

Universidad Carlos III de Madrid

Escuela Politécnica Superior



VocAssist (De-Es)

Proyecto Fin de Carrera

Autor: **José Montero Castillo**
Tutora: **Ana María Iglesias Maqueda**
Fecha: **Septiembre 2010**

Tabla de Contenido

1. Introducción.....	10
1.1. Motivación.....	10
1.2. Objetivos.....	11
1.3. Estructura del Documento	12
2. Gestión del Proyecto.....	14
2.1. Gestión Software.....	14
2.1.1. Metodología de Desarrollo.....	14
2.1.2. Ciclo de Vida	15
2.2. Planificación Inicial	17
3. Estado del Arte	19
4. Análisis.....	23
4.1. Captura de Requisitos	23
4.2. Requisitos de Usuario.....	24
4.2.1. Funcionalidades del Sistema	25
4.2.2. Restricciones del Sistema	26
4.2.3. Usuarios del Sistema.....	26
4.2.4. Entorno Operacional.....	27
4.2.5. Especificación de Requisitos	27
4.3. Requisitos de Software.....	27
4.3.1. Especificación de Requisitos	27
4.3.1.1. Requisitos Funcionales	29
4.3.1.2. Requisitos No-Funcionales	43
5. Diseño	46
5.1. Base de Datos.....	46
5.1.1. Esquema Conceptual.....	47
5.1.1.1. Diagrama E/R.....	48
5.1.1.2. Supuestos Semánticos.....	49

5.1.2. Esquema Lógico	52
5.1.2.1. Grafo Relacional.....	53
5.1.2.2. Dominios.....	54
5.1.2.3. Comprobaciones	55
5.1.2.4. Aserciones.....	56
5.1.2.5. Disparadores	57
5.2. Aplicación Software	57
5.2.1. Contexto	57
5.2.2. Arquitectura	58
5.2.3. Diseño de Componentes	60
5.2.3.1. Diagrama de Componentes	60
5.2.3.2. Responsabilidades	63
5.2.3.2.1. Responsabilidades Generales	63
5.2.3.2.2. Responsabilidades Específicas	65
5.2.3.2.3. Matriz de Rastreabilidad: CO - RS	66
6. Implementación y Pruebas	68
6.1. Base de datos	68
6.1.1. Selección de SGBD	68
6.1.2. Implementación y Pruebas.....	71
6.2. Aplicación Software	74
6.2.1. Selección del Lenguaje de Programación.....	74
6.2.2. Estructuración en Incrementos	74
6.2.3. Implementación de la Arquitectura	75
6.2.4. Implementación del Sistema	76
6.2.5. Pruebas	77
7. Planificación Final y Análisis de Costes.....	79
7.1. Planificación Final	79
7.2. Análisis de Costes.....	81

7.2.1. Recursos Humanos	81
7.2.2. Equipamiento	82
7.2.3. Fungibles	82
7.2.4. Otros	83
7.2.5. Coste Total	83
8. Conclusiones y Trabajos Futuros	84
8.1. Conclusiones.....	84
8.2. Trabajos Futuros	85
9. Glosario de Términos y Acrónimos.....	86
10. Bibliografía	91
Anexo 1 – Pruebas de Usabilidad.....	92
Anexo 2 – Manual de Usuario	94
Instalación	94
Ejecución y Salida de la Aplicación	94
Gestión de Diccionarios de Usuarios	96
Gestión de Términos	97
Añadir Términos	97
Editar Términos	98
Eliminar Términos	99
Visualizar/Buscar Términos	100
Búsqueda Avanzada	101
Buscar Términos	101
Generar Listados PDF	102
Búsqueda Online.....	104
Buscar Término	104
Añadir Término Online	105
Gestión de Ejercicios	106
Generar Ejercicios	106

Resolver, Corregir y Solucionar Ejercicios.....	107
Abrir y Guardar Ejercicios.....	108
Gestión de Opciones.....	109
Cambiar Idioma de la Interfaz	109
Cambiar Nombre del Usuario	109
Cuadro de Diálogo Añadir/Editar Término.....	110

Índice de Ilustraciones

Ilustración 1 – Ciclo de Vida Incremental.....	16
Ilustración 2 – Panificación Inicial.....	18
Ilustración 3 – Diagrama E/R.....	48
Ilustración 4 – Grafo Relacional.....	53
Ilustración 5 – Diagrama de Contexto.....	58
Ilustración 6 – Arquitectura en 3 Capas.....	58
Ilustración 7 – Arquitectura MVC.....	59
Ilustración 8 – Diagrama de Componentes.....	62
Ilustración 9 – Tiempos de Consultas SQL Principales.....	73
Ilustración 10 – Panificación Final.....	80
Ilustración 11 – Ventana Principal.....	94
Ilustración 12 – Salir de la Aplicación.....	95
Ilustración 13 – Menú Abrir Diccionario de Usuario.....	96
Ilustración 14 – Diccionario de Usuario Abierto.....	96
Ilustración 15 – Añadir Término.....	97
Ilustración 16 – Editar Término.....	98
Ilustración 17 – Eliminar Término.....	99
Ilustración 18 – Visualizar/Buscar Términos.....	100
Ilustración 19 – Búsqueda Avanzada de Términos.....	101
Ilustración 20 – Generar Listado PDF.....	102
Ilustración 21 – Listado PDF Generado.....	103
Ilustración 22 – Búsqueda Online de Término.....	104
Ilustración 23 – Añadir Término Online.....	105
Ilustración 24 – Generar Ejercicio.....	106

Ilustración 25 – Solucionar Ejercicio.....	107
Ilustración 26 – Menú Cambiar Idioma	109
Ilustración 27 – Menú Cambiar Nombre de Usuario.....	109
Ilustración 28 – Cambiar Nombre de Usuario	109
Ilustración 29 – Cuadro de Diálogo Añadir/Editar Término	110

Índice de Tablas

Tabla 1 – Ejemplo de un Requisito de Software	28
Tabla 2 – RSF-01: Crear Diccionario	29
Tabla 3 – RSF-02: Abrir Diccionario	30
Tabla 4 – RSF-03: Definición de Término	32
Tabla 5 – RSF-04: Añadir Términos	32
Tabla 6 – RSF-05: Editar Términos	33
Tabla 7 – RSF-06: Eliminar Términos	33
Tabla 8 – RSF-07: Añadir Etiquetas Organizativas	34
Tabla 9 – RSF-08: Eliminar Etiquetas Organizativas	34
Tabla 10 – RSF-09: Añadir Marca “Estudiar”	35
Tabla 11 – RSF-10: Eliminar Marca “Estudiar”	35
Tabla 12 – RSF-11: Asignar Nombre de Usuario	36
Tabla 13 – RSF-12: Búsqueda Local de Términos.....	36
Tabla 14 – RSF-13: Búsqueda Online de Términos.....	37
Tabla 15 – RSF-14: Búsqueda Avanzada de Términos	37
Tabla 16 – RSF-15: Añadir Términos Automáticamente	38
Tabla 17 – RSF-16: Editar Términos Automáticamente	38
Tabla 18 – RSF-17: Generar Listados de Vocabulario	39
Tabla 19 – RSF-18: Generar Ejercicios – Significados	39
Tabla 20 – RSF-19: Generar Ejercicios – Artículos	39
Tabla 21 – RSF-20: Generar Ejercicios – Plurales.....	40
Tabla 22 – RSF-21: Generar Ejercicios – Verbos.....	40
Tabla 23 – RSF-22: Responder Ejercicios.....	41
Tabla 24 – RSF-23: Corregir Ejercicios	41

Tabla 25 – RSF-24: Solucionar Ejercicios	42
Tabla 26 – RSF-25: Guardar Ejercicios	42
Tabla 27 – RSF-26: Abrir Ejercicios.....	43
Tabla 28 – RSNF-01: Interfaz Gráfica.....	43
Tabla 29 – RSNF-02: Idioma de la Interfaz	44
Tabla 30 – RSNF-03: Multiplataforma.....	44
Tabla 31 – RSNF-04: Sin Instalación	44
Tabla 32 – RSNF-05: Volumen de Datos	45
Tabla 33 – RSNF-06: Futuras Ampliaciones.....	45
Tabla 34 – Matriz de Rastreabilidad: CO –RS	67
Tabla 35 – Comparativa de SGBD	70
Tabla 36 – Tamaño y Organización del Código.....	77
Tabla 37 – Costes en Recursos Humanos.....	81
Tabla 38 – Costes en Equipamiento	82
Tabla 39 – Costes en Fungibles	82
Tabla 40 – Otros Costes	83
Tabla 41 – Coste Total	83
Tabla 42 – Glosario de Términos y Acrónimos	90
Tabla 43 – Evaluación de Prueba de Usabilidad	93

1. Introducción

En esta sección se ofrece la motivación que ha originado este proyecto, una especificación de los objetivos que se desea conseguir y una descripción de la estructura de este documento.

1.1. Motivación

Hoy en día, el aprendizaje de idiomas es una tarea muy importante, especialmente estando inmersos en un proceso de globalización.

Aprender un idioma es un proceso lento y requiere un gran esfuerzo por parte del estudiante. Son muchos los aspectos que hay que estudiar para dominar un idioma, siendo la gramática y el vocabulario la base de todos ellos.

Viviendo en la era de la electrónica, todo el mundo tiende a pensar que existen aplicaciones para hacer cualquier cosa, incluido facilitar el proceso de aprendizaje de idiomas; sin embargo, la realidad no es así. Es cierto que existen muchos cursos (online, presenciales, interactivos, enfocados en situaciones conversacionales...), muchos diccionarios (en papel, online, bilingües, especializados...), muchas gramáticas (para principiantes, avanzadas, con ejercicios...) y muchos otros materiales, pero para poder sacar el mayor provecho de ellos, es necesario que el estudiante posea las herramientas adecuadas.

Centrando nuestro interés en el aprendizaje de vocabulario, hay que decir que las herramientas y técnicas que los estudiantes de hoy utilizan son bastante rudimentarias:

- La mayoría de ellos almacenan su vocabulario en formato papel.
- La información que suelen incluir en su vocabulario es el término en sí y su significado.
- La técnica que suelen utilizar para estudiar su vocabulario es memorizar un término y su significado.

Utilizar este tipo de herramientas y técnicas tienen consecuencias negativas:

- Resulta muy complicado localizar una palabra concreta.
- Resulta muy difícil organizar el vocabulario por temas, campos semánticos u otros criterios que el estudiante considere oportuno.
- Estudiar vocabulario disponiendo tan solo de los términos y sus significados puede acarrear problemas de uso.
- Resulta bastante habitual traspapelar una hoja de vocabulario, perdiendo así información que probablemente costó mucho recopilar.
- Para poder realizar ejercicios de vocabulario necesitan tener acceso a dicho material.

Teniendo en cuenta todos estos problemas (los cuales yo, personalmente, he vivido y los sigo viviendo), parece necesaria la existencia de una herramienta que permita gestionar vocabulario. ¿Existen actualmente herramientas que lo hagan? Tal y como se puede ver en la sección 3, la respuesta a esta pregunta es "No", al menos ninguna lo suficientemente completa para considerarse un gestor de vocabulario.

Debido a la inexistencia de aplicaciones gestoras de vocabulario, surge la necesidad de crear una, con el fin de cubrir un vacío que sin duda afecta a un gran número de personas, incluyéndome a mí.

1.2. Objetivos

El propósito principal de este proyecto es diseñar y construir un gestor de vocabulario que ayude a personas hispanohablantes que estudian alemán a almacenar, organizar, recuperar, estudiar y ampliar su vocabulario de forma fácil y rápida.

Por otra parte, el sistema debe diseñarse e implementarse de la forma más genérica posible con el fin de poder utilizar el resultado de

este proyecto como base para construir gestores de vocabulario para otros idiomas.

Los objetivos principales que debe cumplir el gestor a desarrollar son los siguientes:

- Almacenar vocabulario en un soporte no volátil.
- Almacenar, para un término dado, la mayor cantidad de información posible: significados, características gramaticales, características de uso, ejemplos de uso y frases hechas.
- Clasificar el vocabulario almacenado por temas, campos semánticos y/o cualquier otro criterio que crea conveniente.
- Recuperar el vocabulario almacenado de forma rápida y precisa.
- Permitir que el vocabulario almacenado sea compartido entre distintos usuarios.
- Generar ejercicios de vocabulario a partir del vocabulario almacenado.
- Generar listados convencionales de vocabulario.
- Ampliar el vocabulario almacenado buscando términos online directamente desde el gestor.

Si se desea una visión más completa de los objetivos del sistema, se recomienda leer la sección 4.2.

1.3. Estructura del Documento

El documento actual se estructura en diez secciones principales y un anexo. La primera sección detalla la motivación y los objetivos de este proyecto. La segunda, describe la gestión de proyecto realizada. La tercera, habla sobre el estado del arte. Las secciones cuarta, quinta y sexta describen las fases de análisis, diseño e implementación respectivamente. La sección séptima proporciona la planificación final del proyecto y el análisis de costes realizado. La octava, ofrece conclusiones e ideas para trabajos futuros. Las últimas

secciones principales, las secciones 9 y 10, corresponden al glosario de términos y acrónimos y a la bibliografía. Por último, se presenta un anexo con el manual de usuario del sistema final.

2. Gestión del Proyecto

Todo proyecto debe ser gestionado de forma adecuada para garantizar su éxito. La gestión de proyectos suele intentar responder a preguntas como estas:

- ¿Por qué fases debe pasar el proyecto?
- ¿Qué procedimientos hay que ejecutar en cada fase?
- ¿Qué documentación hay que generar?
- ¿Qué planificación se va a seguir?
- ¿Es la planificación rígida o flexible?

2.1. Gestión Software

La gestión de este proyecto se ha llevado a cabo siguiendo una metodología de desarrollo de software, aplicando el ciclo de vida más conveniente y planificando adecuadamente las tareas de desarrollo.

2.1.1. Metodología de Desarrollo

Una metodología de desarrollo de software es un conjunto estándar de conceptos, prácticas y criterios que se utiliza para estructurar, planear y controlar el proceso de desarrollo de sistemas de información.

Elegir una metodología apropiada para un proyecto es una tarea muy importante, pues el éxito del mismo depende de ésta. Actualmente existen muchas metodologías de desarrollo de software, las cuales se pueden clasificar en dos grandes grupos:

- **Pesadas o tradicionales:** poseen muchas reglas y prácticas para el desarrollo y la documentación del proyecto; sin embargo, estas metodologías tienen una visión muy burocrática del proceso software, dando más importancia a la documentación que al propio software a desarrollar. Estas metodologías suelen utilizarse en proyectos con una o varias de las siguientes propiedades:
 - Equipos de desarrollo grandes (más de 10 integrantes).

- Requisitos bien definidos.
- Una duración superior a un año.
- **Ligeras o ágiles:** se centran más en el desarrollo, dejando la documentación en un segundo plano. Las prácticas y reglas que poseen suelen ser pocas, dejando al equipo de desarrollo una mayor libertad. Estas metodologías suelen utilizarse en proyectos con una o varias de las siguientes propiedades:
 - Equipos de desarrollo pequeños (menos de 10 integrantes).
 - Requisitos cambiantes.
 - Una duración menor de un año.

Para el proyecto actual, que es un proyecto de corta duración (6 meses), con un equipo de desarrollo pequeño (1 integrante) y con requisitos que pueden variar en el tiempo, lo más apropiado es seguir una metodología ligera. Sin embargo, las metodologías ligeras no proporcionan guías ni técnicas para generar documentación de calidad, algo que se desea para este proyecto. Por este motivo, se ha decidido seguir una metodología híbrida (decisión bastante habitual en proyectos con equipos de desarrollo de menos de 3 integrantes), con lo mejor de las metodologías tradicionales y lo mejor de las metodologías ligeras:

- De las metodologías tradicionales, en particular, de la metodología propuesta por la ESA [1], se tomarán algunas guías y prácticas sobre cómo desarrollar software y sobre cómo generar documentos de calidad.
- De las metodologías ligeras, en particular, de la metodología Extreme Programming [2], se tomarán sus valores y principios (simplicidad, comunicación, retroalimentación, respeto y valentía) y se aplicarán sus prácticas de desarrollo.

2.1.2. Ciclo de Vida

El ciclo de vida de un proyecto software determina y describe las fases por las que dicho proyecto debe pasar.

Para el proyecto actual, se ha decidido aplicar un ciclo de vida incremental como el que se muestra a continuación:

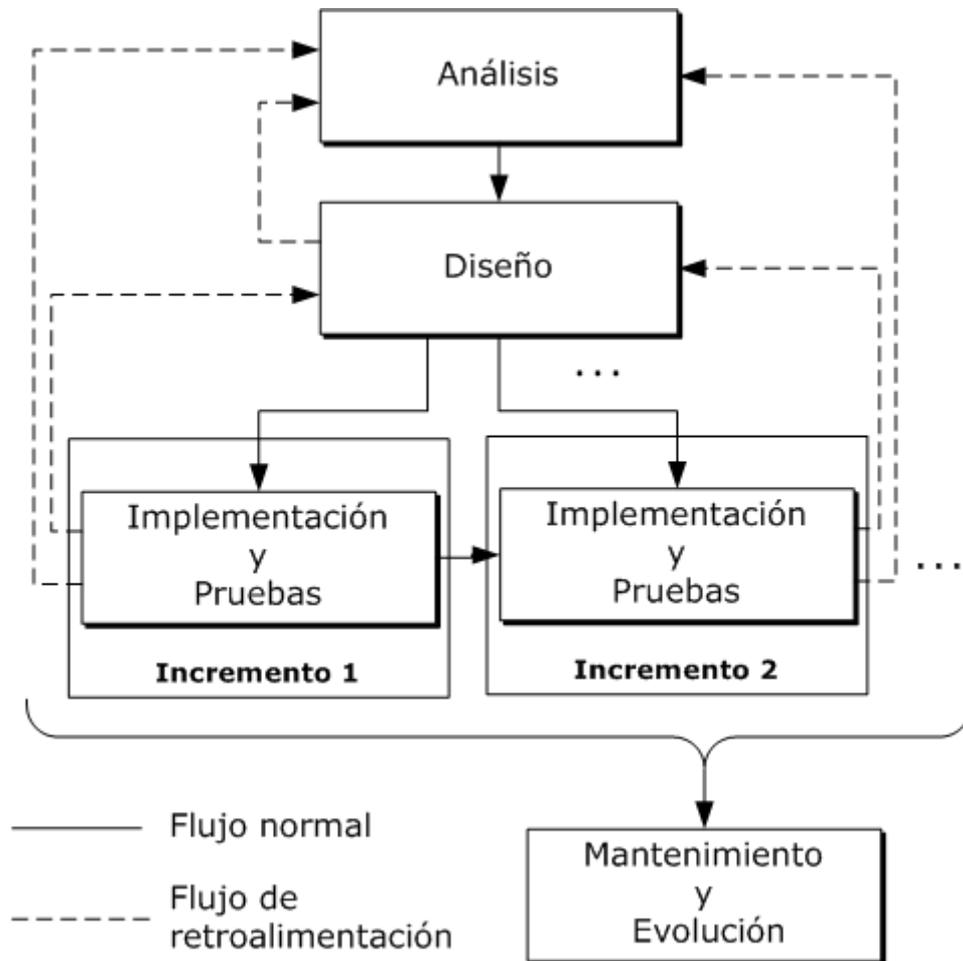


Ilustración 1 – Ciclo de Vida Incremental

Cada una de las fases que compone el ciclo de vida de este proyecto tiene una serie de objetivos:

- **Análisis:** determinar los objetivos del sistema y capturar y especificar los requisitos de usuario y de software.
- **Diseño:** diseñar la base de datos, decidir la arquitectura del sistema y diseñar los componentes que lo formarán.
- **Implementación y pruebas:** implementar y probar la base de datos y la aplicación software a desarrollar. Es importante comentar que esta fase se lleva a cabo en incrementos:
 - Para cada incremento se debe implementar y probar un conjunto de funcionalidades específicas.

- Tras completar un incremento, se debe comenzar el siguiente, utilizando como base el incremento anterior.
- Una vez completado el último incremento del proyecto, se dispondrá de una versión estable y completa del sistema.
- **Mantenimiento y evolución:** garantizar que el sistema resultante es útil a lo largo de toda su vida. Teniendo en cuenta que éste es un proyecto de fin de carrera, el equipo de desarrollo actual no será el encargado de llevar a cabo esta última fase, por lo que no será contemplada en este documento.

2.2. Planificación Inicial

Una tarea muy importante de la gestión de proyectos es hacer una buena planificación inicial: cuanto menor sea el número de cambios que sufra esta planificación a lo largo del desarrollo del proyecto, menor será el coste de éste y mayor será el beneficio.

En un proyecto comercial, a partir de la planificación inicial se suele generar un presupuesto para el cliente (costes de mano de obra, licencias, porcentaje de beneficios...). Como éste es un proyecto de fin de carrera, no ha sido necesario estimar tal presupuesto. Sin embargo, al final del proyecto sí se ha llevado a cabo un análisis de costes (ver sección 7.2) con el fin de darle al sistema resultante un valor desde un punto de vista económico.

A la hora de realizar la planificación inicial de este proyecto se han tenido en cuenta los hechos siguientes:

- Se trabaja de lunes a viernes, 8 horas cada día.
- El equipo de desarrollo está formado por un único integrante. Según las necesidades, éste integrante tomará uno de los siguientes roles: gestor de proyecto, analista/diseñador, diseñador de interfaces y programador.

A continuación se muestra la planificación inicial de este proyecto, que abarca desde el 1 de abril de 2010 hasta el 7 de septiembre de 2010 (en total, 160 días):

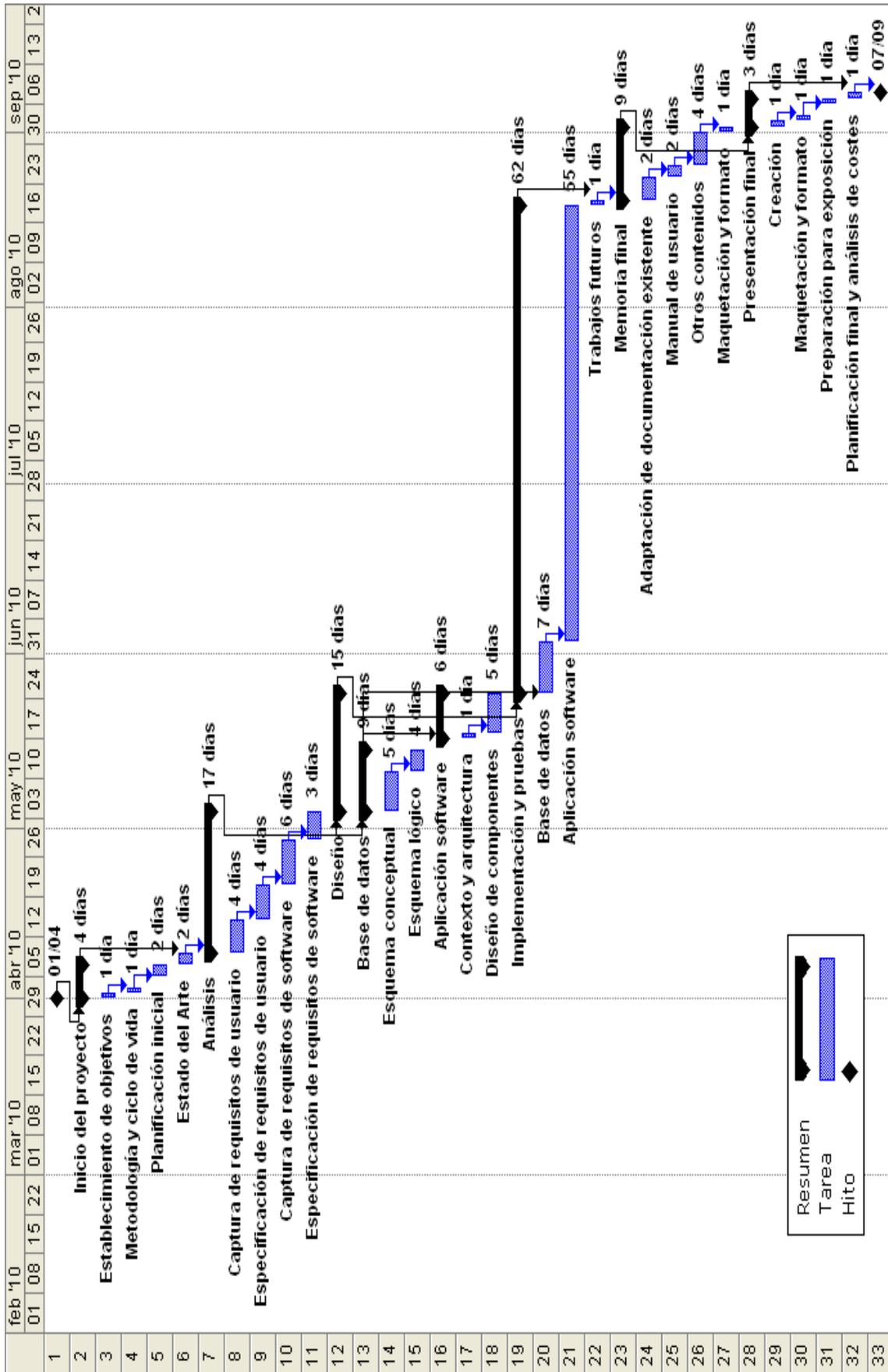


Ilustración 2 - Panificación Inicial

3. Estado del Arte

En esta sección se ofrece una visión global de las tecnologías y productos que pueden ser relacionados con este proyecto.

El primer tipo de productos altamente relacionados con el proyecto son **los diccionarios**. Los diccionarios recogen y explican de forma ordenada términos de una o más lenguas. Así pues, un diccionario es una fuente, más o menos grande, de vocabulario. Algunos diccionarios disponibles en Internet son *PONS*¹, *Hueber*² y *Reverso*³.

En el proceso de aprendizaje de vocabulario, el uso de un diccionario es imprescindible. En particular, el uso de diccionarios bilingües es el más extendido entre estudiantes de lenguas extranjeras, puesto que estos asocian términos de la lengua materna con términos de la lengua extranjera. Sin embargo, a pesar de ser una herramienta imprescindible, su utilización se reduce a la búsqueda de términos desconocidos.

Es importante resaltar que el vocabulario que posee una persona es un concepto mucho más complejo que una colección ordenada de términos y explicaciones, por muy grande que ésta sea:

- El vocabulario de una persona es limitado y no todos los términos de un diccionario pueden formar parte de su vocabulario.
- El vocabulario de una persona posee connotaciones subjetivas. Muchas personas añaden comentarios a términos complicados para facilitar su aprendizaje.
- El vocabulario no está formado por términos independientes: los términos se relacionan entre sí por campos semánticos, por tipos de uso, por caracterizaciones gramaticales...

¹ <http://de.pons.eu/>

² <http://www.hueber.de/woerterbuch/online/>

³ <http://diccionario.reverso.net/aleman-espanol/>

- El vocabulario de una persona necesita ser escrutado con precisión. De nada vale conocer un término si éste no se puede recordar (recuperar) en un momento determinado.

Por todas estas razones, un diccionario no puede considerarse un gestor de vocabulario.

Otro tipo de productos altamente relacionados con el proyecto son los **listados de vocabulario**. Un listado de vocabulario proporciona términos relacionados, habitualmente por campo semántico. Este tipo de recursos es muy útil para estudiantes de lenguas extranjeras, ya que una conocida técnica de aprendizaje de vocabulario es estudiarlo por grupos temáticos. Algunos listados de vocabulario disponibles en Internet son los proporcionados por *Syvum*⁴, *Idiomas Astalaweb*⁵ y *Wikia Education*⁶.

Sin embargo, a pesar de su gran utilidad, los listados de vocabulario no pueden considerarse una forma eficaz de gestionar vocabulario:

- Si un término puede clasificarse de diferentes formas, es necesario duplicarlo en los correspondientes listados.
- Se puede localizar fácilmente el vocabulario relacionado con un tema particular; sin embargo, para cualquier otro criterio de localización, las listas de vocabulario funcionan muy mal.
- Las listas de vocabulario suelen considerarse bloques independientes de vocabulario, lo que dificulta obtener una visión global del mismo.

Otro tipo de productos también relacionados con el proyecto son los **cursos de idiomas**, como pueden ser *Deutsch Interativ*⁷, *Curso de Alemán*⁸ y *German for Beginners*⁹. Un curso de idiomas no se

⁴ <http://es.syvum.com/idiomas/vocabulario/German/>

⁵ http://idiomas.astalaweb.com/alem%C3%A1n/v/_vocabulario.asp

⁶ <http://es.vocabularioaleman.wikia.com/wiki/Portada>

⁷ <http://www.dw-world.de/dw/0,,9572,00.html>

⁸ <http://www.curso-de-aleman.de/>

⁹ <http://www.bbc.co.uk/languages/german/lj/>

centra en el aprendizaje de vocabulario, pero juega un papel importante en él. Los cursos de idiomas suelen presentar vocabulario en situaciones y contextos particulares, lo que resulta ser una característica muy útil para el estudiante, ya que puede ver cómo un término se utiliza en la vida real. Otra característica buena que suelen tener los cursos de idiomas son los ejercicios, que ayudan al estudiante a aprender vocabulario de una forma amena y efectiva. Estas dos características hacen de los cursos una buena herramienta de aprendizaje pero por sí solas, no permiten al estudiante explotar su vocabulario al máximo:

- Localizar términos tiende a resultar complicado.
- El vocabulario no suele encontrarse organizado.
- Los ejercicios de vocabulario son estáticos, así que realizar un ejercicio dos veces no garantiza una mejor retención de vocabulario.

Teniendo en cuenta las ventajas y desventajas que los productos estudiados tienen respecto al proceso de aprendizaje de vocabulario, resulta inevitable buscar una herramienta que posea todas las características positivas hasta ahora mencionadas.

Para ello, se ha realizado una búsqueda exhaustiva de sistemas gestores de vocabulario con características similares a las requeridas por estudiantes de idiomas. Tras mucho buscar, lo único que se encontró fue una pequeña aplicación, *Vocabulary Manager*¹⁰, que permite la creación de vocabulario utilizando para cada término una imagen y dos grabaciones de audio (una para la pronunciación y otra para información adicional). Esta herramienta resulta útil para que el estudiante estudie su vocabulario de forma visual; sin embargo, la creación de vocabulario resulta muy costosa, la cantidad de información que se puede asociar a un término es muy reducida y la localización de términos se lleva a cabo de una forma muy

¹⁰ http://www.sil.org/computing/catalog/show_software.asp?id=96

rudimentaria. Por tanto, esta aplicación no puede considerarse un gestor de vocabulario completo.

Viendo que no existe ningún gestor de vocabulario completo en el mercado, parece una muy buena idea desarrollar uno desde cero y cubrir así una necesidad imperante de la sociedad.

4. Análisis

La fase de análisis es la primera fase del ciclo de vida de este proyecto. Durante esta fase se han llevado a cabo dos grandes tareas: la captura y especificación de requisitos de usuario y también la de requisitos de software.

4.1. Captura de Requisitos

La captura de requisitos es un proceso extremadamente importante en el desarrollo de cualquier sistema puesto que garantiza que se conocen de forma clara y sin ambigüedades las necesidades del cliente y/o usuario final.

En este proyecto, el proceso de captura de requisitos se ha llevado a cabo de forma exhaustiva y meticulosa, utilizando para ello tres técnicas diferentes:

- **Discusiones:** el analista y el cliente se reúnen para discutir sobre las funcionalidades y restricciones que el sistema resultante debe tener y cumplir.
- **Introspección:** el analista se pone en el lugar del usuario final y crea una lista con las funcionalidades y restricciones que cree que pueden ser útiles. Después, el analista y el cliente discuten sobre la necesidad de dichas funcionalidades y restricciones y deciden cuáles debe tener y cumplir el sistema resultante.
- **Estudios de campo - documentación:** el analista estudia a fondo el dominio del proyecto, recopilando la mayor cantidad de información posible con el fin de entender completamente el problema planteado y conocer así las claves de su resolución.

Las primeras versiones de la especificación de requisitos de usuario se obtuvieron a través de una serie de discusiones: las primeras discusiones se emplearon para capturar nuevos requisitos de usuario; las discusiones siguientes se utilizaron principalmente para revisar las descripciones de los requisitos ya capturados y asegurar que su especificación era clara, concisa y sin ambigüedades.

Una vez se disponía de una especificación inicial de requisitos de usuario, se procedió a realizar una sesión de introspección. Gracias a ésta, la especificación inicial se amplió con cuatro nuevos requisitos de usuario (a partir de los cuales se derivaron los requisitos de software *RSF-09*, *RSF-10*, *RSF-17* y *RSNF-02*).

Una vez que el cliente aprobó la especificación de requisitos de usuario, se prosiguió con la captura de requisitos de software.

La captura de requisitos de software se ha llevado a cabo principalmente mediante estudios de campo, en particular, mediante un proceso de documentación. El analista estudió diferentes diccionarios bilingües de alemán-español (*PONS*¹¹, *Hueber*¹² y *Reverso*¹³) y una gramática alemana (*Canoo.net*¹⁴) con el fin de saber qué significa exactamente "vocabulario" en el contexto de este proyecto. La respuesta, aunque requirió un esfuerzo importante, fue satisfactoria

Posteriormente, se llevaron a cabo una serie de discusiones que se emplearon para completar la especificación de requisitos de software que se tenía hasta el momento y para afinar al máximo las descripciones de sus requisitos, puesto que dicha especificación sería la base de la fase de diseño.

4.2. Requisitos de Usuario

Los requisitos de usuario proporcionan una especificación de alto nivel de las funcionalidades y restricciones que el cliente y/o usuario final necesita en el sistema resultante.

La documentación generada para esta especificación se basa en las prácticas recomendadas por la ESA [3].

¹¹ <http://de.pons.eu/>

¹² <http://www.hueber.de/woerterbuch/online/>

¹³ <http://diccionario.reverso.net/aleman-espanol/>

¹⁴ http://canoo.net/index_en.html

4.2.1. Funcionalidades del Sistema

Las funcionalidades que el sistema resultante debe tener son las siguientes:

- Crear y abrir diccionarios de usuario.
- Añadir, editar y eliminar términos de un diccionario de usuario, teniendo en cuenta que un término se define mediante significados, características gramaticales, características de uso, ejemplos de uso y frases hechas.
- Añadir y eliminar etiquetas organizativas de un término. Esto permite al usuario, entre otras cosas, clasificar su vocabulario por cualquier campo semántico.
- Añadir y eliminar la marca "Estudiar" de un término.
- Asignar el nombre del usuario actual a los significados nuevos o editados. De esta forma se puede saber quién creó o editó por última vez un significado, algo muy útil cuando se comparten diccionarios de usuario.
- Buscar términos en un diccionario de usuario (A.K.A. búsqueda local de términos) y/o en un diccionario online (A.K.A. búsqueda online de términos).
- Buscar en un diccionario de usuario todos los términos que cumplan un determinado criterio de búsqueda (A.K.A. búsqueda avanzada de términos).
- Añadir automáticamente a un diccionario de usuario términos obtenidos a partir de búsquedas online de términos.
- Añadir automáticamente a términos existentes en un diccionario de usuario nueva información obtenida a partir de búsquedas online de términos.
- Generar, a partir de un diccionario de usuario, listados de vocabulario cuyos términos cumplan un determinado criterio de búsqueda.

- Generar, a partir de los resultados de una búsqueda avanzada de términos, diferentes tipos de ejercicios (asociación de términos y significados, asociación de sustantivos con artículos definidos, especificación de plurales de sustantivos y especificación de presente, pasado y participio de verbos irregulares).
- Responder ejercicios de vocabulario.
- Comprobar automáticamente si las respuestas dadas a un ejercicio de vocabulario son correctas o incorrectas.
- Obtener automáticamente la solución de un ejercicio de vocabulario.
- Guardar y abrir ejercicios de vocabulario.

4.2.2. Restricciones del Sistema

Las restricciones que el sistema resultante debe cumplir son las siguientes:

- Debe tener una interfaz gráfica de usuario.
- Debe ser multiplataforma.
- No debe requerir la instalación de aplicaciones o componentes, ya sean propios o de terceros.
- Debe ser capaz de manejar un volumen medio de 8000 términos.
- El impacto de futuras ampliaciones debe ser mínimo.

4.2.3. Usuarios del Sistema

VocAssist (De-Es) es una aplicación que no hace distinción entre distintos tipos de usuarios: cualquier usuario que tenga acceso a una instancia de la aplicación puede usarla sin ningún tipo de restricción.

El nivel de informática que requieren los usuarios de *VocAssist (De-Es)* para utilizar correctamente la aplicación es medio-bajo.

4.2.4. Entorno Operacional

El entorno operacional de *VocAssist (De-Es)* es muy simple: un ordenador con acceso a Internet, siendo este acceso sólo necesario para llevar a cabo búsquedas online de términos.

Por otro lado, el ordenador deberá disponer de un entorno de ejecución adecuado y de recursos suficientes para ejecutar la aplicación.

4.2.5. Especificación de Requisitos

Para evitar incluir información duplicada en este documento, se ha decidido omitir la especificación de requisitos de usuario, ya que la especificación de requisitos de software (ver sección 4.3.1) proporciona la misma información pero ampliada.

4.3. Requisitos de Software

Los requisitos de software proporcionan una especificación detallada de las funcionalidades y restricciones que el sistema resultante debe tener y cumplir.

La documentación generada para esta especificación se basa en las prácticas recomendadas por la ESA [4].

4.3.1. Especificación de Requisitos

Con el fin de proporcionar una especificación de requisitos clara, concisa y rastreable, cada requisito de software se ha especificado mediante una tabla como ésta:

ID	RSF-02	Tipo	Funcional
Nombre	Abrir diccionario		
Descripción	El usuario debe poder abrir diccionarios de usuario existentes. Para ello, debe especificar la ruta en disco en la que se encuentre el diccionario de usuario que desee abrir.		

Relaciones	RSF-12	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 1 – Ejemplo de un Requisito de Software

El significado que tiene cada uno de los campos de la tabla anterior se describe a continuación:

- **ID:** identificador del requisito. Por motivos de rastreabilidad, los identificadores de requisitos de software se especifican siguiendo un formato particular:
 - Los requisitos funcionales se especifican como RSF-XX, donde XX es un número entero de dos cifras que coincide con la numeración del requisito de usuario que da origen a éste.
 - Los requisitos no funcionales se especifican como RSNF-XX, donde XX es un número entero de dos cifras que coincide con la numeración del requisito de usuario que da origen a éste.
- **Tipo:** tipo de requisito. Puede tomar valores como "Funcional", "Interfaz", "Portabilidad" y "Escalabilidad".
- **Nombre:** nombre corto y descriptivo que permita al lector hacerse una idea de la finalidad del requisito.
- **Descripción:** descripción detallada del requisito.
- **Relaciones:** requisitos de software con los que se relaciona este requisito. Si existen relaciones, se especificarán los identificadores de los requisitos con los que se relaciona. Si no existen relaciones, se escribirá "-".
- **Dependencias:** requisitos de software de los que depende este requisito. Si existen dependencias, se especificarán los identificadores de los requisitos de los que depende. Si no existen dependencias, se escribirá "-".
- **Verificable:** indica si el requisito puede ser verificado o no. Los únicos valores posibles son "Sí" y "No".

- **Criticalidad:** indica cuán crítico es para el cliente el incumplimiento del requisito. Los únicos valores posibles son "Alta", "Media" y "Baja".
- **Necesidad:** indica cuán necesario es para el cliente el cumplimiento del requisito. Los únicos valores posibles son "Esencial", "Conveniente" y "Opcional".
- **Conveniencia:** indica cuán conveniente es para el cliente el cumplimiento del requisito. Los únicos valores posibles son "Alta", "Media" y "Baja". En el caso de particular de requisitos esenciales, la conveniencia se establece siempre como alta.

4.3.1.1. Requisitos Funcionales

ID	RSF-01	Tipo	Funcional
Nombre	Crear diccionario		
Descripción	El usuario debe poder crear nuevos diccionarios de usuario. Para ello, debe especificar la ruta en disco en la que desee guardarlo. Una vez creado, este diccionario debe abrirse automáticamente.		
Relaciones	-	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 2 – RSF-01: Crear Diccionario

ID	RSF-02	Tipo	Funcional
Nombre	Abrir diccionario		
Descripción	El usuario debe poder abrir diccionarios de usuario existentes. Para ello, debe especificar la ruta en disco en la que se encuentre el diccionario de usuario que desee abrir.		
Relaciones	RSF-12	Dependencias	-

Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 3 – RSF-02: Abrir Diccionario

ID	RSF-03	Tipo	Funcional
Nombre	Definición de término		
Descripción	<p>Un término se define a través de la siguiente información:</p> <ul style="list-style-type: none"> • Término: el término en sí. Un término puede tener cualquier número de entradas. • Entradas: una entrada se compone de un comentario, de la categoría gramatical del término y de la información particular asociada a la categoría gramatical: <ul style="list-style-type: none"> ○ Sustantivos: género, número, si el sustantivo puede variar en número, forma plural en nominativo y forma singular en genitivo. ○ Verbos: tipo de verbo, tipo de prefijo (si tiene), tercera persona del singular del presente y pasado y la forma de participio, el verbo auxiliar que requiere para la formación de tiempos compuestos y la regularidad del verbo. ○ Adjetivo: forma comparativa y superlativa. ○ Artículos: tipo de artículo y formas femenina, neutra y plural en nominativo. ○ Pronombres: tipo de pronombre e información particular asociada al tipo de pronombre: <ul style="list-style-type: none"> ▪ Personales y reflexivos: persona, número, caso y en caso de ser un pronombre en tercera persona del singular, 		

	<p>su género.</p> <ul style="list-style-type: none">▪ Posesivos: persona, número, género y formas femenina, neutra y plural en nominativo.▪ Demostrativos e indefinidos: si el pronombre es invariable y en caso de no serlo, sus formas femenina, neutra y plural en nominativo.▪ Demostrativos e indefinidos: si el pronombre varía en género y número y en caso de no hacerlo, sus formas femenina, neutra y plural en nominativo. <p>El resto de pronombres no llevan ninguna información asociada.</p> <ul style="list-style-type: none">○ Preposición: caso (declinación) regido. <p>El resto de categorías gramaticales no llevan ninguna información asociada.</p> <p>Una entrada puede tener cualquier número de significados y cualquier número de frases hechas, y estos ordenan entre sí a través de un campo orden que el usuario puede modificar.</p> <ul style="list-style-type: none">• Significados: un significado se compone de la siguiente información: el significado en sí, el registro lingüístico y las restricciones geográficas que limitan el uso del significado, las etiquetas organizativas que clasifican significados, la marca "Estudiar" que indica si un significado se desea estudiar, un comentario y el nombre del usuario que añadió o editó el significado. <p>Un significado puede tener cualquier número de ejemplos de uso y cualquier número de frases hechas, y estos ordenan entre sí a través de un campo orden que el usuario puede modificar.</p> <ul style="list-style-type: none">• Ejemplos de uso: un ejemplo de uso se compone del ejemplo de uso en sí y de su
--	--

	traducción. <ul style="list-style-type: none"> • Frases hechas: una frase hecha se compone de la frase hecha en sí y de su traducción. Para más información, ver el diseño conceptual de base de datos (sección 5.1.1).		
Relaciones	RSF-04, RSF-05, RSF-17, RSF-18, RSF-19, RSF-20, RSF-21	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 4 – RSF-03: Definición de Término

ID	RSF-04	Tipo	Funcional
Nombre	Añadir términos		
Descripción	El usuario debe poder añadir términos a un diccionario de usuario abierto. Cualquier información relativa a un término y contemplada por el sistema debe poder ser especificada, excepto el nombre del usuario que se encuentra añadiendo el término, pues dicho nombre debe ser establecido automáticamente por el sistema.		
Relaciones	RSF-12	Dependencias	RSF-02, RSF-03
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 5 – RSF-04: Añadir Términos

ID	RSF-05	Tipo	Funcional
Nombre	Editar términos		

Descripción	El usuario debe poder editar términos de un diccionario de usuario abierto. Para encontrar el término a editar, el usuario debe realizar una búsqueda local de dicho término. Cualquier información relativa a un término y contemplada por el sistema debe poder ser editada, excepto el nombre del usuario que se encuentra editando el término, pues dicho nombre debe ser establecido automáticamente por el sistema.		
Relaciones	-	Dependencias	RSF-02, RSF-03, RSF-12
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 6 – RSF-05: Editar Términos

ID	RSF-06	Tipo	Funcional
Nombre	Eliminar términos		
Descripción	El usuario debe poder eliminar términos de un diccionario de usuario abierto. Para encontrar el término a eliminar, el usuario debe realizar una búsqueda local de dicho término.		
Relaciones	-	Dependencias	RSF-02, RSF-12
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 7 – RSF-06: Eliminar Términos

ID	RSF-07	Tipo	Funcional
Nombre	Añadir etiquetas organizativas		

Descripción	<p>El usuario debe poder añadir etiquetas organizativas a términos, en particular, a sus significados. Para ello, el usuario debe especificar una etiqueta organizativa nueva o seleccionar una de las sugeridas por el sistema. El sistema debe sugerir todas las etiquetas que el diccionario de usuario abierto tenga registradas.</p> <p>Esta operación se debe poder llevar a cabo tanto al añadir términos como al editarlos.</p>		
Relaciones	-	Dependencias	RSF-02, RSF-04, RSF-05
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 8 – RSF-07: Añadir Etiquetas Organizativas

ID	RSF-08	Tipo	Funcional
Nombre	Eliminar etiquetas organizativas		
Descripción	<p>El usuario debe poder eliminar etiquetas organizativas de términos, en particular, de sus significados.</p> <p>Esta operación se debe poder llevar a cabo tanto al añadir términos como al editarlos.</p>		
Relaciones	-	Dependencias	RSF-02, RSF-04, RSF-05
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 9 – RSF-08: Eliminar Etiquetas Organizativas

ID	RSF-09	Tipo	Funcional
Nombre	Añadir marca "Estudiar"		

Descripción	El usuario debe poder añadir la marca "Estudiar" a términos, en particular, a sus significados Esta operación se debe poder llevar a cabo tanto al añadir términos como al editarlos.		
Relaciones	-	Dependencias	RSF-02, RSF-04, RSF-05
Verificable	Sí	Criticidad	Media
Necesidad	Esencial	Conveniencia	Alta

Tabla 10 – RSF-09: Añadir Marca "Estudiar"

ID	RSF-10	Tipo	Funcional
Nombre	Eliminar marca "Estudiar"		
Descripción	El usuario debe poder eliminar la marca "Estudiar" de términos, en particular, de sus significados. Esta operación se debe poder llevar a cabo tanto al añadir términos como al editarlos.		
Relaciones	-	Dependencias	RSF-02, RSF-04, RSF-05
Verificable	Sí	Criticidad	Media
Necesidad	Esencial	Conveniencia	Alta

Tabla 11 – RSF-10: Eliminar Marca "Estudiar"

ID	RSF-11	Tipo	Funcional
Nombre	Asignar nombre de usuario		
Descripción	El usuario debe poder registrar un nombre de usuario en el sistema. Dicho nombre será asignado automáticamente a significados nuevos o editados. Si el usuario no registra ningún nombre de forma explícita, se debe utilizar el nombre de usuario del		

	sistema operativo.		
Relaciones	RSF-04, RSF-05	Dependencias	RSF-02
Verificable	Sí	Criticidad	Media
Necesidad	Conveniente	Conveniencia	Media

Tabla 12 – RSF-11: Asignar Nombre de Usuario

ID	RSF-12	Tipo	Funcional
Nombre	Búsqueda local de términos		
Descripción	El usuario debe poder buscar términos en un diccionario de usuario abierto. Para ello, el sistema debe proporcionar al usuario una lista (siempre actualizada) de todos los términos existentes en el diccionario de usuario abierto. El usuario también debe poder realizar la búsqueda especificando el término que desea encontrar. El término encontrado debe visualizarse en algún componente de texto que permita el copiado de su contenido.		
Relaciones	RSF-04, RSF-05, RSF-06	Dependencias	RSF-02
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 13 – RSF-12: Búsqueda Local de Términos

ID	RSF-13	Tipo	Funcional
Nombre	Búsqueda online de términos		
Descripción	El usuario debe poder buscar términos en un diccionario online. Para ello, el usuario debe especificar el término que desea encontrar. El término encontrado debe visualizarse en algún componente de texto que permita el copiado de su contenido.		

Relaciones	RSF-15, RSF-16	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 14 – RSF-13: Búsqueda Online de Términos

ID	RSF-14	Tipo	Funcional
Nombre	Búsqueda avanzada de términos		
Descripción	El usuario debe poder buscar en un diccionario de usuario abierto todos los términos que cumplan un determinado criterio de búsqueda, entendiendo éste como la unión de una serie de criterios unitarios, los cuales se listan a continuación: texto contenido en el término, categoría gramatical, texto contenido en el significado, marca “Estudiar”, etiqueta organizativa, nombre de usuario, registro lingüístico y restricción geográfica. Los términos encontrados deben mostrarse en una lista y deben visualizarse (de forma individual) en algún componente de texto que permita el copiado de su contenido.		
Relaciones	RF-17	Dependencias	RSF-02
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 15 – RSF-14: Búsqueda Avanzada de Términos

ID	RSF-15	Tipo	Funcional
Nombre	Añadir términos automáticamente		
Descripción	El usuario debe poder añadir automáticamente a un diccionario de usuario abierto términos obtenidos a partir de búsquedas online de términos. El usuario debe poder añadir el término tal cual se obtuvo de Internet o editarlo para añadir solo la información que		

	desea.		
Relaciones	-	Dependencias	RSF-02, RSF-04, RSF-13
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 16 – RSF-15: Añadir Términos Automáticamente

ID	RSF-16	Tipo	Funcional
Nombre	Editar términos automáticamente		
Descripción	El usuario debe poder añadir automáticamente a términos existentes en un diccionario de usuario abierto nueva información obtenida a partir de búsquedas online de términos. El usuario debe poder editar la información obtenida antes de añadirla al término.		
Relaciones	-	Dependencias	RSF-02, RSF-05, RSF-13
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 17 – RSF-16: Editar Términos Automáticamente

ID	RSF-17	Tipo	Funcional
Nombre	Generar listados de vocabulario		
Descripción	El usuario debe poder generar, a partir de los resultados de una búsqueda avanzada de términos, listas de vocabulario en formato PDF.		
Relaciones	-	Dependencias	RSF-02, RSF-03, RSF-14
Verificable	Sí	Criticidad	Media

Necesidad	Conveniente	Conveniencia	Alta
------------------	-------------	---------------------	------

Tabla 18 – RSF-17: Generar Listados de Vocabulario

ID	RSF-18	Tipo	Funcional
Nombre	Generar ejercicios – Significados		
Descripción	El usuario debe poder generar, a partir de los resultados de una búsqueda avanzada de términos, ejercicios de asociación de términos y significados. El usuario debe poder indicar el número máximo de preguntas que desea que tenga el ejercicio.		
Relaciones	-	Dependencias	RSF-02, RSF-03, RSF-14
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 19 – RSF-18: Generar Ejercicios – Significados

ID	RSF-19	Tipo	Funcional
Nombre	Generar ejercicios – Artículos		
Descripción	El usuario debe poder generar, a partir de los resultados de una búsqueda avanzada de términos, ejercicios de asociación de sustantivos y artículos definidos. El usuario debe poder indicar el número máximo de preguntas que desea que tenga el ejercicio.		
Relaciones	-	Dependencias	RSF-02, RSF-03, RSF-14
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 20 – RSF-19: Generar Ejercicios – Artículos

ID	RSF-20	Tipo	Funcional
Nombre	Generar ejercicios – Plurales		
Descripción	El usuario debe poder generar, a partir de los resultados de una búsqueda avanzada de términos, ejercicios de especificación de plurales de sustantivos. El usuario debe poder indicar el número máximo de preguntas que desea que tenga el ejercicio.		
Relaciones	-	Dependencias	RSF-02, RSF-03, RSF-14
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 21 – RSF-20: Generar Ejercicios – Plurales

ID	RSF-21	Tipo	Funcional
Nombre	Generar ejercicios – Verbos		
Descripción	El usuario debe poder generar, a partir de los resultados de una búsqueda avanzada de términos, ejercicios de especificación de presente, pasado y participio de verbos irregulares. El usuario debe poder indicar el número máximo de preguntas que desea que tenga el ejercicio.		
Relaciones	-	Dependencias	RSF-02, RSF-03, RSF-14
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 22 – RSF-21: Generar Ejercicios – Verbos

ID	RSF-22	Tipo	Funcional
Nombre	Responder ejercicios		

Descripción	El usuario debe poder especificar respuestas a ejercicios de vocabulario. Para ello, el usuario debe poder interactuar con ejercicios.		
Relaciones	RSF-18, RSF-19, RSF-20, RSF-21, RSF-23, RSF-24, RSF-25, RSF-26	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 23 – RSF-22: Responder Ejercicios

ID	RSF-23	Tipo	Funcional
Nombre	Corregir ejercicios		
Descripción	El usuario debe poder comprobar automáticamente si las respuestas dadas a un ejercicio de vocabulario son correctas o incorrectas. Al corregir un ejercicio, el sistema debe indicar claramente por cada pregunta si ésta ha sido respondida y si lo ha sido, si la respuesta dada es correcta o incorrecta.		
Relaciones	RSF-18, RSF-19, RSF-20, RSF-21, RSF-24, RSF-25, RSF-26	Dependencias	RSF-22
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 24 – RSF-23: Corregir Ejercicios

ID	RSF-24	Tipo	Funcional
Nombre	Solucionar ejercicios		
Descripción	El usuario debe poder obtener automáticamente la		

	solución de un ejercicio de vocabulario. Al solucionar un ejercicio, el sistema debe indicar claramente por cada pregunta si la respuesta dada es correcta o incorrecta y en caso de ser incorrecta, proporcionar la solución.		
Relaciones	RSF-18, RSF-19, RSF-20, RSF-21, RSF-23, RSF-25, RSF-26	Dependencias	RSF-22
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 25 – RSF-24: Solucionar Ejercicios

ID	RSF-25	Tipo	Funcional
Nombre	Guardar ejercicios		
Descripción	El usuario debe poder guardar ejercicios de vocabulario generados. Para ello, debe especificar la ruta en disco en la que desee guardarlo.		
Relaciones	RSF-22, RSF-23, RSF-24, RSF-26	Dependencias	RSF-18, RSF-19, RSF-20, RSF-21
Verificable	Sí	Criticidad	Media
Necesidad	Conveniente	Conveniencia	Alta

Tabla 26 – RSF-25: Guardar Ejercicios

ID	RSF-26	Tipo	Funcional
Nombre	Abrir ejercicios		
Descripción	El usuario debe poder abrir ejercicios de vocabulario existentes. Para ello, debe especificar la ruta en disco en la que se encuentre el ejercicio que desee abrir.		

Relaciones	RSF-18, RSF-19, RSF-20, RSF-21, RSF-22, RSF-23, RSF-24	Dependencias	RSF-25
Verificable	Sí	Criticidad	Media
Necesidad	Conveniente	Conveniencia	Alta

Tabla 27 – RSF-26: Abrir Ejercicios

4.3.1.2. Requisitos No-Funcionales

ID	RSNF-01	Tipo	Interfaz
Nombre	Interfaz gráfica		
Descripción	La aplicación debe ofrecer al usuario una interfaz gráfica de alta calidad, debiendo seguir los principios de la usabilidad.		
Relaciones	RSNF-02	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 28 – RSNF-01: Interfaz Gráfica

ID	RSNF-02	Tipo	Interfaz
Nombre	Idioma de la interfaz		
Descripción	La interfaz de la aplicación debería poder leerse tanto en español como en alemán. Para ello, el usuario debería poder seleccionar qué idioma utilizar para localizar/internacionalizar la interfaz del sistema.		
Relaciones	-	Dependencias	RSNF-01
Verificable	Sí	Criticidad	Baja
Necesidad	Opcional	Conveniencia	Media

Tabla 29 – RSNF-02: Idioma de la Interfaz

ID	RSNF-03	Tipo	Portabilidad
Nombre	Multiplataforma		
Descripción	La aplicación debe ser multiplataforma, pudiendo utilizarse tanto en plataformas Windows como en plataformas Linux.		
Relaciones	-	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 30 – RSNF-03: Multiplataforma

ID	RSNF-04	Tipo	Escalabilidad
Nombre	Sin instalación		
Descripción	La aplicación debe poder ser utilizada sin la necesidad de instalar aplicaciones o componentes, ya sean propios o de terceros. Simplemente teniendo el ejecutable del sistema, éste debe poder ejecutarse correctamente.		
Relaciones	-	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 31 – RSNF-04: Sin Instalación

ID	RSNF-05	Tipo	Escalabilidad
Nombre	Volumen de datos		
Descripción	La aplicación debe ser capaz de manejar un volumen medio de 8000 términos, donde cada término tendrá una media de dos significados y cada significado		

	tendrá asociados una media de dos ejemplos de uso y dos frases hechas. El tiempo de respuesta de operaciones sobre un diccionario de usuario con un volumen de datos como éste no debería superar los 15 segundos.		
Relaciones	-	Dependencias	-
Verificable	Sí	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 32 – RSNF-05: Volumen de Datos

ID	RSNF-06	Tipo	Escalabilidad
Nombre	Futuras ampliaciones		
Descripción	El impacto de futuras ampliaciones debe ser mínimo. Para ello, el sistema debe desarrollarse en módulos tan independientes como sea posible y separando claramente la lógica de negocio de la lógica de presentación.		
Relaciones	-	Dependencias	-
Verificable	No	Criticidad	Alta
Necesidad	Esencial	Conveniencia	Alta

Tabla 33 – RSNF-06: Futuras Ampliaciones

5. Diseño

Una vez finalizada la fase de análisis, se comienza la fase de diseño. Durante esta fase, y teniendo como base la especificación de requisitos de software, se han llevado a cabo dos grandes tareas: diseñar la base de datos y diseñar la aplicación software.

5.1. Base de Datos

Antes de llevar a cabo el diseño de la base de datos, hubo que decidir qué tipo de base de datos se iba a utilizar:

- El uso de una base de datos **deductiva** no tiene sentido teniendo en cuenta el tipo de consultas que se deberán realizar.
- El uso de una base de datos **distribuida** tampoco tiene sentido puesto que la aplicación es centralizada.
- El uso de un **almacén de datos** tampoco tiene sentido teniendo en cuenta que el volumen de datos que se espera manejar no es muy grande.
- El uso de **ficheros convencionales** resulta poco viable ya que los datos a manejar son muchos y muy heterogéneos, y construir un gestor de datos desde cero es una tarea muy costosa y complicada.
- El uso de una base de datos **jerárquica** o **en red** tampoco resulta viable puesto que estos modelos son muy costosos de llevar a cabo y muy sensibles al cambio.
- El uso de una base de datos **orientada a objetos** puede resultar muy interesante. Sin embargo, su utilización aún no está muy extendida conque puede resultar difícil encontrar personal cualificado que gestione este tipo de base de datos.
- El uso de una base de datos **relacional** resulta el más adecuado: el proceso de diseñar y construir una base de datos relacional es muy asequible, ampliarla o modificarla a posteriori es más que factible y existen numerosos gestores gratuitos disponibles en el mercado.

Una vez decidido que la base de datos será de tipo relacional, se diseña siguiendo la metodología especificada por Adoración de Miguel en [5]. Esta metodología utiliza una notación particular para modelar el diagrama E/R del esquema conceptual y el grafo relacional del esquema lógico. Para este proyecto, se ha decidido utilizar dicha notación al ser ésta una de las notaciones que más semántica recogen, más incluso que la notación del modelo EER [6], una de las más usadas internacionalmente para el diseño de bases de datos

5.1.1. Esquema Conceptual

Un esquema conceptual ofrece una descripción completa de los datos de un determinado dominio y la forma en la que éstos están relacionados. Cabe destacar que esta descripción es completamente independiente del tipo de base de datos a diseñar (en nuestro proyecto, relacional).

5.1.1.1. Diagrama E/R

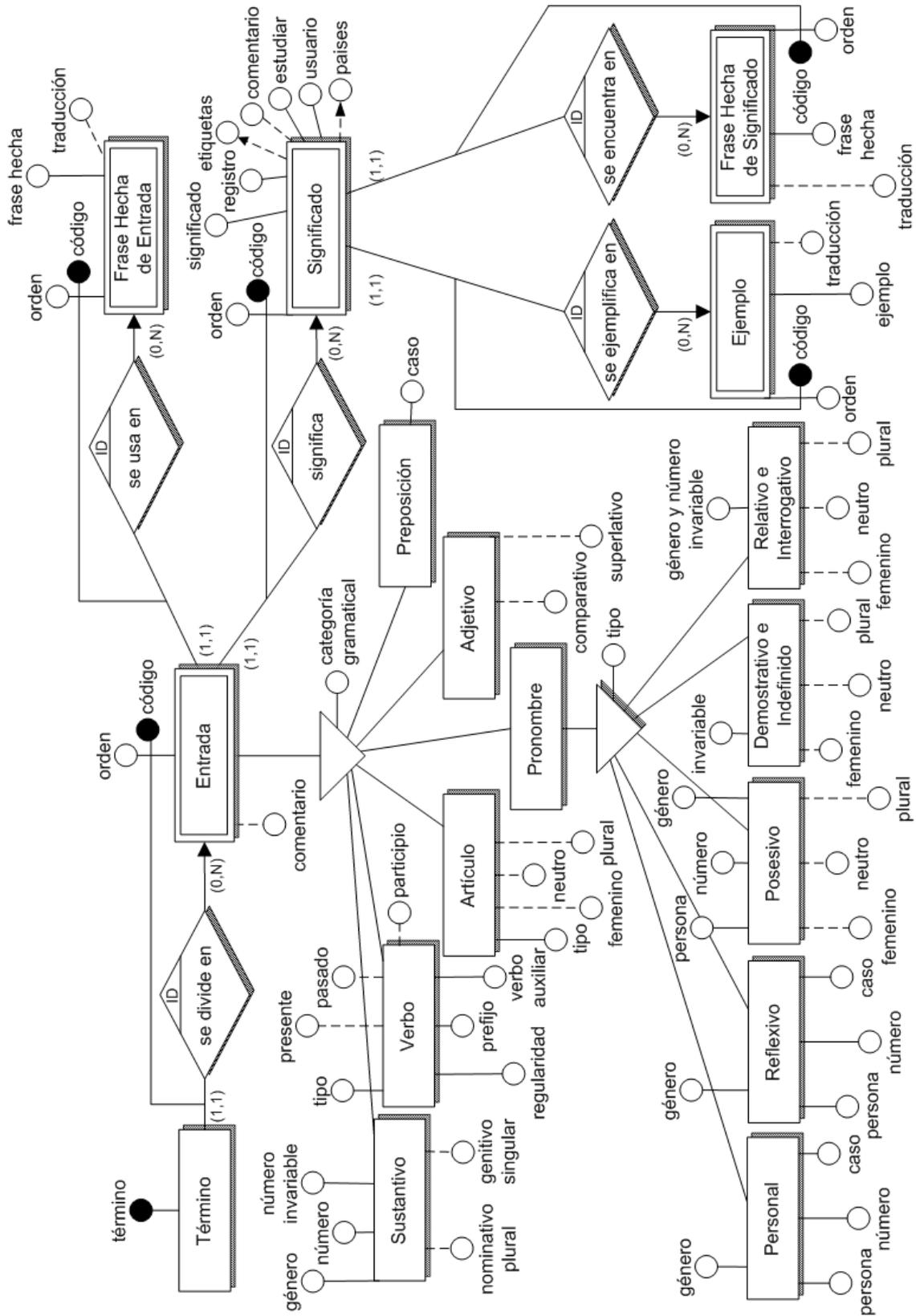


Ilustración 3 – Diagrama E/R

5.1.1.2. Supuestos Semánticos

Los supuestos semánticos asociados al esquema conceptual de la base de datos de este proyecto son los siguientes:

1. El atributo *categoría gramatical* de la entidad *Entrada* debe tomar alguno de los siguientes valores: "Sustantivo", "Verbo", "Artículo", "Pronombre", "Adjetivo", "Preposición", "Adverbio", "Conjunción" o "Interjección".
2. El atributo *género* de las entidades *Sustantivo*, *Personal*, *Reflexivo* y *Posesivo* debe tomar alguno de los siguientes valores: "Masculino", "Femenino", "Neutro" o "Ninguno".
3. El atributo *número* de las entidades *Sustantivo*, *Personal*, *Reflexivo* y *Posesivo* debe tomar alguno de los siguientes valores: "Singular" o "Plural".
4. El atributo *número invariable* de la entidad *Sustantivo*, el atributo *invariable* de la entidad *Demostrativo e Indefinido*, el atributo *género y número invariable* de la entidad *Relativo e Interrogativo* y el atributo *estudiar* de la entidad *Significado* deben tomar alguno de los siguientes valores: "Sí" o "No".
5. El atributo *tipo* de la entidad *Verbo* debe tomar alguno de los siguientes valores: "Transitivo", "Intransitivo", "Reflexivo", "Impersonal", "Auxiliar" o "Modal".
6. El atributo *prefijo* de la entidad *Verbo* debe tomar alguno de los siguientes valores: "Ninguno", "Separable" o "Inseparable".
7. El atributo *verbo auxiliar* de la entidad *Verbo* debe tomar alguno de los siguientes valores: "Haben" o "Sein".
8. El atributo *regularidad* de la entidad *Verbo* debe tomar alguno de los siguientes valores: "Regular" o "Irregular".

-
9. El atributo *tipo* de la entidad *Artículo* debe tomar alguno de los siguientes valores: "Determinado" o "Indeterminado".
 10. El atributo *tipo* de la entidad *Pronombre* debe tomar alguno de los siguientes valores: "Personal", "Reflexivo", "Recíproco", "Posesivo", "Demostrativo", "Relativo", "Interrogativo" o "Indefinido".
 11. El atributo *persona* de las entidades *Personal*, *Reflexivo* y *Posesivo* debe tomar alguno de los siguientes valores: "Primera", "Segunda" o "Tercera".
 12. El atributo *caso* de las entidades *Personal* y *Reflexivo* debe tomar alguno de los siguientes valores: "Nominativo", "Acusativo", "Dativo" o "Genitivo".
 13. El atributo *caso* de la entidad *Preposición* debe tomar alguno de los siguientes valores: "Acusativo", "Dativo" o "Genitivo".
 14. El atributo *registro* de la entidad *Significado* debe tomar alguno de los siguientes valores: "Neutro", "Formal", "Coloquial" o "Vulgar".
 15. El atributo *países* de la entidad *Significado* debe tomar alguno(s) de los siguientes valores: "Austria", "Suiza" o "Alemania del sur".
 16. Si el atributo *número* de la entidad *Sustantivo* toma el valor "Plural", el atributo *género* debe tomar el valor "Ninguno" y el atributo *número invariable*, el valor "Sí".
 17. El atributo *género* de las entidades *Personal*, *Reflexivo* y *Posesivo* debe tomar un valor distinto de "Ninguno" si el atributo *persona* toma el valor "Tercera" y el atributo *número*, el valor "Singular"; en caso contrario, el valor que debe tomar el atributo *género* es "Ninguno".

-
18. Si el atributo *invariable* de la entidad *Demostrativo e Indefinido* toma el valor "Sí", los atributos *femenino, neutro y plural* deben tomar un valor nulo.
 19. Si el atributo *género y número invariable* de la entidad *Relativo e Interrogativo* toma el valor "Sí", los atributos *femenino, neutro y plural* deben tomar un valor nulo.
 20. Los atributos *género y número invariable* de la entidad *Sustantivo* forman clave alternativa para un sustantivo particular.
 21. Los atributos *tipo y prefijo* de la entidad *Verbo* forman clave alternativa para un verbo particular.
 22. El atributo *tipo* de la entidad *Artículo* es clave alternativa para un artículo particular.
 23. Los atributos *persona, número, género y caso* de las entidades *Personal y Reflexivo* forman clave alternativa para un pronombre personal o reflexivo particular.
 24. Los atributos *persona, número y género* de la entidad *Posesivo* forman clave alternativa para un pronombre posesivo particular.
 25. El atributo *caso* de la entidad *Preposición* es clave alternativa para un término particular.
 26. Los atributos *orden y significado* de la entidad *Significado* forman clave alternativa para un significado particular.
 27. Los atributos *orden y ejemplo* de la entidad *Ejemplo* forman clave alternativa para un ejemplo particular.
 28. Los atributos *orden y frase hecha* de la entidad *Frase Hecha de Significado* forman clave alternativa para una frase hecha de significado particular.

29. Los atributos *orden* y *frase hecha* de la entidad *Frase Hecha de Entrada* forman clave alternativa para una frase hecha de entrada particular.

5.1.2. Esquema Lógico

Un esquema lógico ofrece una descripción completa de la estructura de una base de datos en términos de estructuras de datos procesables por un determinado tipo de SGBD. En nuestro proyecto, el SGBD es de tipo relacional. Cabe destacar que esta descripción depende del tipo de SGBD, no del SGBD específico.

5.1.2.1. Grafo Relacional

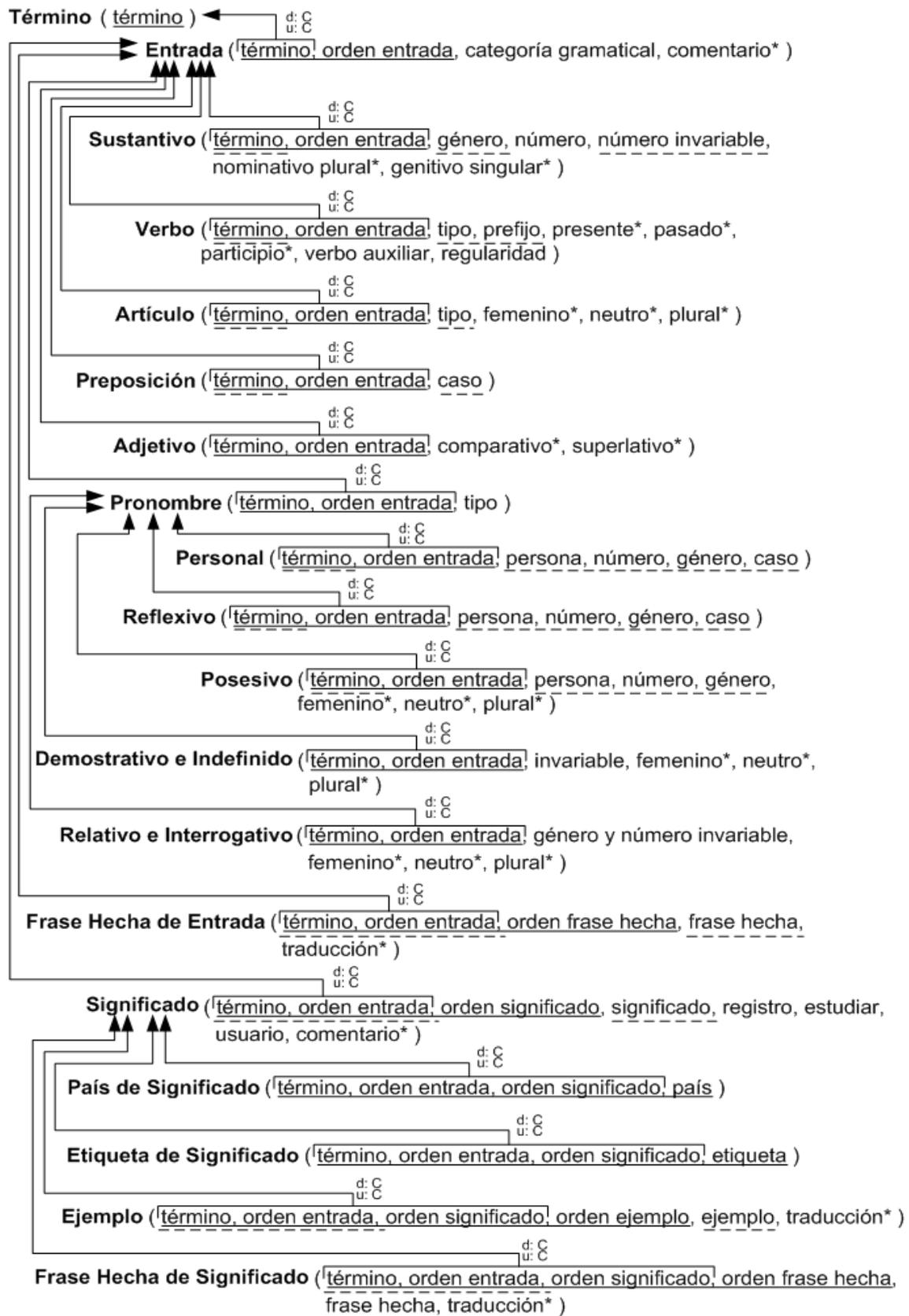


Ilustración 4 – Grafo Relacional

5.1.2.2. Dominios

Los dominios asociados al esquema lógico de la base de datos de este proyecto son los siguientes:

1. *Categorías Gramaticales*: enumeración formada por los siguientes elementos: "Sustantivo", "Verbo", "Artículo", "Pronombre", "Adjetivo", "Preposición", "Adverbio", "Conjunción" e "Interjección".
2. *Géneros*: enumeración formada por los siguientes elementos: "Masculino", "Femenino", "Neutro" y "Ninguno".
3. *Números*: enumeración formada por los siguientes elementos: "Singular" y "Plural".
4. *Booleanos*: enumeración formada por los siguientes elementos: "Sí" y "No".
5. *Tipos de Verbo*: enumeración formada por los siguientes elementos: "Transitivo", "Intransitivo", "Reflexivo", "Impersonal", "Auxiliar" y "Modal".
6. *Prefijos*: enumeración formada por los siguientes elementos: "Ninguno", "Separable" y "Inseparable".
7. *Auxiliares*: enumeración formada por los siguientes elementos: "Haben" y "Sein".
8. *Regularidades*: enumeración formada por los siguientes elementos: "Regular" e "Irregular".
9. *Tipos de Artículo*: enumeración formada por los siguientes elementos: "Determinado" e "Indeterminado".
10. *Tipos de Pronombre*: enumeración formada por los siguientes elementos: "Personal", "Reflexivo", "Recíproco", "Posesivo", "Demostrativo", "Relativo", "Interrogativo" e "Indefinido".

11. *Personas*: enumeración formada por los siguientes elementos: "Primera", "Segunda" y "Tercera".
12. *Casos 4*: enumeración formada por los siguientes elementos: "Nominativo", "Acusativo", "Dativo" y "Genitivo".
13. *Casos 3*: enumeración formada por los siguientes elementos: "Acusativo", "Dativo" y "Genitivo".
14. *Registros*: enumeración formada por los siguientes elementos: "Neutro", "Formal", "Coloquial" y "Vulgar".
15. *Países*: enumeración formada por los siguientes elementos: "Austria", "Suiza" y "Alemania del sur".

5.1.2.3. Comprobaciones

Las comprobaciones asociadas al esquema lógico de la base de datos de este proyecto son los siguientes:

1. El atributo *categoría gramatical* de la tabla *Entrada* debe tomar un valor del dominio *Categorías Gramaticales*.
2. El atributo *género* de la tablas *Sustantivo*, *Personal*, *Reflexivo* y *Posesivo* debe tomar un valor del dominio *Géneros*.
3. El atributo *número* de las tablas *Sustantivo*, *Personal*, *Reflexivo* y *Posesivo* debe tomar un valor del dominio *Números*.
4. El atributo *número invariable* de la tabla *Sustantivo*, el atributo *invariable* de la tabla *Demostrativo* e *Indefinido*, el atributo *género* y *número invariable* de la tabla *Relativo e Interrogativo* y el atributo *estudiar* de la tabla *Significado* deben tomar un valor del dominio *Booleanos*.
5. El atributo *tipo* de la entidad *Verbo* debe tomar un valor del dominio *Tipos de Verbo*.

6. El atributo *prefijo* de la entidad *Verbo* debe tomar un valor del dominio *Prefijos*.
7. El atributo *verbo auxiliar* de la entidad *Verbo* debe tomar un valor del dominio *Auxiliares*.
8. El atributo *regularidad* de la entidad *Verbo* debe tomar un valor del dominio *Regularidades*.
9. El atributo *tipo* de la entidad *Artículo* debe tomar un valor del dominio *Tipos de Artículo*.
10. El atributo *tipo* de la entidad *Pronombre* debe tomar un valor del dominio *Tipos de Pronombre*.
11. El atributo *persona* de las entidades *Personal*, *Reflexivo* y *Posesivo* debe tomar un valor del dominio *Personas*.
12. El atributo *caso* de las entidades *Personal* y *Reflexivo* debe tomar un valor del dominio *Casos 4*.
13. El atributo *caso* de la entidad *Preposición* debe tomar un valor del dominio *Casos 3*.
14. El atributo *registro* de la entidad *Significado* debe tomar un valor del dominio *Registros*.
15. El atributo *país* de la entidad *País de Significado* debe tomar un valor del dominio *Países*.

5.1.2.4. Aserciones

Las aserciones asociadas al esquema lógico de la base de datos de este proyecto son los siguientes:

1. Si el atributo *número* de la tabla *Sustantivo* toma el valor "Plural", el atributo *género* debe tomar el valor "Ninguno" y el atributo *número invariable*, el valor "Sí".
2. El atributo *género* de las tablas *Personal*, *Reflexivo* y *Posesivo* debe tomar un valor distinto de "Ninguno" si el atributo *persona* toma el valor "Tercera" y el atributo

número, el valor "Singular"; en caso contrario, el valor que debe tomar el atributo *género* es "Ninguno".

3. Si el atributo *invariable* de la tabla *Demostrativo e Indefinido* toma el valor "Sí", los atributos *femenino*, *neutro* y *plural* deben tomar un valor nulo.
4. Si el atributo *género y número invariable* de la tabla *Relativo e Interrogativo* toma el valor "Sí", los atributos *femenino*, *neutro* y *plural* deben tomar un valor nulo.

5.1.2.5. Disparadores

El esquema lógico de la base de datos de este proyecto no tiene ningún disparador asociado

5.2. Aplicación Software

Para diseñar la aplicación software se han realizado principalmente dos tareas: un diseño de la arquitectura del sistema y un diseño de los componentes que la forman.

La documentación generada para el diseño de la aplicación software se basa en las prácticas recomendadas por la ESA [7].

5.2.1. Contexto

Para poder diseñar correctamente un sistema es esencial conocer su contexto, es decir, conocer los sistemas con los que interactúa. El contexto del sistema a desarrollar en este proyecto es bastante sencillo, pues tan sólo se interactúa con un tipo de sistema externo, los diccionarios online, a los cuales se accede a través de Internet.

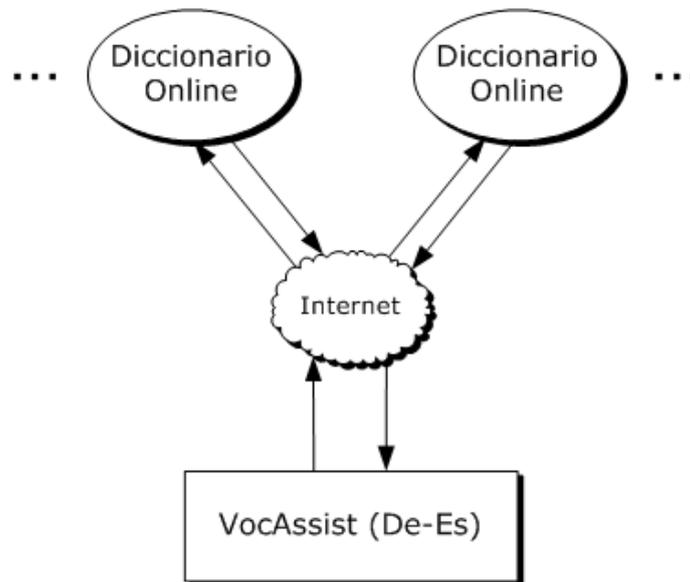


Ilustración 5 – Diagrama de Contexto

5.2.2. Arquitectura

El primer paso a la hora de diseñar un sistema es decidir su arquitectura, es decir, los grandes bloques en los que se organizarán sus componentes y las interacciones entre estos bloques:

- **Arquitectura en 1 capa:** todos los componentes se encuentran al mismo nivel, de tal forma que la lógica de presentación y la de negocio no se separan. Esta arquitectura es muy sencilla de implementar pero también muy difícil de mantener y de ampliar, debido a la alta dependencia entre componentes y a la falta de estructuración del sistema.
- **Arquitectura en 3 capas:** el sistema se divide en tres capas dispuestas en una pila como la de la ilustración siguiente:

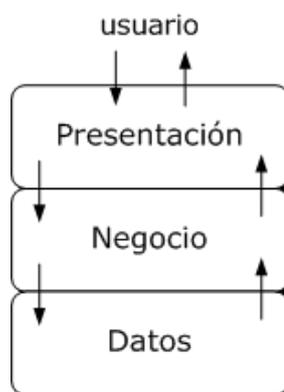


Ilustración 6 – Arquitectura en 3 Capas

La capa de presentación recibe eventos del usuario y se los pasa a la capa de negocio. Como consecuencia, la capa de negocio ejecuta alguna operación. Si la operación requiere manejar datos, la capa de negocio establece comunicaciones con la capa de datos. Cuando la capa de negocio decida que el usuario debe ser informado de algo, pasará la información correspondiente a la capa de presentación, que se encargará de mostrar dicha información de la manera más adecuada.

Implementar esta arquitectura no es una tarea compleja, pero sí conlleva un esfuerzo importante. Sin embargo, un sistema con esta arquitectura puede ser mantenido y ampliado con un esfuerzo mínimo.

- **Arquitectura MVC:** el sistema se divide en tres capas dispuestas en un triángulo como el de la ilustración siguiente:

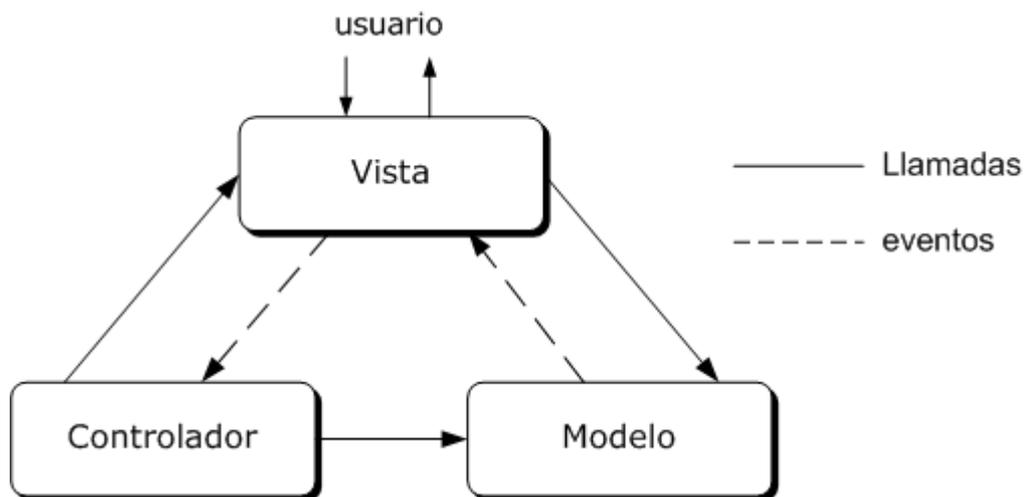


Ilustración 7 – Arquitectura MVC

Esta arquitectura es parecida a la arquitectura en 3 capas excepto por las propiedades siguientes:

- El modelo es la unión de la capa de negocio y de la capa de datos presentes en la arquitectura en 3 capas.
- La vista y el modelo se pueden comunicar entre sí: el modelo puede actualizar la visión que la vista tiene de él y la vista puede preguntar al modelo su estado actual.

- El controlador transforma los eventos del usuario generados en la vista en llamadas al modelo.

De las tres arquitecturas comentadas anteriormente, que son las arquitecturas más utilizadas actualmente, se ha decidido que el sistema a desarrollar implemente una arquitectura MVC. Los motivos de esta decisión son principalmente los siguientes:

- La lógica de presentación está claramente separada de la lógica de negocio, mejorando así la escalabilidad del sistema.
- La capa de controlador reduce el nivel de acoplamiento entre la lógica de presentación y la lógica del modelo, facilitando así el proceso de mantenimiento y evolución del sistema.

5.2.3. Diseño de Componentes

Una vez decidida la arquitectura del sistema, es momento de diseñar la aplicación.

Existen numerosos modelos disponibles para diseñar aplicaciones software: diagramas de componentes, de clases, secuenciales, de colaboración, máquinas de estado... Dependiendo de las necesidades y circunstancias de un proyecto, éste puede modelarse utilizando uno o varios de los modelos anteriores.

El proyecto actual, teniendo en cuenta la metodología seguida y los recursos disponibles, se ha decidido modelar mediante un diagrama de componentes complementado por una especificación de responsabilidades.

5.2.3.1. Diagrama de Componentes

Un diagrama de componentes proporciona una representación gráfica de los componentes que forman el sistema y de las relaciones que estos tienen entre sí.

El lenguaje utilizado para la generación de este diagrama es UML 2[8]. La elección de este lenguaje se debe principalmente a que es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

La nomenclatura utilizada para nombrar los componentes es la siguiente:

- **Identificador:** por motivos de rastreabilidad y consistencia, los identificadores de componentes se especifican en un formato específico: CO-XX donde XX es un número entero de dos cifras.
- **Nombre:** nombre corto y descriptivo que permita al lector hacerse una idea de las funcionalidades que engloba el componente (nótese que el nombre de un componente es único dentro de un determinado paquete de componentes, pero no dentro de todo el sistema).

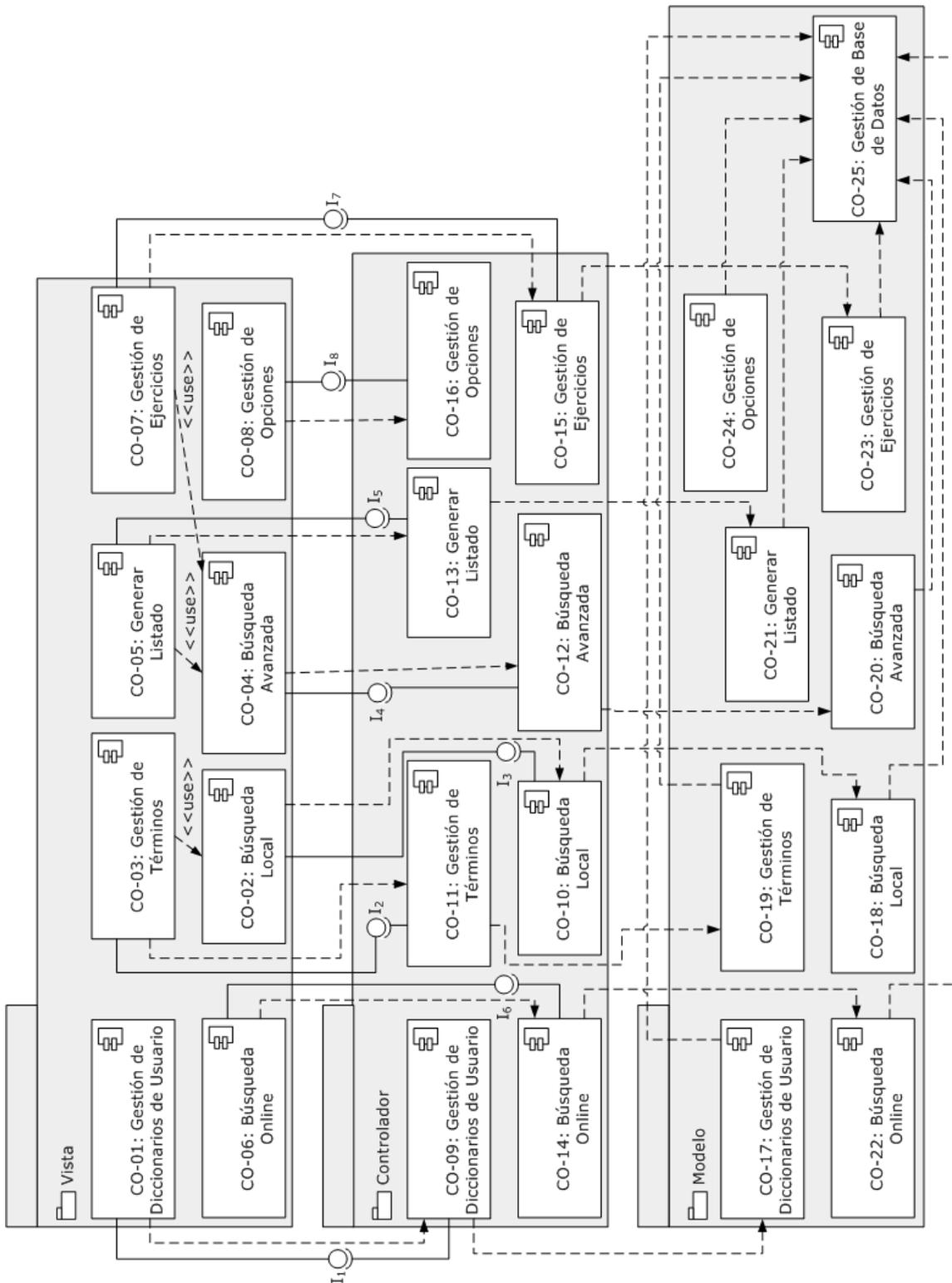


Ilustración 8 – Diagrama de Componentes

Como se puede observar en el diagrama anterior, los componentes se distribuyen en tres grandes bloques, correspondientes a las tres

capas de la arquitectura del sistema, y se relacionan entre sí de la siguiente forma:

- Los componentes de la capa vista ofrecen interfaces a los componentes de la capa controlador, los cuales utilizan estas interfaces para mostrar al usuario los resultados de sus acciones.
- Los componentes de la capa vista dependen de los componentes de la capa controlador, pues son estos los que traducen los eventos de usuario a llamadas de componentes de la capa modelo.
- Los componentes de la capa controlador dependen de los componentes de la capa modelo al ser estos los que realmente llevan a cabo las tareas requerida por el usuario.
- Todos los componentes de la capa modelo, exceptuando el componente CO-25, dependen del componente encargado de gestionar la base de datos, que es precisamente el componente CO-25.

5.2.3.2. Responsabilidades

Cada componente debe cumplir una serie de responsabilidades. Estas responsabilidades se pueden clasificar en generales y específicas. La unión de ambas responsabilidades junto con la descripción ofrecida en el diagrama de componentes proporciona una descripción completa del componente.

5.2.3.2.1. Responsabilidades Generales

Estas responsabilidades dependen exclusivamente de la capa arquitectónica en la que los componentes se encuentran dispuestos:

- Cada componente de la capa vista tiene la responsabilidad de proporcionar al usuario una o varias interfaces gráficas que le permitan:

- Acceder a las operaciones (responsabilidades específicas) ofrecidas por el componente. Para ello, se debe hacer uso del componente de la capa controlador correspondiente.
- Visualizar los resultados de dichas operaciones. Estos resultados serán proporcionados por el componente de la capa controlador correspondiente.
- Cada componente de la capa controlador tiene las siguientes responsabilidades:
 - Traducir los eventos del usuario a llamadas de componentes de la capa modelo. Los eventos del usuario se reciben a través del componente de la capa vista correspondiente.
 - Enviar al usuario los resultados obtenidos a partir de sus acciones. Esta información se manda a través de la interfaz que el componente de la capa vista correspondiente ofrece.
- Cada componente de la capa modelo tiene las siguientes responsabilidades:
 - Ejecutar la operación solicitada por el usuario. Esta operación se ejecuta como resultado de una llamada realizada por el componente de la capa controlador correspondiente. Para la ejecución de la operación es muy probable que se deban realizar llamadas sobre el componente encargado de gestionar la base de datos.
 - Enviar los resultados de la operación al usuario. Para ello, la llamada realizada por el componente de la capa controlador correspondiente debe retornar la información necesaria.
- El componente encargado de gestionar la base de datos tiene la responsabilidad de:
 - Insertar, actualizar, eliminar y recuperar información de la base de datos del sistema. Tan sólo componentes de la capa modelo pueden hacer uso de este componente.

5.2.3.2.2. Responsabilidades Específicas

Las responsabilidades específicas, que se corresponden con las funcionalidades que un componente debe ofrecer, se encuentran especificadas en las descripciones de los requisitos de software que cada componente tiene asociado.

A continuación se especifica para cada componente o conjunto de estos (ver Ilustración 8 – Diagrama de Componentes) los requisitos de software que especifican sus responsabilidades específicas (ver sección 4.3.1):

- **CO-01, CO-09, CO-17 (gestión de diccionarios de usuario):** RSF-01 y RSF-02.
- **CO-02, CO-10, CO-18 (búsqueda local):** RSF-12.
- **CO-03, CO-11, CO-19 (gestión de términos):** RSF-04, RSF-05, RSF-06, RSF-07, RSF-08, RSF-09-RSF-10 y RSF-11.
- **CO-04, CO-12, CO-20 (búsqueda avanzada):** RSF-14.
- **CO-05, CO-13, CO-21 (generar listado):** RSF-17.
- **CO-06, CO-14, CO-22 (búsqueda online):** RSF-13, RSF-15 y RSF-16.
- **CO-07, CO-15, CO-23 (gestión de ejercicios):** RSF-18, RSF-19, RSF-20, RSF-21, RSF-22, RSF-23, RSF-24, RSF-25, y RSF-26.
- **CO-08, CO-16, CO-24 (gestión de opciones):** RSF-11 y RSNF-02.
- **CO-25 (gestión de base de datos):** RSF-03.
- **CO-01, CO-02, CO-03, CO-04, CO-05, CO-06, CO-07, CO-08 (componentes de presentación):** RSNF-01.

Nótese que los requisitos de software *RSNF-03*, *RSNF-04*, *RSNF-05* y *RSNF-06*, aunque no aportan ninguna funcionalidad, afectan de forma indirecta a todos los componentes del sistema. No obstante, es

en la fase de implementación en las que estos requisitos deben tenerse seriamente en cuenta.

5.2.3.2.3. Matriz de Rastreabilidad: CO - RS

Para poder visualizar fácil y rápidamente las relaciones entre requisitos de software y componentes, se ha decidido generar una matriz de rastreabilidad. Esta matriz demuestra que en el diseño del sistema se han tenido en cuenta todas las necesidades del usuario y/o cliente final.

	CO-01	CO-02	CO-03	CO-04	CO-05	CO-06	CO-07	CO-08	CO-09	CO-10	CO-11	CO-12	CO-13	CO-14	CO-15	CO-16	CO-17	CO-18	CO-19	CO-20	CO-21	CO-22	CO-23	CO-24	CO-25	
RSF-01	X								X								X									
RSF-02	X								X								X									
RSF-03																									X	
RSF-04			X								X								X							
RSF-05			X								X								X							
RSF-06			X								X								X							
RSF-07			X								X								X							
RSF-08			X								X								X							
RSF-09			X								X								X							
RSF-10			X								X								X							
RSF-11			X					X			X					X			X					X		
RSF-12		X								X								X								
RSF-13						X							X									X				
RSF-14			X								X								X							
RSF-15						X							X									X				
RSF-16						X							X									X				
RSF-17				X								X									X					
RSF-18							X								X								X			
RSF-19							X								X								X			
RSF-20							X								X								X			
RSF-21							X								X								X			
RSF-22							X								X								X			
RSF-23							X								X								X			
RSF-24							X								X								X			
RSF-25							X								X								X			
RSF-26							X								X								X			
RSNF-01	X	X	X	X	X	X	X																			
RSNF-02								X								X									X	
RSNF-03	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RSNF-04	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

	CO-01	CO-02	CO-03	CO-04	CO-05	CO-06	CO-07	CO-08	CO-09	CO-10	CO-11	CO-12	CO-13	CO-14	CO-15	CO-16	CO-17	CO-18	CO-19	CO-20	CO-21	CO-22	CO-23	CO-24	CO-25
RSNF-05	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RSNF-06	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Tabla 34 – Matriz de Rastreabilidad: CO –RS

6. Implementación y Pruebas

Una vez finalizada la fase de diseño, se inicia la fase de implementación y pruebas. Durante esta fase, y teniendo como base la especificación de requisitos de software, el esquema relacional y el diseño de componentes, se han llevado a cabo dos grandes tareas: implementar y probar la base de datos e implementar y probar la aplicación software.

6.1. Base de datos

Para implementar y probar una base de datos se requieren dos elementos importantes: un esquema relacional y un SGBD adecuado. El esquema relacional ya se encuentra disponible gracias a la fase de diseño; el SGBD, hay que seleccionarlo al comienzo de esta fase.

6.1.1. Selección de SGBD

Para elegir el SGBD que mejor se adapte a las necesidades del proyecto, lo primero que se ha hecho es determinar las características mínimas que debe tener el SGBD elegido:

- Según el requisito *RSNF-04*, el SGBD debe poder utilizarse sin necesidad de instalación, es decir, ejecutarse en modo embebido.
- Según el requisito *RSNF-05*, el SGBD debe ser rápido y potente, lo que implica que se deben poder utilizar índices de distintos tipos.
- Según el requisito *RSNF-06*, el SGBD debe ser escalable, y por tanto, debe permitir alterar bases de datos existentes, añadir índices para acelerar consultas y funcionar correctamente con grandes volúmenes de datos.
- Debido a las limitaciones económicas del proyecto, el SGBD debe poder utilizarse sin ningún tipo de coste.
- El SGBD debe ser compatible con el lenguaje de programación que se utilice en la implementación del sistema, que en nuestro caso, es Java.

A partir de estas características mínimas, se pueden descartar los SGBD tradicionales como Oracle¹⁵ y MySQL¹⁶, que son sistemas que requieren instalación, y algunos SGBD embebidos como VistaDB¹⁷ y FlashFiler¹⁸, que no se están disponibles para Java.

De los SGBD aún en juego, los cuatro más utilizados dentro del entorno Java son: Java DB¹⁹ 10.5.3.0, Berkeley DB JE²⁰ 4.0.103, SQLite²¹ 3.6.14.2 (wrapper SQLiteJDBC²² 0.56) y HSQLDB²³ 1.8.1.3. Para cada uno de estos SGBD se han estudiado las siguientes características, las cuales se encuentran ordenadas de mayor a menor importancia para el proyecto:

1. **SQL 92:** ¿soporta el SGBD el uso del lenguaje SQL 92? Si lo hace, ¿cuántas características de este dialecto soporta?
2. **Rapidez:** ¿cómo de rápido es el SGBD en general?
3. **Claves ajenas:** ¿puede el SGBD manejar claves ajenas?
4. **Integridad referencial:** ¿garantiza el SGBD integridad referencial entre tablas?
5. **Modificación estructural:** ¿permite el SGBD realizar modificaciones estructurales en una base de datos existente?
6. **Tamaño máximo:** ¿cuál es el tamaño máximo que puede alcanzar una base de datos?
7. **Huella:** ¿qué tamaño tiene la librería que permite utilizar el SGBD dentro de Java?

¹⁵ <http://www.oracle.com/global/es/products/database/index.html>

¹⁶ <http://www.mysql.com/>

¹⁷ <http://www.vistadb.net/>

¹⁸ <http://sourceforge.net/projects/tpflashfiler/>

¹⁹ <http://www.oracle.com/technetwork/java/javadb/overview/index.html>

²⁰ <http://www.oracle.com/technetwork/database/berkeleydb/overview/index-093405.html>

²¹ <http://www.sqlite.org/>

²² <http://www.zentus.com/sqlitejdbc/>

²³ <http://hsqldb.org/>

8. **Concurrencia:** ¿permite el SGBD concurrencia? Si lo hace, ¿responde bien en entornos de alta concurrencia?
9. **Comprobación de tipos:** ¿comprueba el SGBD que el contenido de una columna sea válido con respecto al tipo de datos que tiene asociado?

A continuación se presenta una comparativa de las características (anteriormente descritas) que los SGBD estudiados poseen:

	Java DB 10.5.3.0	Berkeley DB JE 4.0.103	SQLite 3.6.14.2	HSQldb 1.8.1.3
SQL 92	5	0	3	5
Rapidez	2	3	4	5
Claves Ajenas	Sí	Sí	No	Sí
Integridad Referencial	Sí	Sí	No	Sí
Modificación Estructural	5	5	3	5
Tamaño Máximo	∞	256 PB	∞	8 GB
Huella	2,5 MB	2 MB	2,6 MB	700 KB
Concurrencia	5	5	2	3
Comprobación de Tipos	Sí	Sí	No	Sí

Tabla 35 – Comparativa de SGBD

Para interpretar correctamente la comparativa anterior es necesario saber interpretar los valores de algunas de las características estudiadas:

- **SQL 92:** el 0 significa que el SGBD no soporta el dialecto SQL 92. Enteros del 1 al 5, ambos inclusive, indican que sí se

soporta el dialecto; cuánto más alto sea el entero, mayor será el número de características del dialecto soportadas.

- **Rapidez:** cuanto más alto sea el entero, más rápido se considera el SGBD. Puede tomar enteros del 0 al 5, ambos inclusive. Esta característica se ha determinado a partir de la comparativa de *PolePosition*²⁴.
- **Modificación estructural:** cuanto más alto sea el entero, mayor número de operaciones de modificación estructural se pueden realizar. Puede tomar enteros del 0 al 5, ambos inclusive.
- **Tamaño máximo:** el ∞ significa que el SGBD no limita el tamaño, pero hay que tener en cuenta que el sistema operativo y el tipo de sistema de ficheros sí lo hace.
- **Concurrencia:** el 0 significa que el SGBD no permite concurrencia. Enteros del 1 al 5, ambos inclusive, indican que sí se permite concurrencia; cuánto más alto sea el entero, mejor responde el SGBD ante tasas altas de concurrencia.

A partir de la comparativa anterior, resulta evidente que el SGBD que mejor se adapta a las necesidades del proyecto es HSQLDB 1.8.1.3.

6.1.2. Implementación y Pruebas

Una vez elegido el SGBD a utilizar en el proyecto, se empezó a implementar la base de datos.

Gracias al esquema relacional generado en la fase de diseño, el proceso de implementación fue bastante directo. En particular, se llevaron a cabo las siguientes acciones: crear tablas, restringir atributos, establecer claves primarias, alternativas y ajenas y crear índices de dichas claves. La implementación de dominios, comprobaciones asociadas a estos dominios y aserciones no se llevó

²⁴ <http://polepos.sourceforge.net/results/html/index.html>

a cabo directamente sobre la base de datos; en su lugar, estas restricciones fueron implementadas a nivel de aplicación. Las razones para tomar esta decisión fueron principalmente dos:

1. Este tipo de restricciones suelen implementarse, en entornos Java, como código Java asociado a la base de datos. Por tanto, al implementar estas restricciones a nivel de aplicación lo único que se hace es cambiar la localización del código.
2. Algunos SGBD no permiten asociar código Java a una base de datos. Otros SGBD ni siquiera permiten implementar este tipo de restricciones. Así pues, al implementar este tipo de restricciones a nivel de aplicación, se garantiza que el impacto de cambiar de SGBD será mínimo.

A medida que se iba implementando la base de datos, se iban realizando pruebas unitarias (inserciones, borrados, actualizaciones y consultas simples), las cuales intentaban responder las siguientes preguntas: ¿las claves primarias, secundarias y ajenas se comportan cómo se espera? ¿La integridad referencial se mantiene al eliminar y/o actualizar claves ajenas? ¿Las restricciones de los atributos son adecuadas?

Tras finalizar la implementación y pruebas de la base de datos, se diseñaron, implementaron y probaron las sentencias SQL (inserciones, borrados, actualizaciones y consultas) necesarias para el funcionamiento de la aplicación. Entre las pruebas realizadas, las más importantes y no triviales fueron las encargadas de comprobar que las consultas implementadas tuvieran tiempos de respuesta aceptables (según requisito *RSNF-05*Tabla 32 – *RSNF-05*: Volumen de Datos). Para alcanzar este fin, se llevaron a cabo las siguientes tareas:

1. Crear una base de datos y poblarla con 8000 términos, donde cada término tiene una entrada con dos significados (con comentarios y dos etiquetas organizativas) y cada significado tiene dos ejemplos de uso y dos frases hechas.

2. En un ordenador con prestaciones medias (Pentium III, 888 MHz, 512 MB RAM), ejecutar las principales consultas implementadas contra la base de datos anterior y medir los tiempos que las consultas tardan en devolver los resultados.
3. Comprobar los tiempos medidos y en caso de encontrar alguno inaceptable, estudiar cómo optimizar la consulta: reescribirla, crear índices...

A continuación se muestra una gráfica con los tiempos medidos en estas pruebas:

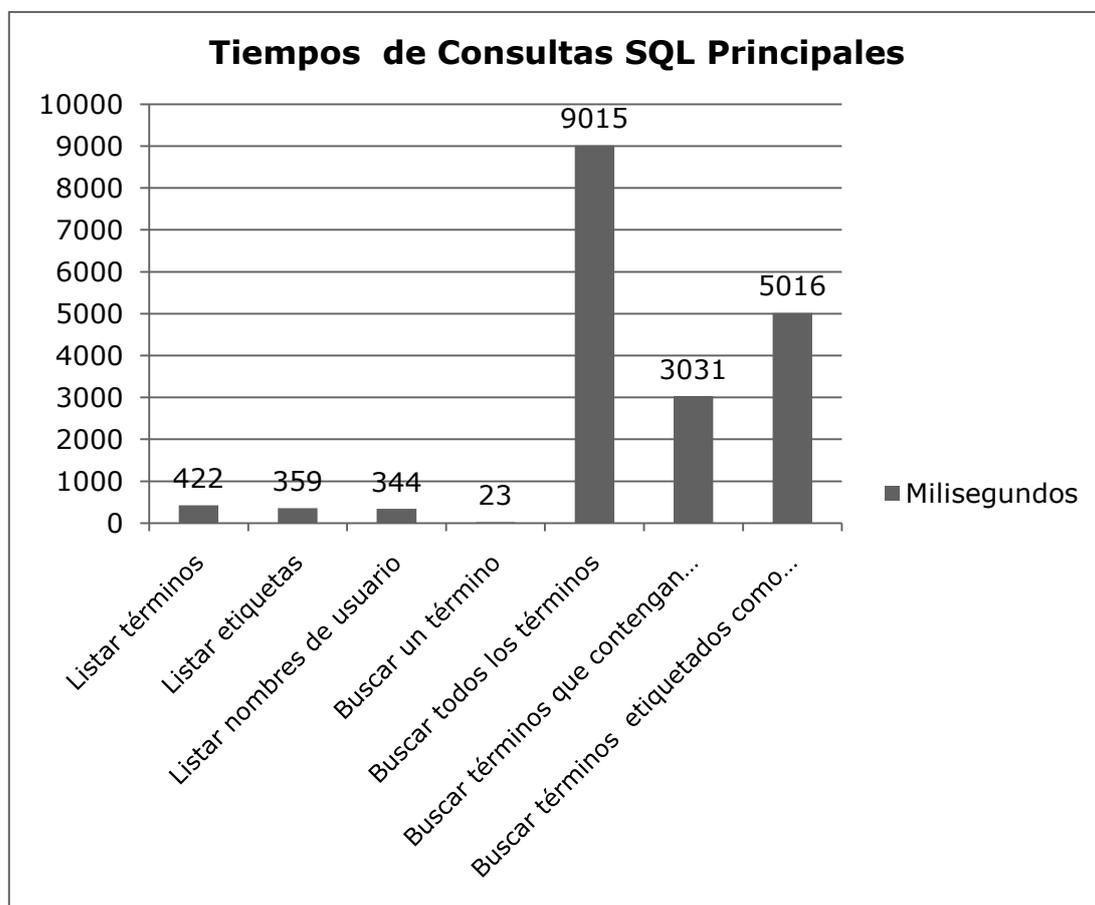


Ilustración 9 – Tiempos de Consultas SQL Principales

Como se puede observar, ninguna de las consultas probadas tardó más de 10 segundos en devolver resultados. Gracias a esto, no se tuvieron que llevar a cabo operaciones de optimización para ninguna consulta.

6.2. Aplicación Software

La implementación y pruebas de la aplicación software se han llevado a cabo partiendo del diseño arquitectónico y del diseño de componentes generados en la fase de diseño.

6.2.1. Selección del Lenguaje de Programación

Antes de poder empezar a codificar, fue necesario decidir qué lenguaje de programación era el más conveniente:

- Según el requisito *RSNF-03*, la aplicación debe poder ejecutarse en plataformas Windows y Linux. Por tanto, el lenguaje elegido debe ser portable (de lo contrario, habría que codificar una versión para Windows y otra versión para Linux).
- Tomando en consideración la planificación del proyecto y los recursos disponibles, es esencial elegir un lenguaje de programación con el que el programador del proyecto se sienta realmente cómodo.

Teniendo en cuenta las consideraciones anteriores, se decidió que el lenguaje de programación más conveniente para implementar la aplicación software era Java.

6.2.2. Estructuración en Incrementos

Tal y como se especifica en el ciclo de vida del proyecto (ver sección 2.1.2), el proceso de implementación y pruebas se ha llevado a cabo en incrementos.

Teniendo en cuenta el diseño de componentes, se decidió dividir el proceso en ocho incrementos:

- Cada uno de estos incrementos tiene como objetivo implementar y probar unos determinados componentes del sistema, ya sea parcial o totalmente.
- Los incrementos se deben desarrollar por orden de criticidad, de tal forma que las funcionalidades más importantes estén disponible cuanto antes.

A continuación se listan los incrementos en los que se ha dividido la fase de implementación y pruebas (ordenados secuencialmente) y se especifican los componentes del sistema que deben implementar:

1. **Gestión de diccionarios de usuario:** CO-01, CO-09, CO-17 y CO-25.
2. **Búsquedas locales:** CO-02, CO-10, CO18 y CO-25.
3. **Gestión de términos:** CO-03, CO-11, CO-19 y CO-25.
4. **Búsquedas avanzadas:** CO-04, CO-12, CO-20 y CO25.
5. **Búsquedas online:** CO-06, CO-15, CO-22 y CO-25.
6. **Gestión de ejercicios:** CO-07, CO-15, CO-23 y CO-25.
7. **Gestión de opciones:** CO-08, CO-16, CO-24 y CO-25.
8. **Generación de listados:** CO-05, CO-13, CO-21 y CO-25.

Nótese que todos los componentes, excepto el componente CO-25, deben ser completamente implementados y probados dentro de su incremento. El componente CO-25, al ser el encargado de gestionar la base de datos, debe implementarse parcialmente en cada uno de los incrementos.

6.2.3. Implementación de la Arquitectura

La implementación de la arquitectura MVC en entornos de escritorio Java no es una práctica muy documentada. Por ello, antes de comenzar a implementar el sistema, se estudió la mejor forma de implementar esta arquitectura.

El resultado del estudio dio como resultado una serie de reglas que deben seguirse a la hora de codificar el sistema:

- Si una vista se comunica con otra vista (por ejemplo, una ventana comunicándose con uno de los paneles que la forman), deberá hacerlo a través de una interfaz particular.
- Generalmente, si un usuario genera un evento en una vista, éste debería manejado por su controlador. Sin embargo, si el evento generado tan solo afecta a la vista (por ejemplo,

cambiar el idioma de la interfaz), éste debe ser manejado directamente por la vista, reduciendo así el número de llamadas al controlador.

- Todos los controladores deben implementarse mediante el patrón de diseño singleton²⁵.
- Un controlador puede controlar una o varias vistas, aunque en este proyecto, cada controlador controla una única vista.
- Los controladores se deben comunicar con sus vistas a través de una interfaz particular.

6.2.4. Implementación del Sistema

Para implementar cada uno de los incrementos en los que el sistema está dividido, se llevaron a cabo una serie de tareas:

- **Diseño detallado:** antes de codificar un incremento, se realizaba una pequeña tarea de diseño: se creaba de manera informal (en papel y sin intenciones de mantenimiento) algunos diagramas de clases y secuenciales con el fin de asegurar que la codificación se realizaría sobre una base sólida.

Cabe destacar el especial esfuerzo dedicado a diseñar clases genéricas (abstracciones de diccionarios online, gestores genéricos de base de datos...), facilitando así el proceso de evolución del sistema

Nótese que esta tarea podría considerarse una fase en sí misma (sería una fase, presente en todos los incrementos, previa a la fase de implementación). Sin embargo, su caracterización informal la posiciona más como una tarea de implementación.

- **Diseño IU:** todas las interfaces de usuario necesarias dentro de un incremento se diseñaban en papel, y una vez validadas por el cliente, se codificaban.

²⁵ <http://www.javaworld.com/javaworld/jw-04-2003/jw-0425-designpatterns.html>

- **Codificación:** tarea puramente programacional.
- **Documentación:** toda clase, método y atributo público que se iba codificando, se iba documentando con comentarios javadoc. De esta forma, a día de hoy, se dispone de una documentación de código amplia y de calidad, lo que facilitará enormemente el proceso de mantenimiento y evolución.

Para proporcionar una ligera idea del tamaño y de la organización del código desarrollado, se han recopilado una serie de cifras que se presentan a continuación:

Concepto	Cifra
Número de líneas de código	~9000
Número de clases	155
Número de paquetes	56

Tabla 36 – Tamaño y Organización del Código

6.2.5. Pruebas

Para probar la implementación de cada uno de los incrementos en los que el sistema está dividido, se llevaron a cabo una serie de pruebas:

- **Pruebas unitarias:** a medida que se iban implementando funcionalidades de un incremento, se iban ejecutando pruebas unitarias para verificar el correcto funcionamiento del sistema.
- **Pruebas de usabilidad:** cuando un incremento terminaba de implementarse, éste se sometía a pruebas de usabilidad:
 - Cada incremento era probado por 5 usuarios finales (los mismos usuarios para todos los incrementos).
 - A cada uno de estos usuarios finales se le entregaba un formulario (ver Anexo 1 – Pruebas de Usabilidad1) en el que se le pedía realizar y valorar (desde el punto de vista de la usabilidad) una serie de tareas.

- Durante un periodo de 5 a 15 minutos, cada usuario, por separado, intentaba llevar a cabo las tareas encomendadas.
- Al finalizar este tiempo, se pedía a los usuarios que dejaran de utilizar la aplicación (si aún la estaban usando) y que si tenían sugerencias o mejoras con respecto a las funcionalidades probadas, las escribieran.

Tras recoger los formularios, las respuestas de los usuarios se evaluaban y se decidía si el incremento era usable o no (ver Anexo 1 – Pruebas de Usabilidad1):

- Si lo era, se procedía a realizar la prueba de validación.
- Si no lo era, se hacían los cambios pertinentes y después, se realizaba la prueba de validación.

Cabe destacar que de este tipo de pruebas salieron algunas ideas interesantes, por ejemplo, la de mostrar un mensaje inicial cuando la aplicación se inicie sin ningún diccionario de usuario abierto.

- **Pruebas de validación:** esta prueba consistía en mostrar al cliente el incremento en funcionamiento (una pequeña demo del incremento). Si el cliente consideraba que se habían cumplido los objetivos del incremento, éste se consideraba terminado. Si el cliente observaba que algún objetivo no había sido cumplido, el incremento debía retomarse hasta que todos los objetivos fueran cumplidos.

En el proyecto, todas las pruebas de validación fueron satisfactorias.

7. Planificación Final y Análisis de Costes

En esta sección se proporcionan la planificación final del proyecto y el análisis de costes llevado a cabo en este proyecto.

7.1. Planificación Final

Al comienzo del proyecto se realizó una planificación inicial, en la cual se estimaba el tiempo que las diferentes tareas conllevarían. A medida que el proyecto avanzaba, esta planificación iba sufriendo pequeños cambios:

- La captura de requisitos de software duró 2 días más de lo estimado: determinar la información que un término alemán posee fue una tarea bastante costosa, ya que hubo que estudiar y analizar varios diccionarios de alemán-español y una gramática alemana.
- La implementación y pruebas de la base de datos duraron 1 día menos de lo estimado: la implementación terminó antes de lo previsto gracias al esquema relacional.
- La implementación y pruebas de la aplicación software duraron 9 días más de lo estimado:
 - Aplicar una arquitectura MVC a una aplicación de escritorio no es una práctica muy documentada; por ello, antes de empezar a programar, fue necesario estudiar la mejor forma de implementar dicha arquitectura, tarea que no fue prevista en la planificación inicial.
 - El diseño de las interfaces gráficas de usuario también llevó más tiempo del esperado. En particular, el diseño de la interfaz utilizada para añadir y editar términos (manual o automáticamente) se prolongó bastante debido a la gran cantidad de información heterogénea que debe manejar.

Finalmente, la planificación de este proyecto abarca desde el 1 de abril de 2010 hasta el 21 de septiembre de 2010, en total, 174 días, 14 días (naturales) más de los estimados inicialmente.

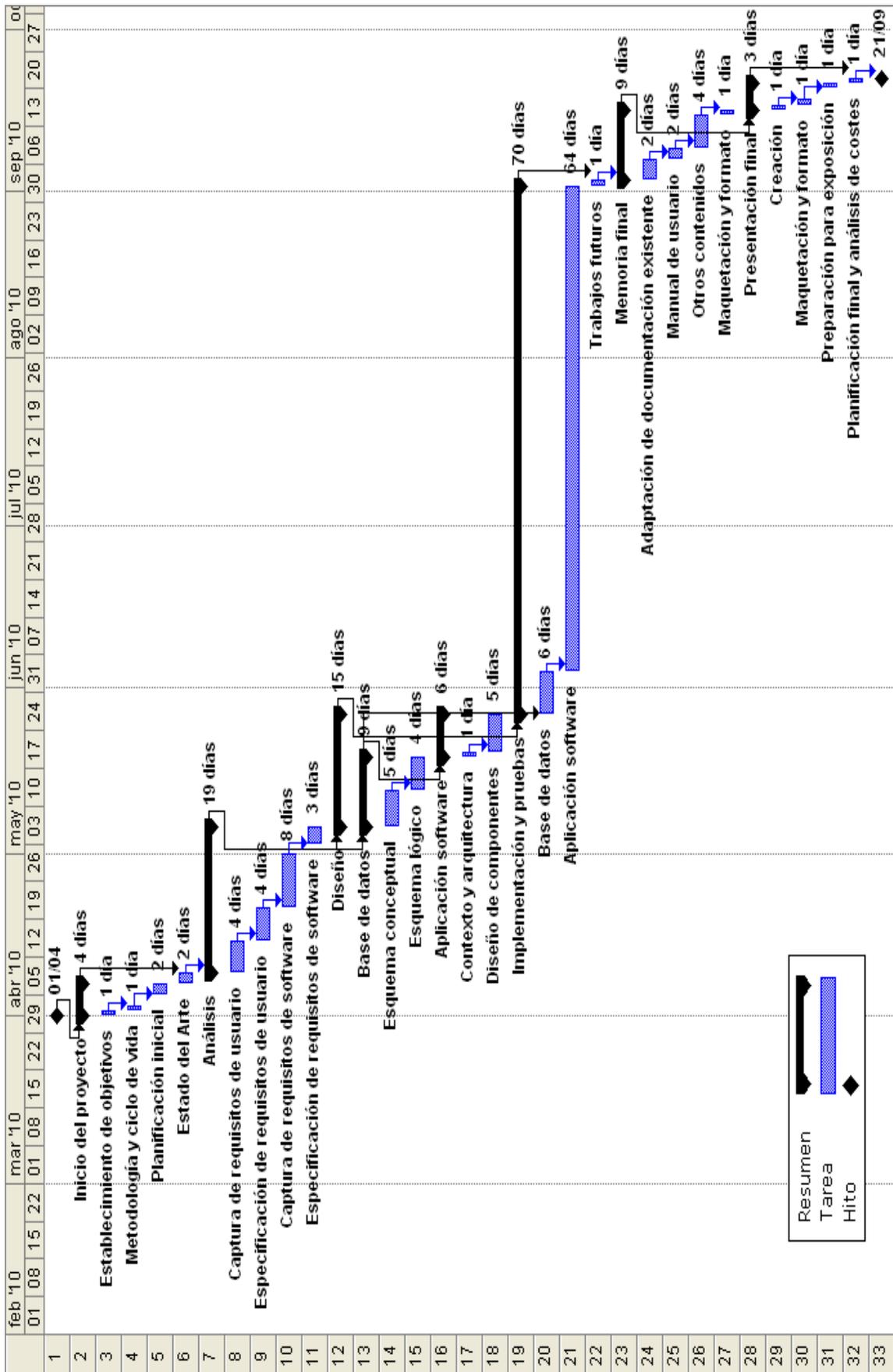


Ilustración 10 - Panificación Final

7.2. Análisis de Costes

Con el fin de valorar este proyecto desde un punto de vista económico, se ha realizado un análisis de costes.

7.2.1. Recursos Humanos

Los recursos humanos utilizados en el proyecto han sido:

- Un gestor de proyecto que trabaja durante todo el proyecto.
- Un analista/diseñador que trabaja en las fases de análisis y diseño.
- Un diseñador de interfaces de usuario y un programador que trabajan, no al mismo tiempo, durante la fase de implementación y pruebas.
- Un traductor para traducir todos los textos de las interfaces de usuario.

Los costes derivados de recursos humanos se especifican en la siguiente tabla:

Rol	€/Hora	Horas	Coste (€)
Gestor de Proyecto	18	992	17.856
Analista/Diseñador	14	248	3.472
Diseñador IU	10	30	300
Programador	9	538	4.842
Traductor ²⁶	-	-	150
Total			26.620

Tabla 37 – Costes en Recursos Humanos

²⁶ El traductor no cobra por horas sino por lotes de trabajo.

7.2.2. Equipamiento

El equipamiento hardware necesario para el desarrollo del proyecto ha sido un ordenador de trabajo (torre, monitor, teclado y ratón) y una impresora. En cuanto al equipamiento software, los únicos productos comercial necesarios han sido un paquete de Microsoft Office 2007, Microsoft Visio 2007 y Microsoft Project 2007.

Los costes asociados a equipamiento hardware y software se detallan a continuación:

Producto	Coste (€)
Ordenador de trabajo	700
Impresora	90
Microsoft Office 2007	139
Microsoft Visio 2007	330
Microsoft Project 2007	775
Total	2.034

Tabla 38 – Costes en Equipamiento

7.2.3. Fungibles

Los gastos en fungibles y material de oficina se especifican en la siguiente tabla:

Producto	Coste (€)
Tóner para impresora	60
Papel	10
CDs	5
Total	75

Tabla 39 – Costes en Fungibles

7.2.4. Otros

En la tabla siguiente se especifican otro tipo de costes que también han sido generados a partir del desarrollo del proyecto:

Concepto	Coste (€)
Transporte	200
Luz, teléfono, internet...	600
Total	800

Tabla 40 – Otros Costes

7.2.5. Coste Total

El coste total del proyecto se presenta desglosado a continuación:

Concepto	Coste (€)
Recursos Humanos	26.620
Equipamiento	2.034
Fungibles	75
Otros	800
Total	29.529

Tabla 41 – Coste Total

Como se puede observar en la tabla anterior, el coste total del proyecto asciende a 29.529 €. Téngase en cuenta que si éste fuera un proyecto comercial, al coste total calculado habría que sumarle el porcentaje de beneficios (por ejemplo, un 10%) que la empresa desarrolladora cobraría.

8. Conclusiones y Trabajos Futuros

En esta sección se ofrecen conclusiones e ideas para mejorar la aplicación desarrollada en este proyecto.

8.1. Conclusiones

El aprendizaje de vocabulario es un proceso costoso y las técnicas que la mayoría de los estudiantes utilizan actualmente para estudiarlo son rudimentarias y poco efectivas.

Hoy en día, en el mercado existen muchos diccionarios, cursos, recopilaciones de vocabulario y otro tipo de recursos; sin embargo, no existe ninguna herramienta que el estudiante pueda utilizar para almacenar, organizar, recuperar, estudiar y ampliar su vocabulario. Es por esto que el proyecto que aquí se presenta resulta innovador y necesario, pues existe una necesidad que debe ser cubierta.

En cuanto al desarrollo del proyecto, me gustaría decir que ha sido un placer llevarlo a cabo: hacía años que tenía la idea en mi mente pero nunca tuve tiempo suficiente para llevarla a la práctica.

Los resultados conseguidos son, en mi opinión y en la de los usuarios finales que participaron en las pruebas de usabilidad (ver sección 6.2.5), excelentes: la aplicación funciona en diversas plataformas, es agradable a la vista, intuitiva, fácil de utilizar, pero ante todo, permite al estudiante de alemán tener a su alcance un completo gestor de vocabulario.

Por último, gracias a la forma en la que el sistema ha sido diseñado (arquitectura MVC), codificado (clases genéricas, bien organizadas y comentadas) y documentado (aplicando prácticas de la metodología de la ESA), el sistema actual resulta altamente escalable: añadir funcionalidades interesantes e incluso adaptar el sistema para gestionar otros idiomas son operaciones que podrían llevarse a cabo de una forma bastante sencilla.

8.2. Trabajos Futuros

A continuación se proponen algunas ideas para ampliar y mejorar la versión actual del sistema gestor:

- Ampliar el criterio de búsqueda de las búsquedas online para permitir al usuario elegir el diccionario online sobre el cual realizar una consulta.
- Ampliar el criterio de búsqueda de las búsquedas avanzadas para permitir al usuario buscar términos con entradas que posean determinadas características gramaticales inherentes a una categoría gramatical particular. De esta forma, se podrían realizar búsqueda como "*verbos irregulares*", "*sustantivos plurales*" o "*pronombres indefinidos*".
- Manejar el concepto de sinónimos y antónimos y permitir relacionar significados entre sí a través de estos conceptos.
- Generar nuevos tipos de ejercicios: asociar frases hechas con sus significados, traducir frases del alemán al español y viceversa (utilizando los ejemplos de uso disponibles), asociar sinónimos, asociar antónimos...
- Asociar a un término con una transcripción fonética e incluso con un archivo de audio que contenga su pronunciación.
- Crear versiones del gestor para otros idiomas o modificar la versión actual para permitir la carga dinámica de módulos especializados en un idioma concreto. Cabe destacar que la mayor parte del sistema puede utilizarse para cualquier idioma.
- Importar los contenidos de un diccionario de usuario en otro diccionario de usuario.
- Crear una extensión para navegadores Web que permita añadir rápidamente (por ejemplo, mediante CTRL+clic) términos encontrados en la Web a diccionarios de usuario existentes.
- Modificar la base de la aplicación para convertirla en una herramienta colaborativa.

9. Glosario de Términos y Acrónimos

Término/Acrónimo	Definición
A.K.A.	<p>Also Known As (traducido al español, También Conocido Como).</p> <p>Expresión utilizada para indicar una segunda forma de referirse a algo o alguien.</p>
Aserción	<p>En el contexto de las bases de datos, una aserción es un tipo de restricción en el que el valor de un atributo depende del valor de uno o más atributos de la misma tupla.</p>
Características gramaticales	<p>Algunos ejemplos de características gramaticales son la categoría gramatical (nombre, adjetivo, verbo...), el género de un nombre y la regularidad de un verbo.</p>
Características de uso	<p>Algunos ejemplos de características de uso son el registro lingüístico (términos formales, informales...) y el uso geográfico.</p>
CD	<p>Compact Disc (traducido al español, Disco Compacto).</p>
CO	<p>Componente.</p> <p>En el contexto del desarrollo software, un componente es una unidad de composición que especifica las interfaces y las dependencias contextuales que éste posee.</p>
Comentarios javadoc	<p>Comentarios (para código Java) con un formato especial que tienen como objetivo principal describir clases y métodos.</p>

Término/Acrónimo	Definición
Comprobación	En el contexto de las bases de datos, una comprobación es un tipo de restricción en el que un atributo sólo puede tomar valores de un determinado conjunto de valores.
Diagrama E/R	Diagrama Entidad / Relación. Tipo de diagrama conceptual. Proporciona una representación gráfica, más o menos completa, de los datos de un determinado dominio y la forma en la que éstos están relacionados.
Diccionario online	Diccionario u otro tipo de recurso disponible en Internet que permita obtener información relacionada con términos alemanes.
Diccionario de usuario	Unidad lógica utilizada por el sistema para gestionar todo el vocabulario de un usuario.
Disparador	En el contexto de las bases de datos, un disparador es una función que se ejecuta al insertar, actualizar y/o borrar una tupla.
Dominio	En el contexto de las bases de datos, un dominio es el conjunto de valores que puede tomar uno o varios atributos.
EER	Extended Entity/Relationship (traducido al español, Entidad/Relación Extendido). Es un tipo de modelo conceptual de datos, cuyo uso está muy extendido a nivel internacional.

Término/Acrónimo	Definición
ESA	<p>European Space Agency (traducido al español, Agencia Espacial Europea).</p> <p>La Agencia Espacial Europea, aparte de ser la puerta de acceso al espacio del continente europeo, posee una serie de estándares de aplicación ingeniería del software. Hoy en día, estos estándares se utilizan mucho en proyectos de ámbito europeo.</p>
Etiqueta organizativa	<p>Todos los términos que comparten alguna etiqueta organizativa están relacionados de alguna manera. La razón más habitual para usar etiquetas organizativas es la clasificación de términos por campos semánticos.</p>
Grafo relacional	<p>Tipo de diagrama lógico. Proporciona una representación gráfica, más o menos completa, de la estructura de una base de datos en términos de estructuras de datos procesables por un determinado tipo de SGBD.</p>
IU	<p>Interfaces de Usuario.</p>
Marca "Estudiar"	<p>Dentro del contexto de este proyecto, los significados poseen una marca o atributo que indica si el usuario desea estudiar un significado concreto. La utilidad de esta marca está en que a la hora de realizar búsquedas avanzadas de términos se puede incluir este atributo como parte del criterio de búsqueda.</p>

Término/Acrónimo	Definición
Matriz de rastreabilidad	En este proyecto, la matriz de rastreabilidad se ha utilizado para mostrar las relaciones existentes entre requisitos de software y componentes.
MB	Megabyte.
MHz	Megahercio.
Multiplataforma	Se dice que una aplicación es multiplataforma si puede funcionar correctamente en diversas plataformas.
PB	Petabyte. Equivale a 10^{15} bytes.
PDF	Portable Document Format (traducido al español, Formato de Documento Portátil).
RAM	Random Access Memory (traducido al español, Memoria de Acceso Aleatorio). Memoria volátil que posee el ordenador para guardar programas y datos en uso.
Registro lingüístico	Los significados de un término pueden considerarse formales, coloquiales, vulgares o neutros. Estas características de uso se conocen como registro lingüístico.
Restricción geográfica	Cuando el significado de un término sólo se emplea en determinadas zonas geográficas, se dice que posee una restricción geográfica.
RS	Requisito de Software. Necesidad del cliente y/o usuario final que posee un nivel de detalle elevado.
RSF	Requisito de Software Funcional.

Término/Acrónimo	Definición
RSNF	Requisito de Software No-Funcional.
SGBD	Sistema Gestor de Bases de Datos. Sistema cuyo objetivo es gestionar un determinado tipo de bases de datos. Los SGBD más comunes son los que se encargan de gestionar bases de datos relacionales.
Singleton	Patrón de diseño software, también conocido como <i>instancia única</i> , que impide a una clase ser instanciada varias veces.
SQL	Structured Query Language (traducido al español, Lenguaje de Consulta Estructurado). Lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre éstas.
Supuesto semántico	Información del esquema conceptual/lógico que no puede ser contemplada en el diagrama conceptual/lógico debido a limitaciones de la notación.
Término	En el contexto del vocabulario, un término se define como una palabra y toda la información necesaria para poder utilizarla de forma correcta en cualquier contexto.
Tupla	En el contexto de las bases de datos, una tupla es una fila de una tabla.

Tabla 42 – Glosario de Términos y Acrónimos

10. Bibliografía

La bibliografía utilizada durante el desarrollo de este proyecto es la siguiente:

1. **European Space Agency.** ESA Software Engineering & Standardisation. [En línea]
http://www.esa.int/TEC/Software_engineering_and_standardisation/.

2. **Extreme Programming.** Extreme Programming: A gentle introduction. [En línea] <http://www.extremeprogramming.org/>.

3. **ESA Software Engineering & Standardisation.** Guide to the User Requirements Definition Phase. [En línea]
<ftp://ftp.estec.esa.nl/pub/wm/wme/bssc/PSS0502.pdf>.

4.—. Guide to the Software Requirements Definition Phase. [En línea] <ftp://ftp.estec.esa.nl/pub/wm/wme/bssc/PSS0503.pdf>.

5. **de Miguel, Adoración, Piattini, Mario y Marcos, Esperanza.** *Diseño de Bases de Datos Relacionales*. s.l. : Ra-Ma, 1999.

6. **Elmasri, R. y Navathe, S.B.** *Fundamentals of Database Systems*. s.l. : Pearson/Addison Wesley, 2007.

7. **ESA Software Engineering & Standardisation.** Guide to the Software Architectural Design Phase. [En línea]
<ftp://ftp.estec.esa.nl/pub/wm/wme/bssc/PSS0504.pdf>.

8. **Agile Modeling.** UML 2 Component Diagrams. [En línea]
<http://www.agilemodeling.com/artifacts/componentDiagram.htm>.

Anexo 1 – Pruebas de Usabilidad

En este anexo, se muestra, a modo de ejemplo, el formulario utilizado en las pruebas de usabilidad del incremento 3 (gestión de términos) y el proceso de evaluación de las respuestas proporcionadas por los usuarios finales para dicho formulario:

Dispone de 10 minutos para intentar realizar las tareas listadas abajo. Intente realizar el máximo número de tareas, pero no se preocupe si no le da tiempo a realizar alguna.

Valore todas las tareas que haya realizado o intentado realizar. Si no ha tenido tiempo de empezar alguna tarea, por favor, no la valore.

Para la valoración de tareas, por favor use la siguiente escala:

- 1** – No supe cómo realizar la tarea.
- 2** – Realicé la tarea aunque el mecanismo me resultó poco intuitivo.
- 3** – Realicé la tarea aunque tuve problemas con la disposición de los componentes gráficos.
- 4** – Realicé la tarea sin problemas.

Tareas	Valoración
1. Cree un nuevo diccionario de usuario llamado <i>MiDiccionario</i> .	
2. Añada manualmente el siguiente término al diccionario que acaba de crear: <i>Haus</i> ⇔ sustantivo neutro singular, con plural; su nominativo plural es <i>Häuser</i> y su genitivo singular es <i>Hauses</i> ; su significado principal es <i>casa</i> y puede usarse en cualquier país de habla alemana y en cualquier situación.	
3. Edite el término <i>Haus</i> de tal forma que su significado <i>casa</i> tenga una etiqueta organizativa llamada <i>Edificio</i> .	
4. Elimine el término <i>Haus</i> del diccionario de usuario actual.	

Comentarios:

A partir de las respuestas de los usuarios, se calculan la media aritmética y la desviación típica de las valoraciones de cada una de las tareas. Estas dos cifras se utilizan entonces para evaluar la usabilidad de las funcionalidades del incremento:

- Si media aritmética ≥ 3.5 y desviación típica $\leq 0.6 \Rightarrow$ la funcionalidad inherente a la tarea se considera usable.
- Si media aritmética ≥ 2.5 y < 3.5 ó media aritmética ≥ 3.5 y desviación típica $> 0.6 \Rightarrow$ la funcionalidad inherente a la tarea se considera usable pero también mejorable. Se estudian las posibles mejoras y se decide si se debe o no cambiar algo.
- Si media aritmética $< 2.5 \Rightarrow$ la funcionalidad no es usable. Las interfaces gráficas relacionadas con la funcionalidad deben rediseñarse para mejorar la usabilidad.

Tarea	Valoración		Evaluación
	Media	Desviación	
1	3,8	0,447214	Usable
2	3,4	0,547723	Usable pero mejorable
3	3,4	0,547723	Usable pero mejorable
4	3,6	0,547723	Usable

Tabla 43 – Evaluación de Prueba de Usabilidad

Anexo 2 – Manual de Usuario

Instalación

La aplicación no requiere ningún tipo de instalación. Para poder ejecutarla, tan solo es necesario que su ordenador disponga del entorno de ejecución Java, el cual se puede descargar gratuitamente de Internet²⁷.

Ejecución y Salida de la Aplicación

Existen varias formas de ejecutar la aplicación:

- Si dispone de un sistema operativo con interfaz gráfica, puede ejecutar la aplicación haciendo doble clic sobre el ejecutable.
- Si, por algún motivo, no tiene, no puede o no desea utilizar la interfaz gráfica de su sistema operativo, puede ejecutar la aplicación desde línea de comandos utilizando el comando `java <nombre-del-ejecutable>`.

Al ejecutar la aplicación, la ventana principal del sistema aparecerá.

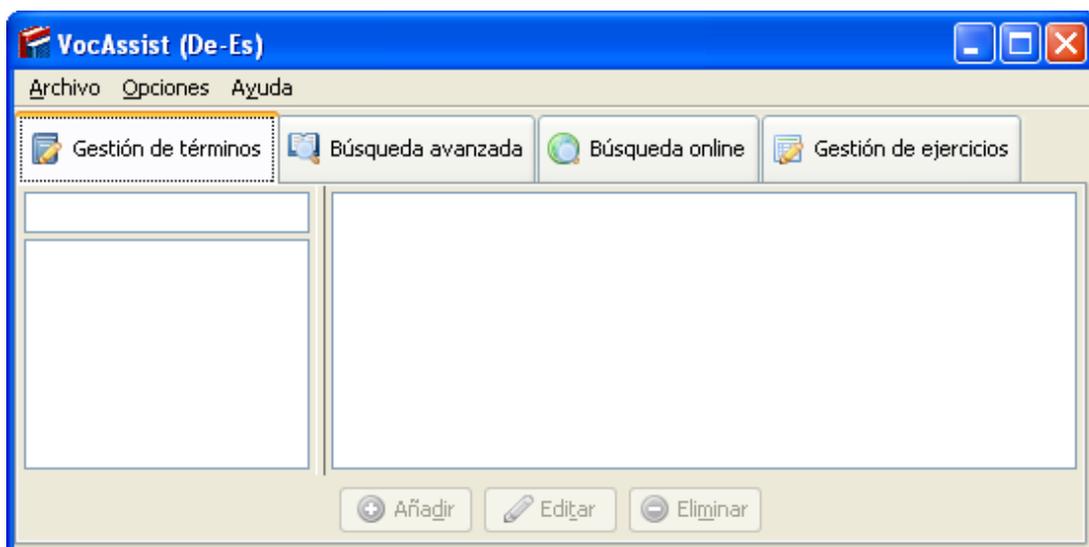


Ilustración 11 – Ventana Principal

²⁷ <http://www.java.com/es/download/>

Si es la primera vez que ejecuta la aplicación, se mostrará un mensaje de información advirtiéndole al usuario que hasta que no abra o cree un nuevo diccionario de usuario, la mayor parte de las funcionalidades del sistema no estarán disponibles.

Para salir de la aplicación, puede utilizar el botón cerrar de la barra de títulos, el menú de la aplicación *Archivo + Salir*, o la combinación de teclas *CTRL+Q*.

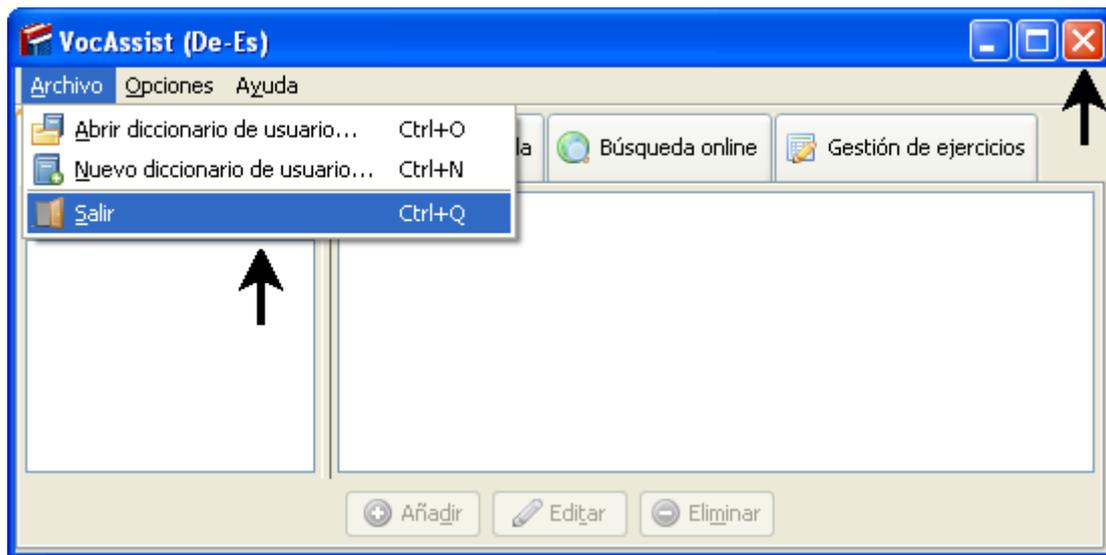


Ilustración 12 – Salir de la Aplicación

Gestión de Diccionarios de Usuarios

Para poder gestionar vocabulario, es necesario disponer de un diccionario de usuario (un diccionario de usuario es un directorio especial en el que se guarda vocabulario).

Si ya dispone de un diccionario de usuario, puede abrirlo a través del menú de la aplicación *Archivo + Abrir diccionario de usuario...*, ó mediante la combinación de teclas *CTRL+O*.

Si no dispone de ningún diccionario de usuario, puede crear uno a través del menú de la aplicación *Archivo + Nuevo diccionario de usuario...*, ó mediante la combinación de teclas *CTRL+N*.

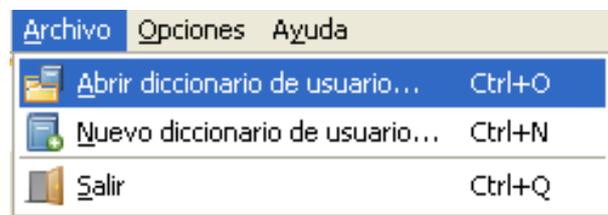


Ilustración 13 – Menú Abrir Diccionario de Usuario

Nótese que al crearse un diccionario de usuario nuevo, éste se abre automáticamente.

Cuando un diccionario de usuario está abierto, su nombre aparece en la barra de títulos de la ventana principal, entre corchetes.

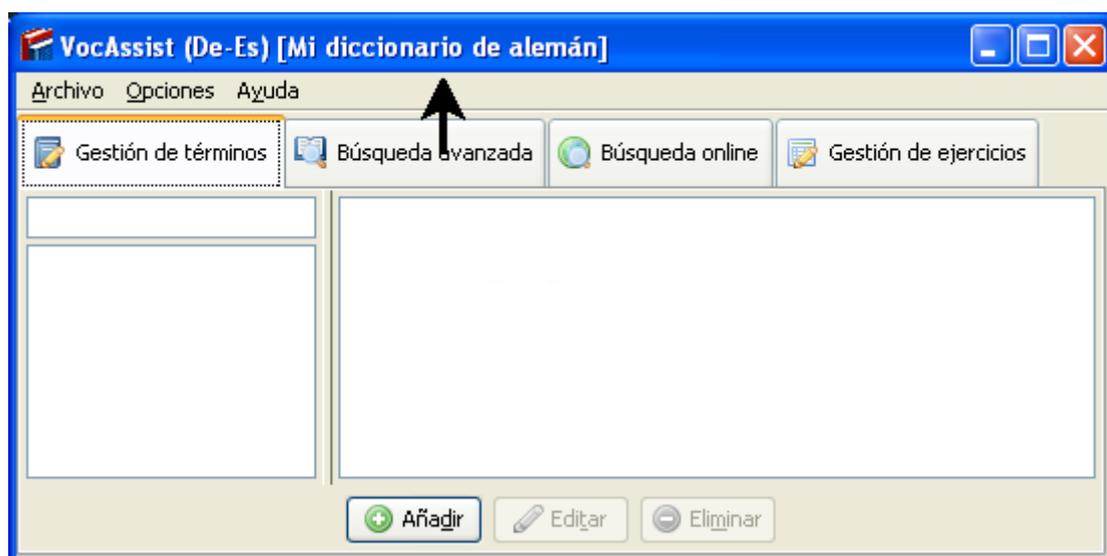


Ilustración 14 – Diccionario de Usuario Abierto

Gestión de Términos

Si desea añadir, editar, eliminar, visualizar o buscar términos en el diccionario de usuario abierto, seleccione la pestaña *Gestión de términos* de la ventana principal. Para ello, puede hacer clic sobre la pestaña o pulsar la combinación de teclas *ALT+1*.

Añadir Términos

Para añadir un nuevo término, pulse el botón *Añadir*. Esto provocará la aparición de un cuadro de diálogo que le permitirá especificar la información del nuevo término. Para detalles de cómo especificar dicha información, ver sección Cuadro de Diálogo Añadir/Editar Término.

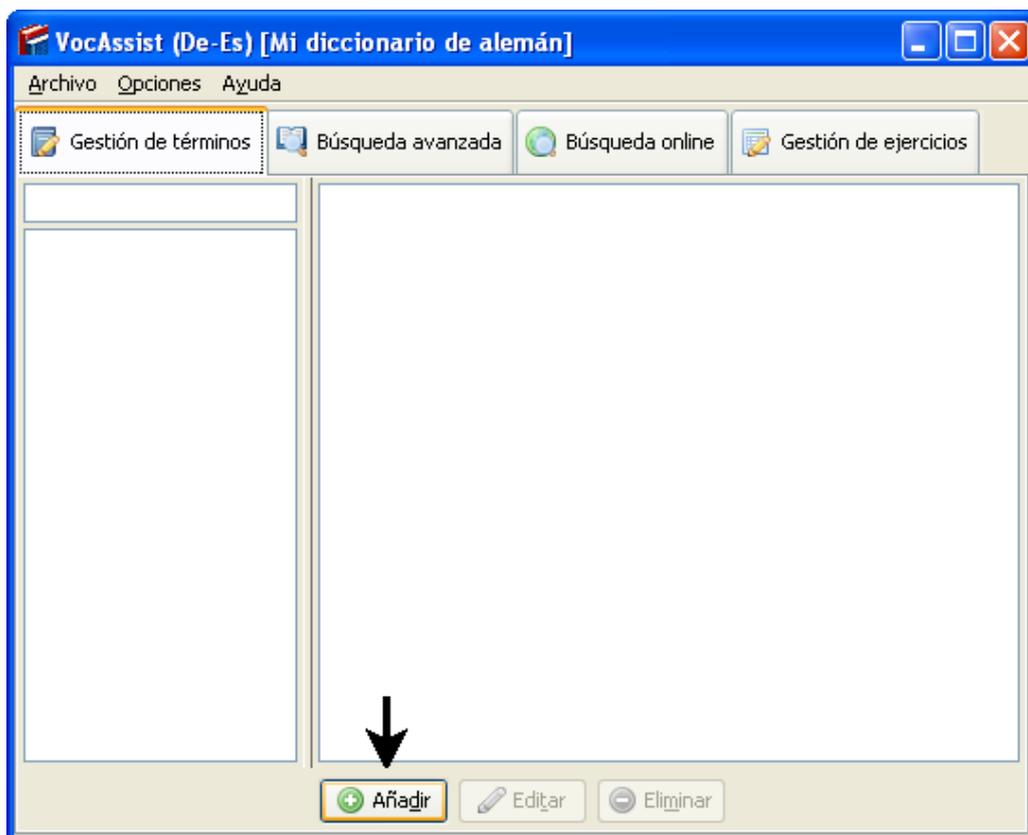


Ilustración 15 – Añadir Término

Editar Términos

Para editar un término existente, en la lista de términos, seleccione el término que desee editar y pulse el botón *Editar*. Esto provocará la aparición de un cuadro de diálogo que le permitirá editar la información del término existente. Para detalles de cómo editar dicha información, ver sección Cuadro de Diálogo Añadir/Editar Término.

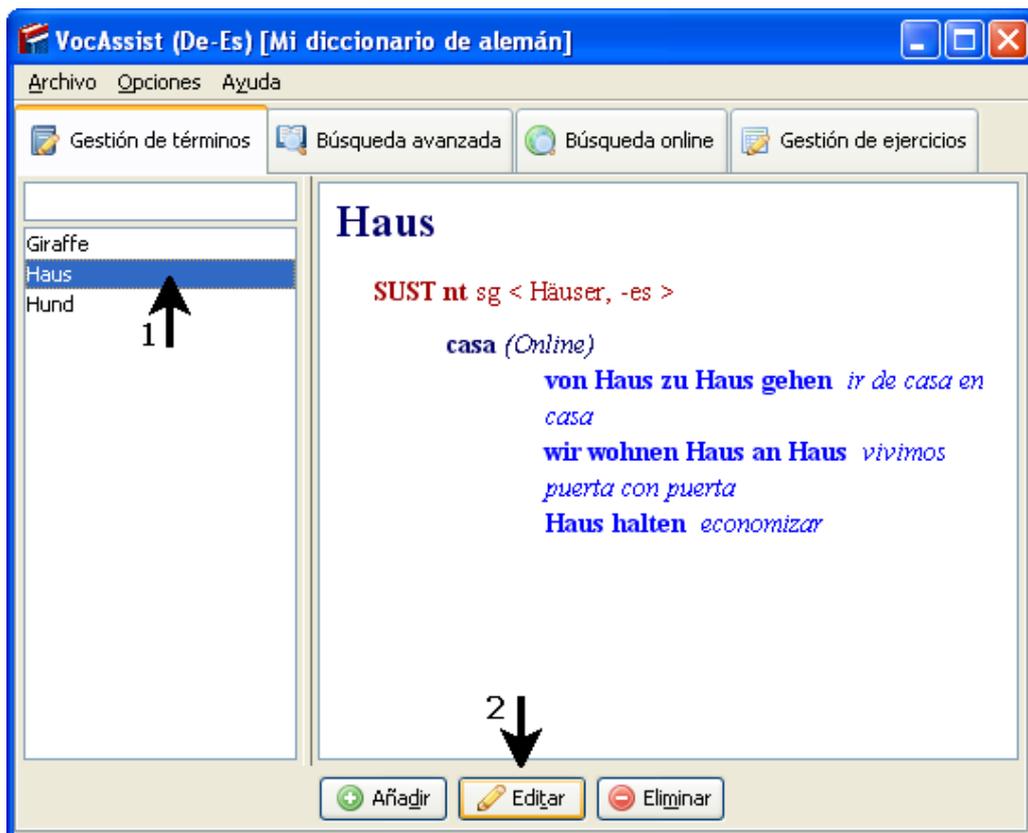


Ilustración 16 – Editar Término

Eliminar Términos

Para eliminar un término existente, en la lista de términos, seleccione el término que desee editar, pulse el botón *Eliminar* y pulse *Sí* en el mensaje de confirmación que aparecerá.

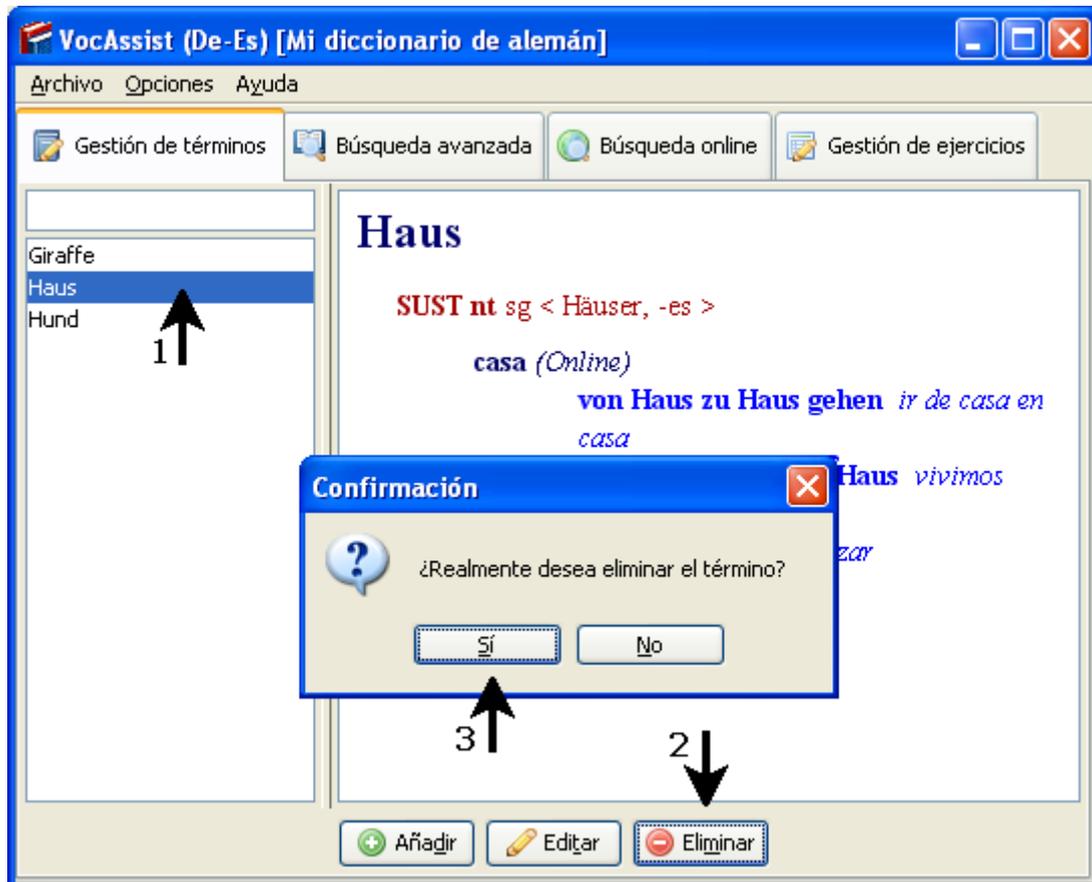


Ilustración 17 – Eliminar Término

Visualizar/Buscar Términos

Para visualizar un término, selecciónelo de la lista de términos o búsquelo utilizando la caja de búsquedas. Si utiliza la segunda opción, simplemente escriba el término que desee visualizar/buscar; tan pronto como éste se encuentre, será seleccionado de la lista de términos y visualizado.

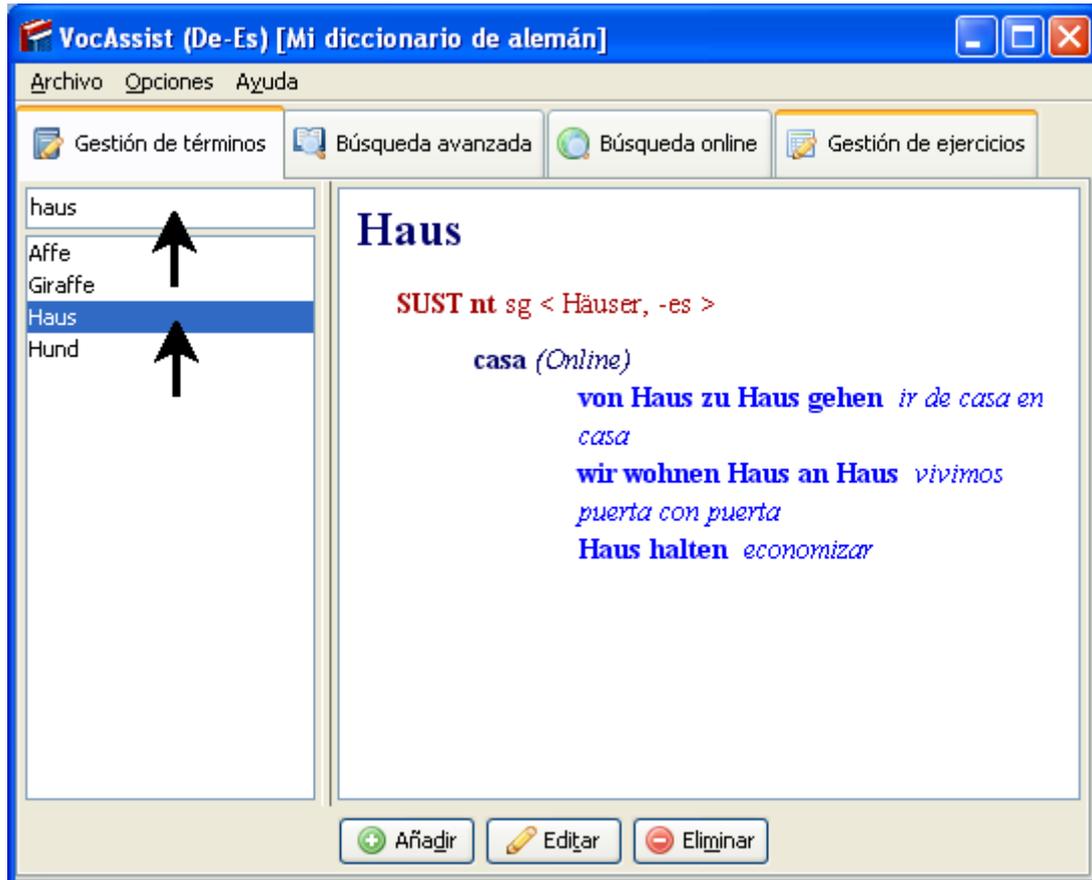


Ilustración 18 – Visualizar/Buscar Términos

Búsqueda Avanzada

Si desea buscar términos que cumplan un determinado criterio de búsqueda o generar listados de vocabulario, seleccione la pestaña *Búsqueda Avanzada* de la ventana principal. Para ello, puede hacer clic sobre la pestaña o pulsar la combinación de teclas ALT+2.

Buscar Términos

Para buscar todos los términos que cumplan un determinado criterio, especifique un criterio de búsqueda (por ejemplo, términos que contengan el texto "affe" y que estén etiquetados como "animales") y pulse el botón *Buscar*:

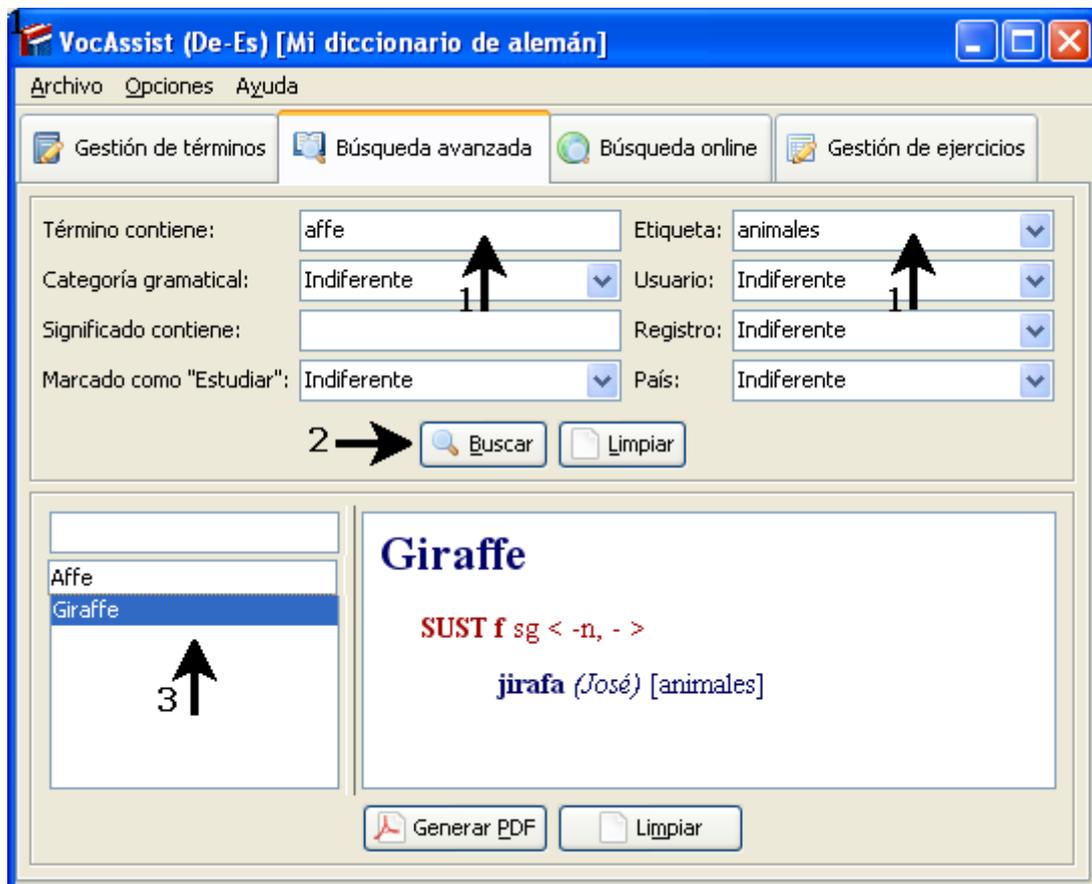


Ilustración 19 – Búsqueda Avanzada de Términos

- Si la búsqueda obtiene resultados, estos se mostrarán en el panel de resultados. Si desea borrar dichos resultados, pulse el botón *Limpiar* situado al lado del botón *Generar PDF*.

- Si la búsqueda no obtiene resultados, un mensaje informativo se mostrará y el panel de resultados se mostrará en blanco.

Si desea especificar un nuevo criterio de búsqueda desde cero, pulse el botón *Limpiar* situado al lado del botón *Buscar*.

Generar Listados PDF

A partir de los resultados de una búsqueda avanzada, se puede generar un listado de vocabulario en formato PDF. Para hacerlo, tan sólo tiene que pulsar el botón *Generar PDF* y en el selector de archivos que se abrirá, elija la ruta en la que guardar el listado PDF y pulse el botón *Guardar*.

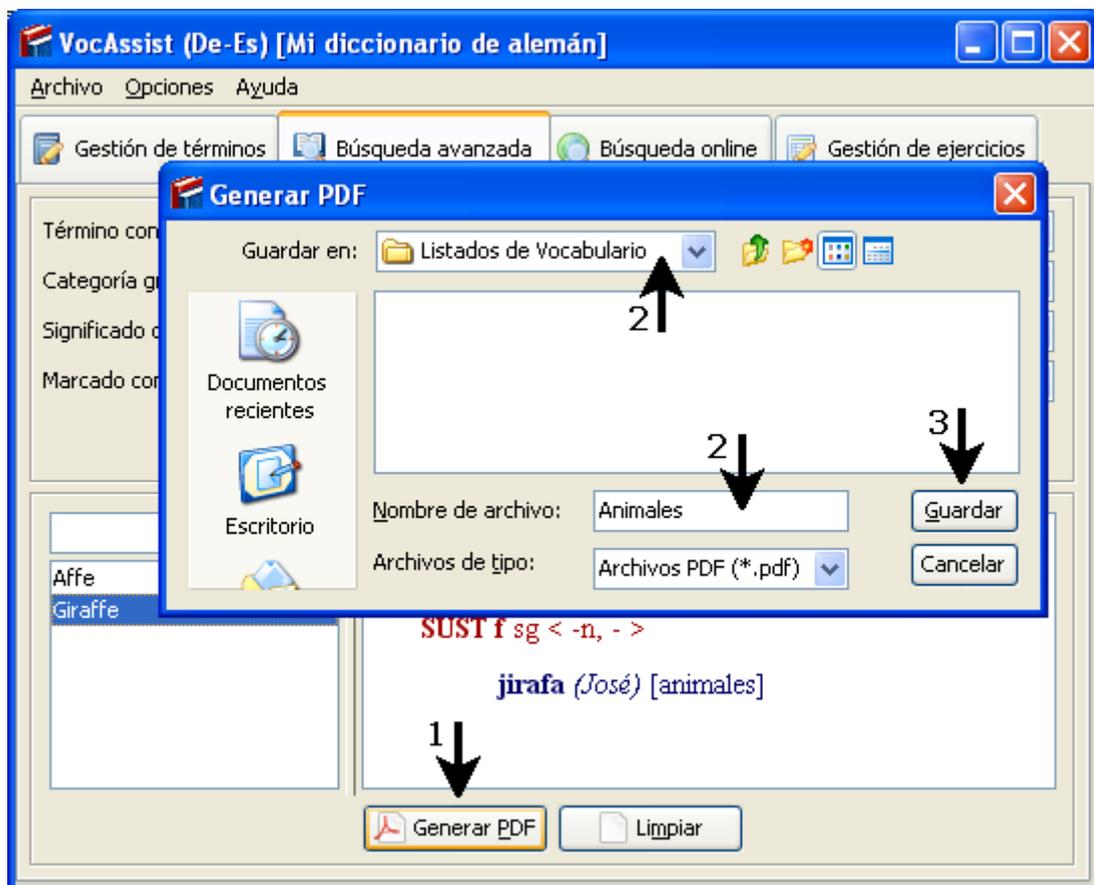


Ilustración 20 – Generar Listado PDF

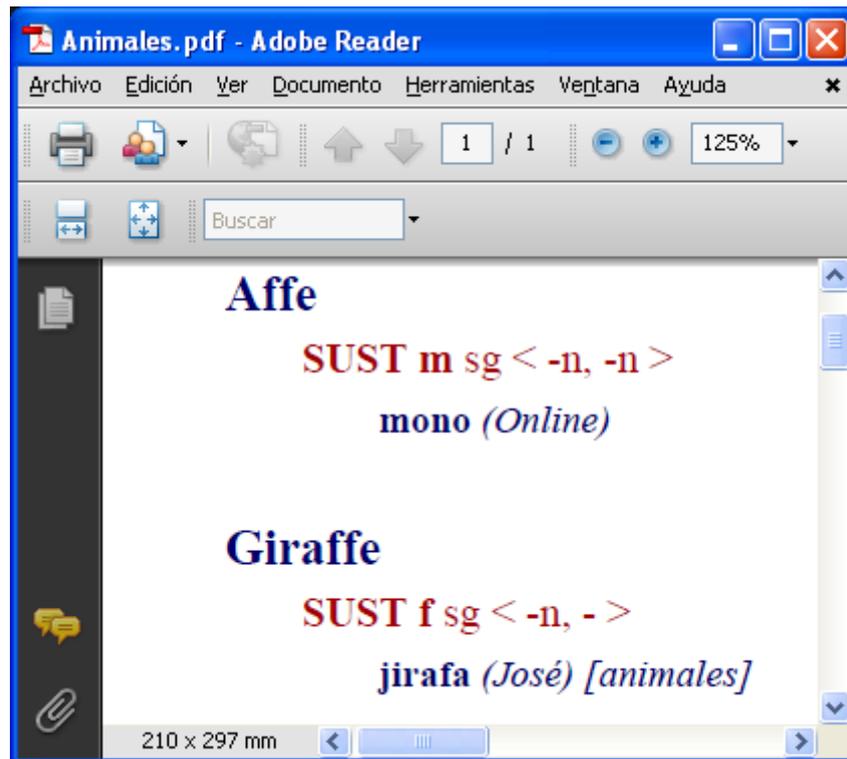


Ilustración 21 – Listado PDF Generado

Búsqueda Online

Si desea buscar un término en un diccionario online, seleccione la pestaña *Búsqueda Online* de la ventana principal. Para ello, puede hacer clic sobre la pestaña o pulsar la combinación de teclas ALT+3.

Buscar Término

Especifique el término que desee buscar en el diccionario online y pulse el botón *Buscar*:



Ilustración 22 – Búsqueda Online de Término

- Si la búsqueda obtiene resultados, estos se mostrarán en el panel de resultados. Si desea borrar dichos resultados, pulse el botón *Limpiar* situado al lado del botón *Añadir*.
- Si la búsqueda no obtiene resultados, un mensaje informativo se mostrará y el panel de resultados se mostrará en blanco.

Si desea borrar el término especificado en la caja de búsqueda, pulse el botón *Limpiar* situado al lado del botón *Buscar*.

Añadir Término Online

El término obtenido a partir de una búsqueda online puede ser añadido al diccionario de usuario abierto pulsando el botón *Añadir*. Al pulsar este botón, se mostrará el cuadro de diálogo para añadir y editar términos (ver sección Cuadro de Diálogo Añadir/Editar Término) y se cargará en él la información del término:

- Si el término online no existe en el diccionario de usuario abierto, toda su información será cargada.
- Si el término online ya existe en el diccionario de usuario abierto, se cargará la información guardada en el diccionario de usuario y se completará con la información del término online.

Edite la información del término a su gusto y pulse el botón *Guardar cambios*.

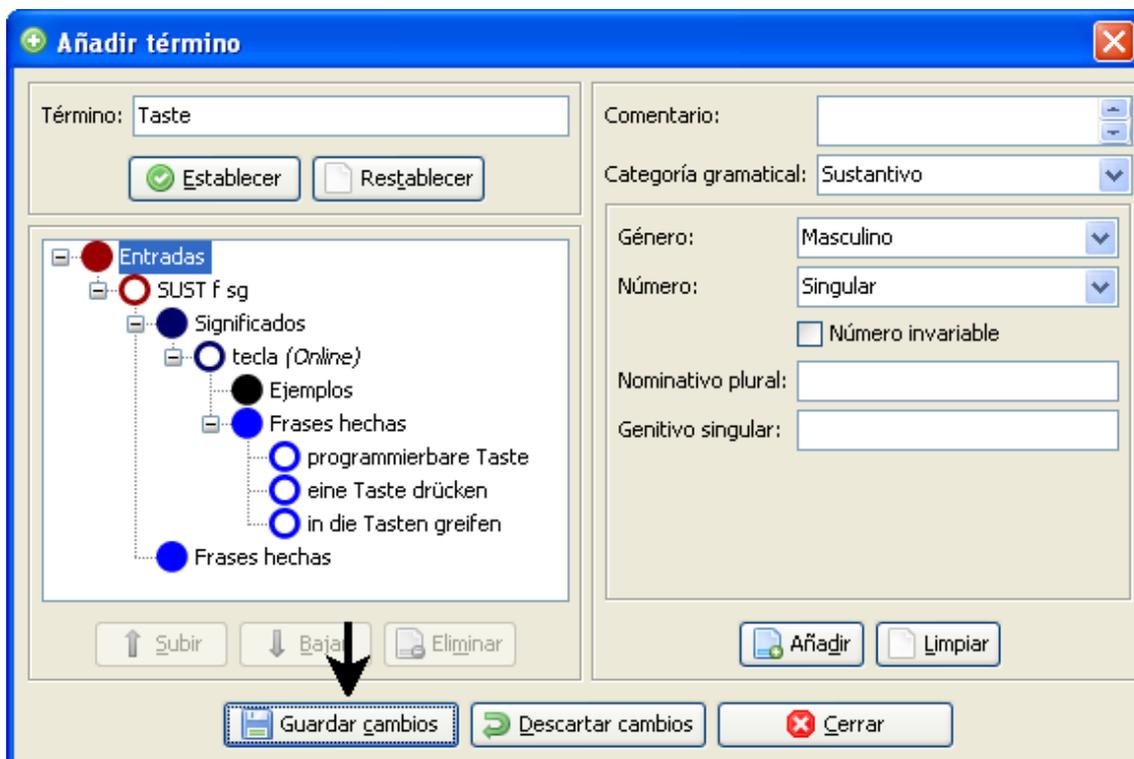


Ilustración 23 – Añadir Término Online

Gestión de Ejercicios

Si desea realizar ejercicios de vocabulario, seleccione la pestaña *Gestión de Ejercicios* de la ventana principal. Para ello, puede hacer clic sobre la pestaña o pulsar la combinación de teclas ALT+4.

Generar Ejercicios

Para generar un ejercicio basado en todos los términos que cumplan un determinado criterio, especifique un criterio de búsqueda (por ejemplo, sustantivos etiquetados como "animales"), seleccione el tipo de ejercicio a generar, el número máximo de preguntas que el ejercicio debe tener y pulse el botón *Generar*:

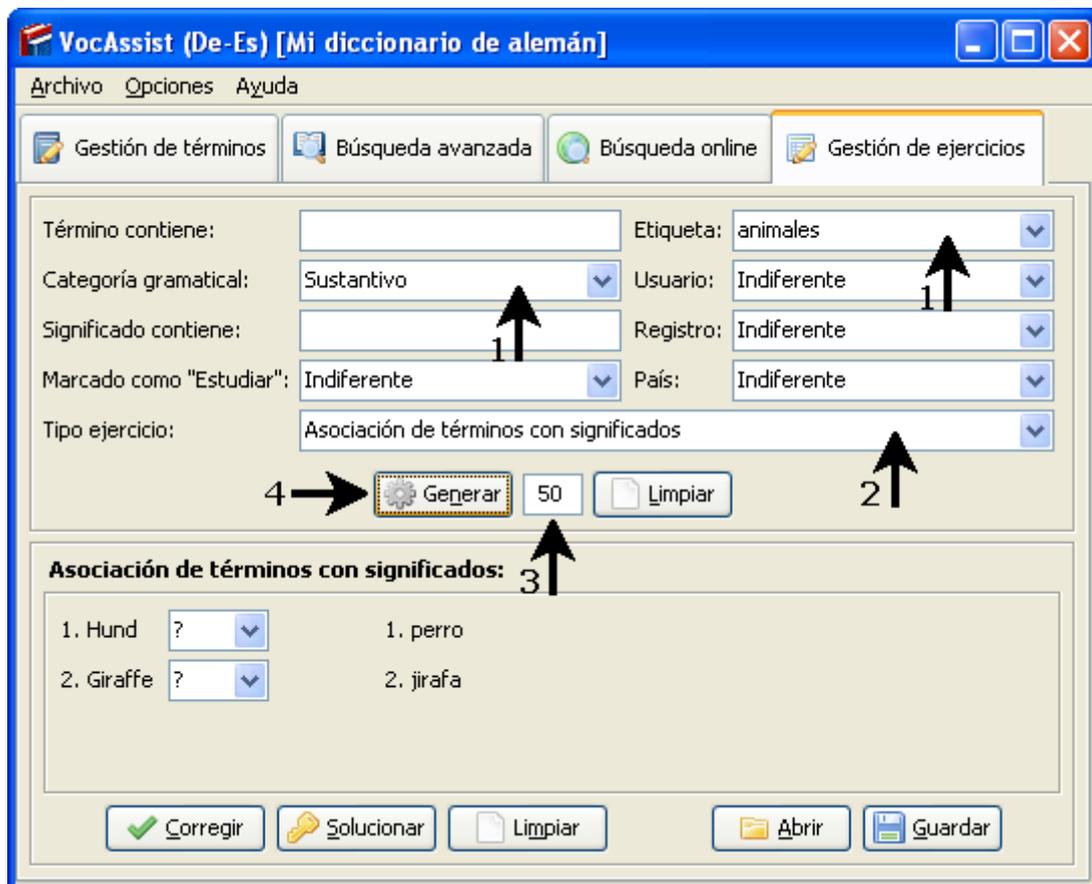


Ilustración 24 – Generar Ejercicio

- Si la búsqueda obtiene resultados útiles para la generación del ejercicio, el ejercicio se generado y se carga en el panel de ejercicios. Si desea eliminar el ejercicio, pulse el botón *Limpiar* situado al lado del botón *Solucionar*.

- Si la búsqueda no obtiene resultados útiles para la generación del ejercicio, un mensaje informativo se mostrará y el panel de ejercicios se mostrará en blanco.

Si desea especificar un nuevo criterio de búsqueda desde cero, pulse el botón *Limpiar* situado al lado del botón *Buscar*.

Resolver, Corregir y Solucionar Ejercicios

Resuelva el ejercicio intentando responder correctamente a cada pregunta.

Cuando haya terminado, compruebe si sus respuestas son correctas pulsando el botón *Corregir*.

Si desea que el sistema le muestre las soluciones a las preguntas que no supo contestar o que contestó incorrectamente, pulse el botón *Solucionar*.

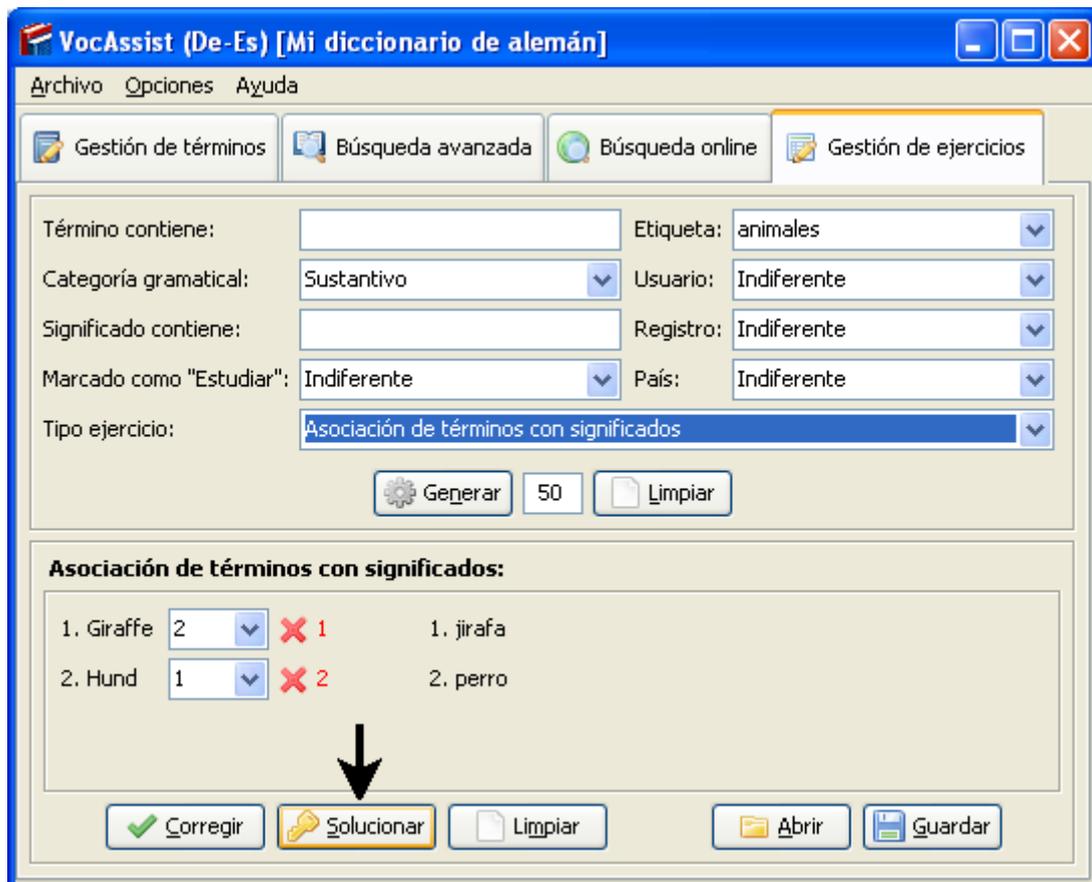


Ilustración 25 – Solucionar Ejercicio

Abrir y Guardar Ejercicios

Si desea abrir un ejercicio guardado en disco, pulse el botón *Abrir* y en el selector de archivos que se abrirá, elija el ejercicio que desee abrir y pulse el botón *Abrir*.

Si desea guardar un ejercicio, pulse el botón *Guardar* y en el selector de archivos que se abrirá, elija la ruta en la que guardar el ejercicio y pulse el botón *Guardar*.

Gestión de Opciones

Cambiar Idioma de la Interfaz

Si desea cambiar el idioma de la interfaz, utilice el menú de la aplicación *Opciones + Idioma + (Español / Alemán)*.

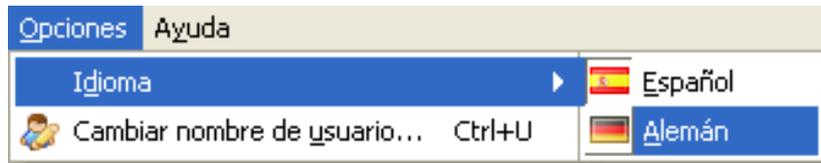


Ilustración 26 – Menú Cambiar Idioma

Cambiar Nombre del Usuario

Si desea cambiar el nombre del usuario de la aplicación, abra el cuadro de diálogo con tal fin. Para ello, puede utilizar el menú de la aplicación *Opciones + Cambiar nombre de usuario...* ó la combinación de teclas *CTRL+U*.

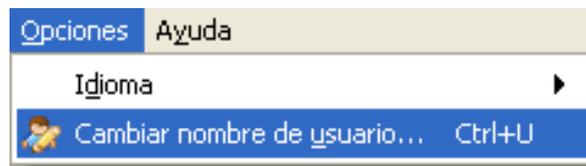


Ilustración 27 – Menú Cambiar Nombre de Usuario

Dentro del cuadro de diálogo, escriba el nuevo nombre del usuario y pulse el botón *Guardar*. Puede utilizar cualquier nombre de usuario excepto "Online", pues éste está reservado como nombre de usuario para la información proveniente de términos online.

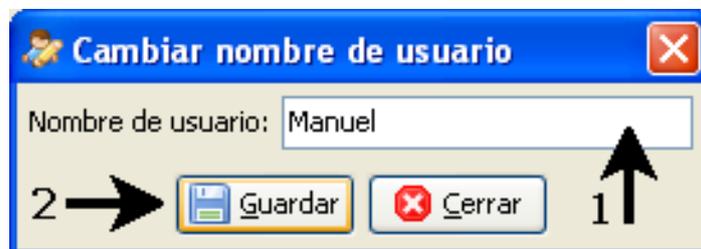


Ilustración 28 – Cambiar Nombre de Usuario

Cuadro de Diálogo Añadir/Editar Término

El cuadro de diálogo que permite añadir y editar términos está dividido en cuatro áreas:

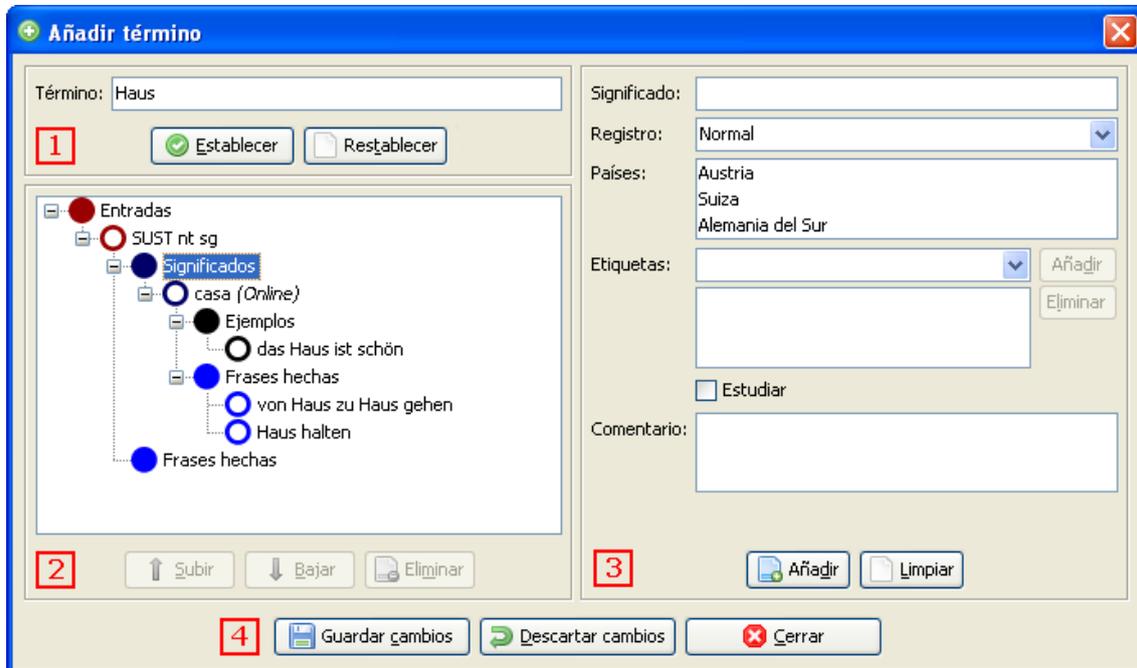


Ilustración 29 – Cuadro de Diálogo Añadir/Editar Término

1. **Establecimiento del término:** permite identificar un término con una palabra o conjunto de éstas (por ejemplo, "Haus").
 - Si se está añadiendo un nuevo término, la caja de texto etiquetada como *Término* aparecerá inicialmente vacía. Escriba en ella la(s) palabra(s) que desee usar como identificador del término y pulse el botón *Establecer*. Si todo va bien, se mostrará un mensaje de confirmación. Si hubo algún problema, se mostrará un mensaje de error detallando la causa.
 - Si se está editando un término existente, la caja de texto etiquetada como *Término* contendrá la(s) palabra(s) que lo identifican. Si las desea modificar, hágalo y pulse el botón *Establecer*. Si todo va bien, se mostrará un mensaje de confirmación. Si hubo algún problema, se mostrará un mensaje de error detallando la causa.

2. **Árbol de información:** muestra en una estructura de árbol la información más relevante del término. Cada nodo del árbol sirve o bien para añadir un nuevo elemento de información (entrada, significado, ejemplo de uso o frase hecha) o bien para editar uno existente.

- Los nodos etiquetados como *Entradas*, *Significados*, *Ejemplos* y *Frases hechas*, llamados nodos de adición, se utilizan para añadir nuevos elementos de información. Al seleccionar alguno de estos nodos, el panel de elementos de información (área 3) cambia para permitir añadir el elemento seleccionado.
- El resto de nodos (llamados nodos de edición) representan elementos de información existentes. Al seleccionarlos, el panel de elementos de información cambia para permitir editar el elemento seleccionado.
- Para ordenar los elementos de información a su gusto, puede subir y bajar un elemento de información utilizando los botones *Subir* y *Bajar*, respectivamente.
- Para eliminar un elemento de información, utilice el botón *Eliminar*.

3. **Panel de elementos de información:** permite añadir y editar elementos de información (entradas, significados, ejemplos de uso o frases hechas).

- Si en el árbol de información hay un nodo de adición seleccionado, este panel le permitirá añadir un nuevo elemento de información.

Para añadirlo, proporcione la información oportuna y pulse el botón *Añadir*. Si todo va bien, se mostrará un mensaje de confirmación. Si hubo algún problema, se mostrará un mensaje de error detallando la causa.

Si aún no ha añadido el elemento de información y desea empezar a proporcionar la información desde cero, puede utilizar el botón *Limpiar* situado al lado del botón *Añadir*.

- Si en el árbol de información hay un nodo de edición seleccionado, este panel le permitirá editar el elemento de información seleccionado.

Para hacerlo, realice los cambios oportunos y pulse el botón *Guardar*. Si todo va bien, se mostrará un mensaje de confirmación. Si hubo algún problema, se mostrará un mensaje de error detallando la causa.

Si aún no guardado los cambios y desea empezar a editar el elemento de información desde cero, puede utilizar el botón *Restablecer* situado al lado del botón *Guardar*.

4. **Fin de operación:** los cambios realizados en el cuadro de diálogo no son realmente guardados en el diccionario de usuario hasta que no se pulsa el botón *Guardar cambios*. Si aún no ha guardado los cambios en el diccionario de usuario y desea empezar a añadir y editar elementos de información desde cero, puede utilizar el botón *Deshacer cambios*. Para cerrar el cuadro de diálogo, pulse el botón cerrar de la barra de títulos o pulse el botón cerrar situado al lado del botón *Deshacer cambios*.