

estudia las acciones del discurso y las situaciones en las cuales el lenguaje es usado.

Lingüística Computacional

Moreno (1998) en su libro "Lingüística computacional. Introducción a los modelos simbólicos, estadísticos y biológicos" la define como una disciplina que trata de dos cosas: lenguas naturales y computadoras. Moreno hace también mención a Lubliner y Grishman (2012) que también hace una definición como: "*estudio de los sistemas de computación utilizados para la comprensión y la generación de las lenguas naturales*".

La lingüística computacional trata esencialmente: lenguas naturales y computadoras. La Asociación de la lingüística computacional (*Association for Computational Linguistics, ACL*) fundada en 1962, ha establecido límites con otras áreas de conocimiento. Se podrían citar numerosos teóricos para llegar a la misma conclusión que el Procesamiento de Lenguaje Natural PNL y la Lingüística computacional LC, tratan del desarrollo de programas de computadoras que simulan la capacidad lingüística humana. Siendo la primera parte de la inteligencia artificial y la segunda de la lingüística informática.

Modelo computacional del lenguaje

La lingüística computacional está enfocada como un proceso comunicativo donde el emisor y receptor procesan determinada información en función de un conocimiento lingüístico y un conocimiento del mundo compartido. En ese enfoque la tarea de la lingüística computacional es reflejar la organización y funcionamiento de las estructuras y procesos lingüísticos por una parte y de las estructuras y procesos cognitivos por otra. Finalmente, toda lengua tiene un modelo matemático previo relacionado. (Moreno, 1998)

Se pueden determinar dos tipos de modelos matemáticos del lenguaje:

- a) **Modelos simbólicos**, conocidos como algebraicos o axiomáticos. Construidos a partir de la teoría de conjuntos y de la lógica matemática, que reflejan la estructura lógica del lenguaje. Uno de los ejemplos más conocidos son las gramáticas generativas, las gramáticas categoriales y las gramáticas de dependencias.
- b) **Modelos probabilísticos**, o estadísticos. Construidos a partir de la Teoría de la Información y la estadística. Estudia la frecuencia de cada fonema, morfema, categoría sintáctica, probabilidad de significado en un determinado contexto.

Ingeniería Lingüística

La Ingeniería Lingüística provee formas de aumentar y optimizar el uso de la lengua, para convertirla en un medio más efectivo. Basado en los vastos conocimientos sobre las lenguas y su funcionamiento recolectados mediante investigaciones. Utiliza recursos lingüísticos como diccionarios y gramáticas electrónicas, y bancos y corpus terminológicos elaborados a lo largo del tiempo. Las investigaciones dicen que necesitamos saber sobre las lenguas y perfeccionan las técnicas necesarias para comprenderlas y manipularlas. Los recursos representan la base de conocimientos necesaria para reconocer, validar, comprender y manipular las lenguas utilizando la potencia de los ordenadores. La aplicación de estos conocimientos lingüísticos nos permite crear nuevas maneras de solucionar problemas que se plantean en los ámbitos

político, social y económico. (Departamento de Ciencias de la Computación, 2017)

Para la mejora de sistemas informáticos, la ingeniería lingüística utiliza los conocimientos de la lengua para perfeccionar nuestra interacción con los sistemas, para asimilar, analizar, seleccionar, utilizar y presentar información de manera más eficaz, creando medios para la generación de lenguaje natural y de traducción. (Vez, 2017) La ingeniería lingüística pretende que una máquina pueda reconocer escritura natural y voces en varias lenguas, para poder traducirla y general resultados orales o impresos.(Departamento de Ciencias de la Computación, 2017).

Técnicas de traducción

Peter Newmark en su libro "*Aproches to Translation*" (Enfoques de la traducción, 1988) habla de la teoría y el arte de la traducción llamando al siglo veinte como la "era de la traducción" ya que fue en los mil novecientos que la traducción de libros en varios idiomas tuvo su mayor auge, dando a la traducción una importancia política y cultural.

Evidentemente una traducción automática por computadora ni siquiera estaba dentro de ese contexto temporal. Lo importante en ese espacio de tiempo era que un texto en un idioma o lengua origen pueda llegar a la mayor cantidad de lectores posibles en un idioma destino. De esta manera, con una traducción correcta o no, el lector podría enterarse de la obra escrita: libro, investigación, estudio, resumen, etc.; y así conseguir mayores ventas o mayor espacio de propagación.

Así surgieron diferentes técnicas utilizadas por traductores para pasar de un idioma a otro y así facilitar la traducción y agilizar la entrega de materiales. Para poder hablar de las técnicas de traducción primero hay que definir lo que es la traducción, y ella consiste en el intento de reemplazar un mensaje en un idioma con el mismo mensaje en otro idioma. La traducción deriva de la lingüística comparada y su aspecto más importante es la semántica, de la cual derivan todas las preguntas sobre la teoría de la traducción. Los sociolingüistas quienes estudian los registros sociales del lenguaje y el problema de los idiomas en contacto con ellos mismos y su entorno nos dirige hacia una teoría, que el lenguaje no puede ser apartado de su contexto. (Alvarenga, 2006)

La traducción lleva consigo una serie de inconvenientes donde el primero de todos es que del paso de un idioma a otro puede haber pérdidas de significado debido a factores culturales que para subsanarlos se aumentan los detalles o la generalización, concluyendo que la traducción de idiomas puede ser solamente aproximada. En segundo lugar, si se toman en cuenta ambos idiomas con sus características básicas nos encontraremos con las diferencias de contexto en cuanto al léxico, la gramática, los sonidos y conceptos intelectuales diferentes, por eso se dice que cuanto más cercanas son las culturas y el lenguaje; existe mayor aproximación en la traducción. En tercer lugar, se encuentra el uso individual de los idiomas, ya que cada escritor o persona tiene su propio estilo personal de utilizar las palabras que el traductor debe respetar para que no pierda el sentido. Finalmente, el traductor tiene diferentes teorías de significado y diferentes valores para la interpretación de un idioma. (Newmark, 1988)

Para definir las técnicas de traducción primero hay que analizar la mayor cantidad de significados que puede tener una frase. Para el ejemplo se utiliza una frase en guaraní, y los aspectos estudiados corresponden a un estudio de traducción de lengua francesa a inglesa (Newmark, 1988)

Extracto del texto: Che angirũ ohetũ va'ekue chupe mbo'eha kotýpe. (Frase original en francés: *Mon ami l'a embrassée dans le hall de l'hôtel*. Inglés: *My friend kissed her in the hotel hall*, se modificó al contexto guaraní).

Traducción lingüística: Mi amigo la besó en el salón del colegio.

Para entender el significado faltan muchos detalles como "el hombre que la ama y quien ella ama" para saber si el hecho de que una mujer sea besada por un hombre en un espacio público donde las personas que están alrededor pueden verlos resulta embarazoso o no para ella.

Referencia: José besó a Verónica en el salón del colegio San Luis Guanella a las 3 pm, el 5 de junio de 2014. Salón se traduce como mbo'eha kotýpe para evitar ambigüedades del lugar donde ocurrió el hecho.

Traducción Automática

Traducción Automática proviene del Inglés *Machine Translation* (MT) estudia diversos métodos para cambiar un texto de un lenguaje origen (LO) a otro texto semánticamente equivalente en una lengua destino (LD), sin intervención humana posible. Esta tarea compleja que pertenece al área del Procesamiento de Lenguaje Natural (PLN) enlaza analizar el texto origen, revisando ambigüedades a niveles léxico - estructural - funcional - pragmático, y generar un texto en idioma destino. (Tertoolen, 2012)

Traducir lenguajes naturales de manera mecanizada, fue una idea norteamericana que surgió de comunicaciones informales entre investigadores y se concretó a principio de los años 50. Primeramente, se trabajó en aspectos Lingüísticos complejos que lograban la traducción automatizada de textos sin intervención humana con características similares a las traducciones realizadas por humanos. Este enfoque, llamado *Fully Automatic High Quality* MT, se descartó con el paso del tiempo por la sucesiva carencia de factibilidad de tal tarea. (Altamirano, 2010)

Durante este proceso surgieron algunos métodos como la "traducción directa" (reemplazo de palabras origen por palabras destino, utilizando sólo un diccionario bilingüe), la "traducción por transferencia" (analizando el texto origen, y/o reglas gramaticales de transferencia para pasar de una representación origen a un destino). Esto dio como resultado una cantidad importante de proyectos efectivos y factibles de la TA para algunas tareas específicas como, reportes del tiempo, la traducción de resúmenes escritos en lenguajes controlados, o la traducción de textos técnicos, por ejemplo, matemáticos (Altamirano, 2010).

En los años 80s surge otro enfoque de TA "basado en ejemplos", cuya traducción se realiza comparando el texto dado con una colección de palabras paralelas. En los años 90s, se propone utilizar métodos estadísticos en el proceso de TA, utilizados ya en el área de la teoría de la información, que se conoce como "traducción estadística" y que constituye actualmente el paradigma de investigación más estudiado. (Altamirano, 2010)

Herramientas de desarrollo de aplicaciones web.

Los aspectos más difíciles de resolver en el momento de comenzar una aplicación web es elegir el tipo de tecnología de información a usar ya que tiene que ser compatible con la que está al servicio de la organización que utilizará el sistema. Esta dificultad reside en la variedad de tecnologías y en la rápida evolución de éstas. Además, hay que tener en cuenta la actividad que realiza la organización y el tipo de sistema que necesita. Estos sistemas pueden ser:

Sistemas de información para la gestión (MIS), Sistema de información de oficina (OIS), Sistemas de apoyo a la toma de decisiones (DSS), Sistemas basados en el conocimiento (KBS), Sistemas interorganizacionales (IOS), Infraestructuras de redes (WAN y LAN). (Secretaría de Planificación Estratégica, Ministerio de Educación de la República del Perú. 2009).

Las herramientas mencionadas a continuación son estándares que corresponden al desarrollo de aplicaciones web utilizadas por la mayoría de los desarrolladores.

Motor de Base de Datos MySQL

Una de las bases de datos con código abierto y de libre distribución más utilizada en el mundo es MySQL. Se pueden crear bases de datos escalables y de alto rendimiento. Por tratarse de la base del funcionamiento del sistema Linux, MySQL se distribuye fundamentalmente para Linux, aunque también hay versiones para Windows. Las versiones de MySQL son cuatro:

- a) Estándar: motor estándar y la posibilidad de usar bases de datos InnoDB, sin soporte completo para utilizar transacciones.
- b) Max: herramientas de prueba para realizar opciones avanzadas de base de datos
- c) Pro: versión comercial del MySQL estándar
- d) Classic: igual que la estándar pero no dispone de soporte para InnoDB

El uso de MySQL (excepto en la versión Pro) está sujeto a licencia *GNU public license* (llamada GPL). Esta licencia admite el uso de MySQL para crear cualquier tipo de aplicación. Se pueden distribuir copias de los archivos de MySQL, salvo esas copias se cobren a un tercer usuario. Se prohíbe cobrar por incluir MySQL. Se puede modificar el código fuente de MySQL, pero si se distribuye la aplicación con el código modificado, habrá que obtener una copia comercial y consultar sobre el cobro de la licencia. Al distribuir copias, se tiene que poder obtener información sobre las licencias GNU. Se puede también obtener una licencia comercial que permitiría cobrar las instalaciones MySQL, incluir la base de datos en ordenadores y cobrar por ello, y otras situaciones no reflejadas en la licencia GNU. (Developer Zone, 2010)

La siguiente lista describe algunas de las características más importantes del software de base de datos MySQL.

- Interioridades y portabilidad
- Escrito en C y en C++
- Probado con un amplio rango de compiladores diferentes
- Funciona en diferentes plataformas.
- Usa GNU *Automake*, *Autoconf*, y *Libtool* para portabilidad.
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Uso completo de *multi-threaded* mediante *threads* del kernel. Pueden usarse fácilmente *multiple CPUs* si están disponibles.
- Proporciona sistemas de almacenamiento transaccionales y no transaccionales.
- Usa tablas en disco *B-tree (MyISAM)* muy rápidas con compresión de índice.
- Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia. Un sistema de reserva de memoria muy rápido basado en *threads*.
- *Joins* muy rápidos usando un *multi-join* de un paso optimizado.

- Tablas hash en memoria, que son usadas como tablas temporales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.
- Tipos de columnas: Diversos tipos de columnas: enteros con/sin signo de 1, 2, 3, 4, y 8 bytes de longitud, *FLOAT*, *DOUBLE*, *CHAR*, *VARCHAR*, *TEXT*, *BLOB*, *DATE*, *TIME*, *DATETIME*, *TIMESTAMP*, *YEAR*, *SET*, *ENUM*, y tipos espaciales *Open GIS*.
- Registros de longitud fija y longitud variable.
- Sentencias y funciones: Soporte completo para operadores y funciones en las cláusulas de consultas *SELECT* y *WHERE*. Por ejemplo: `mysql>SELECT CONCAT (first_name, ' ', last_name) FROM citizen WHERE income/dependents > 10000 AND age > 30;`
- Soporte completo para las cláusulas SQL *GROUP BY* y *ORDER BY*. Soporte de funciones de agrupación (*COUNT ()*, *COUNT(DISTINCT ...)*, *AVG()*, *STD()*, *SUM()*, *MAX()*, *MIN()*, y *GROUP_CONCAT()*).
- Soporte para *LEFT OUTER JOIN* y *RIGHT OUTER JOIN* cumpliendo estándares de sintaxis SQL y ODBC¹.
- Soporte para alias en tablas y columnas como lo requiere el estándar SQL.
- *DELETE*, *INSERT*, *REPLACE*, y *UPDATE* devuelven el número de filas que han cambiado (han sido afectadas). Es posible devolver el número de filas que serían afectadas usando una bandera al conectar con el servidor.
- El comando específico de MySQL *SHOW* puede usarse para obtener información acerca de la base de datos, el motor de base de datos, tablas e índices. El comando *EXPLAIN* puede usarse para determinar cómo el optimizador resuelve una consulta.
- Los nombres de funciones no colisionan con los nombres de tabla o columna. Por ejemplo, *ABS* es un nombre válido de columna. La única restricción es que, para una llamada a una función, no se permiten espacios entre el nombre de función y el '(' a continuación.
- Puede mezclar tablas de distintas bases de datos en la misma consulta (como en MySQL 3.22).
- Seguridad: Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está cifrado cuando se conecta con un servidor.
- Escalabilidad y límites: Soporte a grandes bases de datos. Se usa MySQL Server con bases de datos que contienen 50 millones de registros. También conocemos a usuarios que usan MySQL Server con 60.000 tablas y cerca de 5.000.000.000.000 de registros.

¹Connector / ODBC es un controlador de base de datos estandarizado para plataformas Windows, Linux, Mac OS X y Unix. (MySQL.com, 2018)

- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2). Un índice puede usar prefijos de una columna para los tipos de columna *CHAR*, *VARCHAR*, *BLOB*, o *TEXT*.
- Conectividad: Los clientes pueden conectar con el servidor MySQL usando sockets *TCP/IP* en cualquier plataforma. En sistemas Windows de la familia NT (NT, 2000, XP, o 2003), los clientes pueden usar named pipes para la conexión. En sistemas Unix, los clientes pueden conectar usando ficheros *socket* Unix.
- En MySQL 5.0, los servidores Windows soportan conexiones con memoria compartida si se inicializan con la opción *--shared-memory*. Los clientes pueden conectar a través de memoria compartida usando la opción *--protocol=memory*.
- La interfaz para el conector *ODBC (MyODBC)* proporciona a MySQL soporte para programas clientes que usen conexiones *ODBC (Open Database Connectivity)*. Por ejemplo, puede usar *MSAccess* para conectar al servidor MySQL. Los clientes pueden ejecutarse en Windows o Unix. El código fuente de *MyODBC* está disponible. Todas las funciones para *ODBC 2.5* están soportadas, así como muchas otras. Consulte Sección 25.1, "MySQL Connector/ODBC".
- La interfaz para el conector *J MySQL* proporciona soporte para clientes Java que usen conexiones *JDBC*. Estos clientes pueden ejecutarse en Windows o Unix. El código fuente para el conector *J* está disponible. Consulte Sección 25.4, "MySQL Connector/J".
- Localización: El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas.
- Soporte completo para distintos conjuntos de caracteres, incluyendo *latin1 (ISO-8859-1)*, *german*, *big5*, *ujis*, y más. Por ejemplo, los caracteres escandinavos 'à', 'ä' y 'ö' están permitidos en nombres de tablas y columnas. El soporte para *Unicode* está disponible
- Todos los datos se guardan en el conjunto de caracteres elegido. Todas las comparaciones para columnas normales de cadenas de caracteres son *case-insensitive*.
- La ordenación se realiza acorde al conjunto de caracteres elegido (usando colación Sueca por defecto). Es posible cambiarla cuando arranca el servidor MySQL. Para ver un ejemplo de ordenación muy avanzada, consulte el código Checo de ordenación. MySQL Server soporta diferentes conjuntos de caracteres que deben ser especificados en tiempo de compilación y de ejecución.
- Clientes y herramientas: MySQL server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas. Estos comandos están disponibles a través de la línea de comandos y el cliente *mysqlcheck*. MySQL también incluye *myisamchk*, una utilidad de línea de comandos muy rápida para efectuar estas operaciones en tablas *MyISAM*.
- Todos los programas MySQL pueden invocarse con las opciones *--help* o *-?* para obtener asistencia en línea (Developer Zone, 2010)

InnoDB

En MySQL 5.0, usando el motor de almacenamiento MyISAM, el máximo tamaño de las tablas es de 65536 *terabytes* ($256^7 - 1 \text{ bytes}$). Por lo tanto, el tamaño efectivo máximo para las bases de datos en MySQL usualmente los determina los límites de tamaño de ficheros del sistema operativo, y no por límites internos de MySQL.

El motor de almacenamiento InnoDB mantiene las tablas en un espacio que puede ser creado a partir de varios ficheros. Esto permite que una tabla supere el tamaño máximo individual de un fichero. Este espacio puede incluir particiones de disco, lo que permite tablas extremadamente grandes. El tamaño máximo del espacio de tablas es 64TB. (Oracle, 2018).

En la siguiente tabla se listan algunos ejemplos de límites de tamaño de ficheros de diferentes sistemas operativos.

Tabla 2: Límites de tamaños de ficheros de Sistemas Operativos

Sistema operativo	Tamaño máximo de fichero
Linux 2.2-Intel 32-bit	2GB (LFS: 4GB)
Linux 2.4	(usando sistema de ficheros ext3) 4TB
Solaris 9/10	16TB
Sistema de ficheros NetWare w/NSS	8TB
win32 w/ FAT/FAT32	2GB/4GB
win32 w/ NTFS	2TB (posiblemente mayor)
MacOS X w/ HFS+	2TB

Fuente: Oracle Corporation, 2014

Principales ventajas de InnoDB

Sus operaciones *DML* siguen el modelo *ACID*, con transacciones que ofrece *commit*, *rollback*, y *crash-recovery* con capacidad para proteger los datos del usuario.

El bloqueo a nivel de fila y las lecturas consistentes al estilo de Oracle aumentan la concurrencia y el rendimiento de múltiples usuarios.

Las tablas *InnoDB* organizan sus datos en el disco para optimizar las consultas basadas en claves primarias. Cada tabla *InnoDB* tiene un índice de clave principal denominado índice agrupado que organiza los datos para minimizar la E / S para las búsquedas de clave principal.

Para mantener la integridad de los datos, InnoDB soporta restricciones *FOREIGN KEY*. Las inserciones, las actualizaciones y las eliminaciones se verifican para garantizar que no den como resultado inconsistencias en las diferentes tablas. (Oracle, 2018)

Servidor Web WAMP

Es una plataforma de desarrollo web en Windows para aplicaciones web dinámicas. Utiliza el servidor Apache2, el lenguaje de scripting PHP y una base de datos MySQL. También tiene PHPMyAdmin para administrar sus bases de datos más fácilmente. (WampServer, 2018)

Entorno integrado de Desarrollo ECLIPSE

Una de las plataformas de desarrollo de código abierto basada en Java es Eclipse, consiste en un marco de trabajo y un conjunto de servicios para la construcción del entorno de desarrollo de los componentes de entrada. Tiene un

conjunto de complementos, entre ellas las Herramientas de Desarrollo de Java (JDT).

Eclipse es comúnmente usado como un IDE de Java, también incluye el Entorno de Desarrollo de Complementos (PDE), tienen un entorno de desarrollo integrado y unificado para los usuarios. Aunque Eclipse se escribe en el lenguaje Java, los complementos se encuentran disponibles o planificados para incluir soporte para los lenguajes de programación como C/C++ y COBOL. El marco de trabajo de Eclipse puede también utilizarse como base para otros tipos de aplicaciones que no se relacionen con el desarrollo del software, como los sistemas de gestión de contenido.

El ejemplo principal de una aplicación basada en Eclipse es el entorno de trabajo de IBM® *WebSphere® Studio*, que forma la base de la familia de IBM de las herramientas de desarrollo de Java. *Web Sphere Studio Application Developer*, agrega soporte para JSP, servlets, EJB, XML, servicios web y el acceso a la base de datos. (*Eclipse Foundation*, 2019)

Lenguaje de programación de plataforma web JAVA

Java fue lanzada por la compañía *Sun Microsystems* en 1995 en un principio podía generar aplicaciones para controlar electrodomésticos como lavadoras, heladeras, etc., por su robustez e independencia de la plataforma donde se ejecutase el código, desde el origen fue utilizado para programación de aplicaciones independientes y creación de componentes interactivos (*applets*) en páginas Web. Luego se incorporaron servicios HTTP, servidores de aplicaciones, acceso a bases de datos (JDBC), (Gálvez, 2005)

La descarga de Java es gratuita. *Java Runtime Environment* (JRE) es lo que se descarga como software de Java. El JRE está formado por *Java Virtual Machine* (JVM) que son las clases del núcleo de la plataforma Java, además de bibliotecas. Java se ha adaptado a las necesidades de usuarios y empresas con soluciones y servicios para distintos ámbitos tecnológicos desarrollando ediciones distintas. El archivo Java(.jar) agrupa varios archivos en un solo archivo de almacenamiento. (Oracle, 2018)

En la actualidad Java no es solo un simple lenguaje de programación, sino que un conjunto de tecnologías que abarca a todos los ámbitos de la computación con dos elementos en común:

- El código fuente en lenguaje Java es compilado a código intermedio interpretado por una *Java Virtual Machine* (JVM), por lo que el código ya compilado es independiente de la plataforma.
- Todas las tecnologías comparten un conjunto más o menos amplio de APIs básicas del lenguaje, agrupadas principalmente en los paquetes java.lang y java.io. (Galvez, 2005)

Las características principales que nos ofrece Java respecto a cualquier otro lenguaje de programación serían: simple, orientado a objetos, distribuido, robusto, arquitectura neutral, seguro, portable, interpretado, multihilo, y dinámico.

Lenguaje de Scripts JAVA

Los programas en Java Script se llaman scripts. Se pueden escribir directamente en el HTML de una página web y ejecutarse automáticamente a medida que se carga la página. Los scripts se proporcionan y ejecutan como texto plano. No necesitan preparación especial o compilación para ejecutar. En

este aspecto, Java Script es muy diferente de otro lenguaje llamado Java (Kantor, 2019)

JSP.

Acrónimo de *Java Server Pages*. Con un JSP se pueden crear aplicaciones web que se ejecutan en un servidor Web de múltiples plataformas, y construyen páginas Web. Estas páginas están basadas en datos enviados por el usuario, compuestas de código HTML/XML con etiquetas especiales para programar scripts de servidor en sintaxis JAVA. El motor de los JSP se basa en servlets de java destinados a ejecutarse en el servidor. (Álvarez, 2002).

Los JSP ofrecen una tecnología similar a los servlets para agregar contenido dinámico a un archivo HTML, utiliza código escrito en JAVA por medio de tags especiales que son procesados por el servidor web antes de enviarlos al cliente. Los servlets y jsp se diferencian en que los servlets son clases que deben implementar la clase abstracta *HttpServlet*, en el modo *doGet()* o *doPost()* y deben ser previamente compilados, en cambio los archivos JSP contiene código Java entre el código HTML que deben ser interpretado por el servidor en el momento de la petición por parte del usuario. (Barrios, 2001).

jQuery

jQuery es una biblioteca de JavaScript rápida, pequeña y con numerosas funciones. Hace que la manipulación y recopilación de documentos HTML, el manejo de eventos, la animación y Ajax sean más simples con una API fácil de usar que funciona en una gran cantidad de navegadores. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado la forma en que millones de personas escriben JavaScript. (The jQuery Foundation, 2019).

Hojas de estilo CSS

Hojas de Estilo en Cascada (del inglés *Cascading Style Sheets*) o CSS es el lenguaje utilizado para describir la presentación de documentos HTML o XML, esto incluye varios lenguajes basados en XML como son XHTML o SVG. CSS describe como debe ser renderizado el elemento estructurado en pantalla, en papel, hablado o en otros medios.

CSS es uno de los lenguajes base de la *Open Web* y posee una especificación estandarizada por parte del W3C. Desarrollado en niveles, CSS1 es ahora obsoleto, CSS2.1 es una recomendación y CSS3, ahora dividido en módulos más pequeños, está progresando en camino al estándar. (MDNWebdocs, 2018).

Bootstrap

Bootstrap es un conjunto de librerías de CSS, que facilita y estandariza el desarrollo de sitios web. A partir de la versión 3.x ha sido implementado pensando que se adapte tanto a las pantallas de equipos de escritorio como a móviles y tablets. Bootstrap ha sido desarrollado y es mantenido por la empresa Twitter y ha sido liberado como un producto *Open Source*.

Tiene una filosofía muy intuitiva para el maquetado de sitios web que puede ser rápidamente aprendida por desarrolladores que no vienen del mundo del diseño web. (Bootstrap team, 2019).

El Lenguaje

Existen diversas definiciones sobre lenguaje. La Real Academia Española (RAE, 2018) en su página web publica siete acepciones de la palabra lenguaje cuya etimología proviene de la lengua de oc (Occitano, Lengua Romance de Europa) *Lenguatge*.

Los libros de lengua y literatura Castellana utilizados en Paraguay para la enseñanza de la lengua utilizan en su bibliografía diversos autores modernos y trabajos realizados por docentes en capacitaciones y congresos realizados diversas partes del mundo.

De todas las definiciones encontradas la que reúne en una oración todas las definiciones de los diferentes Literatos y estudiosos de la lengua es:

"El lenguaje es un medio para expresar nuestros pensamientos y según su forma de expresión, puede ser: mímico, oral y escrito." (Natalizia, 1982:14)

El lenguaje adopta variantes entre los habitantes de una región determinada a la que se llaman lengua o idioma y es el código por el cual se realiza el proceso de la comunicación. (Ortellado, 2008)

La comunicación por sobre todo es un proceso, que para que sea efectiva necesita de algunos factores: emisor o escritor - hablante, mensaje que proporciona información, receptor o lector - oyente, código que representa y transmite la información, canal o medio por el cual se transmite la información. (Nasser, 1987)

El contexto situacional también es muy importante en el proceso de la comunicación, estos pueden favorecer o interponerse en la interpretación correcta, se toman en cuenta las circunstancias locales, el tiempo en donde se realiza la comunicación, además de la situación social, cultural y psicológica del emisor y receptor.

Idioma Guaraní

En el origen de la lengua guaraní no se encontraron pruebas de que los Guaraníes tuvieran alguna forma de escritura. El traspaso de la lengua se efectuaba de padres a hijos y hoy en día ya como resultado del mestizaje el idioma sigue siendo enseñado de abuelos a padres y estos enseñan a sus hijos. En la época posterior a la colonia, cuando los religiosos necesitaron escribir para difundir la religión fue que aparecieron los primeros escritos en Guaraní. Al principio se utilizaron las letras del idioma castellano para escribir los sonidos, pero era muy pobre para reflejar la riqueza fonética del Guaraní. Después de varias disputas entre la grafía tradicional popular y la académica, comenzó a enseñarse el guaraní con el afán de que todos pudieran hablarlo correctamente. (De Guaranía, 2004)

Fonética y Fonología Guaraní

El autor Pedro Moliniers (1986) en su libro Lecciones de Guaraní, delimita al alfabeto guaraní con 33 signos que a su vez representan la misma cantidad de sonidos.

Tabla 3: Fonemas de la Lengua Guaraní según Moliniers

Fonemas de la Lengua Guaraní
a, ã, ch, e, ê, g, ñ, h, i, î, j, k, l, m, mb, n, nd, ng, nt, ñ, o, õ, p, r, rr, s, t, u, ù, v, y, ÿ, '.

Fuente: Moliniers, 1986. Lecciones de Lengua Guaraní.

A diferencia de los Autores Natalia Krivoshein de Canese y Feliciano Acosta Alcaraz, en su obra Gramática Guaraní Colección Ñemity (2001) hablan como el guaraní tiene alrededor de 30 fonemas. En los hispanismos a menudo se usan los fonemas d, ll, mp, y rr.

Tabla 4: Fonemas de la Lengua Guaraní según la Gramática Guaraní

Fonemas de la Lengua Guaraní
a, ã, ch, e, ê, g, h, i, î, j, k, l, m, mb, n, nd, ng, ñ, o, õ, p, s, t, u, ù, v, y, ÿ, '

Fuente: Krivoshein N. 2001

Luego la secretaria de Políticas Lingüísticas (SPL) dependiente de la presidencia de la República del Paraguay con la Resolución N° 80 del 19 de junio de 2012 estableció el alfabeto oficial provisional compuesto de 33 fonemas principales del idioma y son como sigue: a, ã, ch, e, ê, g, g, h, i, î, j, k, l, m, mb, n, nd, ng, nt, ñ, o, õ, p, r, rr, s, t, u, ù, v, y, ÿ.

A partir de la creación de la SPL en el año 2012 comenzó el proceso de formalización del idioma Guaraní en cada departamento, así como en las diferentes parcialidades indígenas que mantienen la lengua hasta hoy. (Secretaria de Políticas Lingüísticas, 2018).

MATERIALES Y METODOS

El proyecto se realizó en Asunción, Paraguay, desde agosto de 2014 a abril 2019, sobre la construcción de analizadores léxicos, semánticos y sintácticos de la lengua guaraní, consultando al Ateneo de Lengua y Cultura Guaraní, basados en publicaciones de la Secretaria de Políticas Lingüísticas de Paraguay (SPL) y entrevistando a miembros de la Academia de la Lengua Guaraní y otros profesionales de la lengua. El trabajo consistió en desarrollar específicamente una aplicación web de análisis y traducción automática guaraní - español / español guaraní utilizando el servidor WAMPSEVER (Wamp Server. 2018), lenguaje de programación JAVA (Kantor I. 2019) y la base de datos MySQL 5.0, (2010); (Oracle, 2018).

Una vez terminada la recolección de datos se comenzó con la selección de herramientas disponibles para el desarrollo de la aplicación web. Luego del desarrollo se realizaron las pruebas y depuración de errores.

El analizador léxico utilizó un algoritmo de selección de palabras dentro de la base de datos. El analizador sintáctico utilizó un algoritmo de división por puntos, comas y espacios y luego de la actuación del analizador léxico, utilizó un árbol generacional de frases que ordena las palabras de acuerdo a la mayor estadística de coincidencias. El analizador semántico devuelve el equivalente correspondiente, que necesita el ingreso de una palabra o frase para devolver, en el primer caso, todos los resultados posibles de su significado, con los atributos correspondientes, en el segundo caso, la frase resultante que abarque la mayor parte del texto ingresado. Los módulos necesarios para la aplicación web se desarrollaron en lenguaje de programación Java, mediante la plataforma de desarrollo Eclipse. La implementación del esquema de traducción dirigida se realizó mediante consultas a la base de datos dentro de algoritmos de división de entradas y comparación de resultados de las consultas. La base de datos utilizada para las consultas de palabras y frases es MySql (ORACLE, 2018). Se desarrolló una interfaz web sencilla adaptable a todo tipo de dispositivos.

RESULTADOS

De acuerdo con el objetivo específico en el cual se propuso implementar un esquema de traducción se detalla en el apartado IV.2. En la sección IV.4 y IV.5 se describe el objetivo de agregar una base de datos con un diccionario. El diseño y elaboración de los módulos necesarios para el analizador léxico, sintáctico y semántico de la Aplicación web propuesta se detallan en los apartados IV.2 al IV.5. Finalmente, el objetivo de integrar una interfaz sencilla al traductor se encuentra en el apartado IV.6.

Resultados del relevamiento

Las entrevistas revelaron una problemática nacional que con la Academia de Lengua Guaraní intentan subsanar el poco uso escrito que se le da al Guaraní en todas las Instituciones públicas. La Resolución N°103 de la Secretaría de Políticas Lingüísticas del 30 de julio de 2018, por la cual se dispone el plazo de tres años para la exigibilidad en la implementación de las obligaciones derivadas de la ley 4251/2010 de lenguas que requieran una expresión escrita, obliga el uso del Guaraní en todos los Poderes del Estado. Dada esta situación, Secretaría de Políticas Lingüísticas publica la Gramática del Guaraní el 13 de agosto de 2018. Material utilizado para verificar los ejemplos de frases utilizadas en la base de datos. Se realizaron entrevistas abiertas a diferentes profesionales de la lengua. La mayoría docentes de Lengua y Literatura Guaraní Castellano.

Funcionamiento del sistema

El usuario ingresa una palabra o frase. Comprueba si se ingresó una palabra o una frase. Si ingresó palabra, busca si la palabra existe en la base de datos en guaraní, sino busca en la base de datos del español. Si existen algunas de las dos, agrupa los resultados de acuerdo con los atributos. En caso de que sea una frase, separa en grupos por puntos y almacena en el vector punto. Luego separa por comas y almacena en el vector coma. Finalmente separa por espacios y almacena el resultado en el vector espacio, las palabras del vector espacio se concatenan con los espacios correspondientes mientras se comprueba si existe la frase en la base de datos. En caso de que exista se almacena en una lista. Luego se concatenan los resultados del vector coma, a medida que se verifica en la base de datos si existe. En caso de que existe se almacena en la misma lista. Luego se recorre el vector punto, concatenando los resultados y verificando en la base de datos, si existe, se almacena en la misma lista. Este proceso se realiza con un recorrido por derecha y por izquierda. Luego de este proceso, se recorre la lista resultante y se comprueba cuáles son las frases con mayor coincidencia en el texto original. Y se eliminan las de menor coincidencia. Luego, se verifican los resultados obtenidos con el texto original, las frases o palabras no reconocidas, se buscan su equivalente en la base de datos, los datos obtenidos se almacenan en una lista, finalmente se concatenan todos los resultados y se muestra en pantalla.

Gramática generativa transformacional de Chomsky aplicada al Guaraní

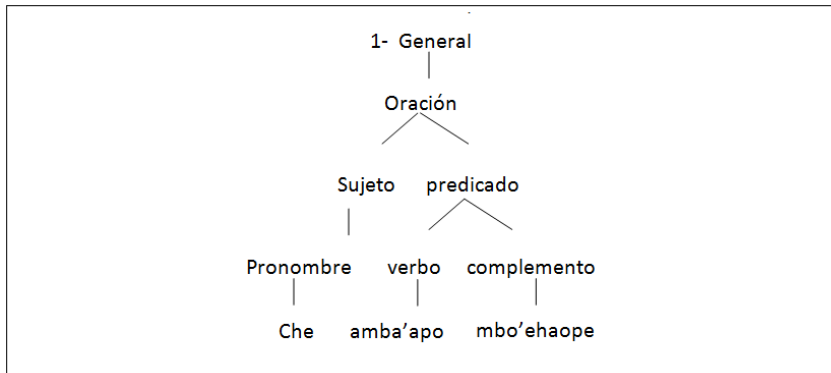
Con la revisión bibliográfica y las observaciones se pudo llegar a una muestra de gramática básica del guaraní utilizando la Gramática Generativa Transformacional de Chomsky, para representar el desplazamiento de constituyentes del lenguaje natural. (Chomsky, 1957).

General	→	oración
oración	→	oración sujeto predicado sujeto predicado oración , oración oración. oración oración Conjunción oración Conjunción oración
Sujeto	→	Sustantivo Pronombre sustantivo Adjetivo sustantivo Sustantivo adjetivo Nombre Pronombre
Predicado	→	Verbo Verbo complemento

Fuente: Elaboración propia

Figura 1: Gramática de la Lengua Guaraní v1.1

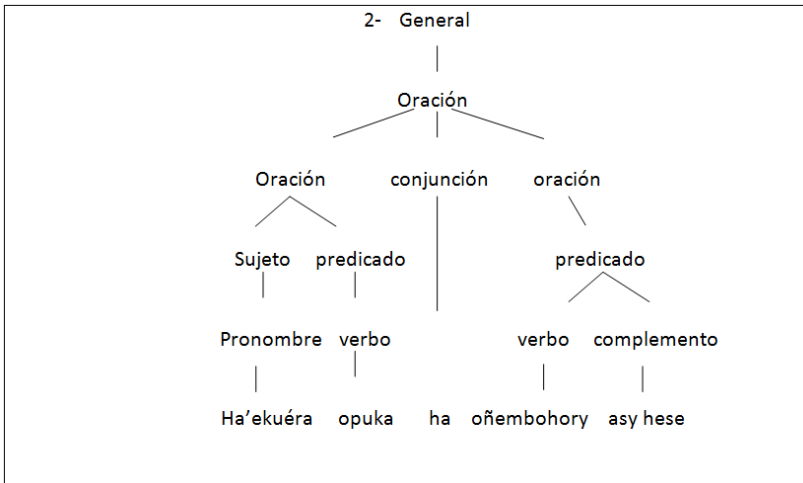
Para comprobar el desplazamiento de la gramática se utilizaron seis ejemplos preliminares, utilizando la técnica de derivación de izquierda a derecha (Suárez, 2007):



Fuente: Elaboración propia

Figura 2: Ejemplo de Gramática 1

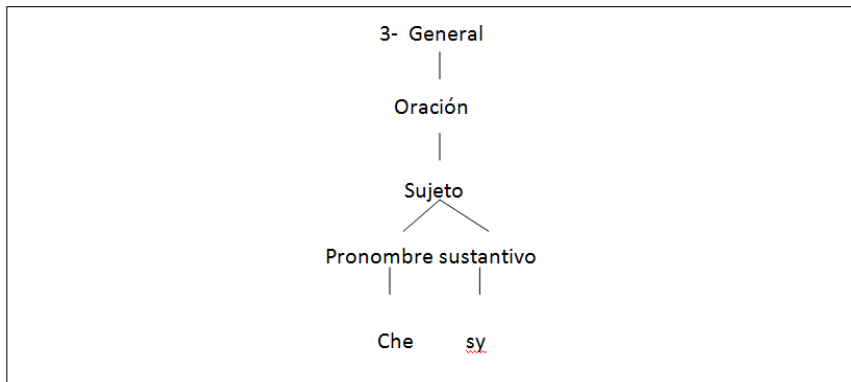
En la Figura 1 se muestra el primer ejemplo de Gramática, donde toda frase se forma a partir de una idea general, general deriva en oración, oración deriva en sujeto y predicado, el sujeto deriva en pronombre, y el predicado deriva en verbo y complemento, siendo esta la última derivación.



Fuente: Elaboración propia

Figura 3: Ejemplo de Gramática 2

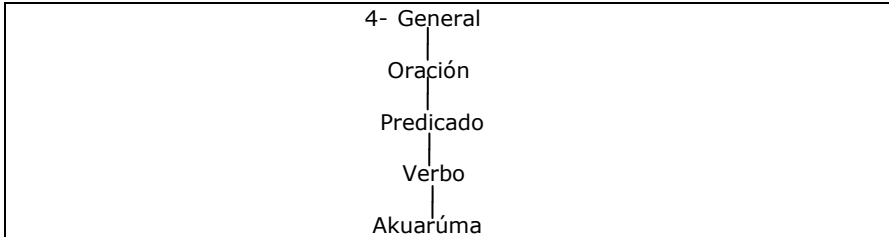
En la Figura 3 se introduce la primera recursividad donde oración deriva en sí misma, además de una nueva forma en la tercera derivación, con la introducción de una conjunción uniendo dos ideas, quedando oración, conjunción, oración; el primer término consecuente oración vuelve a derivar en sujeto y predicado, y el último consecuente oración deriva en predicado.



Fuente: Elaboración propia

Figura 4: Ejemplo de Gramática 3

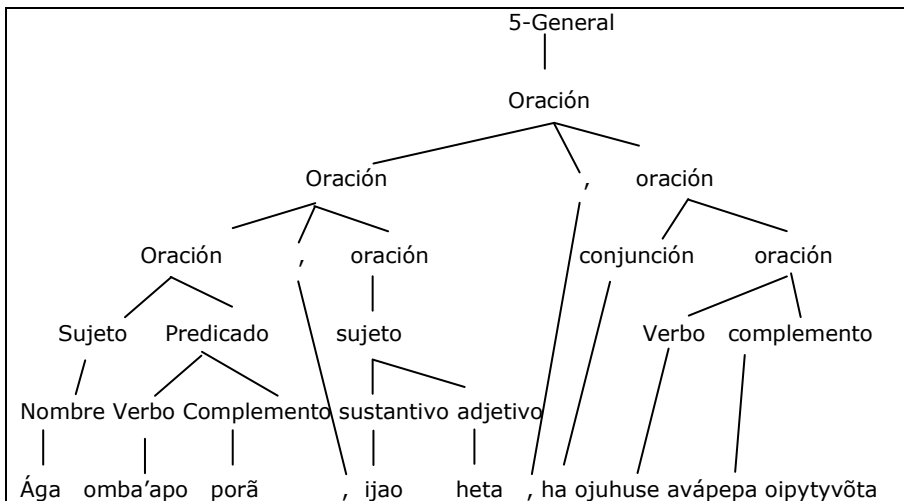
En la Figura 4, se observa la construcción de una oración unimembre de donde se construye solo el sujeto.



Fuente: Elaboración propia

Figura 51: Ejemplo de Gramática 4

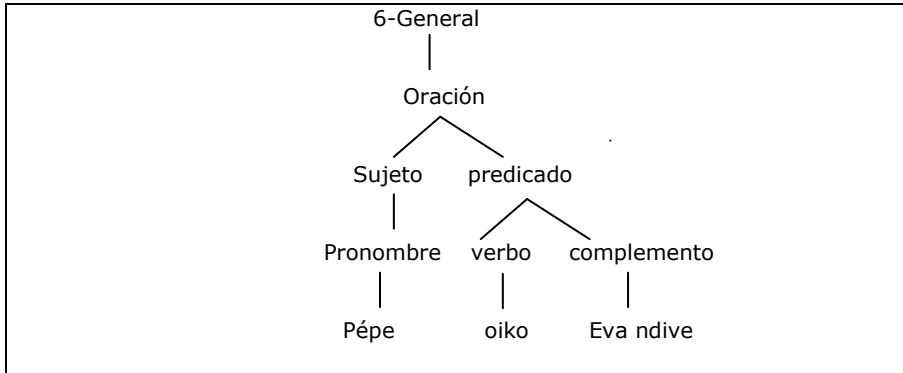
El ejemplo 4, Figura 5, construye otra oración unimembre diferente constituida solo por el verbo.



Fuente: Elaboración propia

Figura 6: Ejemplo de Gramática 5

En la Figura 6, similar al ejemplo 2, en la cuarta derivación se puede ver un nuevo caso de recursividad, donde el primer consecuente de oración, oración deriva en oración, oración.



Fuente: Elaboración propia

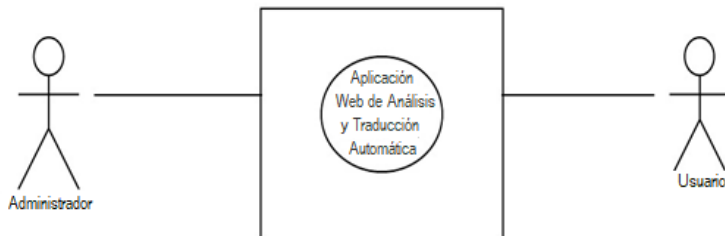
Figura 7: Ejemplo de Gramática 6

En la Figura 7, se puede observar un ejemplo de complemento circunstancial de lugar.

Modelado del Sistema: Casos de Uso Niveles

Caso de Uso: Nivel Cero

Presenta la interacción entre el sistema y los actores. Representado en la Figura 8.

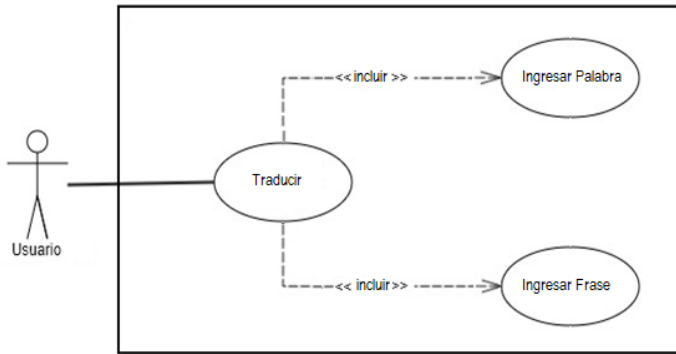


Fuente: Elaboración propia.

Figura 8: Caso de Uso Nivel 0 - Usuarios de la Aplicación Web de Análisis y Traducción Automática.

Caso de Uso Nivel 1 – Caso de Uso Rol del Usuario

El Usuario puede Traducir al introducir palabras o frases.



Fuente: Elaboración propia.

Figura 9: Caso de Uso Nivel 1 - Rol del usuario.

Tabla 5: Descripción del Caso de Uso Traducir

Nombre	Traducir – Ingresar Palabra	
Actor	Usuario	
Precondición	El actor debe acceder a la página web.	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el actor quiere traducir	
Secuencia Normal	Paso	Acción
	1	El actor escribe una palabra o frase
	2	El traductor busca en la base de datos
Post condición	El traductor muestra la traducción de la palabra o frase.	
Excepciones	Paso	Acción
	1	El traductor informa que no se encuentra ningún resultado.

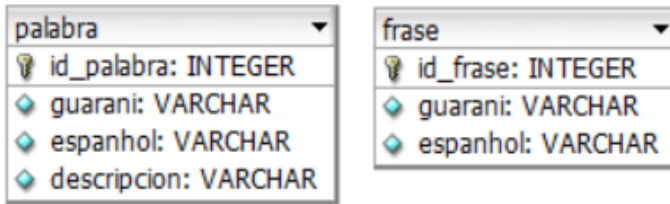
Fuente: Elaboración propia

Modelado de Datos

Tabla 6: Entidades que se representan en el DER

Nombre	Descripción
Palabra	Contiene información de las palabras registradas en el sistema.
Frase	Contienen las frases registradas en el sistema.

Fuente: Elaboración Propia.



Fuente: Elaboración propia

Figura 10: Tablas palabra, frase

Diccionario de datos

Tabla 7: Tabla palabra

Nombre Tabla: palabra **Nro.:** 1

Descripción Tabla: Contiene información de la descripción de las palabras.

Campo	Tipo	Descripción
id_palabra	integer (10)	Clave primaria identificadora de la fila
Español	varchar(200)	Palabra en español
Guarani	varchar(200)	Palabra en guaraní
Descripción	varchar (200)	Descripción gramatical de la palabra

Clave: PRIMARY

Fuente: Elaboración Propia.

Tabla 8: Tabla frase

Nombre Tabla: frase **Nro.:** 2

Descripción Tabla: Contiene información de las frases.

Campo	Tipo	Descripción
id_frase	integer (10)	Clave primaria identificadora del registro.
Español	varchar(200)	Frase en español
Guarani	varchar(200)	Frase en guaraní

Clave: PRIMARY

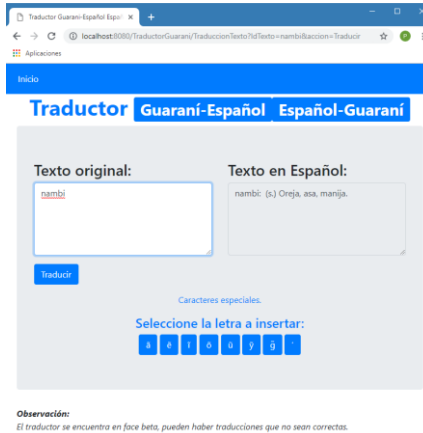
Fuente: Elaboración Propia.

Presentación de Pantallas

La página principal del traductor contiene las opciones: Inicio, texto original, caracteres especiales, traducir.

Inicio

El traductor recarga la página con los campos limpios.



Fuente: Elaboración propia

Figura 11: Presentación de pantalla **Texto original**

El usuario puede ingresar un texto original que puede contener una palabra, una frase, una oración, varias oraciones unidas por comas o puntos. Si la palabra o frase es reconocida por el traductor, devuelve la palabra o frase traducida. Sino devuelve “No se encontrada ningún resultado”.

Caracteres especiales

El usuario puede agregar los caracteres especiales propios del guaraní: ã, ẽ, õ, õ, õ, õ, õ, õ, '.

Traducir

El usuario al seleccionar el botón traducir el traductor devuelve el resultado de la traducción.

Detalles de la codificación.

El usuario ingresa un texto original, luego el traductor comprueba si se ingresó una palabra o una frase buscando espacios como vemos en la Figura 12.

```
String[] Aux=texto.split(" ");
```

Fuente: Elaboración propia

Figura 12: Codificación 1

Si se ingresa palabra, busca si la palabra existe en la base de datos en guaraní, sino busca en la base de datos del español. Utiliza el mismo procedimiento para una frase. Se puede observar parte del código en la Figura 12.

```

if (Aux.length<=2) {
    lista = INTERFACEEjecucion.getTextoINTERFACE().listarg(texto);
    if (lista.size(>0) {

        text = INTERFACEEjecucion.getTextoINTERFACE().buscargp(texto);
    }else {
        lista = INTERFACEEjecucion.getTextoINTERFACE().listare(texto);
        if (lista.size(>0) {

            text = INTERFACEEjecucion.getTextoINTERFACE().buscarep(texto);
        }else {
            text = INTERFACEEjecucion.getTextoINTERFACE().buscarg(texto);
        }
    }
}
session.setAttribute("objetos", text);
}else {
    texto = " " + texto;
    text = INTERFACEEjecucion.getTextoINTERFACE().traducir(texto);

    if (text.getTipo().contentEquals("fraseGuara")) {
        session.setAttribute("objetos", text);
    }else if (text.getTipo().contentEquals("frase")) {
        textofinal = INTERFACEEjecucion.getTextoINTERFACE().traducire(texto);
        if (textofinal.getTipo().contentEquals("fraseEspa")) {
            session.setAttribute("objetos", textofinal);
        }else if (textofinal.getTipo().contentEquals("frase")) {
            textofinal.setEspanho(null);
            textofinal.setGuarani(texto);
            session.setAttribute("objetos", textofinal);
        }
    }
}
}
}

```

Fuente: Elaboración propia

Figura 13: Codificación 2

Analizador Sintáctico

Verifica si existe alguna palabra o frase, agrupa los resultados de acuerdo con los atributos, sino muestra mensaje "No se encuentra ningún resultado". En caso de que sea una frase, separa en grupos por puntos y almacena en el vector punto. Luego separa por comas y almacena en el vector coma. Finalmente separa por espacios y almacena el resultado en el vector espacio.

Analizador Léxico

Las palabras del vector espacio se concatenan con los espacios correspondientes mientras se comprueba si existe la frase en la base de datos. En caso de que exista se almacena en una lista. Luego se concatenan los resultados del vector coma, a medida que se verifica en la base de datos si existe. En caso de que existe se almacena en la misma lista. Luego se recorre el vector punto, concatenando los resultados y verificando en la base de datos. Si existe, se almacena en la misma lista. Este proceso se realiza con un recorrido por derecha y por izquierda. Figura 13


```

String FraseCompleta = tex;
String[] Aux_punt = FraseCompleta.split("\\. |\\.");
int tamanho, tamanho2, tamanho3;
String TemTexesp = null;
String TemTexcom = null;
String TemTexpun = null;
ArrayList<FraseBD> listagder = new ArrayList<FraseBD>();
FraseBD frase = null;
tamanho= Aux_punt.length;
int li=0;
for (int i=0; i<tamanho ; i++) {

    String[] Aux_com = Aux_punt[i].split(", |,");
    tamanho2=Aux_com.length;
    for (int j=0; j<tamanho2 ; j++) {
        String[] Aux_esp = Aux_com[j].split(" ");
        tamanho3 = Aux_esp.length;
        for (int k=0; k<tamanho3; k++) {
            if (k==0) {
                TemTexesp = Aux_esp[k];
            } else {
                TemTexesp = TemTexesp + " " + Aux_esp[k];
                if (INTERFACEEjecucion.getTextoINTERFACE().buscarfg(TemTexesp).getEspañol() != null) {
                    frase = INTERFACEEjecucion.getTextoINTERFACE().buscarfg(TemTexesp);
                    frase.setZzz(k);
                    frase.setYyy(j);
                    frase.setXxx(i);
                    listagder.add(frase);
                    li++;
                }
            }
        }
    }
}

```

Fuente: Elaboración propia

Figura 14: Codificación 3

Analizador Semántico

Luego del proceso anterior, se recorre la lista resultante y se comprueba cuáles son las frases con mayor coincidencia en el texto original. Y se eliminan las de menor coincidencia (Figura 14).

```

for (int i=0; i<palabraslista.size(); i++) {
    if (i==0) {
        lista.add(palabraslista.get(i));
    }else if (palabraslista.get(i).getPosicion()==palabraslista.get(i-1).getPosicion()) {
        if (palabraslista.get(i).getTamaño()<palabraslista.get(i-1).getTamaño()) {

        }else if (palabraslista.get(i).getTamaño()>palabraslista.get(i-1).getTamaño()) {
            lista.remove(lista.size()-1);
            lista.add(palabraslista.get(i));
        }
    }else if (palabraslista.get(i).getPosicion()> (palabraslista.get(i-1).getPosicion()+palabraslista.get(i-1).getTamaño()-1)){
        lista.add(palabraslista.get(i));
    }
}

```

Fuente: Elaboración propia

Figura 15: Codificación 4

Finalmente se concatenan todos los resultados y se muestra en pantalla.

```

for (int i=0; i<Textocompleto.size(); i++) {
    textofinal += Textocompleto.get(i);
}
TextoBD tb=new TextoBD();
tb.setEspanhol(textofinal);
tb.setGuarani(texto);
if (banderaidioma == 1) {
    tb.setTipo("fraseGuara");
}else {
    tb.setTipo("frase");
}
tb.setId(0);
return tb;

```

Fuente: Elaboración propia

Figura 16: Codificación 5

Técnica de Prueba

Algoritmo de Proceso de traducción

Cuerpo Principal

INICIO

- | Mientras página activa
- | | Ejecutar Proceso inicio
- | | Ejecutar Proceso Principal
- | | Ejecutar Proceso Final
- | Fin Mientras

FIN

Rutinas

P_INICIO

- | El usuario Ingresa la palabra o
- | frase a traducir.

RETORNO

P_PRINCIPAL

- | El traductor realiza el proceso
- | de análisis sintáctico
- | El traductor realiza el proceso de análisis
- | léxico
- | SI existe
- | | El traductor realiza el
- | | proceso de análisis semántico
- | SI NO
- | | Mensaje No se encuentra ningún
- | | resultado
- | FIN SI

RETORNO

P_FINAL

- | Se muestra en pantalla
- | el resultado del análisis
- | semántico

Fin de la tarea

RETORNO

CONCLUSIONES

Cuando se fijó el objetivo de desarrollar una aplicación web de análisis y traducción automática Guaraní – Español y Español – Guaraní se hizo el primer planteamiento de utilizar parte del tratamiento de lenguajes formales, para ello se necesitó conseguir la gramática del español y del guaraní. Conseguir la gramática llevó gran parte del inicio de la investigación, a la par que se estudiaba el funcionamiento de los traductores existentes. Recurriendo a diferentes fuentes, se pudo encontrar los puntos comunes en diferentes textos y de esa manera formular la gramática generativa transformacional aplicada al guaraní, con seis ejemplos iniciales. Para llegar a esa formulación se realizaron numerosas entrevistas, que fueron documentadas de acuerdo al aporte significativo para la investigación.

Luego se investigó en numerosas plataformas de traductores con sistemas expertos o con inteligencia artificial, el cual podían ser útiles si se contaba con un gran corpus léxico y con una gramática bien definida. Al no contar con ninguna de las dos cosas, se abocó a la idea de ir descartando las teorías y las herramientas que no contribuían a la continuidad de la investigación.

Para el desarrollo de la aplicación de acuerdo al objetivo específico de **diseñar un analizador léxico, sintáctico y semántico**, se necesitó elegir adecuadamente las herramientas a utilizar. En el momento en que los módulos estaban en su fase de inicio, fueron surgiendo las necesidades, fue así como los analizadores fueron cambiando de un primer planteamiento de traducir palabra por palabra con el uso de la gramática; a utilizar un método que permita analizar frases y que de acuerdo con la mayor coincidencia devuelva una traducción asertiva.

Luego la aplicación comenzó su etapa de diseño, que también luego de numerosas investigaciones sobre herramientas de desarrollo, se logró **elaborar los módulos necesarios para la aplicación web propuesta**, luego se realizaron los algoritmos de reconocimiento y búsqueda de palabras, y finalmente se eligió una base de datos gratuita.

Fue así como, para **implementar un esquema de traducción dirigida mediante los analizadores léxico, semántico y sintáctico**, además de **agregar una base de datos con un diccionario propuesto e integrar una interfaz sencilla a la aplicación web**, se utilizó WAMP SERVER con la base de datos MySql y la plataforma Eclipse para el lenguaje de programación JAVA.

Para los analizadores se utilizó un algoritmo con enfoque de traducción basada en corpus, implementando traducción automática basada en ejemplos. En el desarrollo de los módulos de análisis se utilizó un vector tridimensional para poder ordenar y concatenar los resultados en una nueva variable de acumulación.

El corpus léxico utilizado en la base de datos fue obtenido de textos sugeridos por la Secretaría de Políticas Lingüísticas (SPL), libros de textos sugeridos por el Ministerio de Educación y Ciencias, glosarios y diccionarios no oficiales impresos o publicados en internet.

Finalmente **se logró el objetivo de realizar la aplicación web** de análisis y traducción que puede traducir en forma automática palabras, frases o textos. Esto contribuirá significativamente al uso del idioma guaraní, además de ser una aplicación web única.

REFERENCIAS BIBLIOGRAFICAS

- Altamirano, I. (2010). *Traducción automática basada en conocimiento lingüístico y simplificación textual*. FaMAF. <https://cs.famaf.unc.edu.ar/~romina/PlanTrabajo.pdf>
- Álvarez, M. (2002). *¿Qué es un JSP? en Manual de JSP*. DesarrolloWeb.com. <https://desarrolloweb.com/manuales/73/>
- Alvarenga, N. (2006). *Translation Techniques for songs from English to Spanish*. Facultad de Lenguas Vivas. Universidad Evangélica del Paraguay. Asunción
- Arnold, D. y John B. (2003). *Why machine translation is difficult for computers*. In Somers, H. L.(coordinator): *Computers and Translation: a translator's guide*. <https://pdfs.semanticscholar.org/0006/02db30b8cd726148e48fae35e6ce644751d6.pdf>
- Artemano, C. y otros. (2006). *Apertium, una plataforma de código abierto para el desarrollo de sistemas de traducción automática*. En: *Computational Processing of the Portuguese Language: 7th Workshop on Computational Processing of Written and Spoken Portuguese, PROPOR 2006. Lecture* <http://www.dlsi.ua.es/~mlf/docum/armentano07p.pdf>
- Barrios, J. (2001). *Java Servlets*. Universidad de Chile <https://users.dcc.uchile.cl/~jbarrios/servlets/jsp.html>
- Benítez, L. G. 1959 *Lecciones de historia americana*. Asunción, Imprenta Comuneros.
- Bootstrap team. (2019). *Bootstrap home*. <https://getbootstrap.com/>
- Chomsky, N. (1957). *Syntactic structures*. La Haya, Muto.
- Congreso de la Nación Paraguaya. 2010. *Ley 4251 de Lenguas*. <http://www.diputados.gov.py/plenaria/101209-SO/pdf101209so/10.pdf>
- Constitución Nacional del Paraguay*. 1992. http://www.diputados.gov.py/ww5/application/files/3015/3312/8867/CN_Reglamento_CodEtica.pdf
- Contreras H. (2001). *Procesamiento del Lenguaje Natural basado en una "gramática de estilos" para el idioma español*. Universidad de los Andes, Facultad de Ingeniería, Postgrado en Computación
- De Guaranía, F. (2004). *Gramática y Literatura Guaraní*. Serie Ateneo de Lengua y Cultura Guaraní.
- Departamento de Ciencias de la Computación. (2017). *Ingeniería Lingüística. ¿Cómo aprovechar la fuerza del lenguaje?* Universidad de Chile. <https://users.dcc.uchile.cl/~abassi/WWW/Lengua/ingenieria.html>
- Developer Zone. Oracle Corporation and/or its affiliates. 2010. *Manual de Referencia*. <http://ftp.tcr.edu.tw/MySQL/doc/refman/5.0/es/features.html>
- Dirección General de Estadísticas, Encuestas y Censos. Revisión (2015). *Paraguay – Proyección de la Población Nacional, Áreas Urbana y Rural por Sexo y Edad, 2000-2025*. <http://www.dgeec.gov.py/Publicaciones/Biblioteca/proyeccion%20nacional/Estimacion%20y%20proyeccion%20Nacional.pdf>
- Eclipse Foundation. (2019). *Eclipse Help*. www.eclipse.org
- Eberhard, D., Gary S., and Charles F. (2019). *Ethnologue: Languages of the World*. Twenty-second edition. Dallas, Texas: SIL International. Online version: <http://www.ethnologue.com>
- Fondo de Población de las Naciones Unidas (UNFPA). *Población Mundial 2017*. <https://www.unfpa.org/es/swop>
- Gálvez, S y Luis O. (2003). *Java a Tope: J2ME (Java 2 Micro Edition)*. Edición Electrónica. <http://www.lcc.uma.es/~galvez/ftp/libros/J2ME.pdf>
- iGuaraní (2012) *Diccionario y Traductor Guaraní*. www.iguarani.com
- Kantor I. (2019). *Java Script* <https://javascript.info/intro>
- Krivoshin N. (2001). *Gramática de la lengua guaraní*. Colección Ñemity.
- Lubliner, S., & Grisham, D. L. (2012). *Cognate Strategy Instruction. Providing powerful literacy tools to Spanish-speaking students*. En J. Fingon, & S. Ulanoff (Eds.), *Learning from Culturally and Linguistically Diverse Classrooms: Using Inquiry to Inform Practice* (pp. 105-123). New York: Teachers College Press.
- Manaris, B. y S., Brian, (1996). *Interactive Natural Language Processing: Building on Success*. Computer, IEEE.

- MDNwebdocs. (2018). *Hojas en Estilo en Cascada CSS*.
<https://developer.mozilla.org/es/docs/Web/CSS>
- Moliniers, P. 1986. Lecciones de Guaraní 1 Guaraní Peteña. Industrial Gráfica Comuneros S.A., Asunción, 116 pp.
- Moreno, A.(1998). *Lingüística computacional. Introducción a los modelos simbólicos, estadísticos y biológicos*. Madrid. Editorial Síntesis
- MySQL.com. (2018). *Manual MySQL*. <https://www.mysql.com/about>
- Nasser, E. Rolando, N. 1987. *Lenguaje 2*. Editorial Gráfica Comuneros S.A. Asunción.
- Natalizia, R. (1982). *Nihil – Roma – Maius Gramática Latina y una Síntesis de Gramática Histórica del Castellano*. Editorial Gráfica Comuneros. S.A., Asunción.
- Newmark, P. (1988). *Approaches to Translation*. Prentice Hall International (UK) Ltd., 195 pp
- Oracle Corporation. 2018.*MySQL Reference Manual*.
<https://dev.mysql.com/doc/refman/8.0/en/innoDB-introduction.html>
- Ortellado, B. (2008). *Lenguaje Creativo 7º y 8º*. Editora Ricor Grafic S.A. Lambaré.
- RAE. (2018). *Diccionario de la lengua española*. Real Academia Española. España.
<https://www.rae.es/>
- Secretaría de Planificación Estratégica, Ministerio de Educación de la República del Perú. (2009). *Estándares para el Uso de Herramientas de Desarrollo y Plataformas de Aplicaciones Web*. V4
- Secretaría De Políticas Lingüísticas Del Paraguay.(2018). *Corpus de Referencia del Guaraní Paraguayo Actual – COREGUAPA*. 2018.<http://www.spl.gov.py>
- Secretaría De Políticas Lingüísticas Del Paraguay.(2018). *Resolución 103*.
<http://www.spl.gov.py>
- Shannon C. E.. (1948). *A mathematical theory of communication*. Bell system technical, 27
<http://affect-reason-utility.com/1301/4/shannon1948.pdf>
- Suárez J. (2007). *Apuntes para compiladores II Un enfoque práctico en el proceso de Análisis y Síntesis en la Construcción de Compiladores*. UCSA.<http://juliosuarez.webs.com/APUNTES-II.pdf>
- The jQuery Foundation. (2019). *jQuery write less, do more*. <https://jquery.com/>
- Tertoolen, R.(2012).*Desarrollos en la traducción automática: esperando aún una traducción de calidad*. Utrecht University.
<http://revistacaracteres.net/wp-content/uploads/2012/11/Caracteresvol1n2noviembre2012-traduccion.pdf>
- Vez, J. y otros. (2017). *Aplicaciones de las nuevas tecnologías en el aprendizaje de la Lengua Castellana*. Secretaría General Técnica. Ministerio de Educación, Cultura y Deporte.
<https://sede.educacion.gob.es/publiventa/PdfServlet?pdf=VP11006.pdf&area=E>
- Wamp Server. (2018). *About*.<http://www.wampserver.es/#about>