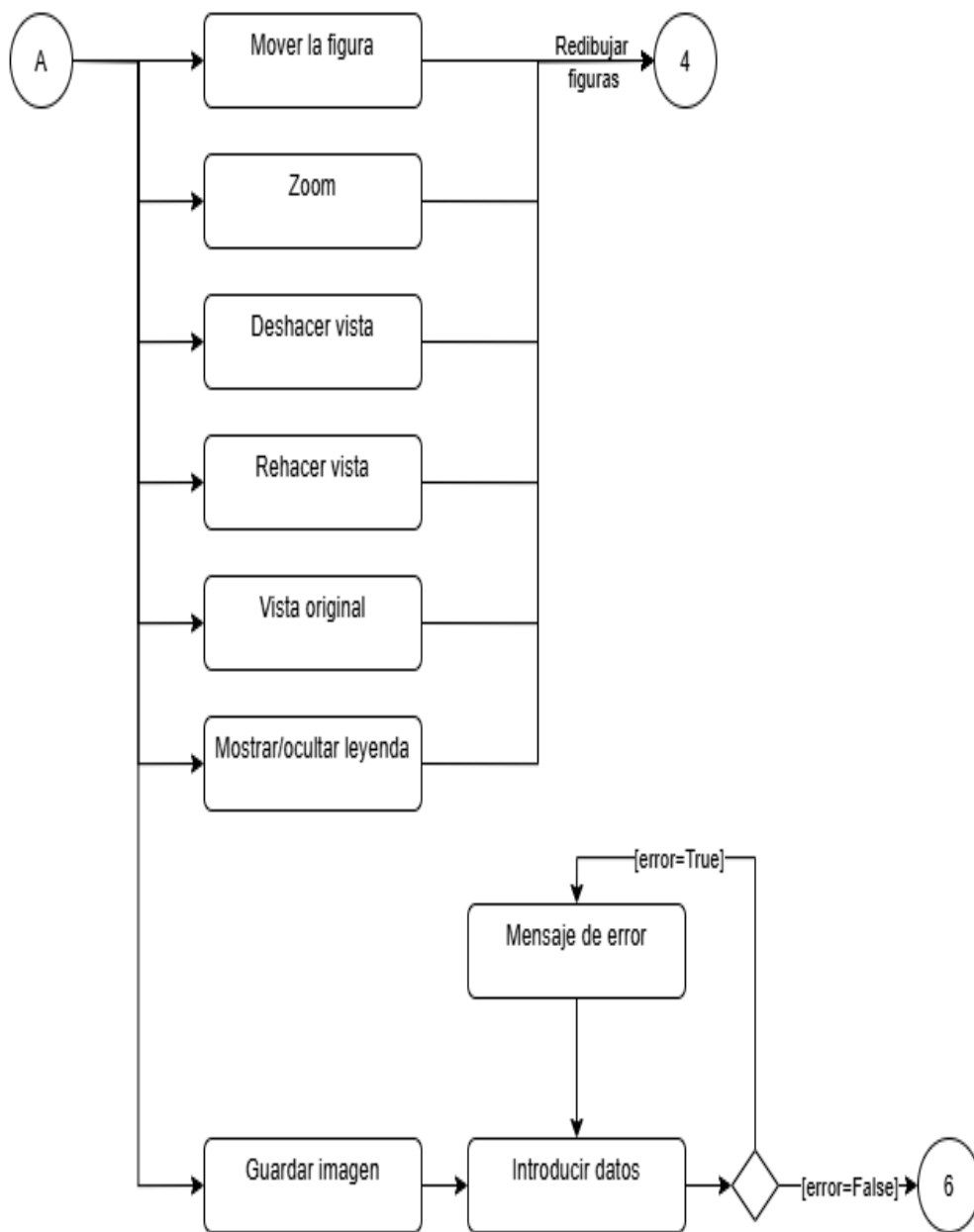


Figura 5.9: Diagrama de Actividad. Detalle C.

La figura 5.10 representa el flujo de todas aquellas acciones que editan la información en la base de datos.

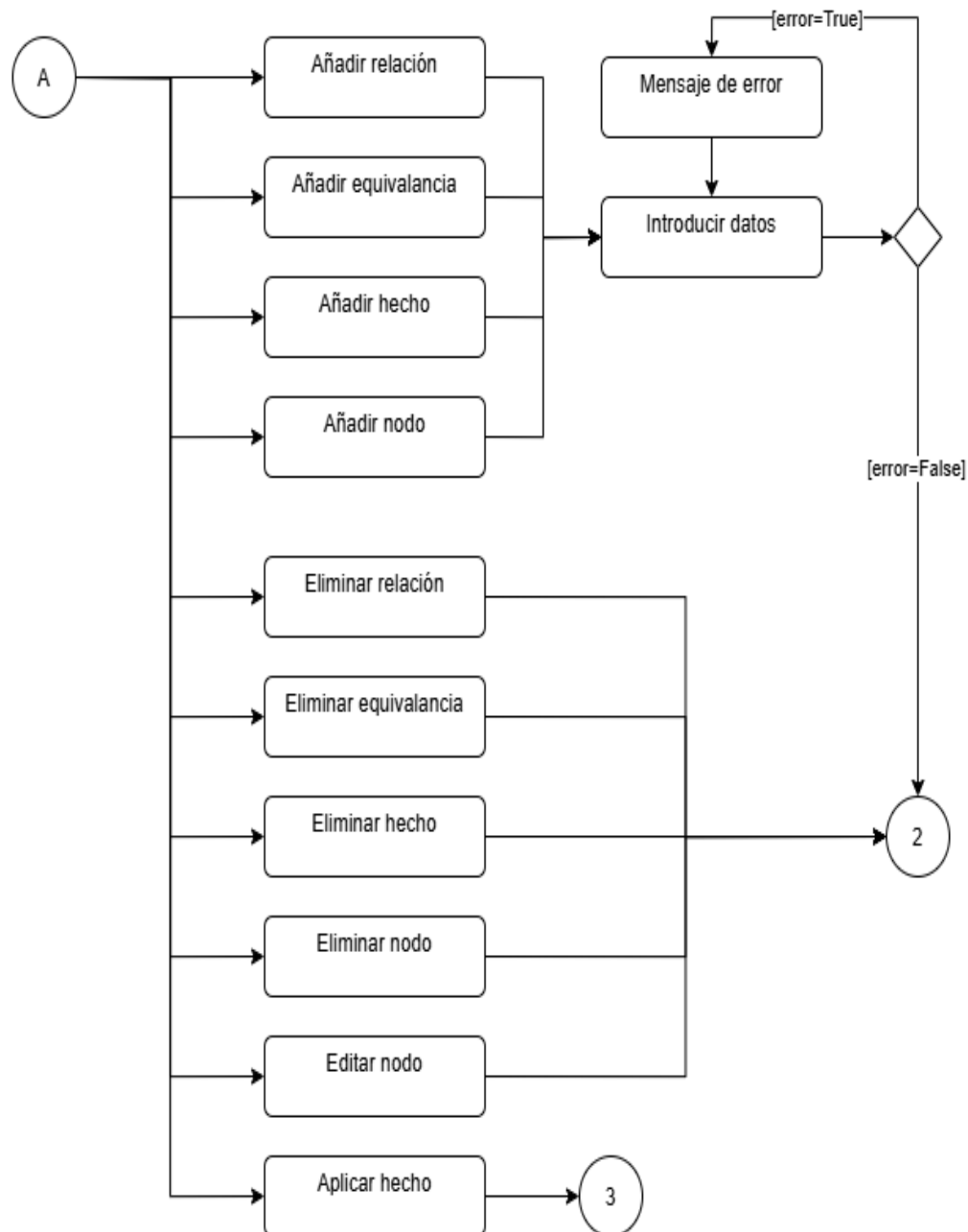


Figura 5.10: Diagrama de Actividad. Detalle D.

La figura 5.11 muestra el flujo de actividad de las acciones de personalización del grafo.

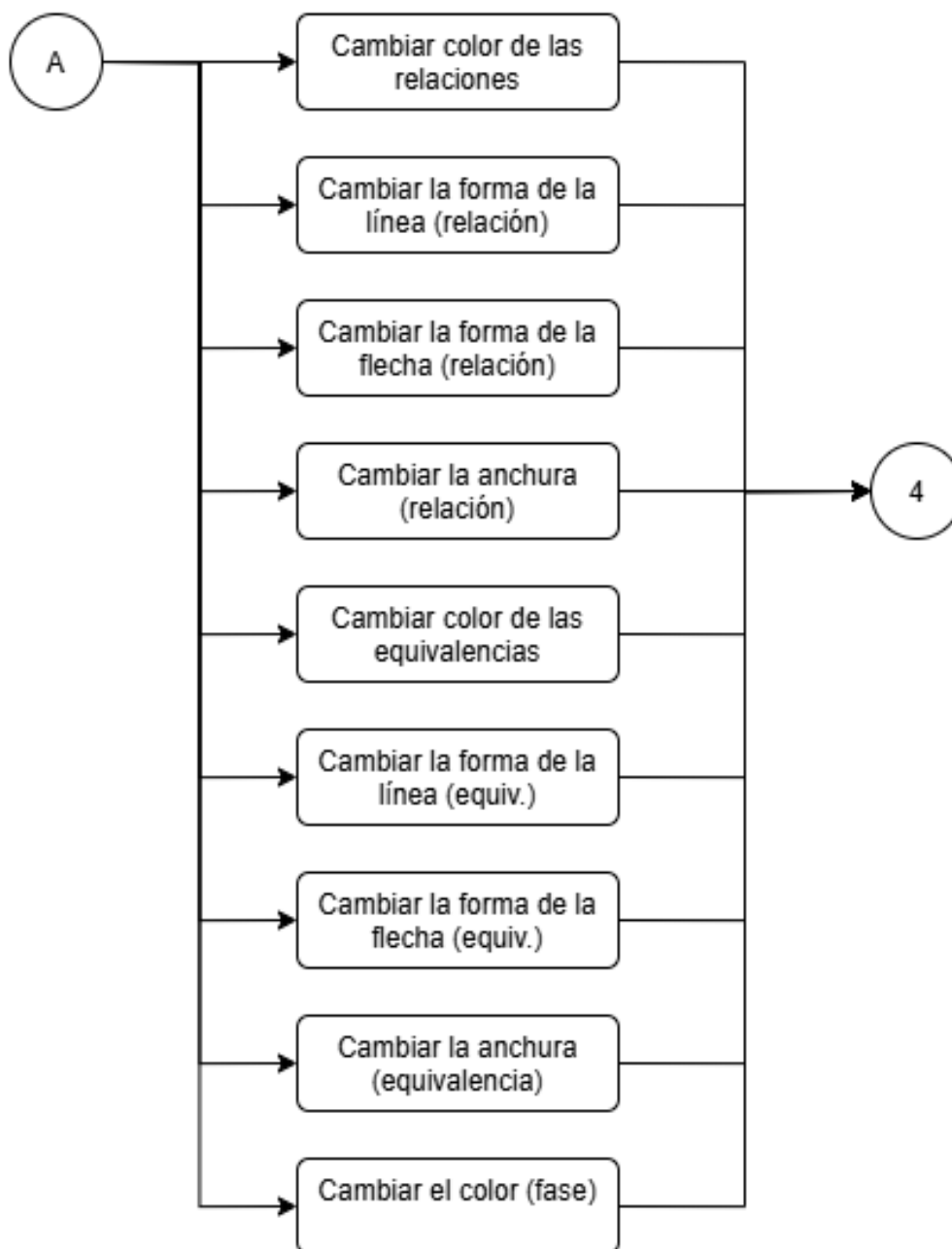


Figura 5.11: Diagrama de Actividad. Detalle E.

La figura 5.12 representa el flujo de actividad de las acciones relacionadas con el filtrado.

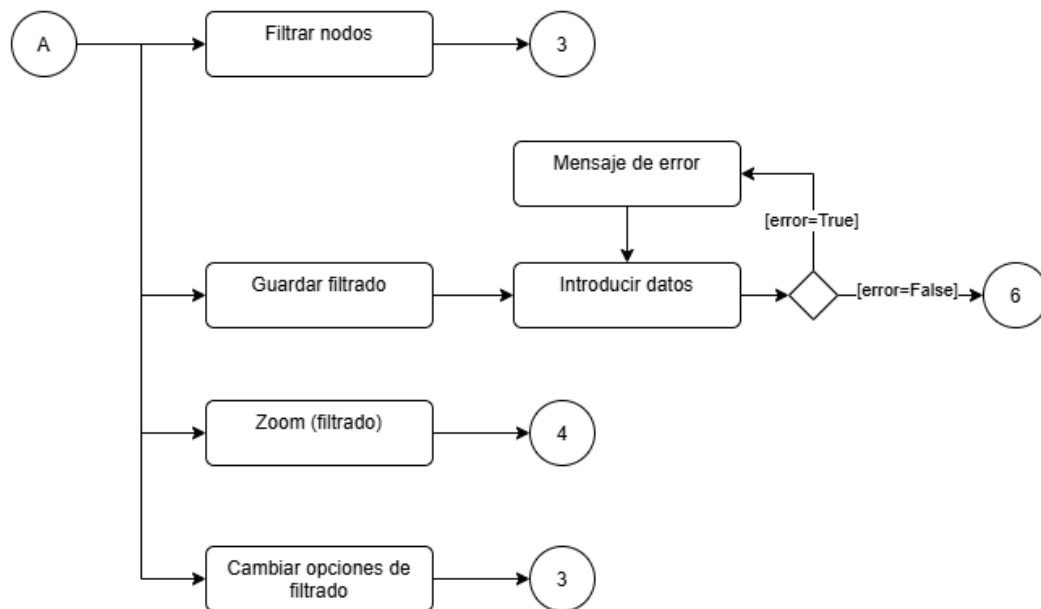


Figura 5.12: Diagrama de Actividad. Detalle F.

Capítulo 6

Diseño

En este apartado se explicarán cuáles han sido las principales problemáticas de diseño de la aplicación, las distintas soluciones que se barajaron para cada problemática y cuál fue la decisión final que se tomó respecto a cada una.

6.1. Plataforma

La plataforma se refiere al formato en el que se presenta la aplicación. Había dos opciones: **aplicación web** o **aplicación de escritorio**. Aquí se analizan las ventajas e inconvenientes en cada una y cuál fue la decisión tomada y por qué.

Una **aplicación web** es un programa de software que se ejecuta en un servidor web en vez de en un sistema operativo. Esto quiere decir que no se instalan en el dispositivo local sino que están en la nube, pudiéndose acceder desde un buscador.

Por otra parte, una **aplicación de escritorio** es una aplicación que está instalada en el dispositivo y es ejecutada por el sistema operativo.

En la tabla 6.1 se realiza una comparativa de las características generales de cada tipo de aplicación según la plataforma¹.

¹<https://medium.com/theymakedesign/web-app-vs-desktop-app-3841e8cb3996>

Tabla 6.1: Comparación entre Web y Escritorio

	Web	Escritorio
Accesibilidad	<ul style="list-style-type: none"> ■ Acceso desde casi cualquier dispositivo con buscador y acceso a Internet. ■ Limitante cuando hay baja conectividad. ■ Más adaptable en general. 	<ul style="list-style-type: none"> ■ Solo se puede acceder desde el dispositivo donde está instalado.
Rendimiento	<ul style="list-style-type: none"> ■ Limitado por el buscador. ■ Limitado por la velocidad del Internet. 	<ul style="list-style-type: none"> ■ Utilizan los recursos del sistema. ■ Más potente para procesamiento de datos y visualización gráfica.
Seguridad	<ul style="list-style-type: none"> ■ Más expuesto a ataques informáticos. ■ La seguridad depende de la seguridad del servidor. 	<ul style="list-style-type: none"> ■ Menos expuestos a ataques informáticos, puesto que la información se guarda en el dispositivo. ■ Menos seguro si se usa con conectividad. ■ La seguridad depende de la seguridad del dispositivo.
Escalabilidad	<ul style="list-style-type: none"> ■ Las actualizaciones se realizan sin que el usuario tenga que instalar otra versión o actualizaciones. 	<ul style="list-style-type: none"> ■ Si hay una nueva versión hay que instalarla manualmente en el dispositivo.

6.1.1. Conclusión

Tras analizar ambas opciones en la tabla 6.1, podemos llegar a la siguiente conclusión: las aplicaciones web son más flexibles a nivel de accesibilidad y escalabilidad, mientras que las aplicaciones de escritorio son más robustas en cuanto a seguridad y rendimiento.

Para este proyecto hay que tener en cuenta que puede haber hasta mil nodos por grafo, por lo que el rendimiento es algo a tener en cuenta. Puesto que los arqueólogos trabajan con información susceptible a ser sensible (coordenadas de las excavaciones, hallazgos dentro de una excavación, etc...) la seguridad de la aplicación también debe ser tomada en cuenta. Se podría concluir que una aplicación de escritorio es la mejor opción en este caso.

Otro motivo para rechazar la opción web es que dependiendo de la velocidad de Internet cada acción puede tomar de media más tiempo que si fuese una aplicación de escritorio. Además, los arqueólogos suelen pasar largos períodos de tiempo en excavaciones que por lo general suelen estar en zonas alejadas de centros urbanos donde rara vez tienen una conexión a la red, y si la tienen, suele ser de mala calidad. Por lo tanto, es mejor una aplicación a la que puedan acceder sin preocuparse por si tienen señal o no [Pop02].

Cabe mencionar que la mayoría de aplicaciones ya existentes para diseñar matrices de Harris son aplicaciones de escritorio, por lo que tiene sentido desarrollar el programa como una aplicación de escritorio, ya que los usuarios que ya hayan trabajado con plataformas así estarán más familiarizados con este tipo de programas.

Decisión final: Aplicación de escritorio.

6.2. Diseño de la interfaz

Inicialmente me basé en el diseño de la interfaz de *Harris Matrix Composer* para la interfaz de mi programa, de la cual se puede ver una captura en la figura 4.5. La figura 6.1 muestra el boceto inicial de la interfaz de la aplicación. Se mostrarán imágenes de la aplicación terminada para ilustrar las decisiones de diseño tomadas.

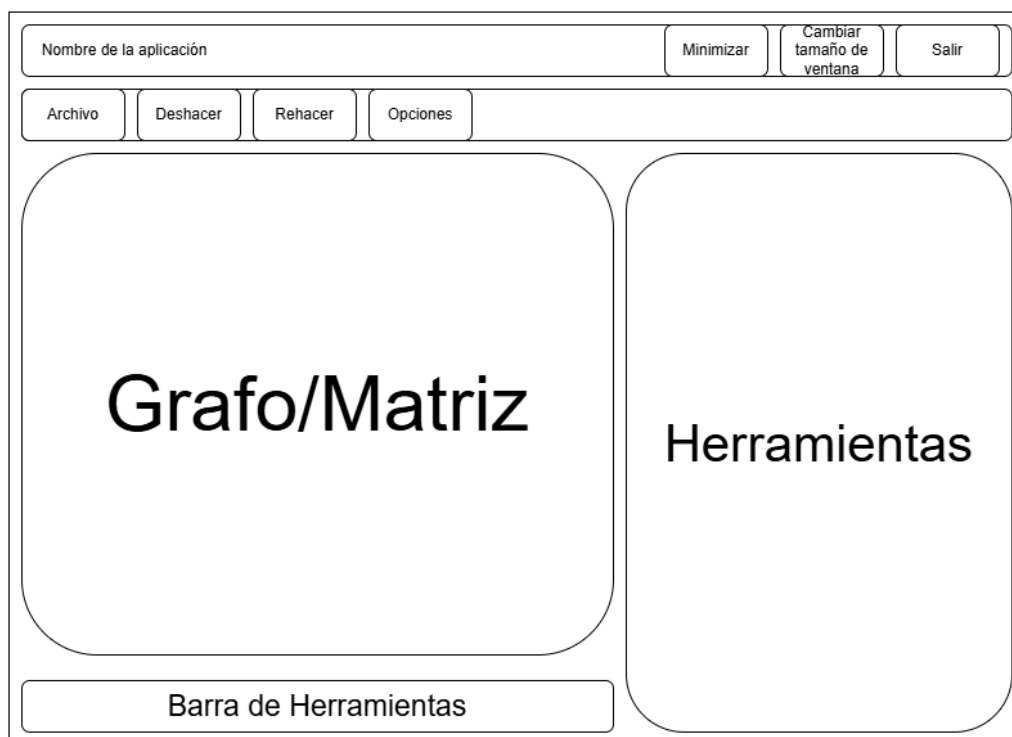


Figura 6.1: Boceto inicial de la interfaz de la aplicación.

Se añadió una barra de herramientas para la vista del grafo justo debajo de la ventana donde se dibuja el grafo (véase la Figura 6.2: Detalle: Barra de herramientas para la vista). A dicha barra se le añadió un botón nuevo que permitía mostrar o no la leyenda en la figura.



Figura 6.2: Detalle: Barra de herramientas para la vista

En la barra de herramientas de vista se pueden apreciar los siguientes botones:

- **Inicio:** Cambia la vista actual a la original.
- **Deshacer:** Vuelve a la vista anterior.
- **Rehacer:** Avanza a la vista posterior.
- **Mover:** Mueve la vista de la figura por los ejes X e Y.
- **Zoom:** Permite hacer zoom in o zoom out sobre una zona de la vista.

- **Guardar:** Genera una imagen de la vista actual.
- **Mostrar leyenda:** Muestra la leyenda de los elementos de la figura en la vista.

6.2.1. Ventana de las figuras

Tras meditar distintas alternativas para permitir poder elegir entre visualizar el grafo o la matriz se decidió crear un *notebook* con una pestaña para el grafo y otra para la matriz para alternar entre ambas, como se puede ver en las figuras 6.3 y 6.4.

Se puede apreciar un grafo con forma $T \rightarrow Unexcavated \rightarrow G$. Este es el grafo que se muestra al iniciar la aplicación *Harris Matrix Composer*, que fue de donde se tomó la inspiración.

El nodo T sería la superficie, *Unexcavated* los estratos arqueológicos sin excavar, y G la interfaz de la geología [TNI08]. También es la matriz que se carga si se solicita crear una nueva matriz de Harris al sistema.

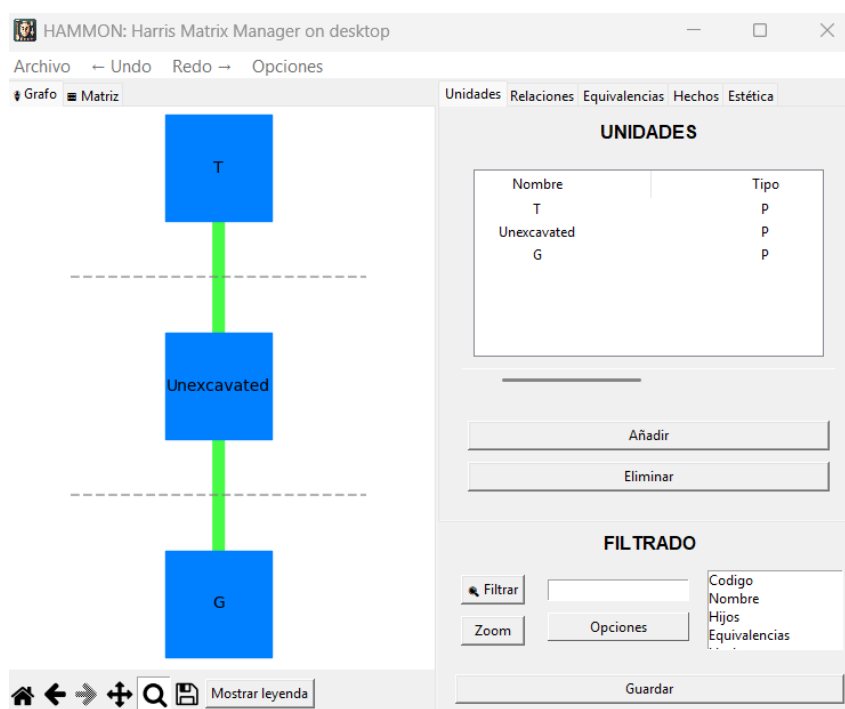


Figura 6.3: HAMMON: Aplicación con la pestaña Grafo abierta.

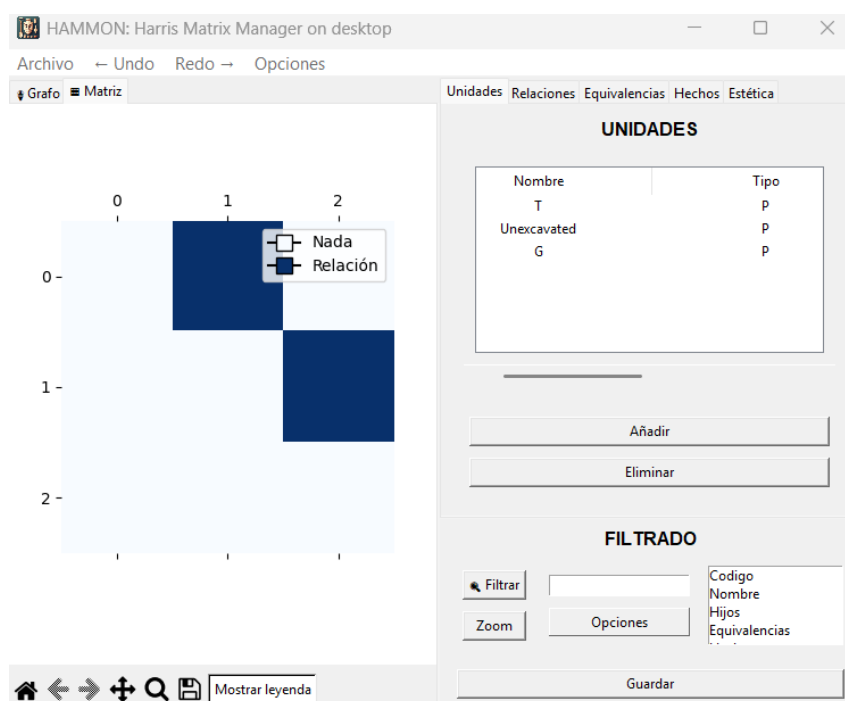


Figura 6.4: HAMMON: Aplicación con la pestaña Matriz abierta.

6.2.2. Nodos

Al hablar de nodos, se refiere tanto de las unidades estratigráficas como de los hechos. En el grafo de nodos cada tipo de nodo se dibuja de manera diferente ²:

- **Unidad positiva:** Refiere a cuando un estrato ha sido generado añadiendo material (muro, roca, etc...). Se representa como un cuadrado.
- **Unidad negativa:** Refiere a cuando un estrato ha sido generado eliminando (agujero, zanja, etc...) material. Se representa como un círculo.
- **Hecho:** Se refiere a cuando se agrupan varias unidades u otros hechos en un solo nodo. Se representa como un hexágono.

Se puede ver una representación de todos estos elementos en la figura 6.5.

²<https://archgoodpractice.com/glossary/>

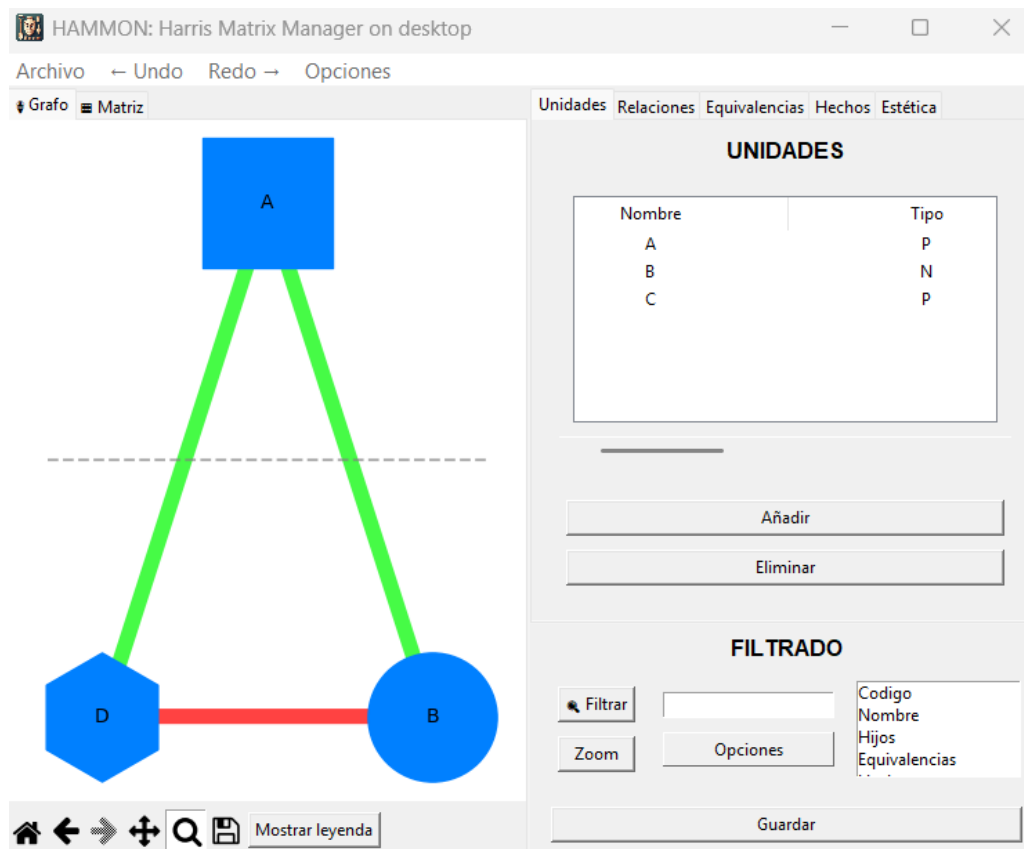


Figura 6.5: En la ventana de HAMMON se muestra un grafo de nodos con los tres tipos de nodos. En el sentido contrario de las agujas del reloj: A es una unidad positiva (cuadrado), B es una unidad negativa (círculo) mientras que D es un hecho (hexágono).

6.2.3. Información de la matriz

En la aplicación todos los elementos de la matriz de Harris que se pueden editar están en distintas ventanas de un *widget* o herramienta llamado **notebook**. Tal como su nombre indica, puedes ir a distintas secciones del *notebook* pulsando sobre la pestaña correspondiente. Las pestañas disponibles son:

- **UNIDADES:** Tiene una tabla que muestra los nodos positivos y negativos, un botón de adición y un botón de eliminación (véase la Figura 6.6: HAMMON: Unidades).

Relaciones Equivalencias Hechos Unidades Estética

UNIDADES

Nombre	Tipo
T	p
Unexcavated	p
G	p

Añadir

Eliminar

Figura 6.6: HAMMON: Unidades.

- **RELACIONES:** Cuenta con una tabla que muestra las relaciones, dos desplegables con todos los nodos positivos y negativos, un botón de adición y otro de eliminación (véase la Figura 6.7: HAMMON: Relaciones).

The screenshot displays the 'RELACIONES' (Relationships) window of the HAMMON application. At the top, there is a tabbed interface with five tabs: 'Relaciones' (selected), 'Equivalencias', 'Hechos', 'Unidades', and 'Estética'. The main area is titled 'RELACIONES' and contains a table with two columns: 'Unidad 1' and 'Unidad 2'. The first row of the table shows 'T' under 'Unidad 1' and 'Unexcavated' under 'Unidad 2'. Below the table, there is a control bar with an 'Añadir' (Add) button, two dropdown menus (both currently showing 'T'), and an 'Eliminar' (Remove) button.

Unidad 1	Unidad 2
T	Unexcavated

Añadir T T Eliminar

Figura 6.7: HAMMON: Relaciones.

- **EQUIVALENCIAS:** Igual que en RELACIONES pero para equivalencias (véase la Figura 6.8: HAMMON: Equivalencias).

The interface is a window titled "HAMMON: Equivalencias". It features a tabbed interface at the top with the following tabs: "Relaciones", "Equivalencias" (which is the active tab), "Hechos", "Unidades", and "Estética". The main content area is titled "EQUIVALENCIAS" and contains a large, empty rectangular box. This box is divided into two columns by a vertical line, with the left column labeled "Unidad 1" and the right column labeled "Unidad 2". At the bottom of the window, there is a control bar. This bar contains an "Añadir" button on the left, followed by two dropdown menus. Both dropdown menus currently display the letter "T". To the right of these dropdowns is an "Eliminar" button.

Figura 6.8: HAMMON: Equivalencias.

- **HECHOS:** Dispone de dos ventanas: **INFORMACIÓN INTERNA** y **AGRUPAMIENTOS**.

La ventana **INFORMACIÓN INTERNA** Tiene una tabla que muestra los nodos de tipo hecho, un botón de adición y un botón de eliminación (véase la Figura 6.9: HAMMON: Hechos - Información interna).

Relaciones Equivalencias Hechos Unidades Estética

Información Interna Agrupamiento

INFORMACIÓN INTERNA

Nombre	Tipo
--------	------

Añadir

Eliminar

Figura 6.9: HAMMON: Hechos - Información interna.

En la ventana **AGRUPAMIENTOS** hay tabla que muestra los nodos tipo hecho y los nodos que engloban, un desplegable con los nodos tipo hecho, otro desplegable con todos los nodos, un botón de adición, un botón de eliminación, y un botón de aplicación de hechos junto a una entrada de texto (véase la Figura 6.10: HAMMON: Hechos - Agrupamiento).

Relaciones Equivalencias **Hechos** Unidades Estética

Información Interna Agrupamiento

AGRUPAMIENTO

Hecho	Unidades
-------	----------

Añadir [▼] T [▼]

Eliminar

Aplicar []

Figura 6.10: HAMMON: Hechos - Agrupamiento.

- **EDICIÓN ESTÉTICA:** Tanto para relaciones como para equivalencias cuenta con un botón para editar el color, que al pulsarlo te muestra una ventana para seleccionar el color, un desplegable para seleccionar la forma de la línea, otro desplegable para seleccionar la forma de la flecha y una barra de desplazamiento horizontal que permite cambiar la anchura en un rango de [1, 8]. También tiene una tabla con cada fase y su color asociado. Al pulsar sobre una línea de la tabla de fases, aparece una ventana que permite cambiar el color (véase la Figura 6.11: HAMMON: Edición estética).

Relaciones Equivalencias Hechos Unidades **Estética**

EDICIÓN ESTÉTICA

RELACIONES

Color

Línea solid

Flecha -

Anchura 1

EQUIVALENCIAS

Color

Línea solid

Flecha -

Anchura 1

FASES

Fase	Color
------	-------

Figura 6.11: HAMMON: Edición estética.

Cabe aclarar que dentro de la base de datos de la aplicación los hechos se representan como otro tipo de nodos, junto con los positivos y con los negativos.

6.2.4. Estética

Como se ha explicado en el Capítulo 4: Estado del arte en la Subsección 4.1.1: ArchEd, *ArchEd* cuenta con la funcionalidad de cambiar el color de los nodos que representan las unidades estratigráficas. Tomando inspiración de esta función, se incluyó en el diseño final de la aplicación una ventana que permitía cambiar la estética de las aristas de relaciones y de equivalencias y cambiar el color de todos los nodos dentro de una fase. La figura 6.11 muestra una imagen de la ventana de edición estética en la aplicación final.

6.2.5. Filtrado

El filtrado tiene una sección propia y no una pestaña. La razón principal por la que está separado y no es una pestaña propia es sobre todo porque es una funcionalidad que afecta a toda la información de la matriz y no solo a un elemento o parte concreta. La figura 6.12 muestra el resultado final.



Figura 6.12: HAMMON: Filtrado

Se puede observar lo siguiente:

- Hay dos botones con las etiquetas **Filtrar** y **Zoom**. Estos dos botones, junto con la entrada de texto plano, una lista de selección de opciones y otra con las columnas, se encargan de filtrar los nodos que aparecen en las figuras y de hacer zoom sobre estas, respectivamente.
- En la entrada de texto se indica la cadena de texto con la que trabajará el filtrado y el zoom. Puedes trabajar con más de una cadena de texto usando comas (*valor1, valor2,...*).
- La lista de selección muestra todas las columnas del fichero subido. Se puede seleccionar sobre cuáles se aplica el filtrado o zoom. Si no se escoge ninguna, el programa trabaja sobre todas.

- La lista de opciones tiene los valores *Coincidir mayúsculas y minúsculas*, *Coincidir palabras completas* y *Coincidir diacríticos*. Si están activadas, se aplica sobre cada valor de la entrada de texto el criterio homónimo.
- El botón **Guardar** guarda la información filtrada en un archivo .csv.

6.3. Prevención de errores

La matriz de Harris no debe presentar errores en su información interna. Algunas de las comprobaciones a realizar son:

- No se producen ciclos, es decir, un nodo es a la vez ancestro y descendiente de otro. Una unidad estratigráfica no puede estar encima y debajo de otra a la vez.
- Una unidad debe ser o positiva o negativa.
- Una unidad o hecho no puede estar agrupado por dos hechos distintos a la vez.

Estos errores, junto con su implementación, se explicarán más en detalle en la Sección 7.3: Comprobación de errores.

6.4. Redundancia y reducción transitiva

En la aplicación se ha incluido la opción de aplicar o no redundancia al grafo y la matriz. Cuando en esta memoria se hable de redundancia, se referirá a la **no transitividad** de un grafo dirigido de nodos. Podemos usar la siguiente definición para saber cuándo un grafo dirigido es transitivo: “Un grafo dirigido G^t se dice que es una reducción transitiva del grafo dirigido G si (i) G^t tiene un camino dirigido del vértice u al vértice v si y solo si G tiene un camino dirigido del vértice u al vértice v y (ii) no hay un grafo con menos arcos que G^t que satisfacen la condición (i)” [AGU72].

Es decir, si están guardadas las siguientes relaciones $A \rightarrow B, B \rightarrow C, A \rightarrow C$, cuando no se aplique redundancia solo se representarán las dos primeras relaciones, mientras que si se aplica redundancia se representarán todas las relaciones.

Por tanto, cuando se aplique la redundancia, el grafo de nodos no será un diagrama de Hasse al no aplicar reducción transitiva.

6.5. Icono

El icono en la figura 6.13 fue generado mediante inteligencia artificial, más específicamente, usando AI Icon Generator. Se usa como el icono de la memoria.



Figura 6.13: Icono de la aplicación

Capítulo 7

Implementación

En este capítulo se explican algunas decisiones tomadas sobre la implementación de la aplicación. Primero se habla de los distintos lenguajes que se barajaron para programar la aplicación y cuál se escogió al final. Posteriormente se habla de la extensión del archivo de la aplicación y del formato de los ficheros que se cargan al programa, además de la comprobación de posibles errores que puedan aparecer en estos últimos. Finalmente se muestra el diagrama de clases de la aplicación.

7.1. Elección del lenguaje

En esta sección se debaten lenguajes de programación candidatos a usarse para programar la aplicación y se explica cuál se ha decidido y por qué.

Se consideraron dos candidatos potenciales: **C++** y **Python**. La principal razón por la que se tuvieron en cuenta fue por la experiencia previa de haberlos usado antes: para muchas de mis asignaturas, especialmente de primer y segundo año, el lenguaje que más se utilizaba era C++, mientras que en asignaturas de mi mención (Computación y Sistemas Inteligentes) el lenguaje por excelencia era Python. Si bien utilicé otros lenguajes en otras asignaturas como Ruby o Java, no los he utilizado tan asiduamente como para sentirme cómodo programando con ellos. Además, tanto C++ como Python tienen una gran cantidad de documentación, por lo que las dudas que surgiesen en torno a uso iban a ser más fáciles de resolver.

7.1.1. C++

Según [Str86], C++ es un lenguaje de propósito general, siendo una versión mejorada del lenguaje C que permite abstracción de datos y programación orientada a objetos.

Entre los puntos fuertes de C se encuentran la posibilidad de aplicarlo en una gran variedad de campos (flexibilidad), su parecido con la computación tradicional (eficiencia), su facilidad para compilar en casi cualquier dispositivo (disponibilidad) y la capacidad de pasar un programa C de una máquina a otra (portabilidad).

Dentro de C++ se encuentran algunas bibliotecas para diseño de GUI, como **Qt 4** [BS08] y **wxWidgets** [BDV09], que destacan por su capacidad multiplataforma, y **gtkmm** dentro de GTK, una biblioteca de C++ enfocada al desarrollo de GUI ¹.

7.1.2. Python

El lenguaje **Python** toma su nombre de *Monty Python's Flying Circus*. Según [Pre03]: “Python es un lenguaje de programación interpretado, interactivo y orientado a objetos que combina una potencia notable con una sintaxis muy limpia”.

El principal motivo por el que se considera como candidato es por haberlo usado más asiduamente en las últimas asignaturas cursadas de la carrera. Otro motivo de peso es el gran número de librerías que posee Python, por lo que tiene una gran variedad de herramientas a las que recurrir. Además, Python tiene una sintaxis más sencilla que C++ [Bal22].

En Python existe un módulo llamado **Tkinter** que proporciona una interfaz al kit de herramientas de Tk GUI. Aunque hay más bibliotecas para desarrollo de GUI como PyQt y WxPython, a mi parecer Tkinter es el mejor por una serie de razones:

- **Fácil de aprender:** Dado que será la primera vez que desarrolle una aplicación con Python, lo mejor es usar una biblioteca que pueda aprender en poco tiempo.
- **Poco código para crear aplicaciones:** Lo mejor es que una aplicación ocupe el menor espacio posible.
- **Multisistema:** Se puede portar a varios sistemas operativos, por lo general, a la mayoría de sistemas Unix además de Windows y Macintosh.
- **Accesible:** Suele estar preinstalado con las distribuciones estándar de Python.

Además, como suele ser la biblioteca que más se usa para desarrollo GUI en Python, hay más documentación disponible a la que recurrir [Lun99; BE+16; Pod19].

¹<https://gtkmm.gnome.org/en/index.html>

7.1.3. Conclusión

Tras analizarlo, finalmente se ha decidido decantarse por utilizar **Python** para desarrollar la aplicación. A continuación se presentan las razones por las que se ha tomado esta decisión:

- **Conocimiento previo:** Ya se había usado Python anteriormente para otras asignaturas, y aunque se podría decir lo mismo de C++, Python es el lenguaje que más se ha usado durante las últimas asignaturas cursadas.
- **Facilidad de uso:** Desde mi experiencia, programar con Python es más sencillo que en C++. Era preferible sacrificar algo de potencia por una mayor legibilidad del código, ya que al ser un proyecto largo, era mejor que la implementación fuera fácil de realizar.
- **Tkinter:** Aunque C++ tiene bibliotecas de desarrollo GUI, Tkinter pareció más intuitivo y fácil de usar.

Decisión final: Python y Tkinter.

7.2. Formato de los archivos

A continuación se explicarán las extensiones de los distintos archivos y ficheros que se usan en la aplicación y el formato de los ficheros que contienen la información de la matriz.

7.2.1. Aplicación

La codificación de la aplicación está guardada en un archivo con extensión **.py** indicando que está escrito en Python.

La aplicación como tal se puede ejecutar desde un archivo con extensión **.exe**. Esta clase de archivos son ejecutables, y como su nombre indica, inician una aplicación o programa al ser usados.

7.2.2. Ficheros de matrices

La información de las matrices de Harris está almacenada dentro de ficheros con formato **.csv**. Un fichero CSV (*Comma-Separated Values*) guarda la información de manera tabular, es decir, se guarda en un formato de filas y columnas. Las columnas se separan mediante comas, mientras que las filas se separan mediante saltos de páginas. Cada fila contiene información de un registro, y cada registro tiene uno o más campos separados por comas (columnas). Hay varios programas que permiten guardar información como fichero CSV. En mi caso utilicé **Microsoft**

Excel.

En los ficheros, cada fila representa una unidad estratigráfica distinta, y cada columna representa un dato de la unidad (véase la Figura 7.1: Fichero CSV con formato correcto).

	A	B	C	D	E	F	G
	Nombre	Hijos	Equivalencias	Hecho	Tipo	Fase	Descripción
2	T	5001			N		
3	5001	5003,5002			P		
4	5003	MR5004			P		
5	5002	5007,MR5012			P		
6	MR5004	Unexcavated			P		
7	5007	5010,5011,5014,FS5015			P		
8	MR5012	Unexcavated			P		
9	Unexcavated	G			P		
10	5010	5009			P		
11	5011	5009			P		
12	5014	5009			P		
13	FS5015	5009			N		
14	G				N		
15	5009	5008			P		
16	5008	Unexcavated			P		

Figura 7.1: Fichero CSV con formato correcto.

Cada fichero CSV con información de la matriz debe tener al menos las siguientes columnas:

- **Nombre:** El nombre de la unidad estratigráfica. Dos unidades no pueden tener el mismo nombre.
- **Hijos:** Si la unidad es de tipo positivo o negativo, indica las unidades estratigráficas inmediatamente inferiores. No se pueden crear ciclos entre unidades (A es hijo de B y B es hijo de A).
- **Equivalencias:** Si la unidad es de tipo positivo o negativo, indica las unidades estratigráficas que son equivalentes a ella. Una unidad no puede ser hija de otra y a la vez ser la equivalente a esta.
- **Hecho:** Si la unidad es de tipo hecho, especifica todas las unidades estratigráficas que engloba. No puede haber ciclos entre ellas (A engloba a B y B engloba a A) ni una misma

unidad puede ser englobada por dos unidades tipo hecho distintas (A engloba a C y B engloba a C).

- **Tipo:** El tipo de una unidad puede ser positivo, negativo o hecho y en el fichero se indica con P, N o H respectivamente. Una unidad es positiva si está formada por la presencia o adición de material, como un muro; mientras que es negativa si se refiere a la eliminación de material, como podría ser un agujero en el suelo. Las unidades de tipo hecho agrupan a otras unidades de tipo positivo, negativo o hecho, y permiten que una matriz sea más legible y sencilla de interpretar.
- **Fase:** La fase normalmente se refiere al período histórico del que data la unidad.
- **Descripción:** Un texto que da información adicional sobre la unidad estratigráfica.

7.3. Comprobación de errores

En esta sección se desarrolla más en profundidad lo explicado en Sección 6.3: Prevención de errores. La información cargada al sistema, junto con el filtrado, afecta a las tablas que muestran la información, y estas indican lo que se dibuja en el grafo de nodos y la matriz. Por eso es importante asegurarse que no hay errores ni al cargar información de una matriz de Harris ni al modificar dicha información. En esta sección se procederá a explicar los distintos errores a evitar y por qué es fundamental que no estén.

7.3.1. Errores en los ficheros

El sistema comprueba que el archivo subido no contenga los errores que se van a explicar a continuación:

- **Las columnas principales existen.** Las 7 columnas representan la información mínima que debe tener una unidad estratigráfica o hecho en una matriz de Harris.
- **No hay una columna con el nombre *Codigo*.** Una vez se comprueba que un archivo subido no tiene errores, se añade una nueva columna de nombre *Codigo* que se usará para identificar a las unidades estratigráficas y hechos en las operaciones internas del sistema. Es mejor que no haya duplicados para evitar errores.
- **No hay dos o más columnas con el mismo nombre.** Que haya dos o más columnas del mismo nombre puede provocar errores en el funcionamiento del programa, por lo que es mejor que no haya columnas con nombres repetidos.

	A	B	C	D	E	F	G	H	I
1	Nombre	Hijos	Equivalencias	Hecho	Tipo	Fase	Descripcion	Nombre	Nombre
2	A	B			P			Trabajo	de
3	B				P			Fin de	Grado

Figura 7.2: Fichero csv tres columnas *Nombre*.

Sin embargo, se ha comprobado que cuando hay dos o más columnas con el mismo nombre en el fichero, al ser subidas se cambia el nombre a las columnas repetidas a *columna.X* siendo X el número de duplicado de la columna. Es decir, si hay tres columnas llamadas *Nombre* las dos últimas el programa las nombrará *Nombre.1* y *Nombre.2*. Se puede comprender mejor observando las figuras 7.2 y 7.3. La primera es el archivo descrito anteriormente y la segunda es del fichero cargado en memoria.

Nombre.1	Nombre.2
Trabajo	de
Fin de	Grado

Figura 7.3: Columnas con nombre duplicado en la aplicación.

- **En la columna *Nombre* no hay nombres con coma (,) o punto y coma (;).** En los ficheros *csv* las comas y punto y comas se usan para separar la información interna del fichero. Si bien estos ficheros usan el entrecomillado cuando un valor contiene separadores (por ejemplo, una fila que contiene “A, B”, C, D donde un valor incluye a A y B) sigue siendo preferible que un nombre no tenga separadores porque no es común que un nombre incluya comas o puntos y comas.
- **No hay filas sin nombre.** Todas las unidades estratigráficas y hechos deben tener nombre incluso aunque no se usen.
- **No hay nombres duplicados.** No hay dos unidades estratigráficas, dos hechos o una unidad y un hecho con el mismo nombre.

- **Las unidades en las columnas *Hijos*, *Equivalencias* y *Hecho* existen en la columna *Nombre* si no están vacíos.** No puede existir una relación, equivalencia o agrupación con una unidad que no exista. La única excepción es el valor vacío, ya que pueden existir unidades pero sin relaciones, equivalencias o agrupaciones con otra.
- **No hay valores duplicados en las columnas *Nombre*, *Hijos*, *Equivalencias* o *Hecho* dentro de la misma fila.** Una unidad no puede tener una relación, equivalencia o agrupación consigo misma, de la misma manera que dos unidades no pueden tener más de un tipo distinto de conexión a la vez.
- **No hay ciclos de nodo padre-hijo.** Hay que asegurarse que una unidad no puede ser ancestro y descendiente de otra al mismo tiempo.
- **La equivalencia se indica en ambas unidades.** Si dos unidades son equivalentes, debe indicarse en su respectiva columna *Equivalencia* escribiendo las unidades con las que son equivalentes.
- **No hay valores duplicados dentro de *Hecho*.** Una unidad solo puede estar en una única agrupación dentro de un hecho. Si no, se corre el riesgo de que haya ciclos dentro de los hechos (A dentro de B y B dentro de A).
- **En la columna *Tipo* solo existen los valores *P*, *N* o *H*.** *P* es *Positivo*, *N* es *Negativo* y *H* es *Hecho*. No hay más tipos de nodos.
- **Las unidades positivas (*P*) y negativas (*N*) no pueden tener valores en la columna *Hecho*.** Como no son hechos, no pueden agrupar a otras unidades o hechos.
- **Los hechos (*H*) no pueden tener valores en *Hijos* o *Equivalencias*.** Como no son unidades, no pueden tener ni hijos ni equivalencias.
- **No puede haber hechos en *Hijos* o *Equivalencias*.** Un hecho no puede aparecer ni como hijo ni como equivalencia de una unidad estratigráfica.

No solo al cargar un fichero se comprueba si hay errores, también se verifica al añadir una relación, equivalencia, agrupación, unidad estratigráfica o hecho.

7.3.2. Añadir relaciones

Si se añade un relación, por ejemplo, *A es padre de B* o $A \rightarrow B$ se realiza lo siguiente:

- **Verificar que no se produce un ciclo.** A no puede ser al mismo tiempo ancestro y descendiente de B, porque un estrato no puede estar encima y debajo de otro a la vez.
- **Verificar equivalencias del nodo hijo.** Si se quiere añadir $A \rightarrow B$ y B es equivalente a C ($B = C$) entonces debe cumplirse que A es padre de C ($A \rightarrow C$), porque si B y C son el mismo estrato, A no puede ser el padre de uno y no del otro.

- **Verificar equivalencias del nodo padre.** Si se quiere añadir $A \rightarrow B$ y A es equivalente a D ($A = D$) entonces debe cumplirse que D es padre de B ($D \rightarrow B$), porque si A y D son el mismo estrato, B no puede ser el hijo de uno y no del otro.
- **Comprobar que no existe como relación.** Es innecesario añadir una relación que ya existe.
- **Comprobar que no existe como equivalencia.** Si ya existe como equivalencia, no puede existir también como relación.

7.3.3. Añadir equivalencia

Al añadir una equivalencia, por ejemplo $A = B$, las comprobaciones a hacer son muy parecidas a las comprobaciones al añadir una relación:

- **Verificar que no se produce un ciclo.** Si $A \rightarrow B$ y $B \rightarrow C$, entonces añadir $A = C$ crearía un ciclo: $A \rightarrow B \cap B \rightarrow C \Rightarrow B \rightarrow A$.
- **Verificar padres.** Hay que comprobar que ambas unidades tengan los mismos padres, porque si una tiene un padre que la otra no, entonces no son equivalentes.
- **Verificar hijos.** Hay que comprobar que ambas unidades tengan los mismos hijos, porque si una tiene un hijo que la otra no, entonces no son equivalentes.
- **Comprobar que no existe como relación.** Si ambas unidades ya están en una relación, no pueden estar al mismo tiempo en una equivalencia. Por dar un ejemplo, si $A \rightarrow B \cap A = B \Rightarrow A \rightarrow A$.
- **Comprobar que no existe como equivalencia.** No es necesario añadir una equivalencia que ya existe.

7.3.4. Añadir agrupación

Al añadir una agrupación $B \in A$ se tiene que tener en cuenta lo siguiente:

- **Verificar que no se producen ciclos.** Sobre todo cuando un hecho agrupa a otro hecho. Si un hecho B está agrupado dentro del hecho A, el hecho A no puede estar también agrupado dentro del hecho B.
- **Comprobar que no existe como agrupación.** Evitar que se introduzcan agrupaciones inútiles. Si $A \in B$ y $B \in C$, es redundante incluir $A \in C$.

7.3.5. Añadir o editar una unidad estratigráfica

Tanto al añadir como editar una unidad estratigráfica se comprueba que el nombre no sea ni una cadena vacía ni haya otra unidad o hecho con ese nombre. Si al editar el nodo se cambia el tipo a H , la unidad se transforma en hecho, su información se pasa a la tabla de hechos y todas las relaciones y equivalencias en las que estaba involucrada son eliminadas.

7.3.6. Añadir o editar un hecho

Muy similar al añadir o editar una unidad estratigráfica, se comprueba que el nombre no sea una cadena vacía o duplicado del de una unidad o hecho. Si se cambia el tipo a P o N , el hecho pasa a ser una unidad, su información pasa a la tabla de unidades y todas las agrupaciones en las que participaba se eliminan.

7.4. Diagrama de clases

Un diagrama de clases explica la estructura de un sistema representando las clases, sus atributos y métodos y las relaciones entre estas. La figura 7.4 muestra el diagrama de clases de HAMMON.

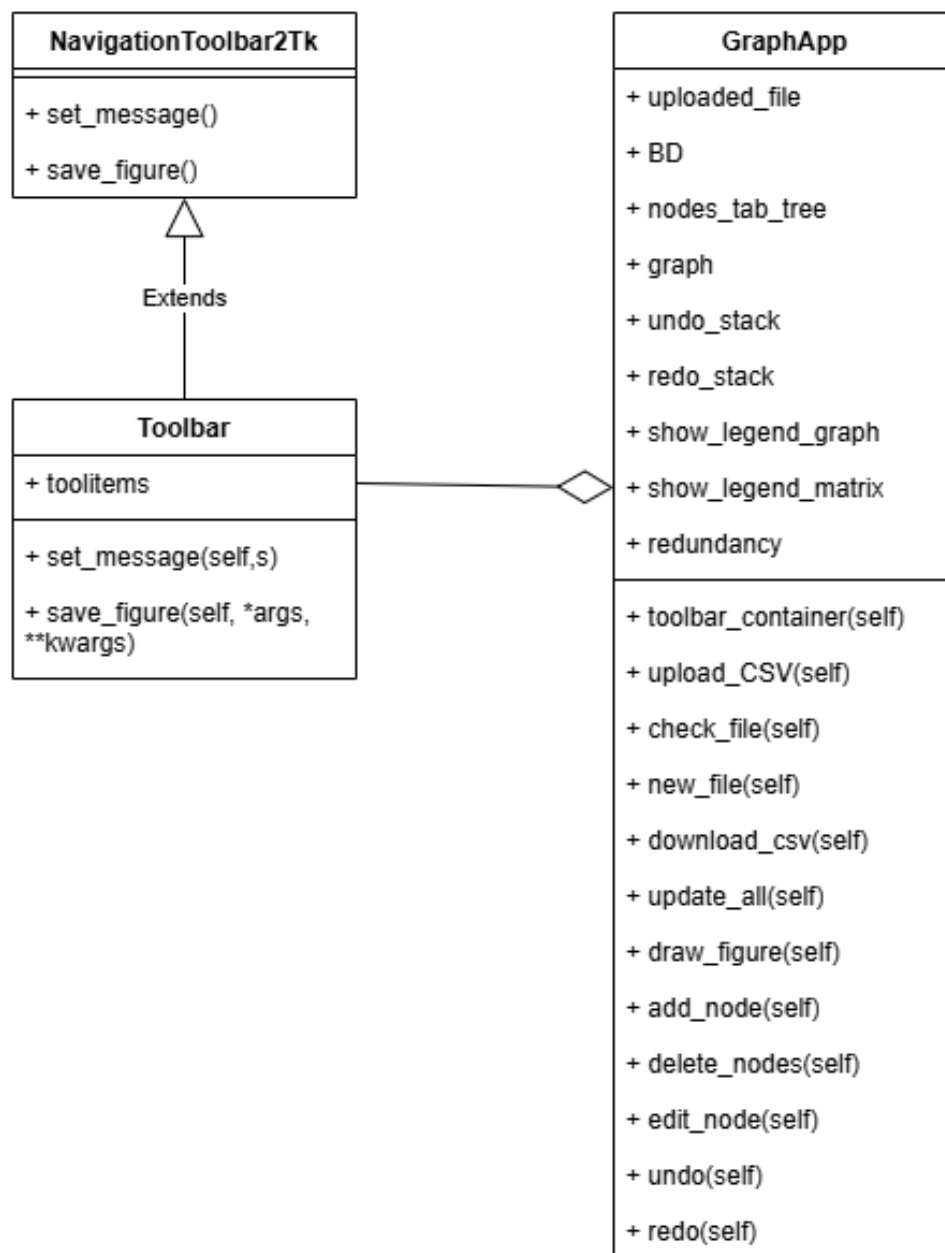


Figura 7.4: Diagrama de las clases de HAMMON.

Capítulo 8

Conclusiones y trabajos futuros

Este es el último capítulo de la memoria antes de pasar a los apéndices y la bibliografía. El Capítulo 8: Conclusiones y trabajos futuros contiene dos partes diferenciadas: en primer lugar, en la Sección 8.1: Conclusiones se explica si se han cumplido los objetivos o no, cómo se han cumplido y qué habilidades se han reforzado o aprendido. En segundo lugar, la Sección 8.2: Trabajos futuros se hablan de las posibles mejoras que se le podrían hacer a la aplicación y por qué son importantes.

8.1. Conclusiones

Tal como se explicó en el Capítulo 2: Objetivos, el objetivo principal del proyecto era crear una aplicación de desarrollo de matrices de Harris alternativa a las ya existentes, especialmente para personas que estuvieran empezando a trabajar con matrices de Harris. En el Apéndice C: Matriz contiene 2 imágenes de HAMMON con dos matrices de Harris dibujadas que corresponden a una matriz de Harris proporcionada por el arqueólogo y por lo tanto usando datos reales que corresponden a la excavación **La Mesa de Fornes (Fornes, Granada) 2023. Dir. ANDRÉS MARÍA ADROHER Y MANUEL ABELLEIRA.**

A continuación repasaremos los distintos objetivos en Capítulo 2: Objetivos y si han podido cumplirse o no, y en caso afirmativo explicar cómo.

- **OBJ-1. Investigar las distintas alternativas que ya existen y analizar sus ventajas y desventajas:** Como se ha visto en el Capítulo 4: Estado del arte se han estudiado 3 aplicaciones distintas: *ArchEd*, *Stratify* y *Harris Matrix Composer*. Si bien se ha podido encontrar información sobre estas, mucha era relativamente antigua y bastante escasa, especialmente en el caso de *ArchEd* y *Stratify*. Cabe mencionar que no fue posible hacer pruebas con estas dos últimas aplicaciones, pues dieron problemas durante el intento de instalación, probablemente debido a la antigüedad que tienen. En el caso de *Harris*

Matrix Composer, sí se encontró más información disponible, y además se pudo probar la aplicación.

- **OBJ-2. Estudiar las distintas alternativas de diseño que se pueden tomar para desarrollar la aplicación:** Todo el contenido del Capítulo 6: Diseño está dedicado a investigar las distintas posibilidades que se podrían haber tomado para desarrollar la aplicación, desde decidir si hacerla web o de escritorio o distintas cuestiones relacionadas con la implementación. Por otra parte el Capítulo 7: Implementación explica las decisiones tomadas sobre el lenguaje de programación utilizado y el formato de los ficheros, que son cuestiones más relacionadas con la implementación de la aplicación.
- **OBJ-3. Otorgar a la aplicación una interfaz lo más sencilla e intuitiva posible para los usuarios con menos experiencia usando aplicaciones:** Como se explica en el Capítulo 6: Diseño, se usó como base la interfaz que tenían otros programas de edición de matrices como *ArchEd* o *Harris Matrix Composer*. La interfaz es simple, fácil de seguir y no tiene demasiadas funcionalidades para que un usuario nuevo no se sienta abrumado. El Apéndice D: Manual de Usuario contiene las instrucciones de la aplicación para mayor comodidad del usuario.
- **OBJ-4. Desarrollar el proyecto, tanto el software como la memoria como código abierto u *open source*, dando más transparencia al trabajo y permitiendo a otras personas usar su contenido libremente:** Se han incluido licencias tanto para el código fuente como para la memoria, que se especifican en el Apéndice A: Licencias. Se ha realizado una investigación para las licencias y finalmente se seleccionaron las que se consideraron apropiadas. Para hacer un resumen rápido, las licencias permiten utilizar los contenidos de la memoria y el código siempre y cuando se cite al autor y se permiten hacer trabajos derivados para fines comerciales y no comerciales.

A lo largo del desarrollo del TRABAJO DE FIN DE GRADO también se han adquirido o reforzado las siguientes capacidades:

- Se ha mejorado la capacidad de planificar proyectos grandes a lo largo de un período de tiempo extenso.
- El tratamiento de bibliografía (búsqueda, clasificación, selección, ...) ha sido reforzado.
- Se ha aprendido a utilizar patrones de diseño que facilitan la implementación del código. A su vez permite que los cambios a la codificación sean mucho más sencillos de realizar.
- Aunque ya se hubiese usado *Python* con anterioridad en asignaturas como Aprendizaje Automático, esta fue la primera vez que se utilizó fuera del ámbito de la Inteligencia Artificial. De hecho, ha sido la primera vez que se ha implementado una aplicación de escritorio con *Python*. De la misma forma, es la primera vez que se usa la biblioteca *Tkinter*.
- Los conocimientos sobre el diseño de interfaces se han visto reforzados.

- Aunque ya se había usado previamente *Latex* para escribir documentos, se han aprendido muchas técnicas y herramientas que anteriormente eran desconocidas.
- Se ha hecho una toma de contacto por primera vez con los distintos tipos de licencias y qué permite cada una.

8.2. Trabajos futuros

Tras varios meses de trabajo, se logró terminar una aplicación de diseño de matrices de Harris completamente funcional, eficiente y con varias funcionalidades interesantes para los usuarios. Desde este punto de partida, hay varias mejoras o añadidos que estarían interesantes considerar.

- **Multilingüe:** Sería interesante explorar la posibilidad de añadir la opción de cambiar el idioma de la aplicación y hacerlo accesible a más usuarios.
- **Expandir la edición estilística:** La aplicación ya cuenta con funciones que cambian los colores de todos los nodos de una fase o que cambian la anchura de las aristas de relación o equivalencia. Incluir más opciones para cambiar el estilo del grafo o añadir opciones estilísticas para la matriz sería una idea atractiva.
- **Mayor cantidad de extensiones disponibles:** Aumentar la gama de extensiones de archivo con las que subir o descargar información del programa resultaría en una aplicación más versátil.
- **Mayor accesibilidad:** Hay usuarios que pueden contar con problemas de visión o que tengan dificultades para leer, por lo que una posible ruta de continuación del trabajo podría ser hacer el programa más accesible.
- **Ventana interactiva:** *Harris Matrix Composer* contaba con una ventana interactiva donde se podían añadir nodos y aristas directamente desde la ventana. Si bien no se pudo implementar por su dificultad y falta de tiempo, no sería descabellado que en un futuro se implemente la capacidad de incluir elementos directamente en el grafo.
- **Multisistema:** Que la aplicación pueda usarse entre distintos sistemas operativos aumenta la accesibilidad de esta.

Apéndice A

Licencias

A.1. Licencia de la memoria

La memoria está protegida bajo el copyright de **Creative Commons**, más específicamente **CC BY-SA 4.0**.

HAMMON: Harris Matrix Manager on desktop © 2025 by Carlos Martín Sánchez is licensed under Creative Commons Attribution-ShareAlike 4.0 International. To view a copy of this license, visit <https://creativecommons.org/licenses/by-sa/4.0/>

A.2. Licencia del código fuente

El código está protegido bajo la licencia **GNU AFFERO GENERAL PUBLIC LICENSE Version 3**, o **GNU AGPL v.3**, en la dirección <https://github.com/CarlosMarSan/HAMMON-Harris-Matrix-Manager-on-desktop.git>.

Apéndice B

Glosario

Glosario

Diagrama de Hasse Tipo de diagrama matemático que representa un conjunto finito parcialmente ordenado aplicando la reducción transitiva. 125

Equivalencia Conexión entre dos unidades estratigráficas que se pueden considerar una sola. 125

Hecho Agrupación de 2 o más unidades estratigráficas. 125

Matriz de Harris Diagrama secuencial que define las relaciones entre las distintas unidades estratigráficas. 125

Relación Conexión entre una unidad estratigráfica (nodo padre) y otra unidad estratigráfica debajo de la primera (nodo hijo). 125

Unidad estratigráfica Conjunto de estratos que se pueden definir e identificar por una serie de propiedades o características. 125

Apéndice C

Matriz

Cabe destacar que se ha utilizado la aplicación en un caso real. En las figuras C.1 y C.3 se muestran extractos de una matriz de Harris proporcionada por el arqueólogo y diseñada en HAMMON. El fichero completo de la matriz se encuentra en el siguiente repositorio <https://github.com/CarlosMarSan/HAMMON-Harris-Matrix-Manager-on-desktop.git> bajo el nombre *Matriz Manuel Completa.csv*. La matriz completa corresponde a la siguiente excavación: **La Mesa de Fornes (Fornes, Granada) 2023. Dir. ANDRÉS MARÍA ADROHER Y MANUEL ABELLEIRA.**

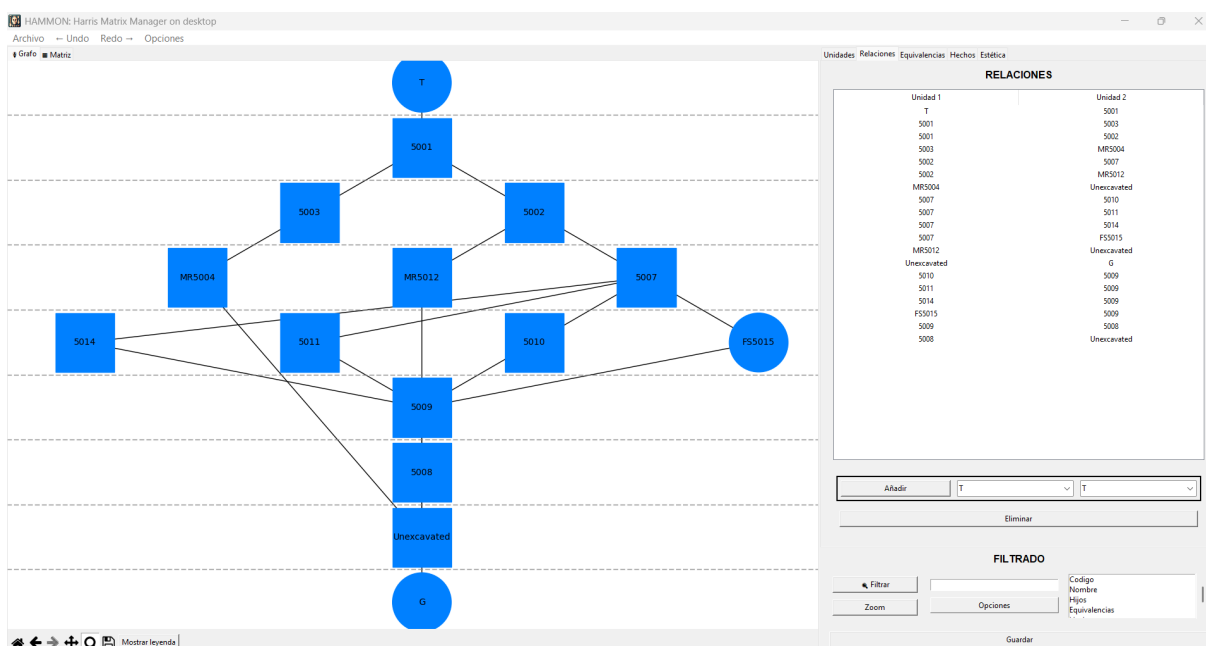


Figura C.1: Extracto de la matriz de ejemplo proporcionada por el arqueólogo 1.

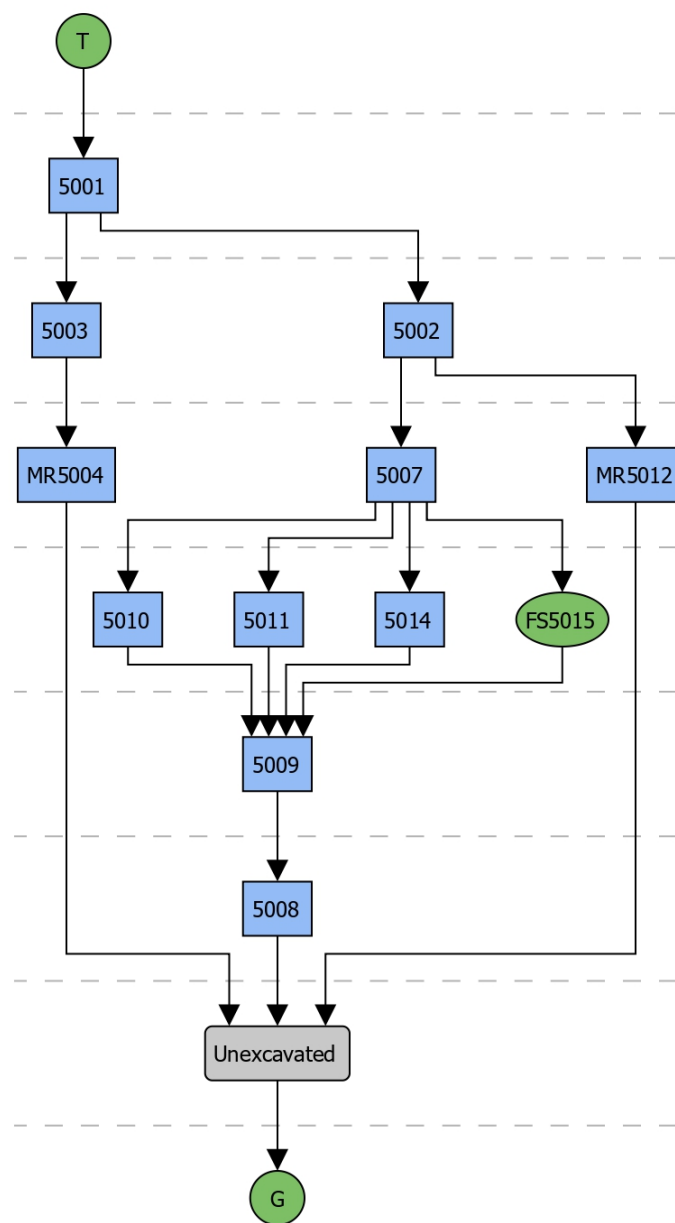


Figura C.2: Submatriz 1. Proporcionada por el arqueólogo.

En las figuras C.2 y C.4 se muestran las submatrices dadas por el arqueólogo. Corresponden con las representaciones de las figuras C.1 y C.3 respectivamente.

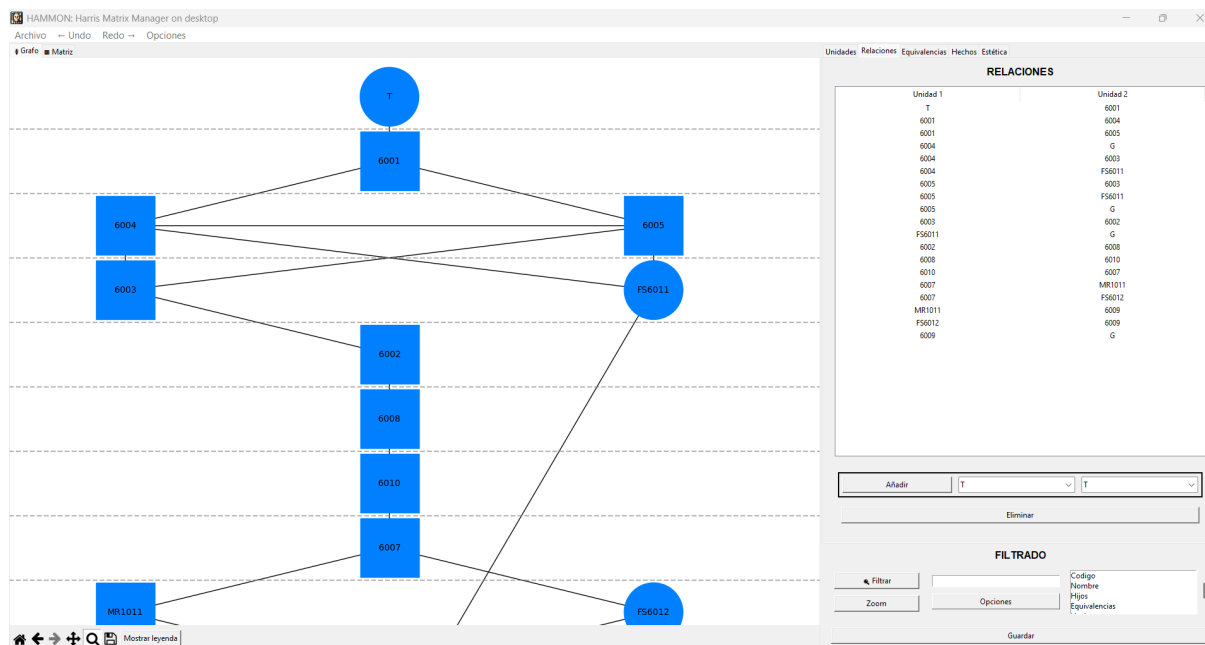


Figura C.3: Extracto de la matriz de ejemplo proporcionada por el arqueólogo 2.

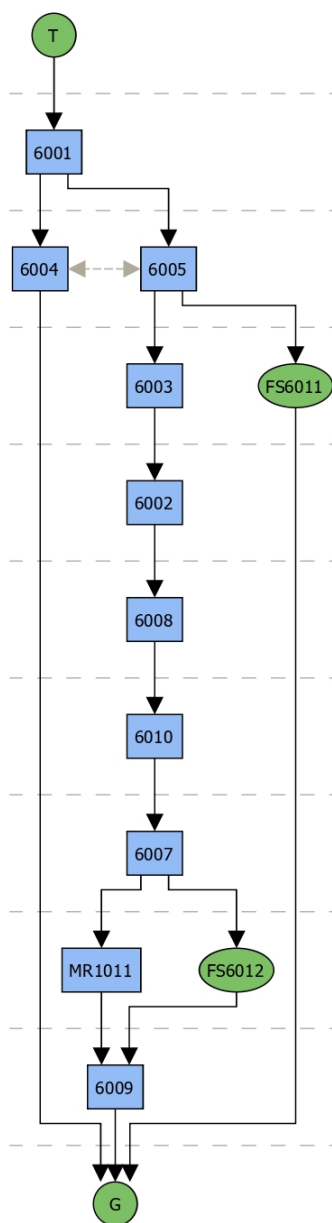


Figura C.4: Submatriz 2. Proporcionada por el arqueólogo.

Apéndice D

Manual de Usuario

D.1. Inicio

La figura D.1 contiene una captura de pantalla que muestra cómo se ve la interfaz cuando se inicia la aplicación.

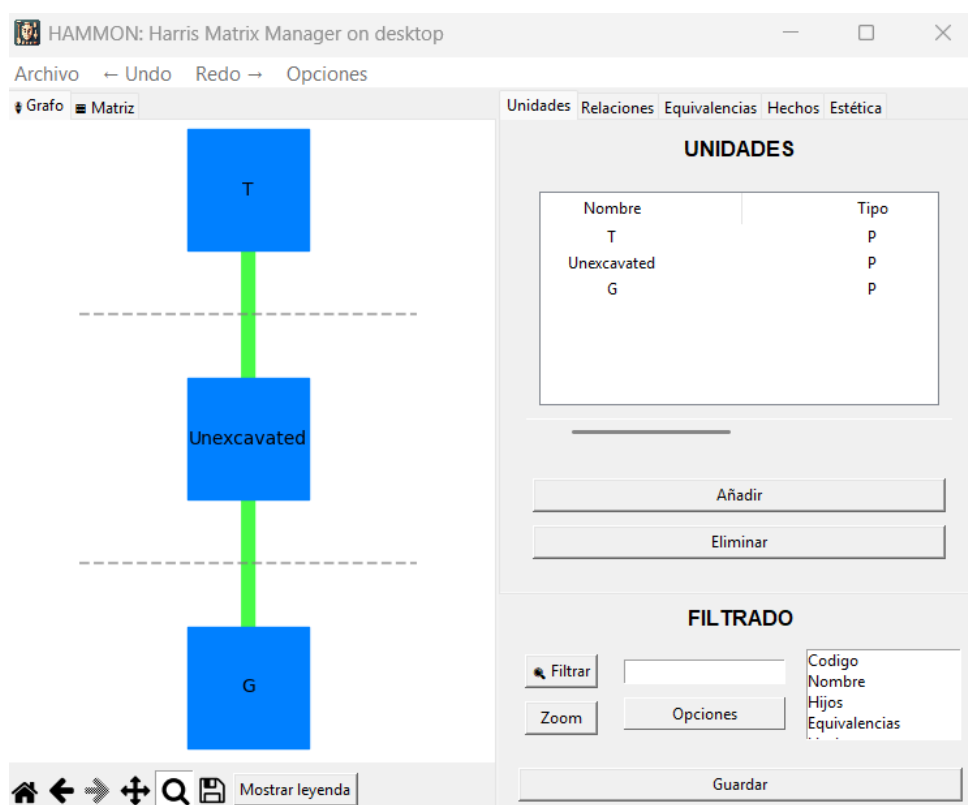


Figura D.1: Interfaz de *HAMMON* al iniciar la aplicación.

D.2. Barra de herramientas

La barra de herramientas contiene las siguientes opciones:

- **Archivo:** Dentro de archivo se encuentran las siguientes opciones:
 - **Nuevo:** Permite limpiar la información de la matriz cargada en la aplicación y cargar la información de la matriz por defecto de la aplicación ($T \rightarrow Unexpected \rightarrow G$). Solicita confirmación al usuario antes de limpiar y cargar la matriz por defecto.
 - **Cargar CSV:** Permite cargar la información de una matriz de Harris con el formato indicado en la Subsección 7.2.2: Ficheros de matrices. El sistema solicita al usuario que indique el fichero a subir con la ventana que se muestra en la figura D.3. Si se detecta algún error, el sistema muestra un mensaje de error por pantalla informando del tipo de error generado y dónde se encuentra. Manda un mensaje con las unidades que no se han dibujado inicialmente si hay alguna unidad sin relaciones ni equivalencias.
 - **Guardar:** Guarda la edición realizada sobre la información de la matriz de Harris siempre y cuando exista un archivo sobre el que guardar. En caso contrario, aparece una ventana de error que incide en que no hay un archivo sobre el que guardar.
 - **Guardar como:** Se crea un nuevo fichero con la información de la matriz actual y solicita al usuario un nombre y una dirección con las que guardar el nuevo fichero.
- **Undo:** Traducido al castellano: deshacer. Opción para devolver la información de la matriz a un estado anterior.
- **Redo:** Traducido al castellano: rehacer o volver a hacer. Opción para que la información de la matriz regrese a una estado posterior.
- **Opciones:** Dentro se encuentra la siguiente función:
 - **Redundancia:** Opción para aplicar o dejar de aplicar la redundancia sobre el dibujo del grafo.

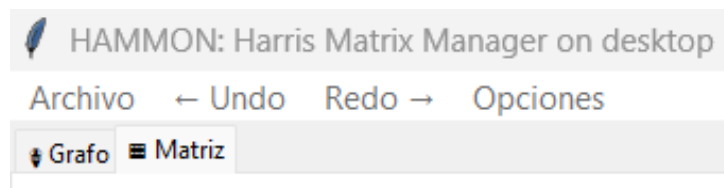


Figura D.2: Barra de herramientas.

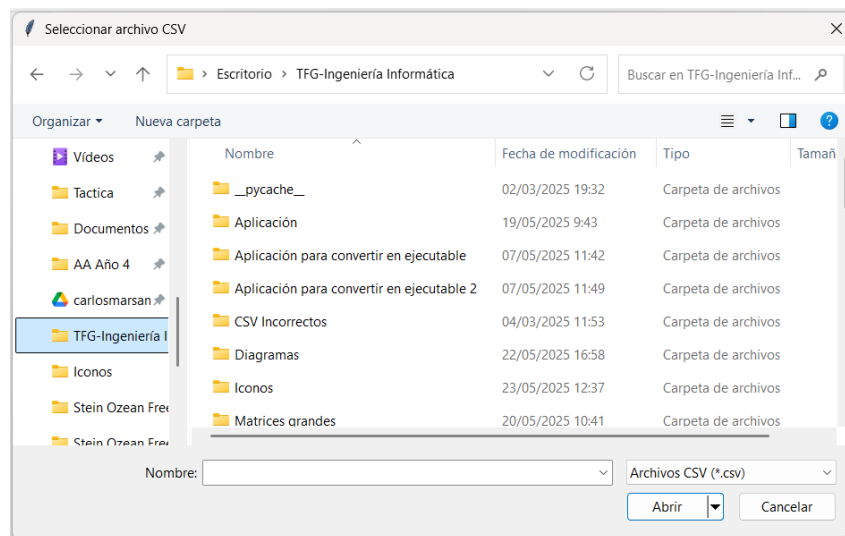


Figura D.3: Ventana con archivos del sistema.

D.3. Ventana de figuras

Se puede observar en la figura D.2 dos pestañas debajo de la barra de herramientas: **Grafo** y **Matriz**. Pulsar sobre cada una de ellas te lleva a la pestaña donde se encuentra la ventana de la figura del grafo y la matriz respectivamente.

D.4. Barra de herramientas de vista

La figura D.4 muestra la barra de herramientas para la vista de las figuras.



Figura D.4: Barra de herramientas de vista.

Esta barra habilita la manipulación de la vista de las figuras, tanto del grafo como de la matriz. Las distintas herramientas con las que cuenta la tabla son:

- **Resetear a la vista original:** Al ser seleccionada, la vista actual de la figura pasa a ser la vista original antes de haber aplicado cambios.
- **Vista anterior:** La vista actual de la figura cambia a la vista anterior.

- **Vista posterior:** La vista actual de la figura cambia a la vista posterior.
- **Moverse con el botón izquierdo del ratón:** Al seleccionar esta opción, permite mover la vista de la figura sobre los ejes X e Y hacia la posición que el usuario indique. Para ello, tiene que presionar sobre un punto de la ventana y arrastrar hacia la dirección contraria a la que pretende desplazar la vista.
- **Zoom con rectángulo:** Cuando se selecciona esta opción, el usuario puede pulsar sobre un punto de la ventana y arrastrar el cursor hacia otra dirección, formando un rectángulo sobre la vista. Cuando suelte el botón, se aplicará zoom sobre el área de la vista dentro del rectángulo. Si el botón usado es el izquierdo, se aplica zoom in, si se usa el botón derecho, zoom out.
- **Guardar la figura:** Guarda una imagen de la vista actual. Cuando se pulsa sobre este botón, el sistema solicita al usuario que introduzca un alto y ancho para la imagen en un rango de [100,5000], y un nombre, dirección y extensión con la que guardar la imagen.
- **Mostrar leyenda:** Un botón que al pulsarlo muestra la leyenda de la figura si no se muestra, o viceversa. Puede verse un grafo con leyenda en la figura D.5.

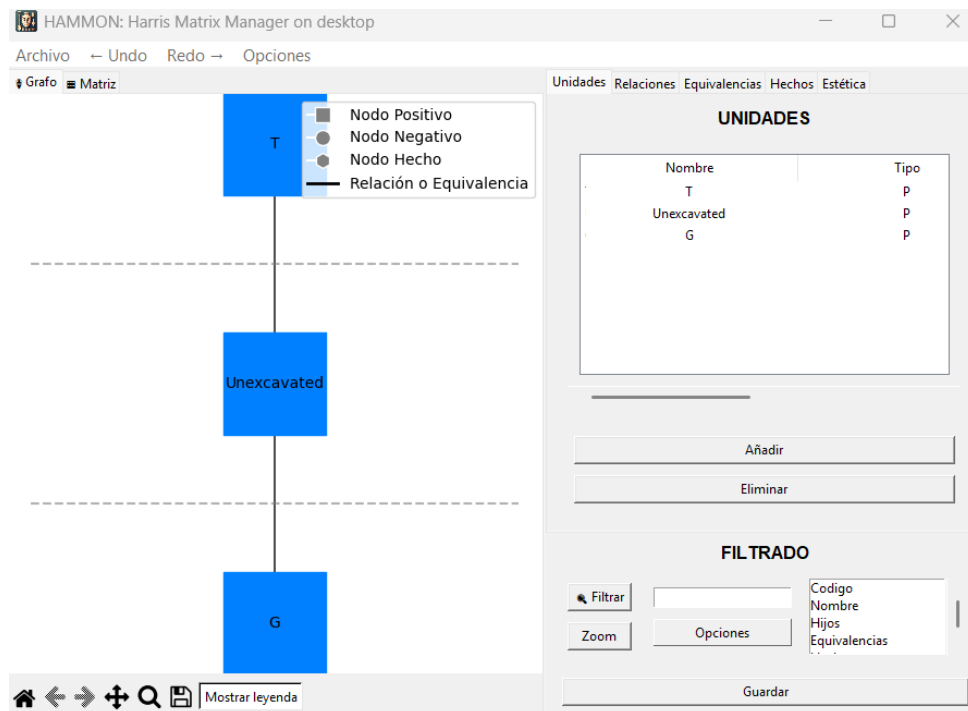
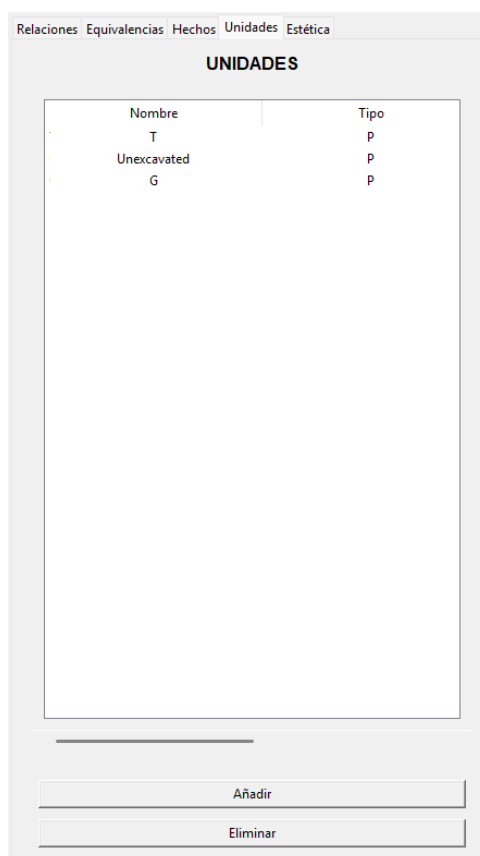


Figura D.5: HAMMON: Matriz con leyenda.

D.5. Unidades

En la tabla se muestra la información de todas las unidades positivas y negativas dentro de la información de la matriz y que cumplen las condiciones de filtrado (véase Sección D.10: Filtrado). No se muestra información de las relaciones o las equivalencias, como se puede ver en la figura D.6.



Nombre	Tipo
T	P
Unexcavated	P
G	P

Añadir

Eliminar

Figura D.6: Pestaña de unidades.

D.5.1. Añadir unidades

Al pulsar el botón *Añadir* se muestra una ventana (véase la Figura D.7: Ventana para añadir unidad) donde hay que introducir los datos de la unidad nueva. Cabe aclarar que el nombre no puede ser el mismo que otra unidad ya existente ni puede ser una cadena vacía. Debe presionarse el botón *Añadir* para confirmar la adición.

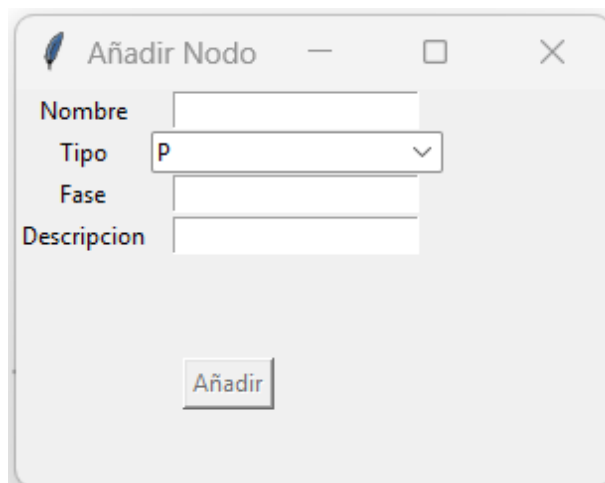
A screenshot of a software window titled "Añadir Nodo" (Add Node). The window has a standard title bar with a feather icon, a minus button, a maximize button, and a close button. Inside the window, there are four labeled input fields: "Nombre" (Name), "Tipo" (Type) which is a dropdown menu currently showing "P", "Fase" (Phase), and "Descripcion" (Description). Below these fields is a button labeled "Añadir" (Add).

Figura D.7: Ventana para añadir unidad.

D.5.2. Eliminar unidades

Se selecciona una o varias filas de la tabla (manteniendo pulsado el botón *ctrl*) que corresponden con las unidades y se pulsa *Eliminar*. La aplicación pedirá una confirmación final al usuario antes de eliminar definitivamente la(s) unidad(es).

D.5.3. Editar unidad

Se puede editar la información interna de una unidad haciendo doble clic sobre una fila de la tabla de unidades. Se mostrará una pestaña parecida a la de añadir unidad (véase la Figura D.7: Ventana para añadir unidad). El nombre cumple las mismas restricciones: no puede ser el mismo que el de otra unidad en la matriz ni puede ser cadena vacía.

Si el tipo se cambia a *H*, esa unidad estratigráfica se convertirá en hecho y pasará a la tabla de hechos (véase Subsección D.8.1: Información interna). Si se edita el tipo, todas las relaciones y equivalencias con dicho nodo son eliminadas. Para guardar los cambios, debe presionarse el botón *Guardar*.

D.6. Relaciones

En la pestaña **RELACIONES** se encuentra una tabla que muestra todas las relaciones que están en la información interna de la matriz y que cumplen los criterios de filtro (véase Sección D.10: Filtrado). La figura D.8 contiene una captura de la pestaña de relaciones.

Relaciones Equivalencias Hechos Unidades Estética

RELACIONES

Unidad 1	Unidad 2
T	Unexcavated
Unexcavated	G

Añadir T T Eliminar

Figura D.8: Pestaña de relaciones.

D.6.1. Añadir relación

Al lado del botón *Añadir* hay dos desplegados donde ambos contienen todos los nodos que representan las unidades estratigráficas positivas y negativas. El usuario solo tiene que seleccionar dos nodos y pulsar *Añadir*. Si la introducción de la relación nueva no produce errores, se edita la información de la matriz para incluir la relación, cambiar el contenido de la tabla de relaciones y redibujar las figuras.

D.6.2. Eliminar relación

Para eliminar una relación, hay que seleccionar una fila de la tabla de relaciones y pulsar el botón *Eliminar*. Se puede seleccionar más de una fila presionando *Ctrl* mientras se seleccionan filas para eliminar dos o más relaciones en una sola operación.

D.7. Equivalencias

Las funcionalidades y la manera de utilizarlas son idénticas a lo descrito en la Sección D.6: Relaciones. La figura D.9 contiene una captura de la pestaña de equivalencias.

La imagen muestra una interfaz de usuario con una barra de pestañas superior que incluye: Relaciones, Equivalencias, Hechos, Unidades y Estética. La pestaña 'Equivalencias' está seleccionada y muestra un título 'EQUIVALENCIAS'. Debajo del título, hay una tabla con dos columnas: 'Unidad 1' y 'Unidad 2'. El cuerpo de la tabla está vacío. En la parte inferior de la interfaz, hay una barra de acciones que contiene un botón 'Añadir', dos menús desplegables (ambos con la letra 'T' seleccionada) y un botón 'Eliminar'.

Figura D.9: Pestaña de equivalencias.

D.8. Hechos

La pestaña de hechos cuenta con un cuaderno con dos pestañas: **Información interna** y **Agrupamiento**.

D.8.1. Información interna

Contiene una tabla con la información interna de los hechos, que se representan como nodos tipo *H*. También cuenta con un botón de *Añadir* y *Eliminar*. El funcionamiento de esta pestaña es idéntico al de la pestaña de unidades (véase Sección D.5: Unidades), es decir, tiene las

funcionalidades de añadir, eliminar y editar hechos. Respecto a esta última funcionalidad, si se cambia el tipo H por el tipo P o N , el hecho se convierte en una unidad estratigráfica y pasa a la tabla de la pestaña de unidades. Si eso ocurre, todos los agrupamientos con el nodo editado son eliminados. La figura D.10 contiene una captura de la pestaña de información interna.

La imagen muestra una interfaz de usuario con una barra de pestañas superior que incluye: Relaciones, Equivalencias, Hechos (seleccionada), Unidades y Estética. Debajo de esta barra, hay una sub-barra con 'Información Interna' y 'Agrupamiento'. El contenido principal de la pestaña 'Información Interna' es un formulario con el título 'INFORMACIÓN INTERNA'. Este formulario contiene una tabla con dos columnas: 'Nombre' y 'Tipo'. La tabla está actualmente vacía. Debajo de la tabla, hay una barra horizontal y dos botones: 'Añadir' y 'Eliminar'.

Figura D.10: Pestaña de hechos: Información interna.

D.8.2. Agrupamiento

Un agrupamiento consta de dos partes: el hecho y la unidad. Se interpreta como que el hecho agrupa a la unidad. Las funciones de añadir y eliminar agrupamientos son idénticas a las de añadir y eliminar relaciones respectivamente (véase Sección D.6: Relaciones). La única diferencia al añadir un agrupamiento es que en el primer desplegable se muestran todos los hechos y en el segundo se muestran todas las unidades estratigráficas. La figura D.11 contiene una captura de la pestaña de agrupamientos.

Figura D.11: Pestaña de hechos: Agrupamiento.

D.8.3. Aplicar hecho

En la pestañas de agrupamientos se pueden indicar los hechos que se desea mostrar en el grafo de nodos. Hay que introducir una lista con los hechos que se desea aplicar, por ejemplo, *hecho1*, *hecho2*, *hecho3*... y pulsar el botón *Aplicar*. Se cambia el contenido de las tablas de relaciones y equivalencias, más concretamente, si una unidad está dentro de un agrupamiento, se sustituye por el hecho que más la engloba; y se vuelven a dibujar las figuras en consecuencia. Solo se tienen en cuenta los agrupamientos cuyos hechos coinciden con los introducidos. Cabe mencionar que al introducir un hecho a aplicar tienen que coincidir mayúsculas y minúsculas, la palabra completa y los diacríticos.

D.9. Estética

En la pestaña de **Edición estética** se pueden cambiar elementos estéticos del grafo, como el color de los nodos, el color de las aristas de relaciones y equivalencias, etc. La figura D.12 contiene una captura de la pestaña de edición estética.

Relaciones Equivalencias Hechos Unidades Estética

EDICIÓN ESTÉTICA

RELACIONES

Color

Línea solid

Flecha -

Anchura 1

EQUIVALENCIAS

Color

Línea solid

Flecha -

Anchura 1

FASES

Fase	Color
------	-------

Figura D.12: Pestaña de hechos: Edición estética.

Se puede comprobar que hay 3 zonas: **RELACIONES**, **EQUIVALENCIAS** y **FASES**. Las dos primeras funcionan exactamente igual, siendo la única diferencia que la primera se aplica a las aristas que representan relaciones y la segunda se refiere a las aristas que representan equivalencias. Para evitar la redundancia, agruparemos las herramientas de ambas como "herramientas de aristas". La zona de **FASES** contiene una tabla que muestra qué color corresponde a cada fase.

D.9.1. Cambiar color de las aristas

Al hacer clic sobre el botón *Color* aparecerá en pantalla la ventana que aparece en la figura D.13.

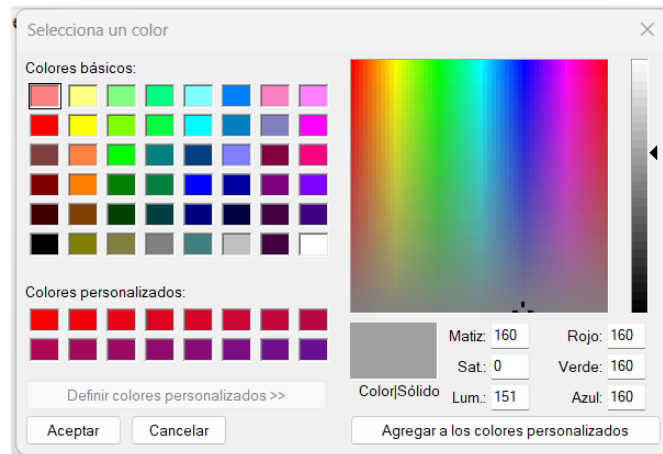


Figura D.13: Ventana de selección de color.

El usuario puede escoger entre los colores predefinidos o definir un nuevo color manualmente. Se debe hacer clic sobre *Aceptar* para aplicar el color nuevo.

D.9.2. Cambiar la forma de la línea de la arista

El desplegable a la derecha de *Línea* indica la forma en la que se está dibujando la línea de la arista. Por defecto está en *solid*, pero hay más opciones como *dotted*, que hace que la línea de la arista sea una sucesión de puntos separados entre sí. Para cambiar la línea, solo hay que pulsar sobre el desplegable y hacer clic sobre la nueva opción que se desea aplicar.

D.9.3. Cambiar la forma de la flecha de la arista

Al lado de *Flecha* hay un desplegable. Funciona de manera similar a cambiar la forma de la línea, pero esta vez se aplica sobre la forma de la flecha de la arista. La flecha refiere a la apariencia que toman los extremos de la arista. Por defecto se representa como -, lo que quiere decir que en los extremos de cada arista no se dibuja una flecha, sino que es una línea recta sin más.

D.9.4. Cambiar anchura de la arista

El deslizable al lado de *Anchura* indica la anchura actual de las aristas. Para cambiar el grosor solo hay que mover el deslizable hacia el valor de anchura que se desee, dentro de un rango de enteros de $[1, 8]$.

D.9.5. Cambiar el color de una fase

En la zona de **FASES** hay una tabla con una columna para las fases y otra para el color. Cada fila indica el color asociado a cada fase. Para cambiar el color de una fase, solo hay que hacer doble clic sobre la fila donde se encuentra la fase cuyo color queramos cambiar y aparecerá la ventana representada en la figura D.13. En la ventana selecciona un color predefinido o define uno si no está en los predefinidos y pulsa *Aceptar* para cambiar el color.

D.10. Filtrado

La sección de filtrado dentro de la aplicación se muestra en la figura D.14.



Figura D.14: Sección de filtrado.

D.10.1. Filtrar

Esta función permite filtrar los nodos que aparecen en las tablas de unidades y hechos, y en consecuencia los que aparecen en las tablas de relaciones, equivalencias y agrupamientos. Es decir, si una unidad o hecho no se muestra en su respectiva tabla, no aparecerá en el resto de tablas. No quiere decir que se haya borrado de la información de la matriz cargada en la aplicación, para eso ya existen funcionalidades como la explicada en la Subsección D.6.2: Eliminar relación.

Para poder utilizarla correctamente, se escribe en la entrada de texto sobre *Opciones* una lista de valores a usar, por ejemplo, *valor1, valor2, valor3,* Haz clic sobre *Filtrar* y solo permanecerán las unidades o hechos que dentro de su información interna se encuentre alguno de los valores escritos. Se puede hacer un filtrado más complejo con elementos que se explicarán en la Subsección D.10.3: Opciones y la Subsección D.10.4: Selección de columnas.

D.10.2. Zoom

La función de *Zoom* utiliza la misma entrada de texto que *Filtrar*, y también se ve afectada por el desplegable de *Opciones* (véase Subsección D.10.3: Opciones) y la lista de columnas

(véase Subsección D.10.4: Selección de columnas). Cuando se hace clic sobre *Zoom*, la ventana de la figura cambia para que se muestren todos los nodos que coinciden con el filtrado usando el tamaño mínimo de ventana.

Hay que aclarar que el zoom de la barra de herramientas en vista (véase Sección D.4: Barra de herramientas de vista) y este zoom aunque tienen funciones similares o casi idénticas funcionan de distinta manera, por lo que es importante no confundirlos.

D.10.3. Opciones

Un desplegable con las siguientes opciones:

- **Coincidir mayúsculas y minúsculas:** Al comparar los valores introducidos con la información interna, deben coincidir las mayúsculas y las minúsculas. Por ejemplo, el valor introducido es para, y en una columna de la información interna una unidad tiene el valor Para. No pasaría el filtro porque para \neq Para (No coincide la P mayúscula con la p minúscula).
- **Coincidir palabras completas:** Al comparar los valores introducidos con la información interna, deben coincidir palabras completas. Siguiendo el ejemplo anterior, si en la información interna un nodo contiene el valor Mampara, no pasaría el filtro porque para \neq Mampara (son palabras distintas). Tampoco pasaría para Elisa por ejemplo, porque aunque contenga para, no coincide el valor completo: para \neq para Elisa.
- **Coincidir diacríticos:** Al comparar los valores introducidos con la información interna, deben coincidir diacríticos. La RAE define a los diacríticos así: "Son diacríticos los signos ortográficos que inciden sobre un grafema para indicar algún rasgo o valor distintivo. En español, los únicos signos diacríticos vigentes son la tilde (´) y la diéresis (¨)...". Volviendo al ejemplo anterior, un nodo que contiene en su información interna pära no pasaría el filtro porque para \neq pära.

Al principio ninguna está activa y no se aplican ni al filtrado ni al zoom. Para activar una hay que hacer clic sobre *Opciones* y seleccionar la que se desee activar. Se pueden desactivar las opciones de la misma forma. Varias opciones pueden estar activas a la vez. El filtrado vuelve a aplicarse tras cambiar el estado de una opción.

D.10.4. Selección de columnas

Si se desea aplicar el filtrado o el zoom sobre columnas concretas, se pueden seleccionar dichas columnas en la lista de columnas en el extremo derecho de la sección de filtrado. En un inicio, no hay ninguna seleccionada. Si no se selecciona ninguna columna a la hora de usar el filtrado o el zoom, estos se aplican a todas las columnas.

D.10.5. Guardar

Es una opción parecida a *Guardar* y *Guardar como* (véase Sección D.2: Barra de herramientas). Funciona exactamente igual que *Guardar como*, con la diferencia que guarda un fichero solo con las unidades y los hechos que han pasado el filtrado, no con toda la información interna. Por ejemplo, si un fichero cuenta con las unidades *A*, *B*, *C* y *D* y tras aplicar el filtrado no aparece *B* en la lista de unidades, el fichero que se guarde tendrá solo la información interna de las otras 3 unidades, y si alguna de estas últimas está en una relación, equivalencia o agrupamiento con *B*, no aparecerá dicha información en el fichero.

De la misma forma que con *Guardar como*, solicitará al usuario un nombre y dirección con el que guardar la información.

D.10.6. Bibliotecas

El código utiliza varias bibliotecas que es necesario tener instaladas para que el programa funcione.

Bibliotecas de la biblioteca estándar de Python

No requieren instalación si la versión de Python es 3.6 o más.

- **tkinter**: En Linux o Ubuntu puede requerir instalación mediante

```
$ sudo apt-get install python3-tk
```

- **collections**
- **re**
- **unicodedata**
- **time**
- **os**
- **sys**

Bibliotecas externas

Estas bibliotecas necesitan instalarse antes de poder ser usadas. Estas son:

- **matplotlib**

- **networkx**
- **numpy**
- **pandas**

Se pueden instalar mediante el siguiente comando:

```
$ pip install matplotlib networkx numpy pandas
```

Compilación

El comando para compilar el programa es:

```
$ pyinstaller --clean --onefile --windowed --icon="StoneMaskIcon.ico"  
--add-data "StoneMaskIcon.png;." HAMMON.py
```

Los archivos *StoneMaskIcon.ico* y *StoneMaskIcon.png* deben estar en la misma dirección que el archivo de la aplicación *HAMMON.py* o la compilación no será exitosa.

También hay que tener instalado *pyinstaller*. Para instalar:

```
$ pip install pyinstaller
```

Se puede comprobar si ya está instalado con:

```
$ pyinstaller --version
```

Bibliografía

- [AGU72] A. V. Aho, M. R. Garey y J. D. Ullman. «The Transitive Reduction of a Directed Graph». En: *SIAM Journal on Computing* 1.2 (1972), págs. 131-137. DOI: 10.1137/0201008. eprint: <https://doi.org/10.1137/0201008>. URL: <https://doi.org/10.1137/0201008>.
- [Bal22] MO Balogun. «Comparative analysis of complexity of C++ and Python programming languages». En: *Asian J. Soc. Sci. Manag. Technol* 4.2022 (2022), págs. 1-12. URL: <https://www.ajssmt.com/Papers/420112.pdf>.
- [BDV09] Michal Bližňák, Tomáš Dulík y Vladimír Vasek. «A persistent cross-platform class objects container for C++ and wxWidgets». En: *WSEAS Transactions on Computers* 8 (ene. de 2009). URL: https://www.researchgate.net/publication/228740954_A_persistent_cross-platform_class_objects_container_for_C_and_wxWidgets.
- [BE+16] Douglas Beniz, Alexey Espindola et al. «Using Tkinter of python to create graphical user interface (GUI) for scripts in LNLS». En: *WEPOPRPO25* 9 (2016), págs. 25-28. URL: <https://accelconf.web.cern.ch/pcapac2016/papers/wepoprpo25.pdf>.
- [BS03] Kurt Bittner y Ian Spence. *Use case modeling*. Addison-Wesley Professional, 2003. URL: https://books.google.es/books/about/Use_Case_Modeling.html?id=zvxfXvEcQjUC&redir_esc=y.
- [BS08] Jasmin Blanchette y Mark Summerfield. *C++ GUI programming with Qt 4*. Prentice Hall Professional, ene. de 2008. ISBN: 9780137143979. URL: https://www.researchgate.net/profile/Jasmin-Blanchette/publication/234788146_C_GUI_Programming_with_Qt_4_Second_Edition/links/0a85e53668db46ecba000000/C-GUI-Programming-with-Qt-4-Second-Edition.pdf.
- [Caq91] André Caquot. «P. Xella. Baal Hammon. Recherches sur l'identité et l'histoire d'un dieu phénico-punique». En: *Revue de l'histoire des religions* 208.4 (1991), págs. 442-444. URL: https://www.persee.fr/doc/rhr_0035-1423_1991_num_208_4_1654.

- [CHA75] KI HONG CHANG. «Unconformity-bounded stratigraphic units». En: *GSA Bulletin* 86.11 (nov. de 1975), págs. 1544-1552. ISSN: 0016-7606. DOI: 10.1130/0016-7606(1975)86<1544:USU>2.0.CO;2. eprint: <https://pubs.geoscienceworld.org/gsa/gsabulletin/article-pdf/86/11/1544/3443719/i0016-7606-86-11-1544.pdf>. URL: [https://doi.org/10.1130/0016-7606\(1975\)86%3C1544:USU%3E2.0.CO;2](https://doi.org/10.1130/0016-7606(1975)86%3C1544:USU%3E2.0.CO;2).
- [Dor81] George T Doran. «There's a SMART way to write managements's goals and objectives». En: *Management review* 70.11 (1981). URL: <https://research.ebsco.com/c/jdsrkr/search/details/wddtf7ti55?db=bsu>.
- [ES12] David Eppstein y Joseph A Simons. «Confluent hasse diagrams». En: *Graph Drawing: 19th International Symposium, GD 2011, Eindhoven, The Netherlands, September 21-23, 2011, Revised Selected Papers 19*. Springer. 2012, págs. 2-13. URL: https://link.springer.com/chapter/10.1007/978-3-642-25878-7_2.
- [Hed58] Hollis D. Hedberg. «Stratigraphic Classification and Terminology1». En: *AAPG Bulletin* 42.8 (ago. de 1958), págs. 1881-1896. ISSN: 0149-1423. DOI: 10.1306/0BDA5B85-16BD-11D7-8645000102C1865D. eprint: https://pubs.geoscienceworld.org/aapg/aapgbull/article-pdf/42/8/1881/4388672/aapg_1958_0042_0008_1881.pdf. URL: <https://doi.org/10.1306/0BDA5B85-16BD-11D7-8645000102C1865D>.
- [HG17] Edward Harris y Robert G Gunn. «The use of Harris matrices in rock art research». En: (2017). URL: https://www.researchgate.net/publication/324834187_The_use_of_Harris_Matrices_in_rock_art_research.
- [Hun+98] Christoph Hundack et al. «ArchE: A Graph Drawing System for Archaeology». En: *Graph Drawing* (abr. de 1998). DOI: 10.1007/3-540-63938-1_72.
- [Ikr+15] Muhammad Ikram et al. «Testing from UML Design using Activity Diagram: A Comparison of Techniques». En: *International Journal of Computer Applications* 131 (dic. de 2015), págs. 975-8887. DOI: 10.5120/ijca2015907354.
- [Lun99] Fredrik Lundh. «An introduction to tkinter». En: URL: www.pythonware.com/library/tkinter/introduction/index.htm 539 (1999), pág. 540. URL: <https://ftp.math.utah.edu/u/ma/hohn/linux/tcl/an-introduction-to-tkinter.pdf>.
- [Pod19] Primož Podrzaj. «A brief demonstration of some Python GUI libraries». En: *Proceedings of the 8th International Conference on Informatics and Applications ICIA2019*. 2019, págs. 1-6. URL: https://www.researchgate.net/profile/Natalie-Walker-15/publication/335570218_Proceedings_of_The_8th_International_Conference_on_Informatics_and_Applications_ICIA2019_Japan_2019/links/5d6e072e4585150886098062/Proceedings-of-The-8th-International-Conference-on-Informatics-and-Applications-ICIA2019-Japan-2019.pdf#page=3.
- [Pop02] Paul Pop. «Comparing web applications with desktop applications: An empirical study». En: (2002). URL: https://backend.orbit.dtu.dk/ws/portalfiles/portal/3715402/pop_hci.pdf.

- [Pre03] Lutz Prechelt. «Are scripting languages any good? A validation of Perl, Python, Rexx, and Tcl against C, C++, and Java.» En: *Adv. Comput.* 57 (2003), págs. 205-270. URL: [https://books.google.es/books?hl=es&lr=&id=xqvv7yqtr2kC&oi=fnd&pg=PA205&dq=Lutz+Prechelt.+\(\(Are+scripting+languages+any+good%3F+A+validation+of+Perl,+Python,+Rexx,+and+Tcl+against+C,+C%2B%2B,+and+Java.\)\)\)+En:+Adv.+Comput.+57+\(2003\),+p%C2%B4ags.+205-270.&ots=9PQ8HWIy68&sig=Bh9smQYIZELeqeuj3unkB-u9NeQ&redir_esc=y#v=onepage&q&f=false](https://books.google.es/books?hl=es&lr=&id=xqvv7yqtr2kC&oi=fnd&pg=PA205&dq=Lutz+Prechelt.+((Are+scripting+languages+any+good%3F+A+validation+of+Perl,+Python,+Rexx,+and+Tcl+against+C,+C%2B%2B,+and+Java.)))+En:+Adv.+Comput.+57+(2003),+p%C2%B4ags.+205-270.&ots=9PQ8HWIy68&sig=Bh9smQYIZELeqeuj3unkB-u9NeQ&redir_esc=y#v=onepage&q&f=false).
- [Sen21] Udes S. Senarath. «Waterfall Methodology, Prototyping and Agile Development». En: *Tech. Rep.* (jun. de 2021). DOI: 10.13140/RG.2.2.17918.72001. URL: https://www.researchgate.net/profile/Udes-S-Senarath/publication/353324450_Waterfall_Methodology_Prototyping_and_Agile_Development/links/60f41f71fb568a7098b9d035/Waterfall-Methodology-Prototyping-and-Agile-Development.pdf.
- [Str86] Bjarne Stroustrup. «An overview of C++». En: *SIGPLAN Not.* 21.10 (jun. de 1986), págs. 7-18. ISSN: 0362-1340. DOI: 10.1145/323648.323736. URL: <https://doi.org/10.1145/323648.323736>.
- [TNI08] Christoph Traxler, Wolfgang Neubauer y A Ica. «The Harris Matrix Composer - A New Tool to Manage Archaeological Stratigraphy». En: *Archäologie und Computer-Kulturelles Erbe und Neue Technologien-Workshop*. Vol. 13. Ene. de 2008. URL: https://www.researchgate.net/profile/Wolfgang-Neubauer/publication/228356749_The_Harris_Matrix_Composer_-_A_New_Tool_to_Manage_Archaeological_Stratigraphy/links/00b49531f4cc05009f000000/The-Harris-Matrix-Composer-A-New-Tool-to-Manage-Archaeological-Stratigraphy.pdf.

