

Universidad Central “Marta Abreu” de Las Villas
Facultad Matemática, Física y Computación



Título: Traductor de español a latín para la nomenclatura botánica.

Autores: Marvin Ávila Kotliarov.
Noeslen Mendoza Morales.

Tutores: Dra. Zenaida García Valdivia.
M.Sc. Susana Carreras Gómez.

Carrera: Ciencia de la Computación.

Curso: 2005-2006.

Resumen

Los profesionales de la botánica requieren conocer un sistema preciso y universal que les permita describir las nuevas unidades que van siendo incorporadas al sistema de su ciencia. En la Universidad Central “Marta Abreu” de las Villas (UCLV) existe un Jardín Botánico en el cual se llevan a cabo investigaciones acerca del mundo vegetal. Por solicitud de esta entidad, surgió la idea de desarrollar una aplicación computacional que dotara a sus especialistas de la herramienta básica en la práctica taxonómica.

En el presente trabajo se manejan los conceptos de traductores automáticos, lingüística computacional, inteligencia artificial, los cuales constituyen la base teórica de nuestra investigación.

La aplicación se desarrolló en dos etapas. En la primera de ellas se realizó el diseño del programa que daría solución al problema planteado. Para ello se utilizó la notación del Lenguaje de Modelación Unificado. En la segunda etapa se procedió a la implementación del programa utilizando conjuntamente la programación a través de un lenguaje declarativo, como lo es Prolog, y la programación orientada a objeto, en nuestro caso Object Pascal.

De esta forma quedó desarrollado EsLatín versión 2.0, traductor de español a latín para la nomenclatura botánica, que constituye una herramienta muy útil y eficiente.

Abstract

Botanics professionals require to master an accurate and universal system which allows them to describe the new units incorporated to the system of their science. In the Universidad Central “Marta Abreu” de las Villas (UCLV) there is a Botanical Garden in which investigations regarding this field are carried out. In order to give their specialists the basic tools for the taxonomic practice, the idea of a software application was developed.

The theoretical grounds of our investigation are represented by the concepts of automatic translators, computer science linguistics and artificial intelligence.

The application was developed in two steps. In the first one, we made the program design for the solution of the given problem. We used the notation of the Unified Modeling Language. In the second step we implemented the program we used, together with the programming of a declarative language, as it is Prolog, and the object oriented programming, in this specific case Object Pascal.

In this way we obtained the version EsLatín 2.0, Spanish-Latin translator for the botanical terminology, that constitutes an useful and efficient tool.

“La matemática es la ciencia del orden y la medida, de bellas cadenas de razonamientos, todos sencillos y fáciles.”

René Descartes.

A mi madre, por haber luchado incansablemente para verme realizado como profesional. Por ayudarme a tomar las decisiones correctas cuando la juventud e inexperiencia me hubieran hecho cometer errores.

A mi padre, por sus consejos, por darme siempre la mano cuando la he necesitado, por ayudarme a pensar.

Noeslen.

A mi madre, por haber luchado a mi lado cada día de mi vida.

Marvin.

A mis hermanos Nilssen y Nislen, por su confianza ciega en mí, por brindarme siempre todo su apoyo.

A mi compañera Ianna, por creer en mí, por recordarme todo lo que olvido, por ayudarme en todo.

A Nino por alentarme siempre a continuar estudiando.

A mis suegros Delfina y Joselín, por su preocupación constante, por sus consejos e indicaciones.

A mis compañeros: Eylín, Yanko, Abdel, Alejandro, Jose, Dariem, Lisdán, Dasiel y Mario entre tantas cosas por cinco años de ayuda diaria, por no habernos separado en momentos difíciles, por ser los mejores amigos.

A mis tíos Idalmis y Darío, siempre dando lo mejor, agradezco mucho todas sus ideas.

A mi amigo Iroel, que desde lejos se preocupó y me aconsejó sobre la tesis.

A mis tutoras Zenaida y Susana, por dedicarnos siempre un espacio en su tiempo para allanar las dudas, una y otra vez.

A mi compañero de trabajo Marvin, por poner tanto de su parte para enfrentar el trabajo con la seriedad que lo hicimos, por estar pendiente de todo.

A todos les agradezco el hacer posible la realización de este trabajo que demuestra nuestra valía como profesionales y la capacidad que ya tenemos de enfrentar un nuevo reto en la nueva vida que comenzaremos de hoy en adelante.

Muchas gracias.

Noeslen.

A mi madre, la persona que más quiero en el mundo, por haberme dado la fuerza necesaria durante todos estos años para enfrentarme a las dificultades de la vida.

A Yosán, la hermana que siempre quise tener, y que me ha querido como tal.

A mis amigos Alejandro, Eylín, Yanko, Jose, Noeslen, Abdel, Dasiel, Lisdán (El Billy), Dariem y Mario, que estuvieron junto a mí en los buenos y malos momentos.

A mi novia por su cariño incondicional.

A mis tutoras Zenaida y Susana por su inestimable ayuda.

A mis profesores de clases que con tanta sabiduría me prepararon para ser un buen profesional.

A mi abuela por preocuparse tanto por mí.

A mi compañero de tesis Noeslen, por su esfuerzo incansable y su paciencia infinita.

Y a todas aquellas personas que de una forma u otra, ayudaron a la realización de este trabajo.

Muchas gracias.

Marvin.

Índice

Introducción.....	1
CAPÍTULO I. LA INTELIGENCIA ARTIFICIAL EN LA IMPLEMENTACIÓN DE LOS TRADUCTORES AUTOMÁTICOS.....	7
1.1 Los traductores en la computación.	7
1.2 Uso de las técnicas de Inteligencia Artificial en la implementación de traductores automáticos.	9
1.2.1 Introducción.	9
1.2.2 La lingüística computacional, una rama de la Inteligencia Artificial.	10
1.2.3 El Prolog, un lenguaje de programación de la Inteligencia Artificial.	12
1.3 Uso del latín en la nomenclatura botánica.	16
CAPÍTULO II. IMPLEMENTACIÓN DE ESLATÍN, TRADUCTOR DE ESPAÑOL A LATÍN PARA LA NOMENCLATURA BOTÁNICA.	19
2.1 Introducción.	19
2.2 Descripción y análisis del sistema.	19
2.3 Estados de la interfaz con el usuario.	21
2.4 Componentes del sistema.	23
2.5 Análisis del módulo ejecutable.	25
2.5.1 Consideraciones iniciales.	26
2.5.2 Predicado list_dec/4.	27
2.5.3 Predicados auxiliares.	28
2.6 Principales clases que conforman al sistema.	29
2.7 Acerca del instalador del Software.	34
CAPÍTULO III. MANUAL DE USUARIO.	38
3.1 Características del software.	38
3.2 Áreas de aplicación de la herramienta.	38
3.3 Requisitos del sistema.	39
3.4 Pasos para la instalación del sistema.	39
3.5 Inicio de la sesión de trabajo.	40
3.6 El menú Archivo.	43
3.7 El menú Editar.	44
3.8 El menú Diccionario.	45
3.9 El menú Herramientas.	50
3.10 El menú Ayuda.	53
Conclusiones.....	56
Recomendaciones	57
Referencias bibliográficas.....	58
Bibliografía	59
Anexos	

Introducción

La computación, a pesar de ser la ciencia más joven, es la que más aceleración ha mostrado en su desarrollo. Asimismo, por las ilimitadas posibilidades que brinda, ha logrado que en ella se verifique una integración de todas las artes y las ciencias. Una de sus herramientas, la computadora, es la más utilizada en el procesamiento y almacenamiento de la información, y en su capacidad como desafío intelectual no tiene precedentes en la historia de la humanidad.

Desde el surgimiento de las primeras computadoras, profesores y científicos como F. B. Skinner, Norman Crowder y otros, dedicaron mucho tiempo al estudio de lo que hoy conocemos como “Enseñanza Programada” o “Enseñanza Asistida por Computadora” (EAC). La introducción de la computación en el mundo de la educación y como soporte principal para las investigaciones, ha marcado un notable auge en la educación a nivel mundial.

Resulta realmente ventajoso el uso de la EAC, ya que este método ofrece nuevas posibilidades a los educandos, y logra un nivel muy elevado de satisfacción de sus necesidades de conocimiento. Las posibilidades para usar la tecnología de la información y las comunicaciones de forma innovadora en las escuelas y centros científicos son ilimitadas. Esta tecnología ha propiciado el desarrollo de nuevos conceptos dentro del proceso docente-educativo.

Haciendo uso de la computadora como herramienta de ayuda se han logrado desarrollar aplicaciones para el procesamiento del lenguaje natural, tales como correctores ortográficos, procesadores de textos especializados, reconocedores de voz, analizadores sintácticos y traductores.

Planteamiento del Problema:

Los profesionales de la botánica requieren el conocimiento de un sistema preciso y universal de nomenclatura que les permita describir las nuevas unidades o grupos taxonómicos que van siendo incorporados al corpus que conforma el sistema de su ciencia. Ese sistema existe, y constituye el fruto del esfuerzo que desde hace siglos han desarrollado los más notables botánicos por conseguir tanto una explicación científica del mundo vegetal, una organización de su conocimiento, como una comunicación uniforme, un tecnolecto universal, que derribe barreras geográficas y diacrónicas y permita el global entendimiento acerca de este ámbito que, efectivamente, es patrimonio de toda la humanidad.

Como ha sucedido con la mayoría de las ciencias, el sistema lingüístico elegido para conformar este tecnolecto es el latín, con un alto ingrediente de lengua griega adaptada al sistema latino. Incluso se latinizan los nombres propios de los descubridores de nuevos ejemplares.

En la Universidad Central “Marta Abreu” de las Villas (UCLV) existe un Jardín Botánico, unidad docente de la Facultad de Ciencias Agropecuarias, en el cual se llevan a cabo investigaciones acerca del mundo vegetal. En este centro laboran especialistas investigadores de alta calificación, que al mismo tiempo, en su mayoría, ejercen la docencia en la Facultad a la que están adscritos.

En nuestro país, a diferencia de las comunidades europeas más desarrolladas, y de un número de países de nuestro Continente, no se desarrolla la enseñanza de la lengua latina en instituciones del sistema general de enseñanza, sino sólo en el reducido contexto de la carrera de Letras, que nada ha tenido que ver hasta hace muy poco con los estudios acerca de la botánica. Por estas razones se ha padecido la carencia de una herramienta indefectible en la práctica taxonómica.

Ante esta evidencia, el Jardín Botánico solicitó al Departamento de Letras de la Facultad de Humanidades la impartición de un curso de postgrado de Latín, que dotara a sus especialistas de la herramienta básica en la práctica taxonómica.

Pero al encuentro de ambas disciplinas (latín y botánica), surgió la evidencia de la desinformación de los especialistas de una ciencia acerca de la otra, y la imposibilidad de desarrollar un curso de postgrado con la duración necesaria para capacitar debidamente a los estudiantes. De manera que era necesario buscar alternativas. Surgió entonces la idea de incorporar al problema el empleo de la computación, que ha venido a ser la clave de la solución.

El hecho de que aquí confluyan varias ciencias (cibernética, latín, botánica y pedagogía), ha condicionado el desarrollo por etapas de un proceso que requiere un sistemático y profundo análisis, y que no ha alcanzado aún su meta más inmediata, que consiste en alcanzar la mayor cercanía posible a un traductor automático de taxonomías completas, pues se hace necesaria la valoración sistemática y por etapas de los diferentes especialistas que paulatinamente van encauzando la investigación.

Como antecedente de nuestro trabajo en esta investigación, se desarrolló una primera versión de un software con el fin de automatizar el proceso taxonómico. El mismo fue desarrollado como trabajo de curso por parte del estudiante Noeslen Mendoza Morales en su cuarto año de la carrera Ciencia de la Computación. Por las razones antes mencionadas, esta versión no cumplía aún con todos los requerimientos de la problemática en cuestión. Por ello se decidió dar continuidad al tema a través del presente Trabajo de Diploma.

Objetivo General:

Diseñar e implementar un software que facilite el proceso de traducción del español al latín, y el proceso de latinización de los nombres propios de los descubridores,

utilizando la fusión de la programación orientada a objeto con la programación declarativa, mejorando la capacidad de traducción y las funcionalidades de la primera versión de este software en aras de sentar bases concretas en el proceso de construcción de un definitivo traductor automático de una taxonomía completa.

Objetivos Específicos:

- Analizar cuáles requerimientos de los que posee la versión anterior del software serán mejorados.
- Proveer al software de documentos de ayuda para el contenido.
- Adicionar funcionalidades de modificación de la base de conocimiento.
- Crear un módulo de interacción visual más eficiente y cómodo que el de la versión anterior.
- Adicionar las funcionalidades referentes a la traducción de adjetivos, adverbios y de sustantivos de la tercera, cuarta y quinta declinación del latín.

Preguntas sobre la Investigación:

- ¿Sería conveniente representar el proceso de traducción mediante reglas?
- ¿De que forma se almacenarían los datos del diccionario?
- Después de una modificación al diccionario, ¿Sería posible hacer uso de los nuevos datos sin tener que esperar a que se inicie nuevamente la aplicación?

Justificación de la Investigación:

El proceso de registrar las nuevas especies, particularmente en la Botánica es un proceso bastante exquisito, en el cual juega un papel importante la nomenclatura botánica y la latinización de los nombres propios. Todos los nombres científicos dados a las plantas o grupos de plantas (taxones) son nombres extraídos del latín.

Hasta el momento, los científicos que se ocupan de la investigación en el mundo vegetal, necesitan tener un conocimiento del latín, lo que conlleva a que su trabajo se vea sistemáticamente interrumpido a causa de ello. Realizar el proceso de nomenclatura utilizando el latín resulta, por lo demás, una tarea un tanto engorrosa, en la cual el gasto de tiempo es un agravante que atenta directamente contra los resultados de la investigación. Dicha tarea se realiza manualmente, o bien con el auxilio de diccionarios, o de lo contrario se necesita de la supervisión de personal calificado, preparado profesionalmente. Conjuntamente se puede afirmar que la eficiencia del trabajo se ve afectada en los lugares donde se realiza este tipo de estudios. En el caso del Jardín Botánico que radica en la UCLV, el personal encargado de llevar a cabo la práctica taxonómica, necesita de la ayuda de un software informático para realizar dicha tarea. Resulta de mucha comodidad que se automatice la latinización de nombres propios a través de una aplicación computacional, de la cual se derive un ahorro de tiempo, además de la liberación de los científicos respecto a la necesidad de tener un conocimiento avanzado del latín.

En una primera versión se realizó una aplicación sencilla con un número reducido de funcionalidades, que sirve para realizar en un inicio algunas traducciones. En el presente software se pretende aumentar el nivel de profesionalidad del anterior, y así poder ofrecer una herramienta capaz de realizar las labores anteriormente mencionadas.

El trabajo ha sido organizado en tres capítulos, de los cuales aparece a continuación una efímera descripción:

Capítulo I: En este capítulo se tratan los temas referentes a la adaptación del software como un producto con fines instructivos, teniendo en cuenta los paradigmas de la Lingüística Computacional. Se hace referencia a las facilidades que ofrece la Inteligencia Artificial para el desarrollo de este tipo de aplicaciones y se explica cómo se usa el latín en los procesos de Nomenclatura Botánica.

Capítulo II: El segundo capítulo recoge en sí toda la labor de análisis previo de los problemas y los métodos a utilizar, el diseño y organización de las clases y la explicación de la implementación del software, EsLatín versión 2.0.

Capítulo III: El tercer y último capítulo está dedicado a las orientaciones a seguir por el usuario en el uso del EsLatín como traductor, así como las explicaciones funcionales del software en su totalidad.

CAPÍTULO I. LA INTELIGENCIA ARTIFICIAL EN LA IMPLEMENTACIÓN DE LOS TRADUCTORES AUTOMÁTICOS.

En este capítulo se explican las facilidades de la inteligencia artificial en el procesamiento del lenguaje natural. Se aborda el tema de la programación lógica y sus técnicas en el diseño e implementación de traductores. Se habla además del uso del latín como lengua utilizada para la nomenclatura de nuevas especies en la botánica.

1.1 Los traductores en la computación.

La aspiración de obtener artefactos mecánicos que sirvan para superar las barreras lingüísticas es muy antigua. En el siglo XVII se hablaba de la utilización de diccionarios mecánicos (basados en códigos numéricos universales) para superar las barreras del lenguaje, dentro de un movimiento a favor de la creación de una “lengua universal” no ambigua, basada en principios lógicos y símbolos icónicos, que permitiese comunicarse a toda la humanidad. Este empeño precede por bastante tiempo a la propia existencia del ordenador. Por ello, se puede entender que desde el momento en que un ordenador estuvo disponible en la década de 1940, la traducción automática pasó a convertirse inmediatamente en una de las aplicaciones estrella de la informática.

Los primeros desarrollos informáticos reseñables se realizaron en el famoso ordenador ENIAC en 1946. Entre los investigadores pioneros hay que citar a Warren Weaver, de la Fundación Rockefeller. Según (Barr and Feigenbaum, 1981), Weaver decía que los artículos escritos en Ruso, él consideraba que en realidad estaban escritos en Inglés, pero codificados a través de extraños símbolos. Fue quien dió a conocer públicamente la disciplina anticipando posibles métodos científicos para abordarla.

En 1951 el prestigioso Instituto de Tecnología de Massachusetts (MIT) puso a uno de sus especialistas, Yehoshua Bar-Hillel a trabajar con dedicación exclusiva en la Traducción Automática (TA). La primera demostración pública de un traductor

automático se llevó a cabo en 1954, en la Universidad de Georgetown, con ayuda de IBM y la participación del investigador Leon Dostert.

A mediados de la tercera década se llegó a la conclusión de que no se había obtenido traducción automática para textos científicos genéricos, y tampoco parecía que se iban a obtener a corto plazo, lo cual redujo el presupuesto asignado a la investigación en Estados Unidos. Pese a ello, fue una época de enormes avances en el plano teórico.

La etapa comprendida entre los años 70 y 80 se caracteriza por un fuerte desarrollo de los métodos simbólicos y una gran vitalidad de la investigación en sintaxis (gramáticas basadas en la unificación de rasgos) y en semántica (formalismos basados en la lógica de predicados). Sin embargo, los avances en el plano teórico no acababan de trasladarse al terreno de los resultados.

Los inicios de la década de los 90 está marcada por acontecimientos negativos. En Europa se cancela la financiación de los más importantes y prestigiosos proyectos relacionados con la TA. Paralelamente en Japón hace su aparición en el mercado un nuevo tipo de producto de traducción asistida, de diseño muy distinto a los anteriores. Otro aspecto destacable de esta etapa es el desarrollo de Internet, así como el cambio de enfoque de la traducción hacia la localización.

En la actualidad estamos ante el comienzo de una etapa nueva. El impulsor de todos estos desarrollos es Internet, con implicaciones que todavía no podemos anticipar, pero que en gran medida dependen de cómo se resuelvan la propiedad intelectual y los derechos de explotación de los recursos lingüísticos que se van acumulando. En la actualidad se están realizando estudios en sistemas basados en la estadística, que constituye el enfoque que mejores expectativas genera.

1.2 Uso de las técnicas de Inteligencia Artificial en la implementación de traductores automáticos.

La traducción automática, disciplina que trata de las maneras y métodos para lograr la traducción de textos de forma automatizada, es una rama de la Inteligencia Artificial (IA). Esta ciencia actualmente se halla en un importante período de renovación y revisión de sus fundamentos. Tradicionalmente, el enfoque dado es el estructural o gramatical, entendiendo este último por la descripción por métodos formales procedentes de la programación lógica y de la estructura de sus sistemas de reglas en el procesamiento del lenguaje natural.

1.2.1 Introducción.

Muchos son los logros obtenidos desde que nació la IA. Hoy existen varios traductores automáticos que ya funcionan normalmente a escala industrial entre múltiples pares de idiomas, y sofisticados procesadores textuales dotados con una amplia gama de herramientas de consulta disponibles, incluso on-line, tales como glosarios, tesauros y correctores ortográficos, gramaticales y estilísticos.

En el artículo (2006) *¿Qué es Procesamiento del Lenguaje Natural?* se describe el esquema general de la mayoría de los traductores automáticos:

- El texto en el lenguaje fuente se transforma a una representación intermedia.
- De ser necesario, se hacen algunos cambios a esta representación.
- Luego, esta representación intermedia se transforma al texto en el lenguaje final.

1.2.2 La lingüística computacional, una rama de la Inteligencia Artificial.

En (Domínguez Burgos, 2002) se define a la lingüística computacional (LC) como *una ciencia interdisciplinaria que se ubica entre la lingüística y la informática, con énfasis en la lingüística. Su fin es la elaboración de modelos computacionales que reproduzcan uno o más aspectos del lenguaje humano.*

La LC no es ámbito exclusivo de centros académicos. A decir verdad, hoy en día la LC está siendo impulsada en gran medida por la industria privada. Las empresas dedicadas a la informática han reconocido desde hace tiempo que el procesamiento automático del lenguaje humano constituirá uno de los principales campos de desarrollo en las próximas décadas. No obstante a ello, el objetivo de nuestro trabajo es centrarnos en la parte académica de esta rama. Los lingüistas computacionales trabajan en varios programas, de los cuales resalta por su interés el relacionado con la producción de traducciones automáticas de textos y la ayuda a los traductores humanos en su labor.

Todo aspecto del lenguaje humano puede ser de interés para la LC. Se trabaja en desarrollar aplicaciones para el análisis automático de la fonética, la fonología, la morfología, la sintaxis, la semántica y la pragmática. Los científicos procuran desarrollar sistemas que posibiliten el diálogo entre humanos que usan idiomas diferentes (traducción automática) o entre humanos y máquinas (sistemas de diálogo, sistemas expertos).

Los sistemas de traducción han sido criticados por sus grandes limitaciones frente a un traductor humano, sin embargo los programas actuales pueden producir traducciones aproximadas que ayudan a los seres humanos a determinar la relevancia de un texto en un idioma extranjero determinado (y su traducción por parte de un ser humano) o pueden servir de esbozo de traducción para un editor humano. En algunas áreas de conocimiento bastante limitadas como los servicios de pronóstico del clima, los

softwares de traducción automática pueden generar versiones bastante adecuadas de un texto producido inicialmente en otro idioma.

Los enfoques principales para programas de traducción automática son tres: el enfoque de traducción palabra por palabra, la traducción por transferencia y la traducción por medio de una llamada "interlingua". Como el contenido semántico y pragmático de una palabra no se puede determinar generalmente de manera aislada, es difícil conseguir una traducción adecuada de alguna frase con el reemplazo de una palabra por otra. Es por ello que el primer enfoque es poco utilizado. El enfoque de transferencia es el más usado hoy día. El análisis sintáctico y semántico de las oraciones del texto origen produce árboles de estructuras que son transformados, a nivel sintáctico y morfológico, en oraciones del idioma de destino. En el enfoque de traducción por medio de una interlingua, el programa de traducción automática analiza morfosintáctica, semántica y pragmáticamente el texto original y con esta información produce una representación intermedia en un lenguaje humano natural o artificial (como el inglés o el esperanto) o en una representación totalmente abstracta del significado. A partir de este lenguaje o de esta representación abstracta, el módulo de síntesis procede a elaborar un texto en el lenguaje de destino.

Actualmente se intenta combinar la traducción por transferencia o interlingua. Además, se buscan cada vez más los mecanismos que faciliten la labor del traductor humano, las llamadas Memorias de Traducción. Igualmente, los sistemas de traducción usan cada vez más un componente semántico-pragmático, y en algunos casos están comenzando a resolver anáforas (repeticiones) y otras referencias no sólo a nivel de la oración, sino a nivel del texto.

1.2.3 El Prolog, un lenguaje de programación de la Inteligencia Artificial.

El lenguaje de programación lógica Prolog resulta de mucha utilidad en la realización de traductores automáticos, debido a que ofrece facilidades para representar y utilizar el conocimiento que se tiene sobre un determinado dominio. El dominio es un conjunto de objetos, y en nuestro caso particular se trata del conjunto de palabras que conforman un lenguaje. El conocimiento se representa por un conjunto de relaciones que describen las propiedades de los objetos y sus interrelaciones, lo que vendría a ser la sintaxis y la gramática de dicho lenguaje. Un conjunto de reglas que describa estas propiedades y estas relaciones es un programa Prolog. A continuación se exponen las ventajas que ofrece Prolog para facilitarnos la tarea de traducir de un lenguaje a otro.

Las razones de la elección de este lenguaje de programación están expuestas claramente en (Miguel Vergés, 2001) :

- PROLOG es un lenguaje de programación específico para el procesamiento del lenguaje natural.
- El método de funcionamiento utilizado por PROLOG es un método de razonamiento deductivo muy similar al razonamiento humano, aplicado sobre el conjunto de fórmulas lógicas que componen el programa.
- En la base de PROLOG se encuentra la idea de que un algoritmo puede dividirse en dos partes: la determinación de QUÉ hay que resolver (lógica del programa) y, la indicación de CÓMO utilizar el conocimiento contenido en el programa para obtener la solución buscada (control del programa). El programador únicamente ha de preocuparse de la lógica del programa, dejando que el lenguaje se ocupe del control. El control es siempre el mismo y viene dado de forma automática por la implementación del lenguaje.

Para resolver el problema que se plantea en este trabajo, se pretende dar la mayor utilidad posible al Prolog dentro de los enfoques para programas de traducción que se mencionaban anteriormente.

El primer paso que se realiza en una traducción es la transformación morfológica entre las palabras como las secuencias de letras y la llamada representación morfológica del texto: la secuencia de las estructuras de palabras en la forma del lema (que puede servir como una clave a una base de datos que guarda todas las propiedades de la palabra) y las propiedades específicas en el texto.

La representación morfológica del texto es, en este caso, un conjunto de hechos (siempre verdaderos) en Prolog, que da los lemas y las propiedades de cada palabra del texto. Esto significa que a cada palabra en español corresponde un hecho en el programa donde aparece la palabra traducida en latín y sus propiedades. En este paso, se construye un diccionario morfológico.

Además de los hechos, en programación lógica existen otros dos tipos de sentencia: las interrogantes y las reglas. Estas sentencias también se denominan cláusulas de Horn en honor al lógico Alfred Horn, quien las introdujo en 1951. Este tipo de cláusulas se utilizan para resolver problemas a través de la lógica. Su utilidad radica en que para ellas existe un procedimiento de inferencia de tiempo polinomial. El conocimiento no siempre se puede representar como un conjunto de cláusulas de Horn, pero en el caso de poder hacerlo, el procedimiento de inferencia es sencillo: se utiliza Modus Ponens Universal o Generalizado.

Los programas lógicos están compuestos por este tipo de cláusulas (hechos y reglas) y no por expresiones lógicas. De aquí se derivan según (García Valdivia, 2000) dos importantes consecuencias:

1. Debido a la representación uniforme se puede escribir un sencillo y potente intérprete que resulte eficaz para realizar la inferencia a partir de un programa lógico y una interrogante.
2. La lógica de los sistemas de cláusulas de Horn es decidible (al contrario de la lógica de predicados de primer orden completa). En forma general un problema es decidible si es posible resolver (determinar si tiene solución o no) en un número finito de pasos.

Las reglas de producción son un subconjunto del cálculo de predicados con una componente añadida que indica cómo usar la información almacenada en reglas, esta componente no es más que la estrategia de solución de conflictos que el usuario tiene que definir para el control de la inferencia. En cuanto a la capacidad de expresión, los sistemas de producción son equivalentes a los sistemas construidos usando un lenguaje de programación lógica (por ejemplo PROLOG).

Aunque han surgido muchas técnicas para organizar colecciones de reglas, según (Lezcano, 2000) todas ellas comparten las siguientes características:

- Son modulares: Cada regla define una parte de conocimiento pequeña y relativamente independiente.
- Pueden incrementarse: Se pueden adicionar nuevas reglas a la base de conocimiento con relativa independencia de las ya existentes.
- Modificables: El conocimiento existente se puede variar o refinar independientemente de otras reglas, lo que constituye una consecuencia del hecho de ser modulares.
- Proporcionan transparencia al sistema: Es una importante y distintiva característica de los sistemas basados en reglas de producción. Por transparencia

se entiende su habilidad para explicar sus decisiones y línea de razonamiento, traduciendo la lógica de cada regla empleada en un cierto lenguaje natural.

Es posible utilizar el lenguaje de Programación Lógica Prolog para implementar Sistemas Basados en el Conocimiento (SBC), en este caso se utiliza como Método de Solución de Problemas una búsqueda a ciegas “Primero en profundidad” con una dirección de búsqueda dirigida por objetivos. El programador sólo se ocupa de escribir los programas que darán solución a su problema constituyendo estos la Base de Conocimiento que utiliza a la programación lógica como Forma de Representación del Conocimiento.

Existen según (Bello, 2002) dos conceptos relativos a los SBC que pudieran aclarar las particularidades que distinguen al PROLOG de otros lenguajes de programación, y que por ello hemos decidido utilizarlo:

- La separación del conocimiento de cómo éste es usado (distinción entre conocimiento y estrategia de control).
- El uso de conocimiento muy específico del dominio.

Con respecto al uso del Prolog se ha podido constatar las ventajas siguientes:

- ✓ Problemas complejos se resuelven con facilidad.
- ✓ Los programas se mantienen sin dificultad en su ciclo de vida.
- ✓ Obliga al programador a especificar una descripción bien estructurada del problema.
- ✓ Tiempo corto de desarrollo.
- ✓ Fácil de aprender, de leer y de modificar.

- ✓ Fácil manipulación de estructuras de datos complejas (listas, árboles, grafos, etc).
- ✓ El número de líneas de programas es generalmente una fracción de las requeridas en C, Pascal o Basic.

1.3 Uso del latín en la nomenclatura botánica.

El latín, a pesar de estar presente en la base de todas las lenguas romances y en una considerable porción del léxico de otras lenguas no romances, es innegablemente lo que en lingüística se conoce como “lengua muerta”. Es decir, que ya no existen hablantes de la lengua propiamente dicha, y que la creación natural de su *corpus* se detuvo, haciendo de esta lengua un registro cerrado.

La nomenclatura de la taxonomía botánica ha seguido la suerte del tecnolecto de la mayoría de las ciencias, teniendo como base las lenguas griega y latina. En el caso de la Botánica, el griego aparece a través de una modificación de su sistema lingüístico al sistema latino.

El registro que incumbe a la taxonomía botánica, constituye un subsistema del sistema latino, que aunque incluye un considerable número de términos, es lógicamente mucho menor que la lengua en su totalidad. Asimismo, las descripciones a las que tributa exhiben un comportamiento notablemente uniforme, que prescinde de matices estilísticos y conlleva valores pragmáticos bastante estables. De manera que el proceso de descripción botánica presenta características muy favorables para ser sometido a una traducción automática.

No obstante, el latín presenta una complejidad morfológica que no está presente en el español, que es el hecho de ser una lengua flexible, es decir, *declinable*. En virtud de esta condición, los nombres latinos (sustantivos, adjetivos y pronombres), adoptan diferentes terminaciones, llamadas *caso*, en dependencia de las diferentes funciones

sintácticas que desempeñan dentro de la oración gramatical. Existen doce casos o terminaciones para cada nombre: seis para el singular y seis para el plural. Por otra parte, el sistema nominal latino está compuesto por subsistemas llamados *declinaciones*, en virtud de lo cual los sustantivos se agrupan en cinco subsistemas diferentes, en cada uno de los cuales existen determinaciones de género, que en latín son tres: masculino, femenino y neutro. De manera que sólo para el sustantivo, por ejemplo, se puede encontrar con ciento ochenta terminaciones, dadas por cinco declinaciones, seis casos, dos números y tres géneros.

El otro punto álgido para una traducción automática se encuentra en el sistema adverbial: debido al sistema morfosintáctico que caracteriza a la taxonomía botánica, el empleo de varios adverbios que modifican al mismo tiempo a un mismo adjetivo, es casi obligado. Pero en ese mismo contexto puede suceder que uno de los adverbios esté a su vez modificado por otro adverbio. Conseguir un orden en la colocación de los miembros de esta categoría gramatical, donde quede absolutamente claro a quién modifica cada uno, presupone el dominio cabal de este procedimiento taxonómico.

Este trabajo constituye un importante paso para alcanzar una meta más ambiciosa, que es la traducción automática de una taxonomía completa. Aquí podemos encontrar la traducción y declinación de los sustantivos y adjetivos más empleados en las descripciones tomadas de una amplia y especializada bibliografía. Asimismo, se encuentra la traducción y la regencia de las preposiciones que aparecen en la mencionada bibliografía, dejando abierta la posibilidad de formar sintagmas con alta frecuencia de empleo. Como un dato muy útil a la vez que atractivo, un elevado número de adjetivos aparece acompañado por ilustraciones que muestran las características que definen. Por último, la apertura a la posibilidad de incorporar nuevos elementos a los diferentes apartados, hacen inagotable la utilidad de los presentes resultados.

Una información general acerca del sistema de la lengua latina, con una adecuada orientación metodológica, podrían colocar al especialista de la Botánica en condiciones

de realizar taxonomías de alta calidad lingüística con la ayuda de las herramientas que ofrecen los resultados de este trabajo.

CAPÍTULO II. IMPLEMENTACIÓN DE ESLATÍN, TRADUCTOR DE ESPAÑOL A LATÍN PARA LA NOMENCLATURA BOTÁNICA.

2.1 Introducción.

El traductor de español a latín EsLatín, surge a partir de un trabajo realizado anteriormente por el alumno Noeslen Mendoza Morales en su proyecto de curso de Programación Lógica en el cuarto año de la carrera Ciencia de la Computación. Este nuevo software retoma las ideas del trabajo anterior, sin embargo su diseño e implementación en esta versión son totalmente novedosos. El mismo permite realizar traducciones de palabras del idioma español al latín. También permite realizar actualizaciones al diccionario que sirve como herramienta a la aplicación.

La interfaz con el usuario resulta amigable y a la vez sencilla al ejecutar cualquier operación, además de permitirle al usuario una rápida familiarización con la herramienta. Se incluye un módulo visual de utilidad para reconocer rasgos propios de las plantas, y además sirve para orientar mejor cualquier tipo de búsqueda.

En este capítulo se explican cuáles han sido las herramientas computacionales utilizadas en la implementación del software, así como los diagramas correspondientes al diseño del mismo.

2.2 Descripción y análisis del sistema.

Para la modelación de este sistema se ha utilizado la notación del Unified Modeling Language (UML), lenguaje visual estándar muy utilizado en la especificación, visualización, construcción y documentación de los distintos aspectos relativos al desarrollo de un software.

Para una mejor comprensión de los requisitos del programa se identifican los casos de uso del sistema así como sus actores. Los actores interactúan con el sistema y permiten que se lleven a cabo los procesos de la aplicación, que son considerados los casos de uso.

En el problema que se aborda se han definido dos actores, cada uno con sus respectivos casos de uso. El primero de ellos es el usuario encargado de interactuar con el sistema y realizar cualquier traducción, así como de utilizar el visor de imágenes. El segundo de ellos es el experto en el lenguaje latín, que se encarga de actualizar el diccionario de la aplicación. Los casos de uso son los siguientes:

- Realizar una traducción: Caso de uso que corresponde a las acciones de efectuar la traducción de una palabra y comprobar los resultados de la solución.
- Visualizar imágenes: Utilizar el gestor de imágenes.
- Actualizar el diccionario: Caso de uso que incluye adicionar, eliminar o modificar palabras del diccionario por parte del experto.

La figura 2.1 muestra los actores y casos de usos encontrados durante el diseño del sistema.

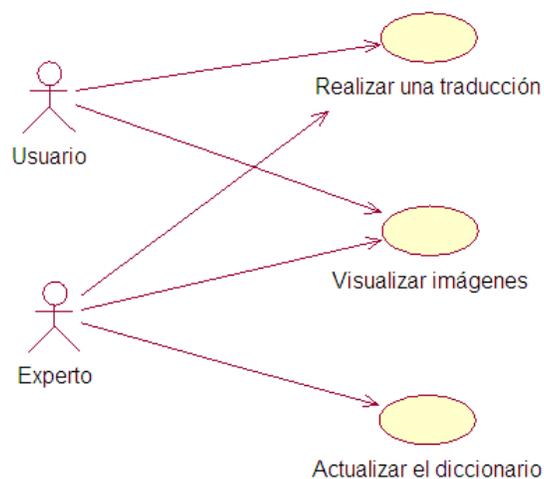


Figura 2.1 Diagrama de casos de uso y actores

También se presenta la expansión del caso de uso, Actualizar el diccionario en la figura 2.2.

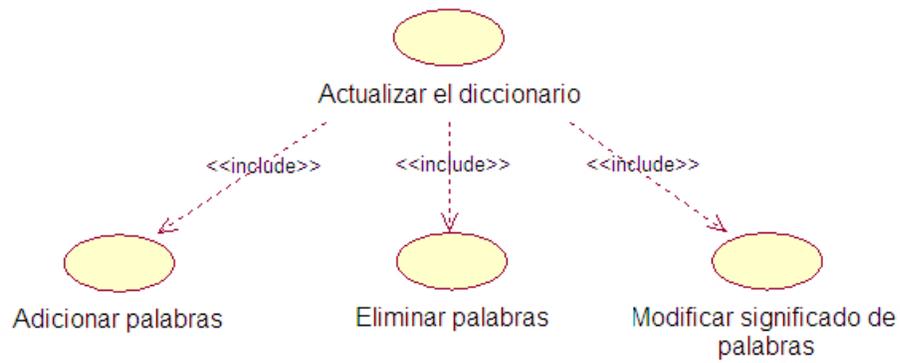


Figura 2.2 Expansión del caso de uso Actualizar el diccionario.

2.3 Estados de la interfaz con el usuario.

La notación UML resulta de mucha utilidad al representar los objetos de la interfaz de una aplicación. Los diagramas de estado sirven para describir la secuencia de los estados de un objeto así como los cambios que ocurren en el mismo ante la entrada de un evento. Un ejemplo esclarecedor lo constituyen las ventanas del sistema, para lo cual se muestran los diagramas de transición de estado para el caso de uso Realizar una traducción.

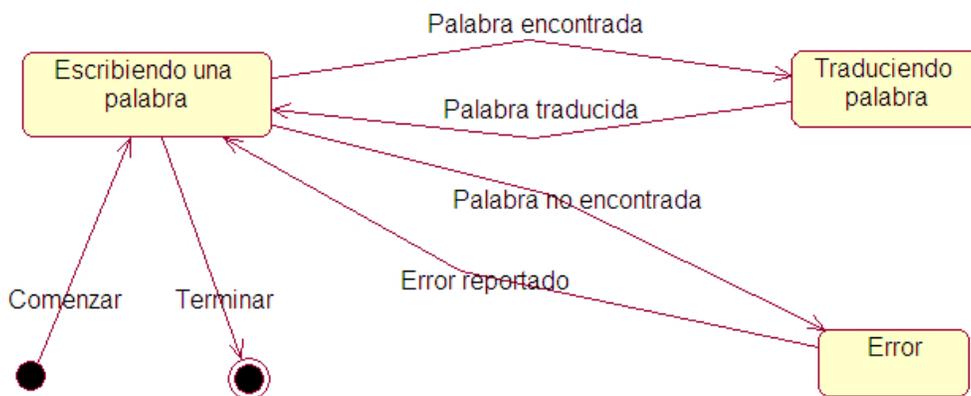


Figura 2.3 Diagrama de estado para caso de uso Realizar una traducción.

Toda vez que se realice una correcta modelación de los diagramas de estado de los objetos de una aplicación, los programadores pueden implementar un software más útil y con menos deficiencias. También se comprende de manera más simple cuáles son las entradas a la interfaz y las salidas del sistema, así como las acciones a realizar ante la ocurrencia de cualquier evento.

Por otra parte, al describir el caso de uso Actualizar el diccionario, es importante realizar una especificación del comportamiento de los objetos de esta interfaz a través de diagramas de estado a más bajo nivel. Se presenta la ventana Actualizar el diccionario (Véase Figura 2.3) y el diagrama de estado que describe una de sus expansiones, en este caso, Modificar el significado de las palabras en el diccionario (Véase Figura 2.4).

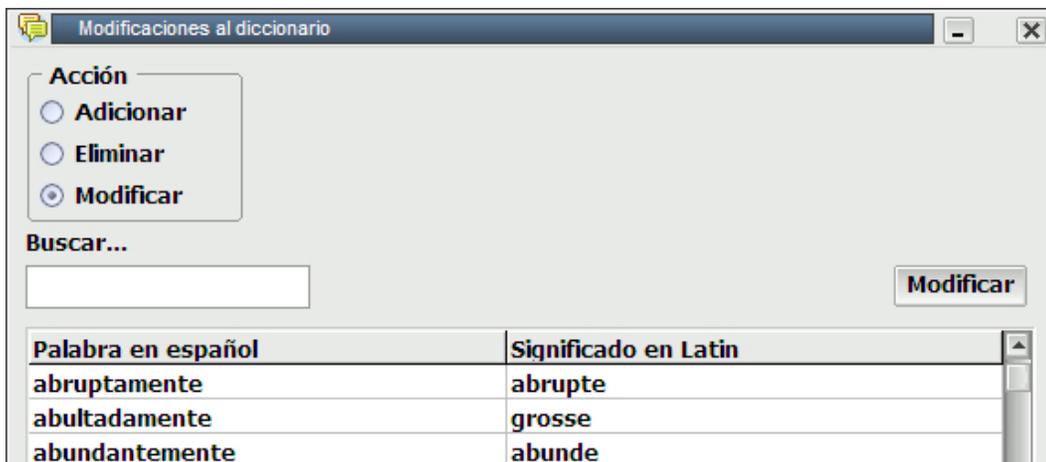


Figura 2.4 Ventana para actualizar el diccionario.

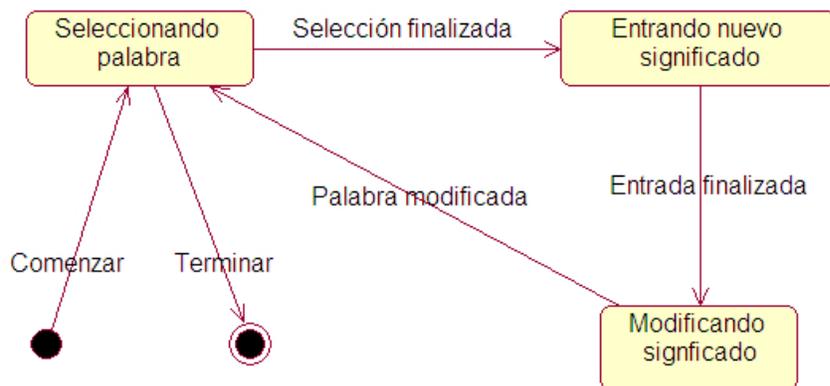


Figura 2.5 Diagrama de estados de Modificar el significado de las palabras.

2.4 Componentes del sistema.

Cuando se llevó a cabo la implementación del sistema se tomaron en cuenta cuáles eran las herramientas computacionales más eficaces para lograr una solución que fuera óptima para el usuario y que además permitiera a los integrantes del proyecto poner en práctica la mayoría de los conocimientos obtenidos durante la carrera. Los lenguajes y ambientes de programación escogidos han sido justificados anteriormente. Como resultado de este trabajo se obtuvieron dos componentes principales: una aplicación ejecutable implementada en Delphi (EsLatín.exe) y un módulo ejecutable simple (EsLatín.xpl) implementado en Amzi Prolog. Además se utilizó la biblioteca de enlace dinámico amzi.dll para lograr el enlace entre ambos módulos.

Una vez que se ha concluido un proyecto en Amzi Prolog, usualmente es compilado y se obtiene un módulo objeto (.plm). Los módulos objetos tienen la ventaja de ejecutarse más rápido que los ficheros fuentes interpretados por el Amzi Prolog. De nuestro programa Prolog se obtuvieron rules.pro y diccionario.pro. Cada uno de ellos fue compilado, obteniendo rules.plm y diccionario.plm.



Figura 2.6 Diagrama del proceso de compilación.

Los ficheros objetos son enlazados en un módulo ejecutable (.xpl) que se encuentran listos para ser utilizados por lenguajes de alto nivel como Delphi, Visual C++, Visual Basic.

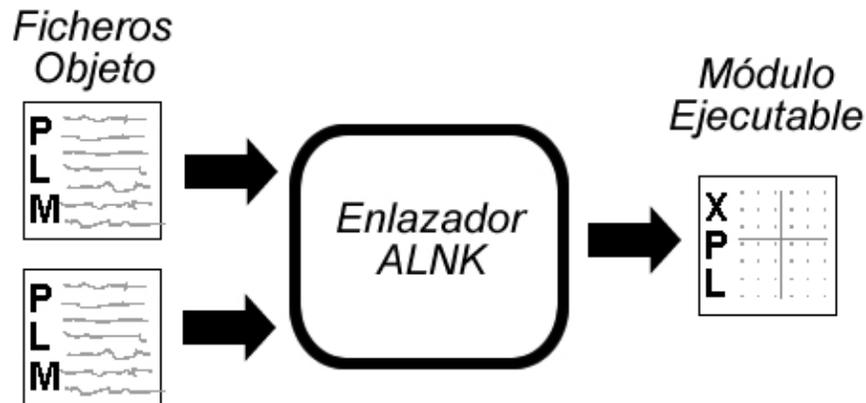


Figura 2.7 Diagrama del proceso de enlace.

En la etapa de implementación del proyecto el fichero rules.pro sólo se compila si las reglas del mismo han sido modificadas. Cuando se procede a implantar el software ya se utiliza sólo el fichero objeto rules.plm. En el caso del fichero fuente diccionario.pro sí es necesario compilar cada vez que se realice una modificación al diccionario.

Por cuestiones de seguridad se decide almacenar el diccionario en un fichero estructurado (diccionario.bin), de tal manera el archivo diccionario.pro no queda expuesto a un posible error de manipulación que afecte el buen funcionamiento de la aplicación. De esta forma antes de realizar la compilación es necesario crear el fichero diccionario.pro a partir de los datos que se encuentran en el fichero protegido.

Cuando se han obtenido ambos ficheros objeto se procede a realizar el enlace de los mismos y esto da como resultado el módulo ejecutable EsLatín.xpl.

Estas dos operaciones, la compilación y el enlace, son dependientes la segunda de la primera. Cuando esta aplicación envía la orden de ejecutarlas sucesivamente se produce un error debido al tiempo que demora el compilador en crear los ficheros que después debe utilizar el enlazador. Se comienza a enlazar cuando aún no se ha terminado la compilación, o sea, los ficheros no están listos para ser enlazados. Este problema se

solucionó implementando a través de las API de Windows un procedimiento de sincronización de los procesos que ejecutan estas acciones.

Amzi Prolog ofrece una interfaz de enlace compuesta de una clase (TLSEngine) que se encuentra en amzi.pas y cuyas funciones interactúan con una biblioteca de enlace dinámico (amzi.dll) encargada de realizar la comunicación entre el Prolog y la aplicación en Delphi.

La implementación de EsLatín.exe se llevó a cabo en Borland Delphi versión 7.0, que es una herramienta de programación muy nutrida de componentes, las cuales resultan de mucha utilidad al realizar aplicaciones visuales, y que permiten lograr ambientes agradables a la vista del usuario. La Biblioteca de Componentes Visuales (VCL en inglés) está compuesta por objetos visuales que son manipulados a gusto en tiempo de diseño. Esto permite conocer con antelación cómo va lucir la aplicación. Además, Borland Delphi permite utilizar el paradigma de la programación orientada a objeto (POO) muy eficaz para representar los programas como colecciones de objetos, lo cual sirve para ganar en funcionalidad así como aclarar la utilidad de las estructuras de datos presentes en una aplicación. Cada objeto constituye la instancia de una clase, la cual describe cualquier tipo de dato que se pueda imaginar. Las clases pueden ser relacionadas unas con otras y también formar jerarquías. Pueden también unirse mediante relaciones de herencia.

2.5 Análisis del módulo ejecutable.

El sistema EsLatín cuenta con un módulo EsLatín.xpl que contiene la representación del conocimiento del traductor. Es el encargado de realizar los análisis correspondientes para ejecutar una correcta traducción. Está compuesto de dos ficheros rules.pro y diccionario.pro, programados en Prolog. En rules.pro se encuentran aquellos predicados que se utilizan para traducir palabras como sustantivos y adjetivos, y otros que facilitan el análisis sintáctico de lo que se desea traducir. En diccionario.pro se encuentran todas las palabras con que cuenta el programa y sus correspondientes significados en latín. El

predicado principal de EsLatín.xpl es list_dec, el cual se describe posteriormente. Este es el encargado de devolver la traducción de una palabra, ya sea sustantivo o adjetivo.

2.5.1 Consideraciones iniciales.

El latín es una lengua donde hay que tener en cuenta algunas particularidades y reglas. Un sustantivo en latín se dice de distintas formas según el contexto donde se utilice. Tiene una parte invariable dentro de la palabra que se denomina Tema. Se puede declinar de cinco formas distintas y en cada una de ellas se especifican Modo, Género y Número. El modo puede ser: nominativo (N.), vocativo (V.), acusativo (Ac.), genitivo (G.), denominativo (D.) y ablativo (Ab.). El género puede ser: masculino (M), femenino (F), masculino y femenino (MF), neutro (N) y mixto (MX). El número al igual que en español es singular o plural. Cada sustantivo pertenece a una única declinación. Una traducción puede ser simple o avanzada. En una traducción simple el sustantivo se define por su nominativo, su genitivo y su género. Por su parte, cuando traducimos de forma avanzada entonces se declina el sustantivo completamente.

Traducción simple: corola → Corolla, Corollae, F.

Traducción avanzada: 1ra. Declinación MF.

Singular:

Plural:

N. → Corolla

N. → Corollae

V. → Corolla

V. → Corollae

Ac. → Corollam

Ac. → Corollas

G. → Corollae

G. → Corollarum

D. → Corollae

D. → Corollis

Ab. → Corolla

Ab. → Corollis

La declinación a la que pertenece un sustantivo se reconoce por la terminación del genitivo, y en este caso “ae” significa que el sustantivo pertenece a la primera declinación. Para una declinación es imprescindible conocer el nominativo y el genitivo.

Los adjetivos, por su parte, pueden ser de tres terminaciones, de dos terminaciones, o de una terminación. En cada caso se declinan igual que los sustantivos, solo con algunas diferencias. Los de tres terminaciones se declinan tres veces, una para género masculino, otra para femenino y una última para neutro. Los adjetivos de dos terminaciones se declinan una vez para masculino y femenino (MF) y otra para neutro. Por último, los adjetivos de una terminación se declinan una sola vez para género mixtos. De esta forma y para el problema que se plantea, se puede considerar un caso particular de los sustantivos al declinarlos, y solo se debe saber a cuál terminación pertenecen.

2.5.2 Predicado list_dec/4.

list_dec (Nom, Theme, ID_DEC, L) :-

findall (Mode/X, dec (Nom, Theme, ID_DEC, Mode, _, X), L).

Argumentos:

- Nom: Es el nominativo del sustantivo o adjetivo en latín que fue anteriormente traducido según el diccionario. Se representa a través de un átomo, por ejemplo: 'Corolla'.
- Theme: Es el tema de la palabra en latín. Se representa a través de un átomo.
- ID_DEC: Es un número que indica a que declinación pertenece el sustantivo o adjetivo. Se obtiene a través del predicado id_dec/5 que se encuentra en el mismo fichero y que se describe más adelante. Se representa con un átomo.
- L: Es una lista donde se van obtener los resultados de la declinación.

El predicado funciona de la siguiente forma: se declina el argumento de Nom en la declinación que indica ID_DEC. Se forma la lista L con las palabras que se obtienen de concatenar el tema (Theme) con las terminaciones de la declinación y los respectivos modos (Mode). Se hace uso de las facilidades que ofrecen los predicados de segundo

orden del Prolog, en este caso findall, mediante el cual se encuentran todas las instancias de X que satisfacen a dec (Nom, Theme, ID_DEC, Mode, _, X).

2.5.3 Predicados auxiliares.

```
dec(Nom, Theme, '1', 'N', 'S', X) :- stringlist_concat([Theme, 'a'],X).
```

Mediante este predicado se construye la palabra del sustantivo o adjetivo Nom concatenando el tema Theme con la terminación “a” que pertenece a la primera declinación, modo nominativo, número singular y se obtiene la variable X con la cual unifica. En el fichero se encuentra un predicado para cada palabra de una declinación.

También se definieron varios predicados que ayudan a la identificación de las declinaciones. Entre ellos se encuentran:

- id_dec(Nom,'AE', Genre, Clasif_Silab,'1'). → Significa que si el genitivo termina en “ae” entonces el sustantivo pertenece a la primera declinación masculinos o femeninos.
- id_dec('US', Gen, Genre, Clasif_Silab,'21') :- genitiveIorII(Gen). → Si el nominativo termina en “us” y el genitivo termina en “i” o “ii” entonces el sustantivo pertenece a la segunda declinación masculinos.
- id_dec('UM', Gen, Genre, Clasif_Silab,'22') :- genitiveIorII(Gen). → Si el nominativo termina en “um” y el genitivo termina en “i” o “ii” entonces el sustantivo pertenece a la segunda declinación neutros.
- id_dec(Nom,'IS', Genre, 'IMPAR','31') :- genreMF(Genre). → Si el genitivo termina en “is” y el nominativo y el genitivo son imparisílabos entonces el sustantivo pertenece a la tercera declinación imparisílabos masculinos o femeninos.
- id_dec(Nom,'IS', 'N', 'IMPAR','32'). → Sustantivos que pertenecen a la tercera declinación imparisílabos neutros.

- `id_dec(Nom,'IS', Genre, 'PAR','33')` :- `genreMF(Genre)`. → Si el genitivo termina en “is” y el nominativo y el genitivo son parisílabos entonces el sustantivo pertenece a la tercera declinación parisílabos masculinos o femeninos.
- `id_dec(Nom,'IS', 'N', 'PAR','34')`. → Sustantivos que pertenecen a la tercera declinación parisílabos neutros.
- `id_dec(Nom,'IS', 'MX', Clasif_Silab,'35')`. → Sustantivos que pertenecen a la tercera declinación mixtos.
- `id_dec(Nom,'US', Genre, Clasif_Silab,'41')` :- `genreMF(Genre)`. → Si el genitivo termina en “us” entonces el sustantivo pertenece a la cuarta declinación masculinos o femeninos.
- `id_dec(Nom,'US', 'N', Clasif_Silab,'42')`. → Sustantivos que pertenecen a la cuarta declinación neutros.
- `id_dec(Nom,'EI', Genre, Clasif_Silab,'5')` . → Si el genitivo termina en “ei” el sustantivo pertenece a la quinta declinación. El género es siempre femenino.

En `Diccionario.pro` se encuentran las traducciones simples de las palabras, ejemplo: `tsust('corola','Corolla, Corollae, f')`. Las palabras del diccionario están conformadas de sustantivos, adjetivos y adverbios. Estos últimos se traducen directamente, o sea, no se declinan y los predicados que hacen referencia a ellos se encuentran sólo en el diccionario.

2.6 Principales clases que conforman al sistema.

La aplicación queda diseñada, como lo exigen las tendencias de programación actuales, como un sistema orientado a objeto (SOO). Las entidades de mayor importancia dentro de esta técnica son las clases que definen a los objetos del programa. Estos se relacionan a través de los atributos y procedimientos que los conforman. En la figura 2.8 se muestra el diagrama de clases del sistema:

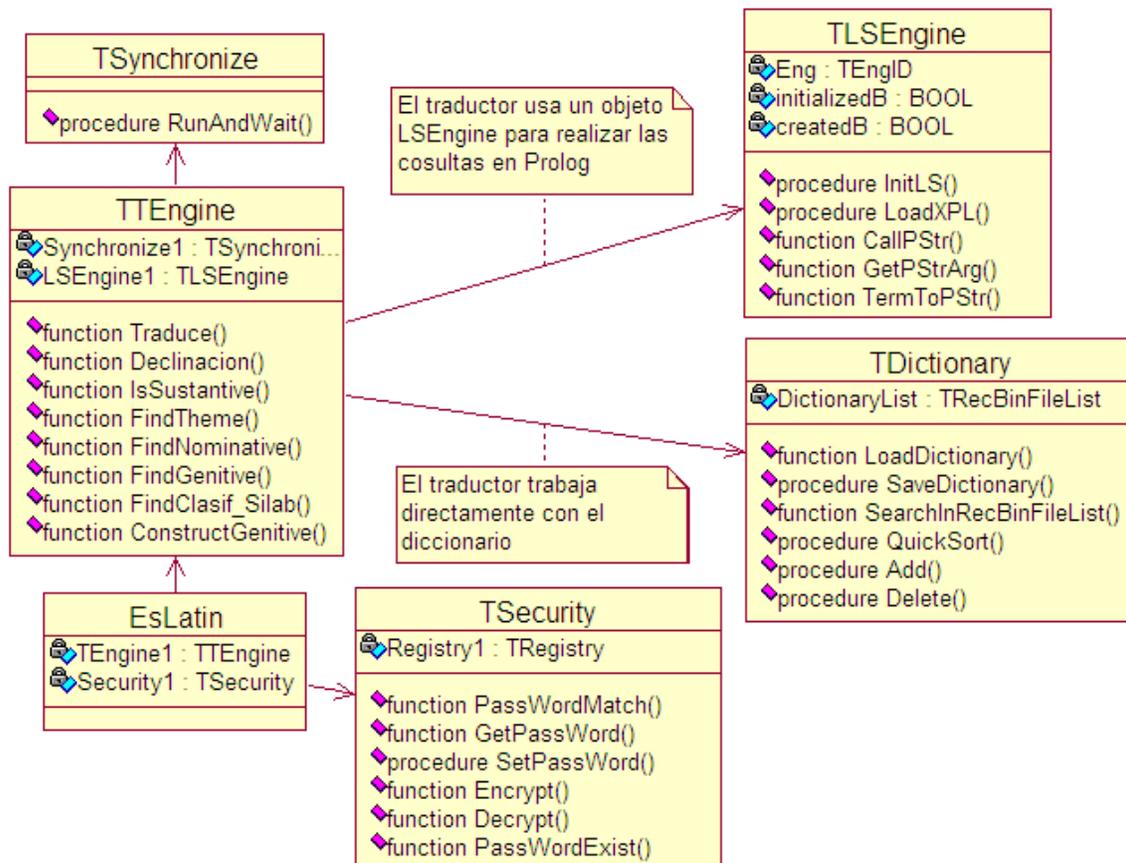


Figura 2.8 Diagrama de clases del sistema.

A continuación se explica detalladamente la implementación y funcionamiento de las clases del sistema.

TDictionary: Clase encargada de la manipulación del diccionario.

Tiene como primera tarea la creación del fichero diccionario.pro a partir del diccionario.bin mediante el método:

```
procedure RecBinFileToProFile (BinFileName,ProFileName:string);
```

Realiza las funciones de recuperar los datos desde el fichero hacia una lista donde se realizan las modificaciones, y también permite guardar la lista modificada en el fichero. Los métodos encargados de ello son respectivamente:

```
function LoadDictionary(FileName: string): RecBinFileList;
procedure SaveDictionary(FileName: string; List: RecBinFileList);
```

Otras de las tareas que se pueden ejecutar mediante esta clase son la búsqueda dentro de la lista y el ordenamiento alfabético de la misma. Esto se logra haciendo uso de:

```
function SearchInRecBinFileList(substr:string;List:TRecBinFileList):arrinteger;
procedure QuickSort(var List:TRecBinFileList;left,rigth:integer);
```

TSynchronize: Clase encargada de la sincronización de los procesos de compilación y enlace de los ficheros del Prolog.

Como se explicaba en el epígrafe 2.4 debe resolverse el problema de sincronización entre estos procesos. El objetivo fue crear un procedimiento que ejecute una aplicación en un proceso independiente y espere a que este termine. Invocando el procedimiento una vez para cada acción quedó resuelto el problema.

```
Procedure RunAndWaitShell
    (ExecutableFile,Args:string;ShowMode:integer;Handle:HWND);
```

Las API de Windows utilizadas para implementarlo fueron:

```
ShellExecuteEx(LPSHELLEXECUTEINFO lpExecInfo );
WaitForSingleObject(HANDLE hHandle,  DWORD dwMilliseconds);
```

TSecurity: Clase encargada de mantener la seguridad del sistema.

Anteriormente se explicaba las medidas adoptadas para mantener la integridad del fichero donde se almacena el diccionario (diccionario.bin). Sin embargo es necesario proteger el diccionario de una incorrecta manipulación por parte de los usuarios en

tiempo de ejecución del programa. Con este fin incluimos en el sistema la posibilidad de establecer una protección mediante el uso de una contraseña. Si esta opción está activada, para acceder a la ventana de modificaciones del diccionario es necesario introducir la contraseña correcta. La contraseña se protege mediante un algoritmo criptográfico y luego se almacena en el registro de Windows. Los procedimientos de la clase son los siguientes:

```
function PassWordMatch(s:string):Boolean;
function Encrypt(s:string):string;
function Decrypt(s:string):string;
function GetPassWord:string;
procedure SetPassWord(s:string);
```

TLSEngine: Clase que ofrece Amzi para realizar las consultas al módulo ejecutable.

Dicha clase encapsula las funciones implementadas en la Biblioteca de Enlace Dinámico (Amzi.dll) y que permiten tener acceso a los módulos ejecutables del Prolog (.xpl). Los métodos que utilizamos son:

procedure InitLS(xplname: String) → Inicializa el Servidor Lógico de Amzi.

procedure LoadXPL(xplname: String) → Carga el módulo ejecutable.

function CallPStr(var tp: TTerm; s: string): Boolean → Realiza una consulta a la base de conocimientos.

s: cadena que contiene la consulta a realizar.

tp: devuelve el término con el cual unificó la consulta realizada.

function GetPStrArg(t: TTerm; n: integer): string → devuelve el argumento n del término t.

function TermToPStr(t: TTerm): string → Devuelve como cadena el término t.

TTraductorEngine: Clase principal del sistema. Se encarga de manipular el proceso de traducción.

Esta clase construye los predicados con los cuales se van a realizar las consultas al módulo ejecutable para realizar una traducción, ya sea simple o que suponga una declinación completa. Los principales métodos implementados para efectuar esta labor son los siguientes:

function Traduce(Paragraph1:TStrings; Mode:integer):TStringList;

En dependencia del tipo de traducción que se desea realizar, simple o avanzada, esta función genera uno o varios predicados para realizar la consulta. En caso de una traducción simple se construye un único predicado. Por el contrario, si está en presencia de una traducción avanzada, se deben realizar diversas consultas. Es necesario recopilar toda la información necesaria, digamos género, tema, número de la declinación, y formar con dicha información los argumentos de los predicados para las consultas.

function Declinacion(s:string):integer; → Devuelve el identificador de la declinación a la que pertenece la palabra que se quiere traducir.

function IsSustantivo(s:string):boolean; → Devuelve si una palabra es un sustantivo o no.

function FindTheme(s:string):string; → Devuelve el tema de la palabra para realizar la declinación.

function FindNominative(s:string):string; → Encuentra el nominativo en una cadena que devuelve el diccionario y que contiene la traducción simple de la palabra a declinar.

function FindGenitive(s:string):string; → Encuentra el genitivo en la cadena que devuelve el diccionario con la traducción simple de la palabra a declinar.

function FindClasif_Silab(s:string):string; → Devuelve si el nominativo y el genitivo son parisílabos o imparisílabos.

function ConstructGenitive(nominative,genitive:string):string; → Si en la cadena que devuelve el diccionario con la traducción simple, se encuentra que el genitivo no es una palabra, sino una terminación, entonces se debe construir el genitivo, utilizando el nominativo y la terminación.

2.7 Acerca del instalador del Software.

Se ha querido que esta aplicación, después de terminada, quede contenida en un fichero de instalación. Esto sirve para agilizar su distribución y permite un control sobre las carpetas y archivos que conforman el software. De esta forma después de haber sido instalado, el traductor optimiza su funcionamiento, ya que los archivos están ordenados según el uso que necesita darles la aplicación. Esto repercute favorablemente a favor del rendimiento que tendrá la computadora que ejecute el software, y dará al usuario una mayor comodidad para trabajar con el sistema en general.

El fichero de instalación ha sido creado utilizando la herramienta InstallShield Express en su versión 3.5, que ofrece Borland adjunto a la versión 7 de Delphi, para crear ficheros de instalación a partir de cualquier aplicación Windows implementada en un lenguaje de programación determinado.

Primeramente se crea un proyecto, a través del cual se actualizan formularios que contienen la información que necesita InstallShield para crear el fichero de instalación. A continuación se muestran los formularios principales en los cuales se trabajó para completar el proyecto de construir el fichero de instalación.

Información general

Permite llenar la información general que va a tener el software, tales como nombre de la aplicación, versión, autor entre otras (véase Figura 2.9).

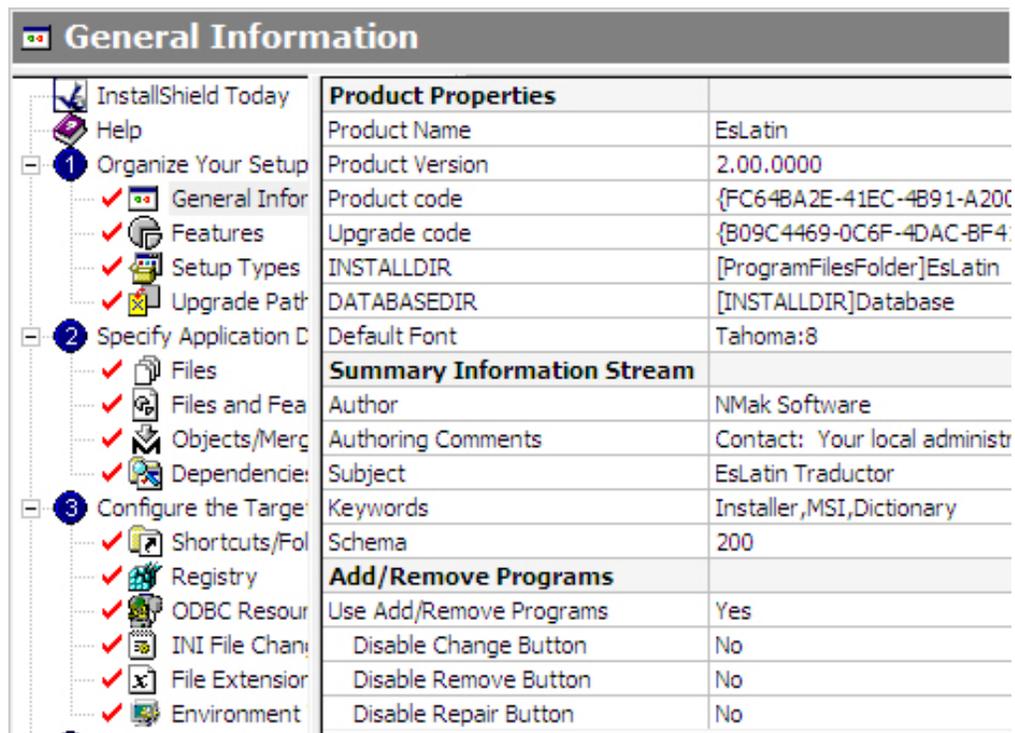


Figura 2.9 Información general.

Archivos

Se organizan los archivos del programa en carpetas, las cuales incluyen los correspondientes permisos para que los usuarios trabajen con el software y se protejan de las malas manipulaciones (véase Figura 2.10).

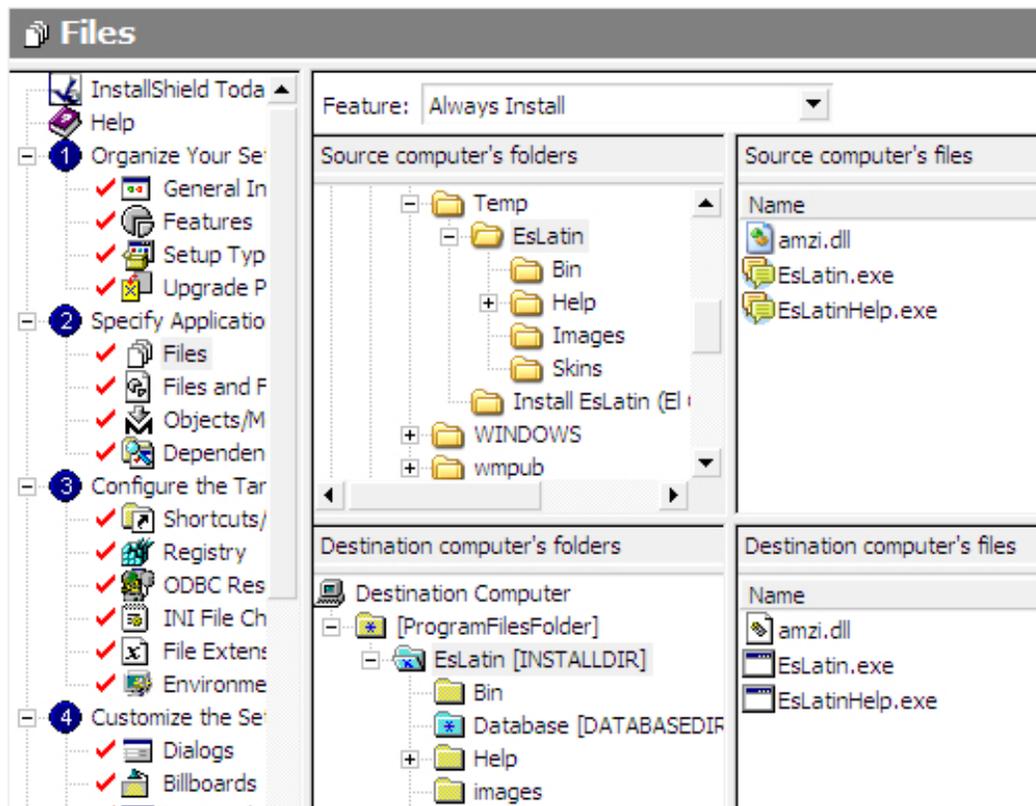


Figura 2.10 Archivos del programa.

Registro

Permite escribir la información correspondiente al contenido del registro de Windows, para trabajar con la contraseña de protección del software. También se otorgan los permisos necesarios a los usuarios para que la aplicación quede protegida (véase Figura 2.11).

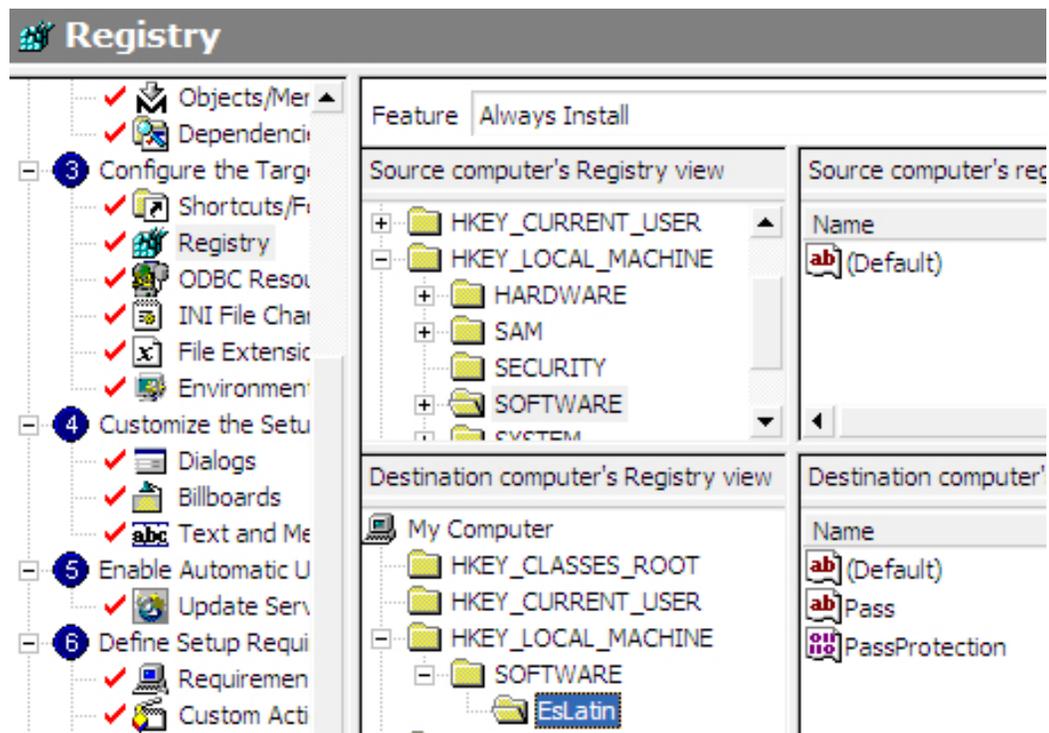


Figura 2.11 El registro de Windows.

Exportar el fichero de instalación

Por último se compila el proyecto y se salva el fichero de instalación. (véase Figura 2.12).

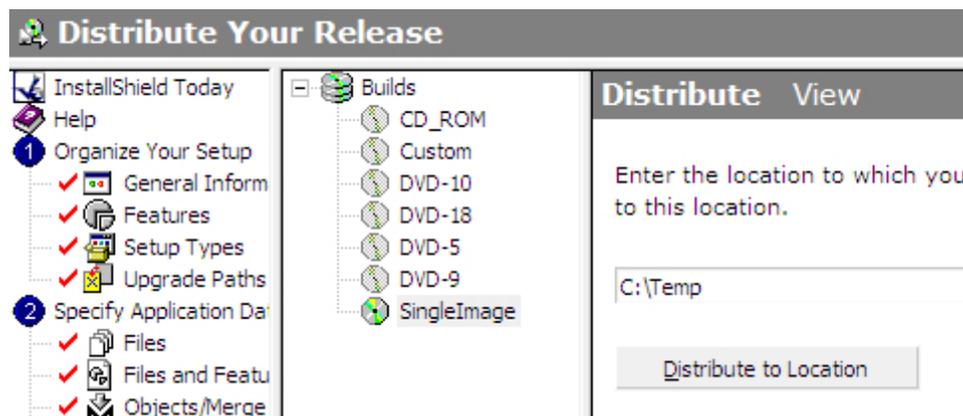


Figura 2.12 Salvar el fichero de instalación.

CAPÍTULO III. MANUAL DE USUARIO.

El traductor de español a latín para la nomenclatura botánica EsLatín versión 2.0, es el primero de su tipo desarrollado en la UCLV. En este capítulo se explican cuáles son los requisitos que deben cumplirse para su correcto uso, al realizar una traducción o para actualizar los datos del mismo. También se hace referencia a los pasos a seguir para la instalación del software.

3.1 Características del software.

Las principales cualidades que reúne el software son las siguientes:

- La interfaz con el usuario es sencilla y amigable, permitiendo que el experto se familiarice rápidamente con el uso de la herramienta. La ayuda del sistema está nutrida de las suficientes explicaciones para que el usuario comprenda cuáles son las definiciones de latín con las que va a tener que trabajar constantemente.
- Se incluye un módulo visual mediante el cual el experto puede comprobar la similitud con las características de su planta y así encontrar directamente la palabra o palabras correctas para su descripción.
- La ventana de actualización del diccionario cuenta con un buscador automático para encontrar fácilmente las palabras que se desean modificar.
- La aplicación cumple con los fines educativos para los que fue creada y sirve de ayuda para aprender varios conceptos del latín que son necesarios conocer para realizar una traducción.

3.2 Áreas de aplicación de la herramienta.

Puede ser utilizado por los especialistas del Jardín Botánico de la UCLV como una herramienta de ayuda en la nomenclatura. También servirá de ayuda a los profesionales

del área de educación que se encargan de impartir los cursos relacionados con el latín en sus facultades o en las esferas de la investigación donde sea necesario.

3.3 Requisitos del sistema.

Para la instalación del software y la utilización del mismo, son necesarios los siguientes requisitos mínimos:

- Procesador Intel 486/100 MHz o superior.
- 16 Mb de memoria RAM.
- Sistema operativo Windows NT o superior.
- Monitor SVGA.
- 30 Mega Bytes de espacio libre de disco duro.
- Macromedia Flash Player.

3.4 Pasos para la instalación del sistema.

El sistema inicialmente no puede ser utilizado hasta que sea instalado en una computadora, la cual deberá cumplir los requisitos planteados en el epígrafe anterior. El fichero para la instalación debe ser ejecutado en la computadora por un usuario con los privilegios necesarios para llevar a cabo dicha tarea. La guía para la instalación del sistema está escrita en idioma inglés. No obstante a continuación se describen los aspectos fundamentales de la instalación que puedan servir de ayuda en caso de dudas.

Primeramente la persona encargada de instalar el software debe asegurarse de que el mismo quede alojado en una carpeta donde los usuarios con menos privilegios tengan permiso de lectura. La guía del instalador ofrece un camino, que por defecto es donde el sistema operativo Windows instala sus programas. El camino es C:\Program Files para su versión en inglés, o C:\Archivos de Programas para su versión en español.

Después que se haya instalado EsLatín versión 2.0 satisfactoriamente, en el escritorio de la sesión de trabajo se copiará un icono de acceso directo al fichero ejecutable de la aplicación. Igualmente el software podrá ser ejecutado desde los Programas del Menú de inicio de Windows dando clic en un icono igual al anteriormente descrito.

3.5 Inicio de la sesión de trabajo.

La sesión de trabajo comienza presentando una imagen que muestra el nombre de la aplicación, donde además se observa la información de los procesos que van ocurriendo cuando el programa se está cargando, en este caso el usuario visualiza que el diccionario está siendo cargado.



Figura 3.1: Cargando diccionario.

A continuación se muestra una ventana que contiene el ambiente de trabajo a partir del cual puede trabajarse en la aplicación, haciendo uso de todas las funcionalidades de la misma. Dicho ambiente constituye el punto de partida de nuestro programa y por lo tanto se mantiene durante toda la ejecución de la aplicación. La figura siguiente muestra el ambiente de trabajo.

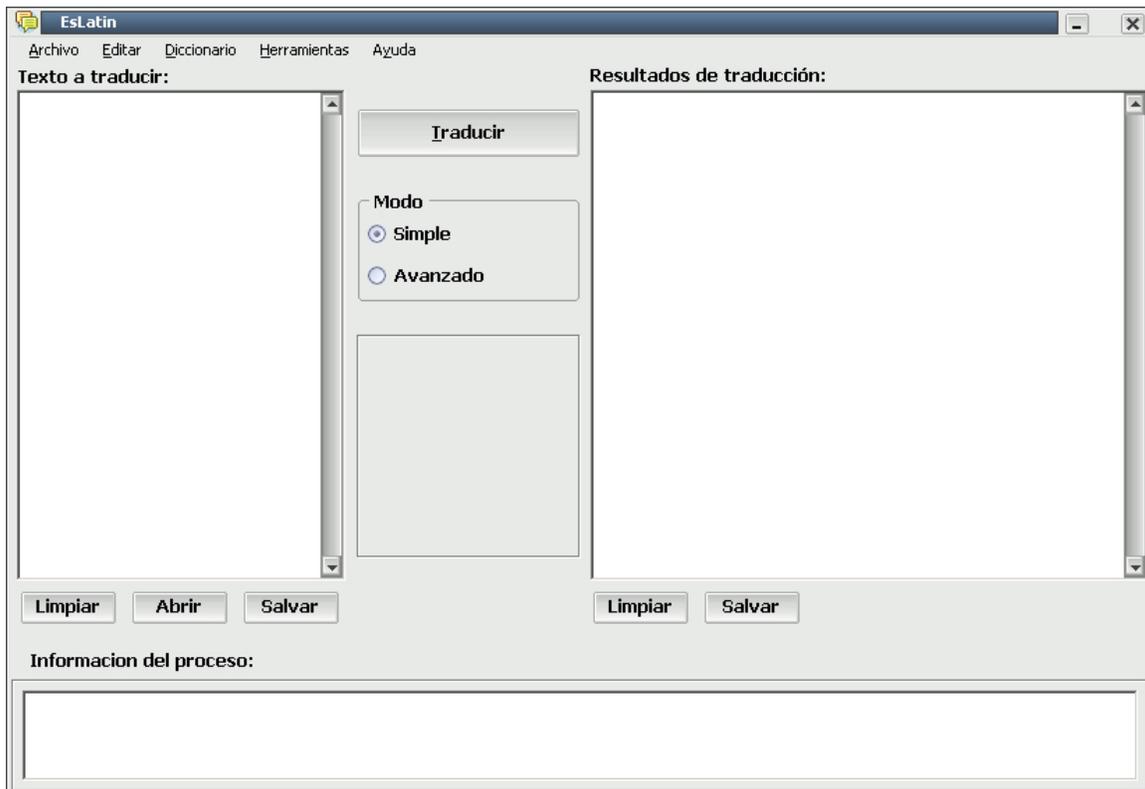


Figura 3.2: Ambiente de trabajo.

A partir de aquí la aplicación se encuentra lista para ser utilizada por el usuario.

En esta ventana principal se muestran dos editores de texto que servirán para realizar las traducciones. Encima de ambos se pueden observar sendas etiquetas de información que describen las acciones que pueden efectuarse en cada uno de ellos.

Texto a traducir

El primero de los editores, que se encuentra a la izquierda es donde el experto va a teclear la palabra o palabras que desea traducir.

Modo

Al centro puede observarse dos botones circulares que sirven para escoger el tipo de traducción que se desea realizar, en este caso, simple o avanzada.

Resultados de la traducción

El segundo editor, que se encuentra a la derecha, es el que va a mostrar los resultados de la traducción efectuada, o un mensaje de información que describa cualquier error que pudiera haber ocurrido durante el proceso.

Traducir

Al presionar este botón es cuando se efectúa el proceso de traducir lo que se encuentra escrito en el editor de la izquierda y aparecen los resultados en el editor de la derecha. Se puede ejecutar la misma acción al presionar la combinación de teclas Alt + t o también si se presiona F5.

En la parte inferior de ambos editores se muestran varios botones que sirven para trabajar con algunas opciones sobre los mismos.

Limpiar

Cuando se presiona este botón se limpia el editor de texto correspondiente al mismo. Se utiliza normalmente cuando se quiere realizar una nueva traducción.

Salvar

Al oprimir este botón se muestra un diálogo de Salvar, mediante el cual el usuario puede salvar hacia un fichero texto (.txt) el contenido del editor al cual pertenece. De esta

forma se ahorra tiempo, ya que de tratarse de una traducción extensa que hayan sido ejecutada con antelación no es necesario volver a teclearla.

Abrir

Cuando se da clic sobre este botón aparece un diálogo de Abrir, mediante el cual se puede cargar hacia el editor un fichero texto que el usuario haya guardado con antelación.

Por último y en la parte inferior de la ventana aparece un tercer editor que muestra la información del último proceso de traducción que haya sido efectuado. En el mismo se muestra un resumen de la cantidad de líneas traducidas, cuáles de estas fueron correctamente traducidas y cuántas palabras no aparecían en el diccionario.

3.6 El menú Archivo.

Al mismo se puede acceder dando clic sobre él o presionando la combinación de teclas Alt + a. El mismo ofrece una reducida cantidad de operaciones a realizar principalmente sobre los editores que muestra la ventana principal de la aplicación.

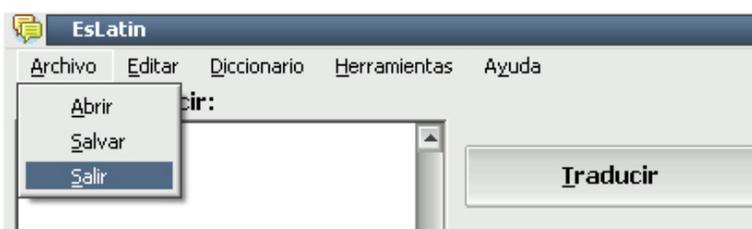


Figura 3.3: El menú Archivo.

Las opciones Abrir y Salvar son las mismas descritas anteriormente. La única diferencia radica en que se ejecutan sobre el editor donde se encuentre posicionado el cursor en ese momento.

La opción Salir finaliza le ejecución del programa. Antes de terminar se pregunta al usuario si realmente está seguro de abandonar la aplicación, mediante una pequeña ventana que permite confirmar la decisión de salida o permite seguir trabajando en el traductor.

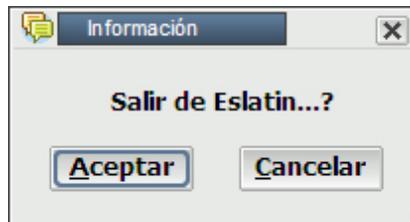


Figura 3.4: Ventana de confirmación de salida.

3.7 El menú Editar.

A este menú se puede acceder presionando sobre él o a través de la combinación de teclas Alt + e. Muestra una serie de acciones que se pueden realizar sobre los editores y que son clásicas de cualquier aplicación de Windows que trabaja con textos. Incluye además las opciones de realizar una traducción y limpiar el contenido de un editor. Dichas acciones han sido anteriormente descritas.

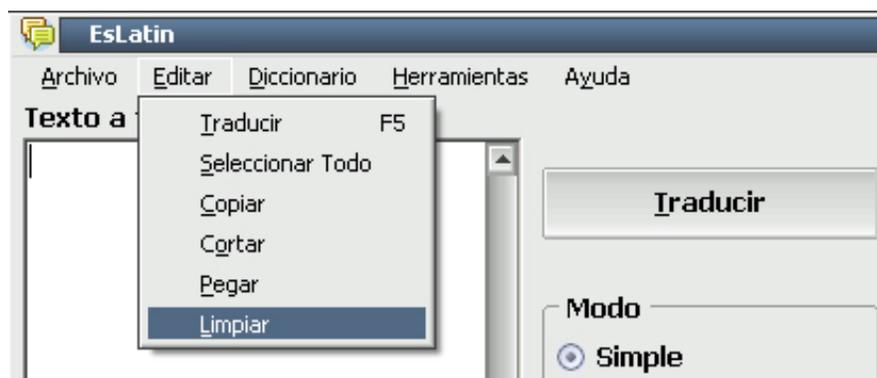


Figura 3.5: El menú Editar.

3.8 El menú Diccionario.

Se puede acceder a esta opción dando clic sobre el menú Diccionario o a través de la combinación de teclas Alt + d. Está formado de dos herramientas:

Visualizar diccionario

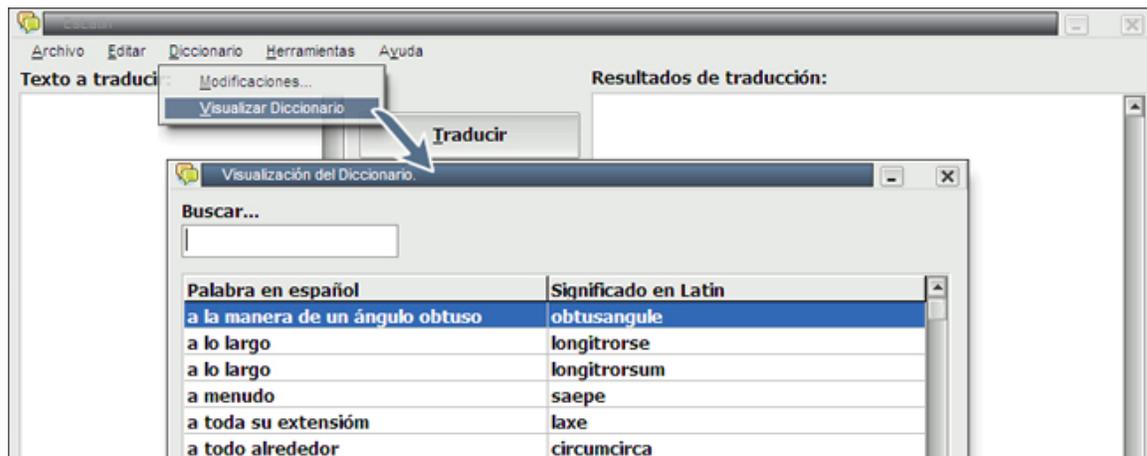


Figura 3.6: Visualizar diccionario.

Con esta opción el usuario puede tener la información de todas las palabras que contiene el diccionario, pero no puede modificarlas. Puede realizar una búsqueda para encontrar una palabra en particular o varias. Las palabras se encuentran en orden alfabético.

Modificaciones al diccionario

Esta opción está diseñada e implementada para que una persona especialista en el idioma latín pueda realizar las modificaciones que estime conveniente sobre el diccionario de la aplicación. Se puede acceder al menú haciendo clic sobre el mismo o mediante la combinación de teclas Alt + d → m.

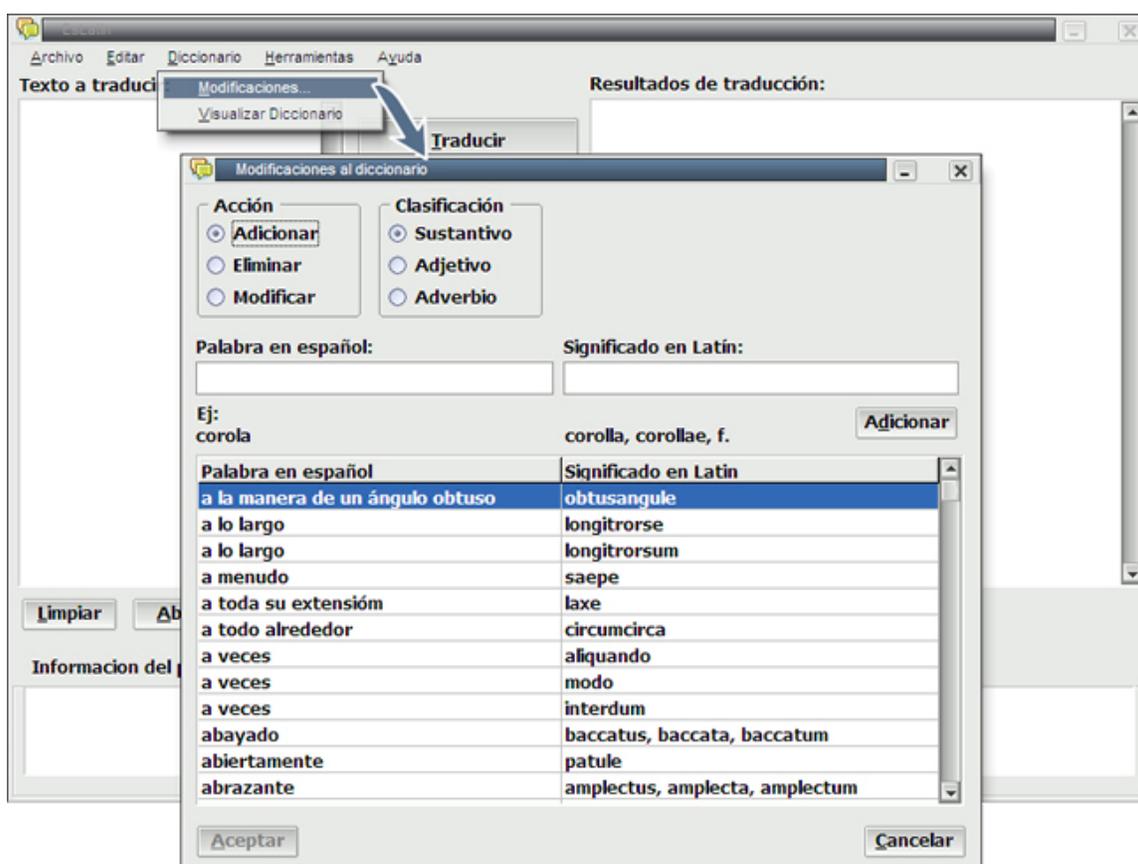


Figura 3.7: Modificaciones al diccionario.

Las opciones que ofrece esta ventana son las de adicionar, eliminar o modificar cualquier palabra que se encuentra en el diccionario. Cada una de ellas se acciona presionando sobre los botones redondos que aparecen en un grupo nombrado “Acción” que se encuentra en la parte superior izquierda de la ventana. Cada uno de estos botones hace que cambie lo que se observa en la misma con el objetivo de que el usuario comprenda mejor como realizar los pasos para actualizar el diccionario.

Adicionar

Cuando se desea agregar una nueva palabra se muestra un segundo grupo de botones circulares nombrado “Clasificación”, situado a la derecha del primero. En ellos se puede observar los tres tipos de palabras con las que trabaja el diccionario, en este caso: sustantivos, adjetivos y adverbios. Si se adiciona un sustantivo la ventana se mantiene

igual que en la figura 3.5. Allí aparecen dos editores de texto. Sobre ellos sus respectivas etiquetas de información que indican que en el primero se escribe el sustantivo en español, y en el de la derecha la traducción simple en latín. Debajo de los editores se pueden ver otras dos etiquetas de información que indican un ejemplo de como debe ser la sintaxis de las palabras que se escriban para evitar errores. Hacia el medio de la ventana, a la derecha y un poco por debajo de los editores aparece un botón que se nombra “Adicionar”. Cuando se presiona sobre él se actualiza una tabla que se encuentra en la parte inferior de la ventana, y que contiene la lista entera de palabras del diccionario. Esta tabla muestra la información de las actualizaciones que se están realizando, para que el usuario decida cuando está listo para copiarlas hacia el diccionario.

Cuando se quiere agregar un adjetivo, la ventana cambia su forma como se muestra a continuación.

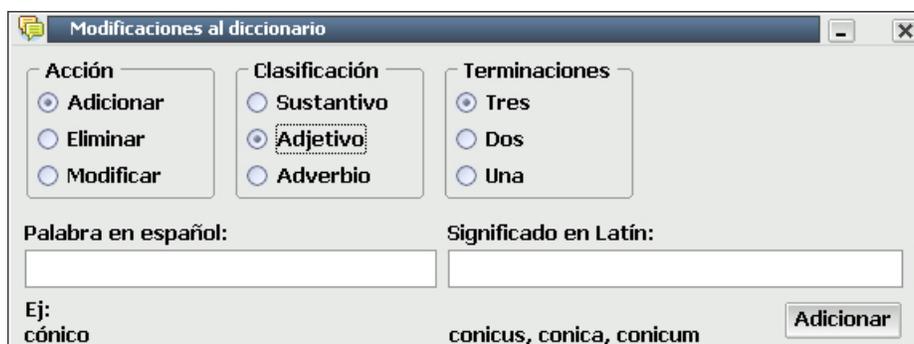


Figura 3.8: Adicionar un adjetivo.

Como se observa, aparece un nuevo grupo de botones circulares que indican el tipo de adjetivo que se desea adicionar, o sea, si es de tres, dos o una terminación. Para cada uno de ellos las etiquetas de información que aparecen debajo de los editores indican cómo debe ser la sintaxis de las palabras que se desean agregar al diccionario. Igual que en el caso anterior las nuevas palabras aparecerán en la tabla que se muestra en la ventana al presionar sobre el botón “Adicionar”.

En el último caso si se desea adicionar un adverbio, la ventana cambia su apariencia como se muestra en la siguiente figura.

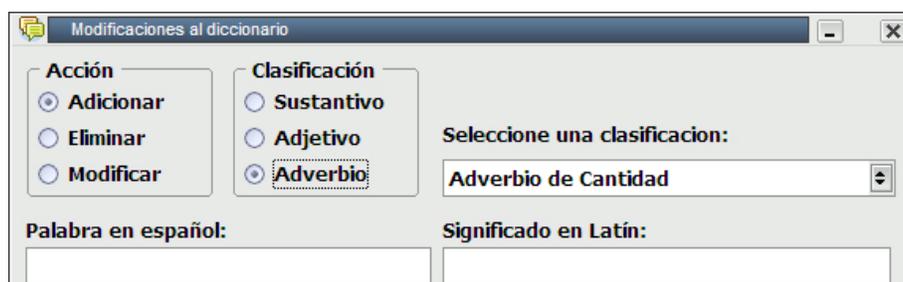


Figura 3.9: Adicionar un adverbio.

Los cambios que se observan es que ha aparecido un editor desplegable, en el que se muestran la clasificación del adverbio que se desea adicionar. Igualmente las nuevas palabras se actualizarán hacia la tabla cuando se presione el botón “Adicionar”.

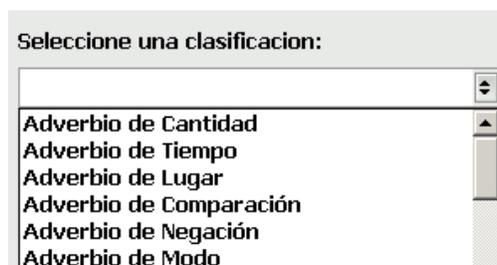


Figura 3.10: Editor desplegable de clasificación de los adverbios.

Eliminar

Cuando se necesita eliminar una o varias palabras del diccionario, o de las que se hayan ido agregando hacia la tabla de la ventana se presiona el botón circular del grupo “Acción”. La ventana se mostrará como a continuación.



Figura 3.11: Eliminar una palabra.

Aparece otro editor de texto similar a los anteriores y el botón “Adicionar” cambia su nombre por “Eliminar”. El editor sirve para ir tecleando la palabra que se desea eliminar y a medida que esto ocurre en la tabla se muestran los resultados de la búsqueda en el diccionario de lo escrito hasta ese momento. Cuando se localice la palabra que se desea eliminar, entonces se presiona sobre el botón “Eliminar”, y la palabra desaparece de la tabla, para que luego, si el usuario está convencido, las borre del diccionario.

Modificar

Para modificar una palabra, la ventana se muestra igual que en la figura 3.9, con el único cambio de que el botón “Eliminar” cambia su nombre por “Modificar”. Podemos realizar una búsqueda, igual que en el caso anterior, de la palabra que se desea modificar. Después de seleccionada la palabra en la tabla basta con dar doble clic sobre ella o presionar el botón “Modificar”. En ese caso aparecerá una nueva ventana como se muestra a continuación.

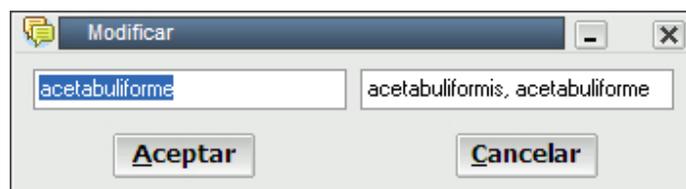


Figura 3.12: Modificar una palabra.

En la misma se puede modificar tanto la palabra en español como su significado en latín. Si se está seguro de que la palabra está correctamente escrita se presiona el botón “Aceptar” (véase figura 3.10) y las palabras se actualizan en la tabla para que el usuario luego las envíe hacia el diccionario.

Después de haber realizado todos los cambios necesarios sobre la tabla que muestra la ventana “Modificaciones al diccionario”, el usuario tiene una idea clara de cuáles son las palabras que va a tener el diccionario de la aplicación a partir de ese momento. Si por alguna razón hubiera algún error y el usuario cree que no debe realizar las modificaciones sobre las que ha estado trabajando entonces solo tiene que presionar el botón “Cancelar” de la ventana. Por el contrario si decide guardar hacia el diccionario todos los cambios realizados entonces deberá presionar “Aceptar”.

3.9 El menú Herramientas.

Para acceder a este menú basta con hacer clic sobre él o presionar la combinación de teclas Atl + h. El mismo muestra las herramientas de la aplicación con las cuáles el usuario puede trabajar. La figura que se muestra a continuación describe con más claridad las acciones que se pueden ejecutar a través de este menú.

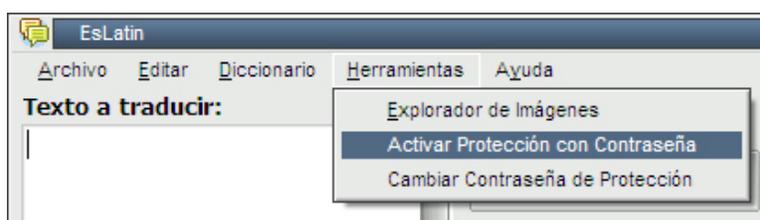


Figura 3.13: El menú Herramientas.

Como primera opción del menú herramientas se encuentra el explorador de imágenes. La utilidad del mismo ha sido descrita con anterioridad. A continuación se muestra cómo trabajar con él.

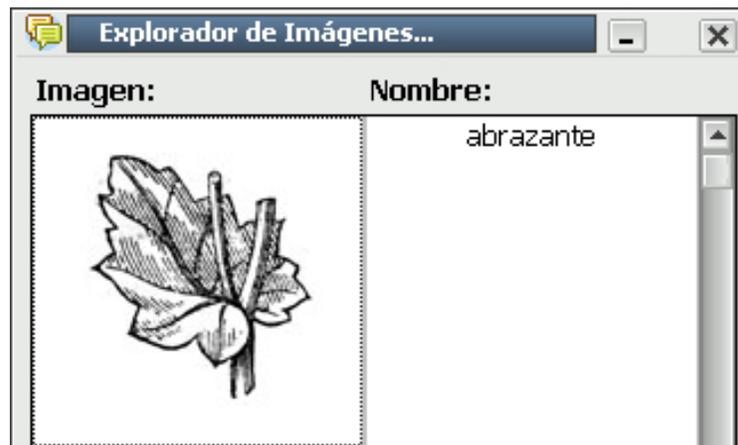


Figura 3.14: Explorador de imágenes.

Tal y cómo se observa la ventana que contiene al explorador de imágenes está formada de dos etiquetas de información que describen el contenido de cada columna de una tabla que aparece en la parte inferior de los mismos. La tabla consta de dos columnas. La primera de ellas, que se encuentra a la izquierda, contiene las imágenes que muestran una o varias características de las plantas. En la columna derecha se muestra el nombre en español, que describe el rasgo más distintivo de la parte de las plantas que se observa. Para comodidad del usuario la tabla puede ser recorrida rápidamente auxiliándose de una barra de desplazamiento que se encuentra en el extremo derecho de la misma. Cuando se hace doble clic sobre cualquier fila de la tabla aparece una nueva ventana con la traducción simple de la palabra en español.

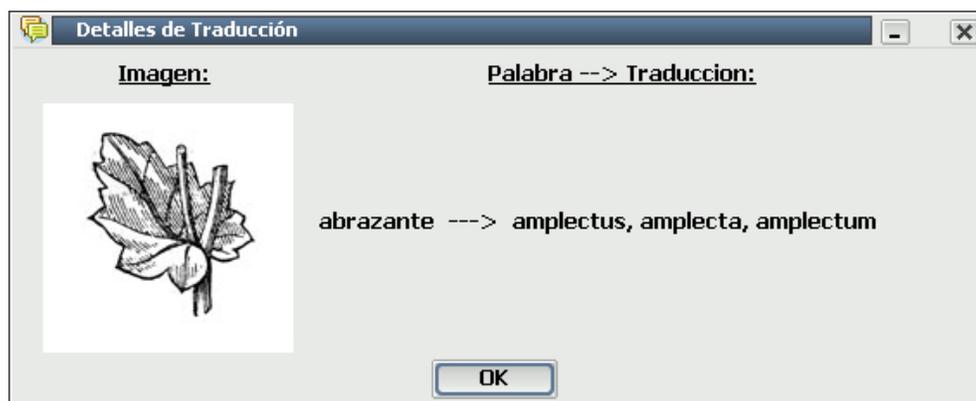


Figura 3.15: Detalles de traducción.

La otra utilidad del menú Herramientas se relaciona con la seguridad que se le ha dado al software para proteger al diccionario de malas manipulaciones por usuarios que no sean expertos en latín. Cuando se escoge la opción “Activar protección con contraseña” se muestra una nueva ventana donde el usuario debe teclear una contraseña o rectificar una antigua si no es la primera vez que quiere proteger el diccionario. La siguiente figura muestra la ventana para trabajar con la contraseña.

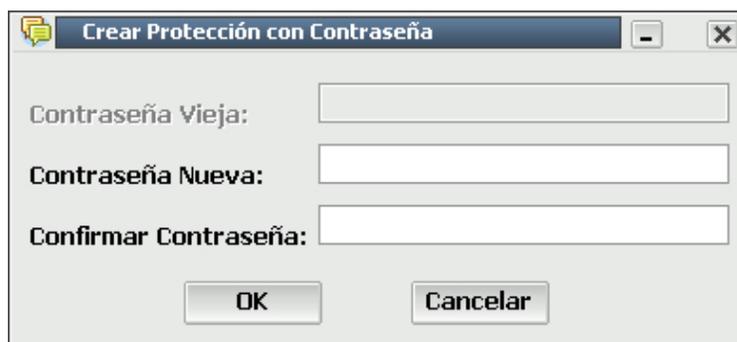


Figura 3.16: Crear protección con contraseña.

Como puede observarse el editor de la vieja contraseña está deshabilitado cuando se quiere proteger el diccionario por primera vez. Es necesario confirmar la contraseña para evitar cualquier error que pudiera ocurrir, de esta manera el usuario comprueba que lo que está tecleando es correcto.

Una última opción del menú Herramientas se refiere al cambio de la contraseña de protección de la aplicación. Se puede utilizar cuando el encargado de mantener la actualización del diccionario no se encuentra a gusto con la contraseña actual y quiere sustituirla por otra. En este caso se muestra una ventana igual a la de la figura 3.14, con la diferencia que el editor de la contraseña vieja se habilita, y lo que teclee el usuario allí debe coincidir con la contraseña anterior para que luego pueda ser cambiada por la nueva, que igualmente tiene que ser confirmada.

Por último es necesario aclarar que desde el momento que se haya activado la protección con contraseña, tanto para acceder a la ventana de actualización del diccionario como para desactivar esta opción se deberá confirmar la contraseña actual a través de la siguiente ventana:



Figura 3.17: Confirmar contraseña de protección.

3.10 El menú Ayuda.

La figura que se muestra a continuación indica cómo se accede a la ayuda del traductor. A través de ella se explican todas las funcionalidades de la aplicación.

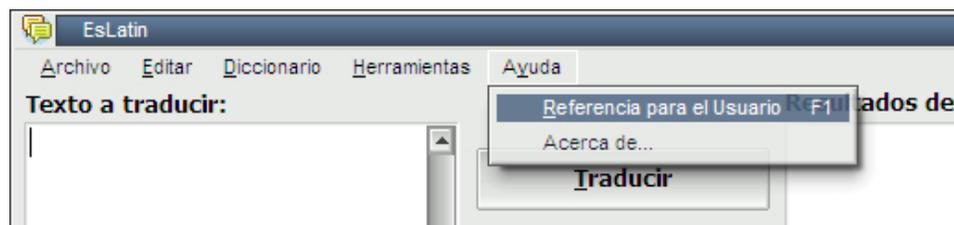


Figura 3.18: El menú Ayuda.

Referencia para el usuario

En la figura 3.16 muestra cómo acceder a la referencia para el usuario. La misma está implementada en RoboHelp y se observa en forma de página web. Dicha página contiene elementos interactivos hechos en Macromedia Flash, que facilitan al usuario la navegación a través de la referencia. Los tópicos de la referencia son en forma de hipertexto, lo cual significa que el usuario puede revisar la referencia de forma no secuencial. A continuación se explica el contenido de la Referencia para el usuario.

Contenido

Muestra toda la información necesaria para que el usuario sepa como está diseñado el traductor y aprenda a trabajar con él. Se divide en cinco partes:

- El traductor: Se explica cuáles son las opciones que ofrece la aplicación y cómo utilizarlas.
- El diccionario: Herramienta con la que cuenta la aplicación, imprescindible para su correcto funcionamiento. Se abordan los temas referentes al uso del diccionario y cómo actualizarlo.
- El explorador de imágenes: Con anterioridad se ha tratado el tema referente al módulo visual del software. En este tópico se explica el funcionamiento del explorador de imágenes y cómo hacer uso de él.
- La seguridad del sistema: Tema también abordado anteriormente. Se explica cómo trabajar con el módulo de seguridad del traductor.
- Opciones: Explica las comodidades del software para guardar, cargar o limpiar los textos con los que trabaja la aplicación.

Temas básicos del latín

Aquí la ayuda del sistema presenta un glosario (véase anexo) donde el usuario puede conocer los conceptos básicos del latín con los que se ha trabajado durante la realización del software y que sirven para conocer mejor cómo se trabaja con el traductor.

Acerca de...

Muestra al usuario breves aspectos como son: nombre del sistema, versión, año e institución donde fue realizada.



Conclusiones

Las ventajas que ofrece la Inteligencia Artificial en la traducción asistida por computadora y la amplia experiencia en la enseñanza del latín por el departamento de Letras de la UCLV, permitieron implementar un software que corrige las deficiencias de su versión anterior y mejora las prestaciones y utilidades de su predecesor. Facilita además el proceso de enseñanza-aprendizaje del latín para aquellas personas que constantemente hacen uso de él y que no son expertas en la materia. La aplicación, en su ayuda al usuario, provee los conceptos y definiciones asociados al latín, los cuales sirven de apoyo al utilizar el traductor.

Se logró que la interfaz con el usuario fuese sencilla y amigable, permitiendo la rápida familiarización con el uso de la herramienta. Se incluyó además un módulo visual que sirve al experto para agilizar la búsqueda de la palabra con la cual desea nombrar su descubrimiento.

La aplicación quedó provista de un diccionario con las traducciones de las palabras con las que trabajan los botánicos en la nomenclatura, tales como sustantivos, adjetivos y adverbios. El mismo puede ser modificado con facilidad por una persona que sea experta en latín.

Se logró construir una aplicación que utiliza dos paradigmas de la programación: la declarativa a través del lenguaje Prolog y la orientada a objeto con Object Pascal. Se consiguió satisfactoriamente enlazar Amzi Prolog con Borland Delphi.

Recomendaciones

Dar continuidad al software EsLatín de manera que se implementen nuevas funcionalidades para la consecución de una meta más elevada: la traducción automatizada de oraciones completas, que en forma yuxtapuesta conformarían una taxonomía.

Desarrollar estudios sobre el enfoque del uso de los traductores automáticos en instituciones educativas como herramienta de ayuda a los profesionales y estudiantes de las lenguas extranjeras.

Poner a disposición de instituciones científicas vinculadas a la Botánica el presente trabajo para su uso y distribución, así como para realizar una correcta evaluación del mismo.

Referencias bibliográficas

- (2006) "¿Qué es Procesamiento del Lenguaje Natural?" [En línea]. Disponible en:
<http://www.ampln.org/NLP.htm> [Consultado el 28 de marzo de 2006].
- BARR, A. & FEIGENBAUM, E. A. (1981) *The Handbook of Artificial Intelligence*.
- BELLO, R. E., ET AL. (2002) Cap 4 Sistemas Basados en Reglas. *Aplicaciones de la Inteligencia Artificial*. Jalisco, México, Universidad de Guadalajara.
- DOMÍNGUEZ BURGOS, A. (2002) *Lingüística computacional: un esbozo*. Universidad Central de Venezuela.
- GARCÍA VALDIVIA, Z., ET AL. (2000) *Introducción a la Inteligencia Artificial*, México, Editorial Pandora.
- LEZCANO, M. (2000) *Prolog y los sistemas expertos*, México, Universidad de Guadalajara. Centro Universitario de la Ciénaga.
- MIGUEL VERGÉS, J. (2001) "La estación de trabajo del traductor". [En línea].
Disponible en: <http://webs.uvigo.es/h06/web573/persoal/jmv/07-bibliografia/a-08/a-08.htm> [Consultado el 9 de abril de 2006].

Bibliografía

- (1981) *Revista del Jardín Botánico Nacional*, Vol. II, Nro. 1, Universidad de La Habana.
- (1983) *Revista del Jardín Botánico Nacional*, Vol. IV, Nro. 2, Universidad de La Habana.
- (1987) *Acta Botanica Hungarica* (en latín e inglés), Vol. 33, Nro. 1-2, Budapest, Academia Kiadó.
- BRATKO, I. (1986) *Prolog Programming for Artificial Intelligence*, London, Addison Wesley.
- CARBONELL, P. (2003) “Aportaciones de la Inteligencia Artificial a la teoría de la traducción”. [En línea]. Disponible en: <http://ctrl.poly.edu/visitors/pjcarbon/pdf/TRADUC97.pdf> [Consultado el 8 de abril de 2006].
- FERNÁNDEZ CHAMIZO, C. (2005) “Tratamiento del lenguaje natural. Universidad Complutense de Madrid”. [En línea]. Disponible en: <http://www.fdi.ucm.es/profesor/carmen/iaic/LenguajeNatural.pdf> [Consultado el 8 de abril de 2006].
- FONT QUER, P. (1967) *Diccionario de botánica*, Edición Revolucionaria, La Habana, Cuba, Instituto Cubano del Libro.
- GARCÍA DE DIEGO, V. (1971) *Diccionario ilustrado latino-español, español-latino*, Barcelona, SPES.
- URBAN, I. (1926) *Plantae Haitienses novae vel rariores* (en latín y alemán), band 17, No. 7, Berlín, Londres, París, Arkiv för botanik.

Willdenowia 32, (2002) *Novitiae Florae Cubensis*, Nro. 6.

Willdenowia 33, (2003) *Novitiae Florae Cubensis*, Nro. 9.

Willdenowia 34, (2004) *Novitiae Florae Cubensis*, Nro. 15.

Willdenowia 34, (2004) *Novitiae Florae Cubensis*, Nro. 18.

Anexo 1 Glosario.

Adjetivo. Palabra cuya función propia es la de modificar al sustantivo —con el que concuerda en género y número—, bien directamente: casa pequeña; magníficas vistas; aquel avión; bien a través de un verbo, caso en el que el adjetivo funciona como atributo (→ atributo) o como predicativo (→ predicativo): La casa es pequeña; Los niños comen tranquilos.

Adverbio. Palabra invariable cuya función propia es la de complementar a un verbo (Hablaba pausadamente), a un adjetivo (menos interesante) o a otro adverbio (bastante lejos, aquí cerca); también puede incidir sobre grupos nominales (solamente los jueves), preposicionales (incluso sin tu ayuda) o sobre toda una oración (Desgraciadamente, no pudo llegar a tiempo). Aportan significados muy diversos: lugar (aquí, cerca, dónde), tiempo (hoy, luego, recién, cuándo), modo (así, bien, cortésmente, cómo), negación (no, tampoco), afirmación (sí, efectivamente), duda (quizá, posiblemente), deseo (ojalá), cantidad o grado (mucho, casi, más, cuánto), inclusión o exclusión (incluso, inclusive, exclusive, salvo, excepto, menos), oposición (sin embargo, no obstante) u orden (primeramente), entre otras nociones.

Caso. Cada una de las formas que, en las lenguas que tienen declinación (→ declinación), adoptan determinadas clases de palabras, como el nombre o el pronombre, según la función sintáctica que deban desempeñar.

Declinación. Conjunto de las diferentes formas que en ciertas lenguas, por ejemplo el latín, adoptan determinadas clases de palabras, como el nombre o el pronombre, según la función sintáctica que deban desempeñar en la oración.

Preposición. Palabra invariable y átona (excepto según) cuya función consiste en introducir un sustantivo o un grupo nominal (llamado término de preposición) con el que forma un complemento que depende sintácticamente de otro elemento del enunciado. En el español actual son las siguientes: a, ante, bajo, con, contra, de, desde, durante, en, entre, hacia, hasta, mediante, para, por, según, sin, sobre, tras. También son preposiciones, de uso más restringido, pro (asociación pro derechos humanos) y vía (Voló a Miami vía Londres).

Sustantivo. Palabra con género inherente que designa personas, animales o cosas y es capaz de funcionar como núcleo del sujeto (→ sujeto). Equivale a nombre (→ nombre).

Régimen. Hecho de regir o exigir una palabra otra u otras para su correcta construcción dentro de una oración. Así, se habla de régimen transitivo de un verbo si se construye con un complemento directo, o de régimen intransitivo si no lo lleva; y cuando se habla del régimen preposicional de un verbo, de un adjetivo o de un sustantivo, se está haciendo referencia a la preposición o las preposiciones con las que se construye su complemento.

Género. Rasgo inherente a los sustantivos por el que estos se dividen, en español, en masculinos y femeninos. Adoptan también el género, a través de la concordancia, los determinantes y adjetivos que los acompañan o los pronombres que los sustituyen. El artículo y algunos pronombres también tienen género neutro (→ neutro), como las formas lo, esto, eso, aquello.

Número. Variación que experimentan los sustantivos y las palabras que conciertan con ellos para expresar, mediante ciertas terminaciones, si se refieren a una sola entidad o a más de una.

Complemento. Palabra o grupo de palabras que depende sintácticamente de otro elemento de la oración.

Complemento circunstancial. Complemento del verbo no exigido por el significado de este y que expresa las circunstancias de lugar, tiempo, modo, instrumento, medio, causa, finalidad, cantidad, etc., relacionadas con la acción verbal: Trabajo en un banco; Amanece a las cinco; Llovía intensamente; Cavé la zanja con una pala; Te llamaré por teléfono; Ahorro para las vacaciones.

Concordancia. Conformidad de accidentes entre dos o más palabras variables. Todas estas, menos el verbo, concuerdan en género y número; y el verbo con su sujeto, en número y persona.