



UNIVERSIDAD DE SANCTI SPÍRITUS “JOSÉ MARTÍ PÉREZ”
FACULTAD DE INGENIERÍA
CARRERA INGENIERÍA INFORMÁTICA

Diccionario Electrónico de Programación Lineal

Trabajo para optar por el título de Ingeniería Informática

Autor: Jorge Brunet González

Tutor: M.Sc. Aurelio Lorenzo Hernández Reyes

Sancti Spíritus

Junio de 2010

Agradecimientos

A mis compañeros de aula y de juego, por estar en las buenas, en las malas y en el estudio.

Al decano y demás profesores de la carrera en todos estos años de guía y formación.

Dedicatoria

A Alicia y Eugenio que me cuidaron cuando era chiquito y se han preocupado por mí siempre.

A Mirella, Álvaro Moisés, Álvaro José y Carlos que siempre han estado conmigo en esta espera.

A mi sobrinito y mi sobrinita.

Resumen

En la actualidad la Programación Lineal es una rama de las Matemáticas que más se usa por la alta aplicación en la vida diaria porque es capaz de ofrecer solución a diversos tipos de problemas, demostrando diariamente su gran importancia y uso obligatorio en cualquier rama ocupacional. El presente trabajo de investigación se desarrolló a partir de la necesidad existente en la Universidad de Sancti Spíritus (UNISS), el cual tuvo como objetivo implementar un Diccionario Electrónico que apoye el proceso enseñanza-aprendizaje de los diferentes términos que existen en la Programación Lineal en la carrera Contabilidad y Finanzas de dicha Facultad en el UNISS. Se logró establecer una línea de trabajo e investigación sobre el uso que se le da a las Tecnologías de la Información y las Comunicaciones (TIC) para la enseñanza-aprendizaje; y su estrecha relación con la investigación. Para la confección del diccionario se contó con la opinión y experiencias de los profesores que imparten la asignatura. La metodología utilizada en el desarrollo de la investigación fue Proceso Unificado de Rational. La implementación del sistema propuesto brindará una herramienta eficaz, capaz de brindar apoyo a los estudiantes de Contabilidad y Finanzas que lo necesiten, incluyendo la selección y muestra de palabras, de diferentes textos referentes a la asignatura, curiosidades, entre otras cosas; y también se incluye la opción de seleccionar los datos que se deseen para la confección de un informe y su posterior impresión.

Abstract:

Lineal Programming is one of the branches of Mathematics more widely used nowadays because of its high application in our lives. It has the capacity to solve different kinds of problems; its importance and obligatory use in any occupational branch has been demonstrated. This research was developed taking into account the necessity of it in The Sancti Spíritus University (UNISS); our main objective was to implement an Electronic Dictionary to improve the instructing-learning process of Lineal Programming terms used in the Faculty of Account and Finances of the UNISS. An investigation line of work about the use given to the Information and Communication Technologies (TIC) applied to the instructing-learning process and its relationship with the investigation was established. The experience and opinion of the professors who teach this subject were taken into consideration to elaborate this program. The methodology used was the Rational United Process. The use of this software will be an important tool which will help the students and professors of the Account and Finances Speciality, because it includes the selection and show of words from different texts related to this subject and curiosities, among other aspects; it also offers the option to choice a desired data to elaborate an information to be printed later on.

Índice

Introducción.....	9
Capítulo 1 Fundamentación Teórica.....	6
1.1 Introducción.....	6
1.2 Principales conceptos asociados al dominio del problema.....	6
1.2.1 El proceso enseñanza-aprendizaje.....	6
1.2.2 Las Tecnologías de la Información y las Comunicaciones (TIC).....	8
1.2.3 Las TIC en la educación.....	8
1.2.4 Las TIC en Cuba.....	10
1.3 Programación Lineal.....	11
1.3.1 Historia de la Programación Lineal.....	12
1.4 Visión y Misión de la Universidad de Sancti-Spíritus.....	13
1.5 El proceso de enseñanza de la Programación Lineal en la UNISS.....	14
1.6 Descripción de sistemas existentes.....	14
1.7 Descripción de la situación problemática.....	15
1.8 Metodologías y tecnologías actuales.....	16
1.8.1 Metodologías actuales a considerar.....	16
Rational Unified Process (RUP).....	19
1.8.2 Tecnologías actuales a considerar.....	20
1.8.3 Lenguajes de Programación y Herramientas de Desarrollo.....	24
Capítulo 2 Modelo de Dominio.....	28
2.1 Introducción.....	28
2.2 Descripción del modelo de dominio.....	28
2.2.1 Principales conceptos asociados al dominio del problema.....	28
2.2.2 Modelo de objetos del dominio.....	29
2.3 Requerimientos.....	30
2.3.1 Requerimientos funcionales.....	30
2.3.2 Requerimientos no funcionales.....	31
2.4 Descripción del Sistema Propuesto.....	33
2.5 Modelo de casos de uso del sistema.....	34
2.5.1 Actores del sistema.....	34
2.5.2 Diagramas de casos de uso del sistema.....	35
2.4.3 Descripción de los casos de uso del sistema.....	36
2.5 Conclusiones.....	47
Capítulo 3 Construcción de la solución propuesta.....	48
3.1 Introducción.....	48
3.2 Diagrama de clases del diseño.....	48
3.3 – Diseño de la base de datos.....	60
3.3.1 Modelo lógico de datos.....	60
3.3.2 Modelo físico de datos.....	61
3.4 Diagrama de implementación.....	62
3.5 Modelo de despliegue.....	62
3.6 Principios de diseño.....	63

3.6.1 Estándares en la interfaz de la aplicación	63
3.6.2 Tratamiento de errores	64
3.7 Estándares de codificación.....	65
3.8 Concepción de la ayuda.....	66
3.9 Conclusiones	66
CAPÍTULO 4. Estudio de Factibilidad.....	67
4.1 Introducción.....	67
4.2 Planificación.....	67
4.3 Costos.....	70
4.4 Beneficios tangibles e intangibles.....	72
4.5 Análisis de costos y beneficios.....	73
4.6 Conclusiones.....	73
Conclusiones.....	74
Recomendaciones	75
Bibliografía.....	76
Anexos.....	78

Índice de Figuras

Figura 2.1 Modelo de objetos del dominio	29
Figura 2.2 Diagramas de casos de uso del sistema	36
Figura 3.1 Diagrama de clases del diseño CU Gestionar Palabra	49
Figura 3.2 Diagrama de clases del diseño CU Gestionar Usuario.....	50
Figura 3.3 Diagrama de clases del diseño CU Gestionar Artículo	51
Figura 3.4 Diagrama de clases del diseño CU Buscar Palabra.....	52
Figura 3.5 Diagrama de clases del diseño CU Buscar Artículo	52
Figura 3.6 Diagrama de clases del diseño CU Mostrar Palabra	53
Figura 3.7 Diagrama de clases del diseño CU Mostrar Artículo	53
Figura 3.8 Diagrama de clases del diseño CU Iniciar Sesión	54
Figura 3.9 Diagrama de clases del diseño CU Cerrar Sesión.....	54
Figura 3.10 Diagrama de clases del diseño CU Cambiar Contraseña	55
Figura 3.11 Diagrama de clases del diseño CU Preparar Reporte de Palabras	56
Figura 3.12 Diagrama de clases del diseño CU Preparar Reporte de Artículos	57
Figura 3.13 Diagrama de clases del diseño CU Generar Reporte de Palabras	58
Figura 3.14 Diagrama de clases del diseño CU Generar Reporte de Artículos	59
Figura 4.2 Modelo lógico de datos	60
Figura 4.3 Modelo físico de datos	61
Figura 4.5 Diagrama de implementación	62
Figura. 4.6 Diagrama de despliegue.	63
Anexos A: Interfaces relacionadas con los Requerimientos del Sistema	78

Índice de Tablas

Tabla 2.1. Descripción de los actores del sistema	34
Tabla 2.2. Descripción del caso de uso de sistema <Gestionar Palabras>.....	37
Tabla 2.3. Descripción del caso de uso de sistema <Gestionar Usuario>	38
Tabla 2.4. Descripción del caso de uso de sistema <Gestionar Artículo>	39
Tabla 2.5. Descripción del caso de uso de sistema <Buscar Palabra>	39
Tabla 2.6. Descripción del caso de uso de sistema <Buscar Artículo>	40
Tabla 2.7. Descripción del caso de uso de sistema <Mostrar Palabras>	41
Tabla 2.8. Descripción del caso de uso de sistema <Mostrar Artículos>	42
Tabla 2.9. Descripción del caso de uso de sistema < Iniciar Sesión >	42
Tabla 2.10. Descripción del caso de uso de sistema < Cerrar Sesión >.....	43
Tabla 2.11. Descripción del caso de uso de sistema < Cambiar Contraseña >	44
Tabla 2.12. Descripción del caso de uso de sistema <Preparar Reporte de Palabras>	44
Tabla 2.13. Descripción del caso de uso de sistema <Preparar Reporte de Artículos>.....	45
Tabla 2.14. Descripción del caso de uso de sistema <Generar Reporte de Palabras>.....	46
Tabla 2.15. Descripción del caso de uso de sistema <Generar Reporte de Artículos>	47
Tabla 4.1 Nombres de algunos controles utilizados.	65
Anexos B: Elementos relacionados a los Diagramas de Clases del Diseño.....	92

Introducción

Con el objetivo de llevar la ciencia, la tecnología, la educación y la cultura a los más apartados rincones del país se lleva a cabo la informatización de la sociedad cubana. Un aspecto importante en el desarrollo de esta política ha sido la formación en toda la población de una cultura en el uso de las Tecnologías de la información y las Comunicaciones (TIC). Todo esto se ha ido desarrollando con un programa de informatización que ha ido llevando nuestro país donde poco a poco se han puesto a disposición de todos los ciudadanos modernos equipamientos con el objetivo de su capacitación y superación en las diferentes ramas de la informática.

Actualmente las universidades del país son parte de dicho programa, junto con la creación de joven club de computación, politécnicos de informática, sedes universitarias, etc, constituyendo un gran potencial para mejorar el acceso y la calidad de la educación, brindándose posibilidades de vencer las distancias geográficas, de fortalecer el aprendizaje autónomo, entre otras cosas. Haciéndose necesario aprender a usar las nuevas tecnologías y utilizarlas para aprender.

La Programación Lineal se ha convertido en una herramienta inseparable para el desarrollo y rentabilidad de muchas tareas, que por muy diferentes que sean, requieran un elemento en común muy necesario en estos tiempos: la economía.

Por su poder y alcance poco a poco se ha ido ganando el lugar cimero que se merece, ya que por sus características y el uso correcto de sus definiciones, reduce drásticamente el número de posibles soluciones óptimas que deberán ser revisadas lográndose un método eficiente para llegar de manera más rápida a determinar una decisión óptima, (o una estrategia óptima o un plan óptimo) de las posibles encontradas, trazándose un objetivo: maximizar o minimizar, logrando una ganancia o ahorro al máximo según el contexto del problema analizado.

Por su complejidad en los cálculos se necesita el análisis utilizando programas en la computadora para llegar a los resultados que se necesitan, pero lo anterior no le quita protagonismo al factor humano, ya que es a nosotros como personas que nos toca una parte importante del trabajo: el análisis profundo del problema para poder adaptarlo de forma correcta a la Programación Lineal y separarlo en los elementos necesarios para poder llegar a la solución esperada, la computadora, como dije anteriormente, solamente se limita a los cálculos.

Desde el punto de vista social la Programación Lineal se imparte en las carreras de, Contabilidad y Finanzas, Economía, en el grupo de las Ingenierías (Eléctrica, Industrial,

Hidráulica, del Transporte, Mecánica e Informática entre otras) Ciencias de la Computación y Cibernética, Matemática Aplicada y otras tantas, muchas de ellas se estudian en casi todas las universidades de nuestro país.

Los docentes aprovechando las facilidades de las TIC buscan de manera amena dotar a los alumnos de herramientas y recursos de aprendizaje que faciliten su autonomía personal, y que los preparen como profesionales competentes y flexibles, capaces de enfrentarse a los constantes cambios científicos y tecnológicos que está viviendo la humanidad. Hoy es frecuente que tanto docentes como asesores asuman la necesidad de enseñar a los alumnos a documentarse, a analizar los datos recogidos, a sintetizar los aspectos relevantes, a comunicar las propias conclusiones de forma clara y precisa, en definitiva que consideren que la máxima " aprender a aprender " es un objetivo educativo de primer orden.

En la actualidad, la Programación Lineal se imparte a los alumnos de la Facultad de Contabilidad y Finanzas de la Universidad "José Martí Pérez" de Sancti Spiritus (UNISS), la misma cuenta con pocos trabajos de diploma que hayan sido dedicados a los temas relacionados con dicha asignatura y partiendo de la necesidad de comenzar a crear investigaciones que apoyen a la facultad y a este tema, por su gran importancia, en su proceso de educación y profundización sobre los diferentes puntos que se imparten; ya que los mismos resultan ser muy diversos, frente a la escasa bibliografía existente y al difícil acceso a Internet, llegando a veces a generar dudas entre los estudiantes sobre alguna palabra o término en específico, que necesita de una comprensión adecuada para su uso posterior, lo explicado anteriormente es una breve presentación de **la situación problemática**.

Todo esto nos lleva al **Problema Científico**, el cual se plantea con la siguiente pregunta:

¿Cómo apoyar el desarrollo del proceso de enseñanza-aprendizaje de la Programación Lineal en los estudiantes de Contabilidad y Finanzas en la Universidad de Sancti Spiritus?

El **Objeto de estudio** de este trabajo es el proceso de enseñanza-aprendizaje. El **Campo de acción** es el proceso de enseñanza-aprendizaje de la Programación Lineal en la carrera Contabilidad y Finanzas.

Para dar solución a la problemática existente se decide como **objetivo general** realizar el análisis, diseño e implementación de un Diccionario Electrónico que contribuya al desarrollo de los conocimientos de la Programación Lineal por parte de los estudiantes de la carrera de Contabilidad y Finanzas en la UNISS.

En el mismo se puede insertar, actualizar, eliminar o revisar un término de manera sencilla y se puede crear e imprimir un informe con los datos que se necesiten.

Del objetivo se desprendieron las siguientes **Preguntas científicas**:

- ¿Cuáles son los fundamentos teóricos–metodológicos que influyen en el proceso de enseñanza-aprendizaje de la Programación Lineal?
- ¿Cómo diseñar un Diccionario Electrónico basado en una metodología como apoyo al proceso de enseñanza-aprendizaje de la Programación Lineal en los estudiantes de Contabilidad y Finanzas de la UNISS?
- ¿Cómo implementar un Diccionario Electrónico cumpliendo con las características definidas en el proceso de análisis y diseño?
- ¿Cómo realizar el análisis de costos y beneficios del Diccionario Electrónico?

Del cual se desprenden los siguientes **objetivos específicos**:

- Determinar con la revisión bibliográfica los fundamentos teóricos – metodológicos que influyen en el proceso de enseñanza-aprendizaje de la Programación Lineal.
- Diseñar un Diccionario Electrónico basado en una metodología para apoyar el proceso enseñanza-aprendizaje de los términos de Programación Lineal en los estudiantes de Contabilidad y Finanzas de la UNISS.
- Implementar el sistema con las características definidas en el proceso de análisis y diseño para apoyar el proceso enseñanza-aprendizaje de la Programación Lineal en los estudiantes de Contabilidad y Finanza en la UNISS.
- Realizar el análisis costos y beneficios para observar la factibilidad del producto final.

La **novedad científica** de esta investigación radica en la implementación de una herramienta que reunirá la mayor cantidad de datos posible recogida de diferentes bibliografías y que facilitará la búsqueda de información sobre temas de Programación Lineal como apoyo al proceso de enseñanza–aprendizaje de la Programación Lineal, partiendo, por supuesto desde el terreno metodológico y un análisis bibliográfico minucioso.

Métodos del nivel teórico: Tienen como objetivo la interpretación de la información empírica obtenida y el establecimiento de generalizaciones, regularidades, teorías y nuevas concepciones, los utilizados fueron los siguientes:

- **Dialéctico-Materialista:** La presente investigación tiene un enfoque dialéctico-materialista apoyado por los métodos teóricos que se relacionan con los procesos lógicos del pensamiento entre ellos:
- **Histórico-Lógico:** Para abordar el estudio de la temática en su secuencia histórica.
- **Inducción-Deducción:** Para transitar desde lo particular a lo general.
- **Vinculación de lo abstracto a lo concreto:** Se parte del conocimiento de la situación problemática a través del análisis y síntesis que permite realizar abstracciones sobre el problema identificado, lo que conllevará a extraer conclusiones de la esencia del mismo, para posteriormente reparar a lo concreto pensado y darle solución a dicho problema.
- **Analítico-Síntesis:** Permitted en un proceso de percepción detallada de cada aspecto del problema planteado. Luego se integraron todos estos elementos entre sí y permitió delimitar el problema para modelar una estrategia con el fin de cambiar la situación existente.
- **Modelación:** Permitted llevar ideas y concepciones a la creación de una representación del modelo.
- **Enfoque del Sistema:** Permitted ver la construcción del software como una secuencia concatenada de pasos.

El presente trabajo está estructurado en 4 capítulos de la siguiente forma:

Capítulo 1: Fundamentación Teórica. Descripción de los principales conceptos que son necesarios para comprender el objeto de estudio. Análisis del objeto de estudio y los sistemas existentes vinculados a él y de las metodologías y tecnologías actuales a considerar para el desarrollo del sistema propuesto.

Capítulo 2: Modelo del Dominio. Descripción de los procesos, actores, trabajadores y casos de uso vinculados al dominio; y diagramas de clases del modelo de objetos del dominio.

Capítulo 3: Construcción de la solución propuesta. Descripción del diseño a través del diagrama de clases del diseño, el de clases persistentes y el modelo físico de datos. Se definen además, el diagrama de implementación y los principios de diseño seguidos en la aplicación.

Capítulo 4: Estudio de factibilidad. Estudio de factibilidad económica realizado para este proyecto, en el que se determina si es factible o no el desarrollo del software propuesto.

Capítulo 1 Fundamentación Teórica.

1.1 Introducción.

En este capítulo se puntualizan los conceptos necesarios para comprender el dominio del problema y el objeto de estudio. Además se analiza el flujo de los principales procesos que se desarrollan en el campo de acción y las causas que originan la situación problemática.

También se realiza un análisis de las tendencias, metodologías y tecnologías actuales a considerar, sobre las cuales se fundamenta la propuesta de automatización.

1.2 Principales conceptos asociados al dominio del problema.

1.2.1 El proceso enseñanza-aprendizaje.

Enseñanza:

“Es el proceso mediante el cual se comunican o transmiten conocimientos especiales o generales sobre una materia. Este concepto es más restringido que el de educación, ya que ésta tiene por objeto la formación integral de la persona humana, mientras que la enseñanza se limita a transmitir, por medios diversos, determinados conocimientos. En este sentido la educación comprende la enseñanza propiamente dicha.” (Edel Navarro, 2004)

“El propósito esencial de la enseñanza es la transmisión de información mediante la comunicación directa o soportada en medios auxiliares, que presentan un mayor o menor grado de complejidad y costo. Como resultado de su acción, debe quedar una huella en el individuo, un reflejo de la realidad objetiva, del mundo circundante que, en forma de conocimiento, habilidades y capacidades, le permitan enfrentarse a situaciones nuevas con una actitud creadora, adaptativa y de apropiación. El proceso de enseñanza produce un conjunto de transformaciones sistemáticas en los individuos, una serie de cambios graduales cuyas etapas se suceden en orden ascendente. Es, por tanto, un proceso progresivo, dinámico y transformador.” (Alfonso Sánchez, 2003)

Aprendizaje:

Refiriéndose al aprendizaje (Edel Navarro, 2004) comentó: “Este concepto es parte de la estructura de la educación, por tanto, la educación comprende el sistema de aprendizaje. Es la acción de instruirse y el tiempo que dicha acción demora. También, es el proceso por el cual

una persona es entrenada para dar una solución a situaciones; tal mecanismo va desde la adquisición de datos hasta la forma más compleja de recopilar y organizar la información.”

“El aprendizaje es un proceso de naturaleza extremadamente compleja, cuya esencia es la adquisición de un nuevo conocimiento, habilidad o capacidad. Para que dicho proceso pueda considerarse realmente como aprendizaje, en lugar de una simple huella o retención pasajera, debe poder manifestarse en un tiempo futuro y contribuir, además, a la solución de problemas concretos, incluso diferentes en su esencia a los que motivaron inicialmente el desarrollo del conocimiento, habilidad o capacidad. El aprendizaje, si bien es un proceso, también resulta un producto por cuanto son, precisamente, los productos los que atestiguan, de manera concreta, los procesos. Aprender, para algunos, no es más que concretar un proceso activo de construcción que realiza en su interior el sujeto que aprende (teorías constructivistas).” (Alfonso Sánchez, 2003)

En otra definición (Alvarez de Zayas, 1999) se menciona que: “El aprendizaje es la actividad que desarrolla el estudiante para aprender, para asimilar la materia de estudio (...) el estudiante es el sujeto de su propio aprendizaje”.

(Labarrere, 1988) confirma el papel del estudiante al determinar que: “(...) Un tercer componente lo encontramos en el sistema de aprendizaje, es decir, en el alumno. El escolar resulta al mismo tiempo objeto y sujeto de la educación. Lo primero está dado por constituir el elemento sobre el cual recae la acción educadora del maestro. Lo segundo se explica al considerar que el alumno es a la vez sujeto activo del aprendizaje”

Según (Bravo López & Cáceres Mesa) “Los objetivos del proceso de enseñanza-aprendizaje constituyen los fines o resultados, previamente concebidos como un proyecto abierto y flexible, que guían las actividades de profesores y estudiantes para alcanzar las transformaciones necesarias en estos últimos. Como expresión del encargo social que se plantea a la escuela reflejan el carácter social del proceso de enseñanza, sirviendo así de vínculo entre la sociedad y la escuela. Como se deduce de la definición anterior; los objetivos constituyen el componente que mejor refleja el carácter social de proceso de enseñanza-aprendizaje e instituyen la imagen del hombre que se intenta formar, en correspondencia con las exigencias sociales que compete cumplir a la escuela. “

“Las reformas educativas, iniciadas en la última década, han insistido en considerar a los alumnos como sujetos activos en la construcción de conocimientos, en la necesidad de

promover aprendizajes en sentido amplio y en asignar un nuevo rol al docente como mediador y facilitador del aprendizaje.” (UNESCO, 2003)

1.2.2 Las Tecnologías de la Información y las Comunicaciones (TIC).

Según (*Las TICs, 2004*), las TIC se pueden definir como:

- Son tecnologías que constituyen nuevos canales de comunicación y entran en las escuelas y los hogares facilitando con su uso el proceso de enseñanza-aprendizaje.
- La denominación de TIC es utilizada para referirse a una serie de nuevos medios como los hipertextos, los multimedia, Internet, la realidad virtual o la televisión por satélite.
- Estas tecnologías se diferencian de las tradicionales en las posibilidades que tienen la creación de nuevos entornos comunicativos y expresivos que facilitan a los receptores la posibilidad de desarrollar nuevas experiencias formativas, expresivas y educativas.

En definitiva, las TIC permiten el procesamiento, tratamiento y comunicación de la información. Es decir, las tecnologías son para actuar sobre la información, transformarla, difundirla y comunicarla.

Facilitan la reproducción, difusión y circulación de documentos, permitiendo así la creación de un gran volumen de información paralela a la industria editorial tradicional y a los servicios de biblioteca.

Permiten una formación individualizada, en donde cada alumno puede trabajar a su ritmo, sin presión de trabajar al mismo tiempo que otros. (López, 2001 y García, 2000)

1.2.3 Las TIC en la educación.

La introducción de las TIC en la educación ocurre a partir de que son tecnologías que giran en torno a cuatro medios: la informática, la microelectrónica, los multimedia y las telecomunicaciones permitiendo la planificación del aprendizaje, en donde cada estudiante define su parámetro de estudio, de acuerdo a su tiempo disponible y a sus posibilidades y necesidades.

La educación no puede permanecer al margen de estos nuevos cambios que ocurren cada día. Debe atender a la formación de los nuevos ciudadanos y la incorporación de las nuevas

tecnologías ha de hacerse con la perspectiva de favorecer los aprendizajes y facilitar los medios que sustenten el desarrollo de los conocimientos y de las competencias necesarias para la inserción social y profesional de calidad. Debe también evitar que la brecha digital genere capas de marginación como resultado de la analfabetización digital. (Ramón Gómez,2004)

El impacto social de las TIC está reflejado en escuelas y universidades, propiciando modificaciones en las formas tradicionales de enseñar y aprender. Es evidente que el avance y perfeccionamiento de la tecnología no parece tener freno, el reto de los centros educativos y en particular de las universidades radica en prepararse como institución y preparar a sus profesores para adaptarse a los cambios de manera rápida y efectiva. Entre las claves fundamentales para el éxito está lograr que el aprendizaje se convierta en un proceso natural y permanente para estudiantes y docentes.

(Martínez Riverol,2006) comentó que "Es tarea de los educadores utilizar las TIC como medios para proporcionar la formación general y la preparación de sus estudiantes, contribuyendo a la formación más desarrollada e integral del individuo."

Se ha observado que las tecnologías de la información suscitan la colaboración en los alumnos, les ayuda a centrarse en los aprendizajes, mejoran la motivación y el interés, favorecen el espíritu de búsqueda, promueven la integración y estimulan el desarrollo de ciertas habilidades intelectuales tales como el razonamiento, la resolución de problemas, la creatividad y la capacidad de aprender a aprender. Para los profesores las tecnologías informáticas han servido hasta ahora para facilitar la búsqueda de material didáctico, contribuir a la colaboración con otros enseñantes e incitar a la planificación de las actividades de aprendizaje de acuerdo con las características de la tecnología utilizada

Según (Caraballoso Rafael,2007) Las TIC se convierten en un instrumento cada vez más indispensable en las instituciones educativas, donde pueden realizar múltiples funcionalidades como son:

- Fuente de información (hipermedia).
- Guiar el proceso de aprendizaje de los estudiantes.
- Controladora de los contenidos a enseñar.
- Ejercitar habilidades, entrenar, motivar.
- Proporcionar simulaciones.
- e-mail, foros telemáticos.

- Medio de expresión y para la creación (procesadores de textos y gráficos, editores de páginas WEB y presentaciones multimedia)
- Instrumento cognitivo y para procesar la información (hojas de cálculo, gestores de bases de datos, etc.)
- Instrumento para la gestión, ya que automatizan diversos trabajos de la gestión de los centros (secretaría, acción tutorial, asistencias, bibliotecas, etc.)

En los últimos años el avance de las TIC ha incrementado de modo considerable su presencia como medio de enseñanza a disposición de los docentes y educandos empleándose generalmente en el sistema educativo de tres maneras distintas: como objeto de aprendizaje, como medio para aprender y como apoyo al aprendizaje

.Las transformaciones observadas en los procesos de enseñanza y aprendizaje se sitúan en la línea de las teorías constructivistas que preconizan estrategias de aprendizaje que hagan de los alumnos elementos activos y dinámicos en la construcción del saber. (Ramón Gómez,2004)

1.2.4 Las TIC en Cuba.

Las limitaciones impuestas por el bloqueo estadounidense determinan sobre una política preferentemente orientada al uso social e intensivo de los recursos que brindan las TIC, buscando extender sus beneficios a la mayor parte posible de la población y las instituciones.

En Cuba se concede una prioridad máxima a la socialización de las TIC. Entre los programas desarrollados y con proyección a largo plazo se encuentran:

- Aplicaciones de las TIC a la educación con alcance para todos y en los diferentes sistemas y niveles de la educación.
- Aplicaciones a la salud y la medicina con acceso de toda la población.
- Uso multidisciplinario en la cultura.
- Informatización de la sociedad, incluyendo servicios comunitarios.
- Informatización de las actividades de producción y servicios como fuente de potenciación de la eficiencia y eficacia social de la actividad humana.
- Colaboración con numerosos países, incluyendo la educación, capacitación y transferencia de tecnología con el uso de las TIC y en la preparación para su asimilación y explotación.

Una característica común y distintiva de estos programas ha sido y será la contribución máxima al desarrollo humano, mediante una cooperación sin fines de lucro.

1.3 Programación Lineal.

La Programación Lineal es un conjunto de técnicas racionales de análisis y de resolución de problemas que tiene por objeto ayudar a los responsables en las decisiones sobre asuntos en los que intervienen un gran número de variables.

El nombre de Programación Lineal no procede de la creación de programas de ordenador, sino de un término militar, programar, que significa 'realizar planes o propuestas de tiempo para el entrenamiento, la logística o el despliegue de las unidades de combate'.

Según (Cardoza Gómez & Guerrero Seide) la programación lineal está principalmente compuesta por:

- * Un conjunto de variables de decisión
- * Una función objetivo
- * Un conjunto de restricciones

Donde se considera que las variables de decisión tienen un comportamiento lineal, tanto en la función objetivo como restricciones del problema. En este sentido, es una de las herramientas más utilizadas en la Investigación Operativa debido a que por su naturaleza se facilitan los cálculos y en general permite una buena aproximación de la realidad, proporcionando un método eficiente para determinar una decisión óptima, (o una estrategia óptima o un plan óptimo) escogida de un gran número de decisiones posibles.

Logrando el cumplimiento del objetivo trazado: ya sea la maximización o minimización de alguna cantidad de elementos que guarden relación con el problema y su solución.

En los problemas de programación lineal, un elemento esencial está dado en su contenido económico, este puede representarse de la manera siguiente:

- Se tienen algunos recursos con el concurso de los cuales se requiere satisfacer algunas necesidades, y están dados los volúmenes de recursos y las magnitudes de las necesidades.

- La utilización de los recursos y la satisfacción de las necesidades, se caracteriza por la medida en que insumen los recursos y en qué volumen satisfacen las necesidades.
- Partiendo de los datos sobre recursos, necesidades y formas alternativas posibles, se necesita conformar un plan o programa de utilización de los recursos y de satisfacción de las necesidades de manera óptima.
- En el análisis consideraremos añadidas las variables de holgura.
- Las variables esenciales representan unidades de productos.
- Los parámetros b representan a los recursos, su valor expresa el nivel de las demandas o disponibilidades.
- Las variables de holgura asumen la dimensión de las restricciones, una variable de holgura asociada a una restricción con sentido:
 - menor o igual (\leq) representa capacidad no utilizada o capacidad ociosa.
 - mayor o igual que (\geq) representa la capacidad utilizada por encima de la mínima exigida.

Por el gran alcance que posee, en nuestro país la Programación Lineal se imparte en las carreras de Contabilidad y Finanzas, Economía, en el grupo de las ingenierías (Eléctrica, Industrial, Hidráulica, del Transporte, Mecánica e Informática entre otras, Ciencias de la Computación y Cibernética, Matemática Aplicada y otras tantas, donde muchas de ellas se estudian en casi todas las universidades de nuestro país. Su versatilidad demuestra día a día la atención e importancia que se le debe dar al ser capaz de darle solución a los diferentes problemas que se le planteen.

1.3.1 Historia de la Programación Lineal.

En los siglos XVII y XVIII, grandes matemáticos como Newton, Leibnitz, Bernouilli y, sobre todo, Lagrange, que tanto habían contribuido al desarrollo del cálculo infinitesimal, se ocuparon de obtener máximos y mínimos condicionados de determinadas funciones.

Posteriormente el matemático francés Jean Baptiste-Joseph Fourier (1768-1830) fue el primero en intuir, aunque de forma imprecisa, los métodos de lo que actualmente llamamos programación lineal y la potencialidad que de ellos se deriva.

En 1941-1942 se formula por primera vez el problema de transporte, estudiado independientemente por Koopmans y Kantarovitch, razón por la cual se suele conocer con el nombre de *problema de Koopmans-Kantarovitch*. Tres años más tarde, G. Stigler plantea otro problema particular conocido con el nombre de régimen alimenticio optimal.

Mucha gente sitúa el desarrollo de la programación lineal entre los avances científicos más importantes de la mitad del siglo XX, y debemos estar de acuerdo con esta afirmación si tenemos en cuenta que su impacto desde 1950 ha sido extraordinario. Se han escrito decenas de libros de texto sobre la materia y los artículos publicados que describen aplicaciones importantes se cuentan ahora por cientos. De hecho, una proporción importante de todo el cálculo científico que se lleva a cabo en computadoras se dedica al uso de la programación lineal y a técnicas íntimamente relacionadas. (Esta proporción se estimó en un 25%, en un estudio de la IBM).

Un modelo de programación lineal proporciona un método eficiente para determinar una decisión óptima, (o una estrategia óptima o un plan óptimo) escogida de un gran número de decisiones posibles.

En todos los problemas de Programación Lineal, el objetivo es la maximación o minimización de alguna cantidad.

1.4 Visión y Misión de la Universidad de Sancti-Spíritus.

1.4.1 Visión

Se consolida la comunidad universitaria con un claustro altamente categorizado, comprometido con la Revolución y con los valores que comparten, asumiendo la prevención y el control de ilegalidades dentro de la cultura organizacional.

Se satisfacen las necesidades de las SUM, logrando perfeccionamiento y consolidación.

Aumenta la calidad de la formación integral de los estudiantes con la acreditación y crecimiento del número de carreras.

La gestión de postgrado satisface las necesidades del territorio. Se amplía la efectividad de la colaboración internacional. Se incrementa el trabajo por proyectos y la comercialización de producciones científicas, lo que permite el autofinanciamiento y desarrollo de los grupos científicos.

Se optimizan las estructuras y métodos de dirección. Se incrementa el uso de la informatización. Las redes aseguran el empleo de las TIC.

1.4.2 Misión

La UNISS, como universidad territorial universalizada sustentable, forma de manera continua profesionales integrados, revolucionarios, tecnológicos, artísticos; fieles a las tradiciones e historia y con voluntad de innovación.

1.5 El proceso de enseñanza de la Programación Lineal en la UNISS..

En la UNISS la enseñanza de la Programación Lineal se desarrolla por medio de un profesor empleando conferencias, clases prácticas, seminarios, y apoyados de cierta forma en la bibliografía existente, como es el caso de los libros de textos y el uso de la Intranet. Esta materia es impartida a estudiantes que cursan en tercer año de Contabilidad y Finanzas como un tema de la asignatura Investigación de Operaciones. Los estudiantes para poder profundizar en el tema deben hacer búsquedas a ciegas en Internet, así como visitar la biblioteca buscando algún documento que trate los diferentes temas. Para muchos estudiantes de la carrera de Contabilidad y Finanzas se torna difícil hacer búsquedas en Internet y en alguno de los casos estas búsquedas no muestran los resultados deseados. Se cuenta además con un software que permite trabajar con problemas de esta índole, el mismo solo comprende lo relacionado con la resolución de ejercicios, no explica teoría alguna y de sentarse un estudiante a trabajar sin conocer los conocimientos teóricos necesarios le sería difícil dar solución a un ejercicio.

1.6 Descripción de sistemas existentes.

En Internet se realizó una búsqueda minuciosa y se pudo llegar a la conclusión de que no se encontró un Diccionario Electrónico de Programación Lineal.

Es válido mencionar que en la actualidad los diccionarios para computadoras en aplicación de escritorio, o en formato online, se realizan bajo diferentes temáticas como son: referentes al idioma, cómo traductor, enciclopedia, acerca de algún tema como la informática, electrónica, etc.

En los casos de las aplicaciones de escritorios, los mismos son comerciales y solamente para la consulta de los datos, no se permite la edición de los mismos.

1.7 Descripción de la situación problemática.

La Programación Lineal se ha convertido en un instrumento poderoso cuando se habla de economía, ya que por mediación de la misma podemos minimizar gastos, maximizar ganancias entre otras tantas aplicaciones que tiene. Cuba por ser un país bloqueado económicamente y subdesarrollado le sería de mucha utilidad el empleo de este tipo de técnicas para lograr una buena planificación de sus recursos. Los estudiantes de Contabilidad y Finanzas como parte de la fuerza laboral de nuestro país, pasarán a manejar la parte económica de cualquier entidad del país; haciendo vital la formación que debe dar la UNISS a estos estudiantes capaces de ayudar a mejorar la economía y tomar las correctas decisiones no solo ante las posibles dificultades, sino también en la planificación y asignación de los recursos.

Es de vital importancia, ante los logros de la Programación Lineal, que los estudiantes adquieran la mayor cantidad de conocimiento con respecto a esta temática. De un estudio realizado en la biblioteca perteneciente a la UNISS se pudo comprobar que de los estudiantes graduados de Contabilidad y Finanzas en Sancti Spiritus desde el año 1998 hasta el año 2009 solo 7 estudiantes de 392 hicieron su trabajo de diploma para optar por el título de Licenciado en Contabilidad y Finanzas con temas en los que se hacía empleo de la Programación Lineal. Representando el 1.8 % de los graduados en el período mencionado anteriormente.

Haciendo un análisis por años vemos que:

Curso	Número de tesis de las que se tiene constancia digital.	Trabajos vinculados con temas relacionados a la Programación Lineal.
1998-1999	6	0
1999-2000	12	0
2000-2001	8	0
2001-2002	22	1
2002-2003	42	0
2003-2004	13	2
2004-2005	43	4
2005-2006	54	0
2006-2007	53	0
2007-2008	81	0
2008-2009	58	0

Todo esto muestra el poco interés y conocimiento que existe actualmente con respecto a la materia abordada. En la actualidad no existe un software que cumpla con la necesidad de los estudiantes de aprender esta temática, en el cual estén inmersos todos aquellos contenidos que necesitan estudiar, y de una manera amena, que cause interés, es por ello que se quiere lograr la implementación de un software diccionario en el cual aparezca todo lo vinculado con la parte teórica-explicativa relacionada con los términos existentes.

1.8 Metodologías y tecnologías actuales.

1.8.1 Metodologías actuales a considerar

Arquitectura de desarrollo por Capas.

Según (Sosa López & Hector Ortiz, 2006) La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio, sólo se modifica el nivel requerido sin tener que revisar todo el código. Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, simplemente es necesario conocer la Interfaz de Programación de Aplicaciones (Application Programming Interface - API por sus siglas en Inglés) que existe entre los niveles.

En el diseño de sistemas informáticos actual se suele usar las arquitecturas multinivel o Programación por capas, teniendo en cuenta las ventajas que esta ofrece. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables es decir que pueden ampliarse con facilidad en caso de que las necesidades aumenten.

El diseño más usado en la actualidad es el de tres capas

1. Capa de presentación
2. Capa de lógica de negocio
3. Capa de datos

1.- Capa de presentación: Esta capa provee la interfaz de usuario (IU). Aquí es donde la aplicación presenta información a los usuarios y acepta entradas o visualiza respuestas. Idealmente, la IU no desarrolla ningún procesamiento de negocios o reglas de validación de negocios. Por el contrario, la IU debería relegar sobre la capa de negocios para manipular estos asuntos. Esto es importante, especialmente hoy en día, debido a que es muy común para una aplicación tener múltiples IU.

2.- Capa de negocio: Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio - LN) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos el almacenamiento o la recuperación de los mismos.

3.- Capa de datos: La capa de acceso a datos (AD) está formada por uno o más

Sistemas Gestores de Bases de Datos (SGBD) que se encargan del almacenamiento y recuperación de los datos que necesita la capa de negocio.

Todas estas capas pueden residir en un mismo equipo. Lo más usual es que haya una multitud de computadoras donde resida la capa de presentación, es decir los clientes de la arquitectura cliente/servidor, y una computadora central que ejerce la función de servidor en dicha arquitectura. Las capas de negocio y de datos pueden residir en un mismo servidor y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más servidores. Así mismo, si el tamaño o la complejidad de la base de datos aumentara, se puede separar en varios servidores de datos, los cuales recibirán las peticiones del servidor donde reside la capa de negocio.

Si por el contrario fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en una o más computadoras que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de datos, y otra serie de ordenadores sobre los cuales corre la base de datos. En este caso se está haciendo referencia a Sistemas de Aplicaciones Distribuidas.

En una arquitectura de tres niveles, los términos Capas o Niveles no significan lo mismo ni son similares. El término capa hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: Presentación/ Lógica de Negocio/ Datos. El término nivel, corresponde a la forma como las capas lógicas, se encuentran distribuidas de forma física.

Lenguaje de Modelamiento Unificado (UML).

En (Peraza Bello & Muñoz Capote, 2006) se definió que El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. Este lenguaje fue creado por un grupo de estudiosos de la Ingeniería de Software formado por: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1995. Desde entonces, se ha convertido en el estándar internacional para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos. Con este lenguaje, se pretende unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

UML no es un lenguaje de programación sino un lenguaje de propósito general para el modelado orientado a objetos y también puede considerarse como un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes.

Entre sus objetivos fundamentales se encuentran:

1. Ser tan simple como sea posible, pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir.
2. Necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son el encapsulamiento y los componentes.
3. Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
4. Imponer un estándar mundial.

A partir del surgimiento de UML, muchas de las metodologías existentes en ese entonces, fueron adaptadas para utilizar este lenguaje, como es el caso de el Proceso Unificado de Desarrollo se concibió desde sus inicios utilizar UML.

Rational Unified Process (RUP).

El Proceso Unificado de Desarrollo, según (Alonso Pinedo & Hernández Mora, 2006) fue creado con el objetivo de producir software de alta calidad, que cumpliera con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos. Es un proceso de desarrollo de software configurable que se adapta a través de los proyectos variados en tamaños y complejidad.

El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica y determina qué entregables producir, cómo desarrollarlas, al tiempo que provee patrones. El proceso unificado es soportado por herramientas que automatizan entre otras cosas, el modelado visual, la administración de cambios y las pruebas.

Su enfoque se caracteriza por la interacción continua, desde un inicio, con el usuario, la mitigación de riesgos antes de que ocurran, liberaciones frecuentes, aseguramiento de la calidad, involucra el equipo en todas las decisiones del proyecto y permite anticiparse al cambio de requerimientos.

Las fases del ciclo de vida del software son: concepción, elaboración, construcción y transición. La concepción es definir el alcance del proyecto y definir el caso de uso. La elaboración es proyectar un plan, definir las características y cimentar la arquitectura. La construcción es crear el producto y la transición es transferir el producto a sus usuarios.

Para apoyar el trabajo con esta metodología ha sido desarrollada por la Compañía norteamericana Rational Corporation la herramienta CASE (Computer Assisted Software Engineering) Rational Rose en el año 2000. Esta herramienta integra todos los elementos que propone la metodología para cubrir el ciclo de vida de un proyecto.

Analizadas estas metodologías, concluye utilizar para llevar a cabo el proceso de desarrollo del software propuesto la metodología RUP. Esto responde fundamentalmente a que esta metodología se ha convertido en un estándar nacional e internacional para guiar el proceso de desarrollo de software y porque se cuenta también con la herramienta CASE Rational Rose del 2003, con la que se han elaborado todos los diagramas.

1.8.2 Tecnologías actuales a considerar.

Gestores de Bases de Datos

1- Microsoft Access.

Microsoft Access (Brunet González, 2007) es un sistema de gestión de bases de datos creado y modificado por la compañía Microsoft para uso personal o de pequeñas organizaciones. Es un componente de la suite Microsoft Office aunque no se incluye en el paquete básico. Su principal función es ser una potente base de datos, capaz de trabajar en sí misma o bien con conexión hacia otros lenguajes de programación, tales como Visual Basic 6.0 o Visual Basic NET. Pueden realizarse consultas directas a las tablas contenidas mediante instrucciones SQL. Internamente trae consigo el lenguaje Visual Basic for Application el cual es similar en forma al Visual Basic 6.

Permite el ingreso de datos de tipos: Numéricos, Texto, Fecha, Sí/No, OLE, Moneda, Memo y Boolean. Pueden desarrollarse aplicaciones completas basadas en Microsoft Access, pues trae consigo las herramientas necesarias para el diseño y desarrollo de formularios, para el ingreso y trabajo con datos e informes y para visualizar e imprimir la información requerida.

Su funcionamiento se basa en un motor llamado Microsoft Jet, y permite el desarrollo de pequeñas aplicaciones autónomas formadas por formularios windows y código VBA (Visual Basic para Aplicaciones). Una posibilidad adicional que ofrece es la de crear ficheros con bases de datos que pueden ser consultados por otros programas. Entre las principales funcionalidades de Access se encuentran:

- Crear tablas de datos indexadas.
- Modificar tablas de datos.
- Relaciones entre tablas (creación de bases de datos relacionales).
- Creación de consultas y vistas.
- Consultas referencias cruzadas.
- Consultas de acción (INSERT, DELETE, UPDATE).
- Formularios.
- Informes.
- Llamadas a la API de Windows.

- Interacción con otras aplicaciones que usen VBA (resto de aplicaciones de Microsoft Office, Autocad, etc.).
- Macros.

Además, permite crear frontends (programa que muestra la interfaz de usuario) de bases de datos más potentes ya que es un sistema capaz de acceder a tablas externas a través de ODBC como si fueran tablas Access.

Es un software de gran difusión entre pequeñas empresas cuyas bases de datos no requieren de excesiva potencia, ya que se integra perfectamente con el resto de las aplicaciones de Microsoft.

Entre sus mayores inconvenientes figuran que no es multiplataforma, pues sólo está disponible para sistemas operativos de Microsoft, y que no permite transacciones. Su uso es inadecuado para grandes proyectos de software que requieren tiempos de respuesta críticos o muchos accesos simultáneos a la base de datos.

2- Adaptive Server Enterprise (ASE).

Acerca del Adaptive Server Enterprise (ASE) en (Brunet González, 2007) se definió que es el motor de bases de datos relacionales insignia de la compañía Sybase. ASE es un sistema de gestión de datos, altamente escalable, de alto rendimiento, con soporte a grandes volúmenes de datos, transacciones y usuarios, y de bajo costo, que permite:

- Almacenar datos de manera segura
- Tener acceso y procesar datos de manera inteligente
- Movilizar datos

Algunas de sus principales características se enumeran a continuación:

- Un optimizador de consultas completamente renovado y más inteligente
- Técnicas de particionamiento semántico de tablas que aumentan la velocidad de acceso a los datos
- Columnas cifradas para mayor seguridad de los datos
- Columnas computadas "virtuales" y materializadas, e índices funcionales, que brindan mayor rendimiento
- Mejoras al lenguaje Transact-SQL, para mayor productividad

- Mejoras a los servicios de Java y XML en la base de datos
- Mejoras a los servicios para consumo y publicación de Servicios Web
- Herramientas mejoradas para la administración y el monitoreo
- Más rendimiento y menor costo total de propiedad

Otras características generales:

- Arquitectura VSA de Sybase
- Administrador lógico de recursos y tareas
- Múltiples esquemas de bloqueo de datos
- Copias de respaldo en línea y de alto rendimiento
- Integración transparente con orígenes de datos remotos
- Programador de tareas
- Conexiones seguras con SSL
- Soporte a LDAP para autenticación de usuarios y conectividad cliente/servidor
- Soporte a múltiples herramientas de desarrollo y lenguajes de programación, como PowerBuilder, Visual Basic, Java, C, PHP, etc.
- Soporte a múltiples protocolos de conectividad, como Open Client (propio de Sybase), ODBC, OLE DB, ADO.NET y JDBC.

ASE está soportado para la mayoría de plataformas comerciales, incluyendo:

- Windows
- Linux
- Sun Solaris
- IBM AIX
- Mac OS
- Silicon Graphics IRIX

3- Microsoft SQL Server.

Microsoft SQL Server pertenece a la familia de los sistemas de administración de base de datos, operando en una arquitectura cliente/servidor de gran rendimiento. Puede manejar perfectamente bases de datos de *TeraBytes* con millones de registros y funciona sin problemas con miles de conexiones simultáneas a los datos. SQL Server permite la creación de procedimientos almacenados, los cuales consisten en instrucciones SQL que se almacenan dentro de una base de datos de SQL Server. Esto agrega una gran practicidad, debido a que permite instrumentar consultas y transacciones muy desarrolladas dentro de los procedimientos almacenados, y después vincularse a ellos mediante la aplicación cliente. Los procedimientos almacenados presentan además otra gran ventaja, se ejecutan más rápido que instrucciones SQL independientes. (Madruga García & Morera Hernández, 2006)

4- MySQL

MySQL es un sistema de gestión de base de datos relacional lo que permite velocidad y flexibilidad ya que los datos se archivan en tablas separadas, en lugar de ser colocados en un único archivo, y son conectados por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso.

Está desarrollado en su mayor parte en ANSI C.

Al contrario de otros proyectos donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios.

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (via dbExpress), Java (con una implementación nativa del driver de Java), PHP, Python, FreeBASIC entre otros. Cada uno de estos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

MySQL funciona sobre múltiples plataformas, incluyendo GNU/Linux, Mac OS X, NetBSD, Novell Netware, OS/2 Warp, SGI IRIX, Solaris, SunOS, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP entre otros.

Fundamentación del Gestor de Base de Datos utilizado.

El gestor de base de datos utilizado fue el Microsoft Access 2003 ya que el mismo es de fácil empleo para la aplicación sobre Windows por la integración del mismo con las aplicaciones de Microsoft. Además se ajusta a los requerimientos del sistema y es el de mayor conocimiento por parte del desarrollador.

1.8.3 Lenguajes de Programación y Herramientas de Desarrollo.

1- Borland Delphi v 7.0

Según (Brunet González, 2007) Delphi es un entorno de desarrollo de software diseñado para la programación de propósito general con énfasis en la programación visual. En Delphi se utiliza como lenguaje de programación una versión moderna de Pascal llamada Object Pascal. Un uso habitual de Delphi (aunque no el único) es el desarrollo de aplicaciones visuales y de bases de datos cliente-servidor y multicapas. Debido a que es una herramienta de propósito múltiple, se usa también para proyectos de casi cualquier tipo, incluyendo aplicaciones de consola, aplicaciones de web servicios COM y DCOM, y servicios del sistema operativo. Object Pascal expande las funcionalidades del Pascal estándar:

- Soporte para la programación orientada a objetos también existente desde Turbo Pascal 5.5, pero más evolucionada en cuanto a:
 - Encapsulación: declarando partes privadas, protegidas, públicas y publicadas de las clases
 - Propiedades: concepto nuevo que luego han adaptado muchos otros lenguajes. Las propiedades permiten usar la sintaxis de asignación para setters y getters.
 - Simplificación de la sintaxis de referencias a clases y punteros.
- Soporte para manejo estructurado de excepciones, mejorando sensiblemente el control de errores de usuario y del sistema.

- Programación activada por eventos (event-driven), posible gracias a la técnica de delegación de eventos. Esta técnica permite asignar el método de un objeto para responder a un evento lanzado sobre otro objeto. Fue adoptada por Niklaus Wirth, autor del Pascal Original, e incorporada a otros de sus lenguajes como Component Pascal.

Delphi dio una implementación muy buena a la idea del uso de componentes, que son piezas reutilizables de código (clases) que pueden interactuar con el EID en tiempo de diseño y desempeñar una función específica en tiempo de ejecución. Desde un enfoque más específico de la herramienta, se catalogan como componentes todos aquellos objetos que heredan de la clase TComponent, donde se implementa la funcionalidad necesaria para interactuar con el entorno de desarrollo, la carga dinámica desde streams y la liberación de memoria mediante una jerarquía.

Una de las principales características y ventajas de Delphi es su capacidad para desarrollar aplicaciones con conectividad a bases de datos de diferentes fabricantes. El programador de Delphi cuenta con una gran cantidad de componentes para realizar la conexión, manipulación, presentación y captura de los datos, algunos de ellos liberados bajo licencias de código abierto o gratuitos. Estos componentes de acceso a datos pueden enlazarse a una gran variedad de controles visuales, aprovechando las características del lenguaje orientado a objetos, gracias al polimorfismo.

2- Java

En (el lenguaje de programación java.pdf) podemos encontrar que el Java es un lenguaje de desarrollo de propósito general, y como tal es válido para realizar todo tipo de aplicaciones profesionales. Incluye una combinación de características que lo hacen único y está siendo adoptado por multitud de fabricantes como herramienta básica para el desarrollo de aplicaciones comerciales de gran repercusión.

Una de las características más importantes es que los programas “ejecutables”, creados por el compilador de Java, son independientes de la arquitectura. Se ejecutan indistintamente en una gran variedad de equipos con diferentes microprocesadores y sistemas operativos. Pero además Java se caracteriza por:

- Ser un lenguaje intrínsecamente orientado a objetos.
- Funcionar perfectamente en red.
- Aprovechar características de la mayoría de los lenguajes modernos evitando sus inconvenientes. En particular los del C++.

- Tener una gran funcionalidad gracias a sus librerías (clases).
- NO tener punteros manejables por el programador, aunque los maneja interna y transparentemente.
- El manejo de la memoria no es un problema, la gestiona el propio lenguaje y no el programador.
- Generar aplicaciones con pocos errores posibles.

Muchos fabricantes de software están migrando sus aplicaciones a este nuevo lenguaje. Posiblemente debido a que valoran más las ventajas y, fundamentalmente la posibilidad de desarrollar una sola aplicación para múltiples plataformas así como la mayor facilidad que ofrece Java para el mantenimiento posterior de las aplicaciones

Java es uno de los lenguajes de programación más utilizados en el desarrollo de aplicaciones de código abierto y posee una variada bibliografía para aquellos programadores interesados en el aprendizaje de este lenguaje.

Por todas estas razones fue tomado en consideración el lenguaje de programación Java para la realización de este proyecto.

Existen varias herramientas de desarrollo que utilizan el lenguaje Java para la realización de diferentes aplicaciones.

Fundamentación del Lenguaje de Programación utilizado.

El lenguaje utilizado fue el Object Pascal, empleando como herramienta de desarrollo el Borland Delphi 7, puesto que el mismo brinda las prestaciones necesarias para el desarrollo del sistema, lo que facilita su implementación y acorta el tiempo de desarrollo. Además el lenguaje y la herramienta seleccionados para la elaboración del sistema son los de mayor dominio por parte del desarrollador.

1.9 Conclusiones.

En la actualidad se puede observar la necesidad de contar con un software que logre reunir la información referente a la Programación Lineal, donde a partir de su funcionamiento se pueda hacer una consulta y edición de datos, apoyando el aprendizaje y aliviando las diversas consultas bibliográficas y el engorroso acceso a Internet por parte de los estudiantes.

En la investigación se pudo reconocer las diferentes metodologías y tecnologías existentes para la construcción del Diccionario Electrónico, donde para el desarrollo del sistema propuesto fue seleccionado como lenguaje de programación el Object Pascal empleando, como herramienta para el uso de este último, el Borland Delphi 7 y como sistema gestor de base de datos se seleccionó a Microsoft Access 2003. La solución a la problemática será modelada según RUP.

Capítulo 2 Modelo de Dominio.

2.1 Introducción

Teniendo en cuenta la descripción de los procesos que se trataron anteriormente, puede afirmarse que la situación tratada en este trabajo tiene muy bajo nivel de estructuración para ser modelada como un negocio, por lo que en el presente capítulo tomando como guía la Metodología RUP, se utiliza uno de los artefactos que brinda dicha metodología: el Modelo de Dominio, los Requerimientos Funcionales y No Funcionales, el Diagrama de Casos de Uso y la descripción de cada uno, los cuales ayudan a modelar y describir la solución propuesta. Además, se presenta una descripción detallada de las reglas de negocio con el objetivo de asegurar el cumplimiento de las restricciones que existen en el dominio.

2.2 Descripción del modelo de dominio.

Un Modelo del Dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos. La modelación del dominio tiene como objetivo fundamental la comprensión y descripción de las clases más importantes en el sistema.

En el modelo del dominio empleado en este trabajo se definen las siguientes entidades: Administrador, Usuario, Profesor; y clases principales: Palabra, Artículo y sus respectivos informes según corresponda.

La esencia del funcionamiento de este dominio consiste en el constante acceso de los usuarios a los datos del diccionario con el objetivo de adquirir conocimientos teóricos de la Programación Lineal, haciendo uso de las definiciones y los artículos, y de la construcción de los reportes de datos según se solicite.

2.2.1 Principales conceptos asociados al dominio del problema.

Concepto	Descripción
Usuario	Persona que interactúa con el sistema, busca los datos que necesita

Administrador	Persona que tiene control total sobre el sistema, puede realizar las modificaciones necesarias, puede actuar como usuario
Palabra	Lo que brinda el sistema a los usuarios, se compone por la palabra y su significado.
Reporte de Palabra	Grupo de palabras reunidas por el usuario
Artículo	Lo que brinda el sistema a los usuarios. Se compone por la información adicional que sea necesaria para comprender la programación lineal.
Reporte de Artículos	Grupo de artículos reunidos por el usuario
Profesor	Usuario registrado con permiso de la edición de los datos existentes

2.2.2 Modelo de objetos del dominio.

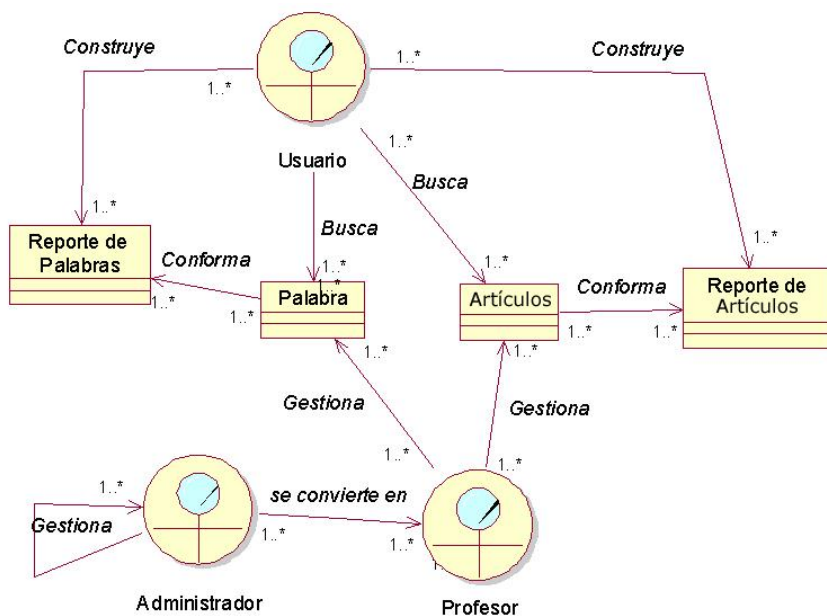


Figura 2.1 Modelo de objetos del dominio

2.3 Requerimientos.

2.3.1 Requerimientos funcionales.

Los requerimientos funcionales son declaraciones de los servicios o funciones que proveerá el sistema, de la manera en que este reaccionará a entradas particulares. Estos dependen del tipo de software y del sistema que se desarrolle y de los posibles usuarios del software. Cuando se expresan como requerimientos del usuario, habitualmente se describen de forma general mientras que los requerimientos funcionales del sistema describen con detalle la función de este, sus entradas y salidas, excepciones, etc.

Listado de Requerimientos Funcionales:

1. Gestionar Palabra
 - 1.1 Insertar Palabra
 - 1.2 Actualizar Palabra
 - 1.3 Eliminar Palabra
 - 1.4 Actualizar Autor
2. Gestionar Usuario
 - 2.1 Insertar Usuario
 - 2.2 Actualizar Usuario
 - 2.3 Eliminar Usuario
3. Gestionar Artículo
 - 3.1 Insertar Artículo
 - 3.2 Actualizar Artículo
 - 3.3 Eliminar Artículo
 - 3.4 Actualizar Autor
4. Buscar Palabra
5. Buscar Artículo
6. Mostrar Palabra
7. Mostrar Escrito

8. Iniciar Sesión
9. Cerrar Sesión
10. Cambiar Contraseña
11. Preparar Reporte de Palabras
12. Preparar Reporte de Artículos
13. Generar Reporte de Palabras
14. Generar Reporte de Artículos

2.3.2 Requerimientos no funcionales.

Los requerimientos no funcionales describen las restricciones del sistema o del proceso de desarrollo; no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de este como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida, en cuanto a prestaciones, atributos de calidad y la representación de datos que se utiliza en la interfaz del sistema.

Listado de Requerimientos no funcionales del sistema.

Apariencia o interfaz externa.

- La interfaz se ajustará al estándar de ventanas que el sistema operativo Windows ha establecido e internacionalizado.
- La interfaz estará diseñada de modo tal que el usuario pueda tener en todo momento el control de la aplicación, lo que le permitirá ir de un punto a otro dentro de ella con gran facilidad. Se cuidará porque la aplicación sea lo más interactiva posible. La interfaz será uniforme de manera que a pesar de tener varios módulos, el usuario los identifique como un mismo sistema.

Usabilidad.

- La explotación del sistema ayudará a la actividad educativa del centro donde se ponga en práctica gracias a las comodidades brindadas por esta aplicación, la rapidez y organización presentado en sus palabras y significados.

- Tendrá una documentación básica que comprenda los aspectos generales a tener en cuenta para trabajar con la aplicación.
- El sistema podrá ser usado por aquellas personas que no tengan experiencia en el uso de la computadora, sólo necesitarían un ligero entrenamiento sobre el funcionamiento de los principales elementos de una interfaz estándar en el ambiente del sistema operativo Windows (uso del Mouse, manejo de menús, botones, cuadros de texto, etc.)

Rendimiento.

- Se requiere de una capacidad de procesamiento media para ejecutar algoritmos de búsqueda y muestra de datos.

Soporte.

- Las pruebas del sistema se realizarán en el centro Universitario “José Martí” de Sancti Spíritus. Dichas pruebas permitirán evaluar en la práctica la funcionalidad y las ventajas de este nuevo producto.
- El sistema deberá dar las posibilidades a futuras mejoras y nuevas opciones que se le quieran incorporar.

Seguridad.

- La información original del software sólo podrá ser modificada por los administradores designados. El sistema utiliza la política de impedir accesos no autorizados que pudieran introducir errores en la información original.
- Existen las cuentas de acceso en caso de la necesidad de gestionar datos

Portabilidad.

- El sistema será utilizado sobre el sistema operativo Windows

Políticos-culturales.

- El nivel social, cultural o étnico; no determinarán una prioridad o limitante a la hora de brindar los servicios que ofrece el producto.

Ayuda y documentación en línea.

- Debe disponerse de una ayuda bien detallada sobre las principales opciones del sistema. Además, se deben tener disponible otros documentos para consulta general.

Software.

- Se debe disponer de Windows 98 o superior, para la instalación de la aplicación

2.4 Descripción del Sistema Propuesto.

El software se iniciará con una pantalla de bienvenida hacia los usuarios, que al cerrarse mostrará el menú principal del sistema.

Al principio solamente se tendrá acceso a mostrar palabras, mostrar escritos, buscar información, generar los reportes y a entrar al sistema mediante una contraseña si está registrado.

Para generar los reportes, antes tienen que prepararlos desde el área de muestra de los datos según el caso (palabra o escrito), los reportes podrán ser generados en dos formatos: uno preparado directamente para imprimir y el otro se guardará en formato de texto enriquecido (RTF) en la misma carpeta de la aplicación, escogido por ser un formato trabajado por muchos procesadores de texto.

Al entrar al sistema mediante una contraseña, se obtendrán más opciones según el tipo de usuario que sea descrito: si es profesor obtendrá acceso a la gestión de los datos (insertar, actualizar, eliminar) y al cambio de su contraseña; y si es administrador tendrá acceso a la gestión de los datos, gestión de los usuarios (insertar, actualizar, eliminar, cambiar contraseña de los usuarios) y al cambio de su contraseña.

El sistema está validado para evitar entradas erróneas y de ocurrir estas se emitirá un mensaje de alerta correspondiente.

En cada ventana al presionar F1 se emitirá la ayuda correspondiente con una explicación detallada de los elementos según sea el caso.

2.5 Modelo de casos de uso del sistema.

El modelado de Casos de Uso es la técnica más efectiva y a la vez la más simple que emplean los desarrolladores de software para modelar los requisitos del sistema desde la perspectiva del usuario. El modelo de casos de uso consiste en actores y casos de uso. Los actores representan usuarios y otros sistemas que interactúan con el sistema y los casos de uso representan el comportamiento del sistema, los escenarios que el sistema atraviesa en respuesta a un estímulo desde un actor. En esencia, el modelado de Casos de Uso nos describe lo que hace el sistema para cada tipo de usuario y nos ofrece un medio correcto para el análisis, el diseño y las pruebas.

2.5.1 Actores del sistema.

Un actor es aquel que interactúa con el sistema, sin ser parte de él y puede asumir el rol que juega una o varias personas, un equipo o un sistema automatizado. A continuación se definen los actores identificados en el sistema propuesto:

Actor	Descripción
Administrador	El administrador es el que va a tener control total del sistema, interactuando completamente con el mismo y haciendo que funcione en todo momento. Tendrá acceso a todos los requerimientos funcionales del sistema
Usuario	Cualquier usuario que interactúe con el sistema. Solamente tendrá acceso a los requerimientos funcionales referentes a la muestra de datos, búsqueda de información y la generación de reportes según sea necesario
Profesor	Usuario registrado del sistema, podrá interactuar como el actor Usuario y aparte tendrá acceso a los requerimientos funcionales: Gestionar Escrito, Gestionar Palabra

Tabla 2.1. Descripción de los actores del sistema

2.5.2 Diagramas de casos de uso del sistema.

Los actores interactúan y usan el sistema a través de casos de uso. Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario.

En nuestro trabajo los casos de uso del sistema quedan representados por:

1. Gestionar Palabra
2. Gestionar Usuario
3. Gestionar Artículo
4. Buscar Palabra
5. Buscar Artículo
6. Mostrar Palabra
7. Mostrar Escrito
8. Iniciar Sesión
9. Cerrar Sesión
10. Cambiar Contraseña
11. Preparar Reporte de Palabras
12. Preparar Reporte de Artículos
13. Generar Reporte de Palabras

14. Generar Reporte de Artículos

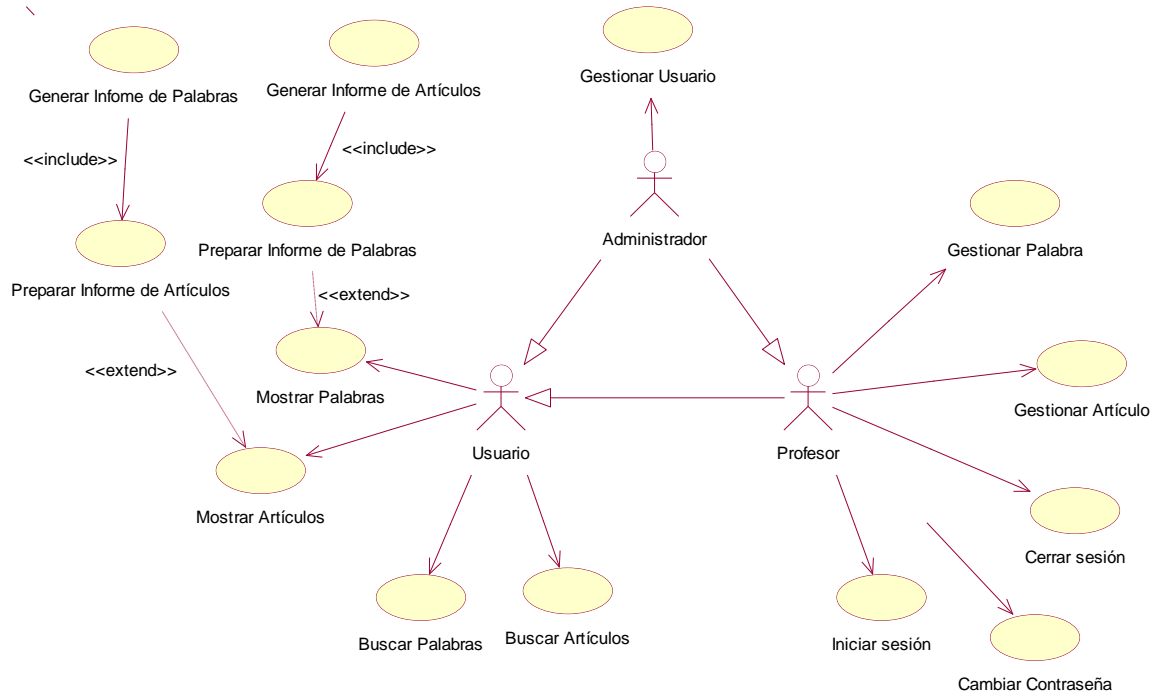


Figura 2.2 Diagramas de casos de uso del sistema

2.4.3 Descripción de los casos de uso del sistema.

Después de haber representado los casos de usos del sistema pasamos a describirlos. Estos serán referenciados a través de los requerimientos funcionales planteados en el epígrafe 2.3.1.

Caso de Uso	Gestionar Palabra
Actores	Profesor
Propósito	Permite al Profesor la gestión de una o más palabras en el sistema

<u>Resumen</u>	
<p>El caso de uso se inicia cuando el Profesor solicita la inserción, cambio o eliminación de una palabra, el sistema muestra el formulario correspondiente en cada caso, el cual deberá ser llenado correctamente con la incorporación de todos los valores; en caso de error, el mismo será notificado para una rectificación en el formulario. El caso de uso finaliza con la culminación exitosa de la acción realizada.</p> <p>La actualización incluye actualizar el autor de la palabra cuando sea necesario</p>	
Referencias	R1
Precondiciones	<p>-El sistema estará conectado correctamente a la base de datos</p> <p>-Existirá al menos un profesor autenticado en el sistema</p>
Poscondiciones	La información entrada por el Profesor es almacenada en la base de datos del sistema
Requisitos Especiales	-
Prototipo	- Anexo A.1

Tabla 2.2. Descripción del caso de uso de sistema <Gestionar Palabras>

Caso de Uso	Gestionar Usuario
Actores	Administrador
Propósito	Permite al Administrador la gestión de uno o más Usuarios en el sistema
<u>Resumen</u>	
<p>El caso de uso se inicia cuando el Administrador solicita la inserción, cambio o eliminación de un Usuario, el sistema muestra el formulario correspondiente en cada caso, el cual deberá ser llenado correctamente con la incorporación de todos los valores; en caso de error, el mismo será notificado para una rectificación en el</p>	

formulario. El caso de uso finaliza con la culminación exitosa de la acción realizada. Actualizar Usuario incluye Actualizar los datos y las contraseñas de los usuarios	
Referencias	R2
Precondiciones	-El sistema estará conectado correctamente a la base de datos -Existirá al menos un Administrador autenticado en el sistema
Poscondiciones	La información entrada por el Administrador es almacenada en la base de datos del sistema
Requisitos Especiales	-
Prototipo	- Anexo A.2

Tabla 2.3. Descripción del caso de uso de sistema <Gestionar Usuario>

Caso de Uso		Gestionar Artículo	
Actores		Profesor	
Propósito		Permite al Profesor la gestión de uno o más Artículos en el sistema	
<u>Resumen</u>			
<p>El caso de uso se inicia cuando el Profesor solicita la inserción, cambio o eliminación de un Artículo, el sistema muestra el formulario correspondiente en cada caso, el cual deberá ser llenado correctamente con la incorporación de todos los valores; en caso de error, el mismo será notificado para una rectificación en el formulario. El caso de uso finaliza con la culminación exitosa de la acción realizada.</p> <p>Insertar Artículo incluye insertar un nuevo tipo de artículo por si es necesario.</p> <p>Actualizar Artículo incluye actualizar autor y actualizar tipo por si es necesario</p>			
Referencias		R3	

Precondiciones	-El sistema estará conectado correctamente a la base de datos -Existirá al menos un Profesor autenticado en el sistema
Poscondiciones	La información entrada por el Profesor es almacenada en la base de datos del sistema
Requisitos Especiales	-
Prototipo	- Anexo A.3

Tabla 2.4. Descripción del caso de uso de sistema <Gestionar Artículo>

Caso de Uso	Buscar Palabra
Actores	Usuario
Propósito	Permite que el usuario tenga acceso a las palabras guardadas del sistema.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita una búsqueda en el sistema de las palabras guardadas, el sistema muestra el formulario correspondiente para dicha acción, que al ejecutarse tomará las palabras que correspondan a la búsqueda; o algunas similares a la misma y las muestra en pantalla para su uso posterior por parte del usuario. El caso de uso culmina cuando la información es mostrada al usuario.	
Referencias	R4
Precondiciones	Deben existir palabras en el sistema
Poscondiciones	Los datos solicitados son mostrados al usuario
Requisitos Especiales	-
Prototipo	- Anexo A.4

Tabla 2.5. Descripción del caso de uso de sistema <Buscar Palabra>

Caso de Uso	Buscar Artículo
Actores	Usuario
Propósito	Permite que el usuario tenga acceso a los artículos guardadas del sistema.
<u>Resumen</u> <p>El caso de uso se inicia cuando el usuario solicita una búsqueda en el sistema de los artículos guardados, el sistema muestra el formulario correspondiente para dicha acción, que al ejecutarse tomará los artículos que correspondan a la búsqueda; o algunos similares a la misma y los muestra en pantalla para su uso posterior por parte del usuario. El caso de uso culmina cuando la información es mostrada al usuario.</p>	
Referencias	R5
Precondiciones	Deben existir artículos en el sistema
Poscondiciones	Los datos solicitados son mostrados al usuario
Requisitos Especiales	-
Prototipo	- Anexo A.5

Tabla 2.6. Descripción del caso de uso de sistema <Buscar Artículo>

Caso de Uso Mostrar Palabras	
Actores	Usuario
Propósito	Permite que el usuario tenga acceso a las palabras guardadas del sistema.
<u>Resumen</u> <p>El caso de uso se inicia cuando el usuario solicita que se muestren las palabras del diccionario, el sistema muestra el formulario correspondiente para la selección y posterior muestra de la palabra deseada. El caso de uso culmina cuando la información es mostrada al usuario.</p> <p>Aquí es donde se puede seleccionar si se desea las palabras que conformarán el</p>	

Informe de Palabras	
Referencias	R6 CU11(Extend)
Precondiciones	- Debe existir conexión a la base de datos - Deben existir palabras en el sistema
Poscondiciones	Los datos solicitados son mostrados al usuario
Requisitos Especiales	-
Prototipo	- Anexo A.6

Tabla 2.7. Descripción del caso de uso de sistema <Mostrar Palabras>

Caso de Uso Mostrar Artículos	
Actores	Usuario
Propósito	Permite que el usuario tenga acceso a los artículos guardados en el sistema.
<p><u>Resumen</u></p> <p>El caso de uso se inicia cuando el usuario solicita que se muestren los artículos del diccionario, el sistema muestra el formulario correspondiente para la selección y posterior muestra de los artículos deseados. El caso de uso culmina cuando la información es mostrada al usuario.</p> <p>Aquí es donde se puede seleccionar si se desea los artículos que conformarán el Informe de Artículos</p>	
Referencias	R7 CU12(Extend)
Precondiciones	- Debe existir conexión a la base de datos - Deben existir artículos en el sistema
Poscondiciones	Los datos solicitados son mostrados al usuario
Requisitos Especiales	-

Prototipo	- Anexo A.7
-----------	-------------

Tabla 2.8. Descripción del caso de uso de sistema <Mostrar Artículos>

Caso de Uso	Iniciar Sesión
Actores	Profesor
Propósito	Permite al Profesor entrar al sistema y comenzar a utilizarlo como usuario registrado
<u>Resumen</u> <p>El caso de uso se inicia cuando el Profesor solicita la entrada al sistema mediante contraseña, se muestra el formulario correspondiente el cual deberá ser usado para entrar como usuario registrado; en caso de error, el mismo será notificado para una rectificación en el formulario. El caso de uso finaliza cuando el profesor entra al programa.</p>	
Referencias	R8
Precondiciones	El programa estará trabajando correctamente
Poscondiciones	El sistema se ampliará con nuevas funciones según el tipo de usuario
Requisitos Especiales	-
Prototipo	- Anexo A.8

Tabla 2.9. Descripción del caso de uso de sistema < Iniciar Sesión >

Caso de Uso	Cerrar Sesión
Actores	Profesor
Propósito	Permite al Profesor salir del sistema y dejar funcionando el diccionario sin alguien conectado

<u>Resumen</u>	
El caso de uso se inicia cuando el Profesor solicita su salida al sistema. El software actuará cerrando la sesión del usuario. El caso de uso finaliza con la salida del profesor del sistema.	
Referencias	R9
Precondiciones	El programa estará trabajando correctamente - Existe alguien conectado
Poscondiciones	El sistema volverá a su estado normal, mostrando las funciones iniciales
Requisitos Especiales	-
Prototipo	- Anexo A.9

Tabla 2.10. Descripción del caso de uso de sistema < Cerrar Sesión >

Caso de Uso	Cambiar Contraseña
Actores	Profesor
Propósito	Permite al Profesor cambiar su contraseña
<u>Resumen</u>	
El caso de uso se inicia cuando el Profesor solicita cambiar su contraseña, el sistema muestra el formulario correspondiente el cual deberá ser llenado correctamente con la incorporación de los valores correspondientes para el cambio de contraseña. El caso de uso finaliza con la actualización de los datos.	
Referencias	R10
Precondiciones	El programa estará trabajando correctamente
Poscondiciones	La información entrada por el Profesor es almacenada en la base de datos del

	sistema
Requisitos Especiales	-
Prototipo	- Anexo A.10

Tabla 2.11. Descripción del caso de uso de sistema < Cambiar Contraseña >

Caso de Uso	Preparar Reporte de Palabras
Actores	Usuario
Propósito	Permite al usuario preparar un reporte con las palabras deseadas
<u>Resumen</u> El caso de uso se inicia cuando el usuario desea preparar un reporte de palabras, las palabras se seleccionan desde Mostrar Palabras y se guardarán en la lista correspondiente.	
Referencias	R11
Precondiciones	Deben existir palabras en el sistema El CU Mostrar Palabras estará en uso actual
Poscondiciones	Los datos seleccionados son guardados por el sistema
Requisitos Especiales	-
Prototipo	- Anexo A.11

Tabla 2.12. Descripción del caso de uso de sistema <Preparar Reporte de Palabras>

Caso de Uso	Preparar Reporte de Artículos
Actores	Usuario
Propósito	Permite al usuario preparar un reporte con los artículos deseados

<u>Resumen</u>	
El caso de uso se inicia cuando el usuario desea preparar un reporte de artículos, los artículos se seleccionan desde Mostrar Artículos y se guardarán en la lista correspondiente.	
Referencias	R12
Precondiciones	Deben existir artículos en el sistema El CU Mostrar Artículos estará en uso actual
Poscondiciones	Los datos seleccionados son guardados por el sistema
Requisitos Especiales	-
Prototipo	- Anexo A.12

Tabla 2.13. Descripción del caso de uso de sistema <Preparar Reporte de Artículos>

Caso de Uso	Generar Reporte de Palabras
Actores	Usuario
Propósito	Permite al usuario generar un reporte con las palabras guardadas
<u>Resumen</u>	
<p>El caso de uso se inicia cuando el usuario solicita generar un reporte, el sistema para llevar a cabo esta función obligatoriamente tendrá que ejecutar el caso de uso 9, donde se guardan las palabras deseadas para que se sean publicadas en el Reporte de Palabras, el formulario correspondiente muestra los datos seleccionados por el usuario para la posterior construcción del Reporte de Palabras. Se mostrarán 2 opciones : Generar el Reporte para imprimirlo directamente y generar un documento RTF para guardar las palabras seleccionadas</p> <p>El caso de uso finaliza con la generación del Reporte mediante alguna de las opciones mencionadas.</p>	
Referencias	R13, CU11(include)

Precondiciones	Deben existir palabras en el sistema
Poscondiciones	Los datos seleccionados son mostrados al usuario Se genera el reporte según la opción seleccionada
Requisitos Especiales	-
Prototipo	- Anexo A.13

Tabla 2.14. Descripción del caso de uso de sistema <Generar Reporte de Palabras>

Caso de Uso	Generar Reporte
Actores	Usuario
Propósito	Permite al usuario generar un reporte con las palabras guardadas
<p><u>Resumen</u></p> <p>El caso de uso se inicia cuando el usuario solicita generar un reporte, el sistema para llevar a cabo esta función obligatoriamente tendrá que ejecutar el caso de uso 10, donde se guardan los artículos deseados para que se sean publicadas en el Reporte de Artículos, el formulario correspondiente muestra los datos seleccionados por el usuario para la posterior construcción del Reporte de Artículos. Se mostrarán 2 opciones : Generar el Reporte para imprimirlo directamente y generar un documento RTF para guardar las palabras seleccionadas</p> <p>El caso de uso finaliza con la generación del Reporte mediante alguna de las opciones mencionadas.</p>	
Referencias	R14, CU12(include)
Precondiciones	Deben existir artículos en el sistema
Poscondiciones	Los datos seleccionados son mostrados al usuario Se genera el reporte según la opción

	seleccionada
Requisitos Especiales	-
Prototipo	- Anexo A.14

Tabla 2.15. Descripción del caso de uso de sistema <Generar Reporte de Artículos>

2.5 Conclusiones.

En el presente capítulo queda definido el modelo de objetos del dominio del Diccionario Electrónico. Se definieron y representaron los principales conceptos asociados al problema y su solución. Tras lograr la modelación del dominio se ha logrado una mejor comprensión de los problemas que se tienen que resolver, y así derivar los requerimientos para la aplicación.

Capítulo 3 Construcción de la solución propuesta.

3.1 Introducción.

En el presente capítulo se realiza una descripción de la construcción de la solución propuesta. En esta descripción se ha utilizado el diagrama de clases del diseño, el diagrama de clases persistentes y el diagrama del modelo físico de datos, artefactos que propone la Metodología de RUP. Para describir los elementos fundamentales de la implementación se muestra el diagrama de implementación, artefacto que también es propuesto por la metodología utilizada. Además se describen los principios de diseño utilizados y las consideraciones de codificación que se tuvieron en cuenta en la implementación de este sistema.

3.2 Diagrama de clases del diseño.

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Normalmente contiene la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información sobre los tipos de atributos.
- Navegabilidad.
- Dependencias.

A diferencia del modelo conceptual, un diagrama de este tipo contiene las definiciones de las entidades del software en vez de conceptos del mundo real. El UML no define concretamente un elemento denominado “Diagrama de clases del diseño”, sino que se sirve de un término más genérico: “Diagrama de clases”. El término diseño se incluye para recalcar que se trata de una perspectiva desde el punto de vista del diseño de las entidades de software y no de una concepción analítica sobre los conceptos del dominio.

Los diagramas se presentan a continuación, y los elementos principales mencionados en los mismos serán descritos en los anexos B.1 y B.2.

En las figuras 3.13 y 3.14, Ppipeline y Ppreport pertenecen al componente para imprimir el Informe; y Ekrtf pertenece al componente para generar el documento Rtf con los datos del Informe.

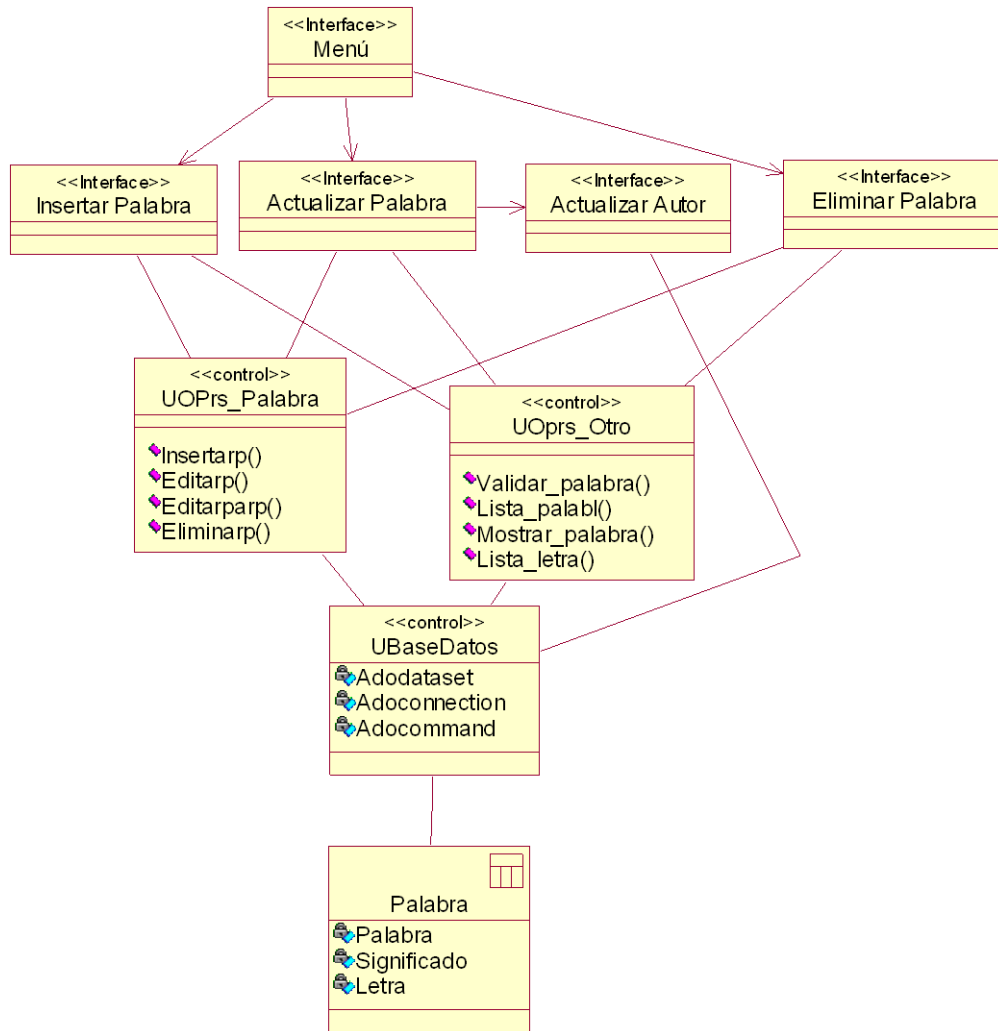


Figura 3.1 Diagrama de clases del diseño CU Gestionar Palabra

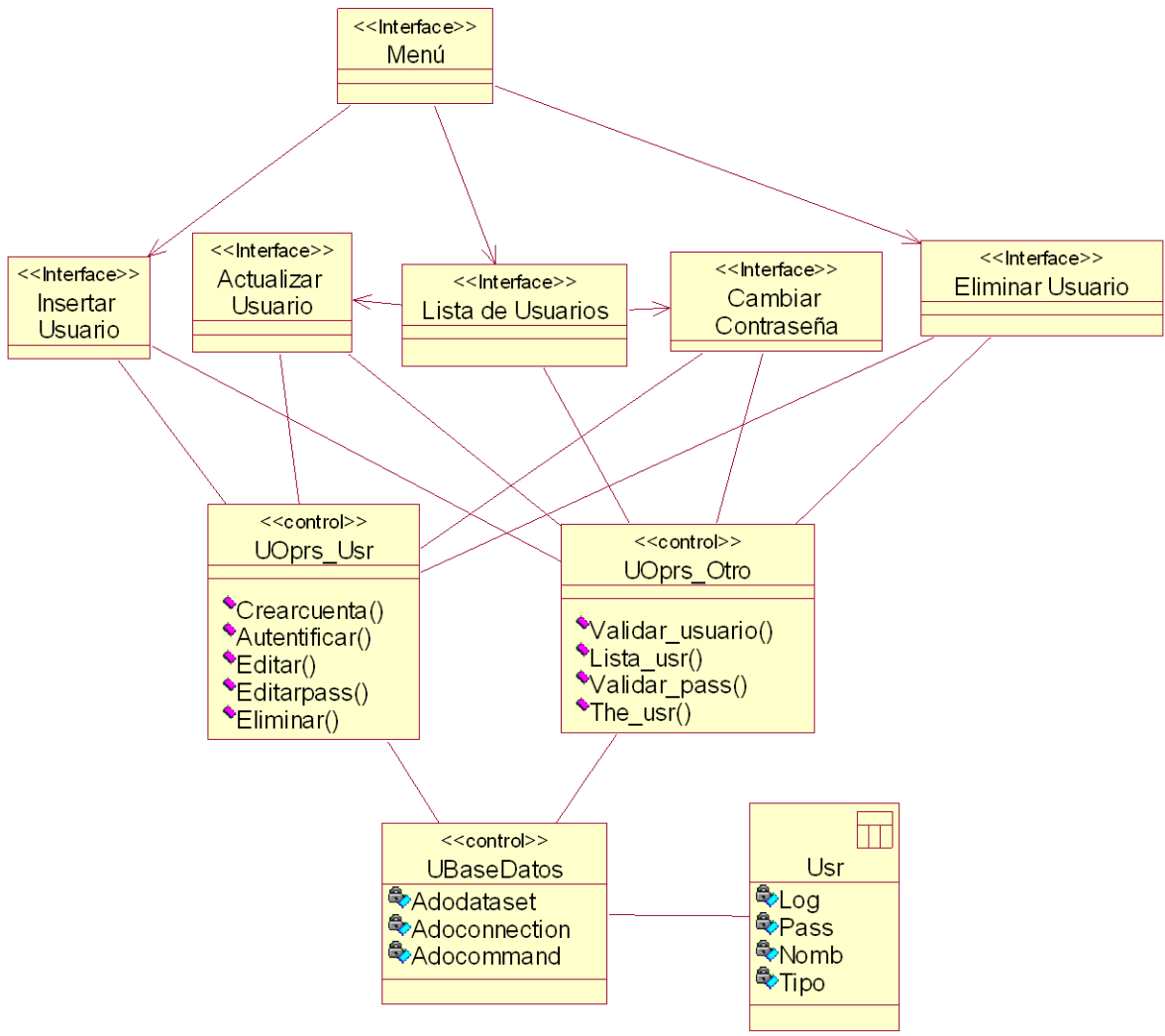


Figura 3.2 Diagrama de clases del diseño CU Gestionar Usuario

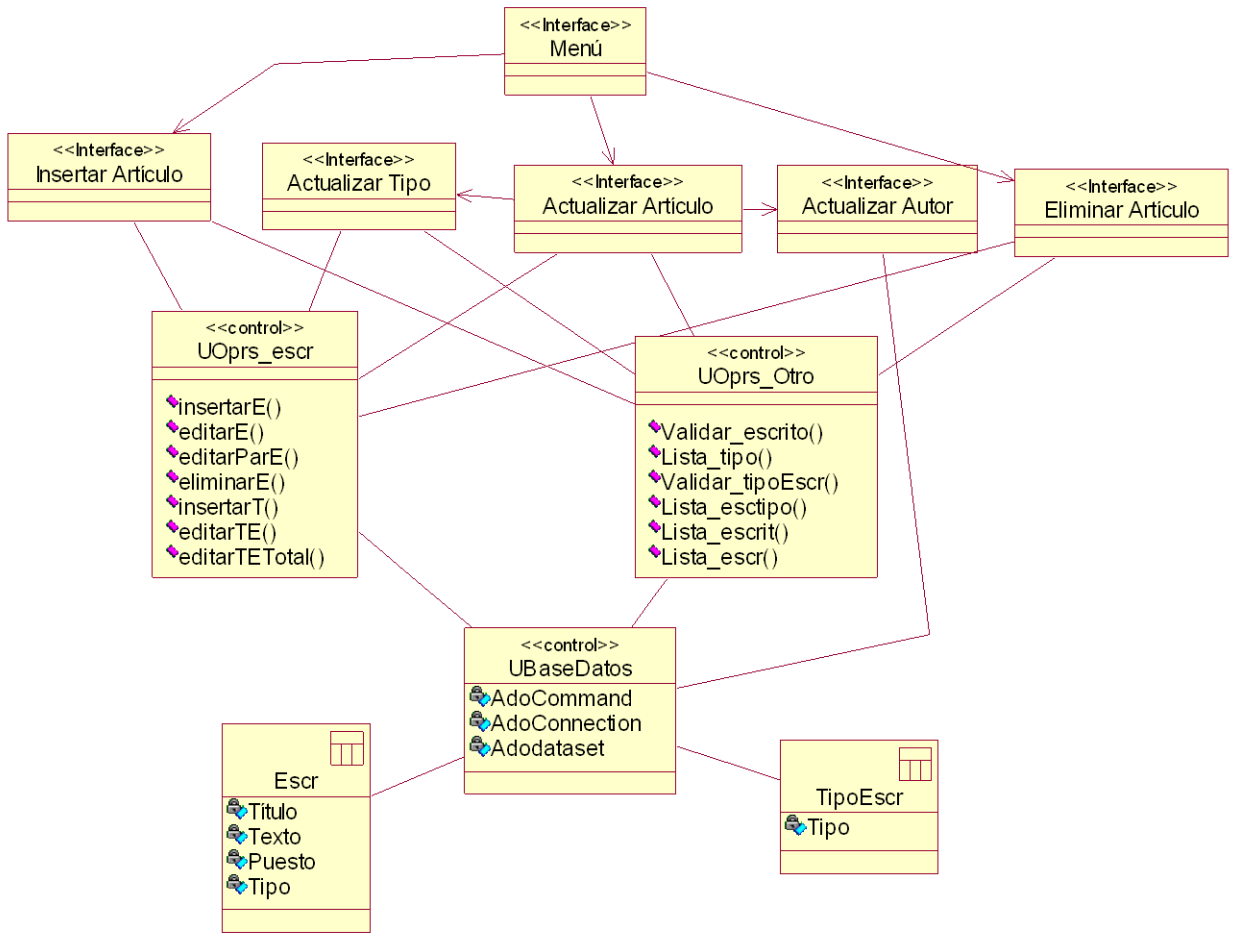


Figura 3.3 Diagrama de clases del diseño CU Gestionar Artículo

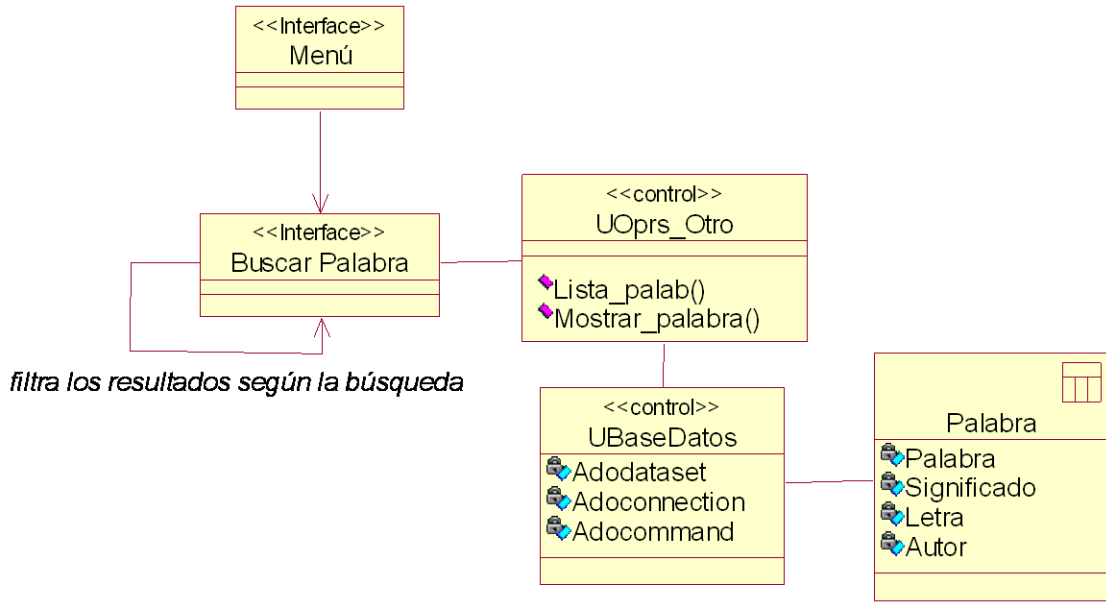


Figura 3.4 Diagrama de clases del diseño CU Buscar Palabra

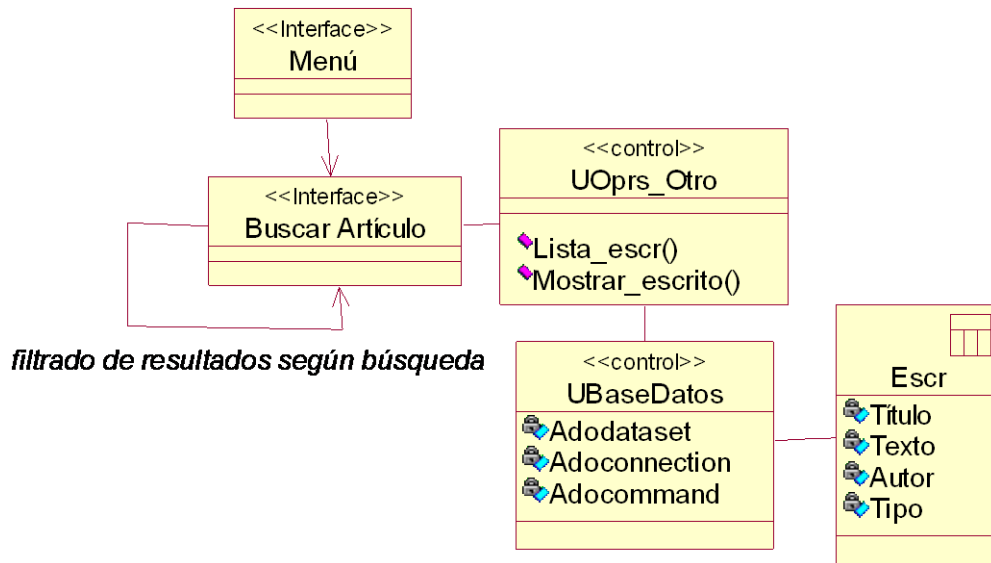


Figura 3.5 Diagrama de clases del diseño CU Buscar Artículo

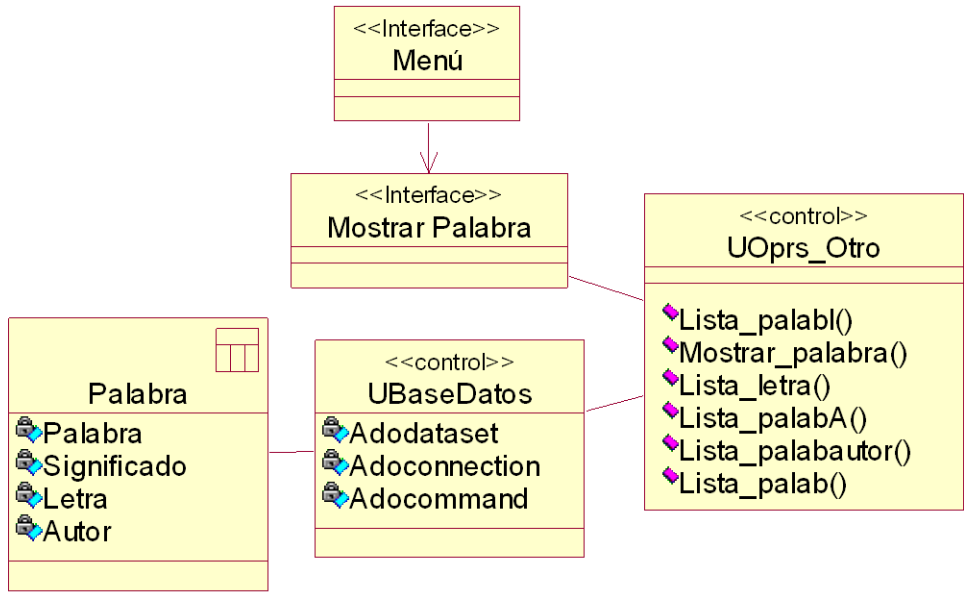


Figura 3.6 Diagrama de clases del diseño CU Mostrar Palabra

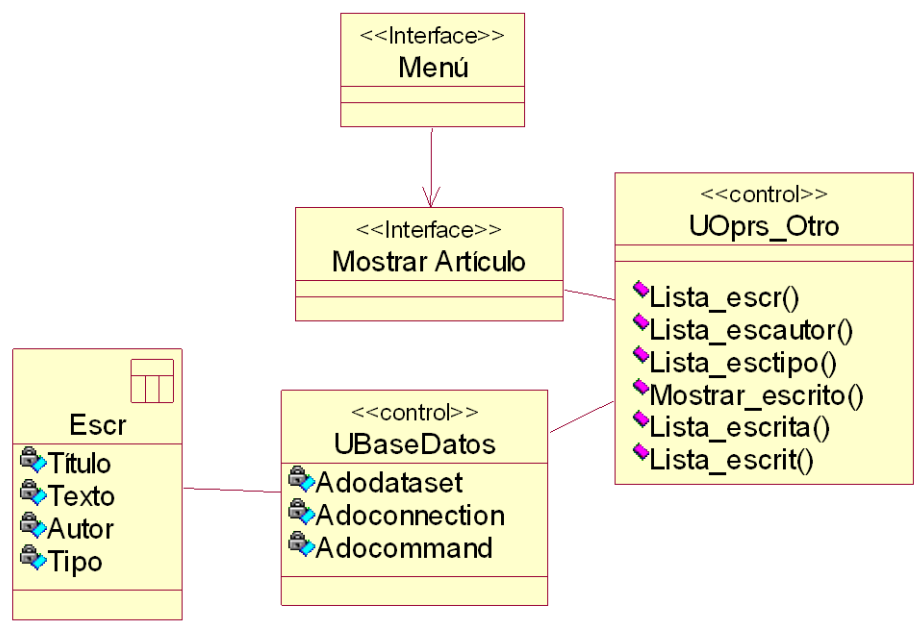


Figura 3.7 Diagrama de clases del diseño CU Mostrar Artículo

El menú se expande según el tipo de usuario registrado

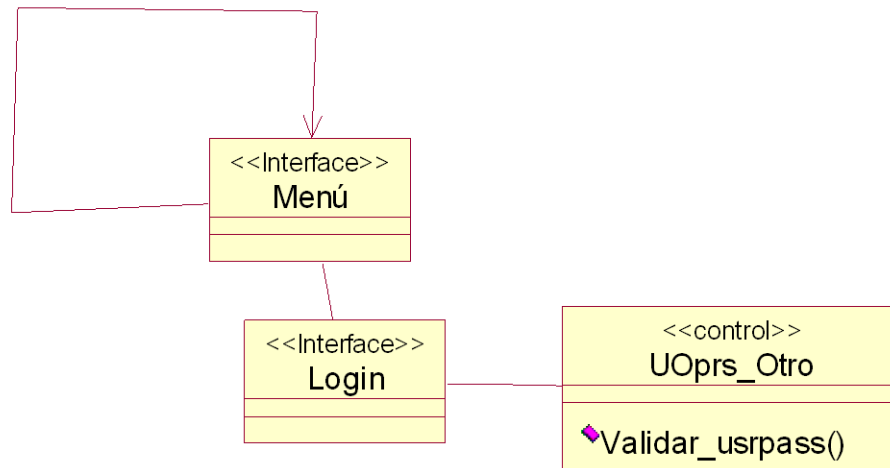


Figura 3.8 Diagrama de clases del diseño CU Iniciar Sesión

El menú regresa su forma inicial. Las funciones mostradas son básicas

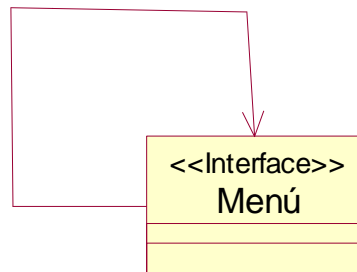


Figura 3.9 Diagrama de clases del diseño CU Cerrar Sesión

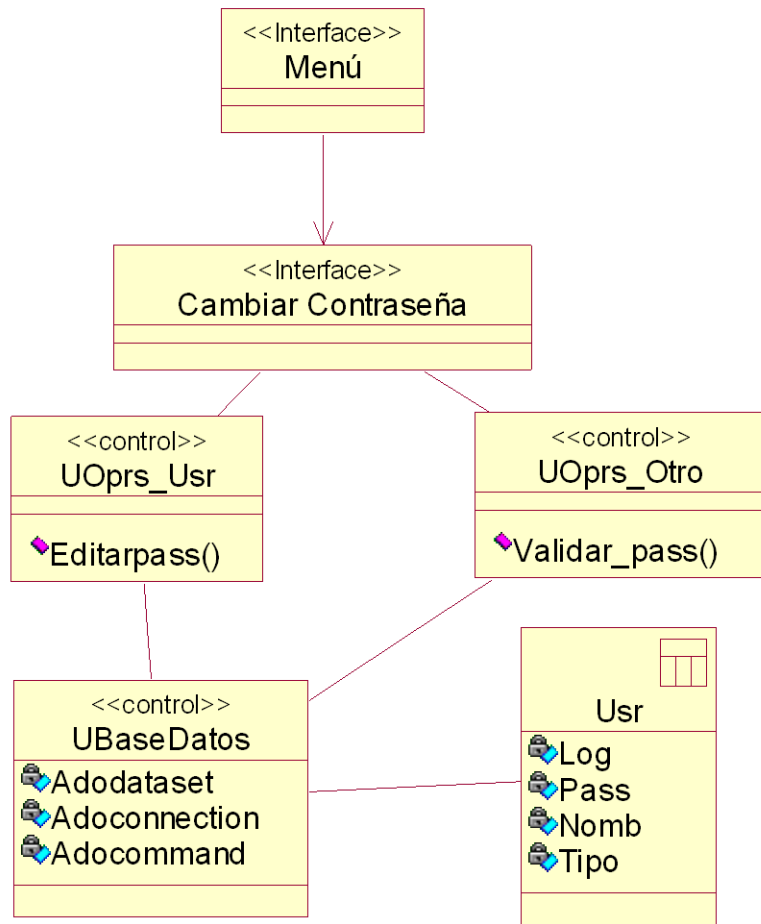


Figura 3.10 Diagrama de clases del diseño CU Cambiar Contraseña

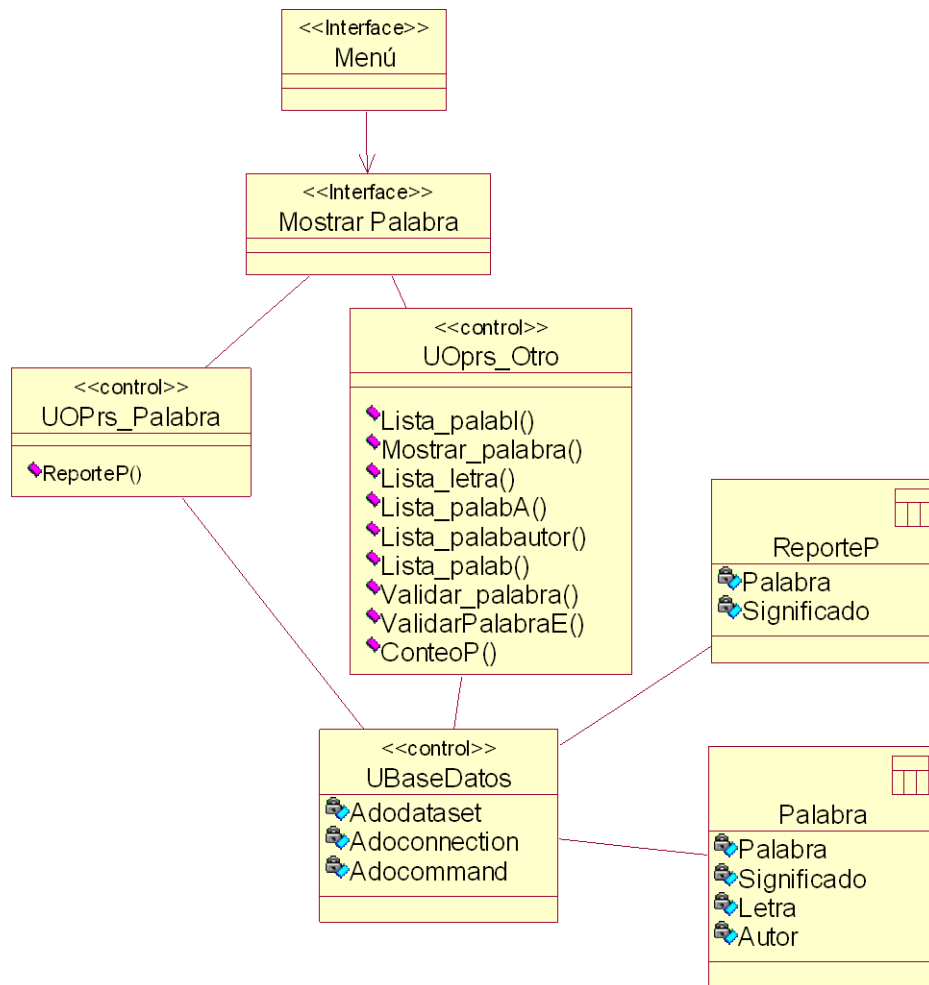


Figura 3.11 Diagrama de clases del diseño CU Preparar Reporte de Palabras

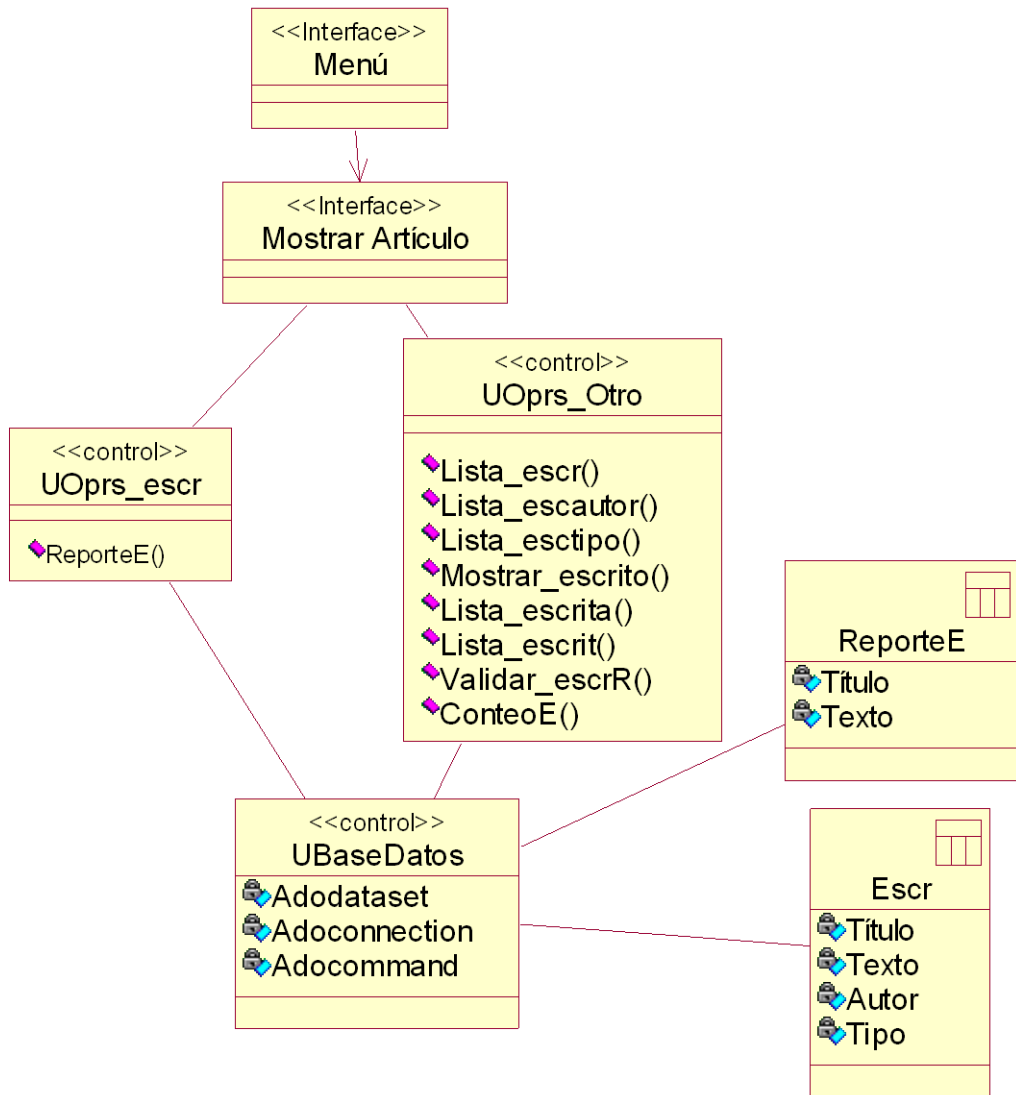


Figura 3.12 Diagrama de clases del diseño CU Preparar Reporte de Artículos

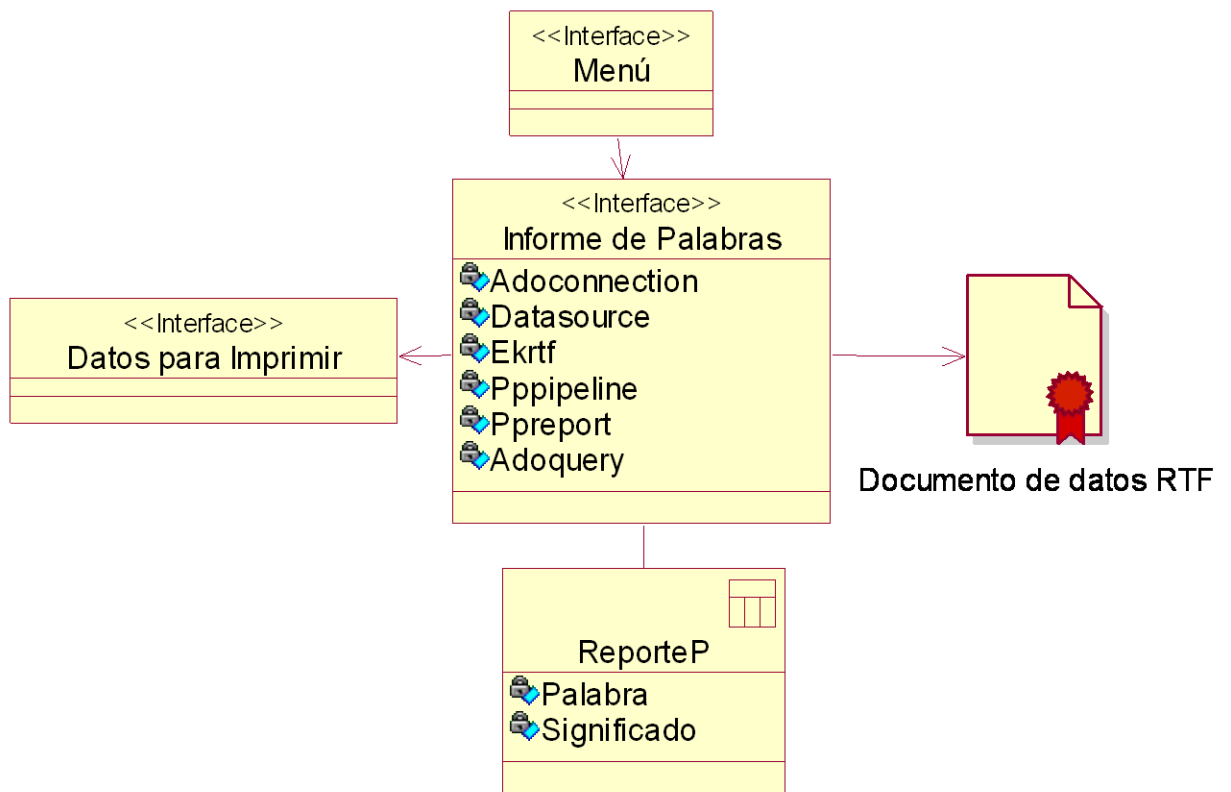


Figura 3.13 Diagrama de clases del diseño CU Generar Reporte de Palabras

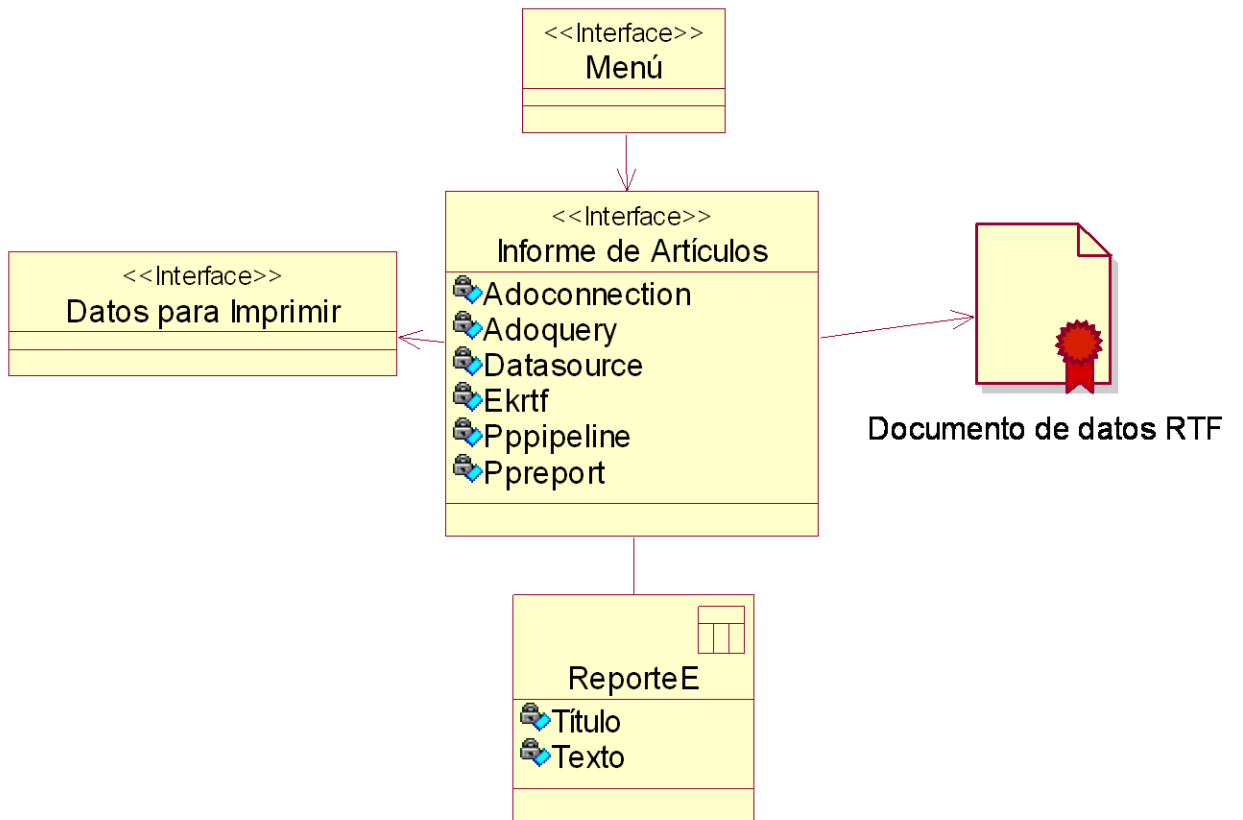


Figura 3.14 Diagrama de clases del diseño CU Generar Reporte de Artículos

3.3 – Diseño de la base de datos.

3.3.1 Modelo lógico de datos.

El diagrama del modelo lógico de datos o diagrama de clases persistentes, muestra las clases capaces de mantener su valor en el espacio y en el tiempo.

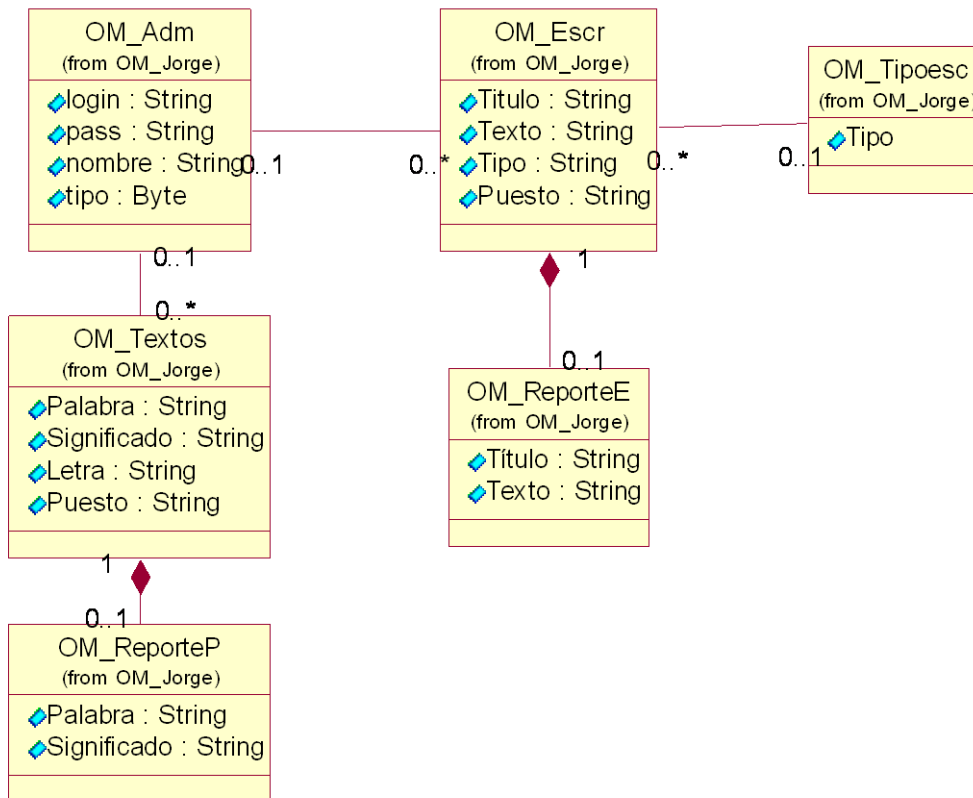


Figura 4.2 Modelo lógico de datos

3.3.2 Modelo físico de datos.

Cuando se define correctamente el modelo lógico, se hace mucho menos engorroso llegar al modelo de datos o modelo físico como también se le denomina en la metodología RUP de la siguiente forma: “el modelo de datos representa la estructura o descripción física de las tablas de la base de datos y es obtenido a partir del diagrama de clases persistentes”.

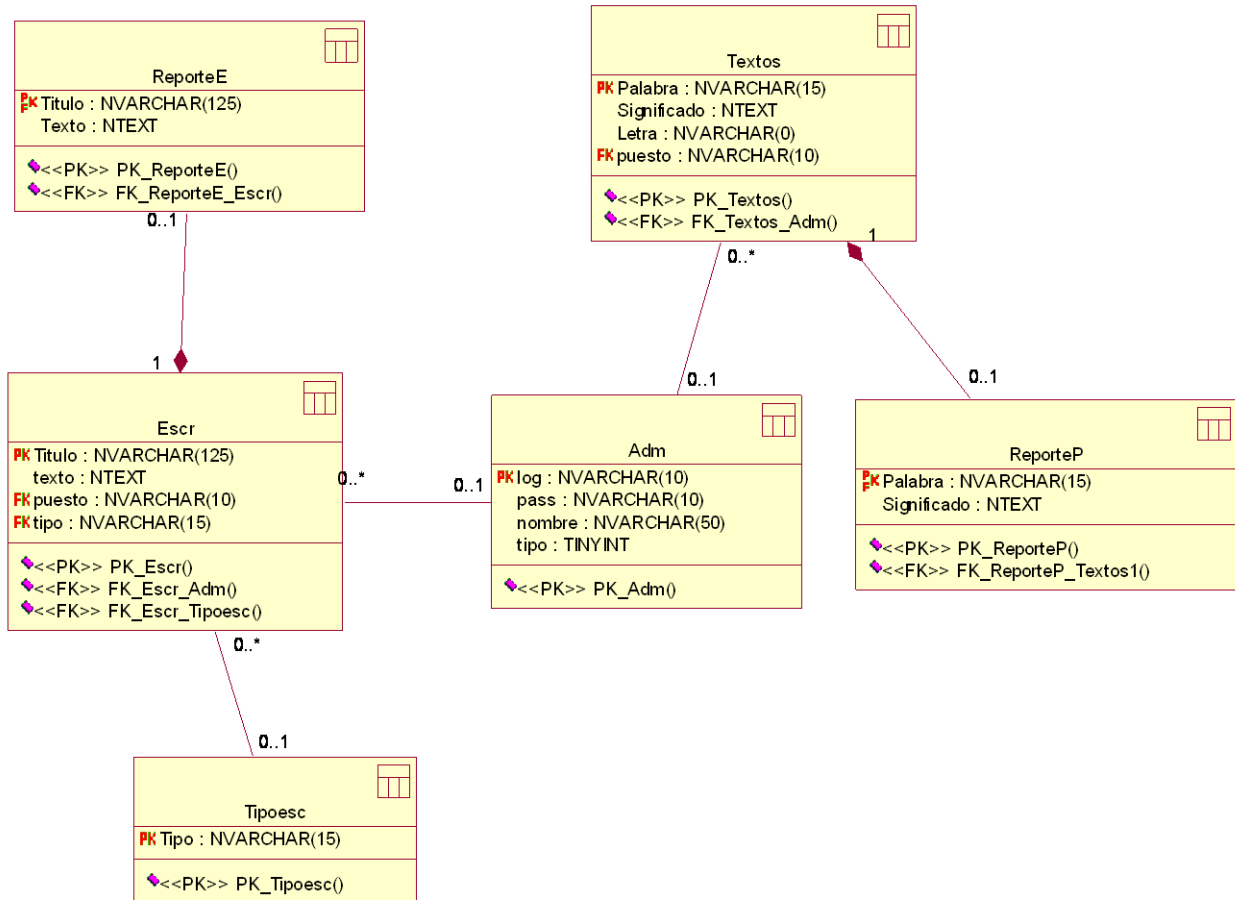


Figura 4.3 Modelo físico de datos

3.4 Diagrama de implementación.

El modelo de implementación denota la implementación del sistema en términos de componentes y subsistemas de implementación. Describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración, y modularización disponibles en el entorno de la implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen los componentes unos de otros

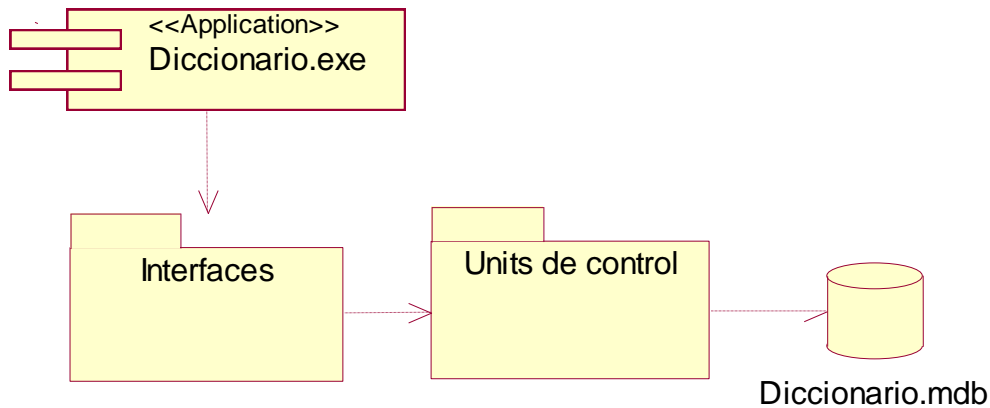


Figura 4.5 Diagrama de implementación.

Aquí presentamos una breve descripción de los componentes presentados:

Componente	Descripción
Interfaces	Este paquete representa las interfaces gráficas, o parte física, que presenta el diccionario para el trabajo con los datos, dándole respuesta a los requerimientos del capítulo 2; como ejemplo podemos mencionar Mostrar Palabras, Mostrar Artículos, Insertar Palabra, entre otros
Units de control	Este paquete representa la parte lógica del Diccionario, donde se encuentran todas las funciones y cálculos utilizados por el paquete Interfaces; como ejemplo podemos mencionar la unit Uoprs_palab, la cual está compuesta por las funciones necesarias para la gestión(Insertar, Actualizar, Eliminar) de las palabras
Diccionario.mdb	Representa la Base de Datos

3.5 Modelo de despliegue.

En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados. Es decir se sitúa el software en el hardware que lo contiene. Cada Hardware se representa como un nodo. En este se muestra el cliente, el cuál es el Diccionario Electrónico y su conexión con la base de datos, que en este caso es Diccionario.mdb

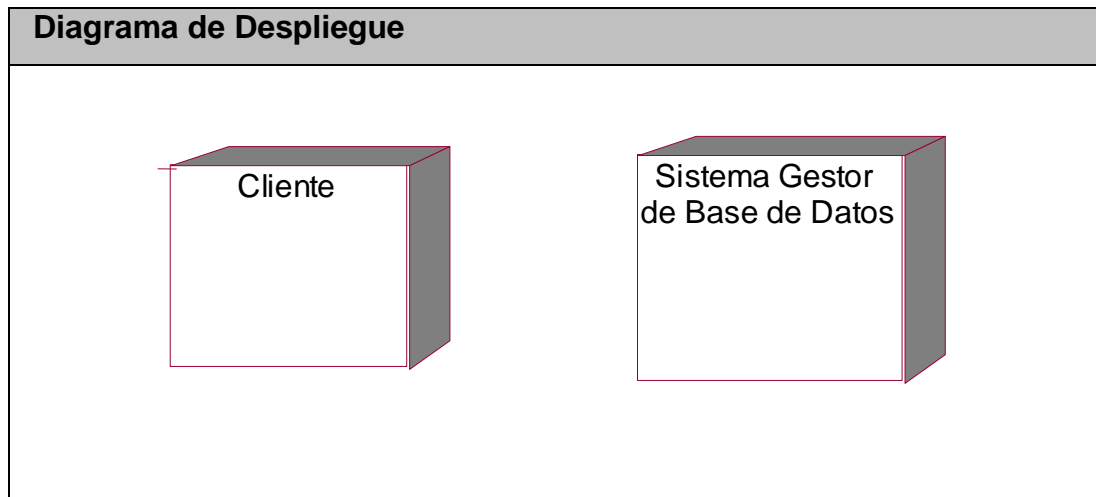
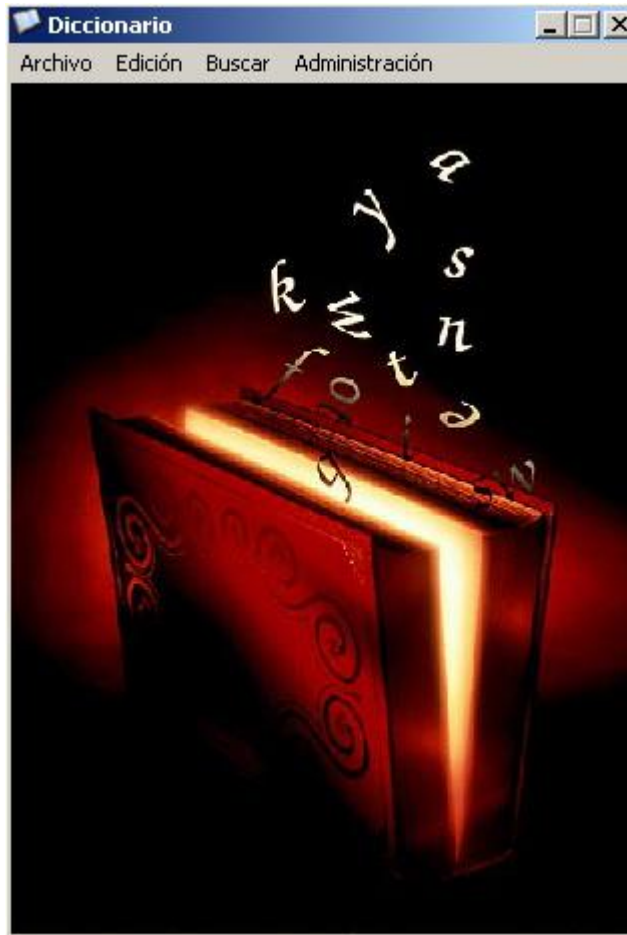


Figura. 4.6 Diagrama de despliegue.

3.6 Principios de diseño.

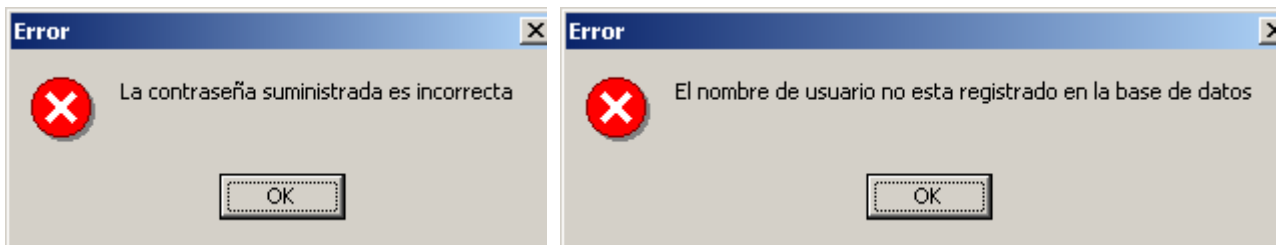
3.6.1 Estándares en la interfaz de la aplicación.

La interfaz diseñada para el sistema está basada en el estándar de ventanas de Windows. El tipo de letra a utilizar será Verdana de estilo regular y tamaño 9 y el diseño de la aplicación será conservador. La carga visual es adecuada y el lenguaje de las opciones que se ha utilizado es de fácil comprensión para el usuario. El sistema tiene un menú para el acceso a las funcionalidades que brinda el mismo. El icono asociado a la aplicación será el logotipo que identifica al sistema. En cuanto a los mensajes de error e informativos que se muestran son breves, pero informando siempre en que consisten.



3.6.2 Tratamiento de errores.

El diseño de la interfaz del sistema es amigable y está dirigido a evitar errores realizando siempre una validación de los datos que se procesan. Los mensajes de error que emite el sistema se muestran en un lenguaje de fácil comprensión para los usuarios, además el usuario debe confirmar las operaciones que impliquen riesgos, tal es el caso de eliminar un registro de la base de datos.





3.7 Estándares de codificación.

Para llevar a cabo el ejercicio de la buena ingeniería de software debe seguirse un buen estilo de código. Es necesario escribir código que sea fácil de entender y que reduzca el tiempo y esfuerzo a la hora de realizar alguna modificación al mismo.

Existen varios aspectos que pueden hacer un código más legible; algunos de estos son el empleo de nombres descriptivos y el uso de una indentación coherente, entre otros. Se describen a continuación algunas convenciones tomadas con relación a estos aspectos.

3.7.1 Convención de Nombres.

Los nombres de las variables, los controles, los procedimientos y funciones fueron adoptados lo más explicativos posibles, siempre respondiendo a su propósito.

Para cada tipo de control se tuvo en cuenta el uso de prefijos, un ejemplo de esto se muestra en la siguiente tabla:

Control	Prefijo	Ejemplo
Button	bt	btGuardar
Form	F	FMenu
Label	Lbl	LlbBA
LabeledEdit	LbE	LbEActP
ListBox	lb	lbCarreras
PageControl	pc	pcPalabras
ComboBox	Cb	CbAutor

Tabla 4.1 Nombres de algunos controles utilizados.

3.7.2 Indentación.

En el caso de la indentación, se establece que todas las líneas dentro de un procedimiento o función, estarán indentadas con respecto a la instrucción que encabeza a este, lo mismo ocurre con todas las líneas que conformen el cuerpo de una estructura repetitiva o condicional.

3.8 Concepción de la ayuda.

La ayuda está hecha con el propósito de informar y señalar todo con lo que el usuario va a trabajar en un momento determinado, por esto la misma se construyó para que se llamara con el botón F1 y que al salir mostrara solamente lo referente a lo que se estaba observando en un principio.

3.9 Conclusiones.

En este capítulo se definieron las clases del diseño y se representaron en el diagrama de clases del diseño. Se describió el diseño de la base de datos a través de los diagramas de clases persistentes y del modelo físico de datos y se representaron los componentes del sistema en el diagrama de implementación. Además, se describieron los principios del diseño seguidos en el sistema propuesto, profundizando específicamente en los estándares de la interfaz, la concepción general de la ayuda y el tratamiento de excepciones. Por último se explicó el estándar de codificación seguido en la implementación

CAPÍTULO 4. Estudio de Factibilidad.

4.1 Introducción.

Es necesario para la realización de un proyecto estimar el esfuerzo humano, el tiempo de desarrollo que se requiere para la ejecución del mismo y también su costo. Estas estimaciones pueden realizarse a través del método de puntos de función del modelo de COCOMO II.

En este capítulo se realizará el estudio de factibilidad del sistema utilizando el modelo de COCOMO II y se analizarán los costos y beneficios del mismo, así como su factibilidad.

4.2 Planificación.

Uno de los factores importantes a tener en cuenta en el diseño o mejoramiento de una aplicación informática, que permita apoyar la toma de decisiones, está relacionado con las ventajas del sistema propuesto que justifiquen o no su costo.

Los sistemas informáticos tienen como objetivo fundamental ofrecer la información adecuada en el momento que se solicite, pero si los ahorros que se obtienen con la información registrada y procesada, no compensan su costo, pueden no ser rentables. Sin embargo, la rentabilidad de un sistema de este tipo a veces resulta difícil de estimar, pues el valor de la información no es fácilmente cuantificable.

Entradas Externas:

Entrada externa	Ficheros	Elementos de Datos	Clasificación
Insertar Palabra	1	2	S
Actualizar Palabra	1	3	S
Eliminar Palabra	1	2	S
Insertar Artículo	2	4	S
Actualizar Artículo	1	3	S
Eliminar Artículo	1	1	S
Insertar Usuario	1	4	S
Actualizar Usuario	1	4	S
Eliminar Usuario	1	4	S
Insertar Palabras al Informe de Palabras	1	3	S
Eliminar Palabras del Informe de Palabras	1	1	S

Insertar Artículos al informe de Artículos	1	3	S
Eliminar Artículos del Informe de Artículos	1	1	S
Cambiar Contraseña	1	3	S
Actualizar Tipo de Artículo	2	4	S
TOTAL	Simple: 15, Media: 0, Compleja: 0.		

Tabla 4.1 Entradas Externas.

Salidas Externas:

Salida Externa	Ficheros	Elementos de datos	de Clasificación
Generar Reporte de Palabras	1	2	S
Generar Reporte de Artículos	1	3	S
Listado de Usuarios	1	3	S
TOTAL	Simple: 3, Media: 0, Compleja: 0.		

Tabla 4.2 Salidas Externas.

Peticiones:

Petición	Ficheros	Elementos de datos	Clasificación
Buscar Palabras	1	2	S
Buscar Artículos	1	2	S
Listar Todas las Palabras	1	2	S
Listar Palabras por Alfabeto	1	3	S
Listar Palabras por Autor	1	3	S
Listar Todos los Artículos	1	2	S
Listar Artículos por Autor	8	3	S
Listar Artículos por Tipo	8	2	S
TOTAL	Simple: 8, Media: 0, Compleja: 0.		

Tabla 4.3 Peticiones.

Ficheros Internos:

Fichero Interno	Records	Elementos de datos	Clasificación
Adm	1	4	S
Es cr	1	4	S

ReporteE	1	2	S
ReporteP	1	2	S
Textos	1	4	S
TipoEscr	1	1	S
TOTAL	Simple: 6, Media: 0, Compleja: 0.		

Tabla 4.4 Archivos Lógicos Internos.

Elementos	S	X Peso	M	X Peso	C	X Peso	PF. Subtotal
Archivos lógicos internos	6	(*7)	0	(*10)	0	(*15)	42
Entradas externas	15	(*3)	0	(*4)	0	(*6)	45
Salidas externas	3	(*4)	0	(*5)	0	(7)	12
Peticiones	8	(*3)	0	(*4)	0	(6)	24
Total							123

Tabla 4.6 Puntos de Función sin ajustar.

Estimación de la cantidad de instrucciones fuente (SLOC):

El cálculo de las instrucciones fuentes, según COCOMO II, se basa en la cantidad de instrucciones por punto de función que genera el lenguaje de programación empleado.

Características	Valor	
Puntos de función desajustados	123	
Lenguaje	Object Pascal	Access
% de utilización en la aplicación	80 % (98.4)	20 % (24,6)
Instrucciones fuentes por puntos de función	29	35
Instrucciones fuentes	2853.6	861
Total Instrucciones fuentes (LOC)	3714,2	

Tabla 4.7 Instrucciones fuentes.

4.3 Costos.

Multiplicadores de esfuerzo:

Multiplicador	Descripción	Valor
RCPX	Se considera que el producto es de complejidad media, con una base de datos de tamaño medio. La necesidad de documentación es básica pues la aplicación tiene un contenido sencillo donde solamente necesitará ayuda en algunos puntos	1
RUSE	Se implementa código reusable para el aprovechamiento de este en toda la aplicación.	1
PDIF	No tiene grandes restricciones en cuanto al tiempo de ejecución. EL Software no tiene limitación de memoria impuesta. La plataforma de aplicación es estable.	1
PERS	Teniendo en cuenta que el proyecto es desarrollado por estudiantes hay una rotación baja del personal. La capacidad de comunicación y colaboración de programadores y especialistas es alta. La disposición para el trabajo en equipo también es alta.	0.83
PREX	La experiencia en el trabajo con las plataformas y las herramientas de desarrollo es de más de tres años por lo que se considera alta.	0.70
FCIL	Se emplean herramientas de alto nivel para cada etapa del ciclo de vida del proyecto.	1
SCED	Los requerimientos de calendario de desarrollo son bajos.	1.14
		0.66

Tabla 4.8 Definición de los Multiplicadores de Esfuerzo (MEj).

Factores de Escala:

Factor	Descripción	Valor
PREC	El equipo de desarrollo posee una comprensión considerable de los objetivos del producto aunque no posee experiencia en la realización de software de este tipo. Aspectos novedosos.	3.72
FLEX	El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las especificaciones de interfaz externa.	3.04

TEAM	La cooperación es alta entre los miembros del equipo	1.14
RESL	La arquitectura es sólida y los riesgos generalmente se mitigan. Poca incertidumbre, riesgos no son críticos.	1.41
PMAT	Relación con el proceso de madurez del software. Nivel 3.	3.12
5 $\sum_{i=1}^5 SFi$		12.43

Tabla 4.9 Definición de los valores de los Factores de Escala (SFi).

7

$$EM = \prod_{i=1}^7 Emi = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED = 0.66$$

5

$$SF = \sum_{i=1}^5 SFi = PREC + FLEX + RESL + TEAM + PMAT = 12.43$$

Calculando el esfuerzo (PM)

El esfuerzo se representa mediante la fórmula:

$$PM = A * (MF)^E * EM$$

Donde:

$$A = 2.94; B = 0.91; C = 3.67; D = 0.24$$

$$E = B + 0.01 * \sum SFi = 0.91 + 0.01 * 12.43 = 1.03$$

$$F = D + 0.2 * (E - B) = 0.24 + 0.2 * (1.03 - 0.91) = 0.27$$

Cálculo del esfuerzo (PM):

$$PM = A * (MSLOC)^E * \prod_{i=1}^7 Emi = 2.94 * (3.7142)^{1.03} * 0.66 = 7.50 \approx \mathbf{8 \text{ Hombres/Mes}}$$

Cálculo del tiempo de desarrollo:

$$TDEV = C * PM^F = 3.67 * (7.50)^{0.27} = 6.32 \approx \mathbf{6 \text{ meses (Estimado)}}$$

Cálculo de la cantidad de hombres:

$$CH = PM / TDEV = 7.50 / 6.32 = 1.19 \approx 1 \text{ hombre}$$

Recalculando tiempo de desarrollo:

Como el equipo de trabajo está formado realmente por 1 persona, se recalcula el tiempo de desarrollo para la cantidad real de hombres.

$$CH^* = 1 \text{ hombres.}$$

$$TEDV = PM / CH^* = 7.50 / 1.19 = 6.31 \approx 6 \text{ meses.}$$

Cálculo del costo:

Asumiendo como salario promedio mensual (SP) \$225.00

$$CHM = CH * SP = 1 * \$ 225.00 = \$ 225.00$$

$$\text{Costo} = CHM * PM = \$ 225 * 6 = \$ 1350$$

Los costos en los que se incurriría de desarrollarse el sistema serían:

Cálculo de:	Valor
Esfuerzo(PM)	8 Hombres - mes
Tiempo de desarrollo(TDEV)	6 meses
Cantidad de hombres(CH)	1 hombre
Salario medio	\$ 225
Costo	\$ 1350

Tabla 4.10 Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

4.4 Beneficios tangibles e intangibles.

Los beneficios y los costos en todas la esferas de la vida son dos parámetros que viajan juntos. El llevar a cabo el proceso de desarrollo de un producto informático implica algún costo y por lo tanto está en dependencia de los beneficios que el mismo puede reportar. Los beneficios pueden ser de orden económico y social y ambos poseen la misma importancia a la hora de considerarlos.

Beneficios intangibles:

Los beneficios que se obtienen con el Diccionario Electrónico son de orden social e intelectual ya que con el uso del mismo se benefician los alumnos por la diversidad de la información almacenada y la facilidad de consulta de la misma para aumentar sus conocimientos sobre la Programación Lineal

4.5 Análisis de costos y beneficios.

El desarrollo de todo producto informático va asociado a un costo, el justificarlo depende de los beneficios tangibles e intangibles que trae consigo.

El apoyo que brinda el Diccionario Electrónico a la institución es la posibilidad de aumentar los conocimientos de Programación Lineal en estudiantes que cursan de Contabilidad y Finanzas.

Un elemento importante que no debe dejarse de considerar a la hora de determinar la factibilidad de este producto, independientemente de todos los beneficios que puedan venir aparejados al mismo, es el costo, el cual fue estimado en \$ 1350 en moneda nacional.

Para la realización de la aplicación no se incurrió en gastos adicionales de equipamiento, compra de algún otro sistema necesario, ni de herramientas de desarrollo y además no hubo necesidad de contratar personal calificado que realizara el trabajo imprescindible para obtener el producto final.

Cuando analizamos los costos apreciamos que son bajos, lo cual, junto a los grandes beneficios de la realización y utilización del sistema que se propone, determina la factibilidad del desarrollo del mismo.

4.6 Conclusiones.

Como método para la estimación se utilizó COCOMO II. El mismo está implementado a través de ecuaciones matemáticas que permiten el cálculo de esfuerzo a partir de valores estimados como son líneas de código fuente y análisis de puntos de función.

Una vez terminado el estudio de factibilidad del sistema, se estima un tiempo de 6 meses para su construcción por cuatro hombres y su costo asciende a \$ 1350.

Este estudio de planificación y factibilidad proporcionó valiosos argumentos que permitieron llegar a la conclusión de que la solución de software, es factible y que la herramienta propuesta trae consigo una serie de beneficios sobre todo intangibles para la organización, pero no menos necesarios e importantes, ya que con su uso va a apoyar el proceso enseñanza-aprendizaje dentro de la misma en los alumnos que la utilicen.

Conclusiones

Partiendo de los objetivos planteados para la realización del presente trabajo se llega a las siguientes conclusiones:

1. Se logró realizar un estudio y reconocimiento de los fundamentos teóricos metodológicos que influyen en el proceso de enseñanza-aprendizaje de la Programación Lineal.
2. Se analizaron las tendencias, metodologías y tecnologías actuales del campo de la Informática, determinándose cuáles emplear en el proceso de diseño del sistema propuesto.
3. Se implementó una aplicación que permite gestionar toda la información almacenada por el sistema y generar informes partiendo de esta información, utilizando los elementos y características definidas en el proceso de diseño.
4. Se determinó la factibilidad del Diccionario Electrónico como apoyo al proceso de enseñanza-aprendizaje de la Programación Lineal en la UNISS.

Recomendaciones.

Poner en práctica las funcionalidades que brinda el sistema durante un período de tiempo que permita garantizar la comprobación de forma práctica del correcto funcionamiento de las mismas para localizar durante el período de pruebas, las insatisfacciones o especificidades que pudieran aparecer.

Ampliar el sistema para insertar imágenes

Hacer pruebas en red para poder determinar la mejor forma de utilizarlo por parte de los profesores, ya sea en una máquina aislada o compartida, en busca de evitar la dualidad del software y su contenido.

Bibliografía.

Alfonso Sánchez, I. (2003). *Elementos conceptuales básicos del proceso de enseñanza-aprendizaje*. Recuperado el 31 de marzo de 2010, de http://www.bvs.sld.cu/revistas/aci/vol11_6_03/aci17603.htm

Alonso Pinedo, A., & Hernández Mora, F. (2006). Trabajo de diploma para optar por el título de Ingeniería en Informática. *Sistema Gestor de Encuestas* , 33-34.

Bravo López, G., & Cáceres Mesa, M. (s.f.). *El proceso de enseñanza-aprendizaje desde una perspectiva comunicativa*. Recuperado el 31 de marzo de 2010, de <http://www.rieoei.org/deloslectores/1289Bravo.pdf>

Brunet González, C. (2007). Trabajo de diploma para optar por el título de Ingeniería en Informática. *GENEX Sistema generador de exámenes para las asignaturas de las carreras de Universalización* , 15,17,20,22.

Caraballosa Rafael, Y. (2007). Tesis en opción al título académico de Master en “Nuevas tecnologías para la educación“. *Tutorial para la enseñanza-aprendizaje de los contenidos de SQL Server en las universidades pedagógicas de Sancti Spíritus* .

Cardoza Gómez, R., & Guerrero Seide, E. (s.f.). *Una metodología para el análisis de la solución óptima un modelo de Programación Lineal*. Recuperado el 12 de marzo de 2010, de <http://www.monografias.com>

Edel Navarro, R. (2004). *El concepto de enseñanza aprendizaje*. Recuperado el 15 de abril de 2010, de redcientifica: <http://www.redcientifica.com/doc/doc200402170600.html>

el lenguaje de programación java.pdf.

Fardales Pérez, J. (mayo de 2007). Tesis presentada en opción al título académico de Máster en Nuevas Tecnologías para la Educación. *Diseño e implementación de un diccionario electrónico para la enseñanza de la Programación Lógica en estudiantes de 4to año de Lic. Ciencias de la Computación*.

Grau, R., Correa, C., & Rojas, M. (2004). *Metodología de la Investigación*. El POIRA Editores S. A.

Jacobson, I. (2004). *El proceso Unificado de Desarrollo de Software*. La Habana: Ed. Félix Varela.

Labarrere, A. F. (1988). *Bases psicopedagógicas de la enseñanza de la solución de problemas matemáticos en la escuela cubana*. La Habana: Pueblo y Educación.

Larman, C. (2004). *UML y Patrones Introducción al diseño orientado a objetos*. La Habana: Ed. Félix Varela.

Las TICs. (s.f.). Recuperado el 21 de enero de 2010 , de <http://www.didactika.com/docentes/files/TICs.ppt>

Madruga García, D., & Morera Hernández, A. (2006). Trabajo de diploma para optar por el título de Ingeniería en informática. *Automatización del proceso de planificación y control del plan de actividades mensual* , 25.

Martínez Riverol, A. (2007). Tesis en opción al título de master en ciencias en Nuevas Tecnologías para la Educación. *Simulador para el aprendizaje del backtracking* , 14.

Peraza Bello, D., & Muñoz Capote, A. (2006). Trabajo de Diploma para optar por el título de Ingeniero Infomático. *SECDC Sistema de Evaluación de Cualidades Dinámicas y de Consumo* , 15-16.

Pérez Medinilla, Y. T. (diciembre de 2004). Tesis en opción al título de Máster en Ciencias de la Educación. *Estrategia de enseñanza aprendizaje para la elaboración de un software educativo de modelación de problemas de Programación Lineal.*

Ramón Gómez, J. (2004). *TIC y educación.* Recuperado el 3 de febrero de 2010, de <http://boj.pntic.mec.es/jgomez46/ticedu.htm>

Sosa López, D., & Hector Ortiz, K. (2006). Trabajo para optar por el título de Ingeniero en Informática. *Desarrollo de un prototipo de intranet para una Facultad de un Centro de Educación Superior: Modulo Docente* , 15-16.

UNESCO. (2003). Proyecto general de educación para América Latina y el Caribe. *Pedagogía 2003.* La Habana.

Victoria González, A. E. (2007). Tesis en Opción al Título Académico de Máster en Nuevas Tecnologías para la Educación. *Diseño de la Biblioteca Virtual para el Joven Club de Computación y Electrónica Central* .

Anexos.

Anexos A: Interfaces relacionadas con los Requerimientos del Sistema

A.1 Interfaces del CU Gestionar Palabra


Insertar Palabra [X]


Insertar Palabra

Palabra

Significado

Opciones

 Insertar Palabra

 Cerrar

Insertar palabra

Actualizar Palabra [X]

Seleccione los datos

Letra:

Palabra:


Área de Actualización


Palabra a actualizar:

Significado a actualizar:

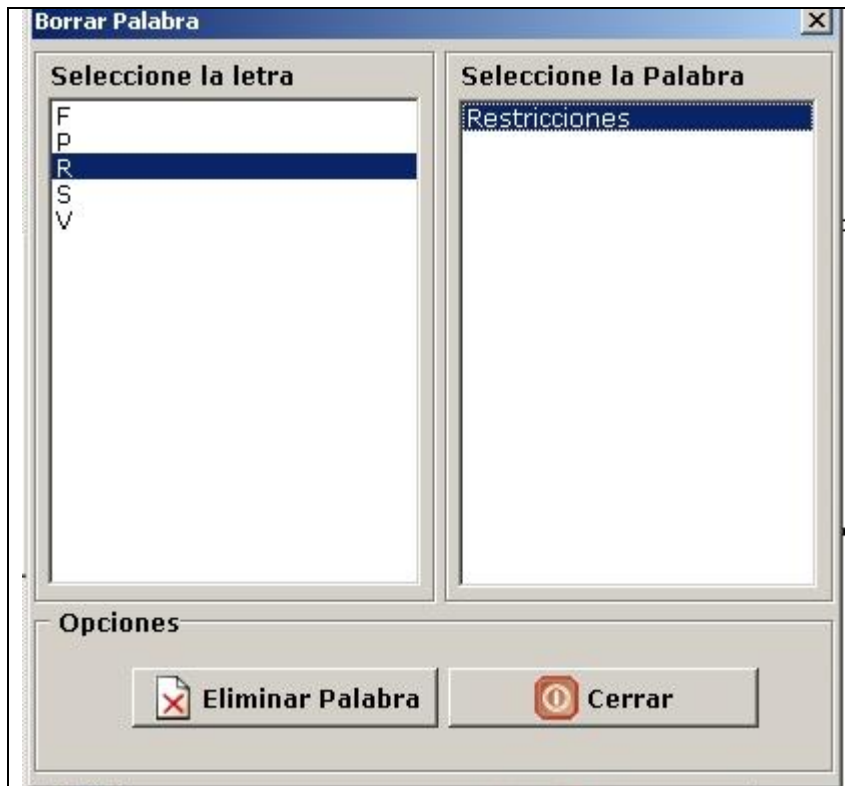
Es una expresión matemática lineal que representa el objetivo del problema. Es la expresión que tendremos que maximizar o minimizar, según el problema a resolver y las variables que hayamos usado en la solución del mismo. Función Objetivo: (Max. ó Min.) $Z = C_1X_1 + C_2X_2 + C_3X_3 + \dots + C_nX_n$

Opciones

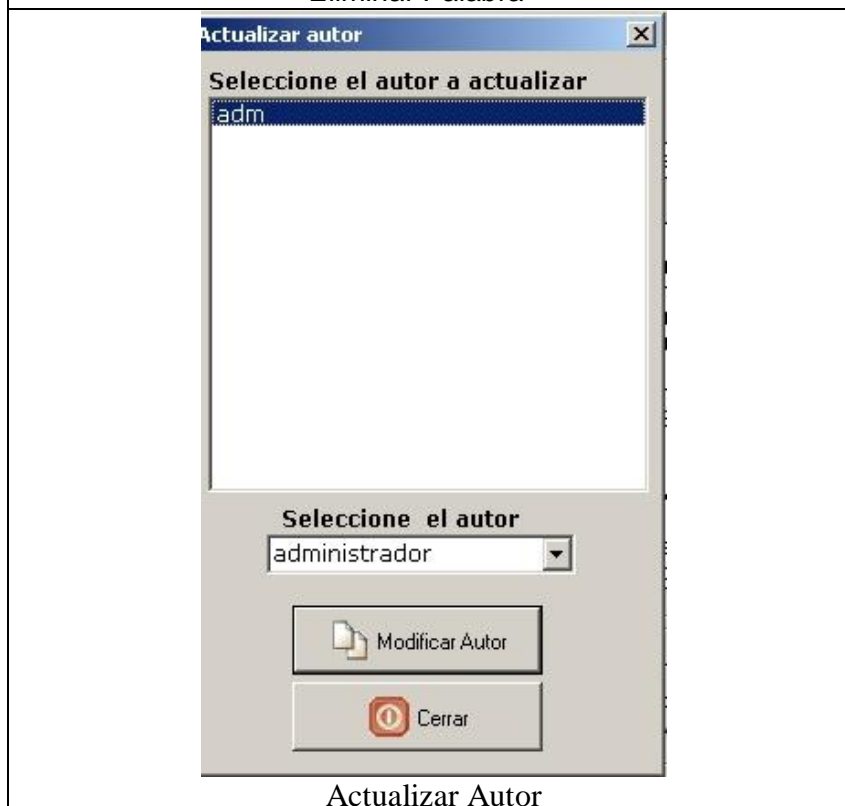
 Modificar Texto

 Cerrar

Actualizar palabra



Eliminar Palabra



Actualizar Autor

A.2 Interfaces del CU Gestionar Usuario

Insertar Usuario

Datos a Insertar

Nombre

Login

Contraseña

Repetir Contraseña

Tipo de Usuario
Especialista

Usuarios registrados
adm
administrador
profesor

Opciones

Insertar usuario

Lista de Usuario

Login	Contraseña	No...
administrador		Jorg...
profesor		Aur...
adm		Jorg...

Opciones

Lista de Usuarios

Actualizar Usuario

Usuarios Registrados	Datos Seleccionados
adm	Login <input type="text" value="adm"/> Nombre <input type="text" value="Jorge Br"/> Tipo <input type="text" value="Administrador"/>
Opciones <input type="button" value="Modificar Usuario"/> <input type="button" value="Cerrar"/>	

Actualizar Usuario

Cambiar Contraseña

Login actual

Contraseña actual

Nueva contraseña

Confirmar contraseña

Cambiar Contraseña

Eliminar Usuario

Usuarios Registrados	Datos Seleccionados
administrador profesor	Login <input type="text" value="administrador"/> Nombre <input type="text" value="Jorge brunet g"/> Tipo <input type="text" value="Administrador"/>
Opciones <input type="button" value="Eliminar Usuario"/> <input type="button" value="Cerrar"/>	

Eliminar Usuario

A.3 Interfaces del CU Gestionar Artículo

The 'Insertar Artículo' dialog box is titled 'Insertar Artículo' and contains the following elements:

- Escrito**: A section header.
- Título**: A text input field.
- Tipos**: Two radio buttons, 'Tipos Existentes' (selected) and 'Agregar Tipo'.
- Tipos**: A dropdown menu and a text input field.
- Texto**: A large text area for entering the article content.
- Buttons**: 'Insertar Artículo' (with a document icon) and 'Cerrar' (with a red stop icon).

Insertar Artículo

The 'Actualizar Artículo' dialog box is titled 'Actualizar Artículo' and contains the following elements:

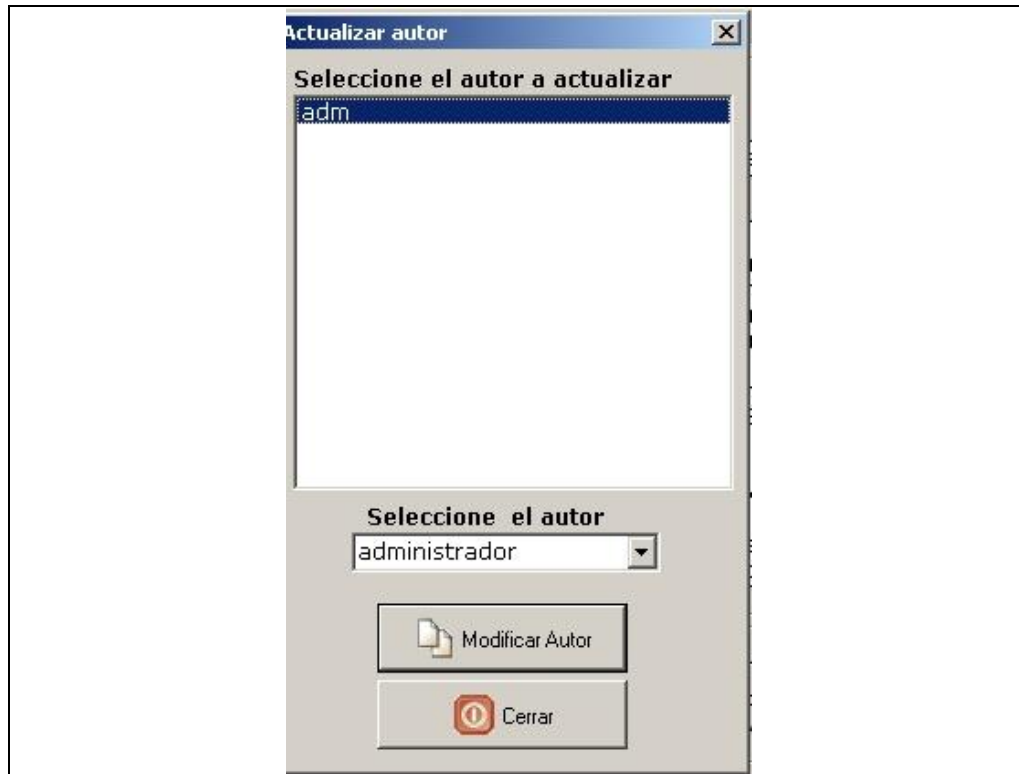
- Seleccionar y Actualizar**: A section header.
- Títulos**: A list of titles, with 'Aspectos de la Programación Lineal 1' selected.
- Título a actualizar**: A text input field containing 'Aspectos de la Programación Lineal 1'.
- Texto**: A text area containing the article content:

En los siglos XVII y XVIII, grandes matemáticos como Newton, Leibnitz, Bernouilli y, sobre todo, Lagrange, que tanto habían contribuido al desarrollo del cálculo infinitesimal, se ocuparon de obtener máximos y mínimos condicionados de determinadas funciones.

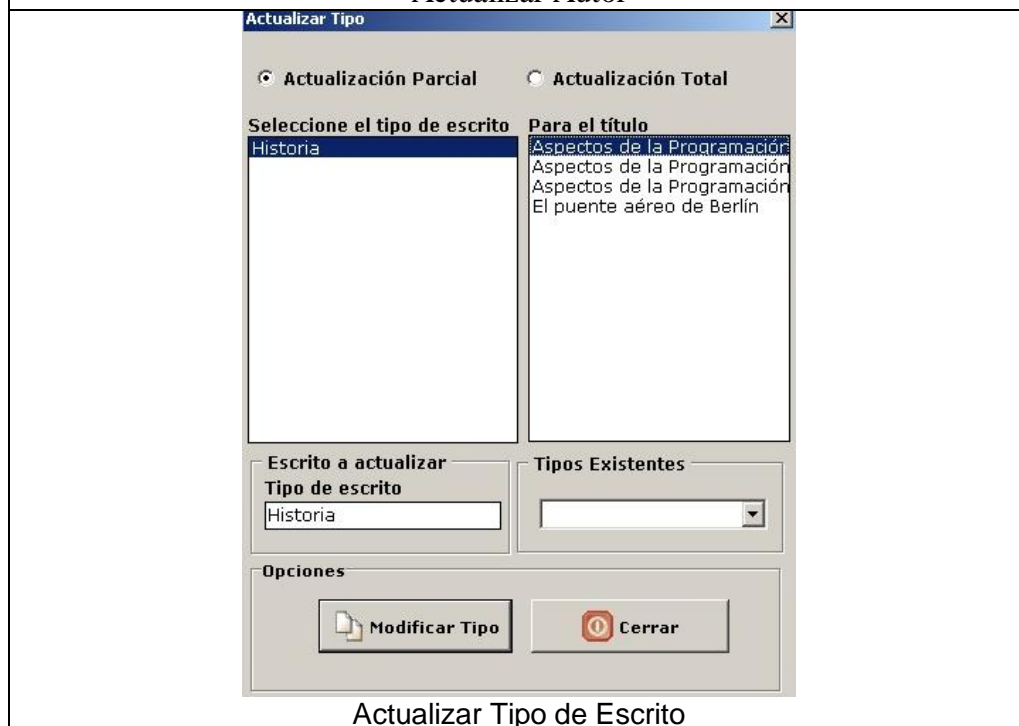
Posteriormente el matemático francés Jean Baptiste-Joseph Fourier (1768-1830) fue el primero en intuir, aunque de forma imprecisa, los métodos de lo que actualmente llamamos programación lineal y la potencialidad que de ellos se deriva.

Si exceptuamos al matemático Gaspar Monge (1746-1818), quien en 1776 se interesó por problemas
- Opciones**: A section header.
- Buttons**: 'Modificar Texto', 'Modificar Tipo', 'Modificar Autor', and 'Cerrar' (with a red stop icon).

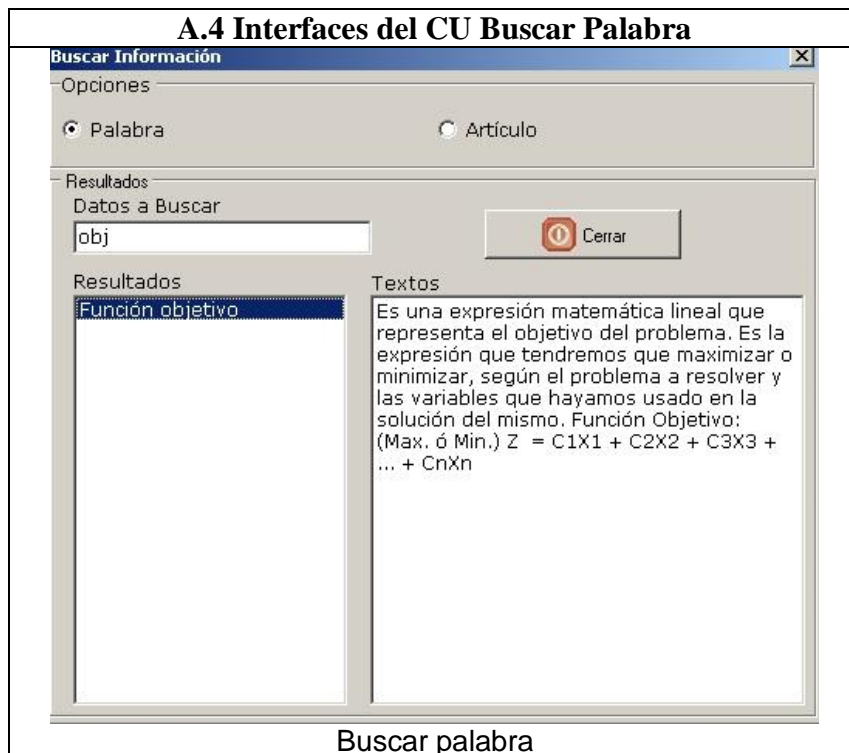
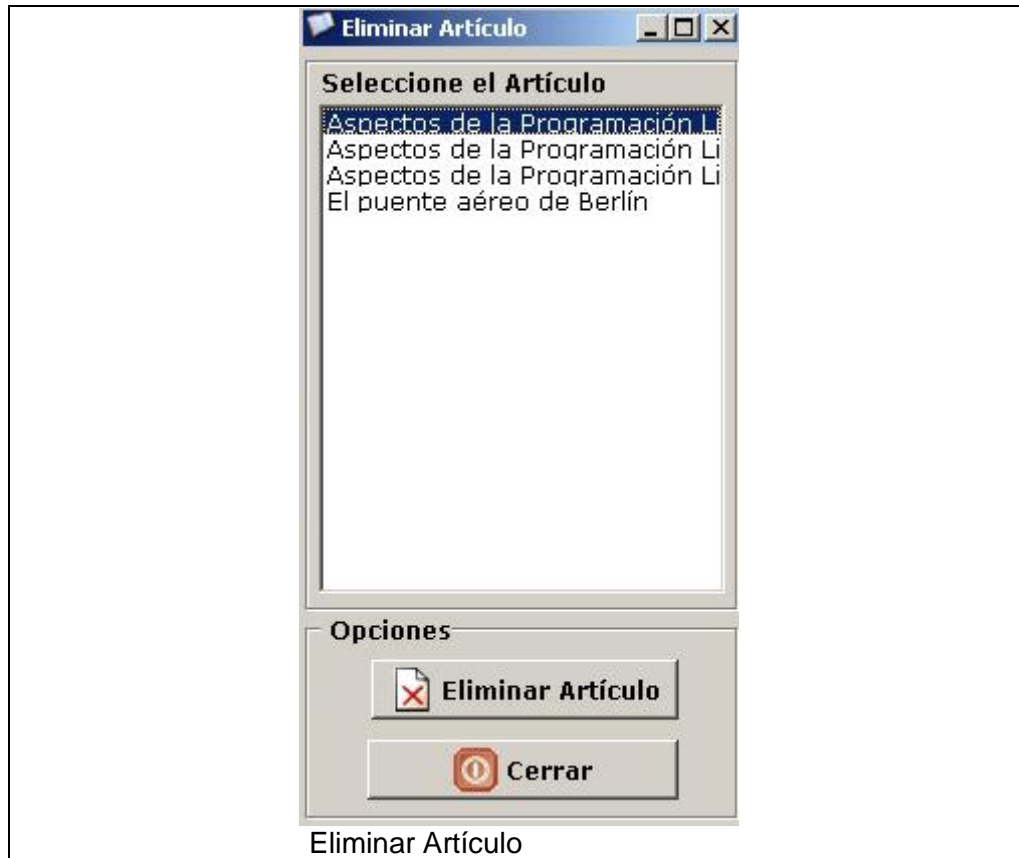
Actualizar Artículo



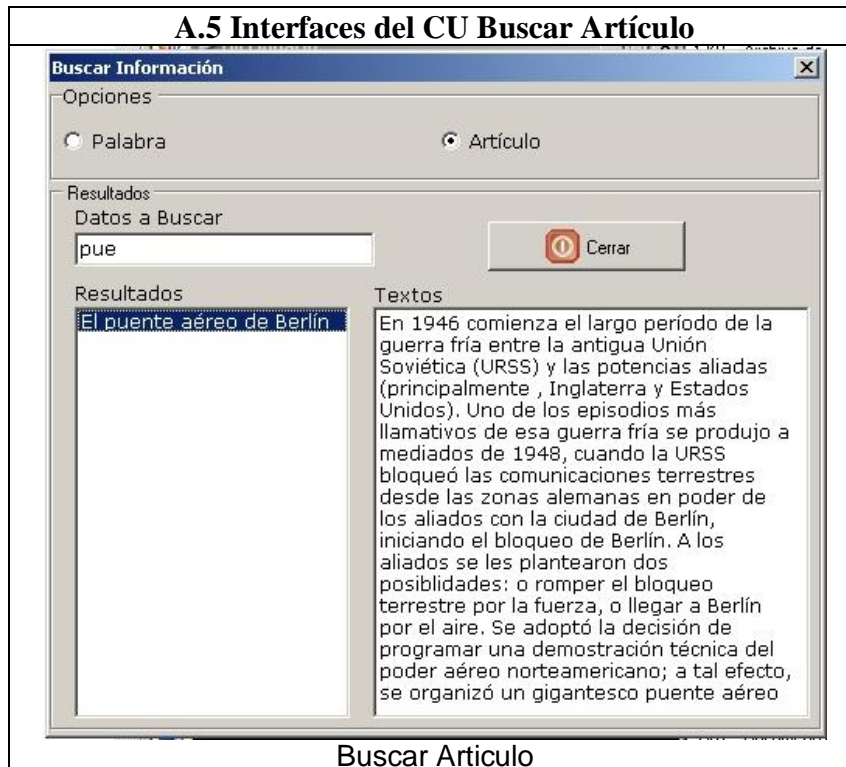
Actualizar Autor



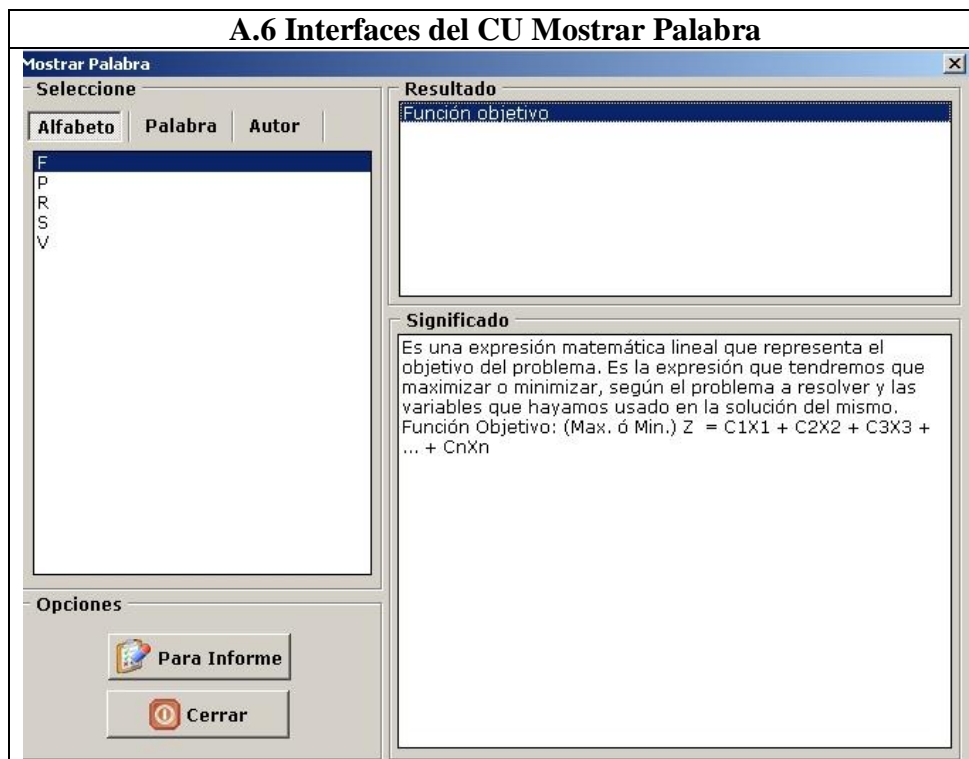
Actualizar Tipo de Escrito

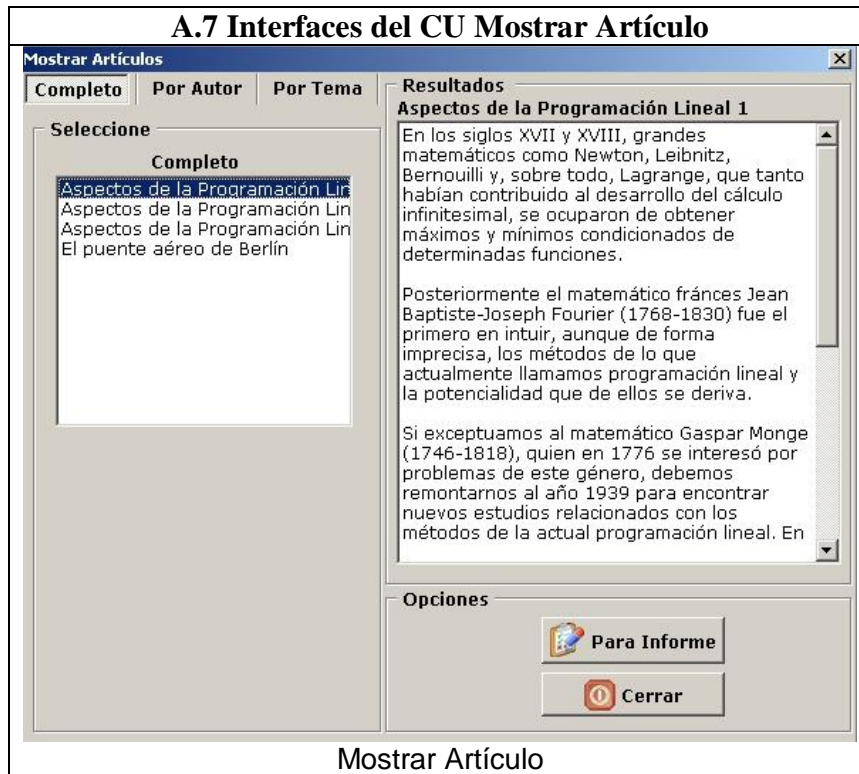


A.5 Interfaces del CU Buscar Artículo



A.6 Interfaces del CU Mostrar Palabra





A.9 Interfaces del CU Cerrar Sesión



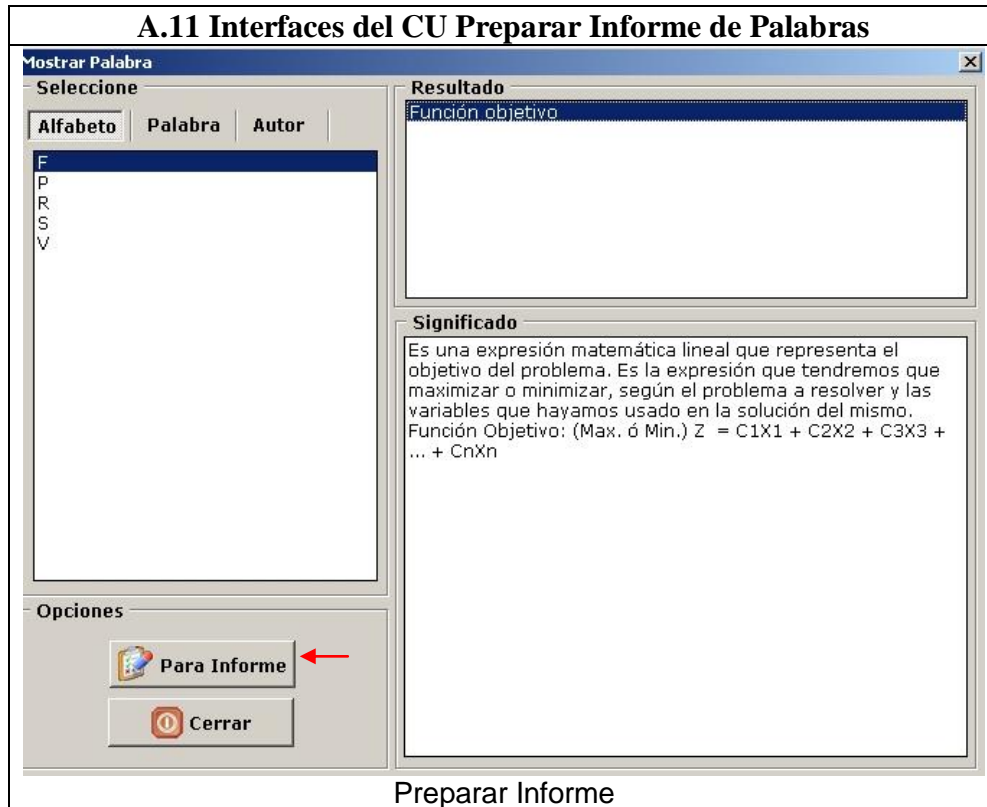
Cerrar Sesión

A.10 Interfaces del CU Cambiar Contraseña

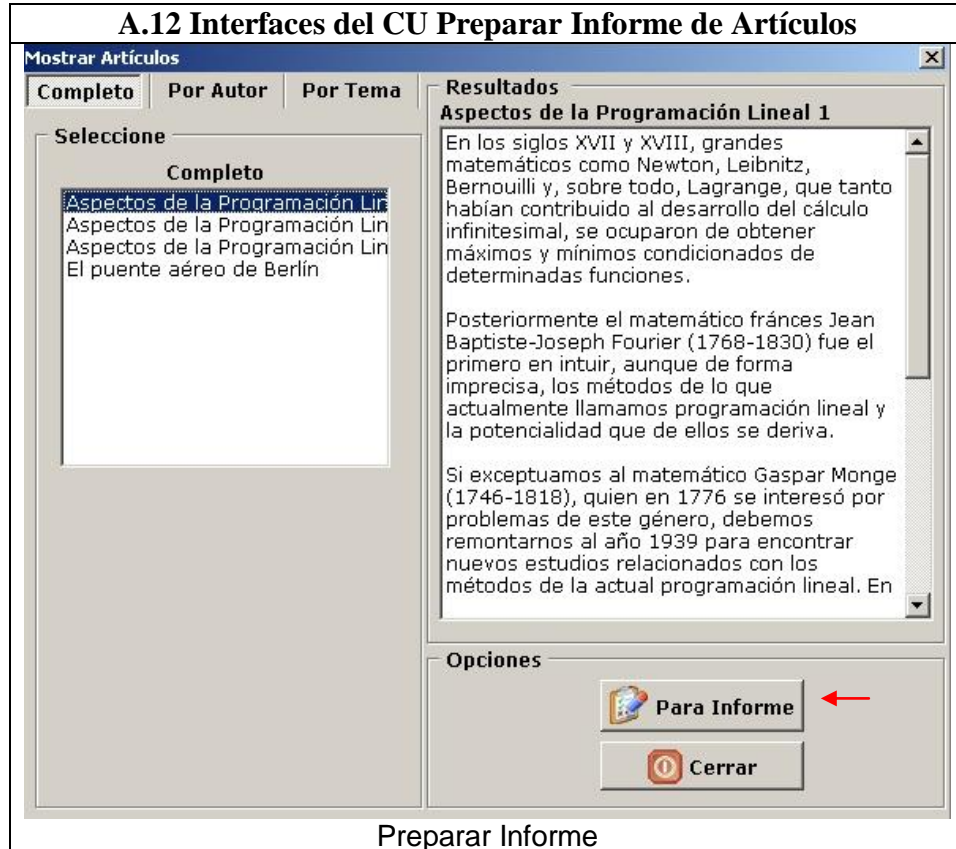
A screenshot of a dialog box titled "Cambiar Contraseña". It contains four input fields: "Login actual" with the value "adm", "Contraseña actual", "Nueva contraseña", and "Confirmar contraseña". At the bottom, there are two buttons: "Modificar Contraseña" (with a document icon) and "Cerrar" (with a red stop icon).

Cambiar Contraseña

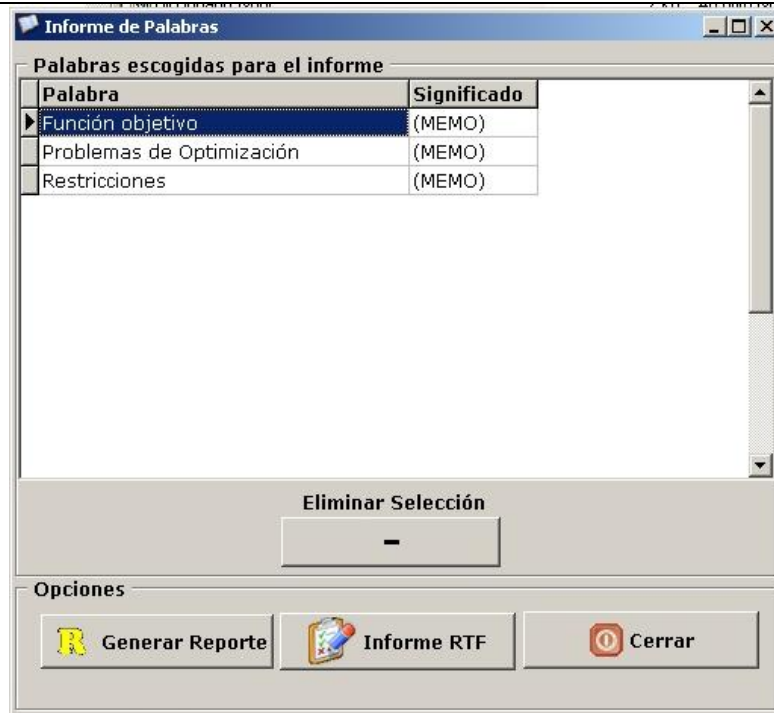
A.11 Interfaces del CU Preparar Informe de Palabras



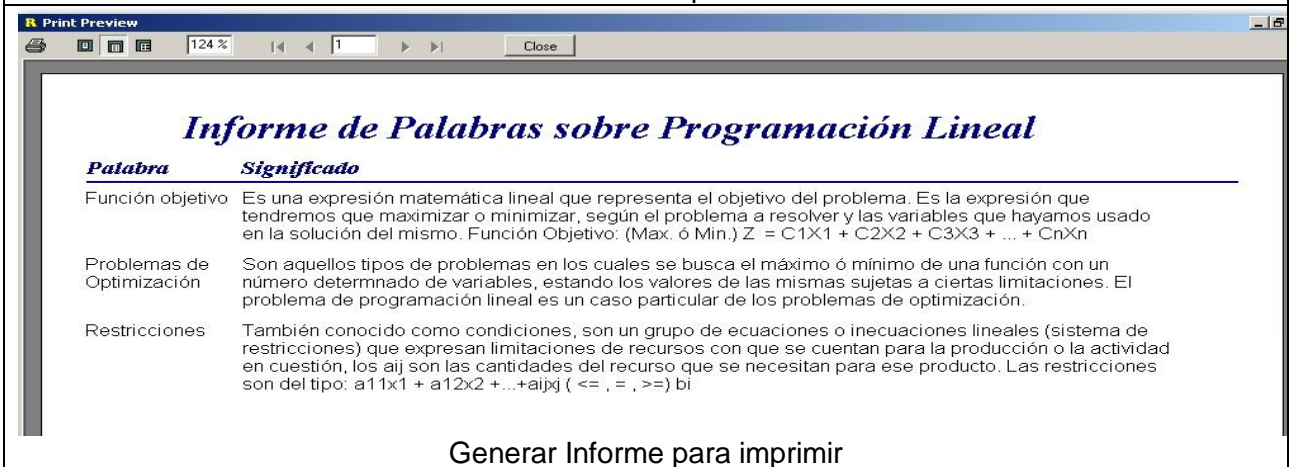
A.12 Interfaces del CU Preparar Informe de Artículos



A.13 Interfaces del CU Generar Informe de Palabras



Selección de opciones



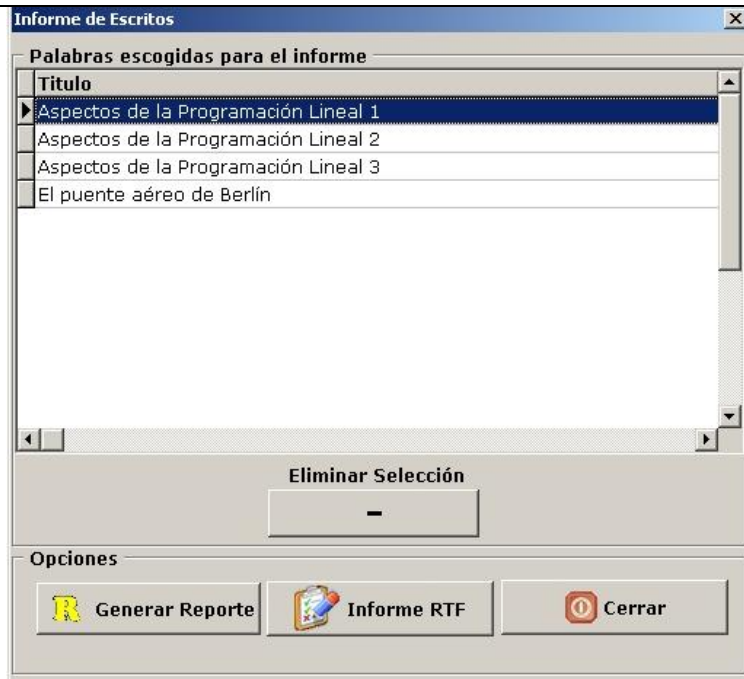
Generar Informe para imprimir

INFORME DE PALABRAS

PALABRA(S)	SIGNIFICADO(S)
Función objetivo	Es una expresión matemática lineal que representa el objetivo del problema. Es la expresión que tendremos que maximizar o minimizar, según el problema a resolver y las variables que hayamos usado en la solución del mismo. Función Objetivo: (Max. ó Min.) $Z = C1X1 + C2X2 + C3X3 + \dots + CnXn$
Problemas de Optimización	Son aquellos tipos de problemas en los cuales se busca el máximo ó mínimo de una función con un número determinado de variables, estando los valores de las mismas sujetas a ciertas limitaciones. El problema de programación lineal es un caso particular de los problemas de optimización.
Restricciones	También conocido como condiciones, son un grupo de ecuaciones o inecuaciones lineales (sistema de restricciones) que expresan limitaciones de recursos con que se cuentan para la producción o la actividad en cuestión, los a_{ij} son las cantidades del recurso que se necesitan para ese producto. Las restricciones son del tipo: $a_{11}x_1 + a_{12}x_2 + \dots + a_{1j}x_j (\leq , = , \geq) b_i$

Generar Informe en formato RTF

A.14 Interfaces del CU Generar Informe de Artículos



Selección de opciones

INFORME DE ESCRITOS

Título(s)	Texto(s)
Aspectos de la Programación Lineal 1	<p>En los siglos XVII y XVIII, grandes matemáticos como Newton, Leibnitz, Bernouilli y, sobre todo, Lagrange, que tanto habían contribuido al desarrollo del cálculo infinitesimal, se ocuparon de obtener máximos y mínimos condicionados de determinadas funciones.</p> <p>Posteriormente el matemático francés Jean Baptiste-Joseph Fourier (1768-1830) fue el primero en intuir, aunque de forma imprecisa, los métodos de lo que actualmente llamamos programación lineal y la potencialidad que de ellos se deriva.</p> <p>Si exceptuamos al matemático Gaspar Monge (1746-1818), quien en 1776 se interesó por problemas de este género, debemos remontarnos al año 1939 para encontrar nuevos estudios relacionados con los métodos de la actual programación lineal. En este año, el matemático ruso Leonodas Vitallyevich Kantarovitch publica una extensa monografía titulada Métodos matemáticos de organización y planificación de la producción en la que por primera vez se hace corresponder a una extensa gama de problemas una teoría matemática precisa y bien definida llamada, hoy en día, programación lineal .</p> <p>En 1941-1942 se formula por primera vez el problema de transporte, estudiado independientemente por Koopmans y Kantarovitch, razón</p>

Generar Informe para imprimir

Print Preview

124% | 1 | Close

Informe de Artículos sobre Programación Lineal

Título	Texto
Aspectos de la Programación Lineal 1	<p>En los siglos XVII y XVIII, grandes matemáticos como Newton, Leibnitz, Bernouilli y, sobre todo, Lagrange, que tanto habían contribuido al desarrollo del cálculo infinitesimal, se ocuparon de obtener máximos y mínimos condicionados de determinadas funciones.</p> <p>Posteriormente el matemático francés Jean Baptiste-Joseph Fourier (1768-1830) fue el primero en intuir, aunque de forma imprecisa, los métodos de lo que actualmente llamamos programación lineal y la potencialidad que de ellos se deriva.</p> <p>Si exceptuamos al matemático Gaspar Monge (1746-1818), quien en 1776 se interesó por problemas de este género, debemos remontarnos al año 1939 para encontrar nuevos estudios relacionados con los métodos de la actual programación lineal. En este año, el matemático ruso Leonodas Vitallyevich Kantarovitch publica una extensa monografía titulada Métodos matemáticos de organización y planificación de la producción en la que por primera vez se hace corresponder a una extensa gama de problemas una teoría matemática precisa y bien definida llamada, hoy en día, programación lineal .</p> <p>En 1941-1942 se formula por primera vez el problema de transporte, estudiado independientemente por Koopmans y Kantarovitch, razón por la cual se suele conocer con el nombre de problema de Koopmans-Kantarovitch.</p> <p>Tres años más tarde, G. Stigler plantea otro problema particular conocido con el nombre de régimen alimenticio optimal.</p>

Generar Informe en formato RTF

Anexos B: Elementos relacionados a los Diagramas de Clases del Diseño

B1 Clases principales de control utilizadas

Clase UOprs_usr	
Maneja las operaciones sobre los usuarios	
Método	Propósito
CrearCuenta()	Insertar usuario al sistema
Autenticar()	Validar el tipo de usuario registrado
Editar ()	Editar los datos de los usuarios
Editarpass()	Editar la contraseña del usuario
Eliminar()	Eliminar un usuario

Clase UBasedatos	
Maneja la conexión y la operación con la base de datos	
Elementos	Propósito
Adodataset()	Estos 3 elementos tienen el propósito de lograr la conexión con la base de datos (Adoconnection) y de operar sobre la misma con las opciones de edición o listado de datos según se necesite (Adodataset, Adocommand)
Adoconnection()	
Adocommand ()	

Clase UOprs_pala	
Maneja las operaciones sobre las palabras	
Método	Propósito
InsertarP()	Insertar palabras al sistema
EditarP()	Editar la palabra seleccionada
EditarParP ()	Editar solamente el texto de la palabra seleccionada
EliminarP()	Eliminar una palabra
ReporteP()	Insertar la palabra en el informe de palabras
ReportePB()	Eliminar todas las palabras del informe

Clase UOprs_Escr	
Maneja las operaciones sobre los artículos	
Método	Propósito
InsertarE()	Insertar artículos al sistema
EditarE()	Editar el artículo seleccionado
EditarParE ()	Editar solamente el texto del artículo seleccionado
EliminarE()	Eliminar un artículo
InsertarT()	Insertar un tipo de artículo
EliminarT()	Eliminar un tipo de artículo
EditarTE()	Editar el tipo para un artículo
EditarTETotal()	Editar el tipo para todos los artículos
ReporteE()	Insertar la palabra en el informe de palabras

ReportePE()	Eliminar todas las palabras del informe
Clase UOprs_otro	
Maneja las operaciones de listado y validación de los datos	
Método	Propósito
Lista_usr()	Listado de los usuarios del sistema
Lista_palab()	Listado de las palabras disponibles
Lista_palabautor ()	Listado de las personas que han puesto palabras
Lista_escr()	Listado de los artículos disponibles
Lista_letra()	Listado de las letras del alfabeto disponibles según las palabras guardadas
Lista_tipo()	Listado de todos los tipos de artículos disponibles
Lista_esctipo()	Listado de todos los tipos de artículos asignados a un texto
Lista_escautor()	Listado de las personas que han puesto artículos
ConteoP()	Revisa si existen palabras para el informe
ConteoE()	Revisa si existen artículos para el informe
Lista_palabl()	Listado de las palabras según la letra con las que comienzan
Lista_escrit()	Listado de los artículos según el tipo al que pertenecen
Lista_escrita()	Listado de los artículos según quien los puso
The_usr()	Lista todos los datos de un usuario
Lista_palabA()	Listado de las palabras disponibles ordenadas según quien las puso
Validar_usuario()	Comprueba que no exista el usuario en el sistema para evitar duplicados
Validar_usrpass()	Comprueba que exista el usuario y que le pertenezca la contraseña
Validar_palabra()	Comprueba que no exista la palabra en el sistema para evitar duplicados
Validar_palabraE()	Comprueba que no existan duplicados en el informe de palabras
Validar_escrito()	Comprueba que no exista el artículo en el sistema para evitar duplicados
Validar_escritoE()	Comprueba que no existan duplicados en el informe de artículos
Validar_tipoEscr()	Comprueba que no exista el tipo de artículo en el sistema para evitar duplicados
Validar_pass()	Selecciona la contraseña de un usuario
Contar_Adm()	Cuenta los administradores del sistema
Mostrar_palabra()	Selecciona la palabra y su significado que va a ser mostrada
Mostrar_escrito()	Selecciona el artículo que va a ser mostrado

B2 Descripción de las tablas de la base de datos		
Nombre: Adm		
Descripción: Almacena los datos de los usuarios registrados.		
Atributo	Tipo	Descripción
log	texto	Identificador del usuario
pass	texto	Contraseña
nombre	texto	Nombre del usuario
tipo	número	Tipo de usuario (profesor o administrador)

Nombre: Escr		
Descripción: Almacena los artículos guardados por el usuario.		
Atributo	Tipo	Descripción
título	texto	Nombre del artículo.
texto	memo	Contenido del artículo
puesto	texto	Identifica al usuario que puso el texto
tipo	texto	Identifica el tipo de artículo
Nombre: Textos		
Descripción: Almacena las palabras puestas por el usuario.		
Atributo	Tipo	Descripción
Palabra	texto	Título de la palabra
Significado	memo	Contenido de la palabra
Letra	texto	Letra con que comienza la palabra
Puesto	texto	Identifica al usuario que puso la palabra
Nombre: TipoEsc		
Descripción: Almacena los tipos de artículo que se pueden definir.		
Atributo	Tipo	Descripción
Tipo	texto	Tipos de artículos que se pueden definir
Nombre: ReporteP		
Descripción: Almacena las palabras seleccionadas para confeccionar el informe.		
Atributo	Tipo	Descripción
Palabra	Similar a la tabla	Similar a la tabla Textos
Significado	Textos	
Nombre: ReporteE		
Descripción: Almacena los artículos seleccionados para confeccionar el informe.		
Atributo	Tipo	Descripción
título	Similar a la tabla Escr	Similar a la tabla Escr
texto		