

**Universidad de las Ciencias Informáticas**

**Facultad 10**



**Título: “Sistema de Traducción Automática para el SIGB Koha”**

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas.

**Autor(es):** Yadier Alarcón Catell-Florit

Dayron Reyes García

**Tutor:** Ing. Edisnel Carrazana Castro

**Ciudad de La Habana, Cuba, 2009.**

**“Año del 50 Aniversario del Triunfo de la Revolución”**

*Hay dos cosas infinitas: el Universo y la estupidez humana. Y del Universo no estoy seguro."*

*Albert Einstein.*

## Declaración de autoría

---

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 10 de la Universidad de las Ciencias Informáticas; así como a dicho centro a consultar cualquier cambio que se decida aplicar al mismo con su autor.

Para que así conste firmamos la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año 2009.

-----  
Dayron Reyes García

-----  
Yadier Alarcón Castell-Florit

-----  
Ing. Edisnel Carrazana Castro

## Agradecimientos

---

Agradecemos a todas aquellas personas que hicieron posible la realización de este trabajo. A Yusdanis Feus por el soporte y capacitación en el SIGB Koha, a Saimel por su ayuda en Perl, a Ángel, Yenner por su ayuda en la metodología SXP, a Serpilla por su ayuda en inglés.

## Dedicatoria

---

Dayron:

Quisiera dedicar este trabajo a toda mi familia, especialmente a mi abuela Mercedes por todo el apoyo y cariño que me ha dado.

A mi mamá Egina por siempre confiar en mí.

A mi papá Reyes, por ser mi guía.

A mi hermano Michel.

Yadier:

Quisiera dedicar este trabajo a toda mi familia, a los padres de Liyelis, y en especial a mi mamá Lázara y a mi hermano Yenier por todo el apoyo y el cariño que me han dado.

A Liyelis por ser mi compañera de los momentos más difíciles.

A mi tía Carito por ser al igual que mi mamá un ejemplo a seguir por todos estos años.

A la familia de Liyelis de Aldabó por haber pasado con ellos los momentos más felices de mis 5 años en La Habana.

A mi único amigo Lexis por haberme aguantado todos los años juntos que pasamos en la UCI.

# Índice general

---

Índice de figuras.....	IX
Índice de tablas.....	X
Resumen.....	XI
Introducción.....	1
1. CAPÍTULO 1: Fundamentación teórica.....	4
1.1. Introducción.....	4
1.2. Traducción automática.....	5
1.2.1. Reseña histórica.....	5
1.2.2. Tipos de traducción automática.....	6
1.2.2.1. Traducción automática basada en reglas.....	6
1.2.2.1.1. Traducción automática por transferencia.....	7
1.2.2.1.2. Traducción automática por lenguaje intermedio.....	10
1.2.2.2. Traducción automática basada en corpus paralelos.....	14
1.2.2.2.1. Traducción automática basada en ejemplos.....	15
1.2.2.2.2. Traducción automática basada en estadística.....	16
1.2.2.3. Traducción automática basada en el contexto.....	18
1.3. Sistemas de traducción automática de uso actual.....	19
1.3.1. Traductores de escritorio:.....	19
1.3.1.1. Traductores de licencia privada:.....	19
1.3.1.2. Traductores de licencia libre.....	20
1.3.2. Traductores web:.....	21
1.4. Tendencias de los sistemas de traducción automática.....	21
1.5. Herramientas, lenguajes y tecnología a usar.....	22
1.5.1. Anjuta.....	22
1.5.2. SWI-Prolog.....	22
1.5.3. Quanta Plus.....	23
1.5.4. Prolog.....	23
1.5.5. HTML.....	23
1.5.6. Perl.....	24
1.5.7. CGI.....	24
1.5.8. Mysql.....	25
1.5.9. Apache.....	25
1.6. Metodologías ágiles a usar.....	26
1.7. Conclusiones.....	30
2. CAPÍTULO 2: Descripción y análisis de la solución propuesta.....	31
2.1. Introducción.....	31
2.2. Valoración crítica del diseño.....	31

2.3. Lista de reserva del producto.....	33
2.4. Historias de usuarios arquitectónicamente más significativas.....	35
2.5. Arquitectura base del sistema .....	38
2.5.1. Arquitectura cliente-servidor.....	38
2.5.2. Patrón arquitectónico Modelo-Vista-Controlador .....	39
2.6. Base de datos.....	40
2.7. Diagrama de componentes.....	41
2.8. Conclusiones.....	43
3. CAPÍTULO 3: Desarrollo ágil de la aplicación propuesta.....	44
3.1. Introducción.....	44
3.2. Planificación por roles.....	44
3.3. Modelo de dominio.....	45
3.4. Historias de usuario y prototipos de interfaces de usuario.....	46
3.5. Plan de release.....	55
3.6. Conclusiones.....	56
4. CAPÍTULO 4: Validación de la solución propuesta.....	57
4.1. Introducción.....	57
4.2. Casos de prueba de aceptación.....	57
4.3. Conclusiones.....	62
Conclusiones generales.....	63
Recomendaciones.....	64
Referencias bibliográficas.....	65
Bibliografía.....	66
Anexos.....	68
Glosario de términos.....	75

## Índice de figuras

---

Figura 1: Funcionamiento de un sistema basado en reglas.....	7
Figura 2: Grafos de traducción.....	12
Figura 3: Traducción interlingua utilizando métodos sintácticos.....	12
Figura 4: Traducción interlingua utilizando métodos de Inteligencia Artificial.....	13
Figura 5: Expediente de proyecto.....	30
Figura 6: Base de datos.....	40
Figura 7: Diagrama de componentes.....	42
Figura 8: Modelo de dominio.....	46
Figura 9: Interfaz de usuario del traductor automático de escritorio “Babylon”.....	68
Figura 10: Interfaz de usuario del traductor automático de escritorio “L&H Power Translator”.....	69
Figura 11: Interfaz de usuario del traductor automático de escritorio “Star-Dict”.....	70
Figura 12: Interfaz de usuario del traductor web “Google Translate”.....	71
Figura 13: Interfaz de usuario del traductor web “Yahoo Babel Fish”.....	72
Figura 14: Prototipo de la Historia de Usuario: “Mostrar en el Koha el resultado de la búsqueda”.....	73
Figura 15: Prototipo de la Historia de Usuario: “Insertar palabras no clasificadas morfológicamente”.....	74

## Índice de tablas

---

Tabla 1: Ejemplos de corpus paralelos.....	16
Tabla 2: Lista de reserva del producto.....	35
Tabla 3: Historia de Usuario BNJM_T_01.....	36
Tabla 4: Historia de Usuario BNJM_T_02.....	37
Tabla 5: Historia de Usuario BNJM_T_03.....	38
Tabla 6: Planificación por roles.....	45
Tabla 7: Historia de Usuario BNJM_T_01.....	47
Tabla 8: Tarea de Ingeniería BNJM_T_01 -1.....	47
Tabla 9: Tarea de Ingeniería BNJM_T_01 -2.....	48
Tabla 10: Tarea de Ingeniería BNJM_T_01-3.....	48
Tabla 11: Historia de Usuario BNJM_T_02.....	49
Tabla 12: Tarea de Ingeniería BNJM_T_02-1.....	49
Tabla 13: Tarea de Ingeniería BNJM_T_02-2.....	50
Tabla 14: Historia de Usuario BNJM_T_03.....	50
Tabla 15: Tarea de Ingeniería BNJM_T_03-1.....	51
Tabla 16: Tarea de Ingeniería BNJM_T_03-2.....	51
Tabla 17: Tarea de Ingeniería BNJM_T_03-3.....	52
Tabla 18: Tarea de Ingeniería BNJM_T_03-4.....	52
Tabla 19: Historia de Usuario BNJM_T_04.....	53
Tabla 20: Tarea de Ingeniería BNJM_T_04-1.....	53
Tabla 21: Tarea de Ingeniería BNJM_T_04-2.....	54
Tabla 22: Tarea de Ingeniería BNJM_T_04-3.....	54
Tabla 23: Historia de Usuario BNJM_T_05.....	55
Tabla 24: Historia de Usuario BNJM_T_06.....	55
Tabla 25: Plan de release.....	56
Tabla 26: Caso de Prueba de Aceptación BNJM-1-1.....	58
Tabla 27: Caso de Prueba de Aceptación BNJM-1-2.....	59
Tabla 28: Caso de Prueba de Aceptación BNJM-1-3.....	59
Tabla 29: Caso de Prueba de Aceptación BNJM-1-4.....	60
Tabla 30: Caso de Prueba de Aceptación BNJM-1-5.....	61
Tabla 31: Caso de Prueba de Aceptación BNJM-1-6.....	62

## Resumen

---

La informática ha constituido una revolución tecnológica que ha cambiado totalmente el modo de vida de los seres humanos, de tal modo que existe gran dependencia hacia ella. No existe actualmente rama alguna de la vida social, donde la misma no se aplique y que no haya tenido un impacto positivo. La bibliotecología no es una excepción, ya que la informática ha contribuido bastante a su desarrollo, debido principalmente a los SIGB (Sistemas Integrados de Gestión Bibliotecaria), los cuales permiten la automatización de muchos de los procesos que se realizan en una biblioteca. Uno de los SIGB más utilizados actualmente es Koha por su gran versatilidad.

Esta investigación brinda una solución integral, que perfecciona el sistema de localización de libros del SIGB Koha, proporcionándole al usuario una mayor comodidad en su búsqueda. Gracias a la posibilidad que posee esta aplicación de traducir al inglés el criterio de búsqueda introducido por el usuario y mostrar todos los libros que se relacionan con dicho criterio de búsqueda ya traducido, la misma proporciona la opción de acceder a libros escritos especialmente para anglo-parlantes y que no han sido traducidos a otros idiomas; por lo cual, permite elevar la cultura general-integral de nuestros ciudadanos.

# Introducción

---

La traducción literal de un idioma a otro, ya sea en forma de documentos escritos o de manera oral, siempre ha sido un desafío importante para la humanidad a lo largo de su historia. Desde épocas primitivas, el Homo Sapiens encontró la necesidad de comunicarse con similares de su misma especie, ya sea para organizar la caza, la pesca, o realizar otras importantes labores necesarias para su supervivencia. De esta forma, surge el lenguaje articulado como una necesidad comunicativa.

En sus inicios, la humanidad se encontraba segregada en pequeñas tribus, por lo cual la manera de comunicarse de cada una de ellas, era diferente a las otras. Estas diferencias se fueron acentuando con el decursar del tiempo, dando lugar a que hoy día, existan una multitud de idiomas y dialectos en todo el planeta, los cuales impiden una comunicación efectiva. Gracias al surgimiento de la era informática, todas las barreras idiomáticas se han minimizado enormemente, debido principalmente a la construcción de valiosos softwares que permiten la traducción de textos, documentos web y de voz.

En las bibliotecas se almacena gran parte de la información que poseen los seres humanos. Las mismas poseen una gran variedad de materiales bibliográficos sobre temáticas muy variadas. Pero en la sociedad actual, no sólo es importante la abundancia de información, sino también el fácil acceso a la misma. Con tal propósito, los SIGB automatizan la mayor parte de los procesos que se realizan en una biblioteca, entre estos la búsqueda de los materiales bibliográficos a través de catálogos en línea. Sin embargo, cuando buscamos una información determinada, es común encontrar materiales escritos en diversos idiomas, que tratan la misma temática. Los SIGB actuales, obligan al usuario a hacer una búsqueda para cada idioma sobre el mismo criterio de búsqueda, siendo esto una limitación. Lo anteriormente planteado pudiera resultar incómodo para muchos usuarios.

El grupo Gestión Bibliotecaria de la Universidad de las Ciencias Informáticas tiene como misión fundamental la automatización de los procesos que se realizan en las bibliotecas, para ello usa el SIGB Koha, el cual aunque satisface las necesidades de las mismas, no está libre de la limitación anteriormente expuesta.

## **Situación problémica:**

El sistema de búsqueda del SIGB Koha no permite localizar títulos de libros que se encuentran en un idioma diferente al del criterio de búsqueda introducido por el usuario, aunque correspondan al mismo

material bibliográfico.

### **Problema Científico:**

¿Cómo lograr que el sistema de búsqueda del SIGB Koha pueda recuperar registros bibliográficos en español e inglés usando un mismo criterio de búsqueda?

El **objeto de estudio** de este trabajo, estará centrado en el estudio de los procesos de traducción automática. Y teniendo como **campo de acción** los procesos de traducción automática en los sistemas de búsqueda para los SIGB.

**El objetivo general** es desarrollar un módulo de traducción automática para el SIGB Koha.

### **Objetivos específicos:**

- Analizar los diferentes sistemas de traducción automática existentes en Cuba y en el resto del mundo.
- Comprender el funcionamiento de los sistemas de traducción automática.
- Diseñar un módulo de traducción para el SIGB Koha.
- Implementar un módulo de traducción para el SIGB Koha.

### **Idea a defender:**

Con la implementación de un módulo de traducción automática al SIGB Koha, se logrará que el mismo pueda recuperar registros bibliográficos en inglés y español usando un mismo criterio de búsqueda.

### **Tareas a realizar:**

- Investigar los sistemas de traducción existentes en Cuba y el mundo, sus características, ventajas y desventajas.
- Estudiar en profundidad los lenguajes de programación a usar para implementar el sistema.
- Describir el funcionamiento de forma general de los sistemas de traducción para lograr una mejor implementación de nuestro sistema.

- Desarrollar un sistema con la mayor robustez y fiabilidad para el usuario final.
- Realizar las pruebas necesarias para eliminar los posibles errores que pueda poseer el producto.

### **Métodos científicos:**

El método Científico de Investigación es la forma de estudiar los fenómenos de la naturaleza y la sociedad para descubrir sus relaciones y su esencia, el mismo se puede clasificar en métodos teóricos y métodos empíricos los cuales están relacionados entre si de forma dialéctica.[9] A continuación se muestran los métodos utilizados para el desarrollo de esta investigación.

### **Métodos Teóricos:**

Estos métodos permiten estudiar las características del problema que no son observables directamente.

- **Método Histórico:** Se ha realizado un estudio del origen y desarrollo de los sistemas de traducción.
- **Método Lógico:** Basándose en el estudio del problema, se expresa la esencia del mismo, explicando su historia.
- **Análisis y Síntesis:** Para un mejor entendimiento y estudio del problema se ha descompuesto el mismo en varios elementos, de esta forma se ha concretado cuales son los fenómenos que lo generan. Luego de haber realizado el análisis de todos los fragmentos del problema se hace una unión del mismo, posibilitando definir sus características generales y las relaciones que existen entre las partes analizadas.

### **Métodos Empíricos:**

- **Entrevista:** Es un método que permite conocer el fenómeno y tiene como objetivo obtener información. Se utiliza para obtener toda la información necesaria del problema y poder así realizar el levantamiento de requisitos. Mediante este se realizaron las entrevistas necesarias a los futuros usuarios del módulo, lo cual permitió tener un mejor entendimiento del fenómeno.

# 1. CAPÍTULO 1: Fundamentación teórica

## 1.1. Introducción

La gran variedad de dialectos e idiomas que han existido desde el comienzo mismo de la humanidad, ha sido un problema significativo que ha obstaculizado las relaciones entre las personas. Desde un principio, el hombre se vio en la necesidad de interactuar con otras civilizaciones con diferentes idiomas y cultura que la suya, motivado principalmente por razones comerciales y de conquista.

Antes de la invención de la computadora, se puede decir que la traducción estaba en pañales, debido a que esa tarea les era reservada a pocas personas llamadas traductores, los cuales eran los únicos que tenían los conocimientos necesarios para poder traducir de un idioma a otro, por lo cual, este proceso dependía exclusivamente de estas personas.

Con la invención de la computadora, se hace un significativo avance en el campo de la traducción, debido a la gran capacidad de cálculo que la misma posee, la cual supera en ese sentido al cerebro humano, y además es capaz de traducir por sí sola textos enteros sin la necesidad de la intervención humana. Esto permitió que el proceso de traducción se expandiera por el mundo, surgiendo así la traducción automática, la cual es una rama de la Traducción, y a la vez un campo importante de investigación en el área de la Informática y las Telecomunicaciones.

La traducción automática ha jugado un papel fundamental desde su invención hasta nuestros días, y ha servido para estrechar los lazos de hermandad entre todas las naciones. Actualmente, la misma es utilizada en todos los sectores, sobre todo en la red de redes, donde gracias a la misma, es posible, por ejemplo, que una persona pueda acceder y comprender un sitio web que posea un idioma diferente al suyo.

Actualmente, no se puede afirmar que la traducción automática ha alcanzado la perfección, debido a pequeñas incongruencias que aún existen, sobre todo a la hora de traducir las llamadas “frases idiomáticas”. Sin embargo, se podría decir que la misma se encuentra en su plena madurez.

### 1.2. Traducción automática

La traducción automática es un área de la lingüística computacional que investiga el uso de software para traducir texto o hablar de un lenguaje natural a otro. En un nivel básico, la traducción por computadora realiza una sustitución simple de las palabras atómicas de un lenguaje natural por las de otro. Por medio del uso de corpora lingüísticos se pueden intentar traducciones más complejas, lo que permite un manejo más apropiado de las diferencias en la tipología lingüística, el reconocimiento de frases, la traducción de expresiones idiomáticas y el aislamiento de anomalías.

#### 1.2.1. Reseña histórica

A través de la historia, el hombre se ha interesado en construir máquinas que le permitieran superar las barreras lingüísticas. En el siglo XVII, se habla de la utilización de diccionarios mecánicos, basados en códigos numéricos universales, para lograr traducciones. En ese tiempo, existía un movimiento a favor de la creación de una lengua universal basada en principios lógicos y símbolos icónicos. A lo largo de los siglos XVIII y XIX, aparecieron muchas otras propuestas de lenguas internacionales, la más conocida de las cuales es el esperanto. Con la invención de la computadora en 1940, la traducción entra en una nueva fase al lograr automatizar este proceso, surgiendo así la traducción automática. De cualquier modo, la idea de una lengua universal aplicada a la traducción automática no apareció desde sus inicios, ya que se empezó a trabajar con pares de lenguas. Fue durante la década de 1956 a 1966 cuando empezaron a estudiarse algunas ideas en Cambridge, dirigidas por Margaret Masterman, en Milán por Silvio Ceccato y en Leningrado por Nikolai Andrev. Las primeras ideas de traducción automática estadística fueron introducidas por Warren Weaver, el cual tenía el propósito de incluir en ella la teoría de la información de Claude Shannon. Durante la década de 1970 cabe destacar los estudios realizados en Grenoble, que buscaban traducir trabajos matemáticos y físicos del ruso al francés, aunque no era puramente mediante una lengua intermedia. Un trabajo similar era el que se desarrollaba en Texas Metal.

En la década de 1980, toman vigencia los sistemas basados en interlingua y los sistemas basados en el conocimiento, con múltiples investigaciones sobre el tema. El factor común a todas estas investigaciones, es que parten de la idea de que es necesaria la comprensión del texto si se quiere conseguir una traducción de calidad. Para ello, la traducción debe estar basada a la vez, en el conocimiento lingüístico y el dominio concreto de la aplicación, intentando la modularidad entre ambos, aunque marcando el énfasis

sobre éste último. Las investigaciones de mayor importancia a partir de esta década fueron el DLT en Utrecht y el sistema Fujitsu en Japón. El DLT trabajaba con una modificación del esperanto. Posteriormente las investigaciones, se enfocan principalmente en la traducción estadística, desarrollando en 1991 sistemas de traducción como CANDICE. Más recientemente, la empresa Meaningful Machine desarrolla el método de traducción automática basada en el contexto.

### **1.2.2. Tipos de traducción automática**

De acuerdo a la forma en la cual se realiza la traducción, los sistemas de traducción automática se pueden clasificar en varios grupos. Actualmente, los más conocidos son: la traducción automática basada en reglas, de la cual se deriva la traducción por transferencia y la traducción por lenguaje intermedio; la traducción automática basada en corpus paralelos de donde heredan la traducción estadística y la traducción basada en ejemplos y finalmente la traducción automática basada en el contexto. A continuación se expone un resumen, donde se habla de los aspectos más relevantes de cada uno de ellos:

#### **1.2.2.1. Traducción automática basada en reglas**

La traducción automática mediante reglas consiste en realizar transformaciones a partir del original, basándose en un conjunto de reglas y reemplazando las palabras por su equivalente más apropiado. En general, en una primera fase se analizará un texto, normalmente creando una representación simbólica interna. Dependiendo de la abstracción de esta representación, también podemos encontrar diferentes grados: desde los directos, que básicamente hacen traducciones palabra por palabra, hasta interlingua, que utiliza una representación intermedia completa.

A continuación mostramos una imagen donde se pone de manifiesto el funcionamiento de la traducción automática basada en reglas:



Figura 1: Funcionamiento de un sistema basado en reglas.

La traducción automática basado en reglas se divide en dos subcategorías: la traducción automática por transferencia y la traducción automática por lenguaje Intermedio.

### 1.2.2.1.1. Traducción automática por transferencia

La traducción automática por transferencia pertenece a la traducción automática basada en reglas, su base fundamental es la interlingua y es uno de los métodos de traducción más usados actualmente. La interlingua es un lenguaje creado en 1951 por expertos en lingüística, como una lengua auxiliar que pudiera ayudar en las comunicaciones internacionales. La misma, es un lenguaje intermedio entre todas las lenguas romances como el español, francés, italiano, portugués, además que posee algo de inglés y ruso. Debido a su sencillez desde el punto de vista gramatical, es un idioma que puede ser comprendido inmediatamente sin necesidad de estudiarlo, principalmente por los hablantes de lenguas neolatinas así como los anglófonos. Esto hace que dicha lengua pueda ser comprendida por casi 1000 millones de personas en el mundo.

Tanto la interlingua, como la traducción automática por transferencia parten del mismo concepto, es decir, para lograr la traducción final se debe crear una representación simbólica que capte la idea original de la oración para traducirlo al lenguaje equivalente. En interlingua, esta representación es independiente del idioma, pues la misma es la base de muchas lenguas, pero sin embargo, en la traducción automática por transferencia está más relacionada a los dos pares de idiomas que se desean traducir.

El funcionamiento de un traductor automático basado en transferencia es variable, pero normalmente sigue un mismo patrón: se aplican conjuntos de reglas lingüísticas, que son las que definen como se corresponde la estructura de una lengua respecto de otra. En un primer paso se analiza la entrada,

morfológica y sintácticamente, creando la representación interna. A partir de esta información, y mediante el uso de reglas gramaticales y vocabulario, se genera la traducción. Gracias a esto, en los sistemas de traducción por transferencia, es posible conseguir traducciones, con una fiabilidad de más del 90%. Gracias a lo cual, los mismos han sido ampliamente usados en los últimos treinta años. Aunque últimamente la mayor parte de las investigaciones en el campo de la traducción automática va dirigido a métodos más eficaces, como traducción automática mediante corpus lingüísticos (ya sea basado en ejemplos o en estadísticas) o traducción automática basada en el contexto, la traducción automática por transferencia sigue jugando un papel fundamental en esta área.

### **Funcionamiento:**

En los sistemas de traducción automática que se basan en la traducción por transferencia, al texto origen se le analiza primero su morfología y posteriormente su sintaxis, obteniendo como resultado una representación sintáctica superficial. Esta representación, se convierte por consiguiente en otra más abstracta que hace especial énfasis en aspectos relevantes para el proceso de traducción e ignora otro tipo de información. El proceso de transferencia convierte esta última representación a una representación al mismo nivel de abstracción pero ligada al lenguaje objetivo. Estas dos representaciones son las llamadas normalizadas o intermedias. A partir de aquí el proceso se invierte: los componentes sintácticos generan una representación del texto y finalmente se genera la traducción.

### **Análisis y transformación:**

Se pueden utilizar distintos artilugios antes de llegar a la traducción final. Dependiendo del diseño del traductor se realiza hincapié en uno u otro, e incluso se pueden llegar a combinar con análisis estadísticos, formando auténticos híbridos. Pero, en los traductores por transferencia normalmente utilizados, se pueden encontrar los siguientes pasos:

- **Análisis morfológico:** Consiste en clasificar los elementos que componen la cadena de texto introducida, desde un punto de vista morfológico en: artículos, verbos, adjetivos, etc. Además, también deben reconocerse abreviaturas y otras expresiones o palabras compuestas.
- **Categorización léxica:** Algunas de las palabras que aparecen en un texto pueden tener más de un significado, causando así ambigüedad a la hora de hacer su análisis. La categorización léxica analiza el contexto, es decir, las palabras que rodean a la que se debe clasificar, y selecciona la correcta clasificación.

- Transferencia léxica: Consiste en traducir palabra por palabra al idioma destino, todos los elementos del lenguaje origen.
- Transferencia estructural: Una transferencia estructural analiza el texto desde un punto de vista más amplio. En vez de centrarse en palabras, amplía su objetivo a fragmentos mayores. De esta manera se pueden encontrar expresiones que puedan requerir un tratamiento especial.
- Generador morfológico: En el momento en que ya se ha realizado el análisis, el generador morfológico es el encargado de asociar cada elemento identificado en fases anteriores con su equivalente en la lengua objetivo, procurando que la interpretación sea lo más fiel posible.[1]

### Ejemplos de transferencia:

A continuación se muestra un ejemplo donde se observa el funcionamiento del método de transferencia superficial para la traducción de un texto del español al inglés. El texto en español es: “**la mujer bella**”, que traducido al inglés sería “**the beautiful woman**”.

Para realizar esta traducción vamos a seguir los pasos que explicamos anteriormente:

1- Análisis morfológico: En este paso se realiza un análisis morfológico de la frase a traducir, o sea, las palabras se clasifican en sustantivos, adjetivos, verbos, pronombres, etc.

#### **la**

- la, artículo, definido, femenino.
- la, pronombre personal, femenino.

#### **mujer**

- mujer, sustantivo, Femenino.

#### **bella**

- bella, adjetivo, femenino.

2- Categorización léxica: Este paso es usado para clasificar correctamente las posibles ambigüedades que pudiesen existir. En este caso, la palabra “**la**” tiene más de una clasificación, por lo que de acuerdo al contexto y a las palabras que se encuentran cercanas a ella, se llega finalmente a la conclusión de que la clasificación mostrada a continuación es la única que tiene sentido en esta frase.

**la:** la, artículo, definido. femenino.

**mujer:** mujer, sustantivo. Femenino.

**bella:** bella, adjetivo, femenino.

3-Transferencia léxica y estructural: Aquí se traduce literalmente palabra por palabra nuestra frase al lenguaje destino, se reordenan las palabras y se busca la concordancia entre género y número de los sustantivos y sus respectivos artículos.

**la** → the (artículo, definido, femenino).

**mujer** → woman (sustantivo, Femenino).

**bella** → beautiful (adjetivo, femenino).

4-Generador morfológico: Llegados a este punto, la traducción por transferencia aplicaría el generador morfológico, obteniendo como resultado la frase traducida: «**the beautiful woman**».

### 1.2.2.1.2. Traducción automática por lenguaje intermedio

La traducción automática mediante lengua intermedia es un tipo de traducción automática basada en reglas. La idea inicial es representar el texto origen en una lengua abstracta, la cual sería independiente a la lengua inicial y final, para posteriormente traducirlo a la lengua destino. Este tipo de traducción realiza el análisis de los textos escritos en la lengua de origen, imponiendo en ellos las características morfológicas, sintácticas y semánticas correspondientes a la lengua objetivo, realizando una primera transferencia del texto analizado a una lengua intermedia. Dicha lengua posee todas las características morfológicas, sintácticas y semánticas de los idiomas a los que serán traducidos, en lugar de traducir directamente de la lengua fuente a la lengua objetivo. En la siguiente figura se muestra la diferencia que existiría entre realizar una traducción directa y realizar una traducción utilizando una lengua puente. En la figura de la izquierda se muestra los grafos de traducciones necesarios en el caso de traducir directamente: el resultado es que se necesitan doce diccionarios de traducción. En la figura a la derecha se muestra el grafo que representa una traducción utilizando un lenguaje intermedio, para lograrlo hacen falta solamente ocho módulos de traducción.

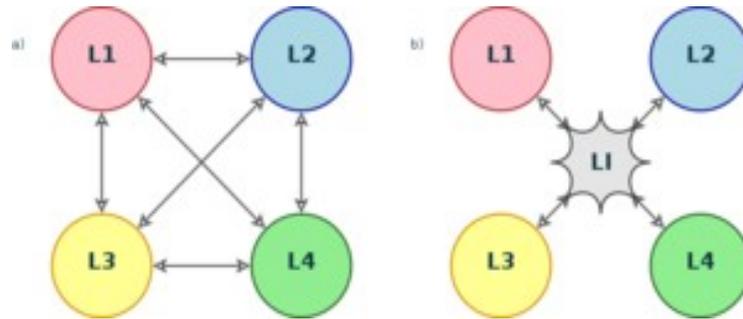


Figura 2 – Grafos de traducción.

En ocasiones se puede utilizar dos lenguas intermedias, ya que esta última tiene posibilidades de adecuarse más a las características de la lengua fuente, esto contribuye notablemente a la fiabilidad en la traducción. La segunda lengua intermedia poseería todas las características estructurales de la lengua objetivo, convirtiendo las traducciones de la primera lengua en un lenguaje mucho más próximo a la lengua final desde el punto de vista gramatical, sintáctico y morfológico. El segundo lenguaje intermedio utiliza un vocabulario más específico y cercano a la lengua de destino y, en orden para concluir la traducción, extrae los análisis estructurales de las oraciones en lengua objetivo.

Este sistema permite optimizar la traducción de un texto original a idiomas estructuralmente cercanos con realizar un sólo análisis. Las características del acercamiento interlingüístico garantizan una integración más fácil en los sistemas de idiomas que no se relacionan tipológicamente, entre los cuales no sería posible garantizar una proyección de la información lingüística sin un análisis exhaustivo de la información contextual y/o semántica del mensaje. Durante su desarrollo, la invención consideraba sólo los idiomas más hablados, pero puso especial atención y cuidado en el desarrollo de sistemas válidos que permitieran extensiones a todas las lenguas.

A continuación se muestra una figura de un grafo que representa la traducción utilizando dos lenguas intermedias:

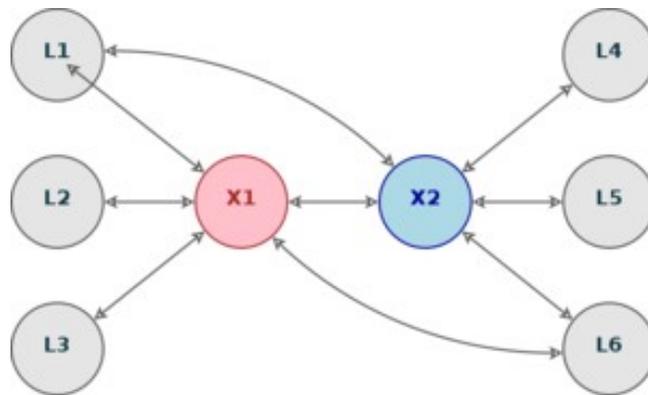


Figura 2: Grafos de traducción.

**Proceso de traducción:**

Básicamente en este tipo de traducción se identifican dos componentes monolingües, el primero es la traducción de la lengua origen a la lengua intermedia y el segundo la traducción de la lengua intermedia a la lengua objetivo. También, se pueden distinguir entre sistemas interlingua, que utilizan métodos sintácticos y los que se basan en técnicas de Inteligencia artificial.

A continuación se muestran dos imágenes que describen a estos dos métodos.

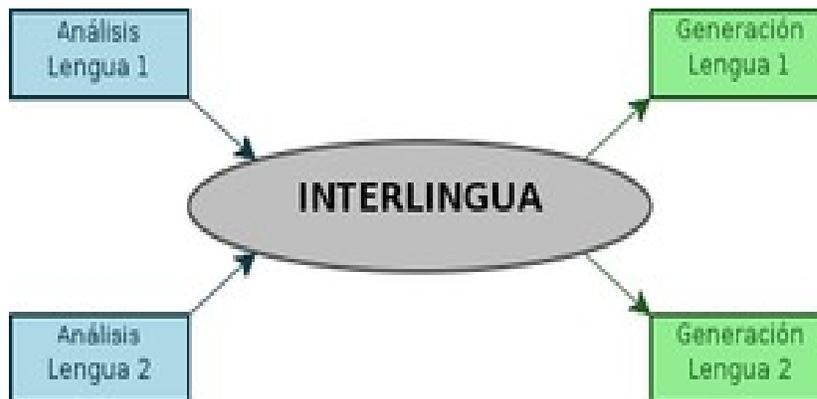


Figura 3: Traducción interlingua utilizando métodos sintácticos.

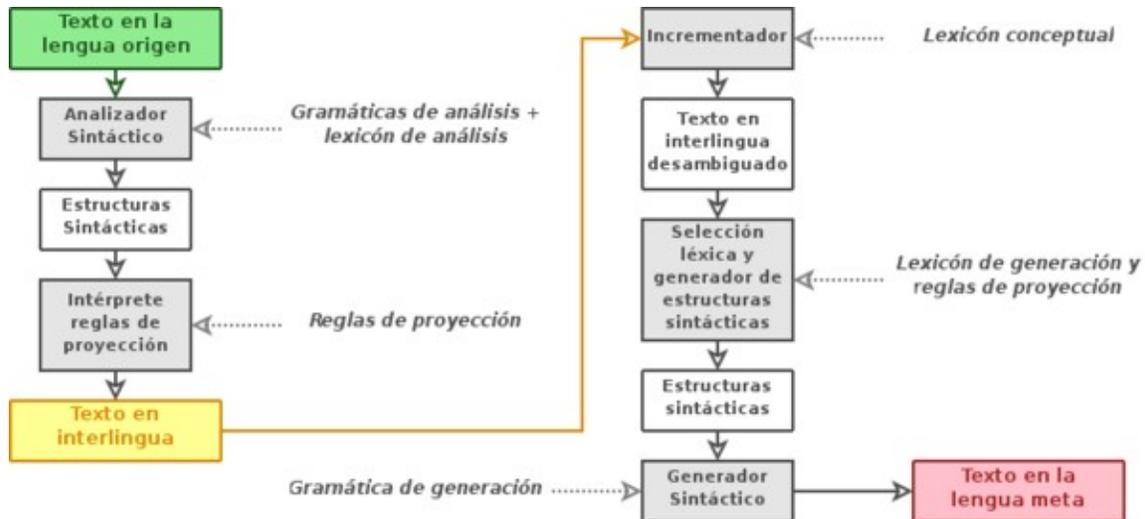


Figura 4: Traducción interlingua utilizando métodos de Inteligencia Artificial.

Los recursos necesarios son, por lo tanto, los siguientes:

- Lexicones de análisis y generación (dependientes del dominio y las lenguas).
- Lexicón conceptual (dependiente del dominio). Es la base de datos de conocimiento sobre eventos y entidades comprendidos por el dominio.
- Reglas de proyección (dependientes del dominio y las lenguas).
- Gramáticas de análisis y generación (dependientes del dominio y las lenguas).

**Ventajas de la traducción automática con lenguaje intermedio:**

Una de las ventajas es que ofrece robustez y ahorro de esfuerzos por parte del equipo de desarrollo, debido a que se necesitan implementar menos módulos de traducción si se utilizara la interlingua, que si se utilizara la traducción directa, por el simple hecho de que en la traducción directa se necesitan dos módulos por cada pareja de lenguajes a traducir, sin embargo, en la traducción por lenguaje intermedio cada lenguaje se relaciona sólo con la lengua intermedia.

Otra de las ventajas es que sería más sencillo incorporar una nueva lengua a nuestro sistema interlingüe, pues únicamente tendríamos que relacionar la nueva lengua con la intermedia, en lugar de relacionarla con cada una de las lenguas existentes en nuestro sistema.

**Desventajas de la traducción automática con lenguaje intermedio:**

Al ser independiente de las lenguas origen y destino, cuando se le incorporen al mismo más lenguas para traducir, el lenguaje intermedio deberá ser más rico y abarcador para realizar una traducción de calidad. Debido a que en este método se hace una representación simbólica en un lenguaje intermedio, es propenso a perder información ya que se muestra lo más relevante de la misma, o sea, la idea esencial. Otra de las desventajas es que es muy abstracto.

### 1.2.2.2. Traducción automática basada en corpus paralelos

La traducción automática basada en corpus paralelos, se basa principalmente en muestras reales con sus respectivas traducciones. Para que se tenga una idea, un corpus paralelo no es más que mismo un texto, documento o libro, traducido a dos idiomas diferentes y que se encuentra en un formato electrónico adecuado. Una condición indispensable es que el mismo se tiene que encontrar alineado a cierto de nivel de granularidad, generalmente a nivel de párrafo, aunque también a nivel de sección, página, o incluso, a veces de palabra. Con alineación, se refiere a que dada una frase o palabra en el texto origen, sea capaz de determinar cuál es su traducción correspondiente en el texto destino. O sea, por ejemplo digamos que se tiene un corpus paralelo que está representado por un mismo texto traducido al español e inglés :

Español: Yo fui a mi **casa** para ver el televisor.

Inglés: I went **home** to watch TV.

Dado el anterior corpus paralelo, si se quisiera conocer cuál es el significado de la palabra **casa** en inglés, el corpus paralelo debe ser capaz de alinear dicha palabra en los dos idiomas y responder correctamente: **house**.

La traducción automática basada en corpus paralelos, posee ventajas por encima del método de traducción basados en reglas, ya que permite eliminar ciertas ambigüedades que pudieran surgir a la hora de traducir, principalmente con las llamadas "frases idiomáticas". Debido a que las mismas se basan en textos, revistas o simplemente cualquier documento que se pueda localizar en Internet y que esté disponible en varios idiomas, poseen vigencia actual. Los sistemas de traducción automática basados en corpus paralelos se dividen en dos tipos principales: traducción automática estadística y traducción automática basada en ejemplos. Cada uno posee características específicas que lo diferencian.

### 1.2.2.2.1. Traducción automática basada en ejemplos

La idea básica de la traducción automática basada en el ejemplo (EBMT: Example Based Machine Translation) se debe principalmente al investigador japonés Makoto Nagao. La EBMT utiliza información estadística pero con métodos diferentes. O sea, el mismo no utiliza algoritmos estadísticos a la hora de traducir, sino para encontrar ejemplos en determinados contextos. Se trata de anotar y clasificar, mediante la alineación oracional de textos paralelos, las formas en que un fragmento de texto de la lengua de origen ha sido traducido a una lengua meta y usar esa información para elegir traducciones de esos fragmentos en otros textos. Los fragmentos no son constituyentes sintácticos, sino secuencias de cadenas de caracteres, es decir, no se usa conocimiento lingüístico alguno.

Se podría resumir la estrategia básica del modo siguiente: dado un fragmento  $O$  de texto de la lengua de origen y una colección de textos bilingües donde se almacenan fragmentos  $O'$  de la lengua de origen, alineados con sus traducciones en una lengua meta (fragmentos  $M'$ ),  $O$  se compara con la parte correspondiente a la lengua origen de la colección de textos. Se selecciona el emparejamiento más cercano para el fragmento  $O'$ , y la traducción de ese emparejamiento,  $M'$ , se acepta como la traducción de  $O$ .

Los pasos que normalmente se siguen en un proceso de traducción por analogía son los siguientes:

1. Alinear el corpus bilingüe a nivel oracional.
2. Encontrar fragmentos de la parte del corpus que corresponde a la lengua de origen que se consideren como los mejores candidatos (ejemplos ideales) para un fragmento *input* determinado (*intra-language matching*).
3. Encontrar el fragmento de la lengua meta que corresponde al fragmento de la lengua de origen (*inter-language matching*).
4. Combinar los resultados a nivel de fragmentos para obtener la "cubierta" para todo el texto. [1]

Inglés	Español
How much is that <b>red umbrella</b> ?	Cuánto cuesta esa <b>sombrilla roja</b>

How much is that <b>small camera</b> ?	Cuánto cuesta esa <b>pequeña cámara</b> .
--	---

Tabla 1: Ejemplos de corpus paralelos.

Con estos tipos de oraciones es muy fácil aprender traducciones de unidades suboracionales. Por ejemplo, un sistema aprende dos unidades de traducción:

1. **green bear** coincide con **oso verde**
2. **chinese bicycle** coincide con **bicicleta china**

Mediante estas unidades, se pueden construir traducciones nuevas. Por ejemplo, si usamos un texto para entrenar el sistema que tiene las oraciones:

**I was in Varadero beach last Sunday.** y **This woman is very beautiful.**

Sería posible traducir la oración **Varadero beach is very beautiful** por sustitución de unidades suboracionales.

### 1.2.2.2. Traducción automática basada en estadística

La traducción automática por estadística es un paradigma en este campo donde las traducciones se generan a partir de modelos estadísticos y de la teoría de la información, los cuales se obtienen al analizar el corpus paralelo. Este método se diferencia de otros métodos tradicionales como la traducción automática basada en reglas, y la basada en ejemplos.

Este método posee ventajas sobre otras metodologías, en primer lugar porque hace un mejor uso de los recursos, ya que por ejemplo, los sistemas de traducción basados en reglas requieren grandes esfuerzos en la implementación, lo que pudiera resultar costoso. Además, generalmente este método no se encuentra adaptado a un par específico de idiomas. Pero la característica principal, es que es capaz de realizar traducciones más naturales que los métodos descritos anteriormente.

La idea básica para traducir mediante técnicas estadísticas es la de utilizar los denominados corpora paralelos (textos traducidos por un humano) para extraer de ellos la información estadística necesaria con el objeto de "entrenar" el sistema de TA. En la actualidad se dispone de un conjunto considerable de estos textos, aunque sólo para algunos idiomas. El primer paso para trabajar con este tipo de corpora es alinear las oraciones de los textos origen y meta para después establecer índices de probabilidad de

traducción (o más bien equivalencia) entre las distintas cadenas de caracteres. En ningún momento se usa información de carácter lingüístico para el análisis ni para la generación.

El equipo responsable de este resurgimiento de los métodos empíricos aplicados a la traducción automática fue el de Peter Brown, quienes retomaron las ideas originales alentados por los buenos resultados de la aplicación de técnicas estadísticas a la síntesis de habla. Su trabajo se basó en el modelo de "canal ruidoso" de Shannon. En este modelo se plantea la situación de recuperar la información perdida de un mensaje al atravesar un canal con ruido, tal como una línea telefónica. Una secuencia de texto correcto ( $I$ ) entra en el canal, y una secuencia corrupta ( $O$ ) sale por el otro lado.

$$I \rightarrow \text{canal ruidoso} \rightarrow O$$

Para recuperar la secuencia original  $I$  se puede tratar de recuperar la secuencia más *probable*  $I'$  teniendo en cuenta todas las secuencias de entradas posibles,  $I$ , y seleccionando la que obtenga la probabilidad más alta,  $\Pr(I|O)$ . Del mismo modo, se parte del supuesto de que cada oración de una lengua es una posible traducción de otra oración en otra lengua. A cada par de oraciones ( $O, M$ ) se le asigna una probabilidad,  $\Pr(O|M)$ , que se interpreta como la probabilidad de que un traductor humano genere  $M$  en el texto meta a partir de  $O$  en el texto origen. El problema de la traducción automática queda así reducido a lo siguiente: dada una oración  $M$  en la lengua meta, se busca la oración  $O$  a partir de la cual el traductor generó  $M$ . Conceptualmente, el programa de traducción busca el dominio de todas las posibles oraciones en el texto origen que maximiza  $\Pr(O|M)$ . Mediante el conocido Teorema de Bayes, esto se puede llevar a cabo mediante la expresión:

$$\Pr(O | M) = \frac{\Pr(O) \cdot \Pr(M | O)}{\Pr(M)}$$

$\Pr(O|M)$  es la probabilidad condicional de que la oración de origen,  $O$  sea, la que buscamos, dado que de hecho observamos la oración meta  $M$ . Del mismo modo,  $\Pr(M|O)$  es la probabilidad condicional de que  $M$  pudiera darse, en caso de que  $O$  se dé. El denominador a la derecha no depende de  $O$ , por lo que basta con escoger un  $O$  que maximice el producto  $\Pr(O) * \Pr(M|O)$ . El primer factor de este producto es la

probabilidad de un modelo de lengua, mientras que el segundo es la probabilidad de traducción de M dado O. [1]

### 1.2.2.3. Traducción automática basada en el contexto

La Traducción Automática basada en el contexto es un método novedoso desarrollado por la compañía Meaningful Machines. Hace pocos años atrás, los sistemas de traducción se dividían en aquellos que utilizaban los métodos estadísticos y el texto paralelo y aquellos que funcionaban mediante las reglas. Sin embargo, esta novedosa técnica ha revolucionado el campo de la traducción automática.

Entre sus principales características destacan: que no utiliza reglas, ni texto paralelo para lograr la traducción. Debido a la metodología que usa para lograr la traducción, posee como ventaja una mayor calidad en la misma, gracias a un mayor uso de la palabra en un determinado contexto, además que posee mayor facilidad de adición de pares de idiomas.

#### **Funcionamiento :**

La traducción automática basada en el contexto utiliza técnicas basadas en hallar la mejor traducción para una palabra, fijándose en el resto de palabras que la rodean. Básicamente este método se basa en tratar el texto en unidades de entre 4 y 8 palabras, de manera que se traduce cada una de ellas por su traducción al idioma destino y se eliminan las traducciones que han generado una "frase" sin sentido. Luego se mueve la ventana una posición (palabra), retraduciendo la mayoría de ellas de nuevo y volviendo a filtrar dejando solo las frases coherentes. Se repite dicho paso para todo el texto. Y luego se pasa a concatenar los resultados de dichas ventanas de manera que se logre una única traducción del texto. El filtrado que se realiza donde se decide si es una frase con sentido utiliza un corpus del lenguaje destino, donde se cuenta el número de apariciones de la frase buscada.

Es por tanto un método basado en ideas bastante simples que ofrecen unos muy buenos resultados en comparación a otros métodos. Como ventajas aporta también la facilidad de añadir nuevas lenguas. Ya que es solo necesario:

- Un buen diccionario, que puede ser cualquier versión comercial adaptada mediante reglas gramaticales para tener los verbos conjugados y los nombres/adjetivos con sus variaciones en número y género.
- Un corpus en el lenguaje destino, que se puede sacar por ejemplo de Internet. Sin que sea necesario traducir ninguna parte, como en los métodos estadísticos.

### **Calidad de traducción:**

La calidad de traducción de los traductores basados en el contexto, es superior a traductores tan reconocidos como SYSTRAN, el cual es utilizado por Google para realizar sus traducciones. El método BLEU es el encargado de dar una puntuación del 0 al 1 de acuerdo a la calidad de traducción de un idioma al otro; normalmente la puntuación de un traductor humano oscila entre 0.7-0.85, la de la traducción basada en el contexto está entre 0.56 - 0.69, y la traducción del traductor SYSTRAN es de 0.55. Aquí se pone de manifiesto como esta novedosa tecnología ha revolucionado el campo de la traducción automática a nivel mundial.

### **1.3. Sistemas de traducción automática de uso actual**

A nivel mundial se han desarrollado disímiles softwares que permiten la traducción ya sea de textos, páginas web, etc. Aunque no se puede afirmar que han alcanzado la perfección, realmente algunos traducen con una exactitud extraordinaria, sobre todo si traducen dos pares de lenguas que sean comunes sintácticamente tales como: el italiano y el español, sin embargo, cuando se trata de traducir lenguas que son muy diferentes entre sí como: el ruso y el español, es donde podrían surgir ambigüedades. Los traductores automáticos poseen disímiles clasificaciones ya que de acuerdo al tipo de licencia se clasifican en libres o privados; y acorde a su distribución se clasifican en traductores web y traductores de escritorio. A continuación se muestran los traductores automáticos más importantes actualmente.

#### **1.3.1. Traductores de escritorio:**

##### **1.3.1.1. Traductores de licencia privada:**

###### **Systrán:**

Diseñado en 1970 por el doctor Peter Toma, es uno de los sistemas comerciales de traducción automática más veteranos. En los últimos años, la tecnología del sistema, tradicionalmente basada en la traducción por reglas, se ha renovado, con la incorporación fundamentalmente de un módulo para la gestión de memorias de traducción.

Systrán es un sistema multiplataforma (puede instalarse tanto en GNU/Linux como en Microsoft

Windows) y es utilizado de manera gratuita en las múltiples instalaciones accesibles por Internet: Google, AQL, Alta vista, o el Instituto Cervantes. La empresa ofrece módulos preparados para más de 40 idiomas. Systrán además integra funciones multilingües muy útiles para aplicaciones como el comercio electrónico, CRM, base de datos, correo electrónico, mensajería instantánea, SMS, WAP, etc.

### **Babylon:**

Babylon es un programa informático comercial de traducción y consulta de información. Tiene una tecnología patentada de OVR y de activación por un solo clic que funciona en cualquier programa de Windows, tales como Word, Outlook, Excel, Internet Explorer y Adobe Reader.

Uno de los aspectos más destacados de este software es la gran cantidad de diccionarios de los que dispone, más de mil, algunos de ellos creados por los propios usuarios mediante una herramienta llamada Babylon Builder. Además de los diccionarios básicos en 75 lenguas, cabe destacar la incorporación de la Enciclopedia Británica Concisa en español y el Diccionario Enciclopédico VOX, en formato digital y directamente accesibles desde el programa. También dispone de varios diccionarios clásicos antiguamente disponibles sólo en papel, listos para ser accedidos de forma digital. Asimismo se incluyen glosarios de términos técnicos, como términos informáticos y médicos, y diccionarios de acrónimos. Algunos de estos diccionarios, especialmente los clásicos como las enciclopedias, requieren el pago de una tarifa anual adicional por cada uno de ellos, que se agrega al coste de la licencia del programa en sí. (ver anexo #1)

### **L&H Power Translator:**

Es uno de los mejores traductores que están en el mercado actualmente, tiene para traducir en más de veinte idiomas incluyendo la traducción a japonés. Actualmente no existen versiones para Linux pero la compañía tiene planes de realizarlo en el futuro. Posee una interfaz gráfica muy intuitiva que facilita el fácil manejo del programa, además que posee atajos de teclas que permiten al usuario operar en el mismo de manera rápida. Posee un nivel de fiabilidad muy bueno, por lo que el mismo es usado en múltiples instituciones de enseñanza.(ver anexo #2)

#### **1.3.1.2. Traductores de licencia libre**

**Star-Dict:** Desarrollado por Hu Zheng es una herramienta liberada bajo el código GPL para acceder a los archivos del diccionario StarDict. El mismo corre bajo Linux y Microsoft Windows. Los diccionarios a

elección del usuario se instalan por separados. Los archivos de diccionario se pueden crear mediante la conversión de archivos DICT. StarDict es compatible con el conocido programa weDict. Está disponible en beta para el iPhone. Con el StarDict mientras está en modo de escaneo, los resultados son mostrados en un Tooltip, permitiendo una búsqueda fácil en el diccionario. Hoy StarDict, que cuenta con un total de más de 1000 diferentes archivos de diccionario. Se considera uno de los más poderosos programas de diccionario en el mundo. (ver anexo #3)

### 1.3.2. Traductores web:

**Google Translate:** Sin lugar a dudas el mejor servicio de traducción en línea. Tiene las funciones de traducción de textos o páginas web completas al idioma deseado y desde el idioma deseado. Soporta un buen juego de idiomas, e inclusive llega a pedirle amablemente a algunos de sus usuarios que añadan algunas traducciones, especialmente en el caso de términos técnicos, los cuales podrán ser incluidos en futuras actualizaciones. A diferencia de otros servicios de traducción como Yahoo Babel Fish, el cual se basan en Systrán, Google Translate tiene su propio software de traducción. Posee limitaciones, ya que aunque el resultado de la traducción de algunos idiomas es bastante buena, no siempre se ven traducciones precisas, dado que no se aplican reglas gramaticales pero sí un análisis estadístico. (ver anexo #4)

**Yahoo Babel Fish:** Usa la tecnología de Systrán, ofreciendo el servicio de traducción gratuita de textos cortos y páginas webs. También permite la inclusión de una "caja de traducción" en las páginas webs para que los visitantes las vean en múltiples idiomas como: inglés, español, alemán, francés, italiano y portugués. (ver anexo #5)

**WorldLingo:** Traductor gratuito bastante popular, cuyo servicio en línea permite traducir partes de texto o páginas webs completas, usando el mismo sitio. El traductor de WorldLingo tiene soporte para inglés, chino(simplificado), chino(Tradicional), danés, francés, alemán, griego, italiano, japonés, coreano, portugués, ruso y español. También permite abrir una ventana con un menú desplegable, donde se puede insertar caracteres especiales (como acentos) y seleccionar para que categoría semántica queremos nuestra traducción (general, legal, técnica etc.).

### **1.4. Tendencias de los sistemas de traducción automática**

Actualmente las dos ramas de investigación en el área de la traducción automática, en la cual los científicos están prestando más interés son: la traducción por estadística y la basada en el contexto. Aunque la mayoría de los traductores actuales usan el método basado en reglas, poco a poco esta tecnología ha resultado obsoleta en comparación con otras más novedosas. Muchos fabricantes de software han optado por descartarla y utilizar los métodos estadístico o basado en el contexto, debido a que han demostrado una gran fiabilidad y calidad en las traducciones. Por lo qué, la tendencia actual en el campo de la traducción automática es la utilización de estas dos tecnologías. Por una parte, la traducción automática estadística ya es una tecnología consolidada que lleva cierto tiempo de utilización, y en las cual los investigadores confían plenamente. Por otro lado, la traducción automática basada en el contexto, es una tecnología novedosa que ha revolucionado el área del la traducción, mostrando excelentes resultados en la misma, pero que no se encuentra muy difundida en el mundo. Estas dos tecnologías, son las que dominarán el mundo de la traducción automática de aquí a cinco años, aunque se prevé que en el futuro todas estas tecnologías se integren en una sola.

### **1.5. Herramientas, lenguajes y tecnología a usar**

Luego de un detallado análisis por parte del equipo de trabajo, se llega a la conclusión de que las tecnologías y herramientas que más se adecuan a nuestro proyecto y que nos permitirán obtener mejores resultados, son las que se muestran a continuación:

#### **1.5.1. Anjuta**

Actualmente Anjuta DevStudio es un entorno de desarrollo integrado (IDE en inglés) para programar en los lenguajes C, C++, Java, Python y Perl, en sistemas GNU/Linux. Su principal objetivo es trabajar con GTK y en el escritorio GNOME, además ofrece un gran número de características avanzadas de programación. Anjuta es software libre, liberado bajo la licencia GPL. Incluye un administrador de proyectos, asistentes, plantillas, depurador, depurador interactivo y un poderoso editor que verifica y resalta la sintaxis escrita.[2]

### 1.5.2. SWI-Prolog

Es uno de los mejores IDE de programación que existen actualmente para Prolog, entre sus ventajas se encuentran que tiene soporte para Unicode, una interfaz de usuario amigable y además posee licencia Open Source. Por dichas características se escogió como el IDE a utilizar para programar en Prolog.

### 1.5.3. Quanta Plus

Es una herramienta libre de desarrollo de páginas webs diseñado para el proyecto KDE que rápidamente se está convirtiendo en un editor maduro que cuenta con varias funcionalidades. Su versión actual es la 3.5 y proporciona un interfaz de múltiples documentos (MDI), poderoso e intuitivo para los desarrolladores webs. Puede incrementar exponencialmente la productividad, a través del uso de acciones personalizadas, guiones y barras de herramientas, con lo que se puede automatizar casi cualquier tarea.  
[3]

### 1.5.4. Prolog

Es un lenguaje de programación ideado a principios de los años 70 en la universidad de Aix-Marseille por los profesores Alain Colmerauer y Phillippe Roussel. Inicialmente se trataba de un lenguaje totalmente interpretado hasta que a mediados de los 70, David H.D. Warren desarrolló un compilador capaz de traducir Prolog en un conjunto de instrucciones de una máquina abstracta denominada "Warren Abstract Machine", o abreviadamente: WAM. Desde entonces Prolog es un lenguaje semi-interpretado. Prolog se enmarca en el paradigma de los lenguajes lógicos, lo que lo diferencia enormemente de otros lenguajes más populares tales como Fortrán, Pascal, C ó Java.

Se eligió Prolog por la gran potencia que posee, sobre todo si se desea diseñar un sistema de traducción automático basado en reglas. Su motor de inferencia permite sacar conclusiones muy rápidas, guiándose en la base de conocimientos y en las reglas que tiene implementada. Lo anteriormente explicado, sería muy difícil de realizar en lenguajes imperativos como C++, C# ó Java.

### 1.5.5. HTML

El HTML, Hyper Text Markup Language (Lenguaje de Marcación de Hipertexto), es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). Fue creado en 1986 por el físico nuclear

Tim Berners-Lee; el cual tomó dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como link o ancla) el cual permite conectar dos elementos entre sí y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc, sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizará en la forma como éste lo entienda. El entorno para trabajar HTML es simplemente un procesador de texto. Estos documentos pueden ser mostrados por los visores o "browsers" de páginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer. [4]

### 1.5.6. Perl

Perl es un lenguaje de programación diseñado por Larry Wall creado en 1987. Perl toma características del C, del lenguaje interpretado shell, AWK, Sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación. Estructuralmente, Perl está basado en un estilo de bloques como los del C o AWK, y fue ampliamente adoptado por su destreza en el procesado de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.

Las características que nos hicieron elegir a Perl, entre los lenguajes a utilizar en nuestro proyecto se encuentran:

- Es un lenguaje libre, o sea no se encuentra bajo el dominio de ninguna empresa privada como es el caso de ASP.NET. Por lo cual es posible estudiar, modificar o distribuir cualquier código de Perl sin tener que pagar un centavo, ni estar sujeto a demandas. Esto también es muy beneficioso, debido a que permite estudiar fragmentos de código de otros traductores automáticos realizados anteriormente en Perl y así mejorar dicho código e incluírselo al nuestro.
- Es multiplataforma, o sea, que el mismo puede funcionar lo mismo en un servidor web que tenga instalado un sistema operativo Windows como uno que tenga instalado un sistema operativo Linux.
- La aplicación a la que se le va a adaptar el sistema de traducción automática es el SIGB Koha, el cual se encuentra implementado principalmente en Perl .

### 1.5.7. CGI

Interfaz de entrada común (en inglés Common Gateway Interface, abreviado CGI) es una importante tecnología de la World Wide Web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos MIME. Las aplicaciones que se ejecutan en el servidor reciben el nombre de CGIs.

Las aplicaciones CGI fueron una de las primeras maneras prácticas de crear contenido dinámico para las páginas web. En una aplicación CGI, el servidor web pasa las solicitudes del cliente a un programa externo. Este programa puede estar hecho en cualquier lenguaje que soporte el servidor, aunque por razones de portabilidad se suelen usar lenguajes de script. La salida de dicho programa es enviada al cliente en lugar del archivo estático tradicional.

CGI ha hecho posible la implementación de funciones nuevas y variadas en las páginas webs, de tal manera que esta interfaz rápidamente se volvió un estándar, siendo implementada en todo tipo de servidores web. [5]

### 1.5.8. Mysql

Se escogió este sistema gestor de bases de datos por disímiles características, primeramente se integra muy bien al lenguaje Perl y al servidor Apache. Para realizar proyectos sencillos, donde no se requiere almacenar gran cantidad de información, ha demostrado ser muy rápido. Aunque actualmente posee licencia propietaria debido a que fue comprada por la Sun Microsystems, las versiones de Mysql que hayan sido liberadas en fechas anteriores a esa adquisición, serán consideradas con licencia libre, por lo cual la versión de Mysql que se usa en este trabajo es libre. Esto supone una gran ventaja si se tiene en cuenta los altos precios que hay que pagar para obtener un gestor de base de datos como Oracle ó SQL Server. Otra de las ventajas es que la misma es multiplataforma, por lo cual funciona lo mismo en servidores con sistema operativo Linux, Unix y Windows.

Todas estas características han hecho que Mysql sea hoy por hoy el gestor de base de datos más utilizado por la mayoría de las páginas webs en Internet.

### 1.5.9. Apache

El servidor Apache surgió como símbolo de cooperación. En 1995 el NSCA (National Center for Super Computing Applications) creó un servidor que llegó a convertirse en el número uno en el mundo, pero, sin embargo, el principal desarrollador de servidores web del NSCA abandonó el proyecto casi en ese momento, por lo cual, el proyecto del servidor comenzó a bloquearse. Entretanto, la gente que estaba utilizando el servidor Web de NCSA, empezó a intercambiar sus propios paquetes para el servidor y pronto pensaron que era necesario un foro para gestionarlos. Nació el grupo Apache.

Debido a las disímiles ventajas que posee este servidor, respecto a sus competidores, lo escogimos como nuestro servidor web. Entre esas características se encuentran:

- Posee licencia libre, por lo cual se pueden realizar múltiples cambios para mejorar su rendimiento sin tener que pagar un centavo.
- Es un servidor altamente configurable, ya que es muy sencillo ampliar las capacidades del servidor Web Apache. Cualquiera que posea una experiencia mínima en la programación con C o Perl puede escribir un módulo para realizar una función determinada. Esto significa que hay una gran cantidad de módulos Apache disponibles para su utilización.
- Es un sistema multiplataforma que funciona a la perfección en sistemas operativos como Linux y Unix, en los cuales brinda un gran rendimiento y escalabilidad. Incluso en el sistema operativo Windows funciona mucho mejor que su otro competidor, el servidor IIS, por lo cual es muy frecuente ver que muchos servidores Windows utilicen Apache en vez de IIS.
- Es un servidor muy potente, que posee una gran seguridad y robustez.

### 1.6. Metodologías ágiles a usar

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, lo que se obtiene es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de dos o tres meses. Lo que se hace con este tipo de proyectos es separar rápidamente el aplicativo en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo. Cuando los proyectos que se van a desarrollar

son de mayor envergadura, ahí si toma sentido el basarse en una metodología de desarrollo, y empezar a buscar cual sería la más apropiada para el desarrollo del software. [6]

### **¿Por qué usar una metodología ágil y no RUP ?**

Actualmente en nuestra universidad se ha puesto muy de moda la utilización de RUP como la metodología a usar en la mayor parte de los proyectos productivos, de tal forma que muchos jefes de proyecto escogen ciegamente la utilización de la misma, sin pensar realmente si se adapta o no a los requisitos de la aplicación. Es cierto que RUP es una metodología muy completa, pero a veces resulta obsoleta utilizarla en proyectos con determinadas características como el nuestro, donde otras metodologías jugarían un mejor papel. Entre las características por lo cual no se escogió la metodología robusta RUP se encuentran:

- Nuestro proyecto no presenta una gran extensión ya que su dificultad se centra principalmente en la complejidad de su código, mientras que RUP está dirigido a proyectos de gran envergadura, donde haya gran cantidad de caso de usos.
- La razón principal por lo cual no se escogió RUP, es que los requisitos de la aplicación son muy variables y existe gran riesgo a la hora de implementarla (debido a que es una aplicación que se le debe aplicar técnicas de Inteligencia Artificial); por lo que esta metodología no responde muy bien a los cambios inesperados de requisitos, ya que la misma sigue una estructura esquemática que define el levantamiento de requisitos solamente al inicio de su realización (Flujo de requisitos).

Por lo anteriormente explicado se decide escoger una metodología ágil, ya que se adapta mejor a este trabajo.

### **Metodologías ágiles**

Las metodologías ágiles forman parte del movimiento de desarrollo ágil de software conocidas anteriormente como metodologías livianas, que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto. Se le denomina ágil como la habilidad de responder de forma versátil al cambio para maximizar los beneficios. Intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados. [7]

Las Metodologías Ágiles se basan en los siguientes principios:

- Realizar entregas cortas en el tiempo y continuas.

- Dar la bienvenida a los cambios.
- Entregas periódicas y frecuentes que funcionen.
- Los clientes forman parte del equipo de desarrollo.
- Equipo con individuos motivados. Darles para ello el ambiente, apoyo y confianza.
- La comunicación directa es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo. Intenta evitar el teléfono, correos electrónicos, fax, etc.
- La medida principal de progreso es el software que funciona.
- Desarrollo sostenible. Es indispensable que exista paz y armonía en el equipo para que el proyecto tenga éxito.
- Buen diseño y calidad técnica.
- La simplicidad es algo básico.
- Equipos autoorganizados.
- El equipo debe realizar reflexiones periódicamente para plantearse cómo llegar a ser más efectivo.

Entre las metodologías más conocidas están:

- XP (Extreme Programming).
- SCRUM.
- Metodología Crystal.
- Dynamic Systems Development Method (DSDM).
- Adaptive Software Development (ASD).
- Feature -Driven Development (FDD).
- Lean Development (LD).
- BUP.
- AUP.

Después de un detallado estudio de las distintas metodologías existentes, se llega a la conclusión que la que más se adapta a nuestro proyecto es la metodología SXP. Para comprender bien por qué se tomó esta decisión se expone a continuación un resumen de dicha metodología.

### SXP

SXP es una metodología compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepamos por dónde andamos.

XP más bien es una metodología encaminada para el desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Consta de 4 fases principales:

- **Planificación-Definición:** Donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- **Desarrollo:** Es donde se realiza la implementación del sistema hasta que esté listo para ser entregado;
- **Entrega:** Es la puesta en marcha.
- **Mantenimiento:** Es la fase donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, Diseño, Implementación, Pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que nos permite mejorar el diseño cada vez que se le añade una nueva funcionalidad.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo.[9]

A continuación se muestra una figura donde se expone el expediente del proyecto de la metodología así

como todos los artefactos que genera.



Figura 5: Expediente de proyecto.

### 1.7. Conclusiones

En este capítulo se ha hecho un profundo estudio en el área de la traducción automática, en donde se ha explicado detalladamente todos los conceptos referentes a dicho tema, los principales softwares de esa rama existente hoy día en el mercado, así como los lenguajes y herramientas propuestas para realizar esta tesis. Se considera que se ha cumplido los objetivos propuestos hasta el momento, debido a que se le ha dado al lector una base teórica extensa que le permitirá comprender los futuros aspectos que se abordarán en este trabajo.

## 2. CAPÍTULO 2: Descripción y análisis de la solución propuesta

### 2.1. Introducción

En este capítulo se dará una panorámica acerca de la descripción y el análisis de la solución que se propuso para resolver el problema que se planteó al inicio de esta tesis. Expondremos los requisitos funcionales y no funcionales de nuestro software así como la arquitectura elegida.

### 2.2. Valoración crítica del diseño

Luego de un detallado estudio de todos los traductores existentes en la actualidad, especialmente los expuestos anteriormente en el capítulo 1, los cuáles constituyen los principales exponentes de la traducción automática existentes hoy día; se llega a la conclusión de que ninguna de estas herramientas representa una solución integral, que se pudiera utilizar en el Koha para resolver el problema descrito al inicio de este trabajo. Entre las causas que justifican lo planteado se encuentran:

- Los mejores traductores que existen hoy día poseen licencia privativa, por lo cual no resulta factible emplearlos en este trabajo, el cual se basa en la licencia GPL.
- Los traductores existentes que poseen licencia GPL, no se pueden utilizar desde el SIGB Koha, ya que la gran mayoría se encuentran en aplicaciones de escritorio y no permiten la interacción este último.
- Una de las opciones que resulta más viable, es utilizar desde Koha el servicio web de traducción que brinda Google, pero se rechaza porque implica una dependencia del mismo.

Debido a esto, se decide implementar un sistema de traducción automática que se logre integrar a Koha y así poder dar solución al problema planteado. Este sistema se diseñará prácticamente desde cero, ya que solamente se utilizará el diccionario de pares de palabras español-inglés que posee el traductor StarDict y se desarrollarán el resto de los componentes que posee un traductor. Para lograr una mejor integración al SIGB Koha se decide realizar un módulo en Perl que realice la función de traducción y se pueda vincular al mismo. Para implementar dicho módulo, se discute acerca del tipo de traducción que poseería el mismo, eligiéndose finalmente la traducción **basada en reglas por transferencia**.

Entre las justificaciones de por qué se tomó esta decisión se encuentran:

- El tipo de traducción **basado en reglas** es sencillo de implementar en este caso, y no requiere de grandes requerimientos de hardware y software .
- Los otros tipos de traducción como los **basados en corpus-paralelos** y **basados en el contexto** presentan dificultades para su implementación, principalmente porque requieren de un corpus que tenga al menos un tamaño mínimo de 100 GB. Dicho corpus debe estar alineado en un formato electrónico llamado TMF y se necesitan grandes recursos de hardware.

La estructura del módulo de traducción **basada en reglas** es la misma que la que posee cualquiera de ese tipo. Está compuesto principalmente por cuatro fases o pasos principales, que componen la transformación que recibe la cadena de texto a traducir. A continuación se explica brevemente cada uno de estos pasos:

1- **Análisis morfológico:** En este paso se realiza un análisis morfológico de la cadena de texto a traducir, o sea, cada una de las palabras se clasifican en sustantivos, adjetivos, verbos, pronombres, etc. Normalmente se tendría una base de datos, en donde estaría cada palabra en español, unido a su significado en inglés, y su clasificación morfológica, por lo que este paso sería relativamente sencillo.

2- **Categorización léxica:** Este paso es usado para clasificar correctamente las posibles ambigüedades que pudiesen existir, o sea, pudiera darse el caso de que una misma palabra tenga dos clasificaciones morfológicas como por ejemplo en el caso de la palabra “**la**”, que se clasifica como un artículo o un pronombre personal dependiendo del contexto donde se encuentre, por lo que tendría dos posibles traducciones al inglés. En este caso se hace imprescindible clasificar estas palabras en dichas situaciones, por lo que se implementará un clasificador morfológico en el lenguaje lógico Prolog, el cual estaría compuesto por disímiles reglas de clasificación morfológica y que pudiera ser incluido y utilizado en nuestro módulo de Perl , gracias a otro módulo llamado **Al::Prolog**.

3-**Transferencia léxica:** Aquí se traduce palabra por palabra al inglés nuestra cadena de texto. Este paso se realiza buscando en la base de datos el respectivo significado en inglés de cada palabra, de acuerdo a la clasificación morfológica que posea en ese momento.

4-**Transferencia estructural y generador morfológico:** En este paso se reordenan las palabras y se busca la concordancia entre género y número de los sustantivos sus respectivos artículos. Además se le aplican varias reglas gramaticales.

## CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Hasta aquí se ha detallado brevemente la solución propuesta así como su funcionamiento. Este trabajo comprende además la interacción del módulo con el OPAC del SIGB Koha, por lo que se explicará a continuación como funcionará dicha interacción:

- Al SIGB Koha se le harán cambios que permitan al usuario escoger si desea realizar la búsqueda por defecto que trae el sistema o la búsqueda que realizará utilizando el módulo de traducción.
- El OPAC de Koha será capaz de capturar la cadena de texto introducida por el usuario, enviársela al módulo de traducción y posteriormente obtener la traducción de la cadena.
- Se buscará en la base de datos de Koha un registro bibliográfico que contenga la cadena anteriormente traducida.
- Se mostrará finalmente el resultado de la búsqueda.

### 2.3. Lista de reserva del producto

La lista de reservas de productos es una lista donde se exponen las tareas a realizar para desarrollar un producto, donde se priorizan las tareas de acuerdo a su importancia. Cuando empieza el proceso de desarrollo, es muy difícil tener claros todos los requerimientos, ya que sólo se poseen los más importantes. A medida que aumente el contacto con el cliente y se conozca más a fondo las especificaciones del proyecto esta lista podría aumentar, aunque sólo se pueden realizar cambios entre iteraciones.

Asignado a	Ítem *	Descripción	Estimación	Estimado por
<b>Prioridad</b>		Muy alta		
Dayron	1	Clasificar morfológicamente cada palabra.	10	Analista

## CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Yadier	2	Traducir palabra por palabra la cadena insertada.	2	Analista
Dayron	3	Aplicar reglas gramaticales a la cadena traducida.	6	Analista
Yadier	4	Gestionar la utilización del módulo de traducción en el OPAC.	2	Analista
Dayron	5	Mostrar en el “Koha” el resultado de la búsqueda.	1	Analista
<b>Prioridad</b>		Alta		
	1	Insertar palabras no clasificadas morfológicamente.	1	Analista
<b>Requisitos no funcionales</b>				
Dayron	1	El sistema deberá funcionar sobre el sistema operativo Debian etch.	1	Analista
Yadier	2	El sistema deberá tener como requisito de software, instalado el intérprete de Perl y de PHP para poder funcionar.	1	Analista
Dayron	3	El sistema deberá tener como requisito de software, instalado el módulo de Perl llamado AI::Prolog.	1	Analista

Dayron	4	El sistema tendrá como requisito de hardware una memoria RAM de 2 GB.	1	Analista
--------	---	---	---	----------

Tabla 2: Lista de reserva del producto.

## 2.4. Historias de usuarios arquitectónicamente más significativas

Una Historia de usuario es una representación de un requerimiento de software escrito en una o dos frases utilizando el lenguaje común del usuario. Las historias de usuarios son utilizadas en las metodologías de desarrollo ágiles para la especificación de requerimientos (acompañadas de pruebas de aceptación). Cada historia de usuario debe ser limitada, esta debería escribirse sobre una nota adhesiva pequeña. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definir las. Las historias de usuarios son una forma rápida de administrar los requerimientos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrar los. Las historias de usuarios permiten responder rápidamente a los requerimientos cambiantes.

Entre los beneficios de la historia de usuario se encuentran:

- Al ser muy corta, la misma representa requisitos del modelo de negocio que pueden implementarse rápidamente (días o semanas).
- Necesitan poco mantenimiento.
- Mantienen una relación cercana con el cliente.
- Permite dividir los proyectos en pequeñas entregas.
- Permite estimar fácilmente el esfuerzo de desarrollo.
- Ideal para proyectos con requerimientos volátiles o no muy claros.

<b>Historia de Usuario</b>	
<b>Número:</b> BNJM_T_01	<b>Nombre Historia de Usuario:</b> Clasificar morfológicamente cada

## CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

	palabra
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Dayron Reyes García	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 10 semanas
<b>Riesgo en Desarrollo:</b> Alto (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12 semanas
<p><b>Descripción:</b> En caso de que una misma palabra tenga dos o más clasificaciones morfológicas insertadas en la base de datos, es capaz de dar la clasificación morfológica correcta de dicha palabra.</p> <p><b>Observaciones:</b> En caso de que no sea capaz de determinar cuál es la clasificación morfológica correcta de la palabra se devolverá un mensaje negativo.</p>	
Prototipo de interfaces:	

Tabla 3: Historia de Usuario BNJM\_T\_01.

Historia de Usuario	
<b>Número:</b> BNJM_T_02	<b>Nombre Historia de Usuario:</b> Traducir palabra por palabra la cadena insertada
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yadier Alarcón Castell-Florit	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 2 semanas

## CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

<b>Riesgo en Desarrollo:</b> Bajo (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 2 semanas
<b>Descripción:</b> Se encarga de traducir del español al inglés todas las palabras introducidas por el usuario en su criterio de búsqueda, siempre y cuando las mismas se encuentren en la base de datos.	
<b>Observaciones:</b> Para poder traducir cada palabra es necesario poseer la clasificación morfológica de la misma, debido a que una palabra que posea diferentes clasificaciones poseerá diferentes significados en inglés.	
<b>Prototipo de interfaces:</b> Ninguno	

Tabla 4: Historia de Usuario BNJM\_T\_02.

Historia de Usuario	
<b>Número:</b> BNJM_T_03	<b>Nombre Historia de Usuario:</b> Gestionar la utilización del módulo de traducción en el OPAC
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b> Yadier Alarcón Castell-Florit	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 2 semanas
<b>Riesgo en Desarrollo:</b> Bajo (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 2 semanas
<b>Descripción:</b> Gestiona la utilización del módulo de traducción en el OPAC.	
<b>Observaciones:</b>	
<b>Prototipo de interfaces:</b> Ninguno	

Tabla 5: Historia de Usuario BNJM\_T\_03.

## 2.5. Arquitectura base del sistema

En los inicios de la informática, la programación se consideraba un arte y se desarrollaba como tal, debido a la dificultad que entrañaba para la mayoría de las personas, pero con el tiempo se han ido descubriendo y desarrollando formas y guías generales, en base a las cuales se puedan resolver los problemas. A estas, se les ha denominado Arquitectura de Software, porque a semejanza de los planos de un edificio o construcción, estas indican la estructura, funcionamiento e interacción entre las partes del software.

### 2.5.1. Arquitectura cliente-servidor

Debido a que esta tesis no consiste en realizar una aplicación independiente, sino desarrollar un módulo que se integre el sistema de búsqueda del gestor bibliotecario Koha; no se hizo necesario implantar una nueva arquitectura, sino más bien desarrollar en base a la que el SIGB Koha posee. Al ser Koha una aplicación web, la misma posee una arquitectura **cliente-servidor**.

Una arquitectura cliente servidor consiste básicamente en un programa cliente que realiza peticiones a otro programa: el servidor, que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora, es más ventajosa en un sistema operativo distribuido a través de una red de computadoras. La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

#### Características de un Cliente:

En la arquitectura Cliente/Servidor el **remite de una solicitud** es conocido como cliente. Sus características son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación

(dispositivo **maestro** o **amo**).

- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

### Características de un Servidor:

En los sistemas Cliente/Servidor el **receptor de la solicitud** enviada por cliente se conoce como servidor. Sus características son:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo **esclavo**).
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

### 2.5.2. Patrón arquitectónico Modelo-Vista-Controlador

El patrón arquitectónico que utiliza el SIGB Koha es el **patrón Modelo-Vista-Controlador** debido a que el diseño del mismo desacopla la Vista del Modelo, con el objetivo de mejorar la reusabilidad, ya que las modificaciones en la Vista impactan en menor medida en la lógica de negocio o datos.

#### Elementos del patrón:

- **Modelo:** Datos y reglas de negocio.
- **Vista:** Muestra la información del modelo al usuario.

- **Controlador:** Gestiona las entradas del usuario.

## 2.6. Base de datos

A continuación se muestra el modelo de datos empleado para la aplicación final. Se empleó el DBDesigner 4.0.5.4 para el diseño de la base de datos. Se utilizó Mysql no solo por su gran rapidez, sino por ser además una de las bases de datos que utiliza el Koha por defecto, por lo cual aumenta la compatibilidad.

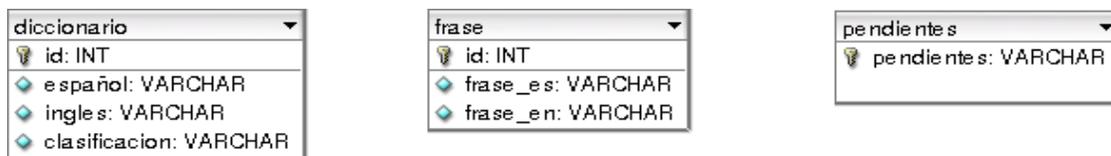


Figura 6: Base de datos.

La base de datos consta de tres tablas: "diccionario", "pendientes" y "frases". La tabla "diccionario" es la más importante de todas, ya que prácticamente es el corazón del sistema, debido a que contiene las palabras en español con sus respectivos significados en inglés y sus clasificaciones morfológicas. La tabla "pendiente" contiene todas aquellas palabras que han sido introducidas por el usuario en su criterio de búsqueda y que no se han encontrado en la tabla "diccionario", por la cual el sistema las introduce automáticamente en esta tabla, para que posteriormente un usuario calificado acceda a una aplicación web y las introduzca a la tabla "diccionario", con su significado en inglés y su clasificación morfológica. Por su parte la tabla "frase" contiene frases idiomáticas propias del inglés, lo cual servirá en caso que un título de un libro corresponda a una frase idiomática.

## 2.7. Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc.

Los pasos a tener en cuenta para la realización del diagrama de componentes son los siguientes :

- Identificar subsistemas de implementación.
- Identificar los componentes, entre ellos los que serán reutilizados. Realizar un diagrama por cada subsistema.
- Identificar estereotipos de los componentes.
- Considerar las relaciones entre componentes.

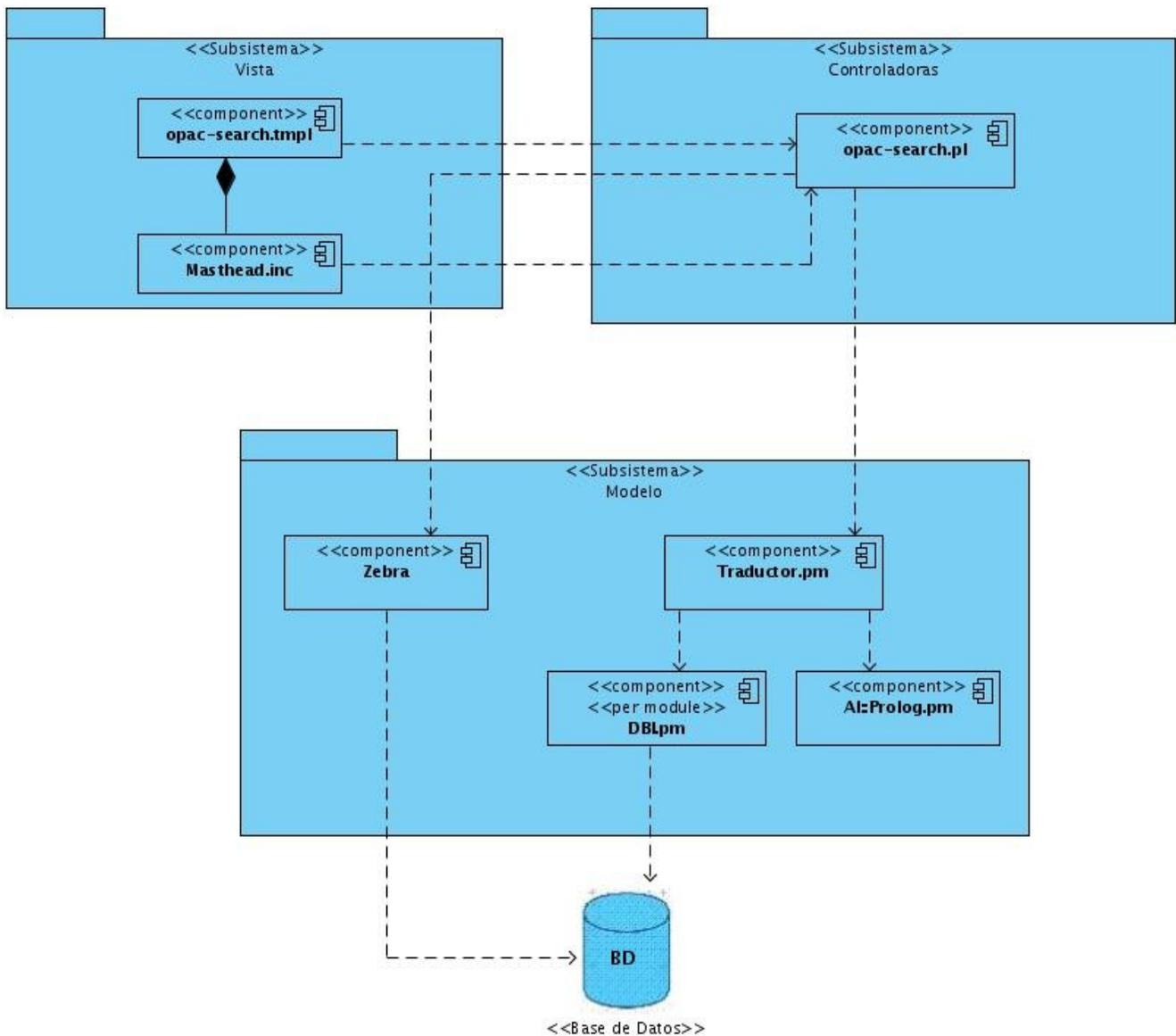


Figura 7: Diagrama de componentes.

El diagrama de componentes anteriormente mostrado se divide en tres subsistemas fundamentales: Modelo, Vista y Controladoras. Dichos subsistemas representan cada una de las capas lógicas que constituyen la aplicación. El subsistema Vista constituye la interfaz de la aplicación sobre la cual el cliente va a interactuar, destacándose el componente **opac-search.tmpl**. El subsistema Controladoras

constituye la capa que se encarga de controlar la aplicación, en el mismo se encuentra el componente **opac-search.pl** . Por último, tenemos al subsistema Modelo que comprende todas aquellas librerías utilizadas en el Koha para poder acceder a bases de datos, la base de datos Mysql, además nuestro módulo de traducción llamado **traducción.pm**.

### 2.8. Conclusiones

Con la realización de este capítulo, se realiza un análisis sobre la solución propuesta. Se describe de forma general como debe funcionar la aplicación y porque se escogió la traducción basada en reglas para implementar el sistema por encima de otros tipos de traducción existentes en la actualidad. Se plantea la arquitectura base que se considera debe poseer el sistema, además del modelo de datos y diagrama de componentes del mismo.

### 3. CAPÍTULO 3: Desarrollo ágil de la aplicación propuesta

#### 3.1. Introducción

Después de analizar exhaustivamente en el capítulo anterior, la solución propuesta para esta tesis. En este capítulo se abordará la fase de implementación, por lo que se mostrarán disímiles plantillas propias de la metodología SXP, las cuales servirán para mostrarle al usuario de una manera intuitiva los pasos seguidos en la realización de la misma. Además se expondrá detalladamente cada **historia de usuario**, incluyendo también las **tareas de ingeniería** asociadas a la misma.

#### 3.2. Planificación por roles

Rol	Responsabilidad	Nombre
Gerente (Management)	Es el responsable de tomar las decisiones finales acerca de estándares y convenciones a seguir durante el proyecto.	Dayron Reyes García Yadier Alarcón Castell-Florit
Cliente (Customer)	El cliente participa en las tareas que involucra la lista de reserva del producto.	Biblioteca Nacional "José Martí"
Programadores (Programmers)	Es el encargado de producir el código y escribir las pruebas unitarias. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.	Yadier Alarcón Castell-Florit Dayron Reyes García
Analista (Analyst)	Es el encargado de escribir las historias de usuario y las pruebas	Dayron Reyes García Yadier Alarcón Castell-Florit

## CAPITULO 3: DESARROLLO ÁGIL DE LA APLICACIÓN PROPUESTA

	funcionales para validar su implementación.	
Diseñadores (Designers)	Encargados del diseño del sistema; así como el de los prototipos de interfaces. Máximos responsables de la realización del diseño de las metáforas y supervisan el proceso de construcción.	Dayron Reyes García Yadier Alarcón Castell-Florit
Encargado de Pruebas (Tester)	Es el encargado de ayudar al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.	Dayron Reyes García Yadier Alarcón Castell-Florit
Arquitecto (Architect)	Se vincula directamente con el analista y el diseñador, debido a que su trabajo tiene que ver con la estructura y el diseño del sistema. Ayuda en el diseño de las metáforas.	Dayron Reyes García Yadier Alarcón Castell-Florit

Tabla 6: Planificación por roles.

### 3.3. Modelo de dominio

Una de las actividades claves que se realizan en la metodología SXP es el Modelo de Historias de Usuario del Negocio, donde se define el Modelo de Negocio como tal. El mismo es importante porque permite comprender la estructura y la dinámica de la organización donde se va a implantar el sistema. En este caso, se consideró que negocio no esté bien definido entre los clientes y los ejecutores del proyecto,

por lo que se realiza el Modelo de Dominio.

Representación de clases conceptuales del dominio del problema

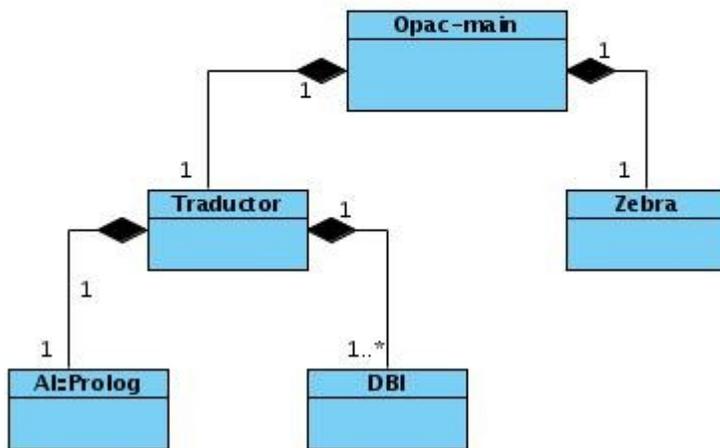


Figura 8: Modelo de dominio.

### 3.4. Historias de usuario y prototipos de interfaces de usuario

A continuación mostramos las distintas historias de usuario, los prototipos de interfaces de usuario, así como cada una de las tareas de ingeniería asociadas y los integrantes del equipo que se encargan de desarrollarlas. Solamente se expone una planificación inicial de la mismas, ya una de las características principales de la metodología SXP es que está diseñada para trabajar en entornos que cambian continuamente como es el caso de esta tesis, por lo que es posible que posteriormente se introduzcan cambios de acuerdo a las necesidades y las propuestas que realice el cliente.

Historia de Usuario	
Número: BNJM_T_01	Nombre Historia de Usuario: Clasificar morfológicamente cada palabra
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Dayron Reyes García	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 10 semanas

**CAPITULO 3: DESARROLLO ÁGIL DE LA APLICACIÓN PROPUESTA**

(Alta / Media / Baja)	
<b>Riesgo en Desarrollo:</b> Alto (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12 semanas
<b>Descripción:</b> En caso de que una misma palabra tenga dos o más clasificaciones morfológicas insertadas en la base de datos, es capaz de dar la clasificación morfológica correcta de dicha palabra.	
<b>Observaciones:</b> En caso de que no sea capaz de determinar cuál es la clasificación morfológica correcta de la palabra se devolverá un mensaje negativo.	
<b>Prototipo de interfaces:</b> Ninguno	

Tabla 7: Historia de Usuario BNJM\_T\_01.

Tarea de Ingeniería	
<b>Número Tarea:</b> BNJM_T_01 -1	<b>Número Historia de Usuario:</b> BNJM_T_01
<b>Nombre Tarea:</b> Verificar existencia de palabras	
<b>Tipo de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b> 0.2 semanas
<b>Fecha Inicio:</b> 13/01/2009	<b>Fecha Fin:</b> 15/01/2009
<b>Programador Responsable:</b> Dayron Reyes García	
<b>Descripción:</b> Verifica que las palabras introducidas por el usuario en su criterio de búsqueda existan en la base de datos que sirve como diccionario, y que además estén clasificadas morfológicamente en la misma. En caso contrario, la insertará a la tabla “pendientes” de la base de datos “Traducción”, en la cual se encuentran las palabras que no existen en la base de datos .	

Tabla 8: Tarea de Ingeniería BNJM\_T\_01 -1.

Tarea de Ingeniería	
<b>Número Tarea:</b> BNJM_T_01 - 2	<b>Número Historia de Usuario:</b> BNJM_T_01
<b>Nombre Tarea:</b> Consultar clasificaciones morfológicas de palabras	

**CAPITULO 3: DESARROLLO ÁGIL DE LA APLICACIÓN PROPUESTA**

<b>Tipo de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b> 1 semana
<b>Fecha Inicio:</b> 16/01/2009	<b>Fecha Fin:</b> 23/01/2009
<b>Programador Responsable:</b> Dayron Reyes García	
<b>Descripción:</b> Realiza una consulta a la base de datos, obteniendo las distintas clasificaciones morfológicas que posee cada palabra introducida por el usuario en el criterio de búsqueda.	

Tabla 9: Tarea de Ingeniería BNJM\_T\_01 -2.

Tarea de Ingeniería	
<b>Número Tarea:</b> BNJM_T_01-3	<b>Número Historia de Usuario:</b> BNJM_T_01
<b>Nombre Tarea:</b> Clasificar morfológicamente las palabras en el Prolog	
<b>Tipo de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b> 5 semanas
<b>Fecha Inicio:</b> 24/01/2009	<b>Fecha Fin:</b> 28/02/2009
<b>Programador Responsable:</b> Dayron Reyes García	
<b>Descripción:</b> Clasifica morfológicamente las palabras que poseen dos o más clasificaciones utilizando un conjunto de reglas implementadas en el Prolog.	

Tabla 10: Tarea de Ingeniería BNJM\_T\_01-3.

Historia de Usuario	
<b>Número:</b> BNJM_T_02	<b>Nombre Historia de Usuario:</b> Traducir palabra por palabra la cadena insertada
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b> Yadier Alarcón Castell-Florit	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 2 semanas
<b>Riesgo en Desarrollo:</b> Bajo (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 2 semanas

### CAPITULO 3: DESARROLLO ÁGIL DE LA APLICACIÓN PROPUESTA

<b>Descripción:</b> Se encarga de traducir del español al inglés todas las palabras introducidas por el usuario en su criterio de búsqueda, siempre y cuando las mismas se encuentren en la base de datos
<b>Observaciones:</b> Para poder traducir las palabras es necesario poseer la clasificaciones morfológicas de las mismas, debido a que una palabra que posea diferentes clasificaciones poseerá diferentes significados en inglés.
<b>Prototipo de interfaces:</b> Ninguno

Tabla 11: Historia de Usuario BNJM\_T\_02.

Tarea de Ingeniería	
<b>Número Tarea:</b> BNJM_T_02-1	<b>Número Historia de Usuario:</b> BNJM_T_02
<b>Nombre Tarea:</b> Ver significados en inglés de las palabras	
<b>Tipo de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b> 0.2 semanas
<b>Fecha Inicio:</b> 29/02/2009	<b>Fecha Fin:</b> 2/03/2009
<b>Programador Responsable:</b> Yadier Alarcón Castell-Florit	
<b>Descripción:</b> Realiza una consulta en la base de datos donde se buscan los significados en inglés que posee cada palabra atendiendo a su significado en español y su clasificación morfológica.	

Tabla 12: Tarea de Ingeniería BNJM\_T\_02-1.

Tarea de Ingeniería	
<b>Número Tarea:</b> BNJM_T_02-2	<b>Número Historia de Usuario:</b> BNJM_T_02
<b>Nombre Tarea:</b> Crear combinaciones de palabras	
<b>Tipo de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b> 2.8 semanas
<b>Fecha Inicio:</b> 3/03/2009	<b>Fecha Fin:</b> 23/03/2009
<b>Programador Responsable:</b> Yadier Alarcón Castell-Florit	
<b>Descripción:</b> Debido a que una palabra que posea distintas clasificaciones morfológicas poseerá a la vez diferentes significados en inglés, cuando el clasificador morfológico implementado en Prolog no es	

capaz de clasificar correctamente una palabra desde el punto de vista morfológico, entonces esta tarea se encarga de crear todas las posibles combinaciones de palabras que pudieran surgir, en donde cada combinación tendría la palabra que no ha podido ser clasificada correctamente con un significado en inglés, más las otras palabras introducidas por el usuario las cuales si han podido ser clasificadas morfológicamente también con sus respectivos significados en inglés, lo cual da todos los significados posibles en inglés. Solo se usa en el caso de que no se pueda clasificar correctamente la palabra desde el punto de vista morfológico.

Tabla 13: Tarea de Ingeniería BNJM\_T\_02-2.

<b>Historia de Usuario</b>	
<b>Número:</b> BNJM_T_03	<b>Nombre Historia de Usuario:</b> Aplicar reglas gramaticales a la cadena traducida
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b> Dayron Reyes García	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6 semanas
<b>Riesgo en Desarrollo:</b> Bajo (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6 semanas
<b>Descripción:</b> Le aplica disímiles reglas gramaticales a la cadena de texto que ha sido traducida palabra por palabra.	
<b>Observaciones:</b>	
<b>Prototipo de interfaces:</b> Ninguno	

Tabla 14: Historia de Usuario BNJM\_T\_03.

<b>Tarea de Ingeniería</b>
----------------------------

**CAPITULO 3: DESARROLLO ÁGIL DE LA APLICACIÓN PROPUESTA**

<b>Número Tarea:</b> BNJM_T_03-1	<b>Número Historia de Usuario:</b> BNJM_T_03
<b>Nombre Tarea:</b> Aplicar regla gramatical del adjetivo primero que el sustantivo	
<b>Tipo de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b> 1.5 semanas
<b>Fecha Inicio:</b> 24/03/2009	<b>Fecha Fin:</b> 3/04/2009
<b>Programador Responsable:</b> Dayron Reyes García	
<b>Descripción:</b> Aplicar la regla gramatical del adjetivo primero que el sustantivo a las palabras traducidas al inglés en donde cada vez que se encuentra una combinación de un sustantivo primero que un adjetivo, como en el caso de <b>"house beautiful"</b> lo convierte a <b>"beatiful house"</b> .	

Tabla 15: Tarea de Ingeniería BNJM\_T\_03-1.

Tarea de Ingeniería	
<b>Número Tarea:</b> BNJM_T_03-2	<b>Número Historia de Usuario:</b> BNJM_T_03
<b>Nombre Tarea:</b> Aplicar regla gramatical de pertenencia	
<b>Tipo de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b> 1.5 semanas
<b>Fecha Inicio:</b> 4/04/2009	<b>Fecha Fin:</b> 15/04/2009
<b>Programador Responsable:</b> Dayron Reyes García	
<b>Descripción:</b> Se le aplica la regla gramatical de "pertenencia" la cual consiste en aplicar una transformación a la cadena de texto traducida palabra por palabra al inglés, cada vez que vea el siguiente patrón "sustantivo + preposición + sustantivo" como es el caso del siguiente ejemplo:  Ejemplo: Cada vez que se encuentra la siguiente ocurrencia <b>"history of toy"</b> lo convierte a <b>"toy history"</b> .	

Tabla 16: Tarea de Ingeniería BNJM\_T\_03-2.

Tarea de Ingeniería	
<b>Número Tarea:</b> BNJM_T_03-3	<b>Número Historia de Usuario:</b> BNJM_T_03
<b>Nombre Tarea:</b> Aplicar regla gramatical de pertenencia1	

**CAPITULO 3: DESARROLLO ÁGIL DE LA APLICACIÓN PROPUESTA**

<b>Tipo de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b> 1.5 semanas
<b>Fecha Inicio:</b> 16/04/2009	<b>Fecha Fin:</b> 26/04/2009
<b>Programador Responsable:</b> Dayron Reyes García	
<p><b>Descripción:</b> Se le aplica la regla gramatical de “pertenencia1” la cual consiste en aplicar una transformación a la cadena de texto traducida palabra por palabra al inglés, cada vez que vea el siguiente patrón “sustantivo + preposición + sustantivo” como es el caso del siguiente ejemplo.</p> <p>Ejemplo: Cada vez que se encuentra la siguiente ocurrencia “<b>house of Dayron</b>” lo convierte a “<b>Dayron’s house</b>”.</p>	

Tabla 17: Tarea de Ingeniería BNJM\_T\_03-3.

Tarea de Ingeniería	
<b>Número Tarea:</b> BNJM_T_03-4	<b>Número Historia de Usuario:</b> BNJM_T_03
<b>Nombre Tarea:</b> Aplicar regla gramatical del “pronombre personal1”	
<b>Tipo de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b> 1.5 semanas
<b>Fecha Inicio:</b> 26/04/2009	<b>Fecha Fin:</b> 2/05/2009
<b>Programador Responsable:</b> Dayron Reyes García	
<p><b>Descripción:</b> Se le aplica la regla gramatical de “pronombre personal1” la cual consiste en aplicar una transformación a la cadena de texto traducida palabra por palabra al inglés, la cual ubica un pronombre personal que hace función de complemento directo detrás del verbo, por ejemplo: el siguiente fragmento en español: “<b>la mató</b>”, se traduce palabra por palabra “<b>her kill</b>”, pero aplicándole esta regla quedaría correctamente de la siguiente manera “<b>kill her</b>”.</p>	

Tabla 18: Tarea de Ingeniería BNJM\_T\_03-4.

Historia de Usuario	
<b>Número:</b> BNJM_T_04	<b>Nombre Historia de Usuario:</b> Gestionar la utilización del módulo de traducción en el OPAC
Modificación de Historia de Usuario Número:	

**CAPITULO 3: DESARROLLO ÁGIL DE LA APLICACIÓN PROPUESTA**

<b>Usuario:</b> Yadier Alarcón Castell-Florit	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 2 semanas
<b>Riesgo en Desarrollo:</b> Bajo (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 2 semanas
<b>Descripción:</b> Gestiona la utilización del módulo de traducción en el OPAC.	
<b>Observaciones:</b>	
Prototipo de interfaces: Ninguno	

Tabla 19: Historia de Usuario BNJM\_T\_04.

Tarea de Ingeniería	
<b>Número Tarea:</b> BNJM_T_04-1	<b>Número Historia de Usuario:</b> BNJM_T_04
<b>Nombre Tarea:</b> Capturar la cadena de texto	
<b>Tipo de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b> 0.5 semanas
<b>Fecha Inicio:</b> 2/05/2009	<b>Fecha Fin:</b> 6/05/2009
<b>Programador Responsable:</b> Yadier Alarcón Castell-Florit	
<b>Descripción:</b> Se captura la cadena de texto introducida por el usuario en el OPAC y se le pasa como parámetro al módulo de traducción.	

Tabla 20: Tarea de Ingeniería BNJM\_T\_04-1.

Tarea de Ingeniería	
<b>Número Tarea:</b> BNJM_T_04-2	<b>Número Historia de Usuario:</b> BNJM_T_04
<b>Nombre Tarea:</b> Capturar resultado del módulo de traducción	

**CAPITULO 3: DESARROLLO ÁGIL DE LA APLICACIÓN PROPUESTA**

<b>Tipo de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b> 0.5 semanas
<b>Fecha Inicio:</b> 6/05/2009	<b>Fecha Fin:</b> 9/05/2009
<b>Programador Responsable:</b> Yadier Alarcón Castell-Florit	
<b>Descripción:</b> Se captura el resultado del módulo de traducción o sea la(s) cadena(s) de texto finalmente traducidas.	

Tabla 21: Tarea de Ingeniería BNJM\_T\_04-2.

Tarea de Ingeniería	
<b>Número Tarea:</b> BNJM_T_04-3	<b>Número Historia de Usuario:</b> BNJM_T_04
<b>Nombre Tarea:</b> Buscar en la base de datos del Koha	
<b>Tipo de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b> 1 semana
<b>Fecha Inicio:</b> 9/05/2009	<b>Fecha Fin:</b> 16/05/2009
<b>Programador Responsable:</b> Yadier Alarcón Castell-Florit	
<b>Descripción:</b> De acuerdo al resultado obtenido en el módulo de traducción, busca en la base datos del Koha algún libro que cumpla ese criterio.	

Tabla 22: Tarea de Ingeniería BNJM\_T\_04-3.

Historia de Usuario	
<b>Número:</b> BNJM_T_05	<b>Nombre Historia de Usuario:</b> Mostrar en el Koha el resultado de la búsqueda
Modificación de Historia de Usuario Número:	
<b>Usuario:</b> Dayron Reyes García	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 1 semana

**CAPITULO 3: DESARROLLO ÁGIL DE LA APLICACIÓN PROPUESTA**

<b>Riesgo en Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 1 semana
<b>Descripción:</b> Se muestra en el Koha, el resultado final obtenido de la búsqueda en la base de datos del Koha, concatenado a la búsqueda por defecto que el mismo realiza.	
<b>Observaciones:</b>	
<b>Prototipo de interfaces:</b> Anexo # 6	

Tabla 23: Historia de Usuario BNJM\_T\_05.

<b>Historia de Usuario</b>	
<b>Número:</b> BNJM_T_06	<b>Nombre Historia de Usuario:</b> Insertar palabras no clasificadas morfológicamente
Modificación de Historia de Usuario Número:	
<b>Usuario:</b> Yadier Alarcón Castell-Florit	<b>Iteración Asignada:</b> 4
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 1 semana
<b>Riesgo en Desarrollo:</b> Bajo (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 1 semana
<b>Descripción:</b> Se insertan en la base de datos, las palabras que se encuentran en la tabla “pendientes”, además de su significado en inglés y su clasificación morfológica	
<b>Observaciones:</b> Se muestra a través de una interfaz web todas las palabras que se encuentran pendientes de clasificación morfológica o que no tienen su respectivo significado en inglés, y el usuario tiene la opción de completarlas.	
<b>Prototipo de interfaces:</b> Anexo # 7	

Tabla 24: Historia de Usuario BNJM\_T\_06.

### 3.5. Plan de release

El plan de release es otra de las plantillas existentes en la metodología SXP, en la cual define cuales son las historias de usuario más significativas e importantes, posicionando a las mismas en las primeras iteraciones, además de organizar el proceso de software en iteraciones y planificar el trabajo a realizar en cada una de ellas. Cada iteración tiene una fecha fija para su consecución, así como un alcance variable.

<b>Release</b>	<b>Descripción de la iteración</b>	<b>Orden de la HU a implementar</b>	<b>Duración total</b>
Iteración1	En esta iteración se desarrollan las condiciones necesarias para que el módulo traduzca al inglés palabra por palabra.	BNJM_T_1 BNJM_T_2	13/01/09 - 23/03/09
Iteración2	En esta iteración el módulo será capaz de traducir al inglés de una manera más refinada, ya que se le aplicarán reglas gramaticales.	BNJM_T_3	24/03/09 - 02/05/09
Iteración3	En esta iteración el módulo estará vinculado al OPAC del Koha y se podrán realizar búsquedas desde el mismo utilizando dicho módulo.	BNJM_T_4 BNJM_T_5	02/05/09 – 16/05/09
Iteración4	En esta iteración, un usuario calificado, a través de una interfaz web , podrá insertar todas aquellas palabras que no se encuentran en la base de datos.	BNJM_T_6	16/05/09 -23/05/09

Tabla 25: Plan de release.

### **3.6. Conclusiones**

A través de las diversas plantillas mostradas anteriormente se ha mostrado todo el proceso de implementación desarrollado en nuestra tesis, por lo que se espera que este capítulo haya servido de documentación para aquellos desarrolladores que quisieran aportar su granito de arena al desarrollo de un sistema de búsqueda inteligente para el sistema de búsqueda del SIGB Koha.

## 4. CAPÍTULO 4: Validación de la solución propuesta

### 4.1. Introducción

Una vez que se ha mostrado la implementación de esta tesis, se pasará a la validación de la misma, con el objetivo de que el producto final sea una aplicación robusta y fiable para el usuario y además asegurar la calidad requerida en nuestra universidad. Para ello se desarrollará una serie de pruebas a través de los llamados casos de prueba de aceptación que se desarrollarán en cada una de las iteraciones, mostrándose los resultados obtenidos en las mismas y presentando además las funcionalidades alcanzadas hasta el período de desarrollo.

### 4.2. Casos de prueba de aceptación

Un caso de prueba de aceptación no es más que la prueba que se le realiza a cada historia de usuario. A continuación se muestra la descripción correspondiente a cada caso de prueba de aceptación.

#### Caso de prueba para la Historia de usuario BNJM\_T\_01:

En esta sección se le realiza una prueba a la historia de usuario: **Clasificar morfológicamente cada palabra**. Su objetivo principal es ver si el módulo es capaz de clasificar correctamente desde el punto de vista morfológico una determinada palabra que posea dos o más clasificaciones, o sea, que determine si la misma es sustantivo, adjetivo, verbo, pronombre personal, etc.

Caso de Prueba de Aceptación	
Código Caso de Prueba: BNJM-1-1	<b>Nombre Historia de Usuario:</b> Clasificar morfológicamente cada palabra
<b>Nombre de la persona que realiza la prueba:</b> Dayron Reyes García	
<b>Descripción de la Prueba:</b> Se clasifican morfológicamente las palabras introducidas.	
<b>Condiciones de Ejecución:</b> Que la palabra que se va a clasificar posea dos o más clasificaciones.	

<p><b>Entrada / Pasos de ejecución:</b> Se entran las palabras a clasificar junto a sus distintas clasificaciones morfológicas, además de las palabras que la rodean, también con sus respectivas clasificaciones.</p>
<p><b>Resultado Esperado:</b> Clasifica correctamente en la mayoría de los casos, ya que existen otros en los cuáles es muy difícil que lo haga, debido a que en el inglés existen un sinnúmero de reglas gramaticales y es muy difícil plasmarlas todas. Además de que la mayoría de los traductores del mundo tampoco son capaces de hacerlo correctamente.</p>
<p><b>Evaluación de la Prueba:</b> Satisfactoria</p>

Tabla 26: Caso de Prueba de Aceptación BNJM-1-1.

**Caso de prueba para la Historia de usuario BNJM\_T\_02:**

A continuación se le hará una prueba a la historia de usuario: **Traducir palabra por palabra la cadena insertada**. Su objetivo principal es ver traducida al inglés cada una de las palabras que componen la cadena de texto insertada.

<b>Caso de Prueba de Aceptación</b>	
Código Caso de Prueba: BNJM-1-2	<b>Nombre Historia de Usuario:</b> Traducir palabra por palabra la cadena insertada
<b>Nombre de la persona que realiza la prueba:</b> Yadier Alarcón Castell-Florit	
<b>Descripción de la Prueba:</b> Se traduce palabra por palabra al inglés la cadena de texto introducida.	
<b>Condiciones de Ejecución:</b> Que todas las palabras a introducir estén clasificadas correctamente desde el punto de vista morfológico.	
<b>Entrada/Pasos de ejecución:</b> Se entran cada una de las palabras a traducir al inglés con sus respectivas clasificaciones morfológicas.	
<b>Resultado Esperado:</b> Traduce correctamente todas las palabras introducidas.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

## CAPITULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 27: Caso de Prueba de Aceptación BNJM-1-2.

**Caso de prueba para la Historia de usuario BNJM\_T\_03:** A continuación se le hará una prueba a la historia de usuario: **Aplicar reglas gramaticales a la cadena traducida**. Su objetivo principal es ver si el módulo es capaz de aplicar varias reglas gramaticales a la cadena de texto que ha sido anteriormente traducida palabra por palabra al inglés.

Caso de Prueba de Aceptación	
Código Caso de Prueba: BNJM-1-3	<b>Nombre Historia de Usuario:</b> Aplicar reglas gramaticales a la cadena traducida
<b>Nombre de la persona que realiza la prueba:</b> Dayron Reyes García	
<b>Descripción de la Prueba:</b> Se le aplica varias reglas gramaticales a la cadena anteriormente traducida palabra por palabra al inglés.	
<b>Condiciones de Ejecución:</b> Que la cadena de texto introducida haya sido anteriormente traducida palabra por palabra al inglés.	
<b>Entrada/Pasos de ejecución:</b> Se entra la cadena de texto previamente traducida palabra por palabra al inglés.	
<b>Resultado Esperado:</b> Se le aplican correctamente las reglas gramaticales a la cadena de texto.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Tabla 28: Caso de Prueba de Aceptación BNJM-1-3.

**Caso de prueba para la Historia de usuario BNJM\_T\_04:**

A continuación se le hará una prueba la historia de usuario: **Gestionar la utilización del módulo de traducción en el OPAC**. Su objetivo principal es ver si esta historia de usuario es capaz de permitir la interacción del modulo de traducción con el OPAC del Koha.

Caso de Prueba de Aceptación	
Código Caso de Prueba:	<b>Nombre Historia de Usuario:</b> Gestionar la utilización del módulo de traducción en el OPAC

## CAPITULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

BNJM-1-4	
<b>Nombre de la persona que realiza la prueba:</b> Yadier Alarcón Castell-Florit	
<b>Descripción de la Prueba:</b> Permitir la interacción del módulo de traducción con el OPAC del Koha permitiendo que sea capaz de capturar la cadena de texto introducida por el usuario al buscar un libro en el OPAC del Koha, pasarle dicha cadena al módulo de traducción y recibir el resultado del mismo. Además de realizar la búsqueda en la base de datos del Koha atendiendo al resultado devuelto por el módulo de traducción.	
<b>Condiciones de Ejecución:</b> Que el módulo de traducción se encuentre funcionando correctamente.	
<b>Entrada / Pasos de ejecución:</b> Se le introduce una cadena de texto consistente en el criterio de búsqueda introducido por el usuario al OPAC del Koha.	
<b>Resultado Esperado:</b> Se gestiona correctamente la interacción del módulo de traducción con el OPAC del Koha, capturando correctamente la cadena de texto introducida por el usuario y pasándole la misma al módulo de traducción, además de capturar satisfactoriamente el resultado devuelto por el mismo y buscar en la base de datos del Koha dicho resultado.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Tabla 29: Caso de Prueba de Aceptación BNJM-1-4.

### Caso de prueba para la Historia de usuario BNJM\_T\_05:

A continuación se le hará una prueba la historia de usuario: **Mostrar en el Koha el resultado de la búsqueda**. Su objetivo principal es ver si esta historia de usuario muestra correctamente en el SIGB Koha el resultado final obtenido de la historia de usuario **gestionar la utilización del módulo de traducción en el OPAC**.

Caso de Prueba de Aceptación	
Código Caso de Prueba: BNJM-1-5	<b>Nombre Historia de Usuario:</b> Mostrar en el Koha el resultado de la búsqueda

<b>Nombre de la persona que realiza la prueba:</b> Dayron Reyes García
<b>Descripción de la Prueba:</b> Se muestra en el Koha el resultado final de la búsqueda.
<b>Condiciones de Ejecución:</b> Tiene que funcionar correctamente el módulo de traducción así como la historia de usuario <b>Gestionar la utilización del módulo de traducción en el OPAC.</b>
<b>Entrada / Pasos de ejecución:</b> Se le introduce la cadena de texto obtenida de la historia de usuario <b>Gestionar la utilización del módulo de traducción en el OPAC.</b>
<b>Resultado Esperado:</b> Se muestra correctamente el resultado final en el Koha.
<b>Evaluación de la Prueba:</b> Satisfactoria

Tabla 30: Caso de Prueba de Aceptación BNJM-1-5.

**Caso de prueba para la Historia de usuario BNJM\_T\_06:**

A continuación se le hará una prueba la historia de usuario: Insertar palabras no clasificadas morfológicamente. Su objetivo es ver si esta historia de usuario es capaz de mostrar en una interfaz web todas las palabras que se encuentran almacenadas en la tabla “pendientes” de la base de datos “traducción” y posteriormente el usuario sea capaz de insertar las mismas en la tabla “diccionario”.

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> BNJM-1-6	<b>Nombre Historia de Usuario:</b> Insertar palabras no clasificadas morfológicamente
<b>Nombre de la persona que realiza la prueba:</b> Yadier Alarcón Castell-Florit	

<b>Descripción de la Prueba:</b> Mostrar en una interfaz web todas las palabras que se encuentran almacenadas en la tabla “pendientes” de la base de datos “traducción” y posteriormente el usuario sea capaz de insertar las mismas en la tabla “diccionario”.
<b>Condiciones de Ejecución:</b> En la tabla “pendientes” de la base de datos “traducción ” debe haber insertada al menos una palabra.
<b>Entrada / Pasos de ejecución:</b>
<b>Resultado Esperado:</b> Se muestran satisfactoriamente las palabras de la tabla “pendiente” y se pueden insertar satisfactoriamente dichas palabras con sus respectivas clasificaciones y significados en inglés en la tabla “diccionario”.
<b>Evaluación de la Prueba:</b> Satisfactoria

Tabla 31: Caso de Prueba de Aceptación BNJM-1-6.

### 4.3. Conclusiones

Una vez terminado este capítulo, en donde se trataron las distintas pruebas que se realizaron de una manera satisfactoria a nuestro software, se puede afirmar que el mismo se encuentra en funcionalidad operativa y listo para ser usado.

## **Conclusiones**

---

Con el gran volumen de información que se maneja actualmente en la bibliotecas, se hace necesario contar con sistemas de gestión que faciliten las búsquedas a los usuarios, incluyendo facilidades que permitan encontrar la información requerida, tales como la traducción automática.

Se han añadido nuevas funcionalidades al SIGB Koha que permiten a los usuarios realizar búsquedas más eficientes, evitando tener que realizar búsquedas innecesarias.

Con la implementación del módulo de traducción automática se han aplicado técnicas de inteligencia artificial al SIGB Koha, con las cuales no se contaba hasta el momento.

## Recomendaciones

---

Se recomienda que:

- Se amplíe el diccionario bilingüe de la aplicación, insertándole más palabras a la base de datos.
- Se le introduzcan más reglas de clasificación morfológica al pequeño clasificador morfológico implementado en Prolog.
- Se le introduzcan más reglas gramaticales a la parte que se encarga de la transferencia léxica y estructural.
- Se hagan investigaciones acerca de la traducción automática basada en el contexto, para estudiar la posibilidad de implementar un sistema de traducción automático basado en este método, ya que el mismo es el más preciso que existe actualmente.

## Referencias bibliográficas

---

- [1] Elies, M. *Traducción*. [citado el: 24 Enero 2009]. Disponible en: <http://elies.rediris.es/elies9/3-3-1.htm>
- [2] Anjuta Team. Anjuta. [citado el: 24 Enero 2009]. Disponible en: <http://anjuta.org/>.
- [3] Eric Laffoon. Quanta Plus. [citado el: 23 Enero 2009]. Disponible en: <http://quanta.sourceforge.net/>.
- [4] ÁLVARO, M, E., Manual Práctico de HTML. [citado el: 25 Enero 2009]. Disponible en: <http://www-app.etsit.upm.es/~alvaro/manual/manual.html>.
- [5] GLADYS MARSI, P. R. MA-GMPR-UR2\_MA-SXP. 23.10.2008  
Sánchez, M. A. M 2002
- [6] Sanchez, Metodologías *de desarrollo.pdf*, December 2002 [citado el: 20 Enero 2009].
- [7] José H. Canós, Patricio Letelier, y Msc Carmen Penadés. Metodologías Ágiles. *pdf*. December 2005 [citado el: 20 Enero 2009].
- [8] GLADYS MARSI, P. R. MA-GMPR-UR2. Metodología ágil para proyectos de software libre, 23.06.2008.
- [9] YANIRIS ROIG MOREJÓN, ROBERTO LAGO BATISTA, Análisis y diseño del módulo gestión de tesis, 1.06.2007.

## Bibliografía

---

- 1-CPAN. Comprehensive Perl Archive Network. [En línea] [Citado el: 11 de enero del 2009.]  
<http://search.cpan.org/~nwclark/perl-5.8.9/lib/strict.pm>.
- 2-Curtis "Ovid" Poe. Módulo AI::Prolog. [En línea] [Citado el: 7 de febrero del 2009.]  
<http://search.cpan.org/~jjore/AI-Prolog-0.740/lib/AI/Prolog.pm>.
- 3-Gerardo Fernández Escribano. Introducción a Xtreme Programming. [En línea] [Citado el: 7 de febrero del 2009.] <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>.
- 4-Ingy dat Net,Audrey Tang. Módulo Perldoc. [En línea] [Citado el: 18 de abril del 2009.]  
<http://search.cpan.org/~ingy/Perldoc-0.20/lib/Perldoc.pm>.
- 5-José Guillermo del Valle. Arquitectura cliente-servidor. [En línea] [Citado el: 29 de enero del 2009.]  
<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>.
- 6-Kent Beck and Cynthia Andres. XP. [En línea] [Citado el: 7 de febrero del 2009.]  
<http://www.amazon.com/reader/0321278658?%5Fencoding=UTF8&ref%5F=sib%5Fdp%5Fpt#reader>.
- 7-Lincoln D. Stein. Módulo CGI. [En línea] [Citado el: 28 de noviembre del 2008.]  
<http://search.cpan.org/~lds/CGI.pm-3.43/CGI.pm>.
- 8-Proyecto Unicornio. Metodología SXP. [En línea] [Citado el: 18 de diciembre del 2008.]  
[http://10.33.2.195/svn/Documentacion\\_General/UNICORNIOS/Tesis/01\\_Tesis\\_Curso\\_2007-08/04\\_Prueba\\_de\\_la\\_propuesta\\_MA-MRV-UR1](http://10.33.2.195/svn/Documentacion_General/UNICORNIOS/Tesis/01_Tesis_Curso_2007-08/04_Prueba_de_la_propuesta_MA-MRV-UR1).
- 9-PhpMyAdmin devel team. Phpmyadmin. [En línea] [Citado el: 15 de diciembre del 2009.]  
[http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php).
- 10-Profesor Alejandro Teruel. Arquitectura en capas. [En línea] [Citado el: 15 de diciembre del 2008.]  
<http://www ldc.usb.ve/~teruel/ci3715/clases/arqCapas.html>.
- 11-Ramiro Lago. Patrones de arquitectura. [En línea] [Citado el: 20 de mayo del 2009.]  
<http://www.proactiva-calidad.com/java/patrones/mvc.html>.
- 12-Rubén Álvarez. Tutorial de SQL. [En línea] [Citado el: 24 de febrero del 2009.]  
<http://www.desarrolloweb.com/manuales/9/>.
- 13-Sam Tregar Módulo Template. [En línea] [Citado el: 27 de abril del 2009.]  
<http://search.cpan.org/~samtregar/HTML-Template-2.9/Template.pm>.

- 14-Sun Microsystems. Mysql Documentation. [En línea] [Citado el: 27 de abril del 2009.]  
<http://dev.mysql.com/doc/>.
- 15-Sun Microsystems. Dbdesigner4. [En línea] [Citado el: 27 de abril del 2009.]  
<http://fabforce.net/dbdesigner4/>.
- 16-Tim Bunce, J. Douglas Dunlop, Jonathan Leffler. Módulo DBI. [En línea] [Citado el: 7 de febrero del 2009.] <http://search.cpan.org/~timb/DBI-1.608/DBI.pm>.
- 17-Uriel Lizama , Marco A. Manzo. Perl Básico, Parte 1. [En línea] [Citado el: 17 de marzo del 2009.]  
[http://perlenespanol.com/tutoriales/aprendiendo\\_perl/perl\\_basico\\_parte\\_1.html](http://perlenespanol.com/tutoriales/aprendiendo_perl/perl_basico_parte_1.html).
- 18-Uriel Lizama, Marco A. Marzo. Tutoriales Perl: Categoría CGI. [En línea] [Citado el: 25 de enero del 2009.] [http://perlenespanol.com/tutoriales/cgi/creando\\_aplicaciones\\_web\\_con\\_c.html](http://perlenespanol.com/tutoriales/cgi/creando_aplicaciones_web_con_c.html).
- 19-Uriel Lizama, Marco A. Manzo. Base de Datos TXT: Parte II. [En línea] [Citado el: 4 de noviembre del 2008.] [http://perlenespanol.com/tutoriales/bases\\_de\\_datos/base\\_de\\_datos\\_txt\\_parte\\_ii.html](http://perlenespanol.com/tutoriales/bases_de_datos/base_de_datos_txt_parte_ii.html).
- 20-Uriel Lizama,, Marco A. Manzo. Base de Datos TXT: Parte I. [En línea] [Citado el: 2 de marzo del 2009.] [http://perlenespanol.com/tutoriales/bases\\_de\\_datos/base\\_de\\_datos\\_txt\\_parte\\_i.html](http://perlenespanol.com/tutoriales/bases_de_datos/base_de_datos_txt_parte_i.html).
- 21-Uriel Lizama, Marco A. Manzo. Tutoriales Perl: Categoría Expresiones Regulares. [En línea] [Citado el: 27 de abril del 2009.]  
[http://perlenespanol.com/tutoriales/expresiones\\_regulares/expresiones\\_regulares\\_intermedio.html](http://perlenespanol.com/tutoriales/expresiones_regulares/expresiones_regulares_intermedio.html).
- 22-Uriel Lizama, Marco A. Manzo. Usar módulos sin instalar. [En línea] [Citado el: 27 de abril del 2009.]  
[http://perlenespanol.com/tutoriales/modulos/usar\\_modulos\\_sin\\_instalar.html](http://perlenespanol.com/tutoriales/modulos/usar_modulos_sin_instalar.html).

# Anexos

## Anexo 1.

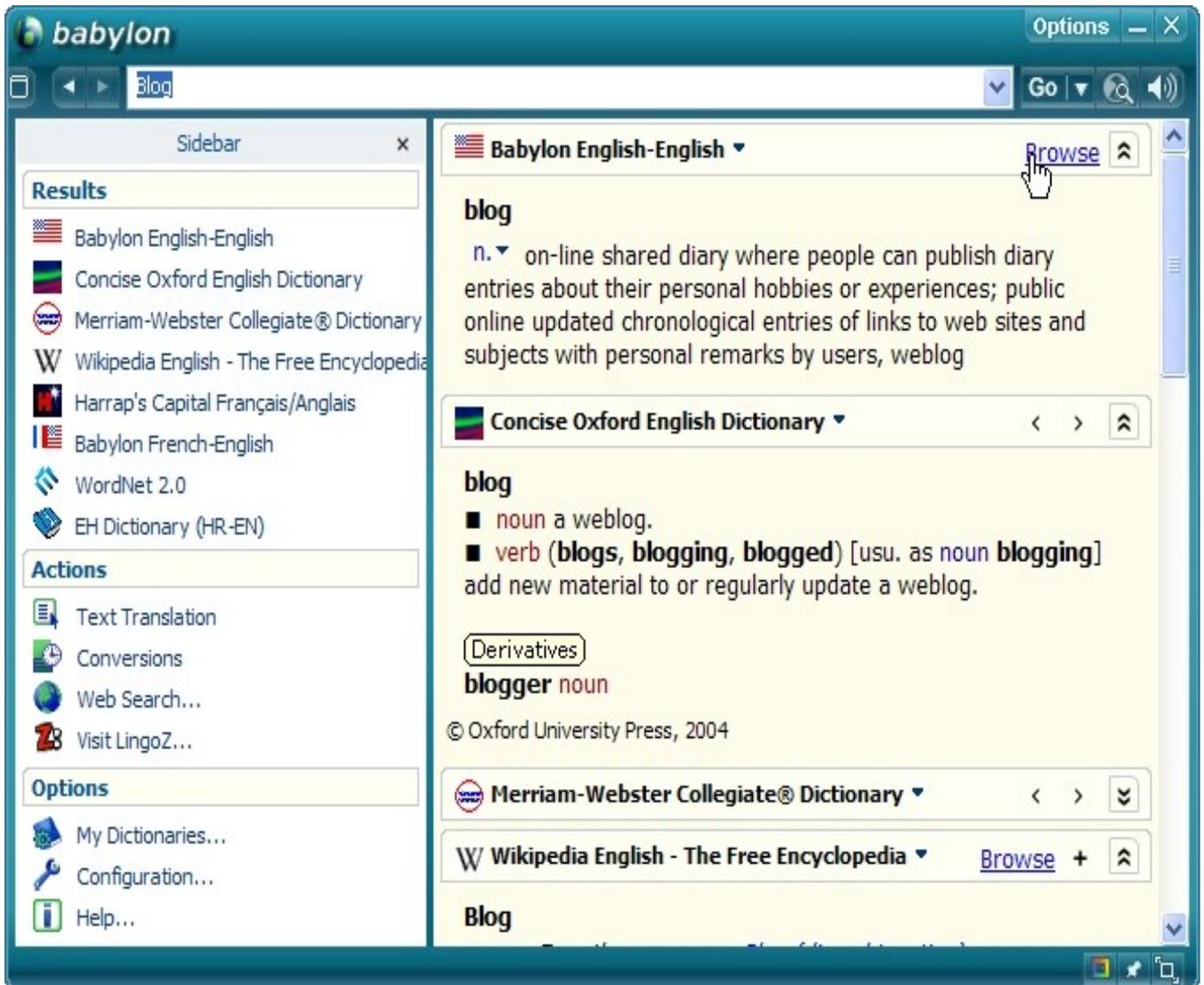


Figura 9: Interfaz de usuario del traductor automático de escritorio "Babylon".

## Anexo 2.



Figura 10: Interfaz de usuario del traductor automático de escritorio "L&H Power Translator".

### Anexo 3.

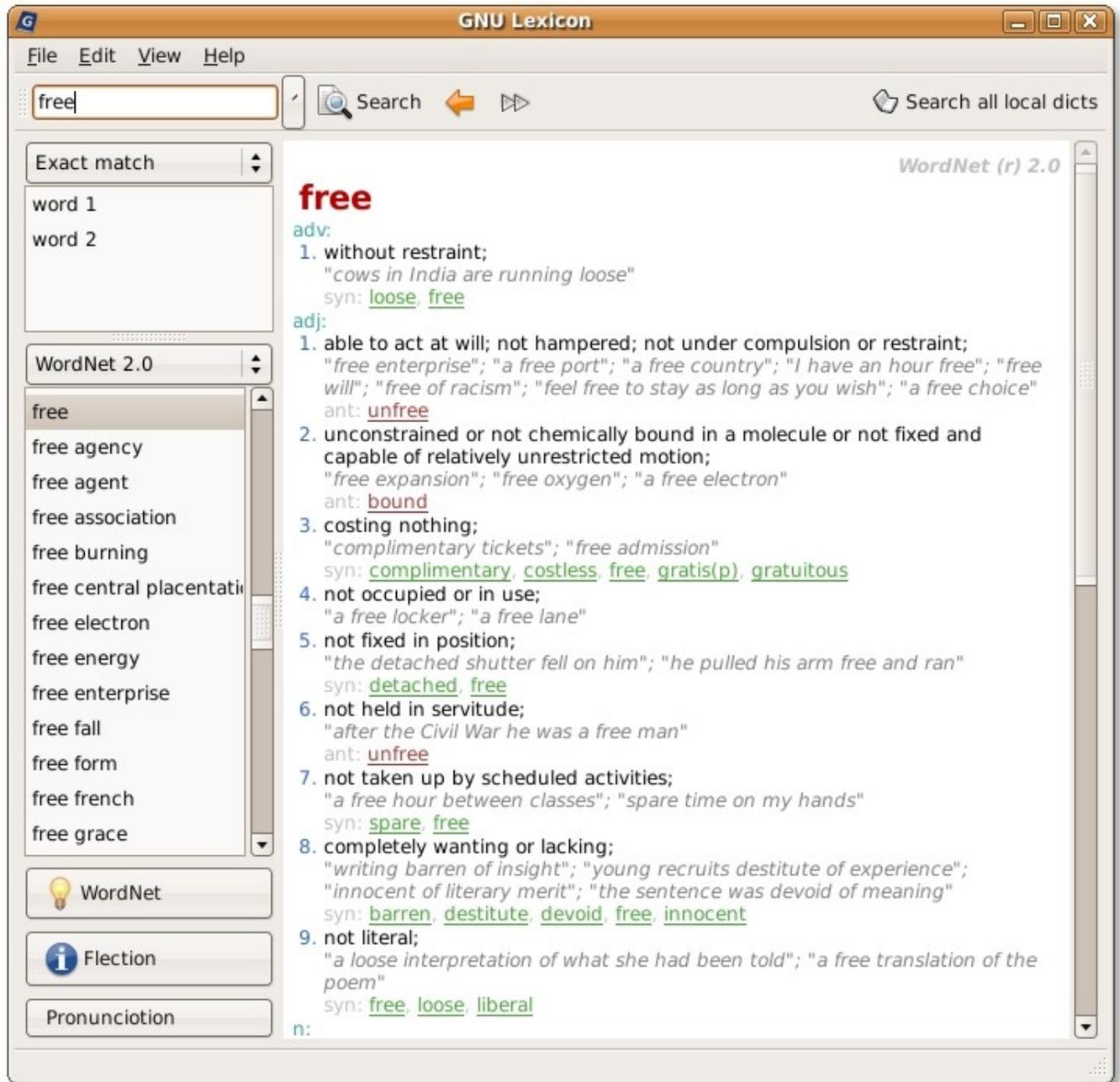


Figura 11: Interfaz de usuario del traductor automático de escritorio "Star-Dict".

Anexo 4.



Figura 12: Interfaz de usuario del traductor web "Google Translate".

Anexo 5.

**Translate a block of text**  ( Enter up to 150 words )

Select from and to languages

---

**Translate a web page** 

Select from and to languages

Figura 13: Interfaz de usuario del traductor web “Yahoo Babel Fish”.

## Anexo 6.

[Ingresar a su cuenta](#)

 Catálogo

Búsqueda en inglés  
Búsqueda avanzada | Buscar por autoridad | Etiquetas del usuario

“mother” devolvió 20 resultados. 

**Refine su búsqueda**

[Select All](#) [Clear All](#) **Add to:**

**Disponibilidad**  
[Limitado a ítems disponibles en la actualidad.](#)

**Bibliotecas**  
[Biblioteca UCI](#)

**Autores**  
[Price, Christine](#)  
[Deller, Sheelagh](#)  
[Deller, Sheelagh0#](#)

**Temas**  
[INGLES](#)  
[ESTUDIO Y ENSEÑANZA](#)  
[LINGUISTICA](#)  
[EDUCACION SUPERIOR](#)  
[CERTIFICACION](#)  
[Mostrar más](#)

1 [Cambridge Certificate of proficiency in english 3 with answers. Examination papers from University of Cambridge ESOL Examinations: English for Speakers of other Languages.](#)  
Publicación: Cambridge University Press 2005 , 161p.  
Fecha:2005  
Disponibilidad: No hay ítems disponibles: Perdido (1),  
Acciones:  [Agregar a la bolsa](#)

2 [Cambridge Certificate of proficiency in english 4 with answers. Examination papers from University of Cambridge ESOL Examinations: English for Speakers of other Languages.](#)  
Publicación: Cambridge University Press 2006 , 192p.  
Fecha:2006  
Disponibilidad: No hay ítems disponibles: Perdido (1),  
Acciones:  [Agregar a la bolsa](#)

3 [Cambridge Certificate of proficiency in english 5 with answers. Examination papers from University of Cambridge ESOL Examinations: English for Speakers of other Languages.](#)  
Publicación: Cambridge University Press 2006 , 192p.  
Fecha:2006  
Disponibilidad: No hay ítems disponibles: Perdido (1),

Figura 14: Prototipo de la Historia de Usuario: “Mostrar en el Koha el resultado de la búsqueda”.

## Anexo 7.



**Sistema de traducción en Koha**  
Su mejor opción

**Bienvenido al SIGB Koha**

Bienvenido a nuestro sitio web. En el podrá insertar en nuestra base de datos todas aquellas palabras que no tengan clasifitacion o que simplemente no se encuentren insertadas en la misma

Espanol:

Ingles:

Ingles:

Figura 15: Prototipo de la Historia de Usuario: “Insertar palabras no clasificadas morfológicamente”.

## Glosario de términos

---

**Homo Sapiens:** Definición científica de la raza humana.

**Traducción:** (del latín *traductiō*, *-ōnis*: hacer pasar de un lugar a otro) es una actividad que consiste en comprender el significado de un texto en un idioma, llamado *texto origen* o "texto de salida", para producir un texto con significado equivalente, en otro idioma, llamado *texto traducido* o "texto meta". El resultado de esta actividad, el *texto traducido*, también se denomina *traducción*.

**Traducción automática:** Es un área de la lingüística computacional que investiga el uso de software para traducir texto o habla de un lenguaje natural a otro.

**Interlingua:** Es una lengua auxiliar internacional basada en vocablos comunes a la mayoría de las lenguas del oeste de Europa y en una gramática anglo-románica simplificada.

**Koha:** Es un *Sistema Integrado de Gestión Bibliotecaria* (SIGB). Que permite automatizar todos los procesos de un Centro de Información y Documentación (CID), como una biblioteca.

**OPAC:** (*Online public access catalog*) es un catálogo automatizado de acceso público en línea de los materiales de una biblioteca. Generalmente, tanto el personal de la biblioteca como el público tienen acceso a él en varias terminales dentro de la biblioteca o desde el hogar vía Internet.

**Corpora-lingüísticos:** Conjunto de textos o fragmentos de textos recopilado para realizar análisis lingüísticos.

**Esperanto:** Es una lengua auxiliar artificial creada por el médico polaco L. L. Zamenhof en 1887 como resultado de una década de trabajo, con la esperanza de que se convirtiera en la lengua auxiliar internacional. Según las estadísticas, ésta es la lengua planificada más hablada del mundo hoy en día.

**Corpus-paralelo:** Recurso lingüístico consistente en textos de dos lenguas (en algún formato electrónico adecuado) que están alineados a cierto nivel de granularidad; generalmente a nivel de párrafo, aunque también a nivel de sección, página o incluso a veces de palabra. Se usa en traducción automática y, principalmente, en memorias de traducción y para el desarrollo de sistemas de traducción estadística.

**Clasificación morfológica:** Clasificación de una determinada palabra atendiendo a que parte de la oración representa, o sea, si es un adjetivo, verbo, pronombre personal, pronombre posesivo, adverbio, conjunción, pronombre interrogativo, pronombre relativo o preposición.

**Inteligencia artificial:** La rama de la ciencia informática dedicada al desarrollo de agentes racionales no vivos. Para explicar la definición anterior, entiéndase a un **agente** como cualquier cosa capaz de percibir

su entorno (recibir entradas), procesar tales percepciones y actuar en su entorno (proporcionar salidas). Y entiéndase a la racionalidad como la característica que posee una elección de ser correcta, más específicamente, de tender a maximizar un resultado esperado.

**Lexicón**: El **lexicón** es el "diccionario" en el que se registran las palabras que conoce un hablante. Este "diccionario" especifica los rasgos característicos de las piezas léxicas (palabras).

**Corpora alineados**: El mismo significado de corpus-paralelos.

**Checkbox**: Elemento gráfico propio del lenguaje HTML; el cual brinda la oportunidad de seleccionar una determinada opción entre varias.

**Semántica**: Se refiere a los aspectos del significado, sentido o interpretación del significado de un determinado elemento, símbolo, palabra, lenguaje o representación formal. En principio cualquier medio de expresión (lenguaje formal o natural) admite una correspondencia entre expresiones de símbolos o palabras y situaciones o conjuntos de cosas que se encuentran en el mundo físico o abstracto que puede ser descrito por dicho medio de expresión.

**Léxico**: Puede significar una lista de palabras; las palabras utilizadas en una región específica, las palabras de un idioma, o incluso de un lenguaje de programación.

**Flooding**: Término en inglés que significa literalmente *inundación*. Se usa en la jerga informática para designar un comportamiento abusivo de la red de comunicaciones, normalmente por la repetición desmesurada de algún mensaje en un corto espacio de tiempo.

**Corpus-monolingüe**: Recurso lingüístico consistente un texto de una misma lengua (en algún formato electrónico adecuado).

**SCRUM**: Es un proceso de desarrollo de software, iterativo y creciente utilizado comúnmente en entornos basado en el desarrollo ágil de software.

**GPL**: La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

**RAM**: Son las siglas de *Random Access Memory*, un tipo de memoria de ordenador a la que se puede acceder aleatoriamente; es decir, se puede acceder a cualquier byte de memoria sin acceder a los bytes precedentes. La memoria RAM es el tipo de memoria más común en ordenadores y otros dispositivos

como impresoras.

**SIGB:** Aplicación informática destinada a automatizar los sistemas y entornos bibliotecarios. Un Sistema Integral de Gestión Bibliotecaria (SIGB) se puede aplicar a las funciones y servicios propios de todo tipo de bibliotecas, tanto públicos como de carácter técnico.