

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 10



**QTOCTAVE: UNA REINGENIERÍA POR UN DESARROLLO
GENUINO**

**Trabajo de Diploma en opción al Título de Ingeniero
en Ciencias Informáticas**

Autores:

Daniel Enrique Pérez del Toro

Luis Manuel Teijón Acosta

Tutor:

Msc. Alexeis Companioni Guerra

Ciudad de la Habana

Junio 2008

*"Nosotros necesitamos una manera sustancialmente distinta de pensar para que la humanidad pueda sobrevivir". **Albert Einstein***

*"No conozco la clave del éxito, pero sé que la clave del fracaso es tratar de complacer a todo el mundo". **Woody Allen***

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firman la presente a los _____ días del mes de _____ de 2008.

Daniel Enrique Pérez del Toro

Luis Manuel Teijón Acosta

Msc. Alexeis Companioni Guerra

Agradecimientos

LA culminación del presente trabajo, no hubiese sido posible sin la contribución de un grupo de personas, amigos todos, que brindaron parte de su tiempo en pos de auxiliarnos en nuestro afán de confeccionar, lo que constituye el punto cumbre de nuestra vida de estudiante. Nos gustaría infinitamente mencionarlos a todos, pero fueron incontables los compañeros que de una forma u otra aportaron a nuestra labor, ya fuera con sus conocimientos o su apoyo moral. No quisiéramos dejar pasar por alto la ocasión, para hacerles llegar nuestro más sincero agradecimiento. La más profunda y especial gratitud a nuestros padres, por habernos guiado por el camino correcto y ser nuestro sostén, nuestra inspiración, nuestra vida. Agradecimientos especiales también a nuestro tutor y amigo y a Clara Gisela Scot Bigñott que han constituido pilares importantes en la materialización de este sueño, quienes con su abnegación, esmero y paciencia ha sido un ejemplo para nosotros. Gracias a Maikel Manuel Fernández, a Yaima Bety Suárez, a Bertha Elena Romero y Adrian Maranje Agramonte por su ayuda y su apoyo. A Pedro L. Lucas por sus aclaraciones y correcciones. En fin, para no olvidar a nadie. . .

A todos, muchas gracias.

Dedicatoria

Daniel Enrique Pérez del Toro:

A mis padres, mi universo, mi razón de ser...

A mis familiares, por su apoyo y su preocupación.

A Yanet, mi aliento, mi inspiración, mi vida...

A mis hermanos, a mis amigos...

Luis Manuel Teijón Acosta:

A mis queridísimos padres, a quienes les debo la vida, y que con mucho amor hicieron posible este sueño.

A mis abuelos manolito, mina, nena, papa y mami Sara que aunque no está hoy con nosotros, siempre confió en mi: por eso nunca dudé en dedicarle este trabajo.

A todos, todos mis familiares...

A Yeleny, por sus consejos, su apoyo, su paciencia, su cariño.

A mis amigos...

Resumen

DE los egipcios a nuestros días, el uso de las matemáticas ha constituido un pilar imprescindible del desarrollo de la ciencia y las tecnologías. La paulatina evolución de los más primitivos métodos de cálculo ha hecho posible contar con un número elevado de asistentes matemáticos de esencia simbólica o numérica, de vital importancia en el ámbito docente, ingenieril y científico. La privatización de estos medios de cálculo modernos ha traído aparejada la búsqueda de alternativas por parte de la comunidad internacional del software libre, desarrollándose proyectos bajo licencias como la GPL que permiten la libre distribución y modificación de los mismos, en aras de alcanzar un mayor progreso y eliminar de cierto modo la brecha digital que ahonda el software privado. Octave, magnífico software de cálculo numérico y sustituto por excelencia de su homólogo propietario de mayor reconocimiento internacional Matlab, aspira marchar en la vanguardia de este tipo de asistentes, pero la carencia de un entorno de desarrollo integrado es la principal desventaja frente a su contrincante propietario. QtOctave, genial frontend surgido al calor de esta búsqueda de soluciones libres, promete impregnarle a Octave toda la integridad y facilidad de uso para colocarlo en el lugar que le corresponde, pero su carencia de documentación del diseño, es una limitante para conseguir tal fin. Confeccionar una documentación del diseño de dicho frontend, que facilite el desarrollo de versiones posteriores, cada vez más funcionales, más eficientes, versátiles y adaptables, es el motivo esencial de este trabajo. Para lo cual se realiza un estudio del origen y evolución de los asistentes matemáticos, de la situación actual en torno a los mismos, así como un levantamiento de las características de QtOctave unido al estudio ingenieril de éste. De manera adicional, se describen los casos de uso, las clases del sistema y se confeccionan los diagramas de diseño y colaboración correspondientes.

INDICE

Introducción	1
1. Fundamentación teórica	8
1.1. Conceptos fundamentales	8
1.1.1. Ingeniería de software	8
1.1.2. Software libre	9
1.1.3. Software privativo	9
1.1.4. Licencia GPL	9
1.1.5. Biblioteca	10
1.1.6. API	10
1.1.7. Bibliotecas de desarrollo Qt	10
1.1.8. Biblioteca estándar de plantillas (STL)	10
1.2. Asistentes matemáticos para el cálculo numérico	10
1.3. Cálculo numérico y cálculo simbólico	11
1.3.1. Cálculo numérico	11
1.3.2. Cálculo simbólico	11
1.3.3. Diferencia entre el cálculo numérico y el cálculo simbólico	11
1.3.4. El asistente Matlab	12
1.3.4.1. Matlab, software privativo	13
1.3.4.2. Precio actual de Matlab	13
1.3.4.3. Matlab en las universidades	14
1.3.5. Scilab	15

1.3.5.1.	Incompatibilidad entre Scilab y Matlab	16
1.3.6.	Octave	17
1.3.6.1.	Octave, sustituto potencial de Matlab	19
1.4.	Frontends	20
1.4.1.	Características del entorno gráfico de Matlab	21
1.4.1.1.	El entorno de trabajo de Matlab	21
1.4.1.2.	Estructuración del entorno de desarrollo de Matlab:	21
1.4.2.	Frontends para Octave	22
1.4.2.1.	KOctave	22
1.4.2.2.	GOctave	23
1.4.2.3.	Octavede	23
1.4.2.4.	Yaog	23
1.4.2.5.	Octave Workshop	24
1.4.2.6.	QtOctave	24
2.	Características del sistema	26
2.1.	Metodología de desarrollo del software	26
2.1.1.	Proceso unificado de desarrollo de software (RUP)	27
2.1.2.	Extreme Programming (XP)	27
2.2.	Reingeniería de software	28
2.3.	Modelo de dominio	29
2.3.1.	Breve descripción de las clases implicadas	30
2.4.	Requerimientos del sistema	33
2.4.1.	Requisitos funcionales	33
2.4.2.	Requisitos de interface	36
2.4.3.	Requisitos de operación	36
2.4.4.	Requisitos de recursos	36
2.4.5.	Requisitos de verificación	36
2.4.6.	Requisitos de transportabilidad	37

2.4.7.	Requisitos de calidad	37
2.4.8.	Requisitos de fiabilidad	37
2.4.9.	Requisitos de mantenibilidad	37
2.5.	Definición de los casos de uso	38
2.5.1.	Definición de los actores	38
2.5.2.	Diagrama de casos de uso del sistema	39
2.6.	Descripción de los casos de uso del sistema	40
3.	Diseño del sistema	83
3.1.	Diagrama de clases del diseño	83
3.2.	Diagrama de paquetes	85
3.3.	Diagramas de secuencia especificados por casos de uso	86
3.4.	Descripción de las clases	87
	Conclusiones	114
	Recomendaciones	115
	Referencias	118
	Bibliografía	119
	Anexos	121
Anexo 1:	Especificación del diagrama de clases del diseño por paquetes	121
Anexo 2:	Diagramas de secuencia del diseño especificados por paquetes	130
Anexo 3:	Deficiencias fundamentales encontradas durante la reingeniería	149

ÍNDICE DE FIGURAS

2.1. Modelo de dominio.	30
2.2. Diagrama de casos de uso.	39
3.1. Diagrama de paquetes.	85
3.2. Gestionar Proyectos.	121
3.3. Gestionar Comandos.	122
3.4. Gestionar Matrices.	123
3.5. Operaciones Matemáticas.	124
3.6. Resolver Ecuaciones.	125
3.7. Gestionar Graficación.	126
3.8. Gestionar Configuración.	127
3.9. Gestionar Menús.	128
3.10. Gestionar Editor de Scripts.	129
3.11. Crear Proyectos.	130
3.12. Modificar Proyectos.	130
3.13. Eliminar Proyectos.	131
3.14. Ayuda Dinámica.	131
3.15. Autocompletar Comandos.	132
3.16. Crear Matriz.	132
3.17. Determinante.	133
3.18. Autovalores y Autovectores.	133
3.19. Inversa.	134
3.20. Transpuesta.	134

3.21. Submatriz.	135
3.22. Calcular Media.	135
3.23. Calcular Mediana.	136
3.24. Calcular Desviación Estandar.	136
3.25. Calcular Covarianza.	137
3.26. Calcular Coeficiente de Relación.	137
3.27. Histograma.	138
3.28. Resolver Ecuación por el Método de Bisección.	138
3.29. Ecuación Lineal.	139
3.30. Ecuación no Lineal.	139
3.31. Plotear.	140
3.32. Plotear en Sistemas de Coordenadas Polares.	140
3.33. Escalar Logarítmicamente los Ejes x e y de Forma Simultánea.	141
3.34. Escalar de Forma Logarítmica el Eje x.	141
3.35. Escalar de Forma Logarítmica el Eje y.	142
3.36. Mostrar Barra de Errores.	142
3.37. Mostrar Gráfica de Barra.	143
3.38. Plotear Gráficos Tridimensionales.	143
3.39. Plotear Gráficos de Contornos de Superficies Tridimensionales.	144
3.40. Establecer Límites de Gráficas en los Ejes Coordenados.	144
3.41. Nombrar y Etiquetar Gráfica.	145
3.42. Exportar ESP.	145
3.43. Exportar JPG.	146
3.44. Exportar PDF.	146
3.45. Gestionar Configuración.	147
3.46. Gestionar Menús.	147
3.47. Gestionar Editor de Scripts.	148
3.48. Cargar Archivos *.m.	148

ÍNDICE DE TABLAS

1.1. Comandos específicos en Matlab, Octave y Scilab para el trabajo con gráficos y polinomios.	17
2.1. Listado de los actores del sistema.	38
2.2. Caso de uso Gestionar Proyecto.	41
2.3. Caso de uso Gestionar Comandos.	43
2.4. Caso de uso Gestionar Matrices.	46
2.5. Caso de uso Operar Matrices.	51
2.6. Caso de uso Manejar Estadísticas.	57
2.7. Caso de uso Resolver Ecuaciones.	61
2.8. Caso de uso Graficar 2D.	70
2.9. Caso de uso Graficar 3D.	72
2.10. Caso de uso Manipular Gráfica.	76
2.11. Caso de uso Gestionar Configuración.	77
2.12. Caso de uso Gestionar Menú.	80
2.13. Caso de uso Gestionar Editor de Scripts.	82
3.1. Descripción de la clase Terminal.	88
3.2. Descripción de la clase Operations.	93
3.3. Descripción de la clase OctaveConnection.	95
3.4. Descripción de la clase MainWindow.	99
3.5. Descripción de la clase Main.	101
3.6. Descripción de la clase GenerateMenu.	103

3.7. Descripción de la clase MenuCallBack.	104
3.8. Descripción de la clase MenuExtCallBack.	106
3.9. Descripción de la clase Table.	109
3.10. Descripción de la clase Autocomplete.	111
3.11. Descripción de la clase Projects.	113

Introducción

EL uso de las matemáticas ha jugado, a través de la historia, un papel fundamental en el avance socio-económico del ser humano. Numerosos hallazgos evidencian que esta ciencia remonta sus orígenes a más de 20 000 años.[1]

Grandes civilizaciones antiguas alcanzaron un desarrollo considerable en la práctica de esta disciplina, primitiva en sus inicios y que fue evolucionando según las necesidades imperantes. Los egipcios son considerados como los primeros en alcanzar cierto progreso en el campo matemático, con el llamado "sistema de numeración jeroglífico"; los babilonios, por su parte, utilizaron un eficaz sistema de notación fraccionaria y en la India se concibió el trascendental sistema de numeración de base 10. Aunque fueron notorias las contribuciones de dichas culturas prehelénicas, es un hecho que ninguna aportó tanto como la Grecia Antigua, donde, prominentes sabios, como Tales de Mileto [2], Arquímedes de Siracusa [3] y Euclides de Megara [4] crearon una obra colosal, invisible y única, sustentada por leyes y teorías, cuya grandeza perdura hasta nuestros días y es conocida como Matemáticas.

El hombre extendió el uso de las matemáticas a todas las esferas de la vida, debido a su gran utilidad en la resolución de problemas de toda índole. Las operaciones de cálculo fueron ganando en complejidad paulatinamente, al punto que fue necesario crear mecanismos que ayudaran a resolverlas, ya que los sistemas existentes como el ábaco [5] y el suwanpan [6], poseían una capacidad de cálculo limitada.

Muchos fueron los intentos de crear una máquina que resolviera tales inconvenientes; en 1617 John Napier ¹ crea lo que más tarde se le conociera como las tablas de Napier, en honor a su nombre, basada en los conceptos y propiedades de los logaritmos. Hacia 1620 fue concebida una regla de cálculo fija, que brindaba facilidades para hacer operaciones con logaritmos, por Edmund Gunter ², la cual fue perfeccionada y extendida en 1630 por William Oughtred ³, quien le agregó una barra deslizante y la llevó a escalas circulares [7]. Por su parte, en 1623, Wilhelm Schickard ⁴ construyó la primera calculadora automática, llamada “Reloj Calculador” [8]. Unos 20 años después, en 1643, Blaise Pascal ⁵ ideó un dispositivo de cálculo más tarde conocido como Pascalina, con el fin de ayudar a su padre que era el encargado del cobro de impuestos en Ruán, Normandía [9]. En años posteriores esta máquina fue usada para el cálculo de impuestos en Francia hasta 1799.

La complejidad del mundo continuaba en ascenso y el último cuarto del siglo XIX igualmente presenció importantes avances en esta rama. Dorr E. Felt ⁶, inventó en 1884 el comptómetro [10], la primera máquina que operada por teclas permitía sumar y calcular (a diferencia de los diseños anteriores, que exigían manipular palancas separadas).

El incesante perfeccionamiento de las calculadoras existentes, propugnó un creciente avance de éstas en la primera mitad del siglo XX, período en el que alcanzaron su cenit con la invención de la máquina sumadora-listadora de Dalton, presentada en 1902, que fue la primera de su tipo en usar sólo diez teclas [11] y la calculadora miniatura Curta [12], presentada por Curt Herzstark ⁷, en 1948. Estos dos acontecimientos marcaron hito en el desarrollo de

¹Matemático escocés (Edimburgo, 1550 - † 4 de abril de 1617).

²Ilustre matemático inglés (1581 in Hertfordshire, Inglaterra - † 10 de diciembre de 1626 London, Inglaterra).

³Eximio científico inglés (5 de marzo de 1574 Eton, Buckinghamshire - † 30 de junio de 1660 Albury, Surrey).

⁴Científico alemán (22 de abril de 1592 Herrenberg - † 24 de octubre de 1635 Tübingen).

⁵Matemático, físico, filósofo y teólogo francés (19 de junio de 1623 - † 19 de agosto de 1662).

⁶Matemático y científico estadounidense (18 de marzo de 1862 Rock County, Wisconsin - † 7 de agosto de 1930).

⁷Prominente científico austriaco (26 de julio de 1902 Vienna - † 27 de octubre de 1988).

los calculadores y determinaron un completo dominio del mercado por parte de estos artefactos mecánicos, hasta que fueron desplazados definitivamente en la década del '70 por sus similares electrónicos.

Con la aparición algunos años antes de las válvulas de vacío, y posteriormente los transistores, se fomentó la invención de calculadoras electrónicas, en un principio grandes y pesadas, pero que con el paso del tiempo y la concepción de los circuitos integrados fueron reduciéndose hasta convertirse en pequeños procesadores de bolsillo, capaces de hacer numerosas operaciones como resolver simbólicamente ecuaciones cuadráticas, integrar funciones, correr procesadores de texto y derivar.

El auge creciente de las tecnologías y con éste, el desarrollo de la informática, trajo consigo la necesidad de un incremento en la producción del software de cálculo simbólico, gráfico y numérico (ver epig. 1.3) para realizar, apoyar o ilustrar problemas de disímiles campos de la ciencia.

La idea de utilizar computadores para estos fines es relativamente antigua, pero el desarrollo de ordenadores y programas que permitan hacerlo es bastante reciente. No sólo se dispone en estos momentos de programas de cálculo numérico, procesadores de texto, programación lógica o inteligencia artificial, sino que se ha ampliado este campo a programas que realizan lo que se conoce como manipulación algebraica, cálculo simbólico, cálculo numérico, algoritmos algebraicos o álgebra computacional.

Hasta hace poco tiempo, la relación entre la Informática y las Matemáticas se vio limitada a una serie de aplicaciones, en el sentido de considerar los computadores como una herramienta de cálculo de gran velocidad, sujeta a los errores típicos de truncamiento y redondeo; no obstante, la rápida evolución de estos medios, condujo a la aparición de ordenadores muy potentes, y con estos, una inmensa gama de aplicaciones capaces de solucionar disímiles funciones y tareas que ayudan a la resolución de problemas matemáticos, físicos, químicos, ingenieriles y científicos.

En la actualidad un número importante de asignaturas de múltiples carreras se imparten con la ayuda de asistentes matemáticos ya sean de esencia simbólica o numérica. En el caso particular de estos últimos el más conocido y empleado internacionalmente es Matlab, dada la cantidad de prestaciones que posee y el nivel de integración, extensibilidad y facilidad de uso que hoy incorpora. Si bien son innegables estos atractivos en dicha herramienta de elevado poder de cálculo, no es menos cierto que la misma posee precios estratosféricos que la hacen completamente inasequibles para universidades y/o centros de investigación o producción fundamentalmente del tercer mundo.

Este factor, unido a todas las limitantes que imponen las licencias privativas a las que están sujetos un sinnúmero de herramientas y sistemas operativos (SO), son sobradas razones para pensar en una migración al software libre (SWL), como una opción que disipará los inconvenientes que implica la privatización del software, y permitirá la búsqueda de aplicaciones homólogas libres que faciliten la realización de trabajos similares con buena calidad.

En el caso particular de Cuba, dadas las consecuencias del bloqueo impuesto, se prohíbe la adquisición de productos informáticos necesarios para el Ministerio de Educación Superior (MES) y por ende para la Universidad de las Ciencias Informáticas (UCI), por lo que es perfectamente entendible la necesidad imperiosa de sustituir SOs y herramientas de software de licencias propietarias, por similares de licencias libres, aunque en algunos casos esto signifique el sacrificio de determinadas prestaciones hoy disponibles en las primeras.

Polémicas como esta, a nivel mundial, han constituido un motor impulsor del desarrollo de aplicaciones libres para el cálculo moderno como Scilab y Octave, que brindan un conjunto de facilidades similares a su homólogo propietario más cercano y que están al alcance de un mayor número de usuarios. Vale destacar en el caso de este último, que no surgió como un reemplazo de Matlab sino que ha ido convergiendo por pura necesidad y que se ha convertido en una alternativa real al mismo.

Hoy la comunidad internacional trabaja afanosamente en esta dirección, pero la carencia de un ambiente de desarrollo integrado para Octave que imprima una mayor didáctica de trabajo, es la principal limitante y desventaja frente a Matlab, por lo que no pocos han sido los intentos de proveer al mismo de un frontend (ver epig. 1.4) que elimine tal inconveniente y lo coloque ciertamente en el lugar que se merece.

Se han desarrollado proyectos como Koctave, Joctave, Goctave, YAOG, Emacs, Octivate y Octave Workshop QtOctave los cuales presentan características similares. En el caso particular de este último se aportan algunas novedades como la gestión de la ayuda, los menús de comandos y la posibilidad de ver las matrices como en hojas de cálculo, que lo hacen sobresalir entre sus semejantes.

Es la Universidad de las Ciencias Informáticas, a nivel nacional, un factor estratégico en lo que a migración al software libre se refiere y de manera especial la facultad 10, sobre la cual cae el peso y responsabilidad de buscar soluciones a las trabas que impone el software privado. En este contexto, el desarrollo de versiones posteriores de QtOctave significa un aporte importante al sistema de estudios, a la calidad de la enseñanza en la UCI y al movimiento del software libre; no obstante, la carencia de una documentación del diseño para dicho frontend constituye un inconveniente para lograr este propósito, por lo que se plantea en el marco de este trabajo como problema a resolver: La inexistencia de una documentación del diseño para el frontend QtOctave.

Contar con una documentación del diseño del antes mencionado frontend, favorecerá el desarrollo de versiones más versátiles y adaptables a las necesidades de la UCI, para lo cual, sustentándose en el proceso de desarrollo de software así como en las fases de inicio y elaboración, se propone como objetivo principal: Proveer al QtOctave de una documentación del diseño que favorezca el desarrollo de versiones superiores.

En consecuencia con este fin, durante el desarrollo del trabajo se dará cumplimiento a una serie de aspectos más específicos, que permiten, una vez concluido este estudio, contar con

documento que respalda el motivo esencial de la presente investigación. Entre estos objetivos pueden citarse:

1. Realizar un estudio detallado de los asistentes matemáticos numéricos libres sobre el SO Linux y en particular del asistente Octave.
2. Realizar un estudio comparativo entre el asistente matemático Octave y su similar propietario de mayor reconocimiento internacional (Matlab).
3. Realizar un estudio comparativo de los frontends disponibles para Octave sobre SO Linux.
4. Confeccionar el modelo del dominio correspondiente al frontend QtOctave.
5. Definir los requisitos funcionales y no funcionales del sistema.
6. Realizar el diseño correspondiente al frontend QtOctave.

El presente documento se estructura en introducción, tres capítulos y un grupo de anexos, donde se expone y da cumplimiento de forma progresiva y elocuente a la totalidad de los objetivos propuestos:

- Capítulo I. Fundamentación teórica:

En este capítulo se enumeran una serie de conceptos que se utilizarán en el trabajo. Se expone además, un análisis de las principales tecnologías en torno al cálculo numérico y la caracterización de algunos de los software matemáticos más representativos, especialmente el Octave. Se realiza un estudio sobre los frontends más notables creados para este asistente y de forma concluyente se converge a optar por QtOctave como frontend rector a nivel mundial gracias a sus potencialidades.

- Capítulo II. Características del sistema:

En este capítulo se definen las características de QtOctave, así como la fundamentación de la metodología de desarrollo que se utilizará. Se confecciona el modelo de dominio

del sistema, se realiza el levantamiento de los requisitos del mismo y se respalda de forma concreta el por qué de la confección de un modelo de dominio. Finaliza el capítulo con la especificación de los casos de uso del sistema.

- Capítulo III. Análisis y diseño del sistema:

En este capítulo se trata lo relacionado con el diseño del software, dígase modelo de diseño especificado por paquetes, unido a los diagramas de secuencia especificados por casos de uso. Se describen de además las clases que intervienen en los mismos.

Capítulo 1

Fundamentación teórica

EL presente capítulo recoge una serie de conceptos importantes que serán utilizados a lo largo del documento y que, de no ser aclarados, podrían obstaculizar la comprensión del presente texto. Se presenta además, una panorámica sobre las principales tecnologías en torno al cálculo numérico, así como una caracterización desde varios puntos de vista, de algunos de los software matemáticos más notables, especialmente el Octave. Por último, se realiza un estudio sobre los frontends más representativos diseñados para este asistente y se exponen un conjunto de conclusiones parciales que enmarcan el desarrollo del presente trabajo.

1.1. Conceptos fundamentales

1.1.1. Ingeniería de software

- *Definición 1:*

Ingeniería de Software es el estudio de los principios y metodologías para desarrollo y mantenimiento de sistemas de software [13].

- *Definición 2:*

Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida

para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como desarrollo de software o producción de software [13].

1.1.2. Software libre

Programa desarrollado y distribuido según la filosofía de dar al usuario las siguientes libertades:

- Libertad 0: la libertad para ejecutar el programa sea cual sea su propósito.
- Libertad 1: la libertad de para estudiar el funcionamiento del programa y adaptarlo a sus necesidades el acceso al código fuente es condición indispensable para esto.
- Libertad 2: la libertad para redistribuir copias.
- Libertad 3: la libertad de mejorar el programa y luego publicarlo para bien de toda la comunidad el acceso al código fuente es condición indispensable para esto [14].

1.1.3. Software privativo

Se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo con o sin modificaciones. El software privativo es también llamado software no libre, software propietario, software privado o software con propietario [14].

1.1.4. Licencia GPL

Licencia de regulación de los derechos de autor de los programas de software libre. Esta licencia, promovida por la Free Software Foundation (FSF) en el marco de la iniciativa GNU¹, permite la distribución de copias de programas, e incluso cobrar por ello, así como la modificación del código fuente de los mismos o su utilización en otros programas [15].

¹Proyecto nacido en 1984 para desarrollar un sistema operativo similar a UNIX, pero bajo el concepto de software libre. En sí, GNU es un acrónimo recursivo que significa "GNU No es Unix".

1.1.5. Biblioteca

En computación, se refiere a un conjunto de rutinas o subprogramas que realizan operaciones comúnmente requeridas por programas. Se les puede encontrar como un fichero aparte, compartida por todos los programas que la utilizan, o físicamente adicionadas a dichos programas[16].

1.1.6. API

Interfaz de Programación de Aplicaciones: Es una interfaz para servicios de sistemas o librerías de software. Las APIs pueden consistir en un conjunto de clases, llamadas a funciones, llamadas de subrutinas, etiquetas descriptivas, etc [17].

1.1.7. Bibliotecas de desarrollo Qt

Qt es un framework² de fácil uso, multiplataforma y con una interfaz GUI³. El popular entorno KDE⁴ para sistemas operativos basados en Unix, utiliza las librerías Qt [18].

1.1.8. Biblioteca estándar de plantillas (STL)

La Standard Template Library (STL) es una colección de estructuras de datos genéricas y algoritmos escritos en C++ .

1.2. Asistentes matemáticos para el cálculo numérico

El software matemático es aquel software que se utiliza para realizar, apoyar o ilustrar problemas matemáticos; entre este tipo de software tenemos motores de álgebra computacional, cálculo numérico y graficadores de funciones, por sólo citar algunos.

²Conjunto de APIs y herramientas destinadas a la construcción de un determinado tipo de aplicaciones de manera generalista.

³Del inglés Graphics User Interface.

⁴Del inglés K Desktop Environment.

1.3. Cálculo numérico y cálculo simbólico

1.3.1. Cálculo numérico

El cálculo numérico o análisis numérico, como también se le conoce, es una rama de las matemáticas cuyos límites no son del todo precisos. Se podría definir como la disciplina que se encarga de describir, analizar y crear algoritmos numéricos que permitan resolver problemas matemáticos fundamentalmente, en los que estén involucradas cantidades numéricas finitas, con un grado de precisión determinado.

1.3.2. Cálculo simbólico

El cálculo simbólico es una rama de las matemáticas que opera no solo números, también realiza operaciones con expresiones, que pueden estar formadas por variables ($x+y+z$) o por números como ($1/3, \text{Pi}, \dots$). En particular, se puede operar con matrices. y realizar cálculos exactos.

1.3.3. Diferencia entre el cálculo numérico y el cálculo simbólico

La principal diferencia es que el cálculo numérico es aproximado; hay que truncar los números para que quepan en el ordenador, por lo que se van cometiendo pequeños errores de redondeo que van acumulándose. Generalmente, los sistemas de cálculo numérico están preparados para corregir estos errores en los casos más sencillos, pero no pueden corregirlos siempre; sin embargo, los cálculos simbólicos son exactos; el resultado es único y no hay posibilidad de error. Los sistemas de computación matemática complejos suelen incorporar la facilidad de realizar cálculos de los dos tipos, tanto simbólicos como numéricos, mientras que los simples se limitan solo al cálculo numérico.

1.3.4. El asistente Matlab

Se trata de un software matemático muy versátil, basado en el cálculo con matrices para computación numérica, que ofrece un Entorno de Desarrollo Integrado (IDE) con un lenguaje de programación propio e interpretado que permite realizar tareas computacionales de una forma relativamente sencilla y desarrollar aplicaciones de una forma más rápida, incluso más veloz que con el uso de lenguajes tradicionales como C, C++, Fortran o cualquier otro tipo de lenguaje compilado sin recolector de basura integrado. Este magnífico asistente debe su nombre a las abreviaturas de las palabras inglesas MATrix LABoratory (laboratorio de matrices) y actualmente se encuentra disponible para las plataformas Unix, Windows y Apple Mac OS X.

Matlab fue diseñado a finales de los '70 por Cleve Moler como una herramienta para evitar a sus estudiantes tener que aprender el lenguaje Fortran. Aparece como producto comercial en el año 1984, oficializado y distribuido por MathWorks y desde entonces ha venido ganando popularidad, principalmente en el campo del control lineal y no lineal así como en la ingeniería en general. En la actualidad, este software dispone de una cantidad considerable de funcionalidades tanto para el cálculo numérico como para la graficación, que justifican el uso que se le da a nivel internacional. Entre sus prestaciones más importantes se encuentran la implementación de las matrices como elemento básico del lenguaje, lo que posibilita una gran reducción del código al no ser necesario implementar el álgebra matricial; la posibilidad de ampliar y adaptar el lenguaje mediante el uso de ficheros "script" y funciones ".m", el análisis numérico, el trabajo con gráficos, la implementación de aritmética lógica, y los toolboxes, los cuales comprenden un conjunto de funciones que extienden las capacidades del software para resolver problemas específicos en esferas tales como el procesamiento de señales, diseño de sistemas de control, simulación de sistemas dinámicos, identificación de sistemas, redes neuronales, entre otros [19].

Independientemente de las facilidades que aporta Matlab en numerosos campos de la matemática computacional, así como su superioridad evidente con respecto a muchos de sus

semejantes, es cierto también que posee ciertas limitantes que restringen considerablemente su uso.

1.3.4.1. Matlab, software privativo

El software privativo, tal y como su concepto lo indica, se refiere a cualquier programa informático con limitadas posibilidades de uso, modificación o redistribución para los usuarios. El usuario de Matlab, por consiguiente, no podrá estudiar el funcionamiento de este software, y al mismo tiempo se verá imposibilitado de adaptarlo a sus necesidades, por la sencilla razón de que no posee el código fuente. De esta manera, el usuario se convierte en un ente completamente dependiente de sus proveedores, quienes deciden cómo y cuándo se actualiza Matlab. Además, si las personas que se dedican al desarrollo y mantenimiento de Matlab, dejaran de desempeñar su labor, el soporte técnico se esfumaría, y con éste, la posibilidad de enmendar errores o adquirir versiones mejoradas. Finalmente, todo aquel que compró una licencia, no podrá más que usar la misma versión del software por el resto de sus días [14].

1.3.4.2. Precio actual de Matlab

Los precios actuales de este software son realmente inalcanzables. Los países del tercer mundo, se ven imposibilitados de adquirir esta herramienta tan importante en el campo de las matemáticas y la ingeniería, la cual es lo suficientemente costosa como para que sea descartada por muchas instituciones educacionales o de investigación y de esta manera comienzan las diferencias entre quienes lo adquieren y aquellos que no pueden, acentuándose aún más la brecha tecnológica y todas las calamidades que de ella derivan y que tanto afectan a las naciones.

Para tener una idea más clara de lo que significa hoy en día el pago de las licencias de Matlab, véase el caso desde una perspectiva primermundista: Hoy grandes multinacionales llevan a la mesa de conversaciones el tema relacionado con el reemplazo de Matlab dado que la adquisición de las licencias necesarias no está siendo rentable. Rolls Royce, por ejemplo, es una multinacional que ocupa uno de los primeros puestos a nivel mundial como fabricante

de motores para aviación. Debido al volumen de información tan elevado que se maneja, es imprescindible que todos los ingenieros que trabajan allí sean capaces de utilizar Matlab, lo que implica que la adquisición del número de licencias necesarias de esta herramienta, se esté convirtiendo en una opción poco viable. Los servidores de licencias, como un paliativo a esta situación, son una solución parcial al problema ya que en los picos de trabajo, cuando se necesitan todas las licencias, es probable que un ingeniero clave no pueda trabajar. Como resultado de esto, el uso del Matlab se ha ido limitando y hasta descartando, en determinadas áreas, mientras que los datos se analizan mediante hojas de cálculo con la consiguiente pérdida de productividad [19].

1.3.4.3. Matlab en las universidades

La educación es una de las necesidades básicas de un país. José Martí, Héroe Nacional de Cuba, como parte de su vasta obra sentenció que ser culto es el único modo de ser libres, y es precisamente en este mundo moderno, donde el acceso a la cultura, y por tanto, a la libertad, depende en gran medida del acceso a la tecnología. Al tanto de estas necesidades, es que muchos centros educacionales invierten en la adquisición de modernas tecnologías y herramientas. Especialmente en la esfera de las Matemáticas y de manera más específica en el cálculo, el empleo de asistentes matemáticos, es una necesidad y un factor de peso en el afán de mejorar la calidad de la enseñanza; es una verdad irrefutable el hecho de que los resultados en el proceso de aprendizaje se encuentran influenciados por la cantidad y calidad de los medios destinados a este fin.

La Universidad Politécnica de Madrid, España, posee de un número considerable de licencias para el uso académico de Matlab. Las mismas son actualizadas frecuentemente con el objetivo de que tanto los alumnos como el personal docente e investigador dispongan de la versión más reciente del software. En algunas escuelas, las prácticas de la asignatura de cálculo numérico se realizan en Matlab, pero, muchos estudiantes sólo usan un pequeño porcentaje de una herramienta que pagan completamente; situaciones de este tipo en diferentes partes del mundo sin lugar a dudas dan pie a interrogantes como estas: ¿Es rentable

entonces utilizar una herramienta tan costosa, cuando existen programas similares capaces de arrojar el mismo resultado? ¿Acaso es lógico comprar esta herramienta al precio de dos computadoras cuando puedes adquirir dichas computadoras e instalarle otra herramienta de prestaciones similares? Octave, por ejemplo.

Esto mismo sucede en muchas universidades del mundo, las condiciones están creadas para un reemplazo, solo se necesita tiempo, voluntad, y en algunos casos plantearse este tipo de interrogantes [20].

1.3.5. Scilab

Scilab es un proyecto científico francés, desarrollado por INRIA (Institut Nationale de Recherche en Informatique et en Automatique) para el cálculo numérico, con un entorno muy amigable para el usuario, de libre distribución y copia, multiplataforma, que incluye una gran cantidad de funciones matemáticas, las cuales pueden ser ampliadas mediante programas escritos en C, Fortran y otros lenguajes. La licencia a la que está sujeto este software (Licencia SCILAB) no es precisamente una licencia de software totalmente libre. ya que no es compatible con la GPL. Es decir, Scilab es un software libre para distribuir, copiar, pero no vender ni modificar.

Este software fue creado esencialmente para realizar cálculos numéricos, aunque también ofrece la posibilidad de hacer algunos cálculos simbólicos simples, como derivadas de funciones polinomiales y racionales. Posee además, un lenguaje de programación propio y un intérprete sofisticado.

Scilab, al igual que Matlab, pone a disposición de los usuarios un número importante de herramientas para la Graficación 2D y 3D, la Animación, el Álgebra Lineal, la Modelación y Simulación con sistemas híbridos⁵(Scicos), el trabajo con Matrices Dispersas, Polinomios,

⁵Sistema que emplea técnicas tanto analógicas como digitales.

Funciones Racionales, entre muchas otras. Scilab es usado, incluso, para el procesamiento de señales y para el aprendizaje de teoría de circuitos [21].

Dichas cualidades le han reportado gran popularidad en el campo de la matemática computacional, al extremo de ser el asistente matemático preferido por muchos. Pero hay algo en Scilab que frena su total aceptación por parte de los usuarios y que lo mantiene al margen cuando de Octave se trata.

1.3.5.1. Incompatibilidad entre Scilab y Matlab

Existe un detalle que no puede pasar por alto y es, a criterio de muchos, determinante a la hora de asumir un desarrollo o una política de migración, y es que Scilab es mucho menos compatible con Matlab que otros asistentes matemáticos como por ejemplo Octave. Esto puede analizarse como un elemento que aporta a la diversidad de enfoques en el mercado de los asistentes, pero deja mucho que desear toda vez que Matlab es la herramienta por excelencia para el trabajo de cálculo numérico y ha marcado el paso en el área desde su creación en 1984; no ser compatible con Matlab, sin lugar a dudas es un problema.

Tal y como se observa en la tabla 1.1, Matlab y Octave usan comandos idénticos para un conjunto de acciones a diferencia de Scilab que opta por utilizar sus propios comandos, detalle que le resta una cantidad considerable de seguidores en un mundo tan competitivo como este.

—Matlab—	—Octave—	—Scilab—
Graficación		
plot	plot	plot, plot2d
polar	polar	polarplot
semilogx	semilogx	plot2d
semilogy	semilogy	plot2d
loglog	loglog	plot2d
mesh	mesh	...
meshgrid	meshgrid	...
hist	hist	histplot
Polinomios		
polyval	polyval	poly
deconv	deconv	pdiv

Tabla 1.1: Comandos específicos en Matlab, Octave y Scilab para el trabajo con gráficos y polinomios.

1.3.6. Octave

Octave o GNU Octave, es un asistente matemático libre de gran utilidad en el campo del cálculo numérico, basado en la filosofía GNU, lo cual implica tener acceso al programa y al código fuente del mismo para transformarlo si se desea, sin la necesidad de abonar un solo dólar por su obtención o utilización, además de poder copiarlo o instalarlo libremente a diferencia del software privativo.

El proyecto fue creado alrededor del año 1988, en la Universidad de Wisconsin en Madison como una herramienta para completar las prácticas de ingeniería química. Posteriormente en el año 1992, se decide extenderlo y comienza su desarrollo a cargo de John W. Eaton.

La primera versión Alpha⁶ fue lanzada el 4 de enero de 1993. Un año más tarde, el 17 de febrero, 1994 aparece la versión 1.0. Pronto se popularizó dentro del mundo GNU/Linux y fue solo cuestión de tiempo el hecho de que haya sido incluido en varias distribuciones como paquete oficial.

Para comprender mejor lo que es Octave se trae a colación las propias palabras de John W. Eaton cuando afirmaba: “Hay diferencias entre Matlab y Octave y algunas seguirán así. No existen porque Octave esté roto sino porque creemos que Matlab lo está (...)”, según estas palabras puede concluirse que Octave no es un clon de Matlab como muchos creen, sino que ha ido convergiendo por pura necesidad sobre la base de algunas mejoras conceptuales realizadas a su parecido más cercano [20].

Octave un fue escrito en C++ usando la librería STL y posee un intérprete de su propio lenguaje que permite ejecutar sentencias de forma interactiva, aunque también posibilita la ejecución por lote, o lo que es lo mismo: la ejecución de acciones sin mediación del usuario. El procesamiento interactivo se realiza a través del lenguaje Octave, un lenguaje interpretado de alto nivel y que soporta gran parte de las funciones de la librería estándar de C y otras similares, las cuales pueden extenderse con el uso de funciones y procedimientos por medio de módulos dinámicos. Entre las funcionalidades que incorpora Octave se pueden mencionar:

1. Herramientas para resolver problemas de álgebra lineal.
2. Calcular raíces de ecuaciones no lineales.
3. Integrar funciones.
4. Manipular polinomios.
5. Integrar ecuaciones diferenciales.

⁶Es el primer paso para acercar el producto al usuario final.

De manera adicional, y por mediación de su propio lenguaje, Octave se comporta como una consola intérprete de comandos (shell) lo cual permite listar el contenido de un directorio, entre otras funcionalidades. Algunos programas como Gnuplot y Grace son usados por Octave para el trabajo con gráficas.

1.3.6.1. Octave, sustituto potencial de Matlab

Muchos ven al software libre como una alternativa utópica al software privado, como algo irrealizable, o como una idea descabellada, tal es el caso de Octave como sustituto potencial de Matlab. Algunos proyectos de software libre argumentan que la libertad de uso es un motivo suficiente como para no utilizar cualquier tipo de software propietario. Usar el primero como elección ante el segundo, significa gozar de todas las libertades que éste ofrece. Octave es libre, de ahí que tiende constantemente a ser más robusto toda vez que cuenta con el apoyo de muchas personas dispuestas a contribuir, tiende a ser más eficiente pues está siendo optimizado continuamente por quienes lo utilizan, y posee una diversidad progresiva debido a la cantidad de necesidades diferentes de la comunidad, lo cual es indudablemente una ventaja insuperable sobre Matlab y al mismo tiempo una vía para el crecimiento sostenible de la cantidad de usuarios hoy, y desarrolladores mañana. Cualquier persona con una computadora y conexión a Internet, puede usar Octave, no así Matlab.

A pesar de los inconvenientes antes mencionados sobre Matlab, no se puede pasar por alto que es un programa magnífico con una interfaz gráfica muy completa y que está disponible para varios sistemas operativos, además posee un compilador envidiable y sus potencialidades para la representación gráfica son sumamente amplias. Sin embargo su pársers⁷ no acepta estructuras sintácticas que serían consistentes en el lenguaje. Por otra parte, la interfaz gráfica, escrita en Java y aunque bastante completa, no es del todo estable y lo mismo sucede con el intérprete; éste es vulnerable ante situaciones relativamente sencillas.

⁷Programa o parte de un programa que analiza una secuencia de palabras usando reglas gramaticales con el fin de crear estructuras que serán procesadas.

Octave, por su parte, parece haber solucionado algunos de las imprecisiones técnicas de Matlab. El párser, aunque no es perfecto, es más listo y acepta estructuras más complejas. El intérprete, al igual que el de Matlab, presenta problemas de estabilidad, pero estos se corrigen rápidamente debido al corto tiempo de publicación entre una versión y otra. Del mismo modo, es fácil de extender en C++ y permite un acceso directo a su maquinaria interna y a sus librerías para utilizarlas en cualquier programa en este lenguaje. Además, hay que contar con el hecho de que Octave es 99% compatible con Matlab, al punto de que se puede trabajar en uno, dejar el trabajo a medias y luego continuar dicho trabajo con otro sin el más mínimo problema, a diferencia de otros asistentes como Scilab, por solo citar un caso [20].

Octave, desde sus inicios y hasta hace relativamente poco tiempo, no contaba con un entorno de desarrollo que lo acercara aún más a su semejante privativo, éste era el factor fundamental por el cual muchos usuarios a nivel mundial se abstenían a la tentativa del cambio de asistente, la facilidad de uso y la calidad didáctica de Matlab era incomparablemente mayor. Hoy la realidad es bien distinta y Octave cuenta con varios frontends disponibles entre los que se pueden mencionar Octavede, Koctave, Goctave, Yaog, Joctave y QtOctave, muchos de los cuales subsanan el inconveniente anterior y abren muchas puertas hacia un futuro de entera libertad.

Después de este pequeño análisis en cuanto a los contrastes entre estos programas matemáticos, así como las diferencias entre sus costos y la similitud en los resultados del trabajo con cada uno de ellos, no cabe duda que Octave es un candidato prometedor al trono del cálculo numérico computacional si tomamos en cuentas nuestras realidades.

1.4. Frontends

Un frontend es un software que provee de una interfaz más “amigable” a otra aplicación que si bien es la que realmente administra la información, su uso directo representa algún

grado de dificultad para el usuario al que está destinada.

1.4.1. Características del entorno gráfico de Matlab

Matlab es un entorno de computación y desarrollo de aplicaciones totalmente integrado, orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos. Matlab integra análisis numérico, cálculo matricial, proceso de señales y visualización gráfica en un entorno completo.

1.4.1.1. El entorno de trabajo de Matlab

El entorno de trabajo Matlab consta de cinco partes fundamentales:

1. El entorno de desarrollo.
2. Las librerías de funciones.
3. El lenguaje de programación Matlab.
4. El manejo de gráficos.
5. La Interfaz de Programas de Aplicación (API)

1.4.1.2. Estructuración del entorno de desarrollo de Matlab:

El entorno de desarrollo de Matlab se estructura de la siguiente manera:

- Ventana de comandos: sirve para ejecutar comandos e invocar funciones y ficheros “.m”.
- Histórico de comandos: recoge las últimas instrucciones.
- Launch Pad: proporciona acceso a herramientas, demos y documentación.
- Help Browser: facilita la búsqueda de documentación.

- Current Directory Browser: permite seleccionar el directorio actual en el que están las funciones que pueden invocarse en cada momento.
- Workspace Browser: en el que se almacenan las variables empleadas en una sesión.
- Editor/Debugger: para crear y depurar ficheros .m y funciones.

Existe además la posibilidad de importar/exportar variables del workspace de/a otras aplicaciones. Del mismo modo, existen herramientas para depurar las prestaciones de los ficheros de Matlab.

1.4.2. Frontends para Octave

El camino a recorrer en aras de confeccionar un producto con características similares al entorno de desarrollo de Matlab es aún bastante largo y escabroso. Numerosos proyectos se han trazado la meta de darle vida a un frontend capaz de impregnarle a Octave la facilidad de uso que hoy incorporan otros asistentes matemáticos como Matlab, con el fin de brindarle al usuario final la mayor comodidad posible, y a la vez, recobrar la fuerza perdida por Octave.

1.4.2.1. KOctave

Koctave es un frontend basado en KDE para GNU Octave bajo la licencia GPL, y portador, por lo tanto, de todas las libertades que la misma otorga. La interfaz gráfica posee un editor de texto con “syntax highlighting”⁸, incluye además, una búsqueda de ayuda, una ventana de comandos y otra de navegación, de una forma muy similar a Matlab, excepto con la ventana de comandos, la cual es una lista de sentencias creada por el usuario y no un historial de comandos como en Matlab. Actualmente se encuentra en fase de desarrollo Beta⁹ [22].

⁸Habilidad para diferenciar texto y elementos estructurales como palabras reservadas, etiquetas, atributos, etc.

⁹El software versión Beta es el que en breve va a ser lanzado.

1.4.2.2. GOctave

Goctave es un frontend basado en Gnome bajo la licencia GPL, característica que lo hace completamente asequible para cualquier usuario con internet. Entre las novedades que incorpora, se encuentra el soporte para macros¹⁰ así como la edición de las mismas en tiempo de ejecución, una ayuda en línea mejorada y una interfaz para editar matrices, además de incluir una librería con numerosas funciones en C++ y Octave [23].

1.4.2.3. Octavede

Octavede es un ambiente de desarrollo integrado para el software matemático Octave, el cual proporciona una aplicación GTK/X11 con funcionalidades similares a otros entornos de desarrollo.

Al igual que sus similares, es multiplataforma, y puede ser descargado de la web bajo la licencia GPL. Actualmente se encuentra en estado de desarrollo Alpha, es decir, que es aún inestable y está en espera de que se eliminen los errores durante el período de prueba [24].

1.4.2.4. Yaog

Yaog es otro de los IDE para Octave, escrito en C usando la librería GTK2 y bajo la licencia GPL. Este IDE posee ciertas características como son un editor de texto integrado con “syntax highlighting”, además existen algunas que serán incorporadas pronto entre las que se encuentran una interfaz de navegación, debugeo en modo gráfico, integración opcional con GNOME, la edición de variables de Octave en tiempo de ejecución y una completa manipulación de las rutas de Octave mediante una interfaz gráfica. Yaog, al igual que Octavede, se encuentre en la fase Alpha de desarrollo [25].

¹⁰Conjunto de comandos que sirven para automatizar tareas que se ejecutan de forma repetitiva.

1.4.2.5. Octave Workshop

Octave Workshop es un entorno de desarrollo integrado para Octave escrito en C++. Este brinda la posibilidad de hacer cálculos rápidos usando el historial del buffer para recordar operaciones anteriores y encontrar rápidamente una función o variable. Para obtener cálculos más complejos se pueden escribir los programas en un editor de texto que automáticamente destaca y autocompleta el código. Con la ficha Archivos, se le puede ver el contenido del directorio de trabajo, así como cargar cualquier (.m) con solo dar doble clic.

- Características:

- Utiliza toda la potencialidad de los comandos de Octave dentro de su ventana principal.
- Permite editar tus propios archivos .m.
- Utiliza la biblioteca de funciones completa de Octave e incluye graficación con GNU Plot.
- Graficación en 2D, 3D y permite la exportación de las gráficas.

Este proyecto en un principio se asemejó mucho a QtOctave, pero en lugar de usar “Strings“ para comunicarse con Octave, usaba código en C++ del propio Octave lo cual es más complejo de mantener a medida que crezca la aplicación[26].

1.4.2.6. QtOctave

QtOctave¹¹ es otro frontend para GNU Octave, desarrollado en Qt4, mucho más acabado y organizado que los hasta ahora vistos. Incorpora un conjunto de asistentes y menú con el objetivo de facilitar enormemente el uso de Octave, pues como se había tratado anteriormente, el trabajo con este programa matemático, desde la consola, suele ser un poco tedioso. QtOctave, presenta además, un poderoso sistema de búsqueda de ayuda, se puede

¹¹En lo sucesivo, cada vez que se mencione QtOctave entiéndase que se está haciendo alusión a la versión 0.7.1.

incluso obtener ayuda de un comando según se va tecleando, mediante “Dynamic Help”. En la actualidad, debido a la diversidad de lenguajes de programación existentes, resulta muy útil la ayuda con los comandos, al punto que un entorno de desarrollo sin esta característica no gozará de la mejor aceptación por parte de los usuarios.

Las características más notables de QtOctave, y probablemente las que lo catapultaron por encima de sus homólogos contemporáneos, las constituye el hecho de tratar a las matrices como hojas de cálculos, lo cual simplifica de manera sustancial la interacción con las mismas, así como el hecho de que los usuarios puedan crear sus propios menú ventanas desde el Octave mediante la aplicación `xmlwidget`, la cual funciona desde prácticamente cualquier lenguaje de programación al mismo tiempo que amplía los horizontes de Octave. Tales funcionalidades le valieron a QtOctave el galardón obtenido en el Concurso de Software Libre celebrado en España con el objetivo de promover el desarrollo de proyectos de Software Libre en las universidades y de esta manera fortalecer a la comunidad en el ámbito universitario [27].

Como puede verse, confeccionar un frontend para Octave es un reto que han asumido un gran número de personas a nivel mundial, algunos en sus momentos de ocio y otros como proyectos serios y bien definidos, pero la realidad es que pocos han sido capaces de lograr un producto de calidad que represente a este asistente matemático en el ámbito internacional, aportándole un grado de integridad y facilidad de uso similar al de Matlab. QtOctave ya incluye un elevado número de prestaciones, respaldadas por un conjunto de herramientas para el fácil manejo de matrices, estadísticas, funciones, gráficos; esto, unido a su intuitivo manejo, su escaso consumo de memoria, su portabilidad y su rapidez, hacen de QtOctave el frontend que marca el paso en la concepción de un ambiente de desarrollo integrado cada vez más eficiente, más funcional, más adaptable y más novedoso.

Capítulo 2

Características del sistema

EL presente capítulo incluye todo lo relacionado con las características de QtOctave, así como la fundamentación de la metodología de desarrollo utilizada. Se confecciona el modelo de dominio del sistema y se hace el levantamiento de los requisitos del mismo. Debido a que la decisión de concebir dicho modelo de dominio, podría dar pie a conjeturas, se respalda de forma concreta el por qué de esa elección. Concluye el capítulo con la especificación de los casos de uso.

2.1. Metodología de desarrollo del software

La informática es una ciencia joven, tan joven que las experiencias acumuladas en el proceso de desarrollo de software son aún escasas. El auge creciente de las tecnologías y su aplicación en todas las esferas de la ciencia y la industria intensifica la demanda de software de fines diversos. Muchos creen poseer herramientas o experiencias suficientes para crear un producto software, prescindiendo de una guía o principios de desarrollo bien definidos y gestionados; sin saber que realmente son esclavos del tiempo y de las exigencias del usuario, las que no son capaces de satisfacer si no se apoyan en un plan de trabajo bien organizado que pueda guiarlos hacia un objetivo concreto.

2.1.1. Proceso unificado de desarrollo de software (RUP)

“El Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto (...) RUP está basado en componentes, lo cual quiere decir que el software en construcción está formado por componentes software interconectados a través de interfaces bien definidas”, pero los verdaderos aspectos que lo hacen único son el hecho de ser “dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental” [13].

“El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos esquemas de un sistema software. De hecho, UML, es una parte esencial del Proceso Unificado – sus desarrollos fueron paralelos” [13].

2.1.2. Extreme Programming (XP)

Extreme Programming o Programación Extrema, es una metodología de desarrollo de software reciente cuya filosofía es satisfacer al máximo las necesidades del cliente, de ahí que lo considera como una parte más del equipo de desarrolladores. En los métodos de desarrollo tradicionales, SSADM por ejemplo, los requerimientos para el sistema son establecidos al principio del proyecto y frecuentemente se mantienen hasta el final, pues cualquier cambio posterior en dichos requisitos posee un elevado costo. XP reduce considerablemente dichos costos al establecer valores básicos, principios y prácticas con el objetivo de que el sistema de desarrollo sea más flexible a los cambios. Esto sucede, naturalmente, cuando el cliente no está muy claro de lo que desea.

De las metodologías de desarrollo expuestas anteriormente se ha considerado pertinente utilizar RUP debido a su versatilidad, organización y adaptabilidad. Este proceso unificado de desarrollo de software brinda los elementos suficientes para realizar la reingeniería a QtOctave. Se estima que XP es una metodología muy joven, superada por RUP en cuanto a documentación y experiencias acumuladas. RUP define en cada momento del ciclo de vida del

proyecto, qué artefactos deben ser creados, con qué nivel de detalle, cuándo y por quién debe ser creado. Aplica además, los conocimientos acumulados durante las iteraciones anteriores, y precisamente esta especie de retroalimentación reduce considerablemente los riesgos.

2.2. Reingeniería de software

La ingeniería inversa del software es el proceso de análisis de un programa con el fin de crear una representación del mismo con un nivel de abstracción más elevado que el código fuente [28]. Consiste principalmente en recuperar el diseño de una aplicación a partir del código.

Este proceso es aplicable a sistemas con las siguientes características:

1. Carencia de una documentación o el estado de la misma es completamente desactualizado.
2. Programación en bloques de códigos muy grandes y/o sin estructurar.
3. La aplicación abarca gran parte de los requisitos y del rendimiento esperado.
4. La aplicación está sujeta a cambios frecuentes, que pueden afectar el diseño.
5. Se avisa que la aplicación pueda tener aún larga vida.

2.3. Modelo de dominio

El modelo del dominio es considerado un subconjunto del modelo de objetos del negocio. El objetivo principal del mismo es comprender, describir y capturar las clases más importantes dentro del contexto del sistema.

Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja dicho sistema. Muchos de estos objetos o clases del dominio, como se considere pertinente llamar, pueden obtenerse de una especificación de requisitos o mediante la entrevista con los expertos del dominio. Este modelo se describe mediante diagramas de UML (especialmente mediante diagramas de clases) que muestran las clases del dominio y cómo se relacionan unas con otras mediante asociaciones.

Un modelo de dominio se lleva a cabo en dependencia del escenario a tratar. En este caso se estimó conveniente no realizar un modelo completo del negocio ya que los procesos del mismo no estaban bien definidos y lo que se necesitaba era representar de manera sencilla la forma en que se relacionaban los módulos presentes en el software, para que el lector tuviese una visión de la estructura de QtOctave. Se describieron a continuación, de forma breve, las principales funciones de cada módulo implicado en el mismo, las cuales no se incluyen en el diagrama por evitar un tamaño exagerado de éste.

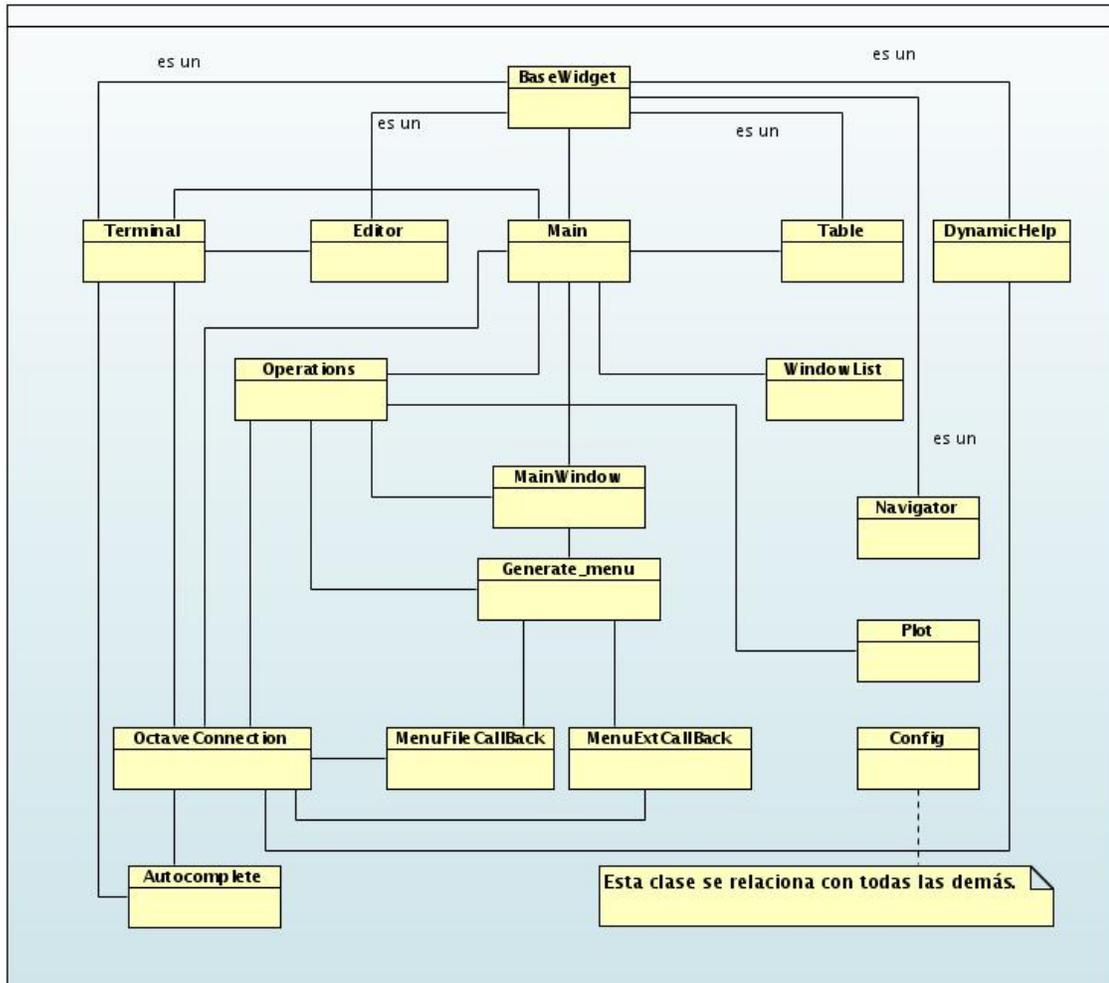


Figura 2.1: Modelo de dominio.

2.3.1. Breve descripción de las clases implicadas

BaseWidget (*Base de componentes gráficos*): Es la base de todas las ventanas mostradas en QtOctave. Sirve para que el sistema pueda tratar todas las herramientas de una forma homogénea. Así se puede replicar un objeto sin de necesidad de conocer sus características.

Help (*Ayuda*): Muestra la ayuda en HTML simple dando la posibilidad de búsqueda a los usuarios.

Main (*Principal*): Ejecuta la aplicación, interconecta y controla todos los módulos de QtOctave.

WindowList (*Lista de ventanas*): Muestra una lista de las ventanas asociadas a un área de trabajo específica, cuando uno de los elementos de la misma es seleccionado la ventana asociada se maximiza automáticamente.

Table (*Tabla*): Muestra una tabla con una hoja de cálculo cuyos valores provienen de una matriz, cuando el usuario hace modificaciones en la tabla, la matriz correspondiente es modificada a la par.

Terminal (*Terminal*): Crea una ventana con un área de texto editable para escribir comandos a Octave y otra para mostrar la salida del mismo.

OctaveConnection (*Conección a Octave*): Ejecuta Octave y controla la entrada y salida de datos del mismo. Puede, además, capturar ciertas salidas que son enviadas como señales a los objetos que lo soliciten.

DynamicHelp (*Ayuda dinámica*): Muestra la ayuda dinámicamente para los comandos de Octave escritos en el área de texto editable.

MainWindow (*Ventana principal*): Crea la ventana principal de QtOctave.

Generate_menu (*Generar menú*): Ayuda a crear los menú de QtOctave. Los menú se pueden guardar en un fichero usando la sintaxis indicada en la documentación de la aplicación, para después ser generados de forma dinámica durante el arranque de la misma.

MenuCallback (*Llamada a los menú*): Es la encargada de la llamada a los menú cuando éstos son ejecutables, y no ficheros *.menu.

MenuExtCallback (*Llamada a los fichero *.menu*): Es la encargada de las llamadas a los menú que no son ejecutables.

Operations (*Operaciones*): Es la encargada de conectar los menú con las operaciones a realizar. QtOctave distingue entre las operaciones propias de la interfaz y las operaciones matemáticas. En este módulo se procura conectar las acciones relacionadas con estas últimas.

Editor (*Editor*): Es el editor del código que introduce el usuario.

Navigator (*Navegador*): Es el navegador de archivos con posibilidad de arrastrar y soltar sobre los otros elementos. Si se arrastra un archivo sobre el editor se abre. Si se arrastra sobre el terminal se ejecuta.

Autocomplete (*Autocompletar*): Es la encargada de autocompletar los comandos al pulsar la tecla tabulador en el terminal.

Plot (*Plot*): Muestra las opciones para dibujar.

Config (*Configuración*): Contiene la configuración de todos los parámetros de la aplicación.

2.4. Requerimientos del sistema

Un requerimiento no es más que la condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. A continuación, se relacionan dichos requisitos desglosados por categorías.

2.4.1. Requisitos funcionales

R1 Gestionar proyecto.

R1.1 Abrir proyectos.

R1.2 Modificar proyectos.

R1.3 Eliminar proyectos.

R2 Gestionar comandos.

R2.1 Enviar comandos a Octave.

R2.2 Mostrar ayuda dinámicamente para los comandos de Octave.

R2.3 Autocompletar comandos.

R3 Gestionar matrices.

R3.1 Crear matriz.

R3.2 Cargar matriz.

R3.3 Salvar matriz.

R4 Operar matrices.

R4.1 Calcular determinante.

R4.2 Calcular autovalores y autovectores.

R4.3 Calcular inversa.

R4.4 Calcular transpuesta.

R4.5 Calcular submatriz.

R5 Manejar estadísticas.

R5.1 Calcular media.

R5.2 Calcular mediana.

R5.3 Calcular desviación estándar.

R5.4 Calcular covarianza.

R5.5 Calcular coeficiente de relación.

R5.6 Histograma.

R6 Resolver ecuaciones.

R6.1 Resolver ecuaciones por el método de bisección.

R6.2 Resolver ecuaciones lineales.

R6.3 Resolver ecuaciones no lineales.

R7 Graficar.

R7.1 Graficar 2D.

R7.1.1 Plotear.

R7.1.2 Plotear en sistemas de coordenadas polares.

R7.1.3 Escalar de forma logarítmica los ejes x y y de forma simultánea.

R7.1.4 Escalar de forma logarítmica el eje x.

R7.1.5 Escalar de forma logarítmica el eje y.

R7.1.6 Mostrar barra de errores.

R7.1.7 Histograma.

R7.1.8 Mostrar gráfica de barra.

R7.2 Graficar 3D.

R7.2.1 Plotear gráficos tridimensionales.

R7.2.2 Plotear gráfico de contornos de superficies tridimensionales.

R7.3 Manipular Gráfica.

R7.3.1 Establecer límites de gráficas en los ejes coordenados.

R7.3.2 Nombrar y etiquetar gráfica.

R7.3.3 Exportar gráfica.

R8 Gestionar configuración.

R8.1 Configurar terminal.

R8.2 Configurar editor.

R8.3 Configurar octave.

R8.4 Configurar ayuda.

R9 Gestionar menú.

R9.1 Crear menú.

R9.2 Editar menú.

R9.3 Eliminar menú.

R10 Contar con un editor de scripts para Octave.

R10.1 Cargar archivos *.m.

R10.2 Guardar archivos *.m.

2.4.2. Requisitos de interface

R11 Capturar las entradas y salidas de Octave.

2.4.3. Requisitos de operación

R12 Un sistema de ayuda con un sistema de búsqueda.

R13 Distinguir comandos en el editor con un color diferente.

R14 Editar las matrices.

R15 Un menú con las operaciones más comunes separadas por categorías.

R16 Que los usuarios puedan configurar su editor favorito.

R17 Un editor del path.

R18 Una ventana/asistente para hacer gráficos.

R19 Un asistente/menú para controlar el formato de salida.

R20 Menú para cambiar la configuración y la apariencia de Octave.

R21 Pestañas/listado de las ventanas/objetos abiertos por Octave.

R22 Autocompletado de comandos.

2.4.4. Requisitos de recursos

R23 Debe funcionar sobre sistemas operativos Linux o Unix

R24 Requerimientos de hardware mínimos para poder usar Qt.

R25 Consumo de memoria inferior a 20Mb.

2.4.5. Requisitos de verificación

R26 Modo de trabajo que visualice el intercambio de información entre QtOctave y Octave.

2.4.6. Requisitos de transportabilidad

R27 Usar el mínimo de bibliotecas.

R28 Usar Qt.

2.4.7. Requisitos de calidad

R29 Los menú de la aplicación deben aparecer en tiempo razonable.

2.4.8. Requisitos de fiabilidad

R30 La aplicación debe ser estable.

2.4.9. Requisitos de mantenibilidad

R31 Adaptable a los cambios.

2.5. Definición de los casos de uso

2.5.1. Definición de los actores

Los actores son terceros fuera del sistema que interactúan con él. Un actor puede que:

1. Solo brinde información de entrada al sistema.
2. Solo reciba información del sistema.
3. Intercambie información con el sistema.

Actores	Justificación
Persona	Es el individuo que puede realizar tanto las actividades del usuario como las del administrador.
Usuario	Es el individuo que interactúa con el sistema pero no posee privilegios administrativos sobre el Sistema Operativo.
Administrador	Es el individuo con privilegios administrativos en el Sistema Operativo.

Tabla 2.1: Listado de los actores del sistema.

2.5.2. Diagrama de casos de uso del sistema

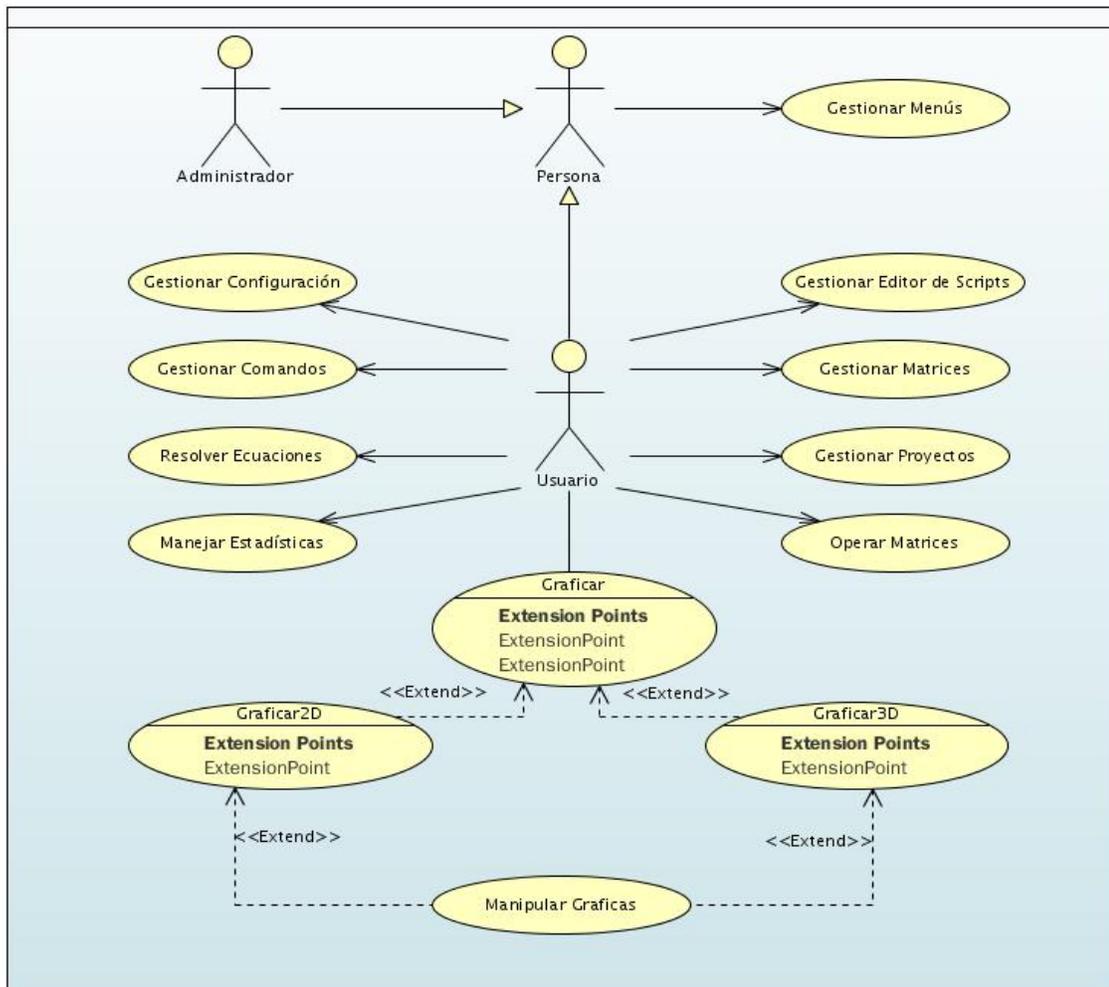


Figura 2.2: Diagrama de casos de uso.

2.6. Descripción de los casos de uso del sistema

A continuación se hace una descripción de cada casos de uso del sistema, desglosandolos en secciones, con el objetivo de ilustrar como fluye cada proceso.

Gestionar Proyectos	
Código del CU	CU-1
Propósito	Permitir que los usuarios creen, modifiquen o eliminen un proyecto.
Actores	Usuario
Resumen	El CU se inicia cuando el usuario ejecuta la aplicación. Inmediatamente se visualiza una ventana en la que se brinda la posibilidad de crear un proyecto nuevo, modificar o eliminar uno de los existentes.
Referencias	R1.1, R1.2, R1.3
Sección Crear Proyecto	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Abrir un proyecto nuevo New o da clic derecho sobre uno de los proyectos existentes y selecciona New Project.	1.1. Mostrar la ventana New Project.
2. Especificar nombre y descripción del proyecto.	2.1. Cerrar la ventana New Project y mostrar la ventana principal de QtOctave.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	2.1. Cerrar la ventana New Project.
Sección Modificar Proyecto	

Continúa...

...Continuación

Gestionar Proyectos	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Dar clic derecho sobre uno de los proyectos existentes y selecciona Modify Project.	1.1. Mostrar la ventana Modify Project.
2. Cambiar el nombre y la descripción del proyecto.	2.1. Cerrar la ventana Modify Project y cambiar el nombre y la descripción del proyecto seleccionado.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	2.1. Cerrar la ventana Modify Project.
Sección Eliminar Proyecto	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Dar clic derecho sobre uno de los proyectos existentes y selecciona Delete Project.	1.1. Mostrar la ventana Delete Project.
2. Presionar el botón Yes.	2.1. Eliminar el proyecto seleccionado.
Flujo alternativo	
Acción del actor	Respuesta del sistema
3. Presionar el botón No.	3.1. Cerrar la ventana Delete Project.
Puntos de extensión	

Tabla 2.2: Caso de uso Gestionar Proyecto.

Gestionar Comandos	
Código del CU	CU-2
Propósito	Permitir a los usuarios enviar comandos a Octave y autocompletar los mismos.
Actores	Usuario
Resumen	El CU se inicia cuando el usuario escribe alguna palabra o letra en el área de comandos.
Referencias	R2.1, R2.2, R2.3
Sección Autocompletar Comandos	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Escribir alguna letra en el área de texto y presiona la tecla Tab.	1.1. Visualizar una ventana de autocompletamiento con una lista de los comandos existentes.
2. Seleccionar el comando que prefiere moviéndose con las teclas de dirección y presionando Enter o dando doble clic sobre el comando.	2.1. Visualizar el comando en el área de comandos.
Flujo alternativo	
Acción del actor	Respuesta del sistema
3. Presionar Esc.	3.1. Cerrar la ventana con la lista de comandos.
Sección Enviar Comandos	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema

Continúa...

...Continuación

Gestionar Comandos	
1. Escribir algún comando en el área de texto y presiona la tecla Enter.	1.1. Enviar el comando a Octave y realizar la orden en cuestión.
Flujo alternativo	
Acción del actor	Respuesta del sistema
2. Escribir un comando con una sintaxis incorrecta.	2.1. Enviar lo que hay en el área de texto editable y mostrar un mensaje de error en el área de texto de salida de datos.
Sección Ayuda Dinámica	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Escribir alguna comando en el área de texto.	1.1. Visualizar en ventana DynamicHelp la ayuda correspondiente al comando.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	1.1. La palabra escrita no es un comando y no se muestra la ayuda.
Puntos de extensión	

Tabla 2.3: Caso de uso Gestionar Comandos.

Gestionar Matrices	
Código del CU	CU-3
Propósito	Permitir que los usuarios realicen determinadas acciones con las matrices.
Actores	Usuario
Resumen	<p>El CU se inicia cuando el usuario selecciona el menú Data. Inmediatamente se visualiza un menú con un listado de operaciones con matrices:</p> <ol style="list-style-type: none"> 1. Table (Crear matriz). 2. Save Matrix (Salvar matriz). 3. Load Matrix (Cargar matriz).
Referencias	R3.1, R3.2, R3.3
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar el menú matriz.	1.1. Visualizar un menú con el listado de operaciones en cuestión.

Continúa. . .

...Continuación

Gestionar Matrices	
2. Escoger una de las operaciones a realizar.	2.1. Ejecutar una de las siguientes acciones: <ol style="list-style-type: none"> 1. Si selecciona Table, ir a la sección Crear Matriz. 2. Si selecciona Load matrix from file, ir a la sección Cargar Matriz. 3. Si selecciona Save matrix from file, ir a la sección Guardar Matriz.
Sección Crear Matriz	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Data, la opción Table.	1.1. Visualizar una interfaz para que el usuario introduzca el nombre de la matriz que va a crear y sus valores iniciales.
2. Introducir los datos.	2.1. Verificar que los datos introducidos sean correctos y que los campos requeridos no estén vacíos.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error para que los campos requeridos sean llenados o para que se introduzcan correctamente los datos.

Continúa...

...Continuación

Gestionar Matrices	
Sección Cargar Matriz	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Data, la opción Load Matrix.	1.1. Mostrar un cuadro de diálogo donde el usuario cargará el fichero con la matriz.
2. Introducir el nombre de la matriz a cargar.	2.1. Verificar que se haya introducido el nombre de la matriz a cargar.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error para que los campos requeridos sean llenados.
Sección Guardar Matriz	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Data, la opción Save Matrix.	1.1. Mostrar un cuadro de diálogo donde el usuario guardará la matriz en un fichero.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	1.1. Emitir un mensaje de error si no se especifica el nombre de la matriz a guardar.
Puntos de extensión	

Tabla 2.4: Caso de uso Gestionar Matrices.

Operar Matrices	
Código del CU	CU-4
Propósito	Permitir que los usuarios realicen determinadas operaciones con matrices.
Actores	Usuario
Resumen	<p>El CU se inicia cuando el usuario selecciona el menú (Matriz). Inmediatamente se visualiza un menú con un listado de operaciones con matrices:</p> <ol style="list-style-type: none"> 1. Determinant (Determinante). 2. Inverse (Inversa). 3. Transpose (Transpuesta). 4. Submatrix (Submatriz). 5. Eigenvalues y Eigenvectors (Autovalores y Autovectores).
Referencias	R4.1, R4.2, R4.3, R4.4, R4.5
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar el menú matriz.	1.1. Visualizar un menú con un listado de operaciones.

Continúa...

...Continuación

Operar Matrices	
2. Escoger una de las operaciones a realizar.	2.1. Ejecutar una de las siguientes acciones: <ol style="list-style-type: none"> 1. Si selecciona Determinant, ir a la sección Determinante. 2. Si selecciona Eigenvalues y Eigenvectors, ir a la sección Autovalores y Autovectores. 3. Si selecciona Inverse, ir a la sección Inversa. 4. Si selecciona Transpose, ir a la sección Transpuesta. 5. Si selecciona Submatrix, ir a la sección Submatriz.
Sección Calcular Determinante	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Matrix, la opción Determinant.	1.1. Mostrar una ventana con el campo Matrix name.
2. Introducir el nombre de la matriz.	2.1. Devolver el determinante de la matriz especificada.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema

Continúa...

...Continuación

Operar Matrices	
	2.1. Emitir un mensaje de error si no existe la matriz especificada.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Calcular Inversa	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Matrix, la opción Inverse.	1.1. Mostrar una ventana con el campo Matrix name.
2. Introducir el nombre de la matriz.	2.1. Devolver la inversa de la matriz especificada.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si no existe la matriz especificada.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.

Continúa...

...Continuación

Operar Matrices	
Sección Calcular Transpuesta	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Matrix, la opción Transpose.	1.1. Mostrar una ventana con el campo Matrix name.
2. Introducir el nombre de la matriz.	2.1. Devolver la transpuesta de la matriz especificada.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si no existe la matriz especificada.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Hallar Submatriz	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Matrix, la opción Submatrix.	1.1. Mostrar una interfaz con varios campos.
2. Escoger una matriz de la que sacara la submatriz, así como el rango de la misma.	2.1. Verificar que exista la matriz especificada, así como que el rango seleccionado sea válido.

Continúa...

...Continuación

Operar Matrices	
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si no existe la matriz especificada.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Puntos de extensión	

Tabla 2.5: Caso de uso Operar Matrices.

Manejar Estadísticas	
Código del CU	CU-5
Propósito	Permitir a los usuarios realizar determinados cálculos estadísticos.
Actores	Usuario
Resumen	<p>El CU se inicia cuando el usuario selecciona el menú Statistics. Seguidamente el sistema visualiza un menú con varias acciones:</p> <ol style="list-style-type: none"> 1. Mean (Media). 2. Median (Mediana). 3. Standard Deviation (Desviación Estándar). 4. Covariance (Covarianza). 5. Correlation Coefficient (Coeficiente de Relación). 6. Histogram (Histograma).
Referencias	R5.1, R5.2, R5.3, R5.4, R5.5, R5.6
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en menú Statistics.	1.1. Visualizar un menú con un listado de operaciones.

Continúa...

...Continuación

Manejar Estadísticas	
2. Escoger una de las operaciones a realizar.	2.1 Ejecutar una de las siguientes acciones: <ol style="list-style-type: none"> 1. Si selecciona Mean, ir a la sección Calcular Media. 2. Si selecciona Median, ir a la sección Calcular Mediana. 3. Si selecciona Standard Deviation, ir a la sección Calcular Desviación Estándar. 4. Si selecciona Covariance, ir a la sección Calcular Covarianza. 5. Si selecciona Correlation Coefficient, ir a la sección Calcular Coeficiente de Relación. 6. Si selecciona Histogram, ir a la sección Histograma.
Sección Calcular Media.	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Statistics, la opción Mean.	1.1. Mostrar una ventana con el campo Vector name.
2. Introducir el nombre del vector.	2.1. Devolver la Media del vector.

Continúa...

...Continuación

Manejar Estadísticas	
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si no existe el vector especificado.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Calcular Mediana.	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Statistics, la opción Median.	1.1. Mostrar una ventana con el campo Vector name.
2. Introducir el nombre del vector.	2.1. Devolver la Mediana del vector.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si no existe el vector especificado.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.

Continúa...

...Continuación

Manejar Estadísticas	
Sección Calcular Desviación Estándar.	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Statistics, la opción Standard Deviation.	1.1. Mostrar una ventana con el campo Vector name.
2. Introducir el nombre del vector.	2.1. Devolver la Desviación Estándar del vector especificado.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si no existe el vector especificado.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Calcular Covarianza.	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Statistics, la opción Covariance.	1.1. Mostrar una interfaz para introducir los datos.
2. Introducir los datos.	2.1. Verificar los datos y devolver la covarianza.

Continúa...

...Continuación

Manejar Estadísticas	
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Devolver un mensaje de error.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Calcular Coeficiente de Relación.	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Statistics, la opción Correlation Coefficient.	1.1. Mostrar una ventana con varios campos.
2. Introducir los datos.	2.1. Verificar los datos y devolver el coeficiente de relación.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Devolver un mensaje de error..
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.

Continúa...

...Continuación

Manejar Estadísticas	
Sección Histograma	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú 2D, el sub-menú Histograma.	1.1. Mostrar una ventana con los parámetros de graficación.
2. Introducir los parámetros de graficación.	2.1 Devolver la gráfica correspondiente.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si hay algún parámetro incorrecto.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	4.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Puntos de extensión	

Tabla 2.6: Caso de uso Manejar Estadísticas.

Resolver Ecuaciones	
Código del CU	CU-6
Propósito	Permitir que los usuarios resuelvan ecuaciones por diferentes métodos.
Actores	Usuario
Resumen	<p>El CU se inicia cuando el usuario selecciona el menú Equations. Inmediatamente se visualiza una lista con las opciones siguientes:</p> <ol style="list-style-type: none"> 1. Solve equation by bisection method (Resolver ecuación por el método de bisección). 2. Linear equation (Ecuación lineal). 3. Nonlinear equation (Ecuación no lineal).
Referencias	R6.1, R6.2, R6.3
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar el menú Equations.	1.1. Visualizar un menú con un listado de operaciones.

Continúa...

...Continuación

Resolver Ecuaciones	
2. Escoger una de las operaciones a realizar.	2.1 Ejecutar una de las siguientes acciones: <ol style="list-style-type: none"> 1. Si selecciona Solve equation by bisection method, ir a la sección Resolver Ecuación por el Método de Bisección. 2. Si selecciona Linear equation, ir a la sección Ecuación Lineal. 3. Si selecciona Nonlinear equation, ir a la sección Ecuación no Lineal.
Sección Resolver Ecuación por el Método de Bisección.	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Equations, la opción Solve equation by bisection method.	1.1. Mostrar una ventana con los campos correspondientes.
2. Introducir los parámetros.	2.1. Devolver el resultado.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si falta algún campo por llenar o si hay algún error en los parámetros.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema

Continúa...

...Continuación

Resolver Ecuaciones	
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Ecuación Lineal.	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Equations, la opción Linear Equation.	1.1. Mostrar una ventana con los campos correspondientes.
2. Introducir los parámetros.	2.1. Devolver el resultado.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si falta algún campo por llenar o si hay algún error en los parámetros.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Ecuación No Lineal.	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema

Continúa...

...Continuación

Resolver Ecuaciones	
1. Seleccionar en el menú Equations, la opción Nonlinear Equation.	1.1. Mostrar una ventana con los campos correspondientes.
2. Introducir los parámetros.	2.1. Devolver el resultado.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si falta algún campo por llenar o si hay algún error en los parámetros.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Puntos de extensión	

Tabla 2.7: Caso de uso Resolver Ecuaciones.

Graficar 2D	
Código del CU	CU-8
Propósito	Permitir a los usuarios graficar funciones en dos dimensiones.
Actores	Usuario
Resumen	El CU se inicia cuando el usuario selecciona el menú Plot. Inmediatamente se visualiza un menú desplegable con diferentes opciones. Seguidamente el usuario selecciona el tipo de graficación e introduce los parámetros y el sistema devuelve la gráfica correspondiente.

Continúa...

...Continuación

Graficar 2D	
Referencias	R7.1.1, R7.1.2, R7.1.3, R7.1.4, R7.1.5, R7.1.6, R7.1.7, R7.1.8
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Plot, la opción 2D.	1.1. Visualizar un submenú con varias opciones de graficación.
2. Seleccionar una opción.	2.1. <ul style="list-style-type: none"> ■ Si selecciona el submenú Plot, ir a la sección Plotear. ■ Si selecciona el submenú Polar, ir a la sección Plotear en Sistemas de Coordenadas Polares. ■ Si selecciona el submenú Log scale for x and y, ir a la sección Escalar Logarítmicamente los Ejes x e y, de Forma Simultánea. ■ Si selecciona el submenú Log scale for x axis, ir a la sección Escalar de Forma Logarítmica el Eje x.

Continúa...

...Continuación

Graficar 2D	
	<p>2.1.</p> <ul style="list-style-type: none"> ▪ Si selecciona el submenú Log scale for y axis, ir a la sección Escalar de Forma Logarítmica el Eje y. ▪ Si selecciona el submenú Error Bar, ir a la sección Mostrar Barra de Errores. ▪ Si selecciona el submenú Histogram, ir a la sección Histograma. ▪ Si selecciona el submenú Bar graph, ir a la sección Mostrar Gráfica de Barra.
Sección Plotear	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú 2D, el submenú Plot.	1.1. Mostrar una ventana con los parámetros de graficación.
2. Introducir los parámetros de la(s) ecuacion(es) a graficar.	<p>2.1. Registrar parámetros de cada ecuación en una lista.</p> <p>2.2. Devolver la(s) gráfica(s) correspondiente(s).</p>

Continúa...

...Continuación

Graficar 2D	
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
3. Cancelar una ecuación de la lista.	3.1. Eliminar los parámetros pertenecientes a la ecuación seleccionada.
	2.2. Emitir un mensaje de error si hay algún parámetro incorrecto.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
4. Seleccionar Copy to clipboard.	4.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Plotear en Sistemas de Coordenadas Polares	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú 2D, el submenú Polar.	1.1. Mostrar una ventana con los parámetros de graficación.
2. Introducir los parámetros de la(s) ecuación(es) a graficar.	2.1. Registrar los parámetros de cada ecuación en una lista. 2.2. Devolver la(s) gráfica(s) correspondiente(s).
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema

Continúa...

...Continuación

Graficar 2D	
3. Cancelar una ecuación de la lista.	3.1. Eliminar los parámetros pertenecientes a la ecuación seleccionada.
	2.2. Se emite un mensaje de error si hay algún parámetro incorrecto.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
4. Seleccionar Copy to clipboard.	4.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Escalar Logarítmicamente los Ejes x e y, de Forma Simultánea	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú 2D, el submenú Log scale for x and y axis.	1.1. Mostrar una ventana con los parámetros de graficación.
2. Introducir los parámetros de la(s) ecuación(es) a graficar.	2.1. Registrar los parámetros de cada ecuación en una lista. 2.2. Devolver la(s) gráfica(s) correspondiente(s).
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
3. Cancelar una ecuación de la lista.	3.1. Eliminar los parámetros pertenecientes a la ecuación seleccionada.

Continúa...

...Continuación

Graficar 2D	
	2.2. Se emite un mensaje de error si hay algún parámetro incorrecto.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
4. Seleccionar Copy to clipboard.	4.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Escalar de Forma Logarítmica el Eje x	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú 2D, el submenú Log scale for x axis.	1.1. Mostrar una ventana con los parámetros de graficación.
2. Introducir los parámetros de la(s) ecuacion(es) a graficar.	<p>2.1. Registrar los parámetros de cada ecuación en una lista.</p> <p>2.2. Devolver la(s) gráfica(s) correspondiente(s).</p>
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
3. Cancelar una ecuación de la lista.	3.1. Eliminar los parámetros pertenecientes a la ecuación seleccionada.
	2.2. Emitir un mensaje de error si hay algún parámetro incorrecto.

Continúa...

...Continuación

Graficar 2D	
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
4. Seleccionar Copy to clipboard.	4.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Escalar de Forma Logarítmica el Eje y	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú 2D, el submenú Log scale for y axis.	1.1. Mostrar una ventana con los parámetros de graficación.
2. Introducir los parámetros de la(s) ecuacion(es) a graficar.	<p>2.1. Registrar los parámetros de cada ecuación en una lista.</p> <p>2.2. Devolver la(s) gráfica(s) correspondiente(s).</p>
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
3. Cancelar una ecuación de la lista.	3.1. Eliminar los parámetros pertenecientes a la ecuación seleccionada.
	2.2. Se emite un mensaje de error si hay algún parámetro incorrecto.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema

Continúa...

...Continuación

Graficar 2D	
4. Seleccionar Copy to clipboard.	4.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Mostrar Barra de Errores	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú 2D, el submenú Error Bar.	1.1. Mostrar una ventana con los parámetros de graficación.
2. Introducir los parámetros de graficación.	2.1 Devolver la gráfica correspondiente.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si hay algún parámetro incorrecto.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	4.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Histograma	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú 2D, el submenú Histograma.	1.1. Mostrar una ventana con los parámetros de graficación.

Continúa...

...Continuación

Graficar 2D	
2. Introducir los parámetros de graficación.	2.1 Devolver la gráfica correspondiente.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si hay algún parámetro incorrecto.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	4.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Sección Mostrar Gráfica de Barra	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú 2D, el submenú Bar graph.	1.1. Mostrar una ventana con los parámetros de graficación.
2. Introducir los parámetros de graficación.	2.1 Devolver la gráfica correspondiente.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si hay algún parámetro incorrecto.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema

Continúa...

...Continuación

Graficar 2D	
3. Seleccionar Copy to clipboard.	4.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Puntos de extensión	

Tabla 2.8: Caso de uso Graficar 2D.

Graficar 3D	
Código del CU	CU-9
Propósito	Permitir a los usuarios graficar funciones en tres dimensiones.
Actores	Usuario
Resumen	El CU se inicia cuando el usuario selecciona el menú Plot. Inmediatamente se visualiza un menú desplegable con diferentes opciones. Seguidamente el usuario selecciona el tipo de graficación e introduce los parámetros y el sistema devuelve la gráfica correspondiente.
Referencias	R7.2.1, R7.2.2
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Plot, la opción 3D.	1.1. Visualizar un submenú con varias opciones de graficación.

Continúa...

...Continuación

Graficar 3D	
2. Seleccionar una de las opciones.	2.1. 1. Si selecciona el submenú Plot of three-dimensional surface, ir a la sección Plotear Gráficos Tridimensionales. 2. Si selecciona el submenú Contour plot of three-dimensional surface, ir a la sección Plotear Gráfico de Contornos de Superficies Tridimensionales.
Sección Plotear Gráficos Tridimensionales	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú 3D, el submenú Plot of three-dimensional surface.	1.1. Mostrar una ventana con los parámetros de graficación.
2. Introducir los parámetros de graficación.	2.1 Devolver la gráfica correspondiente.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si hay algún parámetro incorrecto.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.

Continúa...

...Continuación

Graficar 3D	
Sección Plotear Gráfico de Contornos de Superficies Tridimensionales	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú 3D, el sub-menú Contour plot of three-dimensional surface.	1.1. Mostrar una ventana con los parámetros de graficación.
2. Introducir los parámetros de graficación.	2.1 Devolver la gráfica correspondiente.
Flujo alternativo # 1	
Acción del actor	Respuesta del sistema
	2.1. Emitir un mensaje de error si hay algún parámetro incorrecto.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles para que después el usuario pueda pegarlo en sus scripts de Octave.
Puntos de extensión	

Tabla 2.9: Caso de uso Graficar 3D.

Manipular Gráfica	
Código del CU	CU-10
Propósito	Permitir a los usuarios etiquetar y establecer límites a los ejes coordenados en la graficación, nombrar y exportar las gráficas en formato PDF, EPS ó JPG.
Actores	Usuario
Resumen	El CU se inicia cuando el usuario selecciona el menú Plot. Inmediatamente se visualiza un menú desplegable diferentes opciones.
Referencias	R7.3.1, R7.3.2, R7.3.3
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar el menú Plot.	1.1. Visualizar un menú desplegable con las siguientes opciones: <ol style="list-style-type: none"> 1. Plot 2D (Plotear 2D) 2. Plot 3D (Plotear 3D) 3. Axis scale (Escalar Ejes) 4. Title and labels (Nombrar y Etiquetar) 5. Export (Exportar)

Continúa...

...Continuación

Manipular Gráfica	
2. Escoger una de las operaciones a realizar.	2.1 Ejecutar una de las siguientes acciones: <ol style="list-style-type: none"> 1. Si selecciona el submenú Plot 2D, ir a descripción Plot 2D. 2. Si selecciona el submenú Plot 3D, ir a descripción Plot 3D. 3. Si selecciona el submenú Axis scale, ir a la sección Establecer Límites de Gráficas en los Ejes Coordinados. 4. Si selecciona el submenú Title and labels, ir a la sección Nombrar y Etiquetar Gráfica. 5. Si selecciona el submenú Export, ir a la sección Exportar Gráfica.
Sección Establecer Límites de Gráficas en los Ejes Coordinados	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar el menú Plot, la opción Axis scale.	1.1. Visualizar una ventana para especificar los valores de los límites.
2. Especificar valores.	2.1. Verificar que no existan campos vacíos así como la validez de los límites, y acotar la gráfica con los valores establecidos.

Continúa...

...Continuación

Manipular Gráfica	
Flujo alternativo	
Acción del actor	Respuesta del sistema
	2.1. Mostrar en mensaje de error en caso de que se introduzcan parámetros incorrectos.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles.
Sección Nombrar y Etiquetar Gráfica	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar el menú Plot, la opción Title and labels.	1.1. Visualizar un cuadro de diálogo.
2. Especificar el título de la gráfica y las etiquetas para cada eje de coordenada.	2.1. Verificar que el campo Title no esté vacío y mostrar la gráfica con el título y las etiquetas especificados.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	2.1. Si al menos el título no es especificado, no se muestra la gráfica.
Flujo alternativo # 2	
Acción del actor	Respuesta del sistema
3. Seleccionar Copy to clipboard.	3.1. Copiar el código al porta papeles.

Continúa...

...Continuación

Manipular Gráfica	
Sección Exportar Gráfica	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú Plot, la opción Export.	1.1. Visualizar un cuadro de diálogo para exportar la gráfica.
2. Especificar nombre y formato del fichero a exportar, así como la dirección donde será guardado.	2.1. Verificar que no existan campos vacíos y guardar el fichero en la dirección indicada.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	2.1. Mostrar un mensaje de error en caso de no especificar el nombre del fichero a exportar y su extensión (PDF, PNG, JPG).
Puntos de extensión	

Tabla 2.10: Caso de uso Manipular Gráfica.

Gestionar Configuración	
Código del CU	CU-11
Propósito	Permitir a los usuarios configurar la aplicación.
Actores	Usuario
Resumen	El CU se inicia cuando el usuario selecciona el menú Plot. Inmediatamente se visualiza un menú desplegable con diferentes opciones.
Referencias	R8.1, R8.2, R8.3, R8.4
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar el menú Config.	1.1. Visualizar una interfaz con varias opciones de configuración.
2. Realizar configuración.	2.1. Guardar los cambios y mostrar un mensaje de reinicio de la aplicación.
3. Reiniciar la aplicación.	3.1. Aplicar los cambios.
Flujo alternativo	
Acción del actor	Respuesta del sistema
4. Introducir los parámetros de configuración incorrectamente.	4.1. Error en la aplicación.
Puntos de extensión	

Tabla 2.11: Caso de uso Gestionar Configuración.

Gestionar Menú	
Código del CU	CU-12
Propósito	Permitir al usuario y al administrador crear, modificar y eliminar menú.
Actores	Persona
Resumen	El CU se inicia cuando la persona copia un fichero con el código del menú para una dirección específica. Si es un administrador podrá crear el menú para todos los usuarios que utilicen el sistema; en cambio, si es un usuario, solo él podrá gestionar los menú que él cree.
Referencias	R9.1, R9.2, R9.3
Sección Crear Menú	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. En caso de ser un administrador: copiar el fichero *.menu a la dirección: /usr/local/share/qt octave/menus/ y reiniciar la aplicación. En caso de ser un usuario: copiar el fichero *.menu a la dirección: /home/usuario/.qt octave/menus/ y reiniciar la aplicación.	1.1. Mostrar el nuevo menú en la barra de menú de la aplicación.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	1.1 El menú no se muestra en caso de algún error en el código del fichero.

Continúa...

...Continuación

Gestionar Menú	
Sección Modificar Menú	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. En caso de ser un administrador: editar el fichero *.menu en la dirección: /usr/local/share/qt octave/menus/ y reiniciar la aplicación. En caso de ser un usuario: editar el fichero *.menu en la dirección: /home/usuario/.qt octave/menus/ y reiniciar la aplicación	1.1. Actualizar los cambios en dicho menú.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	1.1 El menú no se muestra en caso de algún error en el código.
Sección Eliminar Menú	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1.En caso de ser un administrador: eliminar el fichero *.menu de la dirección: /usr/local/share/qt octave/menus/ y reiniciar la aplicación. En caso de ser un usuario: eliminar el fichero *.menu de la dirección: /home/usuario/.qt octave/menus/ y reiniciar la aplicación	1.1. Actualizar los cambios y el menú eliminado desaparecerá de la barra de menú.

Continúa...

...Continuación

Gestionar Menú	
Puntos de extensión	

Tabla 2.12: Caso de uso Gestionar Menú.

Editor de Scripts para Octave	
Código del CU	CU-13
Propósito	
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario decide editar un script de octave y abre un editor desde el menú View/Dock Tools/Editor.
Referencias	R10.1, R10.2
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Abrir un editor de scripts.	1.1 Muestrar un editor de scripts.
2. Seleccionar el menú File.	Visualizar un menú desplegable con las siguientes opciones: <ol style="list-style-type: none"> 1. Open (Abrir). 2. Save as (Salvar).

Continúa...

...Continuación

Gestionar Editor de Scripts	
2. Escoger una de las siguientes acciones.	2.1. Ejecutar una de las siguientes acciones: <ol style="list-style-type: none"> 1. Si selecciona el submenú Open ir a la sección Cargar Archivos *.m. 2. Si selecciona el submenú Save as ir a la sección Guardar Archivos *.m.
Cargar Archivos *.m.	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú File la opción Open.	1.1. Mostrar un cuadro de diálogo para que el usuario busque el archivo *.m.
2. Seleccionar el archivo *.m.	2.1. Cargar el archivo en una pestaña del editor.
Flujo alternativo	
Acción del actor	Respuesta del sistema
3. Seleccionar un archivo que no es un *.m.	3.1. Cargar el archivo en una pestaña del editor.
Salvar Archivos *.m.	
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Seleccionar en el menú File la opción Save as.	1.1. Mostrar un cuadro de diálogo para que el usuario seleccione el lugar donde desea guardar el archivo.

Continúa...

...Continuación

Gestionar Editor de Scripts	
2. Seleccionar el directorio.	2.1. Salvar el archivo.
Flujo alternativo	
Acción del actor	Respuesta del sistema
3. Seleccionar un directorio en el que no tiene permisos para escribir.	3.1. Mostrar mensaje de error.
Puntos de extensión	

Tabla 2.13: Caso de uso Gestionar Editor de Scripts.

Luego de una panorámica acerca de cómo funciona el sistema, fundamentada a través del modelamiento del dominio y la descripción de los casos de uso que conforman dicho sistema, el lector cuenta con la información suficiente para adentrarse en el diseño de la aplicación.

Capítulo 3

Diseño del sistema

EL presente capítulo recoge algunos aspectos relacionados con el diseño del software, a través de diagramas y tablas, con el objetivo de brindar una comprensión en profundidad de los requisitos no funcionales, así como sentar las bases para subsiguientes actividades de implementación. Se describen además de forma concluyente, las clases que intervienen.

3.1. Diagrama de clases del diseño

El diagrama de clases del diseño no es más que una representación de la parte estática del sistema, es decir, de las clases del sistema y sus relaciones. Una forma de organizar estos diagramas, es desglosándolos en subsistemas o paquetes, los cuales pueden verse como piezas más manejables que agrupan clases o casos de uso fuertemente asociados, con el fin de lograr un mejor entendimiento del sistema en general.

Durante el análisis previo al diseño, el modelador típicamente coloca el nombre 'business' de un mensaje en la línea del mensaje. Más tarde, durante el diseño, el nombre 'business' es reemplazado con el nombre del método que está siendo llamado por un objeto. Se estimó pertinente entonces no realizar el análisis del sistema, puesto que el fin que se perseguía era ilustrar al grupo de desarrolladores que implementará versiones superiores de QtOctave, para lo cual era suficiente especificar los métodos que realizaban las acciones en cada momento de la vida del caso de uso en cuestión.

Dado el hecho de que el grupo de trabajo pudiera no entender la finalidad de algún método, por la complejidad de su implementación, se procedió entonces a la descripción de las clases implicadas en los diagramas de secuencia elaborados, donde se comentan cada uno de sus métodos.

Para ver la especificación del diagrama de diseño por paquetes consultar anexos en la página (149)

3.2. Diagrama de paquetes

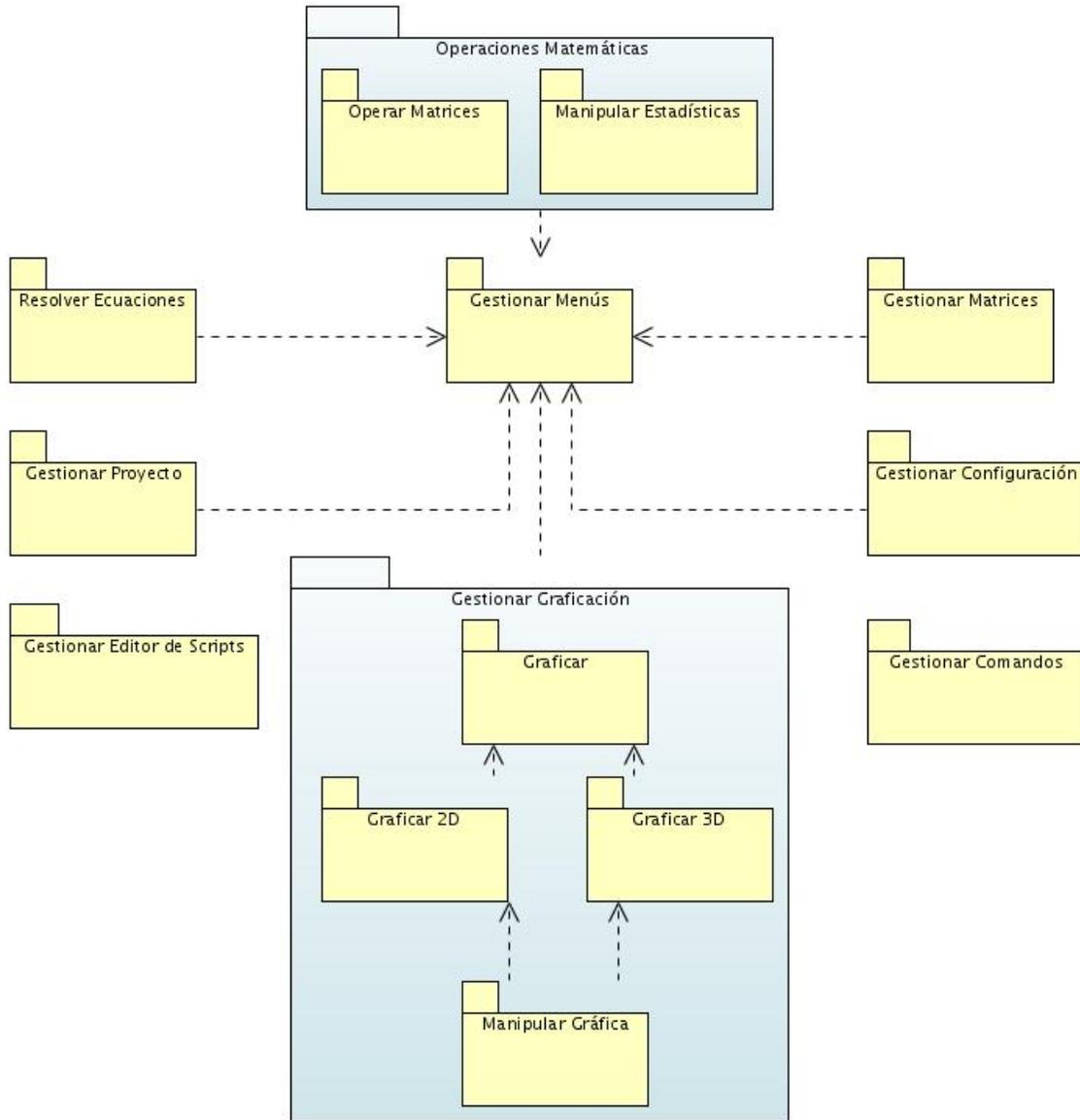


Figura 3.1: Diagrama de paquetes.

3.3. Diagramas de secuencia especificados por casos de uso

El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Este tipo de diagramas muestran la interacción de un conjunto de objetos y clases, así como los detalles de la implementación de un escenario en una aplicación a través del tiempo y se modela para cada caso de uso.

Para ver la especificación de los diagramas de secuencia por casos de uso consultar anexos en la página (130)

3.4. Descripción de las clases

Descripción de la clase Terminal	
Nombre de la Clase	Terminal
Tipo de Clase	Interfaz
Atributo	Tipo
lines_in_terminal	int
cols_in_terminal	int
octave_connection	OctaveConnection *
dynamic_help	Dynamic_help *
combo_box	Autocomplete *
session	Session *
Para cada responsabilidad:	
Nombre:	Terminal(QWidget * parent):BaseWidget(parent)
Descripción:	Crea la ventana de la terminal con la configuración correspondiente.
Nombre:	setOctaveConnection(OctaveConnection *octave_connection)
Descripción:	Establece la conexión con Octave.
Nombre:	getOctaveConnection()
Descripción:	Devuelve la conexión a Octave.
Nombre:	dragEnterEvent(QDragEnterEvent *event)
Descripción:	Evento de arrastrar.
Nombre:	dropEvent (QDropEvent * event)
Descripción:	Carga los scripts que el usuario arrastre y suelte en la terminal.
Nombre:	return_pressed()
Descripción:	Capturar el comando escrito una vez que se presione la tecla ENTER.
Nombre:	getTextEdit()
Descripción:	Devuelve el atributo text.

Continúa...

...Continuación

Descripción de la clase Terminal	
Nombre:	<code>command_enter (const QString &command)</code>
Descripción:	Enviar un comando Octave.
Nombre:	<code>write_output(QString output)</code>
Descripción:	Muestra la salida de Octave.
Nombre:	<code>write_error(QString error)</code>
Descripción:	Muestra mensajes de error en la terminal.
Nombre:	<code>write_command(QString command)</code>
Descripción:	Muestra el texto tecleado por el usuario.
Nombre:	<code>remove_lines(QTextCursor &cursor)</code>
Descripción:	Elimina líneas cuando éstas superan el máximo visible en la terminal, además de dividir aquellas que son demasiado largas.
Nombre:	<code>show_commands_callback()</code>
Descripción:	Muestra una lista de los comandos usados.
Nombre:	<code>dynamic_help_callback()</code>
Descripción:	Llamada a la ayuda dinámica.
Nombre:	<code>completion_matches_callback()</code>
Descripción:	Envía a octave el comando seleccionado de la ventana de completamiento de código.
Nombre:	<code>stop_process_callback()</code>
Descripción:	Termina la conexión con Octave.

Tabla 3.1: Descripción de la clase Terminal.

Descripción de la clase Operations	
Nombre de la Clase	Operations
Tipo de Clase	Controladora
Atributo	Tipo
main_window	MainWindow
active_widget	BaseWidget
octave_connection	OctaveConnection
matrix	QString
copy_clipboard_ok	bool
Para cada responsabilidad:	
Nombre:	Operations(QObject *parent, BaseWidget **active_widget, MainWindow *main_window):QObject(parent)
Descripción:	Establece la conexión de la interfaz gráfica con cada una de las operaciones correspondientes y que se explican a continuación.
Nombre:	setOctaveConnection (OctaveConnection* octave_connection)
Descripción:	Establece la conexión con Octave
Nombre:	setMatrix(QString matrix)
Descripción:	Cambia la matriz por otra pasapa por parámetro.
Nombre:	inverse_callback()
Descripción:	Calcula la inversa de una matriz.
Nombre:	determinant_callback()
Descripción:	Calcula el determinante de una matriz.
Nombre:	eigenvalues_callback()
Descripción:	Calcula los autovalores y autovectores.
Nombre:	transpose_callback()

Continúa...

...Continuación

Descripción de la clase Operations	
Descripción:	Calcula la transpuesta de una matriz.
Nombre:	submatrix_callback()
Descripción:	Halla la submatriz de una matriz.
Nombre:	input_dialog(QString title, QString label, QString &input)
Descripción:	Muestra en cuadro de diálogo con un edit para escribir los comandos que serán ejecutados. Devuelve verdadero si no está vacío el edit al presionar el botón OK del cuadro de diálogo, en otro caso devuelve falso.
Nombre:	simple_comand(QString title, QString label, QString _command)
Descripción:	Procesa el resultado del método: input_dialog().
Nombre:	menu_window(QString title, QStringList parameters_labels, QStringList ¶meters, QStringList output_labels, QStringList &output, QString help=QString(), bool accept_blank_parameters=false)
Descripción:	Muestra un cuadro de diálogo con los datos recibidos por parámetro. Devuelve el tipo de acción que realice el usuario(OK, CANCEL, COPY TO CLIPBOARD).
Nombre:	compex_comand(QString title, QString _command, QStringList parameters_labels, QStringList ¶meters, QStringList output_labels, QStringList &output, QString help=QString(), bool is_table=false, bool accept_blank_parameters=false)
Descripción:	Procesa el resultado del método: menu_window().
Nombre:	mean_callback()
Descripción:	Calcula la media de un vector específico.
Nombre:	median_callback()

Continúa...

...Continuación

Descripción de la clase Operations	
Descripción:	Calcula la mediana de un vector específico.
Nombre:	std_callback()
Descripción:	Calcula la desviación estándar de un vector específico.
Nombre:	cov_callback()
Descripción:	Calcula la covarianza.
Nombre:	corrcoef_callback()
Descripción:	Calcula el coeficiente de relación de un vector específico.
Nombre:	plot_callback()
Descripción:	Llamada el submenú Plot de graficación. Muestra una cuadro de diálogo donde el usuario escribirá los datos para la graficación. Luego envía los comandos a Octave.
Nombre:	semilogy_callback()
Descripción:	Llamada el submenú Semilogy de graficación. Muestra una cuadro de diálogo donde el usuario escribirá los datos para la graficación. Luego envía los comandos a Octave.
Nombre:	semilogx_callback()
Descripción:	Llamada al submenú Semilogx de graficación. Muestra una cuadro de diálogo donde el usuario escribirá los datos para la graficación. Luego envía los comandos a Octave.
Nombre:	log_log_callback()
Descripción:	Llamada al submenú Log_log de graficación. Muestra una cuadro de diálogo donde el usuario escribirá los datos para la graficación. Luego envía los comandos a Octave.
Nombre:	polar_callback()

Continúa...

...Continuación

Descripción de la clase Operations	
Descripción:	Llamada al submenú Polar de graficación. Muestra una cuadro de diálogo donde el usuario escribirá los datos para la graficación. Luego envía los comandos a Octave.
Nombre:	errorBars_callback()
Descripción:	Llamada al submenú (Error Bar) de graficación. Muestra una cuadro de diálogo donde el usuario escribirá los datos para la graficación. Luego envía los comandos a Octave.
Nombre:	axis_callback()
Descripción:	Llamada al submenú (Axis scale) de graficación. Muestra un cuadro de diálogo con los campos correspondientes a los parametros para la graficación. Luego envía los comandos a OCTave.
Nombre:	title_label_callback()
Descripción:	Permite etiquetar las gráficas. Muestra un cuadro de diálogo con los campos correspondientes a las etiquetas y envia los datos a Octave.
Nombre:	show_projects_dialog()
Descripción:	Abre un nuevo proyecto a petición del usuario.
Nombre:	editor_callback()
Descripción:	Muestra el editor a petición del usuario.
Nombre:	open_m_file_callback()
Descripción:	Abre un script en el editor.
Nombre:	cd_callback()
Descripción:	Envia a Octave el comando de navegación (cd) seguido de la dirección.
Nombre:	exit_callback()
Descripción:	Cerrar la ventana principal.

Continúa...

...Continuación

Descripción de la clase Operations	
Nombre:	contour_callback()
Descripción:	Llamada al submenú (Contour plot of three-dimensional surface). Crea una ventana donde el usuario introduce los datos correspondientes a la gráfica.
Nombre:	bar_callback()
Descripción:	Llamada al submenú (Bar graph). Crea una ventana donde el usuario introduce los datos correspondientes a la gráfica.
Nombre:	hist_callback()
Descripción:	Llamada al submenú (Histogram). Crea una ventana donde el usuario introduce los datos correspondientes a la gráfica.
Nombre:	general_config_callback()
Descripción:	Llamada al submenú (Config). Crea una ventana donde el usuario realiza las configuraciones que desea.

Tabla 3.2: Descripción de la clase Operations.

Descripción de la clase OctaveConnection	
Nombre de la Clase	OctaveConnection
Tipo de Clase	Controladora
Atributo	Tipo
lineno	int
colno	int
debugging	bool
octave_path	QString
line_buffer	QString
error_buffer	QBuffer
Para cada responsabilidad:	
Nombre:	OctaveConnection(QObject * parent):QProcess(parent)
Descripción:	Establece la conexión con Octave a través de señales.
Nombre:	setOctavePath()
Descripción:	Cambia el path de Octave.
Nombre:	getOctavePath()
Descripción:	Devuelve el path de Octave.
Nombre:	startOctave(bool quiet)
Descripción:	Inicia Octave.
Nombre:	command_enter (const QString &command, bool show)
Descripción:	Envía comandos a Octave, estos son mostrados o no en el terminal dependiendo del segundo parámetro.
Nombre:	octaveErrorOutputSlot()
Descripción:	Este método se va a ejecutar cuando Octave emita la señal: ready-ReadStandardError().
Nombre:	readyReadStandardOutput()

Continúa...

...Continuación

Descripción de la clase OctaveConnection	
Descripción:	Este método se va a ejecutar cuando Octave emita la señal: ready-ReadStandardOutput().
Nombre:	octaveFinished(int, QProcess::ExitStatus)
Descripción:	Este método se va a ejecutar cuando Octave emita la señal: finished(int, QProcess::ExitStatus).

Tabla 3.3: Descripción de la clase OctaveConnection.

Descripción de la clase MainWindow	
Nombre de la Clase	MainWindow
Tipo de Clase	Interfaz
Atributo	Tipo
QAction *	actionOpen
QAction *	actionRunFile
QAction *	actionOctave_help
QAction *	actionTable
QAction *	actionEditor
QAction *	actionProjects
QAction *	actionOpenMFile
QAction *	actionCompletionMatches
QAction *	actionDynamicHelp
QAction *	actionStopProcess
QAction *	actionClearTerminal
QAction *	actionVariableList

Continúa...

...Continuación

Descripción de la clase MainWindow	
QAction *	actionNavigator
QAction *	actionGeneralConfig
QAction *	actionInverse
QAction *	actionDeterminant
QAction *	actionEigenvalues
QAction *	actionTranspose
QAction *	actionSubmatrix
QAction *	actionMean
QAction *	actionMedian
QAction *	actionStd
QAction *	actionCov
QAction *	actionCorrcoef
QAction *	actionPlot
QAction *	actionPolar
QAction *	actionSemilogy
QAction *	actionSemilogx
QAction *	actionLogLog
QAction *	actionAxis
QAction *	actionTitleLabel
QMenu *	menu2DPlot
QMenuBar *	menubar
QMenu *	menuFile
QMenu *	menuData
QMenu *	menuMatrix
QMenu *	menuStatistics

Continúa...

...Continuación

Descripción de la clase MainWindow	
QMenu *	menuPlot
QMenu *	menuConfig
QMenu *	menuHelp
QMenu *	menuView
QMenu *	menuDocks
QMenu *	menuMode
QMdiArea *	work_space
QAction *	actionModeWorkspace
QAction *	actionModeDock
QAction *	actionModeSDI
QString *	modeWorkArea
QToolBar *	toolBarDocks
QToolBar *	toolBarRun
QToolBar *	toolBarMatrix
QToolBar *	toolBarHelp
QStatusBar *	statusbar
Navigator *	nav
QDockWidget *	dockListVar
QDockWidget *	dockNavigator
QDockWidget *	dockEditor
QDockWidget *	dockTerminal
QDockWidget *	dockCommandHistory
QMap<QString, QAction *>	actions
QMap<QString, QMenu *>	menus
OctaveConnection *	octave_connection

Continúa...

...Continuación

Descripción de la clase MainWindow	
Session *	session
QMessageBox *	closeMessage
QMap<QString, QString>	windowSettings
Para cada responsabilidad:	
Nombre:	MainWindow(OctaveConnection *oc, QWidget *parent):QMainWindow(parent)
Descripción:	Crea la ventana principal de la aplicación con la configuración correspondiente.
Nombre:	createActions()
Descripción:	Crea las acciones de la ventana principal, les asocia un icono y/o un texto descriptivo en caso necesario.
Nombre:	createMenus()
Descripción:	Crea los menú File, Data, Matrix, Statistics y Plot, de la ventana principal, con sus respectivas acciones asociadas.
Nombre:	createMenuView()
Descripción:	Crea el menú View, con sus acciones asociadas.
Nombre:	createStatusBar()
Descripción:	Crea la barra de estado.
Nombre:	createToolBars()
Descripción:	Crea la barra herramientas, con sus acciones asociadas.
Nombre:	createAction(QString action_name, QString name, QString icon)
Descripción:	Crea una nueva acción con un nombre y un icono recibido por parámetro.
Nombre:	createMenu(QString menu_name, QString name, QString icon)

Continúa...

...Continuación

Descripción de la clase MainWindow	
Descripción:	Crea un nuevo menu con un icono y un nombre pasado por parámetro.
Nombre:	clear_menu()
Descripción:	Limpia la barra de menú.
Nombre:	show_config_help_menus()
Descripción:	Crea el menú Config con sus acciones asociadas.
Nombre:	createDockWindows()
Descripción:	Crea las ventanas correspondientes al modo de trabajo Dock.
Nombre:	setModeSDI()
Descripción:	Establece SDI como modo de trabajo.
Nombre:	setModeWorkSpace()
Descripción:	Establece WorkSpace como modo de trabajo.
Nombre:	setModeDock()
Descripción:	Establece Dock como modo de trabajo.
Nombre:	closeEvent(QCloseEvent *event)
Descripción:	Cerrar QtOctave.
Nombre:	showSetVisibleObjects()
Descripción:	Llama al método createMenuView().
Nombre:	hideSetVisibleObjects()
Descripción:	Limpia el menú View.

Tabla 3.4: Descripción de la clase MainWindow.

Descripción de la clase Main	
Nombre de la Clase	Main
Tipo de Clase	Controladora
Atributo	Tipo
MainWindow *	main_window
QMdiArea *	work_space
WindowList *	window_list
Operations *	operations
Terminal *	terminal
Session	session
VariableList *	variableList
Para cada responsabilidad:	
Nombre:	Main(QObject * parent):QObject (parent)
Descripción:	Es el método más importante de la aplicación. Aquí se crea la ventana principal con todas sus ventanas hijas, se crea la conexión a Octave, se crea la terminal con sus respectivas conexiones, se establece el modo de trabajo inicial, se cargan los menú desde los ficheros, entre otros detalles.
Nombre:	widget_activated(BaseWidget *w)
Descripción:	Llamada de retorno para cambiar active_widget, el parámetro w es el widget activado.
Nombre:	help_octave()
Descripción:	Muestra la ayuda de Octave.
Nombre:	help_qtoctave()
Descripción:	Muestra la ayuda de QtOctave.
Nombre:	table(QString text)

Continúa...

...Continuación

Descripción de la clase Main	
Descripción:	Muestra la tabla en forma de hoja de cálculo para crear o editar las matrices.
Nombre:	run_file()
Descripción:	Corre un Octave script (*.m; *.M) mediante un cuadro de diálogo.
Nombre:	variable_list()
Descripción:	Crea la lista de variables que han sido declaradas y la muestra en dependencia del modo de trabajo actual.
Nombre:	windows_layout()
Descripción:	Si el modo de trabajo Workspace está activado, crea una capa de ventanas y la muestra.
Nombre:	setVisibleNavigator()
Descripción:	Muestra el navegador en dependencia del modo de trabajo.
Nombre:	mainWindowWidget()
Descripción:	Devuelve el widget de la ventana principal.
Nombre:	main(int argn, char *argv[])
Descripción:	Inicia la aplicación QtOctave.

Tabla 3.5: Descripción de la clase Main.

Descripción de la clase GenerateMenu	
Nombre de la Clase	GenerateMenu
Tipo de Clase	Controladora
Atributo	Tipo
path	QString
operations	Operations *
mainwindow	MainWindow *
Para cada responsabilidad:	
Nombre:	GenerateMenu(MainWindow *mainwindow, Operations *operations):QObject(mainwindow)
Descripción:	Constructor de la clase.
Nombre:	setPath(QString path)
Descripción:	Especifica la ruta para cargar los menú.
Nombre:	menu_path(QString file_path)
Descripción:	Elimina, de la cadena file_path, la subcadena correspondiente al atributo path.
Nombre:	find_icon(QString file_path, QString menu)
Descripción:	Devuelve la dirección del icono correspondiente al menú pasado por parámetro. Devuelve una cadena vacía si no encuentra el icono.
Nombre:	load_menu()
Descripción:	Carga los menú que están en path.
Nombre:	load_menu(QString dir_name, QMenu *parent_menu)
Descripción:	Crea un menú desde la dirección especificada en los parámetros. Conecta, además, el menú con la ventana principal de la aplicación.

Continúa...

...Continuación

Descripción de la clase GenerateMenu	
Nombre:	process_menu_file(QString _file, QStringList & input_labels, QStringList & input_parameters, QStringList & output_labels, QStringList & output_parameters, QString & command, QString & menu_name, QString & help, bool & accept_blank_parameters)
Descripción:	Parsea los ficheros *.menu, devuelve falso en caso de algún error, de lo contrario devuelve verdadero.

Tabla 3.6: Descripción de la clase GenerateMenu.

Descripción de la clase MenuCallBack	
Nombre de la Clase:	MenuCallBack
Tipo de Clase:	Controladora
Atributo	Tipo
process	QProcess
menu_name	QString
output	QString
octave_connection	OctaveConnection *
Para cada responsabilidad:	
Nombre:	MenuCallBack(QString menu_name, OctaveConnection *oc)
Descripción:	Constructor de la clase.
Nombre:	actions_callback()
Descripción:	Se ejecuta al seleccionar un menú ejecutable.
Nombre:	readyReadStandardOutput()
Descripción:	Este método se ejecuta cuando Octave emite la señal: readyReadStandardOutput().

Continúa...

...Continuación

Descripción de la clase MenuCallBack	
Nombre:	finished(int /*exitCode*/, QProcess::ExitStatus /*exitStatus*/)
Descripción:	Este método se ejecuta cuando Octave emite la señal: finished(int, QProcess::ExitStatus).

Tabla 3.7: Descripción de la clase MenuCallBack.

Descripción de la clase MenuExtCallBack	
Nombre:	MenuExtCallBack
Tipo de Clase:	Controladora.
Atributo	Tipo
oc	OctaveConnection *
input	QList<InputWidget*>
output	QList<InputWidget*>
help	QString
command	QString
output_box	QGroupBox *
input_box,	QGroupBox *
input_area	QWidget *
output_area	QWidget *
accept_blank_parameters	bool
ok_button	QPushButton *
copy_clipboard_button	QPushButton *
help_button	QPushButton *
cancel_button	QPushButton *

Continúa...

...Continuación

Descripción de la clase MenuExtCallBack	
Para cada responsabilidad:	
Nombre:	MenuExtCallBack(QWidget *parent):QWidget(parent)
Descripción:	Construye la ventana del menú.
Nombre:	process_menu_file(QString _file)
Descripción:	Carga el fichero *.menu que recibe como parámetro y lo ejecuta.
Nombre:	ok_button_callback()
Descripción:	Se ejecuta cuando se presiona el boton OK en el cuadro de diálogo correspondiente a un menú de tipo *.menu. Envía los datos a Octave y cierra el cuadro de diálogo.
Nombre:	copy_clipboard_button_callback()
Descripción:	Se ejecuta cuando el usuario presiona el boton Copy clipboard en el cuadro de diálogo correspondiente a un menú de tipo *.menu. Copia en el portapapeles el comando en correspondiente a la acción del menú y cierra el cuadro de diálogo.
Nombre:	cancel_button_callback()
Descripción:	Se ejecuta cuando el usuario presiona el boton Cancel en el cuadro de diálogo correspondiente a un menú de tipo *.menu. Cierra el cuadro de diálogo.
Nombre:	help_button_callback()
Descripción:	Muestra la ayuda de los menú.
Nombre:	actions_callback()
Descripción:	Muestra la ventana del menú.
Nombre:	setOctaveConnection(OctaveConnection *oc)
Descripción:	Establece una conexión con Octave
Nombre:	addInput(InputWidget *input)

Continúa...

...Continuación

Descripción de la clase MenuExtCallBack	
Descripción:	Adiciona un widget de tipo input a la capa de widgets. Luego muestra la lista de widgets.
Nombre:	addOutput(InputWidget *input)
Descripción:	Adiciona un widget de tipo output a la capa de widgets. Luego muestra la lista de widgets
Nombre:	addHelp(QString help)
Descripción:	Cambia el atributo help.
Nombre:	addCommand(QString command)
Descripción:	Cambia el atributo command.
Nombre:	setAcceptBlankParameters(bool accept_blank_parameters)
Descripción:	Cambia el atributo accept_blank_parameters.
Nombre:	generate_command()
Descripción:	Genera los comandos que serán enviados a Octave.

Tabla 3.8: Descripción de la clase MenuExtCallBack.

Descripción de la clase Table	
Nombre:	Table
Tipo de Clase:	Controladora
Atributo	Tipo
table_form	TableForm *
octave_connection	OctaveConnection *
matrix	QString
menu	QMenu *

Continúa...

...Continuación

Descripción de la clase Table	
model	ComplexNumberTableModel *
Para cada responsabilidad:	
Nombre:	Table(QWidget * parent):BaseWidget(parent)
Descripción:	Crea la tabla para editar las matrices, así como las acciones asociadas a cada componente de la tabla.
Nombre:	void setOctaveConnection(OctaveConnection *octave_connection)
Descripción:	Establece una conexión con Octave.
Nombre:	void setMatrix(QString matrix)
Descripción:	Cambiar la matriz.
Nombre:	void change_rows()
Descripción:	Cambia una fila de la matriz.
Nombre:	void change_cols()
Descripción:	Cambia una columna de la matriz.
Nombre:	void build_menu()
Descripción:	Construye el menú para trabajar con la matriz y además el menú Plot para la graficación.
Nombre:	void reloadCell(int row, int col)
Descripción:	Recargar una celda de la hoja de cálculo donde se edita la matriz.
Nombre:	void contextMenuEvent (QContextMenuEvent * event)
Descripción:	Registra un punto de evento del menú y hace el (popup) correspondiente.
Nombre:	void windowActivated (QWidget * w)
Descripción:	Activar la ventana dado un widget.
Nombre:	void windowActivated()
Descripción:	Activar la ventana de la tabla.

Continúa...

...Continuación

Descripción de la clase Table	
Nombre:	void plot(TablePlot::Type)
Descripción:	Plotear si el usuario lo decide en el QDialog y le pasan a Octave según el tipo de TablePlot el comando en cuestión que cojan del mismo.
Nombre:	void plotPlot()
Descripción:	Plotear
Nombre:	void plotPolar()
Descripción:	Plotear en sistemas de coordenadas polares.
Nombre:	void plotLogXandY()
Descripción:	Plotear de forma logarítmica con respecto a los ejes x y y de forma simultánea. a
Nombre:	void plotLogY()
Descripción:	Plotear de forma logarítmica con respecto al eje y.
Nombre:	void plotLogX()
Descripción:	Plotear de forma logarítmica con respecto al eje y.
Nombre:	void plotBar()
Descripción:	Plotear barra ya sea gráfica o de errores.
Nombre:	void line_ready(QString line)
Descripción:	Gestionan las líneas de la matriz mediante un fichero de solo lectura que una vez terminado el proceso destruyen.
Nombre:	void cellChanged (int row, int col, QString value)
Descripción:	Cambiar el valor de una celda.
Nombre:	void rows_changed()
Descripción:	Llaman al método change_rows() y recargan la ventana de de la tabla.

Continúa...

...Continuación

Descripción de la clase Table	
Nombre:	void cols_changed()
Descripción:	Llaman al método change_cols() y recargan la ventana de de la tabla.
Nombre:	void order_changed()
Descripción:	Llaman a los métodos change_rows() y change_cols() y recargan la ventana de la tabla.
Nombre:	void copy_cb()
Descripción:	Copiar al portapapeles.
Nombre:	void copy_matrix_cb()
Descripción:	Copiar matriz al portapapeles.
Nombre:	void paste_cb()
Descripción:	Pegar del portapapeles.
Nombre:	void delete_rows_cb()
Descripción:	Borrar fila, verificación incluida.
Nombre:	void delete_columns_cb()
Descripción:	Borrar columna, verificación incluida.
Nombre:	void insert_column_right_cb()
Descripción:	Insertar una columna a la derecha.
Nombre:	void insert_column_left_cb()
Descripción:	Insertar una columna a la izquierda.
Nombre:	void insert_row_down_cb()
Descripción:	Insertar una fila debajo.
Nombre:	void insert_row_up_cb()
Descripción:	Insertar una fila arriba.

Tabla 3.9: Descripción de la clase Table.

Descripción de la clase Autocomplete	
Nombre:	Autocomplete
Tipo de Clase:	Controladora
Atributo	Tipo
word_list	QStringList
first_match, current_match	QStringList::const_iterator
search_string	QString
tab_flag	bool
commands_entered	QStringList
popup_options	WordList *
octave_connection	OctaveConnection *
Para cada responsabilidad:	
Nombre:	bool event(QEvent *event)
Descripción:	Manejar los eventos del presionado de teclas en el área de comandos.
Nombre:	QString search(QString start)
Descripción:	Buscar el comando que machee con la palabra escrita en el área de comandos y devolverlo.
Nombre:	QString get_next()
Descripción:	Buscar el próximo comando que machee.
Nombre:	void keyPressEvent (QKeyEvent * event)
Descripción:	Validar los eventos por los movimientos de las teclas Up y Down.
Nombre:	void load_from_file(const char *file)
Descripción:	Cargar comando desde la lista de comandos.
Nombre:	void add(QString word)
Descripción:	Adicionar un comando a la lista de comandos.
Nombre:	void remove(QString word)

Continúa...

...Continuación

Descripción de la clase Autocomplete	
Descripción:	Eliminar un comando de la lista de comandos
Nombre:	void clear()
Descripción:	Limpiar la lista de comandos.
Nombre:	QStringList commands()
Descripción:	Retorna los comandos entrados por el usuario.
Nombre:	void set_octave_connection(OctaveConnection *oc)
Descripción:	Establecer conexión con Octave.
Nombre:	void add_completion_match(QString line)
Descripción:	Adicionar comando macheado del autocompletamiento.
Nombre:	void new_command_entered(QStringList list)
Descripción:	Señal que se emite cuando se introduce un comando nuevo.

Tabla 3.10: Descripción de la clase Autocomplete.

Descripción de la clase Projects	
Nombre:	Projects
Tipo de Clase:	Controladora
Atributo	Tipo
project_name	QString
ui	Ui_Projects
context_menu	QMenu *
new_project, modify_project, delete_project	QAction *
Para cada responsabilidad:	
Nombre:	void fill_projects_list()

Continúa...

...Continuación

Descripción de la clase Projects	
Descripción:	Mostrar una lista de los proyectos existentes en un ListView.
Nombre:	static QString navigatorPath(QString project_name)
Descripción:	Retornar un string con la última dirección del navegador de archivos de un proyecto específico.
Nombre:	static void saveNavigatorPath(QString project_name, QString path)
Descripción:	Salvar la última dirección del navegador de archivos de un proyecto específico.
Nombre:	static QStringList listFiles(QString project_name)
Descripción:	Retornar el nombre de un proyecto.
Nombre:	static void saveListFiles(const QString & project_name, const QStringList & files)
Descripción:	Salvar la lista de archivos de un proyecto.
Nombre:	static QStringList listCommandHistory(QString project_name)
Descripción:	Retornar el historial de comandos de un proyecto.
Nombre:	static void saveListCommandHistory(const QString & project_name, const QStringList & command_history)
Descripción:	Salvar el historial de comandos de un proyecto.
Nombre:	QString projectName()
Descripción:	Retornar el nombre de un proyecto.
Nombre:	void new_button_callback()
Descripción:	Mostrar un ventana (dialog) para crear un proyecto nuevo.
Nombre:	void modify_project_callback()
Descripción:	Mostrar un ventana (dialog) para modificar un proyecto.
Nombre:	void delete_project_callback()

Continúa...

...Continuación

Descripción de la clase Projects	
Descripción:	Eliminar un proyecto seleccionado.
Nombre:	void activate_project_callback(QListWidgetItem * item)
Descripción:	Activar un proyecto.
Nombre:	void show_description_callback(QListWidgetItem * current, QListWidgetItem * previous)
Descripción:	Mostrar la descripción de un proyecto.
Nombre:	void contextMenu_cb(const QPoint & pos)
Descripción:	Mostrar menú constectual.

Tabla 3.11: Descripción de la clase Projects.

Cumpliendo con los objetivos de la ingeniería inversa, se ha hecho una evaluación del código fuente, con el fin de comprender y extraer información de diseño de los procedimientos existentes, con los niveles de abstracción correspondientes. Finalmente, se cuenta con una especificación acertada del diseño del sistema. Dicha especificación está compuesta por los modelos de diseño que describen los datos, la arquitectura, las interfaces y los componentes; o sea, se ha recuperado el diseño de QtOctave a partir de su código original.

Conclusiones

Es la obra fecunda cuando no se claudica en el empeño de lograr las metas que se trazan al comienzo, cuando existe constancia en el propósito y firmeza en las ideas, cuando hay aspiraciones, cuando quedan esperanzas, cuando existen sueños. La realización de éste trabajo satisfizo en su totalidad los objetivos trazados, pudiéndose destacar de manera general las conclusiones siguientes:

1. Se confeccionó una documentación del diseño del frontend QtOctave en su versión 0.7.1, que constituye un importante material de apoyo para futuras implementaciones, en aras de desarrollar versiones superiores.
2. Se detectaron un conjunto de errores, deficiencias funcionales, así como estructurales durante el estudio del diseño de QtOctave, los cuales quedaron reflejados en los anexos del documento.
3. Se definió a Octave como el software de cálculo numérico, sobre sistema operativo GNU/Linux, con mayores potencialidades a nivel mundial y como el principal sustituto de su homólogo propietario Matlab.
4. Se fundamentó el hecho de optar por QtOctave como frontend rector, tras el estudio detallado de las aplicaciones de esta naturaleza más representativas disponibles hoy.

Recomendaciones

EL desarrollo de versiones superiores de QtOctave es un aporte considerable a la calidad de la enseñanza en la UCI, al Ministerio de Educación Superior y al movimiento de software libre internacional. Por lo tanto, en pos de fomentar éste importante desarrollo genuino, se recomienda:

1. Una reimplementación total de la herramienta que posibilite una mejor adaptación a las condiciones específicas de la UCI.
2. Poner a disposición de la comunidad de desarrollo los resultados del presente trabajo.

Referencias

- [1] R. H. Gila, “*Un ábaco de hace 20.000 años.*” <http://revistasacitameta.blogspot.com/2008/01/un-baco-de-hace-20000-aos.html>, Enero 2008.
- [2] E. F. R. John J O’Connor, “*Thales of Miletus.*” <http://www-history.mcs.st-andrews.ac.uk/Biographies/Thales.html>, Enero 2008.
- [3] E. F. R. John J O’Connor, “*Archimedes of Syracuse.*” <http://www-history.mcs.st-andrews.ac.uk/Biographies/Archimedes.html>, Enero 2008.
- [4] E. F. R. John J O’Connor, “*Euclid of Alexandria.*” <http://www-history.mcs.st-andrews.ac.uk/Biographies/Euclid.html>, Enero 2008.
- [5] L. Fernández, “*Abacus.*” <http://www.ee.ryerson.ca/~elf/abacus/history.html>, Enero 2008.
- [6] A. B. Patricio Barros, “*Abaco Chino.*” <http://fisicarecreativa.net/swanpan/>, Enero 2008.
- [7] MacTutor, “*William Oughtred.*” <http://www-history.mcs.st-andrews.ac.uk/Biographies/Oughtred.html>, Enero 2008.
- [8] J. P. C. C. José Miguel Zapata García, *Historia de la preinformática: De los primeros guarismos a la máquina de Leibniz.* p.
- [9] F. de Ciencias da Universidade de Lisboa, “*Blaise Pascal.*” <http://www.educ.fc.ul.pt/docentes/opombo/seminario/matematicos.htm>, Enero 2008.

- [10] R. Alonzo, *La calculadora*, vol. 62. Enero 2008.
- [11] C. Vilensky, *Las primeras calculadoras de teclado*. Febrero 2008.
- [12] E. Huélamo, “*Un alarde de precisión y miniaturización.*” <http://www.xnumber.com/xnumber/calculo-mecanico.htm>, Febero 2008.
- [13] R. J. Jacobson I., Booch G., *El Proceso Unificado de Desarrollo del Software*. 2000.
- [14] T. S. S. Culebro Juárez M., Gómez Herrera W., *Software libre vs software propietario*. 2006.
- [15] Definicion.org, “*GPL.*” <http://www.definicion.org/gpl>, Marzo 2008.
- [16] RedHat, “*Biblioteca.*” <http://www.redhat.com/docs/manuals/linux/RHL-5.2-Manual/install-guide/doc137.html>, Marzo 2008.
- [17] B. C. A. of Pennsylvania, “*API.*” <http://www.pcta.com/about/glossary.php>, Marzo 2008.
- [18] Alegsa, “*Libreria de Desarrollo Qt.*” <http://www.alegsa.com.ar/Dic/qt.php>, Marzo 2008.
- [19] T. Mathworks, “*Matlab.*” <http://www.mathworks.com>, Marzo 2008.
- [20] G. Borrell i Nogueras, *Octave: Una alternativa real a Matlab a coste cero*. 2007.
- [21] M. E. H., *Introducción a Scilab*. 2008.
- [22] SourceForge, “*KOctave.*” <http://sourceforge.net/projects/koctave>, Abril 2008.
- [23] K. Straight, “*GOctave.*” <http://www.advogato.org/proj/goctave/>, Mayo 2008.
- [24] SourceForge, “*Octavede.*” <http://sourceforge.net/projects/octavede>, Abril 2008.
- [25] SourceForge, “*Yaog.*” <http://sourceforge.net/projects/yaog>, Abril 2008.

- [26] S. Loisel, “*Octave Worksop.*” <http://www.ics.es.yamanashi.ac.jp/mirror/octave-workshop/>, Mayo 2008.
- [27] P. L. L. Rosado, “*QtOctave un Front-End para Octave.*” <http://qtoctave.wordpress.com/>, Mayo 2008.
- [28] R. Pressman, *Ingenieria del Software, un enfoque práctico*. 5ta ed., 1997.

Bibliografía

- [1] Escobar, H. M. M. *Introducción a SCILAB*. Bogotá, Departamento de Matemáticas, Universidad Nacional de Colombia, 2005. p.
- [2] García, J. M. Z. & Cañizarez, J. P. C. *Historia de la preinformática: De los primeros guarismos a la máquina de Leibniz*. p.
- [3] Jacobson, I.; Booch, G. & Rumbaugh, J. *El Proceso Unificado de Desarrollo de Software*. 2000. 464 p. 84-7829-036-2
- [4] Jiménez, A. J. SCILAB. *Computación numérica bajo Linux y Windows*. Universidad de Cádiz, p.
- [5] Juárez, M. C. Herrera, W. G. G. & Sánchez, S. T. *Software Libre vs. Software Proprietario. Ventajas y desventajas*. México, 2006. p.
- [6] Nogueras, G. B. i. *Octave: Una alternativa real a Matlab a coste de cero*. 7 p.
- [7] Pérez, M. d. l. O. G.; Saus, E. A. & Pérez, E. G. *Sistemas de Computación Algebraicos. Evolución y Aplicaciones*. Universidad de Castilla-La Mancha, p.
- [8] Pressman, R. S. *Ingeniería del Software. Un enfoque práctico*. Quinta Edición. Mc. Graw Hill, 614 p.
- [9] Rebollo, T. C.; Vera, E. C. & Mármol, M. G. *Notas sobre Matlab y Octave*. 29 p.
- [10] *SCILAB Home Page*. <http://www.scilab.org/>, Marzo 2008.

- [11] *Un paseo por la historia de las matemáticas*. 1997. <http://ciencia.astroseti.org/matematicas/articulo.php?num=3521>, Marzo 2008.

Anexos

Anexo 1: Especificación del diagrama de clases del diseño por paquetes

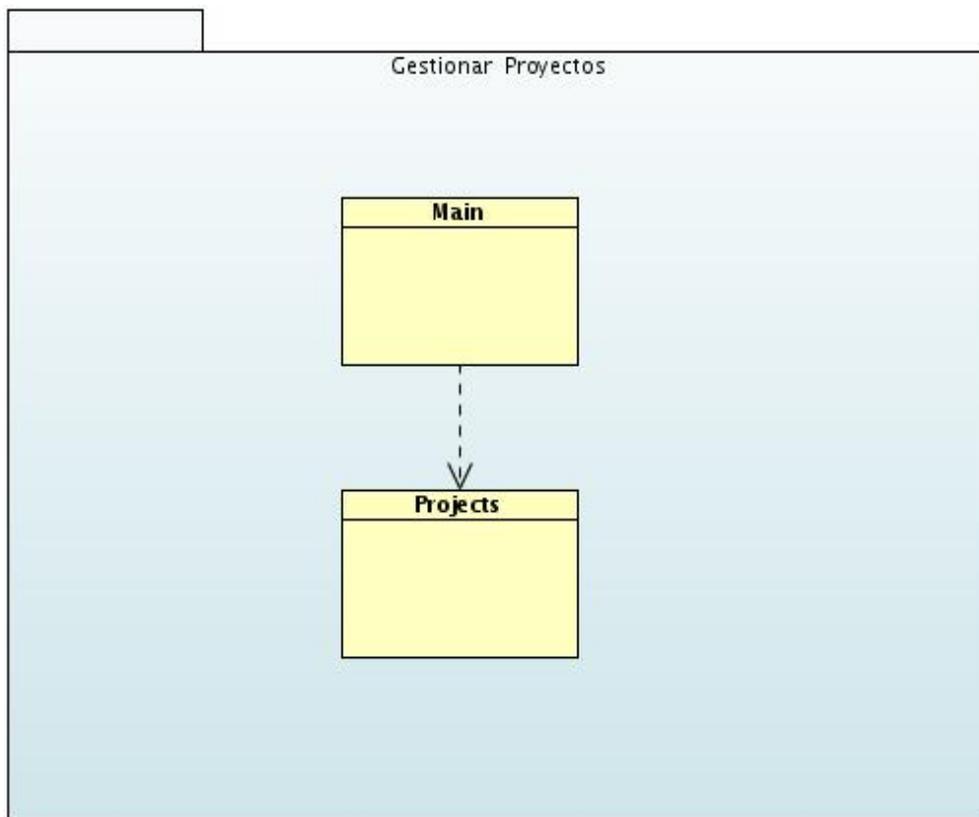


Figura 3.2: Gestionar Proyectos.

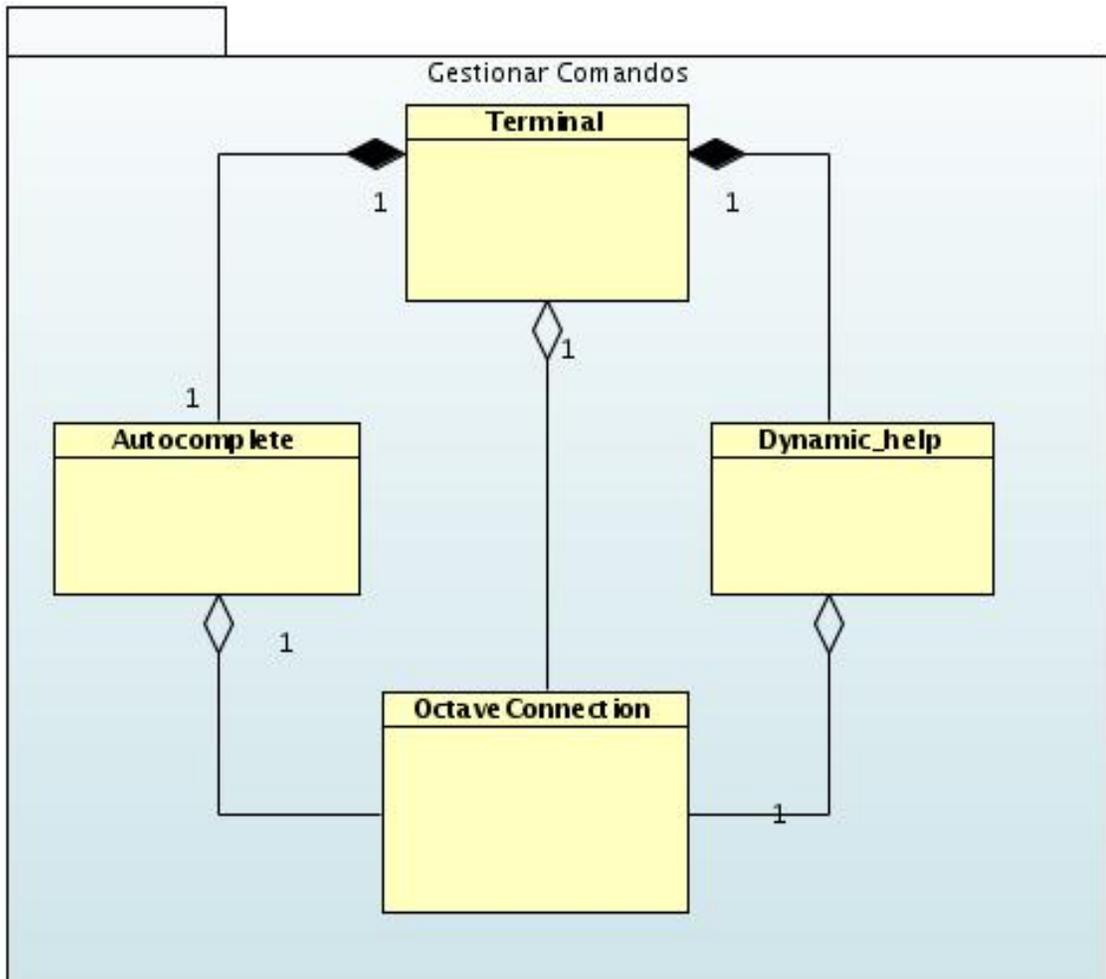


Figura 3.3: Gestionar Comandos.

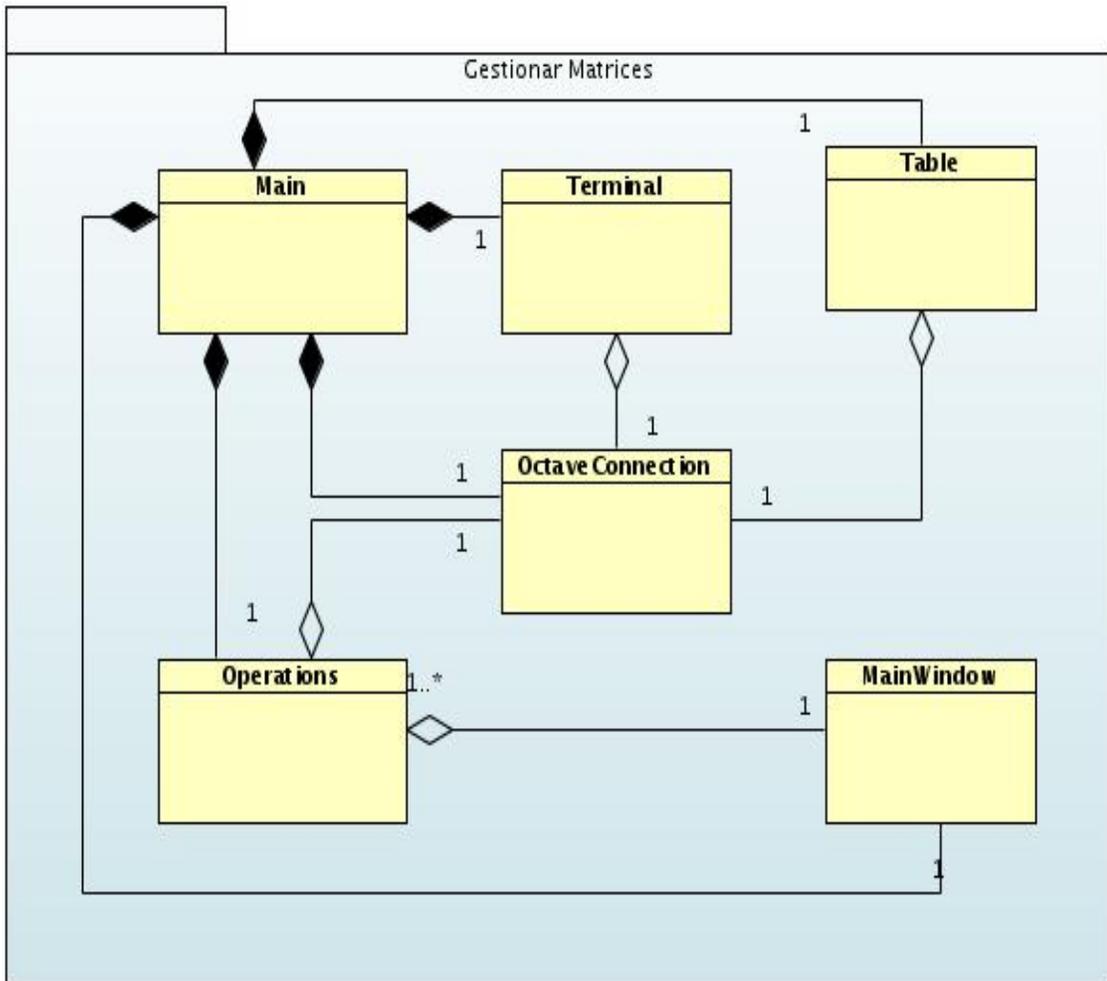


Figura 3.4: Gestionar Matrices.

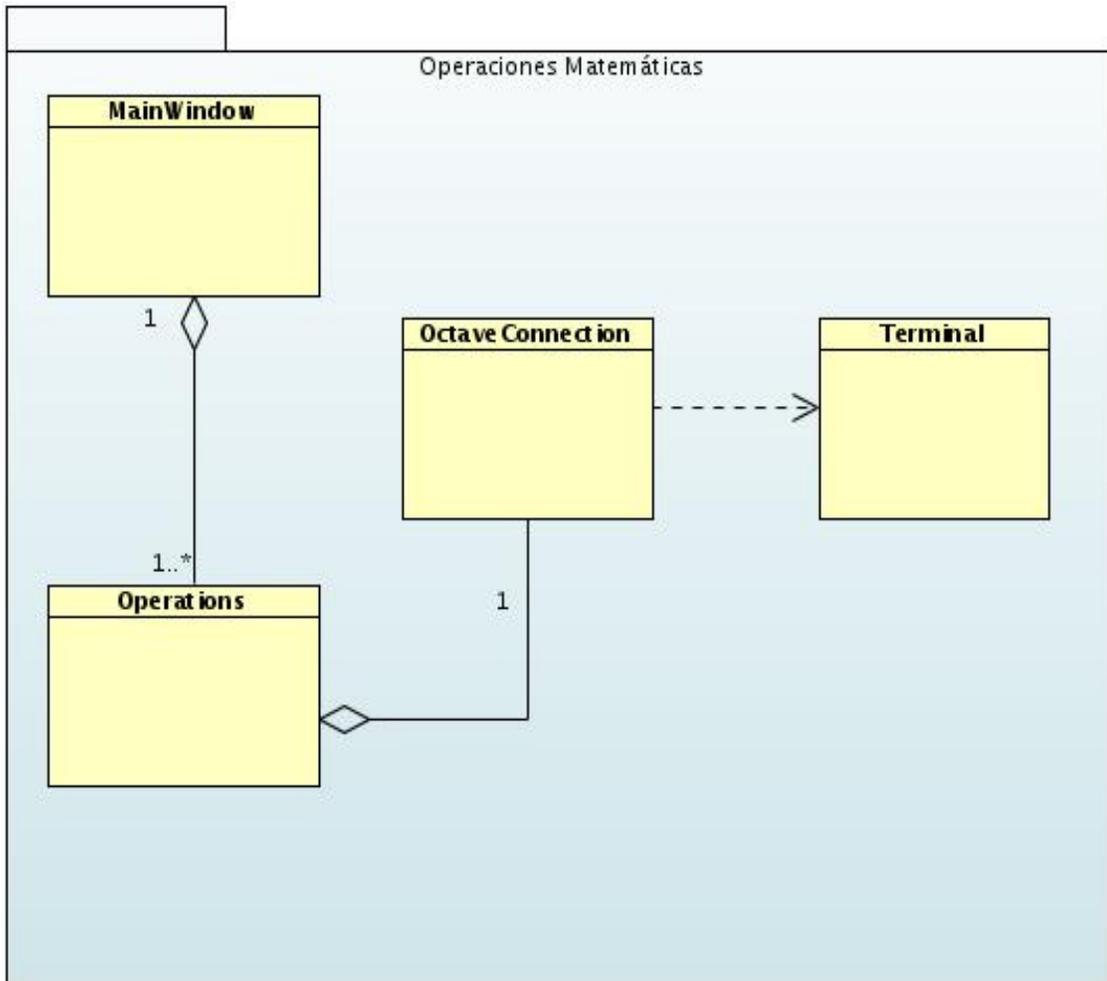


Figura 3.5: Operaciones Matemáticas.

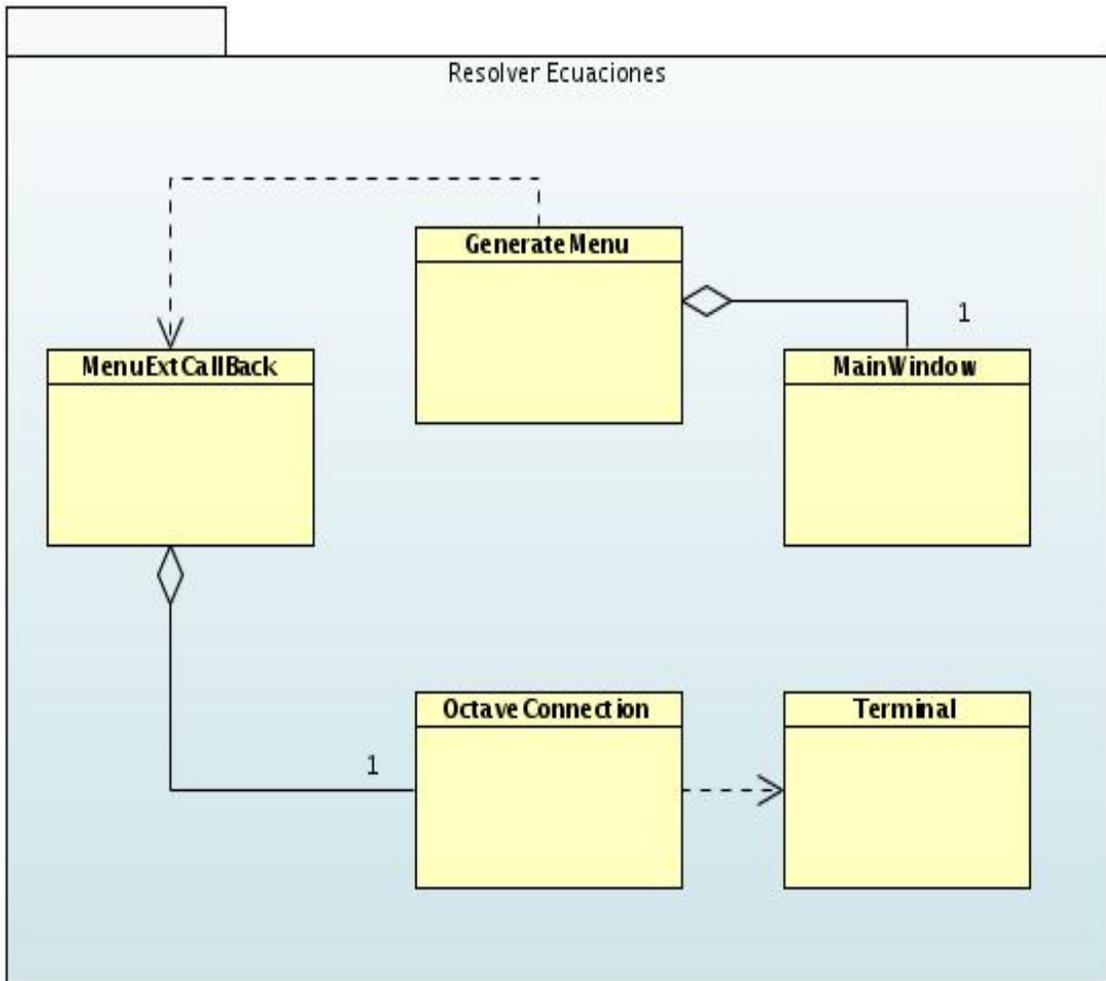


Figura 3.6: Resolver Ecuaciones.

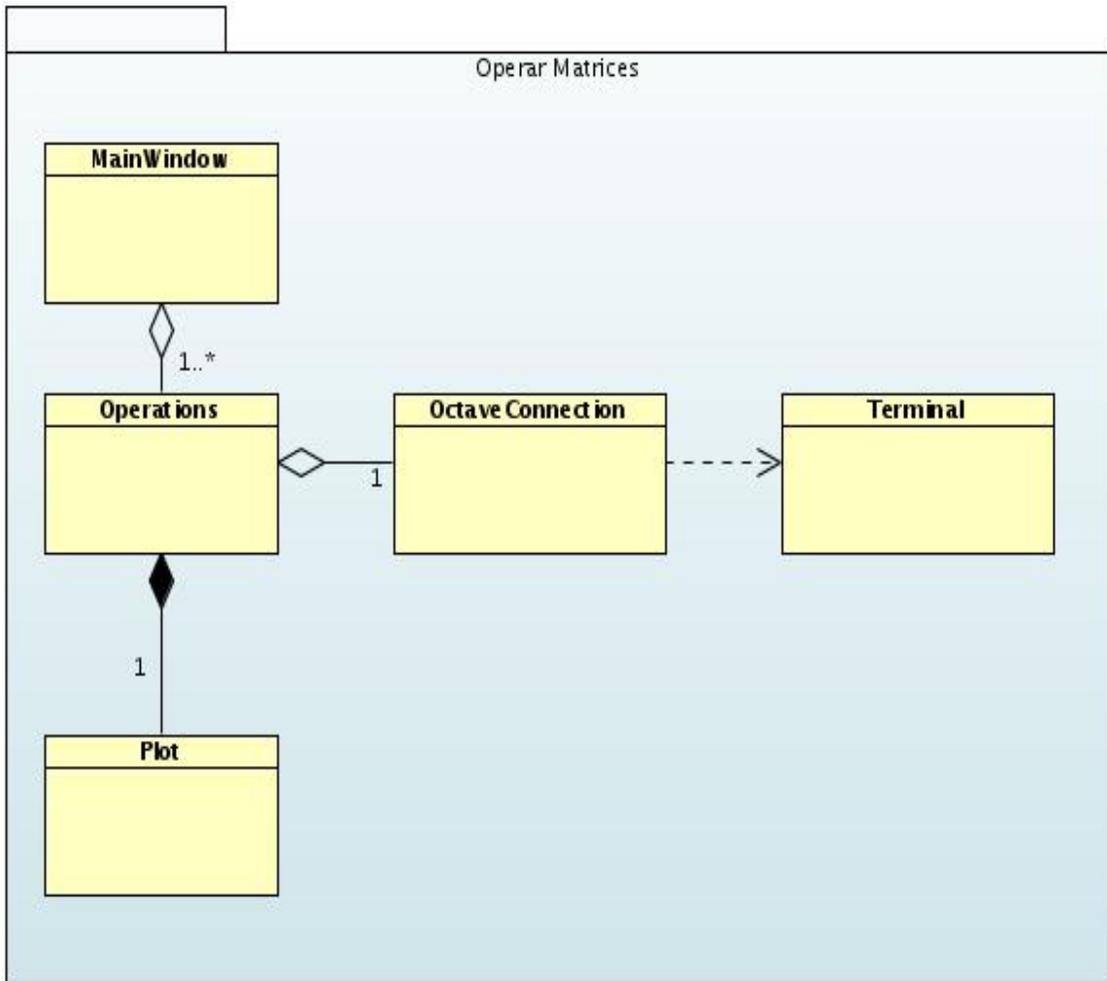


Figura 3.7: Gestionar Graficación.

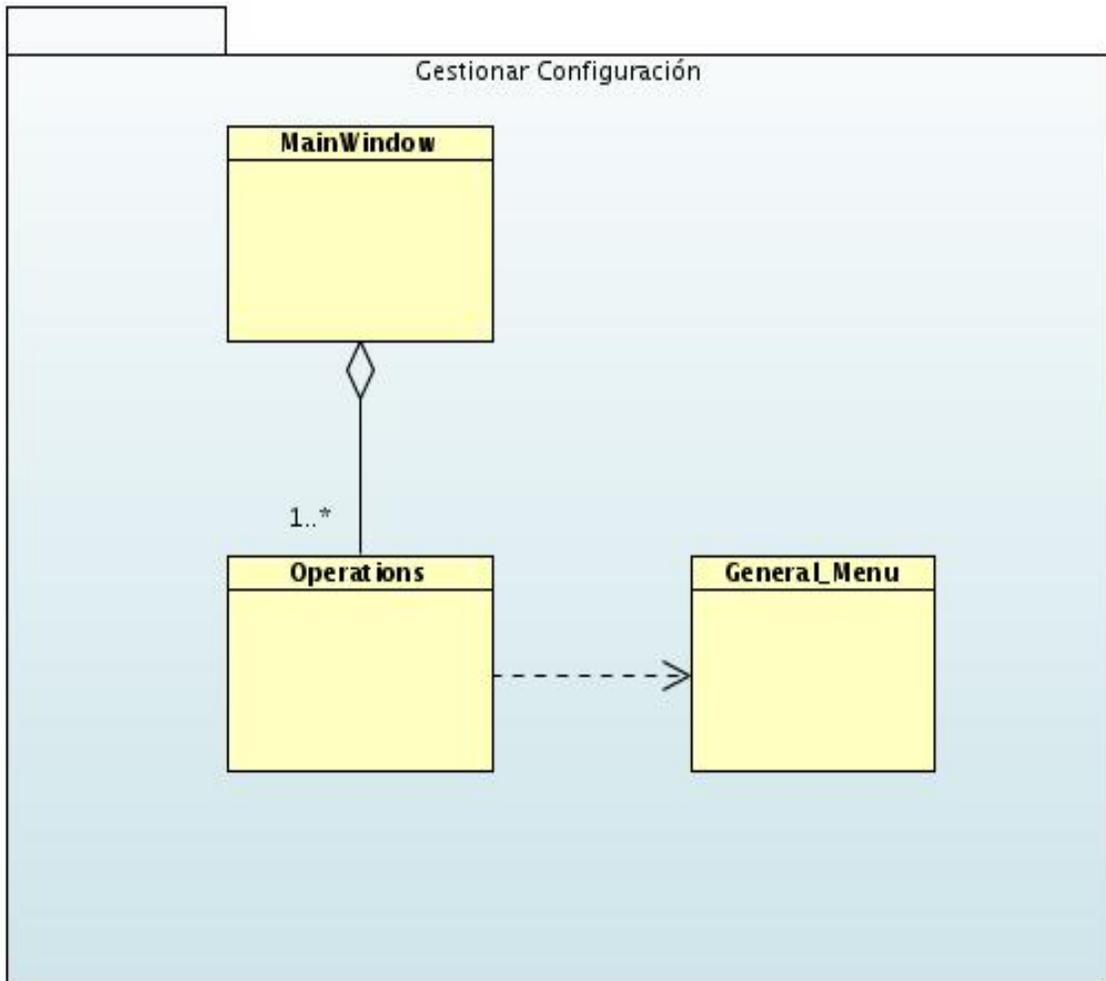


Figura 3.8: Gestionar Configuración.

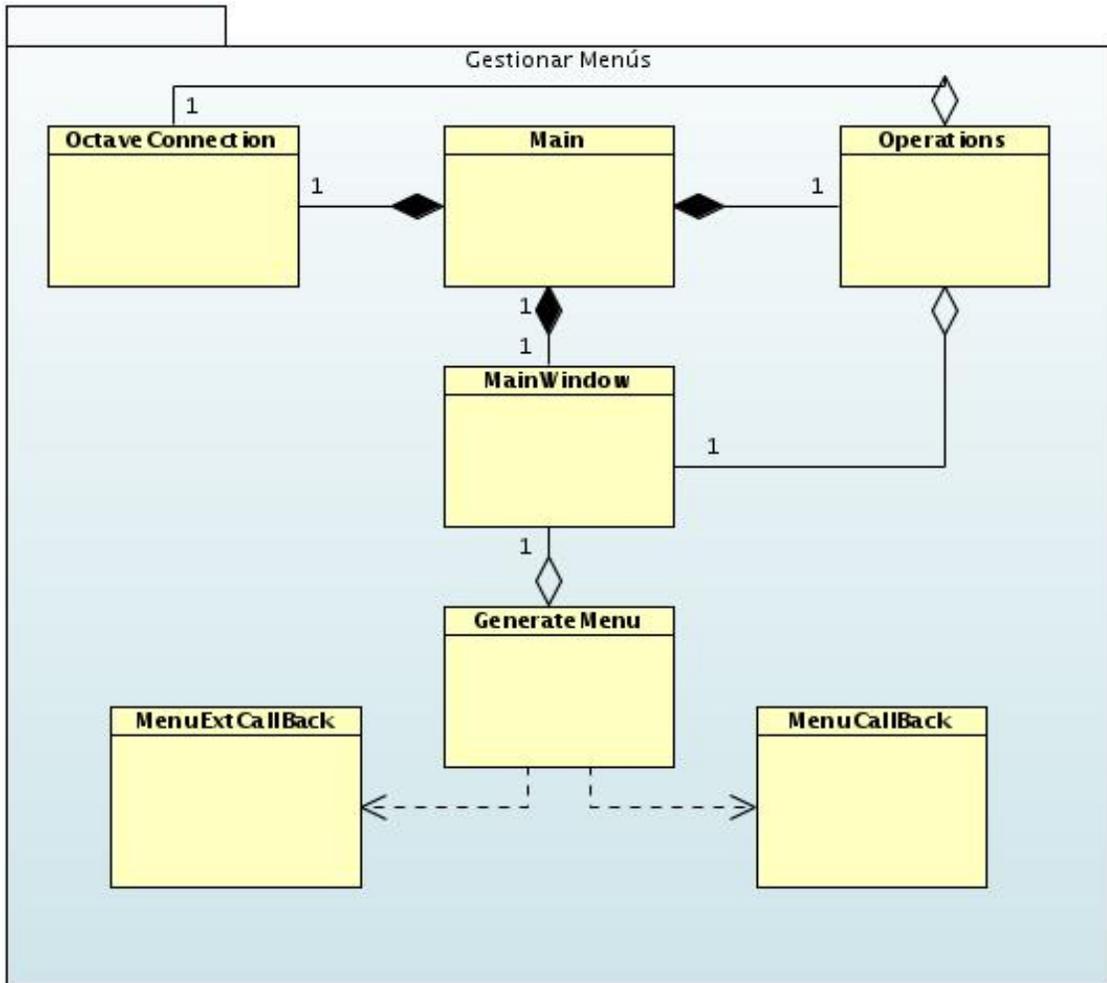


Figura 3.9: Gestionar Menús.

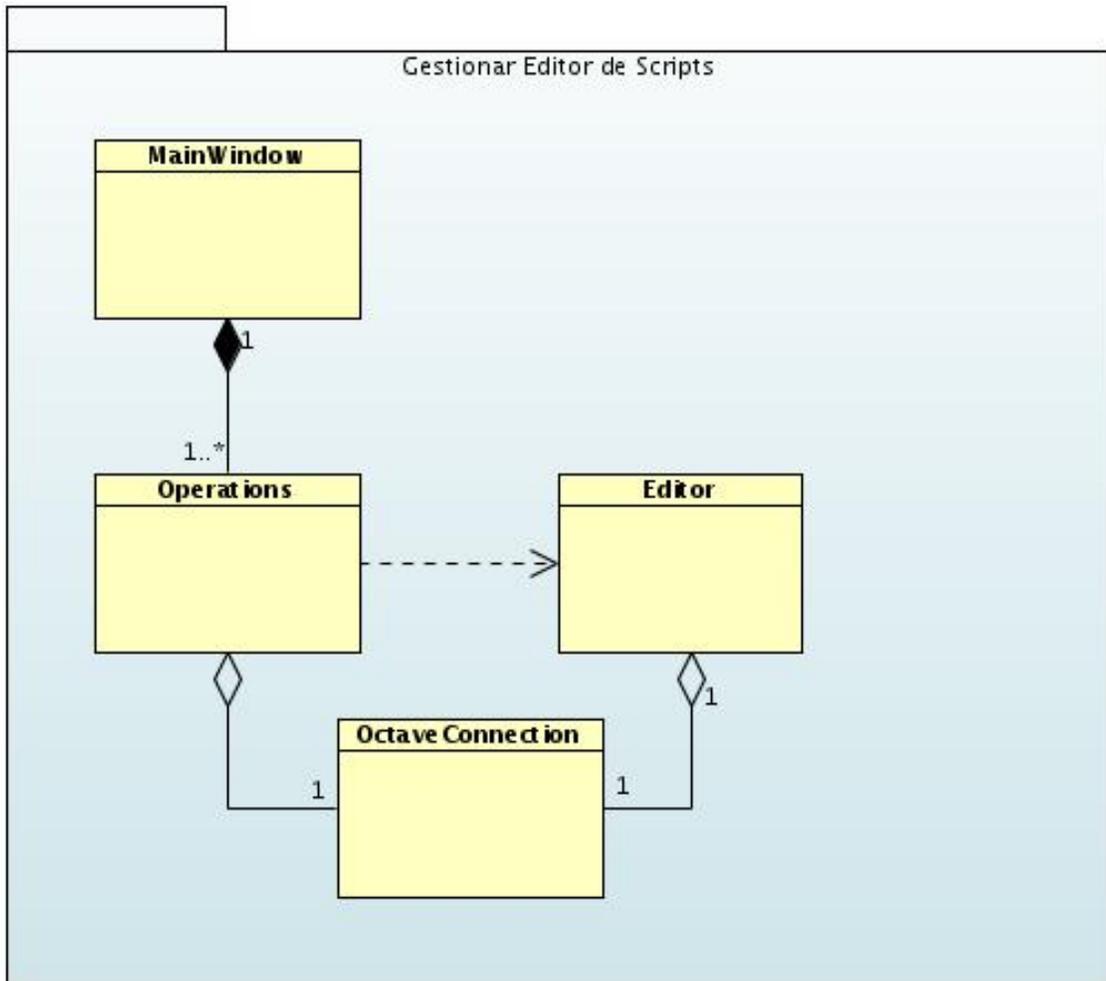


Figura 3.10: Gestionar Editor de Scripts.

Anexo 2: Diagramas de secuencia del diseño especificados por paquetes

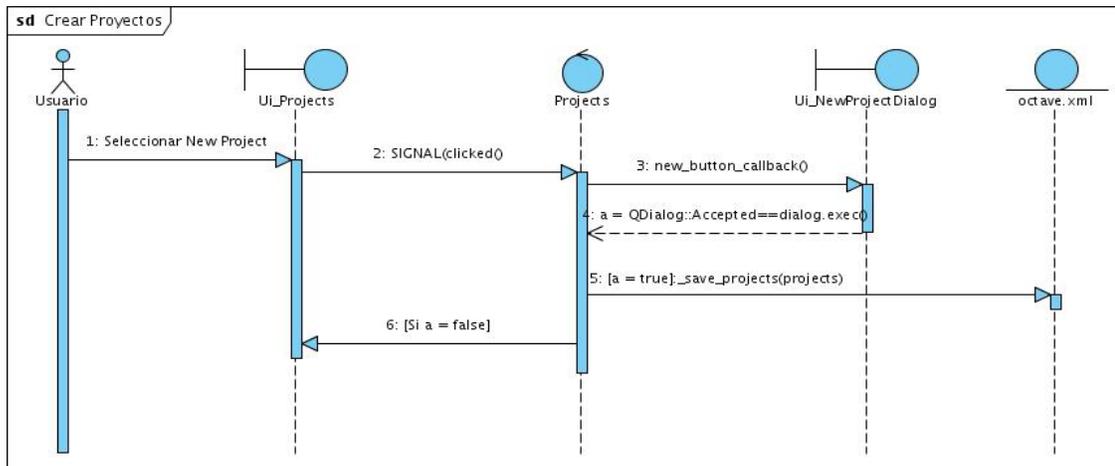


Figura 3.11: Crear Proyectos.

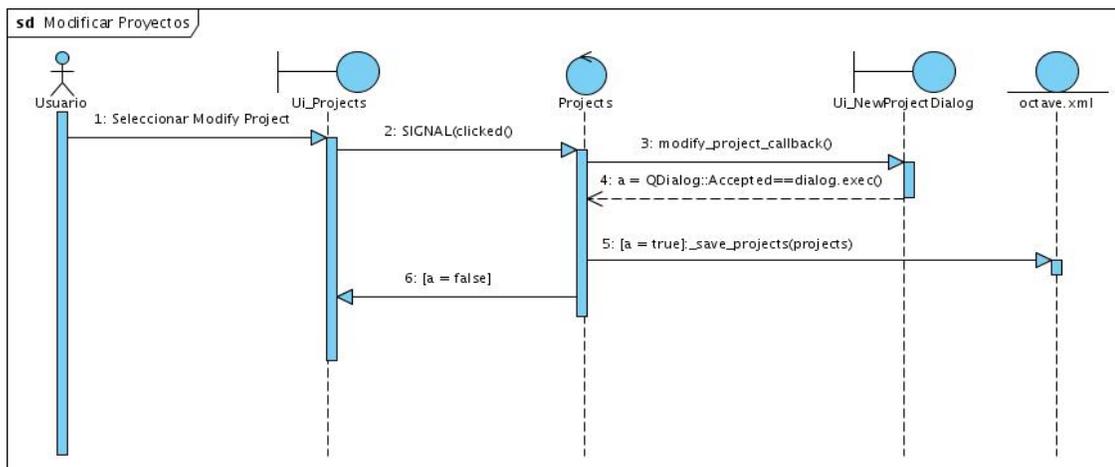


Figura 3.12: Modificar Proyectos.

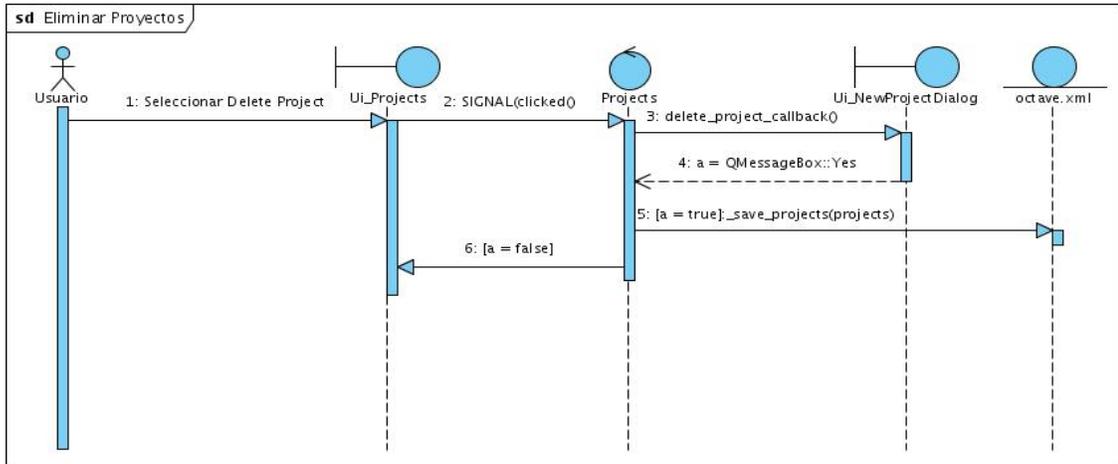


Figura 3.13: Eliminar Proyectos.

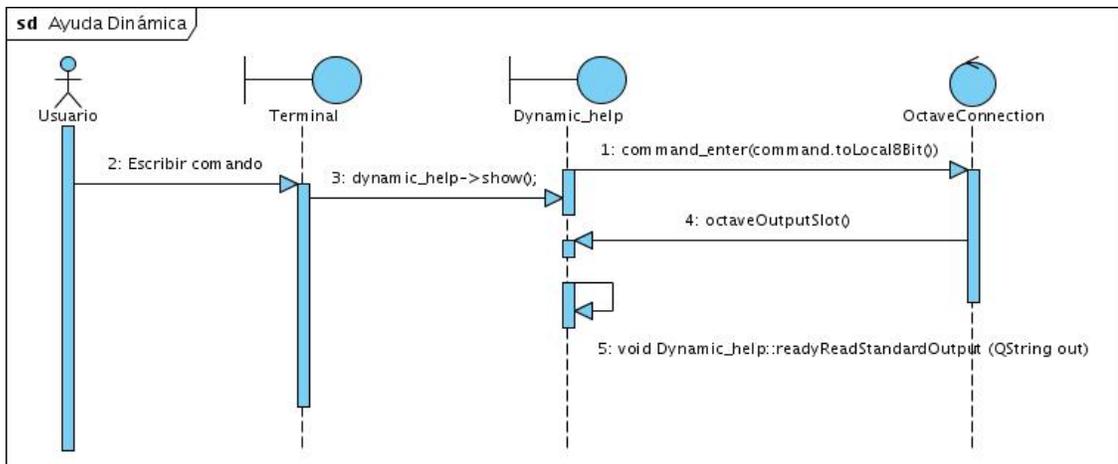


Figura 3.14: Ayuda Dinámica.

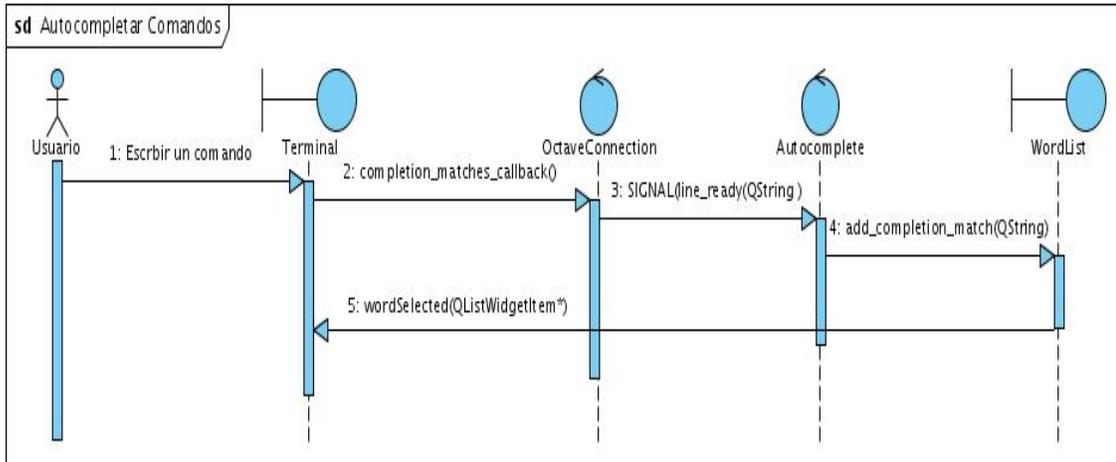


Figura 3.15: Autocompletar Comandos.

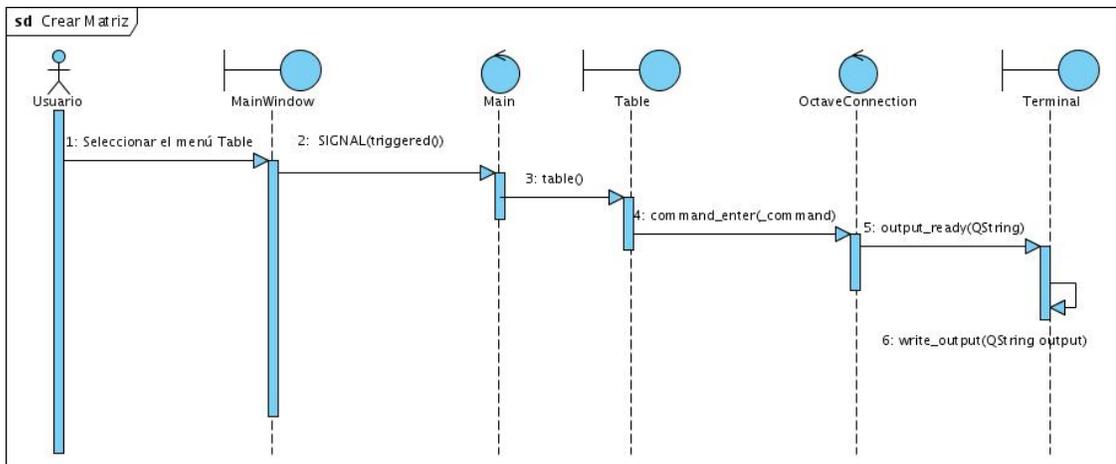


Figura 3.16: Crear Matriz.

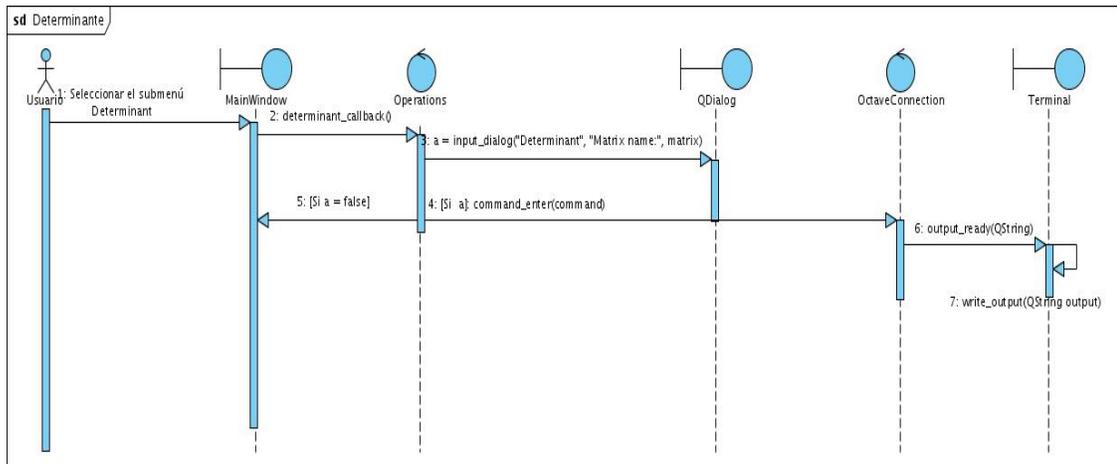


Figura 3.17: Determinante.

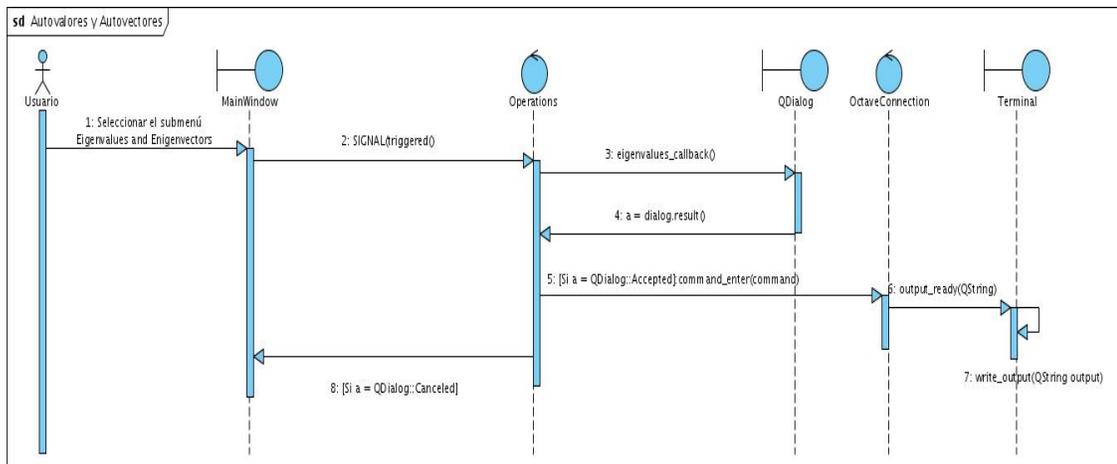


Figura 3.18: Autovalores y Autovectores.

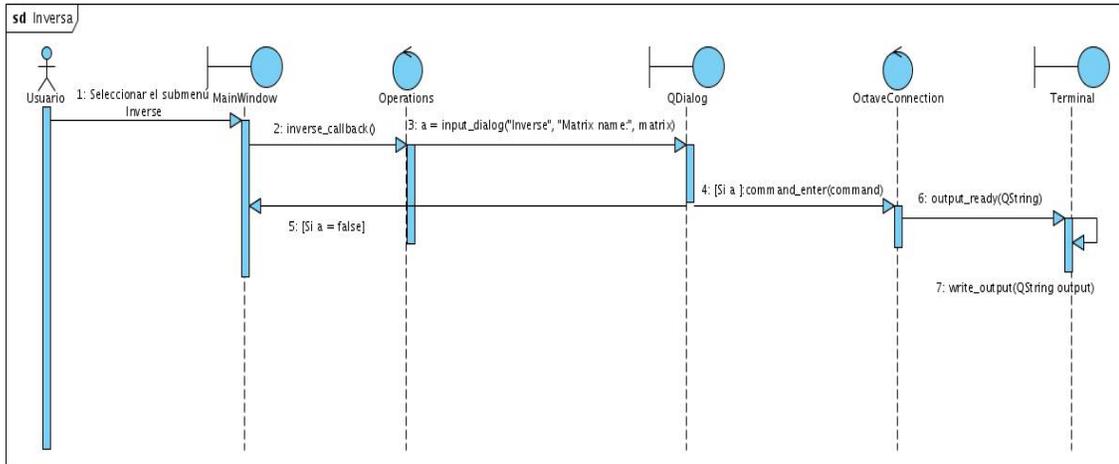


Figura 3.19: Inversa.

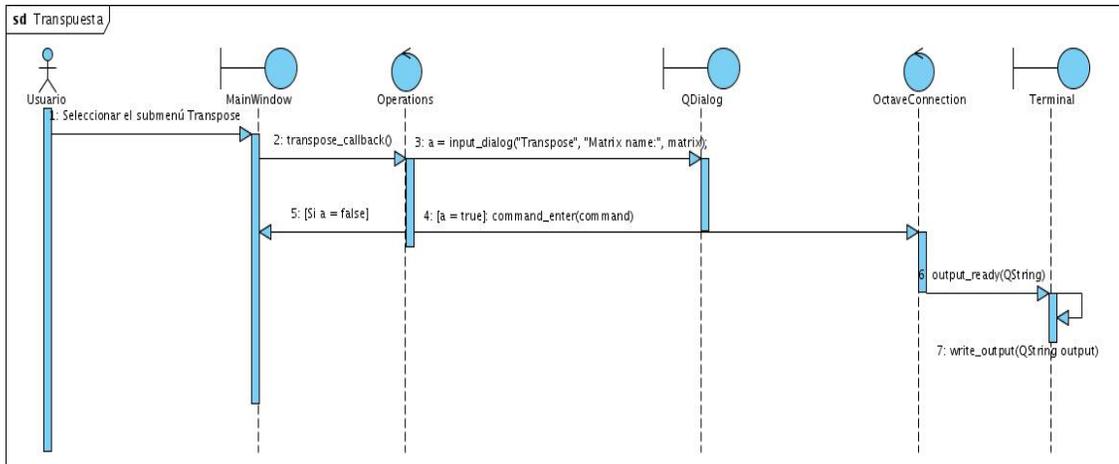


Figura 3.20: Transpuesta.

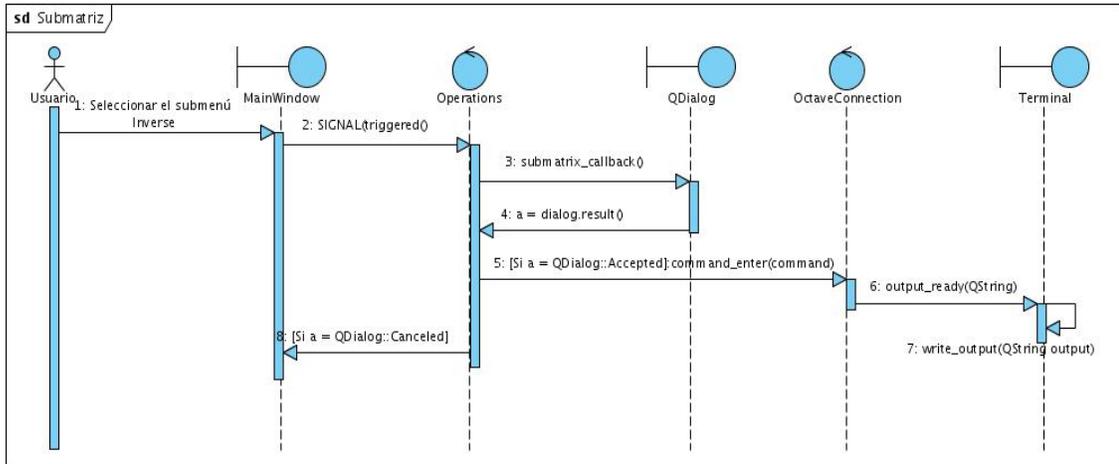


Figura 3.21: Submatriz.

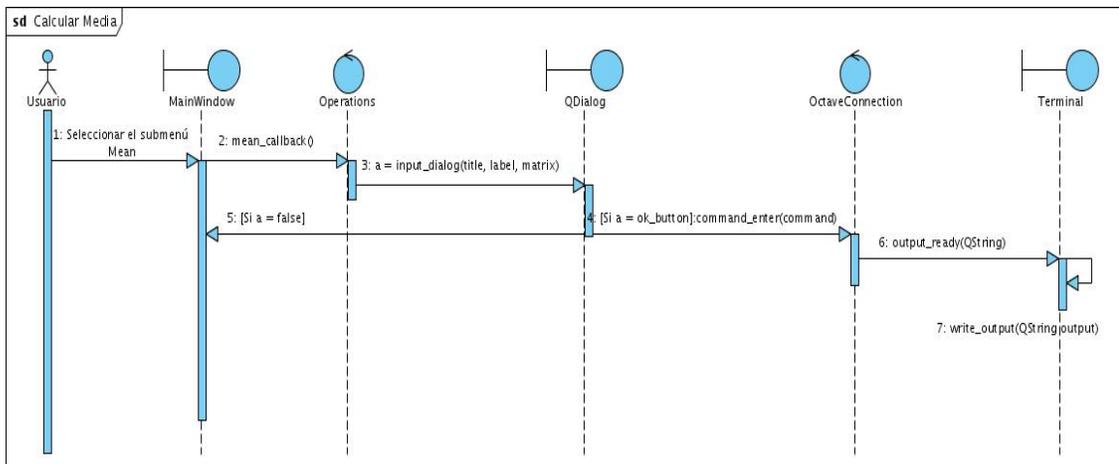


Figura 3.22: Calcular Media.

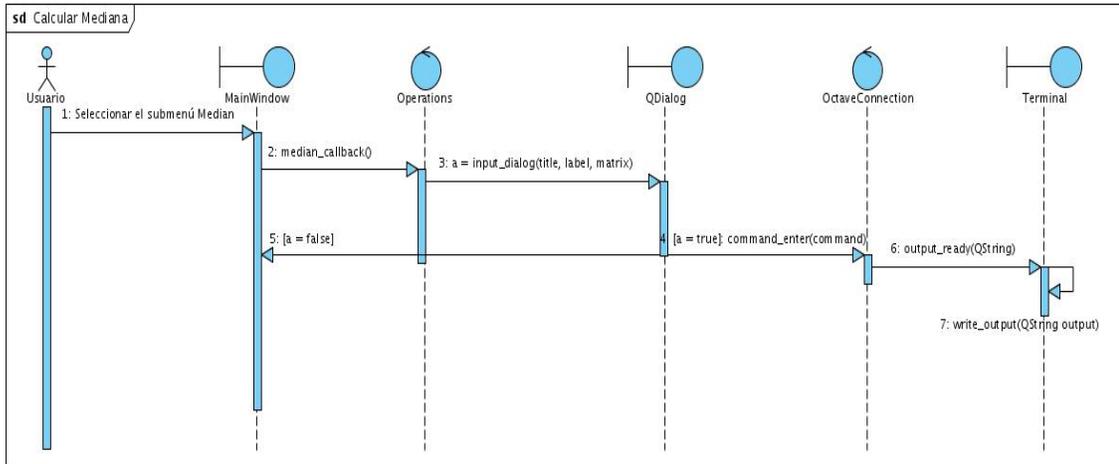


Figura 3.23: Calcular Mediana.

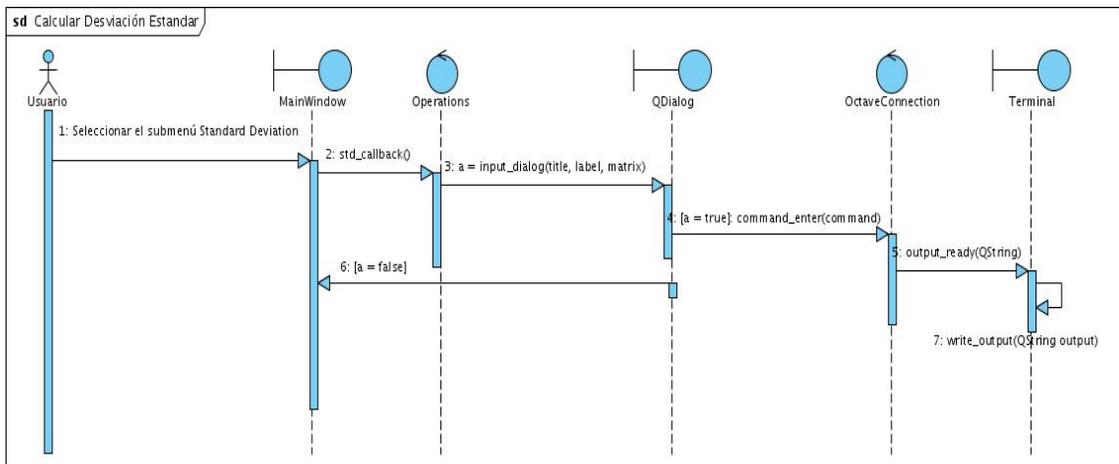


Figura 3.24: Calcular Desviación Estandar.

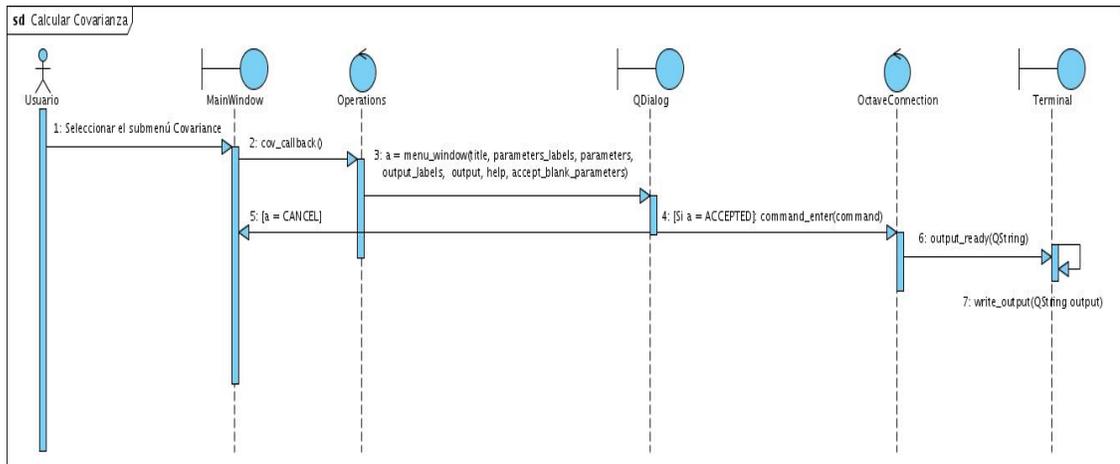


Figura 3.25: Calcular Covarianza.

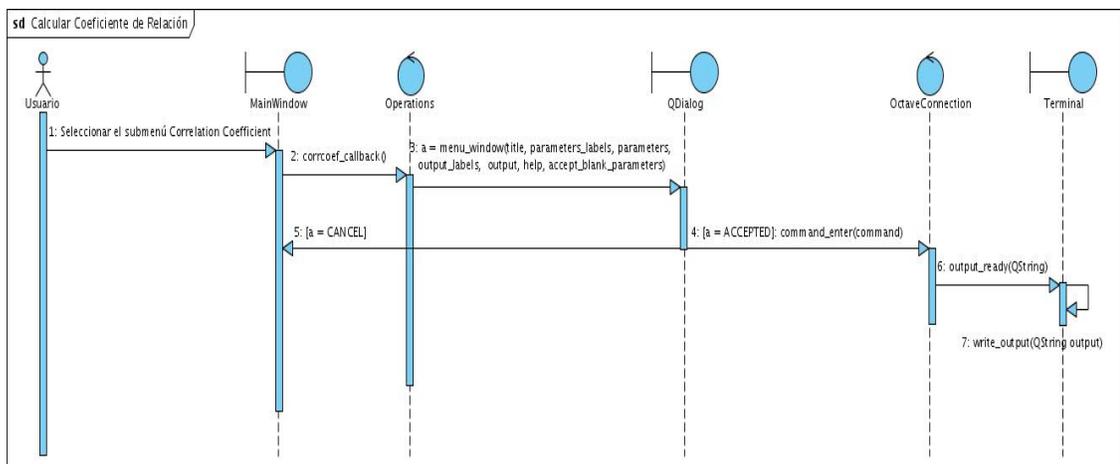


Figura 3.26: Calcular Coeficiente de Relación.

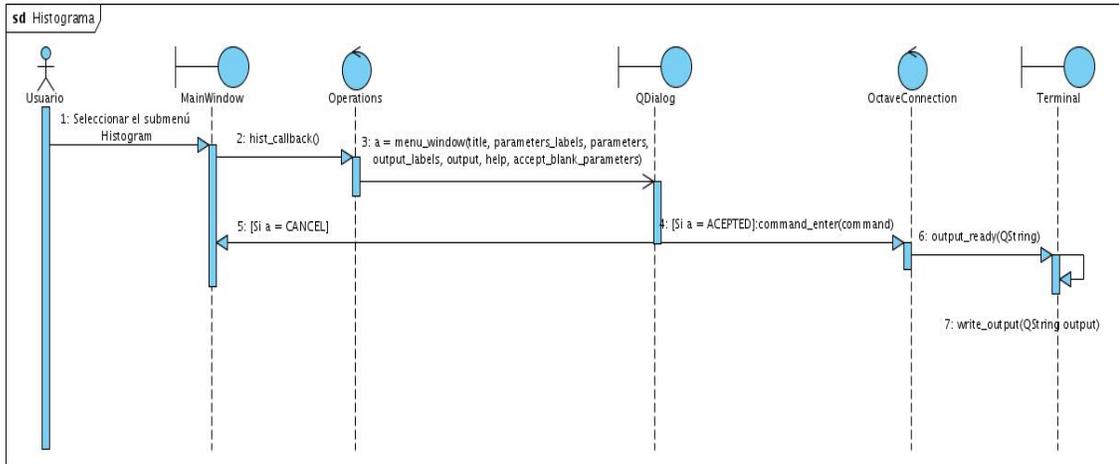


Figura 3.27: Histograma.

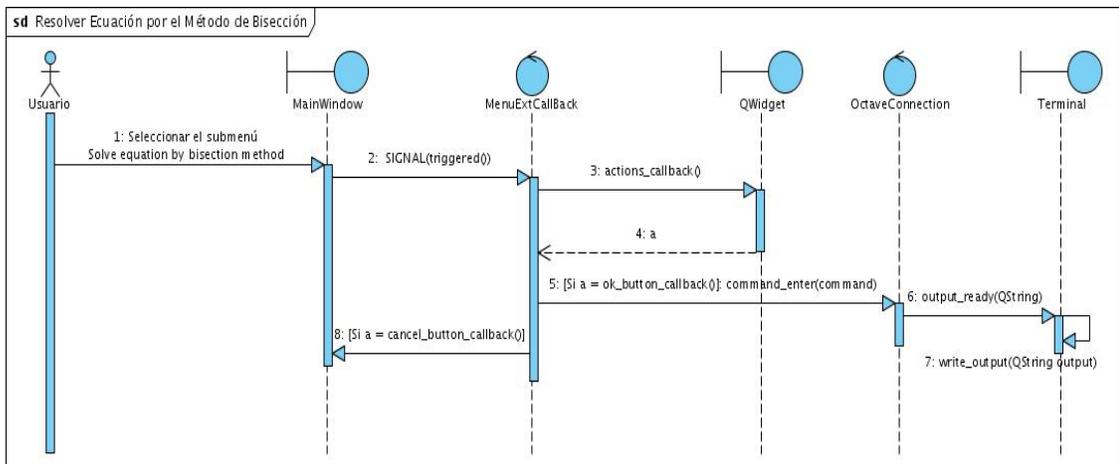


Figura 3.28: Resolver Ecuación por el Método de Bisección.

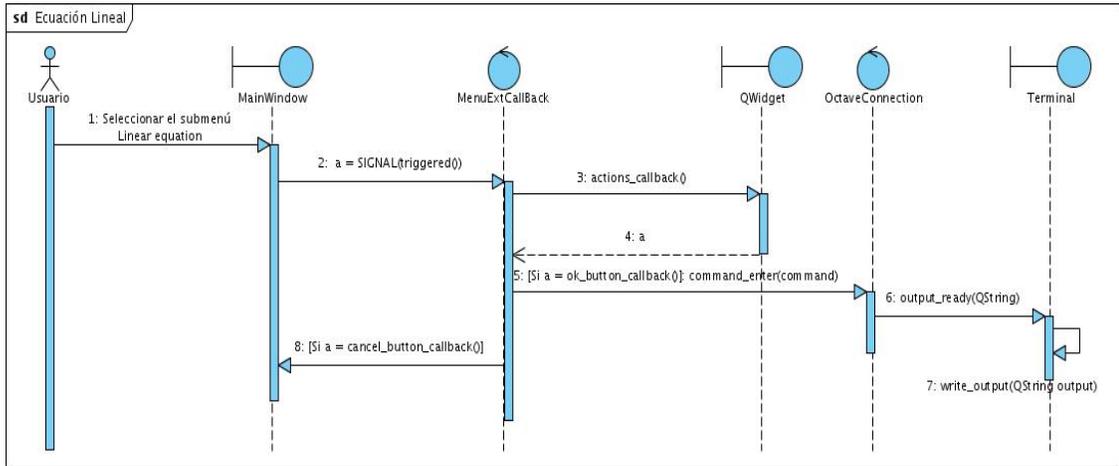


Figura 3.29: Ecuación Lineal.

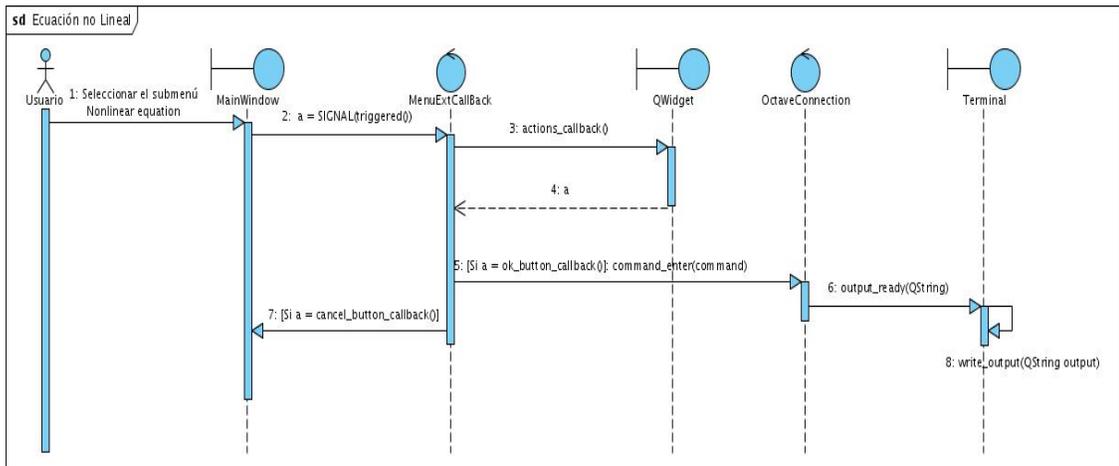


Figura 3.30: Ecuación no Lineal.

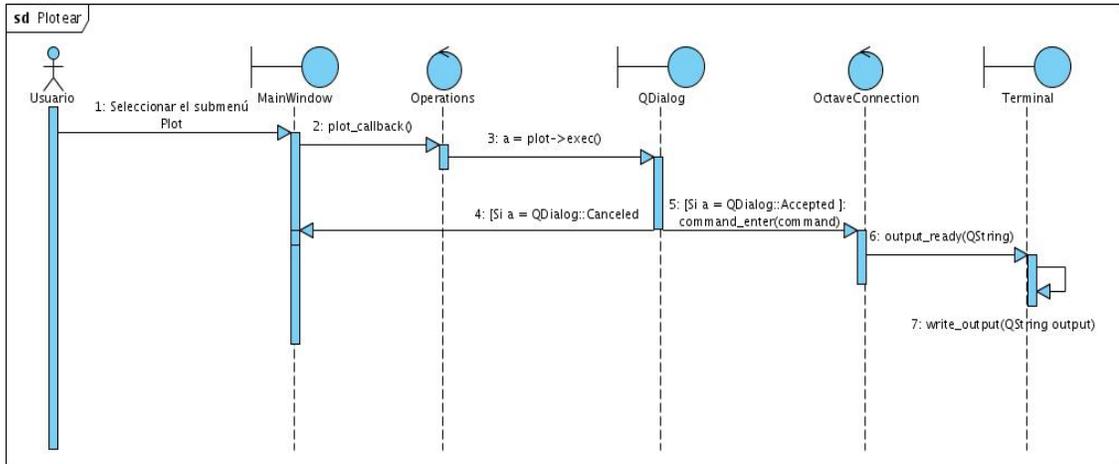


Figura 3.31: Plotear.

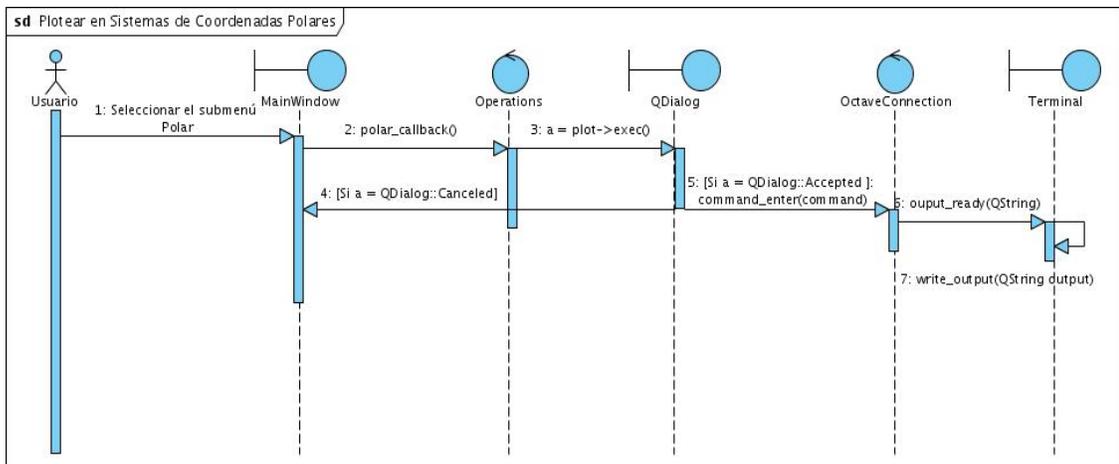


Figura 3.32: Plotear en Sistemas de Coordenadas Polares.

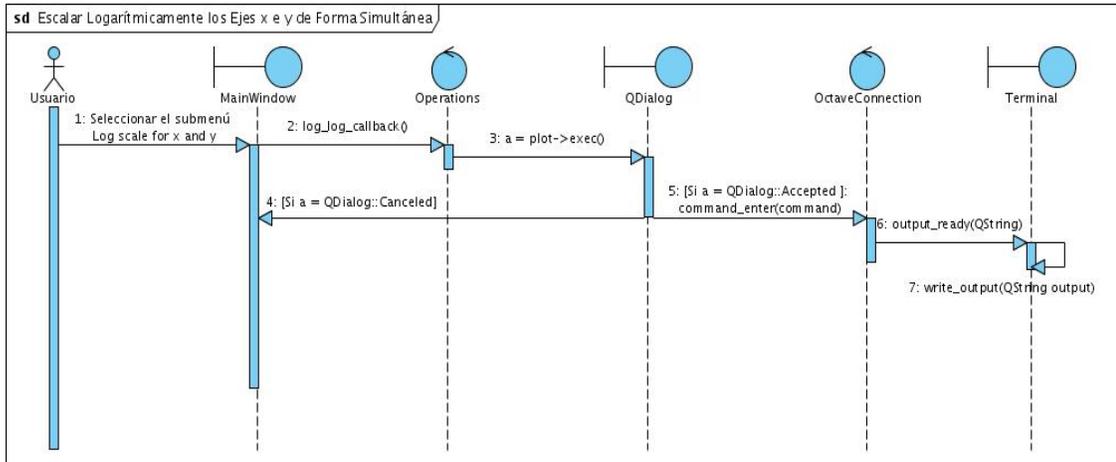


Figura 3.33: Escalar Logarítmicamente los Ejes x e y de Forma Simultánea.

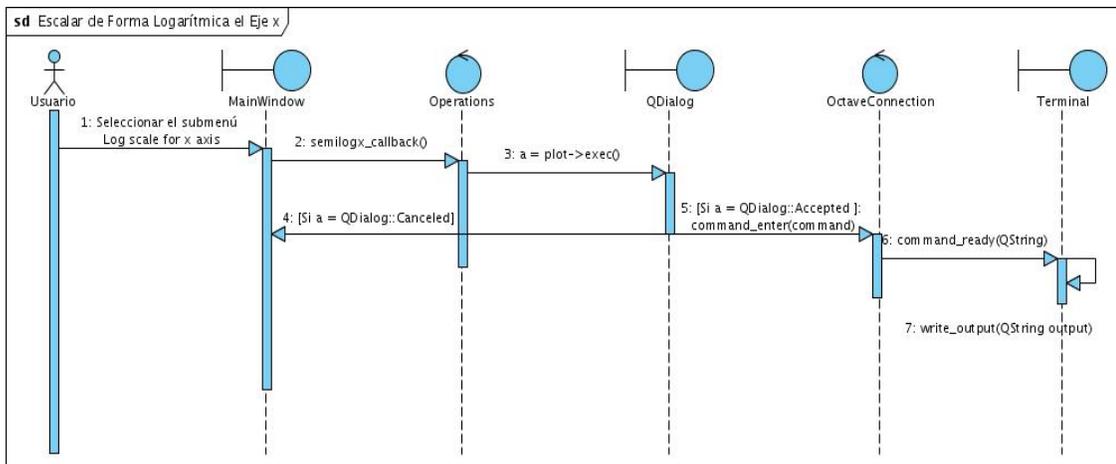


Figura 3.34: Escalar de Forma Logarítmica el Eje x.

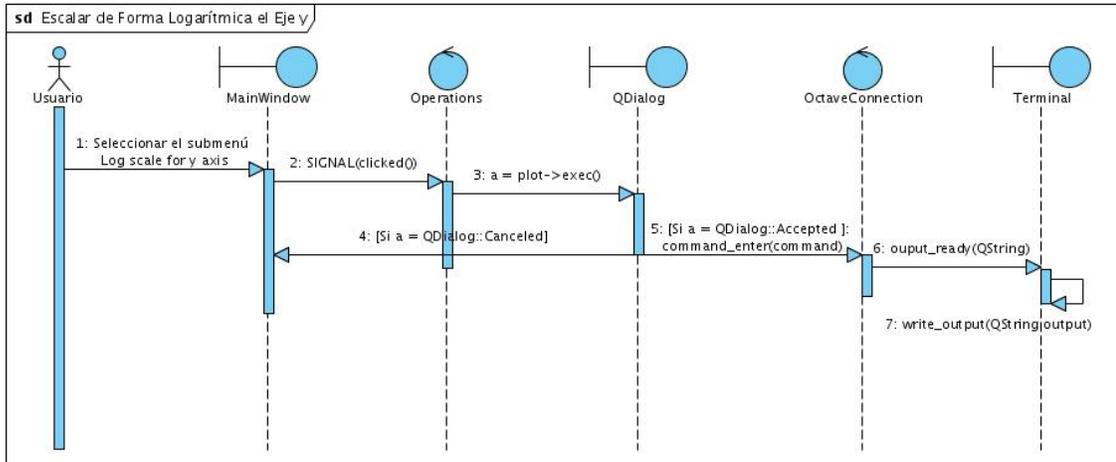


Figura 3.35: Escalar de Forma Logarítmica el Eje y.

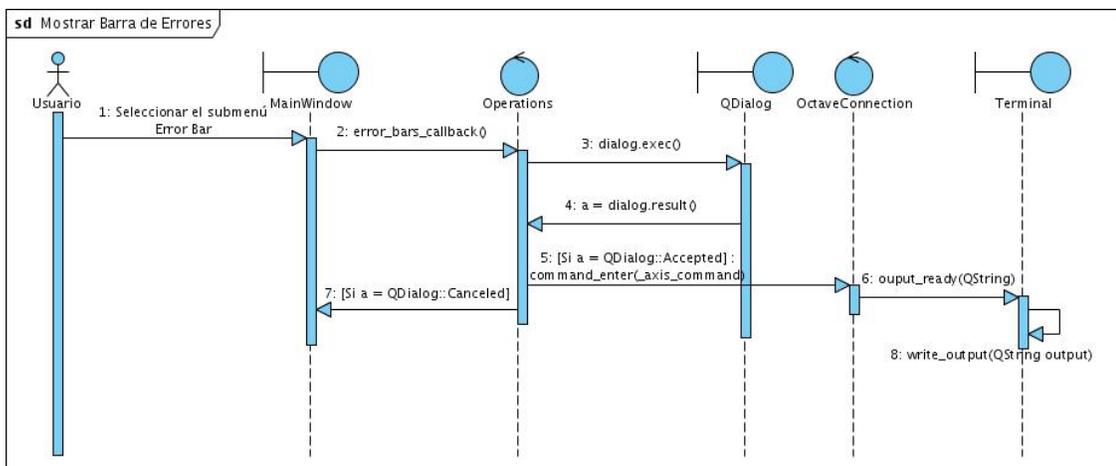


Figura 3.36: Mostrar Barra de Errores.

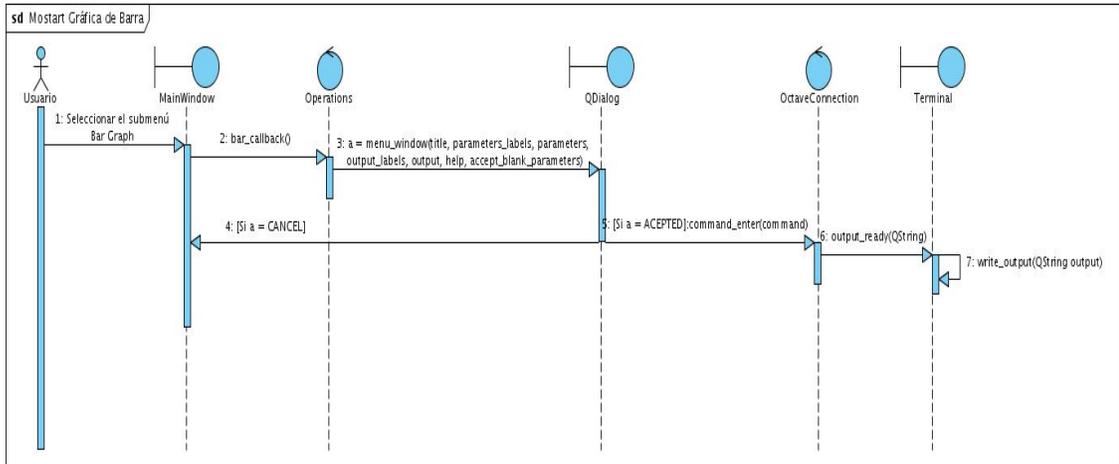


Figura 3.37: Mostrar Gráfica de Barra.

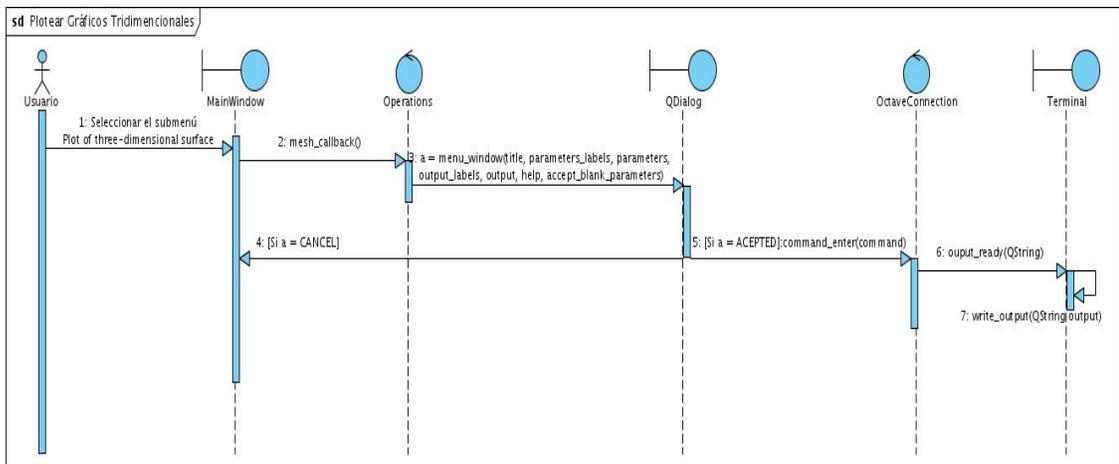


Figura 3.38: Plotear Gráficos Tridimensionales.

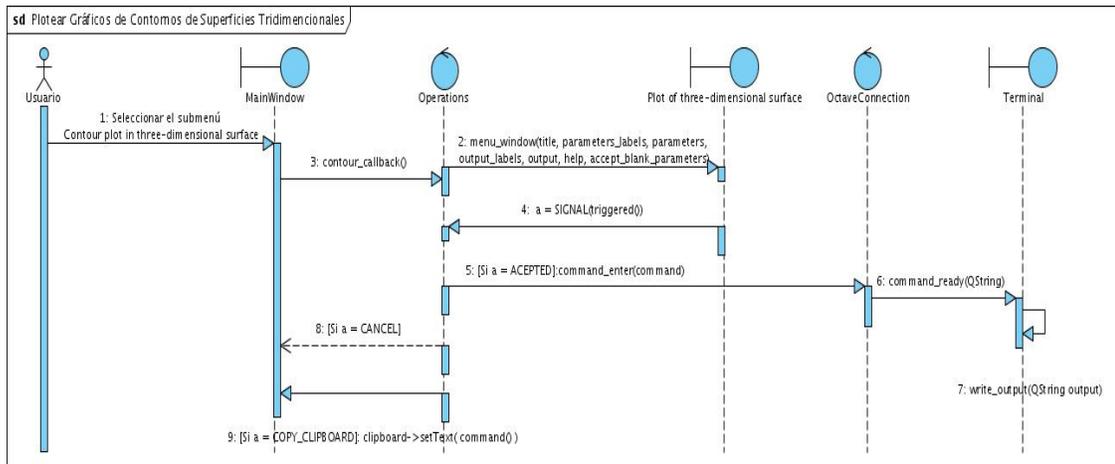


Figura 3.39: Plotear Gráficos de Contornos de Superficies Tridimensionales.

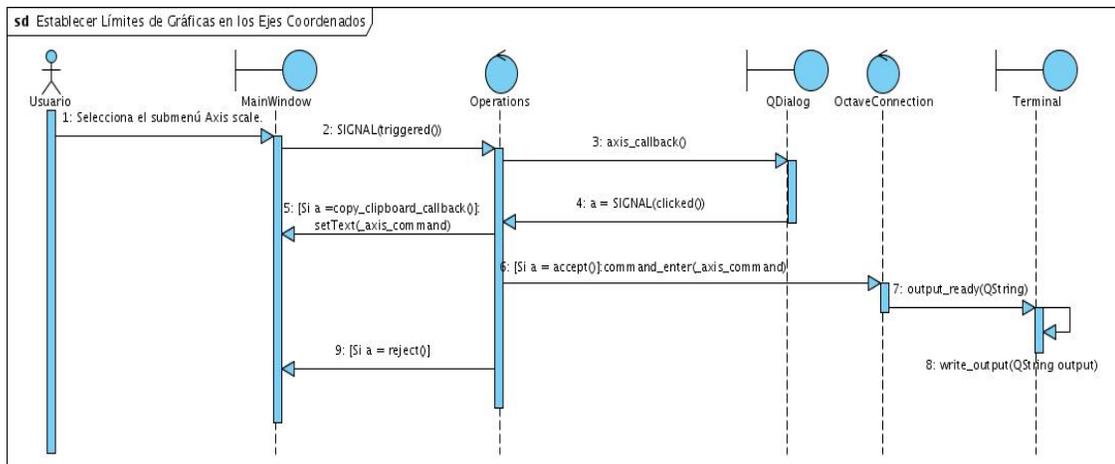


Figura 3.40: Establecer Límites de Gráficas en los Ejes Coordinados.

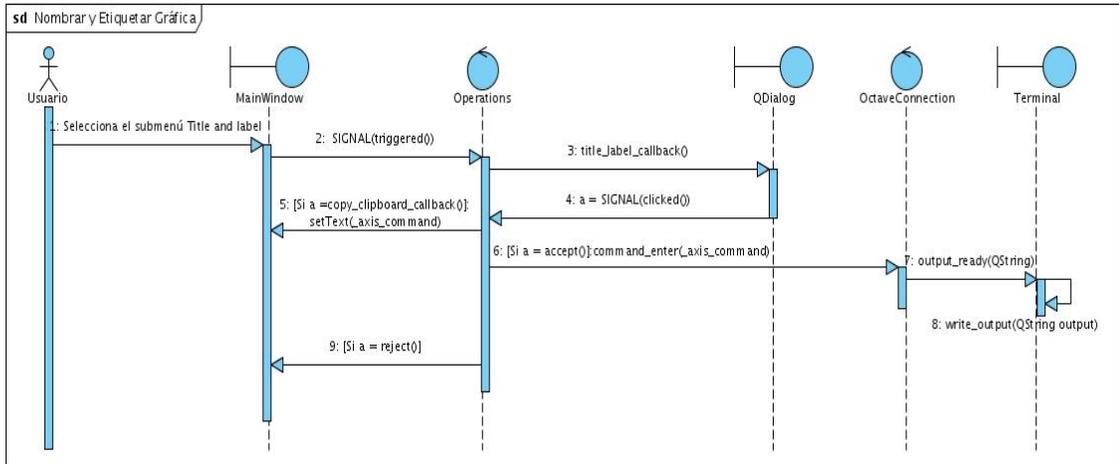


Figura 3.41: Nombrar y Etiquetar Gráfica.

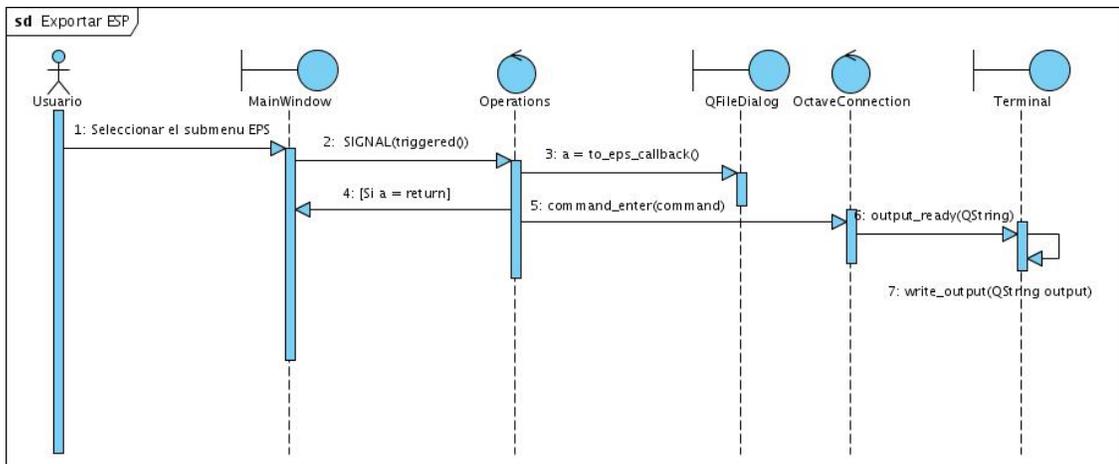


Figura 3.42: Exportar ESP.

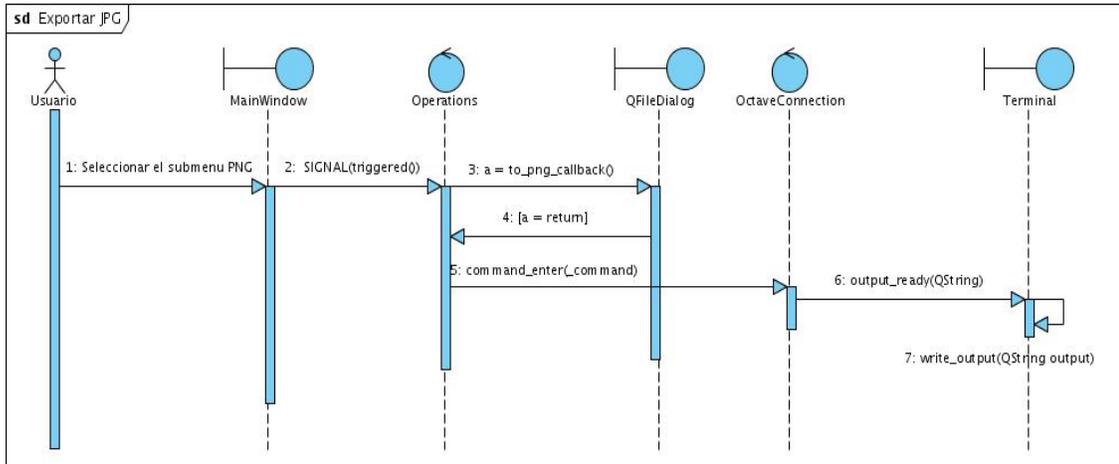


Figura 3.43: Exportar JPG.

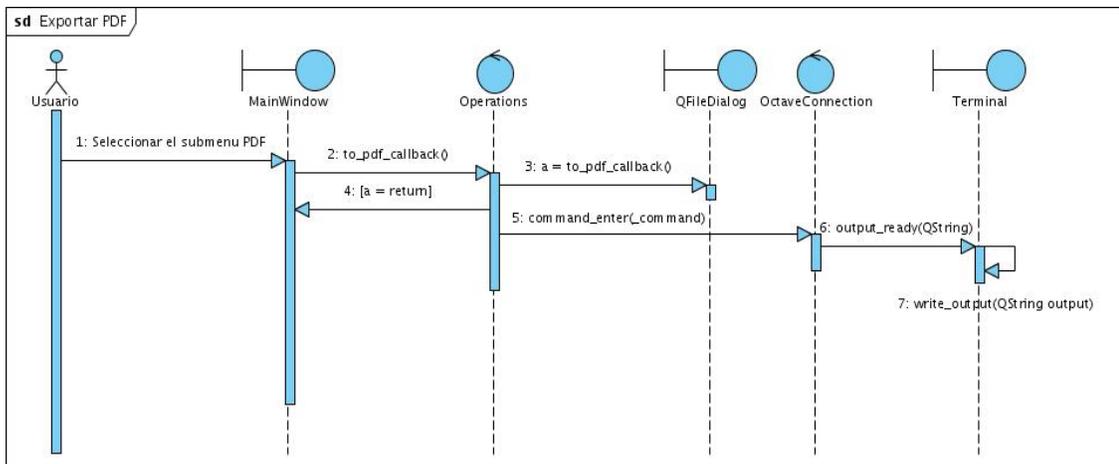


Figura 3.44: Exportar PDF.

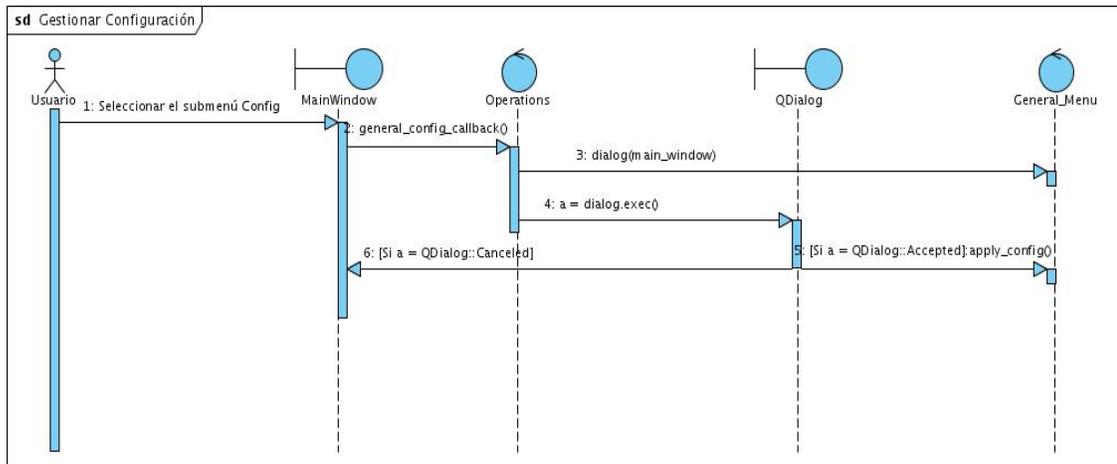


Figura 3.45: Gestionar Configuración.

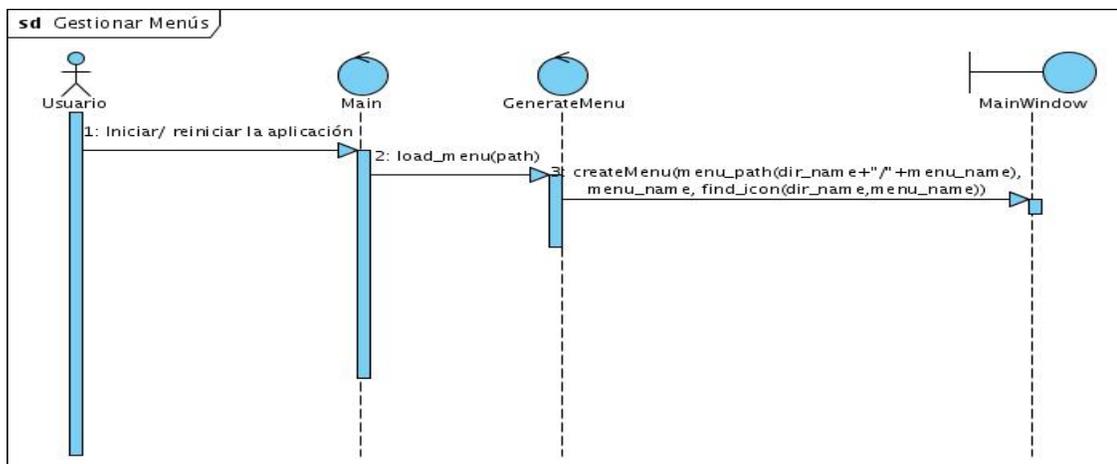


Figura 3.46: Gestionar Menús.

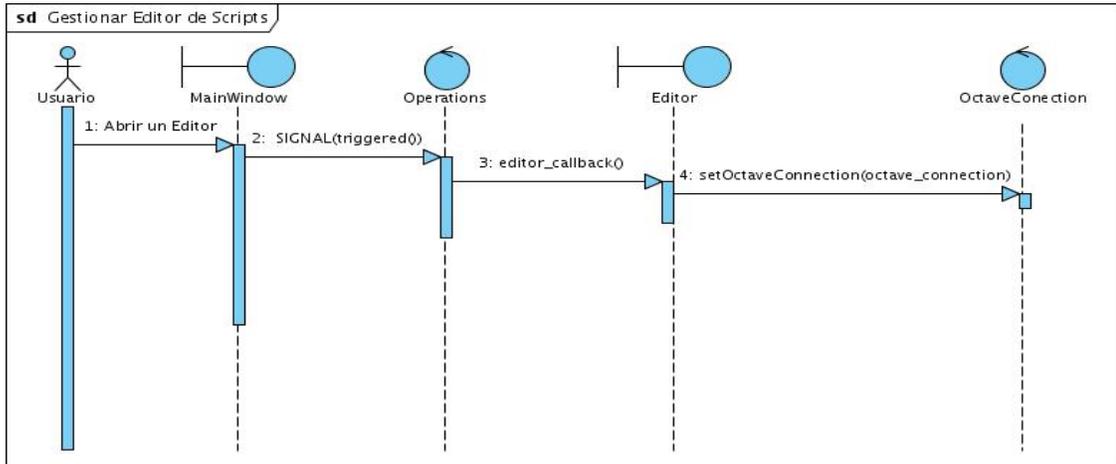


Figura 3.47: Gestionar Editor de Scripts.

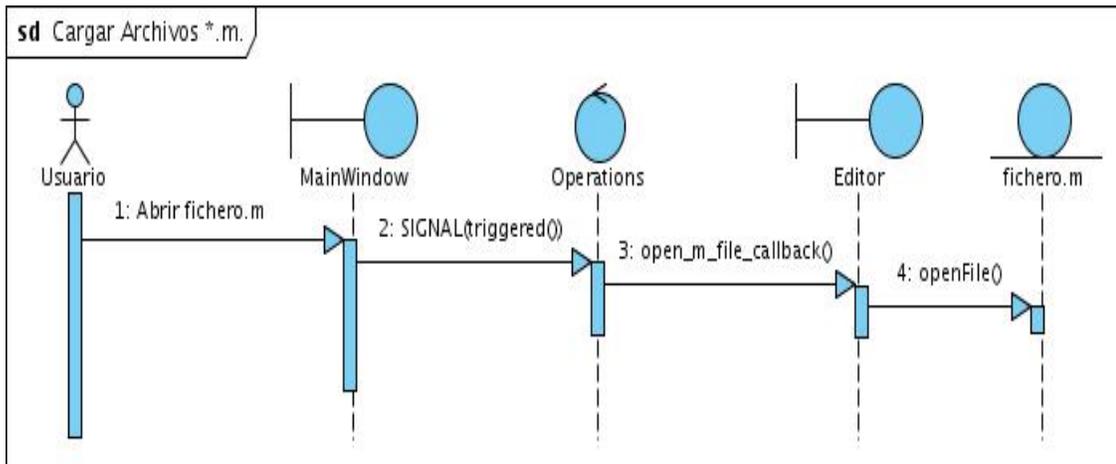


Figura 3.48: Cargar Archivos *.m.

Anexo 3: Deficiencias fundamentales encontradas durante la reingeniería

En este anexo se recogen una serie de errores que se detectaron durante el la reingeniería de QtOctave principalmente en el diseño, así como una serie de sugerencias para erradicarlos en subsiguientes implementaciones.

Deficiencia # 1: Ventana de inicio (Projects):

Esta ventana aparece siempre que se inicia QtOctave, da la posibilidad de crear un proyecto nuevo, modificar uno ya existente o eliminarlo, es la misma que se muestra al accionar el menú (Projects). Trae predefinido crear un proyecto nuevo con el nombre que defina el usuario accionando el boton (New), o un proyecto (Empty) por defecto; un proyecto vacío (Empty) y uno nuevo (New), son la misma cosa, con la diferencia que uno se llama Empty y el otro como le defina el usuario, por eso se estima que uno de los dos sobra en esa ventana.

El hecho de modificar un proyecto se resume solamente en cambiar el nombre de algunos de los ya existentes que se guardan en el archivo projects.xml en el home del usuario corriente. La opción eliminar borra un proyecto de los existentes limpiando los datos correspondientes a éste en el archivo antes mencionado. La opción adicionar funciona de forma análoga.

Seleccionando cualquiera de los casos anteriores se tiene que entrar a la aplicación, incluso presionando el botón (Cancel).

Posible solución:

Si se desea seguir la filosofía que tiene la aplicación actualmente, sería conveniente que se eliminara una de las dos opciones anteriormente mencionadas, crear un proyecto Empty sería más sensato; esto se contempla en el constructor de la clase config donde se le define el nombre al proyecto como Empty.

En las opciones del clic derecho sobre (New Project) esto está reflejado en el constructor de la clase Project. Deben dejarse solamente (Modify Project) y (Delete Project). En cuanto al botón (Cancel) debería implementarse un método que cerrara la aplicación y conectarlo con la señal correspondiente.

Alternativa:

En caso de querer darle otro enfoque al programa, sería oportuno iniciar la aplicación sin mostrar la ventana (Projects). El usuario entraría directamente a QtOctave y salvaría o no el proyecto corriente a su antojo. Para crear un proyecto nuevo se necesitaría una opción Nuevo Proyecto en el menú y dos opciones más para cargar y salvar los mismos, Cargar Proyecto y Guardar Proyecto. Bastaría entonces con llamar a los métodos ya implementados en la clase Project; y la posibilidad de guardar supliría el Renombrar, ya que el usuario lo guardaría con el nombre que desee. La lista de proyectos pasaría a un menú desplegable que se llamaría Proyectos Recientes y mostraría los nombres de los proyectos leídos del xml; si el usuario selecciona alguno de estos proyectos se llamaría al método correspondiente de la clase Project para cargar el proyecto pasándole como parámetro el nombre tomado del menú.

Deficiencia # 2: Abrir una nueva ventana de QtOctave (Open new QtOctave window):

Esto no funciona, lo que hace es cerrar la aplicación, cuando debería lanzar otro proceso QtOctave. Desde un punto de vista funcional sobre, todos los proyectos serios de este mundo obvian la posibilidad de lanzar la aplicación dentro de la propia aplicación, si el usuario quiere ejecutar otro proceso QtOctave lo ejecuta desde la barra de inicio y resuelto el problema.

Posible solución:

Quitar la opción del menú.

Deficiencia # 3:

El código fuente, independientemente de que cumple con la inmensa mayoría de los requisitos funcionales y no funcionales, carece de la organización y los comentarios suficientes para facilitar la comprensión de los procesos inherentes a la aplicación. La "suciedad" de la programación dificulta el uso de una herramienta Case para realizar ingeniería inversa a partir del código, además, la escasa presencia de comentarios es una traba significativa para cualquier persona que decida realizar un estudio de la aplicación; por otra parte, la implementación de métodos que fueron usados en versiones anteriores y otros que son totalmente innecesarios, así como la gran cantidad de líneas comentariadas que incrementan considerablemente el tamaño de las clases, haciendo más engorroso cualquier tentativa de análisis.

Posible solución:

Se debe reorganizar el código, agrupando los componentes y los métodos que los conectan con las llamadas de retorno. Sería pertinente además, comentar dichos métodos para facilitar el trabajo de los desarrolladores.

Deficiencia # 4:

Desde el punto de vista de la Programación Orientada a Objetos, el diagrama de relaciones resultante de QtOctave, no goza de la claridad requerida debido a algunas ambigüedades existentes en la jerarquía de clases. Otro aspecto relevante en el menú es la utilización de una tercera opción en los cuadros de diálogo (Copy to Clipboard), la cual no se usa en los cuadros de diálogo convencionales; y por lo tanto puede ser excluida de la aplicación ya que no satisface ningún requerimiento y de no existir, ahorraría un tiempo considerable al personal encargado de realizar una ingeniería inversa o reingeniería al sistema.

Posible solución:

Redefinir la jerarquía de clases con sus dependencias y funcionalidades, así como eliminar los métodos y/o clases innecesarios.

Deficiencia # 5:

Algunos menú de la interfaz carecen de validaciones básicas, atentando contra la estabilidad de la aplicación, tal es el caso de el menú “Config“ que no valida los parámetros de configuración y el usuario puede hacer que la aplicación colapse una vez que se reinicie, sin mencionar algunos de estos menú que no funcionan en lo absoluto.

Posible solución:

Validar la entrada de datos a la aplicación. Esto evitaría la introducción de información fuera de lugar e innecesaria.