



Universidad de León

**Departamento de Ingeniería Eléctrica
y de Sistemas y de Automática**

Tesis Doctoral

**Utilización del Lenguaje VILA_1 para el
Intercambio de Mensajes Usando el
Vocabulario Chino y el Español**

Xiaochen Yang

Dirigida por:

**Dr. Don Ángel Alonso Álvarez
Dr. Don Héctor Aláiz Moretón**

León, abril de 2010

Dedicatoria

Esta tesis doctoral está dedicada a mis padres.

Agradecimientos

Primeramente quiero mostrar mi mayor agradecimiento al Dr. D. Ángel Alonso Álvarez, y al Dr. D. Héctor Aláiz Moretón, directores de esta tesis doctoral, ya que sin su orientación y apoyo esta tesis no hubiera sido posible.

También quiero agradecer a todas las personas que me han ayudado y alentado a la hora de elaborar esta tesis, pero quiero hacer mención especial a alguno de ellos:

A mis compañeros de despacho Jia Fu, y Javier Alfonso Cendón con los que he compartido muchos momentos, y que siempre han estado a mi lado.

A los profesores Isaías García Rodríguez, Francisco Jesús Rodríguez Sedano, José Manuel Alija Pérez, Ramón Ángel Fernández Díaz, Carmen Benavides Cuellar, José Luis Calvo Rolle, Enrique López González, Luís Panizo Alonso, y Félix Riesco Peláez que siempre me aconsejaron y escucharon.

A mis padres, por el gran esfuerzo que han realizado para darme la educación que hoy me ha permitido llegar hasta aquí.

A mis amigos Yili Qin, Zinan Wu, Chen Chen, Yin Yang, Rongrong Zhong, Lei Song, Dingding Zhang, Chen Ling, Xiayu Yao, y Feng Xu.

También deseo mostrar mi agradecimiento a diversas instituciones, a la Dirección de la Escuela de Ingenierías Industrial e Informática, y a la Dirección del Departamento de Ingeniería Eléctrica y de Sistemas y de Automática, así como a la propia Universidad de León por haber puesto sus recursos a mi disposición cuando así lo solicité.

Espero que el trabajo realizado sea merecedor de toda la ayuda recibida.

Resumen

Esta tesis doctoral parte del lenguaje VILA_1, que es un lenguaje visual para la comunicación genérica entre humanos, o humanos y máquinas. VILA_1 se desarrolló como consecuencia de un cambio de paradigma en la Ingeniería del Conocimiento, que consiste en unificar el lenguaje usado por los seres humanos para generar conocimiento y el lenguaje usado para formalizar el conocimiento en los ordenadores.

El trabajo realizado en esta tesis doctoral ha consistido en diseñar y desarrollar un sistema de mensajería instantánea, para realizar el intercambio de mensajes entre chino y español usando VILA_1.

Abstract

This PhD Thesis is based on VILA_1 language, which is a visual language for generic communication between humans, or humans and machines. VILA_1 was developed as the result of a paradigm shift in Knowledge Engineering involving the unification of the language used by humans to generate knowledge with the language devised to formalize knowledge in computers.

The work of this doctoral dissertation is to design an instant messaging system, and showed in detail how to realize messages transformation between Chinese and Spanish using the VILA_1.

摘要

这篇博士论文是 VILA_1 语言的一部分，它是一种应用于人与人之间或者人与机器之间的通用通讯的虚拟语言。VILA_1 的研发作为一种在知识工程领域的改变的成果的示范，包括如何统一人类所使用的普通语言和计算机所使用的规范语言。

在此博士论文中实现了一个即时通讯系统的设计，开发并详细说明如何使用 VILA_1 语言实现中文和西班牙语的信息转换。

ÍNDICE DE CONTENIDO

DEDICATORIA.....	I
AGRADECIMIENTOS	III
RESUMEN	V
ABSTRACT	VII
摘要.....	IX

ÍNDICESXI

Índice de contenido	XI
Índice de figuras	XV
Índice de tablas.....	XIX

CAPÍTULO 1 (INTRODUCCIÓN)..... 1

1.1 Introducción	3
1.2 La Gramática de VILA_1	6
1.2.1 Las expresiones lingüísticas de identificación	7
1.2.2 Las expresiones lingüísticas para descubrir características	10
1.2.3 Las expresiones lingüísticas para describir acciones	20
1.3 Conclusiones	23

CAPÍTULO 2 (OBJETIVOS Y JUSTIFICACIÓN) 25

2.1 Justificación.....	27
2.2 Objetivos	27
2.3 Metodología de trabajo.....	27
2.4 Estructura de la tesis.....	28

CAPÍTULO 3 (ESTADO DEL ARTE) 31

3.1 Introducción	33
3.2 Mensajería instantánea	33
3.2.1 La historia y desarrollo de Mensajería Instantánea.....	34
3.2.2 Funciones básicas de Mensajería Instantánea	35

3.2.3	Principales aplicaciones cliente	36
3.2.3.1	Principales características de las aplicaciones cliente	37
3.2.3.2	Ventajas	47
3.2.3.3	Inconvenientes	48
3.3	Arquitectura Orientada a Servicios (SOA)	48
3.3.1	Los Servicios Web	50
3.3.2	Cronología	51
3.3.3	Componentes de la arquitectura	52
3.3.4	Funcionamiento de los Servicios Web	53
3.3.5	Extensible Markup Language (XML)	54
3.3.6	XML Schema	55
3.3.7	Web Services Description Language (WSDL)	55
3.3.8	Universal Description, Discovery and Integration (UDDI)	57
3.3.9	SOAP	58
3.4	Conclusiones	61

CAPÍTULO 4 (DISEÑO DE LA APLICACIÓN DE INTERCAMBIO DE MENSAJES CON VILA_1) 63

4.1	Introducción	65
4.2	Análisis y diseño del sistema	65
4.2.1	La definición de los módulos funcionales	66
4.2.2	La definición de los módulos funcionales	68
4.3	El diseño del terminal cliente	70
4.3.1	El diseño de la interfaz de inicio sesión	70
4.3.2	El diseño de la interfaz principal del sistema	71
4.3.3	El diseño de la interfaz de chatea	71
4.3.4	El uso de la aplicación de mensajería instantánea	80
4.2.4.1	El uso de tipo de mensajes comunes	81
4.2.4.2	El uso de tipo de mensajes propios	88
4.4	El diseño de servidor	90
4.4.1	Establecer la conexión	90
4.4.2	Conexión de las bases de datos	91
4.4.3	Formato de transmisión de mensajes	92
4.4.4	La función de traducción	106
4.5	El diseño de la base de datos	112
4.5.1	La base de datos de usuarios	113
4.5.2	La base de datos de lista de amigos	113
4.5.3	La base de datos de palabras VILA_1	113
4.5.4	La base de datos de estructuras VILA_1	115
4.6	Conclusiones	115

CAPÍTULO 5 (VALIDACIÓN EMPÍRICA) 117

5.1 Introducción	119
5.2 Metodología	120
5.2.1 Selección de la población.....	121
5.2.2 Diseño y ejecución de la dinámica de grupo.....	122
5.3 Diseño, realización de encuestas para evaluación de la prueba	137
5.4 Análisis de resultados y conclusiones	139
CAPITULO 6 (CONCLUSIONES FINALES Y TRABAJOS FUTUROS).....	145
6.1 Conclusiones finales.....	147
6.2 Trabajos futuros	147
ABREVIATURAS	149
REFERENCIAS	151

ÍNDICE DE FIGURAS

CAPÍTULO 1 (INTRODUCCIÓN).

Figura 1.1 Paradigma clásico de la Ingeniería del Conocimiento	3
Figura 1.2 Los ejemplos de la expresión lingüística de identificación.....	8
Figura 1.3 Los ejemplos de la aplicación de las expresiones lingüísticas de identificación	9
Figura 1.4 Los ejemplos de las agrupaciones tipo "Y" y tipo "O"	10
Figura 1.5 Los ejemplos de la expresión lingüística de la característica "Adjetivo"	11
Figura 1.6 Los ejemplos de la expresión lingüística de la característica "Valor"	12
Figura 1.7 Los ejemplos de la expresión lingüística de la característica "Relación"	13
Figura 1.8 Los ejemplos de la expresión lingüística de la característica "Espacio"	16
Figura 1.9 Los ejemplos de la expresión lingüística de la característica "Tiempo"	19
Figura 1.10 Los ejemplos de la expresión lingüística de la agrupación de características	20
Figura 1.11 Los ejemplos de la expresión lingüística para describir acciones	22
Figura 1.12 Los ejemplos de las expresiones complementarias de las acciones	23

CAPÍTULO 3 (ESTADO DEL ARTE).

Figura 3.1 Imágenes y características de Windows Live Messenger.....	39
Figura 3.2 Imágenes y características de Pidgin	40
Figura 3.3 Imágenes y características de Google Talk.....	41
Figura 3.4 Imágenes y características de Miranda IM	43
Figura 3.5 Imágenes y características de Skype.....	44
Figura 3.6 Imágenes y características de AOL	45
Figura 3.7 Imágenes y características de Yahoo Messenger.....	47
Figura 3.8 Ejemplo de un archivo WSDL.....	55
Figura 3.9 Representación gráfica del XMLSchema de un archivo WSDL.....	57
Figura 3.10 Ejemplo de un archivo UDDI	58
Figura 3.11 Arquitectura de Servicios Web	59

Figura 3.12 Arquitectura detallada de Servicios Web	60
Figura 3.13 Ejemplo de un mensaje SOAP	60

CAPÍTULO 4 (DISEÑO DE LA APLICACIÓN DE INTERCAMBIO DE MENSAJES CON VILA_1).

Figura 4.1 Estructura de C/S de 3 capas	66
Figura 4.2 Estructura de funcionamiento del sistema de mensajería instantánea	67
Figura 4.3 Los Módulos funcionales	68
Figura 4.4 Interfaz de inicio sesión.....	70
Figura 4.5 Interfaz principal del sistema	71
Figura 4.6 Interfaz de chat	72
Figura 4.7 Ventana de los datos del usuario	73
Figura 4.8 Ventana de editar los datos del usuario	73
Figura 4.9 Captura de pantalla de la interfaz de seleccionar idioma	74
Figura 4.10 Los diez idiomas más utilizados en el Internet	75
Figura 4.11 Captura de pantalla de la interfaz de ayuda.....	76
Figura 4.12 Captura de pantalla de la interfaz de los elementos de tres temas comunes de comunicación	77
Figura 4.13 Captura de pantalla de la interfaz de los elementos para editar mensaje propio.....	78
Figura 4.14 Captura de pantalla de la interfaz de los partes de un elemento.....	79
Figura 4.15 Ventana de ayuda del sistema	80
Figura 4.16 Captura de pantalla de la interfaz de elige tema de preguntar precio	81
Figura 4.17 Captura de pantalla de la interfaz de editar mensaje	82
Figura 4.18 Captura de pantalla de la interfaz de generar un mensaje	83
Figura 4.19 Captura de pantalla de la interfaz de editar mensaje	85
Figura 4.20 Captura de pantalla de la interfaz de generar un mensaje	86
Figura 4.21 Captura de pantalla de la interfaz de los mensajes se sale en el campo de los mensajes	87
Figura 4.22 Interfaz de chat de usuario chino Xiaochen	88
Figura 4.23 Captura de pantalla de la interfaz de editar un mensaje propio	89
Figura 4.24 Captura de pantalla de la interfaz de generar un mensaje propio.....	89
Figura 4.25 Esquema de funcionamiento del proceso de autenticación de usuarios.....	90
Figura 4.26 Ruta de comunicaciones JDBC con las bases de datos.....	91
Figura 4.27 Estructura de un archivo “.vla”	93
Figura 4.28 Un documento completo en VILA_1	95
Figura 4.29 XML del archivo “Acción” del ejemplo	97

Figura 4.30 XML del archivo "Adjetivo" del ejemplo	101
Figura 4.31 XML del archivo "EspacioA" del ejemplo.....	102
Figura 4.32 XML del archivo "Imagen" del ejemplo	103
Figura 4.33 XML del archivo "Mapa" del ejemplo	104
Figura 4.34 XML del archivo "Documento" del ejemplo.....	105
Figura 4.35 XML del archivo "Acción" del ejemplo en chino	106
Figura 4.36 XML del archivo "Adjetivo" del ejemplo en chino.....	108
Figura 4.37 XML del archivo "EspacioA" del ejemplo en chino	109
Figura 4.38 XML del archivo "Imagen" del ejemplo en chino.....	110
Figura 4.39 XML del archivo "Mapa" del ejemplo en chino	111
Figura 4.40 Esquema genérico de las bases de datos del sistema	112
Figura 4.41 Estructura de una tabla de la base de datos de usuarios.....	113
Figura 4.42 Estructura de una tabla de la base de datos de palabras VILA_1	114

CAPÍTULO 5 (VALIDACIÓN EMPÍRICA)

Figura 5.1 Gráfico de resultados de la evaluación del funcionamiento del prototipo.....	140
Figura 5.2 Gráfico de resultados de la evaluación de la interfaz de usuario del prototipo	141
Figura 5.3 Gráfico de resultados de la evaluación de la traducción de las frases desde español a chino.....	142
Figura 5.4 Gráfico de resultados de la evaluación de la traducción de las frases desde chino a español.....	142

ÍNDICE DE TABLAS

CAPÍTULO 4 (DISEÑO DE LA APLICACIÓN DE INTERCAMBIO DE MENSAJES CON VILA_1).

Tabla 4.1 Los tres idiomas más utilizados en la web	75
Tabla 4.2 Tabla MIME de un archivo “.vla”	93
Tabla 4.3 Control de extensiones del sistema	94

CAPÍTULO 5 (VALIDACIÓN EMPÍRICA).

Tabla 5.1 Participantes chinos.....	119
Tabla 5.2 Participantes españoles.....	120
Tabla 5.3 Encuesta para la evaluación del funcionamiento del prototipo	138
Tabla 5.4 Encuesta para la evaluación de la interfaz de usuario del prototipo ..	138

CAPÍTULO 1

Introducción



*En este capítulo se
dará una introducción
sobre la lenguaje
VILA_1, y indicará
este tesis doctoral es un
importante parte del
desarrollo de VILA_1.*



1.1 Introducción

Esta tesis doctoral forma parte de un proyecto de investigación desarrollado por el grupo de Cognomática de la Universidad de León. En el contexto de ese proyecto se ha creado un nuevo lenguaje visual al que se ha denominado VILA_1 [1]. El desarrollo de este lenguaje se ha apoyado en varios trabajos previos de Alonso [2] [3] [4] [5] [6] [7] [8]. El objetivo de este lenguaje es solucionar alguno de los problemas que tiene planteados la Ingeniería del Conocimiento. La Ingeniería del Conocimiento pretende representar el conocimiento de forma que pueda ser procesado automáticamente mediante un ordenador.

Son muchas las dificultades encontradas para automatizar el conocimiento, hasta el punto que, después de treinta años de investigaciones, el tema no está resuelto. En el grupo de investigación sobre Cognomática de la Universidad de León ha llegado a la conclusión de que para seguir avanzando es preciso cambiar de paradigma. A continuación se analizan las razones. En la (*Figura 1.1*) se sintetiza la estrategia seguida hasta la fecha:

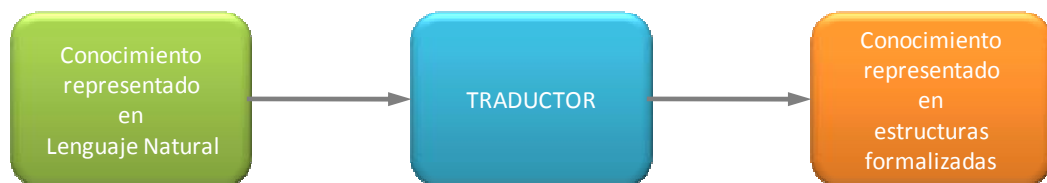


Figura 1.1 Paradigma clásico de la Ingeniería del Conocimiento

El conocimiento es generado en Lenguaje Natural por los seres humanos. Para poder automatizarlo en los ordenadores es preciso representar ese conocimiento en estructuras formalizadas. El paso del conocimiento expresado en Lenguaje Natural a estructuras formalizadas requiere un proceso de traducción. Esta traducción es realizada por los seres humanos.

¿Dónde está el problema? En el proceso de traducción. Para formalizar el conocimiento el traductor necesita eliminar las

ambigüedades propias del Lenguaje Natural, su falta de precisión, las diversas acepciones de los términos, su borrosidad estructural. Y para realizar esta tarea no hay normas, no hay estandarización. Cada traductor aplica sus propios criterios y eso conduce a resultados muy subjetivos. Las estructuras formalizadas por un traductor son difícilmente compatibles con las realizadas por otro. El objetivo básico de la reusabilidad no se consigue.

Además del grave problema citado hay más, entre ellos:

- El traductor debe ser un experto en el dominio de conocimiento que se propone traducir y también debe ser un experto en las técnicas de formalización del conocimiento. Actualmente es difícil encontrar perfiles profesionales que cumplan esta condición.
- La generación de conocimiento en Lenguaje Natural es tan dinámica y voluminosa que se necesitaría una legión de traductores para formalizar el conocimiento generado continuamente.

Después de varios años de reflexión hemos llegado a la conclusión de que es necesario eliminar el proceso de traducción, causante de los males citados. Nuestra propuesta consiste en generar el conocimiento directamente en un lenguaje formalizado, un lenguaje común para el hombre y para la máquina. Y esto representa un cambio de paradigma. Este nuevo lenguaje debe tener resueltos en su estructura los problemas de ambigüedad y borrosidad propios del lenguaje natural. ¿Cómo hacerlo? Para responder a esta pregunta se ha seguido el siguiente razonamiento:

- Puesto que la inmensa mayoría de las transacciones de conocimiento hombre-máquina y hombre-hombre son a través de una pantalla electrónica el lenguaje debe tener una orientación visual.

- Para obtener buen rendimiento de la gran capacidad de procesamiento que ofrece el sentido de la vista el lenguaje debe formularse al menos en dos dimensiones.
- Para que el nuevo lenguaje sea fácilmente aceptado debe introducirse de forma progresiva.

Se ha bautizado al nuevo lenguaje como VILA (Visual Language), y se ha dividido en tres niveles que se denominan VILA_1, VILA_2 y VILA_3.

VILA_1 utiliza muchas palabras de los lenguajes naturales e incorpora un conjunto amplio de elementos gráficos, inicialmente en dos dimensiones, pero que con su desarrollo es previsible que se amplíe con elementos tridimensionales y animados.

VILA_2 mantiene los elementos gráficos de VILA_1 y elimina todas las palabras. Se produce una desconexión total con los lenguajes naturales hasta el punto de que se generará una grafía específica para representar los conceptos que se definan.

VILA_3 es una ampliación de VILA_2. Incorporará representación gráfica directa de los elementos físicos definidos mediante el lenguaje, así como los cálculos matemáticos necesarios para obtener y representar las magnitudes y los parámetros propios de la ciencia y de la tecnología.

VILA_1 ya ha sido desarrollado. Su aprendizaje y manejo es inmediato, ya que los elementos gráficos que incorpora son muy intuitivos y las palabras que usa coinciden con las de los lenguajes naturales, pueden usarse las del idioma que cada uno desee.

En síntesis, lo que se está diciendo es que el Lenguaje Natural, gran artífice del desarrollo de la Humanidad, ha tocado techo. La ciencia y la tecnología necesitan un lenguaje estructurado científicamente. Es una necesidad vital. El Lenguaje Natural fue creado en otro contexto histórico, con unas necesidades muy diferentes a las actuales y sin las

herramientas electrónicas que existen y las que van a existir. El lenguaje es tan importante en el desarrollo del conocimiento que si no evoluciona, el lenguaje, el desarrollo del conocimiento quedará seriamente amenazado.

1.2 La Gramática de VILA_1

La gramática de VILA_1 [1] se estructura sobre los conceptos de *expresión lingüística* y de *rectángulo*. Hay muchos tipos de expresiones lingüísticas. Todas ellas se construyen mediante agrupaciones de los tres tipos básicos. Estos son:

- Las expresiones lingüísticas de identificación. Sirven para identificar las entidades y los conceptos, desde el punto de vista del destinatario de la expresión. Se entiende por entidad cualquier elemento diferenciado de otros, tanto real como virtual.
- Las expresiones lingüísticas para describir características. Sirven para describir las características de las entidades y de los conceptos.
- Las expresiones lingüísticas para describir acciones. Sirven para describir las acciones que producen las fuerzas de la Naturaleza. Cualquier fuerza.

En las expresiones de identificación se supone que todo lo expresado es conocido por el destinatario. Sin embargo en los otros dos tipos de expresiones se supone que se exponen conceptos desconocidas para el destinatario.

El rectángulo sirve para ubicar las expresiones lingüísticas en el plano y para referenciarlas. Hay tantos tipos de rectángulos como tipos de expresiones lingüísticas. Cada expresión lingüística lleva asociado su correspondiente rectángulo. Puede decirse que cada rectángulo es el contenedor de una expresión lingüística.

1.2.1 Las expresiones lingüísticas de identificación

Son el primero de los tres tipos básicos de expresiones lingüísticas. El objetivo de este tipo de expresiones es el de identificar tanto los conceptos como las entidades.

➤ *Identificación mediante término*

Consiste en utilizar un solo término. Puede ser una palabra u otro elemento gráfico cualquiera (signos, dibujos, fotografías, etc.). Algunos ejemplos son los siguientes:

coche

oveja

árbol

libro

ilusión

ordenador

Cuando se usan palabras de un idioma, cualquiera, sólo se utiliza una forma de la misma, la más común. No se usa el plural y la diferenciación entre masculino y femenino sólo se hace al referirse a seres vivos que tengan diferenciación de sexos (Ej.: vaca, buey, etc.). Pero a los objetos o conceptos no se les asigna género.

También se incluyen en este apartado los términos destinados a identificar individualidades o entidades concretas. Es el caso de los denominados nombres propios en los lenguajes naturales. Aunque aquí se incluye cualquier tipo de identificador de carácter alfanumérico como por ejemplo la matrícula de un coche. En todos los casos el identificador aparecerá entre comillas. Esos identificadores pueden constar de cualquier conjunto de caracteres alfanuméricos, incluido el espacio en blanco. Ejemplos:

"España"

"Juan Pérez"

"NCV 33465"

"Pablo VI"

Otro tipo de identificadores que se consideran de término único son aquellos identificadores de carácter genérico, es decir, que sirven para representar conceptos o conjuntos de individualidades, y que constan de varias palabras unidas en forma de frases hechas. En este caso se

unirán todas las palabras de la expresión mediante una barra baja “_”.
Ejemplos:

carnet_identidad Número_bastidor Colegio_profesional Escuela_ingeniería

➤ **Identificación mediante término único con especificadores**

Consiste en concretar el ámbito de identificación de un término único mediante una serie de especificadores gráficos. Entre esos especificadores están los siguientes:

- Negación (X);
- Interrogación (?);
- Singular/plural: por defecto se entiende singular. Para especificar plural (□).
- Cuantificación numérica: un número; En combinación con los números se pueden combinar algunos signos matemáticos como: < ; > ; ≈.
- Cuantificación cualitativa: ninguno (□); poco/s (□); bastante/s (□); mucho/s (□); todo/s (□).
- Determinación (□); La ausencia de determinación se considera indeterminación.

Algunos ejemplos son los siguientes:

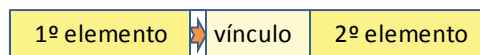
Expresiones en VILA_1	Traducción a lenguaje natural
niño	El niño
6 elefante	Seis elefantes
□ árbol	Ningún árbol
juguete	Muchos juguetes
ilusión	Las ilusiones
> 10 mesa	Más de diez mesas
monumento	Un monumento

Figura 1.2 Los ejemplos de la expresión lingüística de identificación

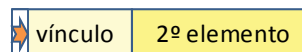
➤ *Ampliación de las expresiones lingüísticas de identificación*

Tanto las expresiones lingüísticas para describir características, como las destinadas a describir acciones pueden usarse para identificar elementos. Cuando así sea se colocará delante el siguiente icono (👉). En el gráfico anterior se muestran algunos ejemplos.

En el caso de las expresiones lingüísticas para expresar características de relación es frecuente utilizar el vínculo y el segundo elemento para identificar el primer elemento. La estructura de la expresión es la siguiente:



Y en este caso se transforma en esta otra:



Algunos ejemplos de este tipo de expresiones son las siguientes:

Expresiones en VILA_1	Traducción a lenguaje natural
👉👇 madre Ana	La madre de Ana
👉👇 autor 📄 trabajo	El autor del trabajo
👉👇 capital España	La capital de España
👉👇 veterinario 🐕 perro	El veterinario del perro

Figura 1.3 Los ejemplos de la aplicación de las expresiones lingüísticas de identificación

➤ *Agupaciones tipo "Y" y tipo "O"*

Las expresiones lingüísticas del mismo nivel pueden agruparse mediante los operadores tipo "y" y tipo "o". Los signos gráficos utilizados para cada tipo son los siguientes:

- Agrupación tipo “Y”: ,
- Agrupación tipo “O”: |

Algunos ejemplos que incluyen agrupaciones de este tipo son los siguientes:

Expresiones en VILA_1	Traducción a lenguaje natural
niño guapo , alto	El niño guapo y alto
estudiante ambicioso , listo	El estudiante es ambicioso y listo
mesa color = castaño roble	Una mesa de color castaño o roble
madre Ana María	La madre de Ana o de María
<div style="border: 1px solid black; padding: 2px;"> Juan , Pedro romper balón </div>	Juan y Pedro rompieron el balón

Figura 1.4 Los ejemplos de las agrupaciones tipo “Y” y tipo “O”

1.2.2 Las expresiones lingüísticas para describir características

El segundo tipo de expresiones lingüísticas básicas lo constituyen las que se utilizan para describir características. Son expresiones que en el lenguaje natural llevan asociados los verbos **ser**, **estar**, **tener** y algunos otros. Estos verbos tienen grandes particularidades y diferencias entre unos idiomas y otros. En VILA_1 no se utilizan verbos para expresar las características de algo. Las características siempre se aplican a uno o más elementos ya identificados, mediante la correspondiente expresión lingüística. Las características tienen el objetivo de trasladar información, en principio desconocida, para el destinatario de la expresión lingüística. Estas se distribuyen en algunos grupos.

➤ **La característica "Adjetivo"**

Los adjetivos de los lenguajes naturales sirven para darle valor cualitativo a magnitudes y otros elementos de valoración. El rectángulo destinado a expresar una característica adjetivo tiene la siguiente forma:



Delante del adjetivo pueden incluirse los siguientes especificadores ya descritos anteriormente: X; ?; [1]; [2]; [3].

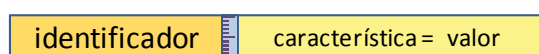
A continuación se describen algunos ejemplos:

Expresiones en VILA_1	Traducción a lenguaje natural
[1] niño [2] guapo	El niño es guapo
[1] estudiante [2] X ambicioso	El estudiante no es ambicioso
[1] cuerda [2] [3] rígido	La cuerda es muy rígida
Antonio [2] ? [3] inteligente	¿Es Antonio muy inteligente?

Figura 1.5 Los ejemplos de la expresión lingüística de la característica "Adjetivo"

➤ **La característica "valor"**

Los adjetivos de los lenguajes naturales, como ya se dijo, engloban en un solo término tanto el tipo de característica como el valor cualitativo asignado a la misma. Pero también es posible especificar la característica y asignarle un valor. Es lo que se hace en el rectángulo que denominamos característica valor. El rectángulo destinado a expresar una característica valor tiene la siguiente forma:



Para la asignación del valor hay varias opciones:

- Que exista un término, palabra o elemento gráfico, que identifique el valor de la característica, por ejemplo “rojo”, “rubio”, etc. En este caso se puede especificar el valor con la palabra correspondiente o con un elemento gráfico que lo represente.
- Que la característica permita cuantificación numérica. En este caso se expresará el número correspondiente, seguido de un espacio y la palabra que identifique la magnitud (Ej.: metro, kilogramo, etc.). El número puede ir precedido de los siguientes especificadores matemáticos: $<$; $>$; \approx . Cuando se quiera especificar un rango se intercalará entre los dos números un guión (–).
- Cuando en una característica no se den ninguno de los dos casos anteriores se recurrirá a los elementos gráficos que representan los conceptos: nada, poco, bastante, mucho, todo.

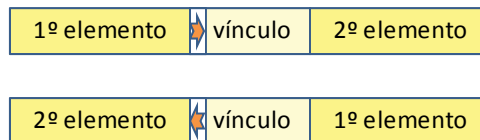
A continuación se describen algunos ejemplos:

Expresiones en VILA_1	Traducción a lenguaje natural
torre altura = 12 metro	La torre tiene una altura de doce metros
tabla ancho = 25 centímetro	Una tabla tiene veinticinco centímetros de ancho
niño ilusión =	Un niño tiene mucha ilusión
moto matrícula = 5267 NJK	La matrícula de la moto es 5267 NJK
mesa color = castaño	Una mesa es de color castaño
Alberto peso = > 100 kilogramo	Alberto pesa más de cien kilos
señal frecuencia = 1,5 – 2 gigahertzio	La señal tiene una frecuencia comprendida entre 1, 5 y 2 gigahertzios

Figura 1.6 Los ejemplos de la expresión lingüística de la característica “Valor”

➤ **La característica "relación"**

La característica relación permite expresar explícitamente el vínculo que existe entre dos elementos, sean entidades o conceptos. Por ejemplo el concepto autor permite vincular a un artista con su obra, el concepto hermano permite vincular a dos personas, etc. Esta es una característica que tiene estructura bidireccional, es decir, que puede leerse hacia adelante y hacia atrás. El símbolo de esta característica tiene forma de flecha, la cual indica en la dirección que debe hacerse la lectura. La característica incorpora, además del símbolo que la identifica, otros tres rectángulos. En los dos de los extremos se especifican los elementos que se vinculan y en el central el tipo de vínculo que tienen. Si no se desea especificar el tipo de vínculo puede dejarse este campo en blanco. La estructura es la siguiente:



A continuación se describen algunos ejemplos:

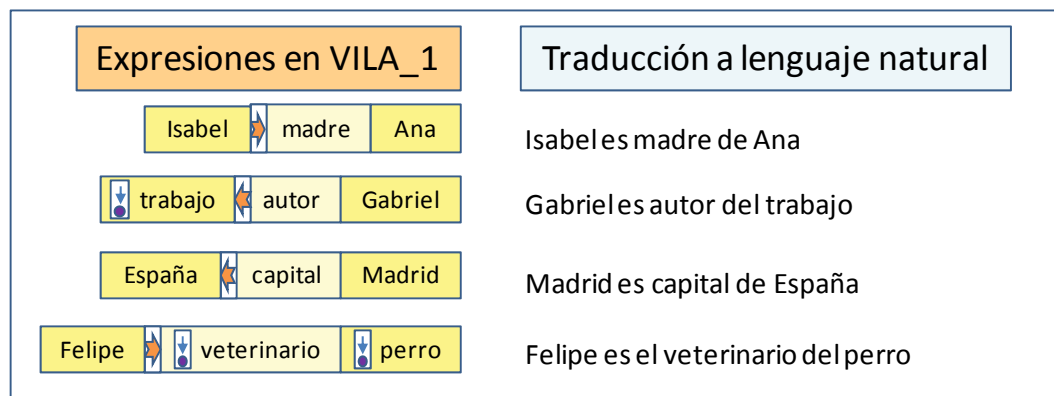


Figura 1.7 Los ejemplos de la expresión lingüística de la característica "Relación"

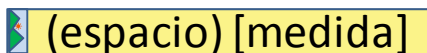
➤ **La característica "espacio"**

Mediante esta característica puede especificarse el espacio físico o virtual en el que se encuentra ubicada la entidad o el concepto que se

está describiendo o donde sucede algo. Hay diversas formas de especificar el espacio. En VILA_1 se usan los siguientes términos relativos como punto de partida para especificar el espacio:

- Dentro
- Fuera
- Arriba
- Debajo
- Derecha
- Izquierda
- Norte
- Sur
- Este
- Oeste
- Al lado
- Cerca
- Lejos
- Alrededor
- Periferia
- Entre
- Desde ... hasta

Exceptuando para los dos últimos términos relativos (*entre* y *desde ... hasta*), el rectángulo del espacio tiene la siguiente estructura:

 (espacio) [medida]

La sintaxis del campo espacio es la siguiente:

- Siempre comienza por un término relativo (*dentro*, *fuera*, *arriba*, etc.), seguido de una flecha. El término relativo *dentro* puede omitirse ya que se considera el término por defecto.
- A continuación se coloca una expresión identificadora de un espacio (armario, jardín, casa, etc.).

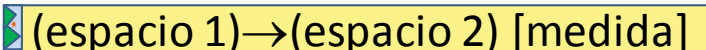
- Las expresiones descritas en los dos puntos anteriores pueden secuenciarse tantas veces como se desee. Irán separadas por “;”.
- Pueden usarse como identificadores de espacio figuras geométricas (**círculo**, **triángulo**, **cuadrado**, etc.). En este caso podrán incluirse entre corchetes todos los datos de identificación de la figura que se deseen, separados por “;”. Por ejemplo: **círculo [centro = ciudad “Madrid”; radio = 80 kilómetro]**
- También podrán usarse como identificadores términos como **ruta** y **trayectoria**, añadiendo entre corchetes los datos que se deseen.

En el campo medida podrá, si se desea, aportarse información sobre longitud, superficie y volumen del espacio definido previamente.

Para el término relativo *entre* el rectángulo del espacio tiene la siguiente estructura:

 (espacio 1)-(espacio 2) [medida]

Y finalmente para el término relativo *desde ... hasta* el rectángulo del espacio tiene la siguiente estructura:

 (espacio 1)→(espacio 2) [medida]

La sintaxis de **espacio 1**, **espacio 2** y **medida** es igual a la descrita previamente para los otros términos relativos. En el caso de la expresión correspondiente al término relativo *desde ... hasta* podrán dejarse en blanco cualquiera de los dos espacios, con lo que la expresión pasa a convertirse en *desde* o *hasta* respectivamente. Estos términos relativos pueden anidarse con los anteriores, formando expresiones todo lo complejas que se desee.

A continuación se describen algunos ejemplos:

Expresiones en VILA_1	Traducción a lenguaje natural
(Catedral)	Dentro de la Catedral
(encima → armario; habitación)	Encima del armario que está en la habitación
(lejos → ciudad "Madrid")	Lejos de la ciudad de Madrid
(dentro → oeste → jardín)	Dentro y al oeste del jardín

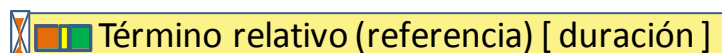
Figura 1.8 Los ejemplos de la expresión lingüística de la característica "Espacio"

➤ **La característica "tiempo"**

Mediante esta característica puede especificarse el momento o el intervalo temporal en el que algo se define o algo sucede. A lo largo del tiempo todo cambia. Pero el ritmo del cambio es diferente o muy diferente para unas cosas y para otras. Hay diversas formas de especificar un intervalo temporal. En VILA_1 se usan los siguientes términos relativos como punto de partida para especificar el tiempo:

- Antes
- Después
- Durante
- Desde
- Hasta
- Desde Hasta
- Entre

En cualquiera de los casos anteriores se puede añadir la duración del intervalo si se desea. La estructura del rectángulo tiempo varía según los términos relativos que se usen. Para los términos *antes*, *después*, *durante desde* y *hasta* la estructura es la siguiente:



Además del icono de identificación del rectángulo tiempo hay otros cuatro campos en el interior del rectángulo. En el primero se puede indicar si se está refiriendo al pasado (marrón), al presente (amarillo) o al futuro (verde). En el segundo se especifica el término relativo a la referencia temporal. En el tercero se indica la referencia temporal y en el cuarto la duración del intervalo.


La sintaxis de los distintos campos es la siguiente:


- Campo genérico:
 - Marrón = pasado
 - Amarillo = presente
 - Verde = futuro
 - Marrón-amarillo-verde = pasado, presente y futuro
 - Marrón-amarillo = pasado y presente
 - Amarillo-verde = presente y futuro
 - Marrón-verde = pasado y futuro
- Los términos relativos pueden ser cualquiera de los siguientes:
 - Antes
 - Después
 - Durante. *Si no expresa nada se sobreentiende este término.*
 - Desde
 - Hasta
- En el campo referencia se indicará una expresión temporal que puede tener los siguientes significados:
 - Intervalo temporal cuando se usa con *durante*.
 - Referencia anterior cuando se usa con *después*.
 - Referencia posterior cuando se usa con *antes*.
 - Punto de partida cuando se usa con *desde*.
 - Punto final cuando se usa con *hasta*.

En todos los casos la expresión temporal puede adoptar las siguientes formas:

- Mediante un nombre. Ej.: enero, lunes, noche, etc.
 - Mediante una unidad temporal y un número. Ej.: (día = 3), (año = 1987), (hora = 12), etc.
 - Mediante una secuencia, separada por “;” de identificadores individuales. Se colocaran primero los identificadores temporales de menor duración. Ej.: (día = 3; marzo), (día = 15; mes = agosto; año = 1990;), (mediodía; mañana), etc.
 - Mediante una expresión lingüística tipo acción. Se analizará posteriormente.
- Campo “duración”. Existen las siguientes posibilidades para identificar la duración:
 - Mediante un número y la correspondiente unidad temporal. Ej.: (3 día), (2 semana), (15 minuto), etc.
 - Combinando varias unidades temporales. Ej.: (1 mes, 6 día)
 - Mediante una expresión lingüística tipo acción. Se analizará posteriormente.

Para los términos *desde ... hasta* y *entre* la estructura es la siguiente:

 desde (ref. 1) hasta (ref. 2) [duración]

 entre (ref. 1) - (ref. 2) [duración]

La sintaxis de las referencias temporales y de la duración es la misma que la descrita anteriormente.

A continuación se describen algunos ejemplos:





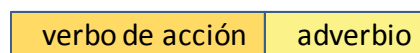



Expresiones en VILA_1	Traducción a lenguaje natural
 desde (hora = 10; ayer) hasta(hora =16; ayer)	Ayer desde las 10 de la mañana hasta las 4 de la tarde
 desde (mañana) [3 mes]	Desde mañana y durante 3 meses
 (año =2004)	Durante el año 2004
 después (agosto)	Después de agosto

Figura 1.9 Los ejemplos de la expresión lingüística de la característica "Tiempo"

➤ **La característica "adverbio"**

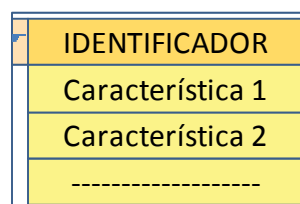
Los adverbios sirven para calificar la forma en la que se realiza una acción. Van siempre asociados a un verbo de acción y por eso no es necesario adjudicarles un icono de identificación. El rectángulo destinado a expresar una característica adverbio tiene la siguiente forma:




Delante del adverbio pueden incluirse los siguientes especificadores ya descritos anteriormente: X; ?; ; ; .

➤ **Agrupación de características**

Cuando se desee expresar varias características de un mismo elemento pueden agruparse todas ellas en la siguiente estructura:



Puesto que el identificador de todas las características es común se especifica una sola vez y de esa forma puede eliminarse de cada una de las características. Teniendo en cuenta que el elemento aglutinador de este tipo de expresiones es un identificador, se le asigna el icono de identificación (), para facilitar la comprensión de la expresión. En el siguiente gráfico se muestran algunos ejemplos. El segundo ejemplo contiene el símbolo (ϵ), para indicar el concepto de pertenencia.

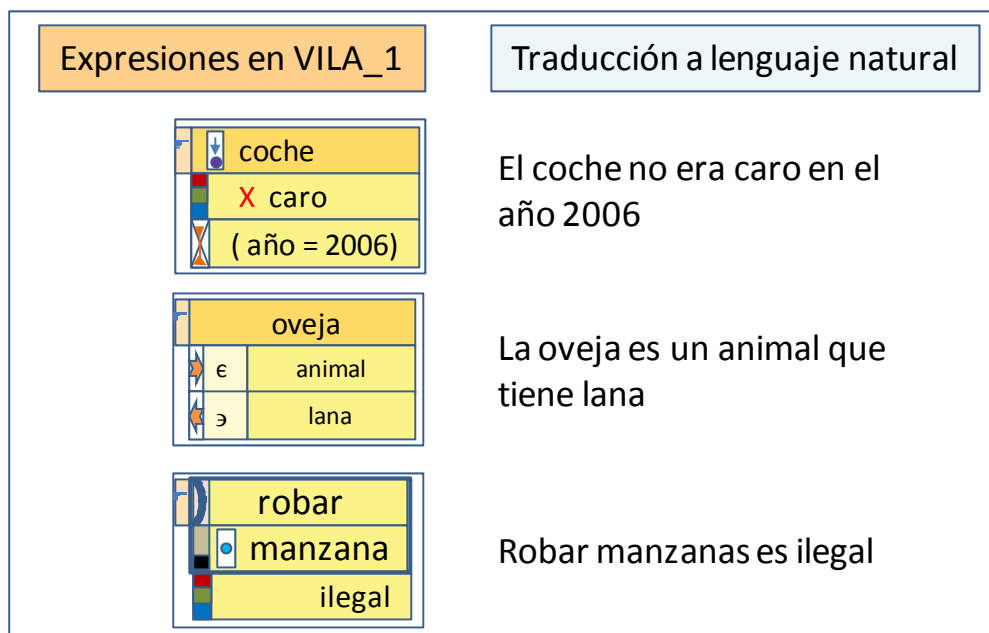


Figura 1.10 Los ejemplos de la expresión lingüística de la agrupación de características

1.2.3 Las expresiones lingüísticas para describir acciones

El tercer bloque de las expresiones lingüísticas básicas lo constituyen las que se utilizan para describir acciones. Son expresiones que sirven para describir los cambios de características de un concepto o entidad. Estos cambios se producen por la acción de fuerzas y siempre están

asociados a un intervalo temporal. La estructura del rectángulo para describir acciones es la siguiente:



El elemento central de este tipo de expresiones es el **verbo**. Con él se describe la acción correspondiente. El verbo se expresa siempre en infinitivo. El **sujeto** es el elemento generador de las fuerzas que producen la acción. El sujeto se expresa con una expresión de identificación. Mediante el **tiempo** se expresa el momento o el intervalo temporal en el que se produce la acción. La estructura del rectángulo tiempo ya se explicó en el apartado correspondiente a las expresiones para describir características. Solo cabe añadir aquí que para expresar el imperativo se dejarán en blanco los rectángulos correspondientes al pasado, al presente y al futuro. Cuando se quiera utilizar el verbo en infinitivo no se usará el rectángulo tiempo.

El **objeto** de la acción es el elemento sobre el que actúan directamente las fuerzas propias de la acción. En algunos casos el objeto de una acción puede ir destinado o dirigido a otro elemento. A este segundo elemento se le denomina **destinatario** de la acción. El rectángulo **espacio** sirve para especificar el lugar donde se desarrolla la acción. Sus características ya se describieron en el apartado correspondiente a las expresiones para describir características. En el gráfico siguiente se muestran algunos ejemplos.

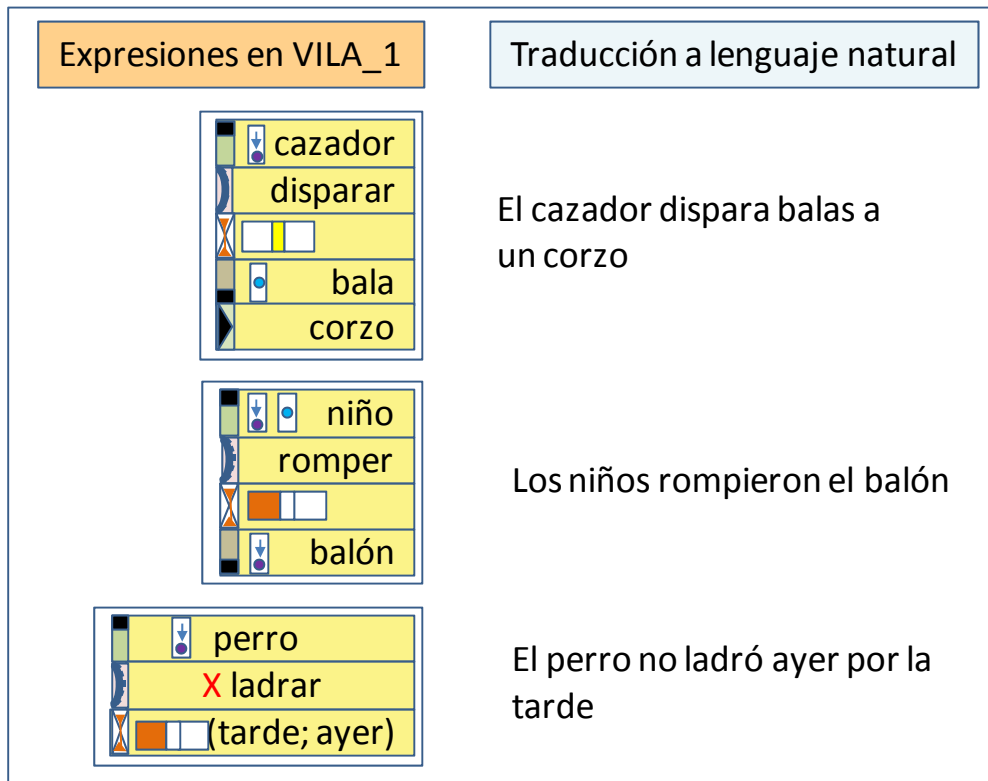


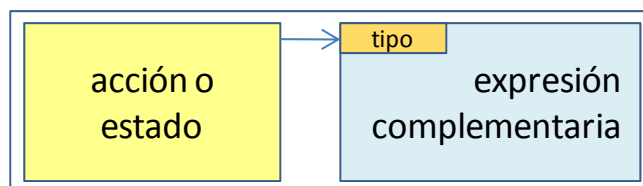
Figura 1.11 Los ejemplos de la expresión lingüística para describir acciones

➤ **Expresiones complementarias de las acciones**

Con frecuencia es preciso complementar las acciones. Son expresiones que sirven para especificar:

- El objetivo de la acción.
- La causa de la acción.
- La condición de la acción.
- Las consecuencias de la acción.

La estructura general es la siguiente:



Algunos ejemplos de este tipo de expresiones son los siguientes:

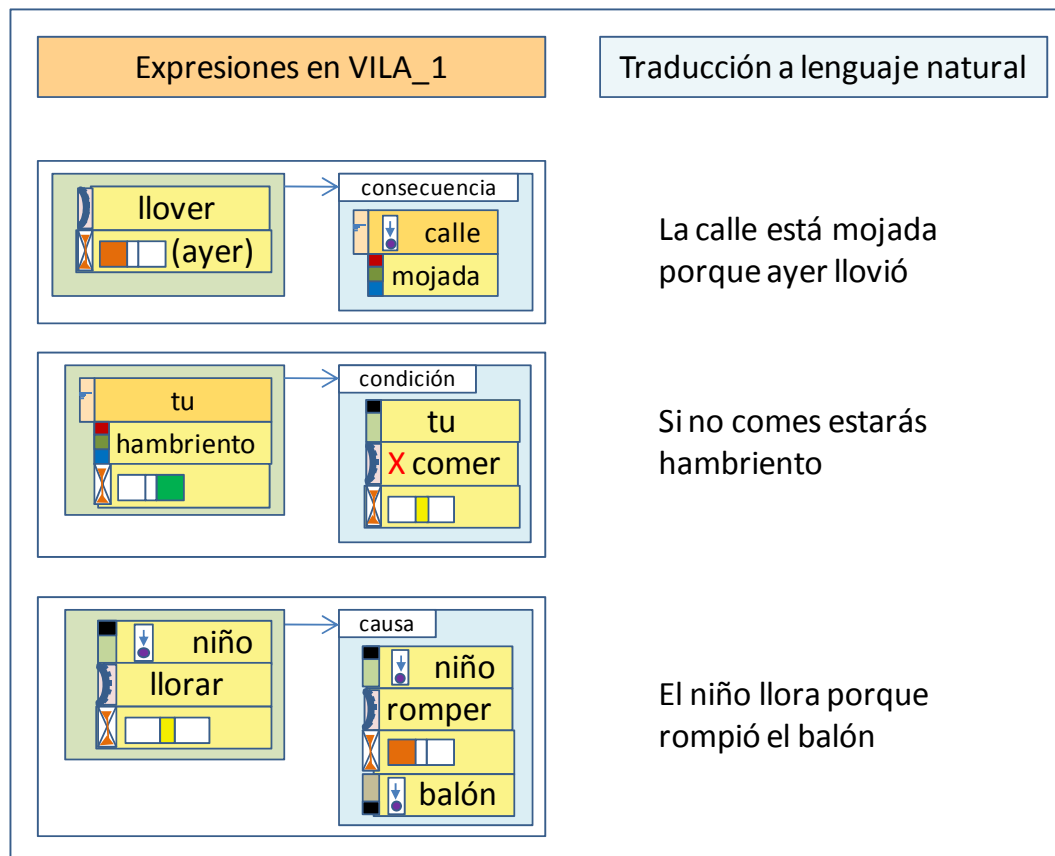


Figura 1.12 Los ejemplos de las expresiones complementarias de las acciones

1.3 Conclusiones

De lo dicho en las páginas anteriores puede deducirse lo siguiente:

- VILA_1 es un lenguaje nuevo, creado para la comunicación tanto hombre-hombre, como hombre-máquina, siempre mediante el uso de una pantalla electrónica.
- VILA_1 es el lenguaje de la ACCESIBILIDAD, y es así porque comparado con los lenguajes naturales es:
 - Mucho menos complejo. Al utilizar 2 dimensiones todo es más sencillo.
 - Menos ambiguo. Porque tiene una estructura semi-formal.

- Es universal. Porque permite la comunicación entre personas de cualquier idioma.
- VILA_1 tiene dos características básicas que le diferencian de los lenguajes naturales:
 - Es un lenguaje diseñado para optimizar la percepción visual, que admite las tres dimensiones espaciales y el tiempo, es decir, la animación.
 - Es un lenguaje semi-formal. Por lo tanto permite un nivel importante de automatización, incluidas todas las búsquedas conceptuales vinculadas a la estructura sintáctica del lenguaje.
- VILA_1 no es un traductor de lenguajes naturales.
- VILA_1 es un lenguaje visual que tiene tantas variantes como idiomas naturales (Hay VILA_1_Español; VILA_1_Chino; VILA_1_Inglés; etc.).
- La traducción entre las distintas variantes de VILA_1 es automática e inmediata.
- El aprendizaje de VILA_1 es extremadamente sencillo.

CAPÍTULO 2

Objetivos, justificación y estructura de la tesis



*En este capítulo se
dará una introducción
sobre la justificación
principal de esta tesis,
analizando sus
objetivos, y presentará
la metodología y
estructura de esta tesis.*



2.1 Justificación

El grupo de investigación dirigido por el Dr. Don Ángel Alonso Álvarez ha desarrollado un lenguaje visual, denominado VILA_1, como se introdujo en el capítulo anterior.

Las características de VILA_1, son muy diferentes a las del resto de los lenguajes visuales existentes en la actualidad, por este motivo no existe ningún tipo de guía, ni de especificación para el desarrollo de una aplicación informática, que permita la edición, y el uso de este lenguaje en Internet, debido a todo esto se plantea la necesidad de realizar una investigación para resolver este problema, de donde parte esta tesis doctoral.

2.2 Objetivos

El objetivo de la tesis consiste en diseñar una aplicación informática que sea eficiente para intercambiar mensaje entre personas de idioma chino y español, usando el lenguaje VILA_1.

Para conseguir el objetivo citado se apoya en otros trabajos que se están realizando en el grupo de investigación, como son el perfeccionamiento de la gramática de VILA_1 y la creación de base de datos que relacionan el vocabulario del idioma chino con el del español.

2.3 Metodología de trabajo

La presente tesis se ha elaborado según la metodología presentada en [9], cuyos pasos principales son: determinación del problema, formulación de la hipótesis, validación de la hipótesis y análisis de resultados, los cuales se detallan a continuación.

- Determinación del problema (Capítulo 3). Este trabajo surgió tras la creación del lenguaje VILA_1, y la necesidad de

desarrollar una plataforma para su uso en Internet. Después de la identificación del problema, se estudio la literatura más relevante de las aplicaciones de mensajería instantánea, y de las Arquitecturas de los Servicios en Internet.

- Formulación de la hipótesis (Capítulo 4). Así pues, la hipótesis de la que se partió en esta tesis es que un lenguaje visual con las características de VILA_1 necesita, para poder ser aprovechado, un diseño de la aplicación de tipo mensajería instantánea que soporta la comunicación entre las personales que tienen idiomas distintos.
- Validación de la hipótesis (Capítulo 5). Partiendo de la hipótesis formulada en el capítulo anterior, se definió un proceso de análisis y evaluación de la hipótesis, que se realizaron pruebas con el prototipo desarrollado en la Universidad de León.
- Análisis de resultados (Capítulo 6). Se analizaron los distintos resultados obtenidos en el desarrollo de la presente tesis, estableciendo una serie de conclusiones y líneas futuras.

2.4 Estructura de la tesis

A continuación se realizará un recorrido global por los capítulos que conforman esta memoria.

- En el capítulo 3, se presenta una síntesis del estado del arte necesario para acometer esta tesis. Incluye aspectos de los principales temas o áreas de conocimiento relacionados con la investigación, de la siguiente manera:
 - En la primera parte de este capítulo se detalla el estado del arte más relevante sobre las aplicaciones de mensajería instantánea.

- En la segunda parte se proporciona información referente a los servicios Web.

Es importante señalar, que de cada uno de los temas presentados, se hará una breve valoración y reseña de por qué han sido incluidos en esta tesis, y su relación directa con ella. Adicionalmente, para terminar este capítulo se presentarán las conclusiones más importantes del estado del arte.

- En el capítulo 4, se especifica el diseño de la aplicación de tipo mensajería instantánea que utiliza el lenguaje VILA_1, y cómo se realiza la función de traducción automática en el intercambio de mensajes.
- En el capítulo 5, se presentará el proceso de validación de la solución propuesta, cuyo objetivo principal es identificar las ventajas y también los problemas prácticos desde una visión externa, y de usuario final, lo que puede ayudar a obtener una retroalimentación sobre la utilidad de esta herramienta y del propio lenguaje como mecanismo de comunicación entre personas.
- En el capítulo 6, se presentarán todas las conclusiones que se han obtenido al terminar esta investigación y los trabajos que se podrán iniciar partiendo de los resultados de la misma.
- Para finalizar, se mostrará la lista de abreviaturas usadas en esta memoria. Después, se señalarán las referencias bibliográficas empleadas durante la investigación.

CAPÍTULO 3

Estado del arte



*En este capítulo se
incluye una
introducción de los
distintos conceptos y
técnicas clave para la
elaboración de la
presente tesis doctoral,
ya sea en su
planteamiento o en su
resolución.*



3.1 Introducción

Se debe partir de una serie de tecnologías claves para el planteamiento de las consideraciones metodológicas que permitan alcanzar los objetivos descritos en el capítulo anterior.

Las tecnologías clave de las que tratará este estado del arte son las siguientes:

- Sistemas de mensajería instantánea existentes en el mercado.
- Aspectos arquitectónicos y de funcionamiento de los Servicios de Internet, así como los formatos serializados existentes en la actualidad, donde se analizarán sus aspectos generales, tipos de servicios existentes, arquitecturas, y metodologías, para así poder especificar de la manera más eficiente, la especificación de la arquitectura interna para el uso en un sistema de mensajería del lenguaje VILA_1.

Este capítulo está estructurado de la siguiente forma; la sección 3.2 introduce los sistemas de mensajería instantánea. La sección 3.3 enumerando las arquitecturas más importantes existentes en la actualidad, así como sus ventajas e inconvenientes. La sección 3.4 se enumeran las conclusiones que se han obtenido del estudio del presente estado del arte.

Después del análisis de la estructura de este capítulo, resulta fundamental comenzar estableciendo los aspectos más relevantes de los distintos tipos de lenguajes, debido a que uno de los objetivos de esta tesis doctoral es estudiar la estructura y características de VILA_1, y para ello es necesario conocer su clasificación.

3.2 Mensajería Instantánea

La Mensajería Instantánea, permite a las personas identificar los usuarios en red y realizar el intercambio de mensajes en tiempo real por internet, se han denominado la mejor forma de comunicación de

internet desde la invención del correo electrónico [10]. Una aplicación de Mensajería Instantánea típica funciona de la siguiente forma: Un usuario se identifica e intenta conectar a través de un ordenador, para enviar un mensaje y así establecer una conversación de chat y comunicarse mediante el intercambio de mensajes. La Mensajería Instantánea tiene más espontaneidad que el correo electrónico o la sala de chat, incluso se puede navegar por la web mientras se intercambian los mensajes en tiempo real [11].

3.2.1 La historia y desarrollo de Mensajería Instantánea

Las aplicaciones de Mensajería Instantánea no tienen una larga historia de desarrollo. Una primera forma de mensajería instantánea fue la implementación en el sistema *PLATO* (Programmed Logic for Automated Teaching Operations) usado al principio de la década de 1970. Más tarde, el sistema *TALK* implementado en *UNIX/LINUX* comenzó a ser ampliamente usado por ingenieros y académicos en las décadas de los 80 y 90 para comunicarse a través de internet. En noviembre de 1996, la empresa *Mirabilis* [12] lanzó el primer sistema de mensajería instantánea para ordenadores con sistema operativo distinto a *UNIX/LINUX*, se llama *ICQ* [13] —en inglés “I see you”. Cuando apareció, fue muy popular entre los usuarios de internet, y estaba de moda en todo mundo. Después, la investigación sobre sistemas de Mensajería Instantánea entró en un periodo de rápido desarrollo. Desde 1997 hasta ahora, han aparecido un gran número de productos de Mensajería Instantánea en el mercado, los clientes de mensajería instantánea más utilizados son *Yahoo!Messenger* [14], *Windows Live Messenger* [15], *Google Talk* [16], *AIM(AOL Instant Messenger)* [17], *QQ* [18]. El sistema de Mensajería Instantánea *QQ*, se inventó en febrero de la 1999 en china, hasta ahora tiene registrados más que 530 millones de usuarios, entre ellos, hay más de 200 millones son usuarios activos, y el mayor récord de número de usuario en línea al mismo tiempo ya ha superado 1,95 millones. *Windows Live Messenger*, se basa en las ventajas estratégicas de una empresa como

Microsoft [19], vinculando este a su sistema operativo Windows, tiene aproximadamente 270 millones usuarios. Los sistemas de mensajería instantánea no solo sirven para intercambio de mensajes, han surgido muchas funciones nuevas. Los usuarios ya se pueden intercambiar fotos, ficheros, videos, se puede jugar o ver las películas con amigos a distancia, se puede hacer una llamada de voz o video, incluso se puede realiza la operación remota o videoconferencia. Es decir, los sistemas de Mensajería Instantánea ya funcionan como un terminal multimedia, en la vida cotidiana, tanto en el ocio como en el trabajo, desempeñan un papel importante cada vez más.

3.2.2 Funciones básicas de Mensajería Instantánea

Todos los sistemas de Mensajería Instantánea tienen dos funciones básicas:

1. Mostrar los usuarios conectados.
2. Intercambio de mensajes.

Como se indicó anteriormente, también tienen algunas funciones nuevas y avanzadas, tales como:

- **Gestion de contactos**

Los usuarios tienen una lista de contactos propia, a la cual pueden añadir o borrar los usuarios. Para añadir se tiene que enviar una solicitud explicando los motivos al otro usuario para obtener el permiso. También se puede editar la lista, para agrupar, borrar y crear grupos tales como: familia, compañeros, amigos, etc. El sistema muestra los estados de los usuarios de la lista: conectado, no conectado, y en el estado conectado incluye varios estados diferentes, como sin actividad, vuelvo enseguida, está ocupado. Cuando un usuario cambia de estado de no conectado a conectado, el sistema siempre manda un aviso. También en algunos sistemas avanzados se puede mostrar más informaciones de los usuarios de la lista. Por ejemplo, qué

-

canción está escuchando, qué juego está jugando, la novedad de su blog.

- Chat

Hay varios tipos de mensajes a enviar, depende de los estados de los usuarios.

Si un usuario en la lista de contactos tiene estado no conectado, se pueden enviar mensajes, estos se guardan en el servidor, y cuando el usuario se identifica, el servidor se los envía otra vez.

Si un usuario en la lista de contactos tiene estado conectado, los mensajes llegan en tiempo real. Cuando se envía un mensaje, el sistema genera una ventana de chateo, y todos los mensajes que son escritos se presentan en la ventana. Algunos sistemas soportan el envío de iconos o el pegado de fotos en la ventana.

- Otros

La mayoría de los clientes de mensajería instantánea, permiten el envío de archivos, generalmente usando protocolos punto a punto (P2P). Hay otra función más popular como es la llamada de voz o video.

3.2.3 Principales aplicaciones cliente

Una aplicación cliente es una versión instalable de un sistema de mensajería instantánea que en un principio sirve para que las personas puedan comunicarse de forma instantánea desde cualquier parte del mundo, pero que poco a poco las aplicaciones han ido creciendo y ofreciendo al usuario múltiples funciones y servicios adicionales que van desde la personalización del programa a envío de material multimedia pasando por videoconferencias y muchos otros servicios.

3.1.3.1 Principales características de las aplicaciones cliente

Existen múltiples ofertas en la web para ofrecernos éste servicio de forma gratuita e instantánea. Además también los propios programas nos sirven diferentes versiones y formatos para adaptarse a las necesidades del usuario y a las características del equipo informático del mismo, ya sea por el sistema operativo que use o por tener equipos antiguos con recursos pocos recursos de memoria.

A continuación vamos a presentar una relación con las características de los principales sistemas de mensajería instantánea en aplicación cliente.

❖ WINDOWS LIVE MESSENGER

Windows Live Messenger de microsoft fue pionero y es uno de los grandes referentes en cuanto a mensajería instantánea.

Comenzó en julio de 1999 con el nombre de MSN Messenger con un novedoso sistema de mensajería instantánea que permitía mantener conversaciones entre personas de todas partes del mundo de manera simple y eficaz y aunque desde el 13 de diciembre de 2005 cambió el nombre a Windows Live Messenger, y sus funcionalidades han aumentado, su esencia sigue siendo la misma [15].

Son muchas las novedades que incorporan las últimas versiones del programa, pero entre ellas, las mejoras que más destacan son: la posibilidad de realizar conferencias de voz e imagen, el sistema windows live call con el que permite llamar a teléfonos móviles y fijos, enviar mensajes a usuarios que no están conectados o posibilidad de conectarse con usuarios de: Hotmail, Microsoft y Yahoo!, entre otras con las que ha ganado una gran calidad haciendo su uso más atractivo.

Por otra parte en la opción de personalización, permite: agregar fotos o vídeos cortos del usuario, con posibilidad de cambiarlos según el

- estado de ánimo; mostrar a los demás usuarios lo que se está escuchando y personalizar la escena en la ventana de conversación.

Como novedad, Windows live messenger además de agrupar los contactos en conectados y no conectados o en diferentes grupos personalizados por el usuario, agrega un nuevo grupo llamado “favoritos” donde puedes meter a los contactos con los que más hables.

Y ofrece la posibilidad de compartir un álbum de fotos durante una conversación con sólo arrástralas. De manera interactiva permite agregar, descargar, suprimir fotos o verlas en modo presentación dentro del Windows Live Messenger.

En definitiva, Windows Live es una plataforma que agrupa una serie de servicios web, que en su mayoría, pretenden brindar al usuario una diversidad de alternativas de comunicación que van desde el correo electrónico, conversaciones en tiempo real, calendario, blogs, entre otros (Windows Live Hotmail, Windows Live Favorites, Windows Live OneCare, Windows Live Messenger, Windows Live Search, Windows Live Alerts, Windows Live Maps, etc). Todo ejecutable desde un navegador web, con el software y los datos del usuario almacenados en servidores de Microsoft.

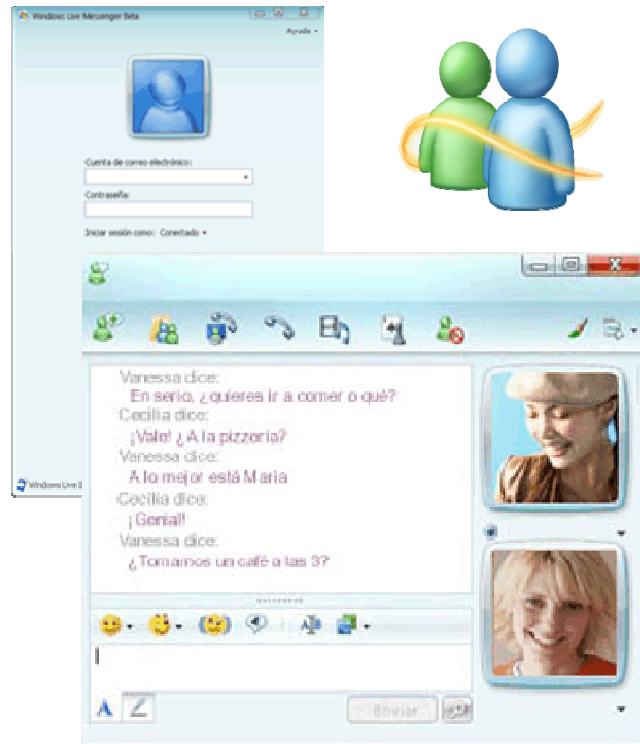


Figura 3.1 Imágenes y características de Windows Live Messenger

❖ Pidgin (ANTES GAIM)

En 2007 Gaim anunció los resultados de su acuerdo con AOL, que comportaban varios cambios: Gaim pasó a ser Pidgin, libgaim pasó a llamarse libpurple, y gaim-text se convirtió en Finch.

Pidgin es software libre, protegido por la GPL de GNU. Fue escrito originalmente por Mark Spencer para sistemas de tipo Unix, pero ahora funciona sobre muchas plataformas, incluyendo Microsoft Windows, GNU/Linux, SkyOS y el sistema de Trolltech para PDA Qt Extended. Para usuarios de Mac OS X la página de Pidgin recomienda Adium.

Pidgin fue dividido entre la interfaz gráfica de usuario (GUI), que se encarga de mostrar los datos al usuario, y el motor, que es el que implementa toda la funcionalidad de conexión con las redes de

mensajería. De esta forma, es posible escribir programas cliente usando diferentes interfaces.

Uno de sus puntos fuertes es que existe también una versión portátil de Pidgin que puede ser transportada y usada directamente desde una memoria USB sin necesidad de instalarse en el ordenador. Pero tiene una carencia bastante importante y es que no soporta chat multimedia [20].

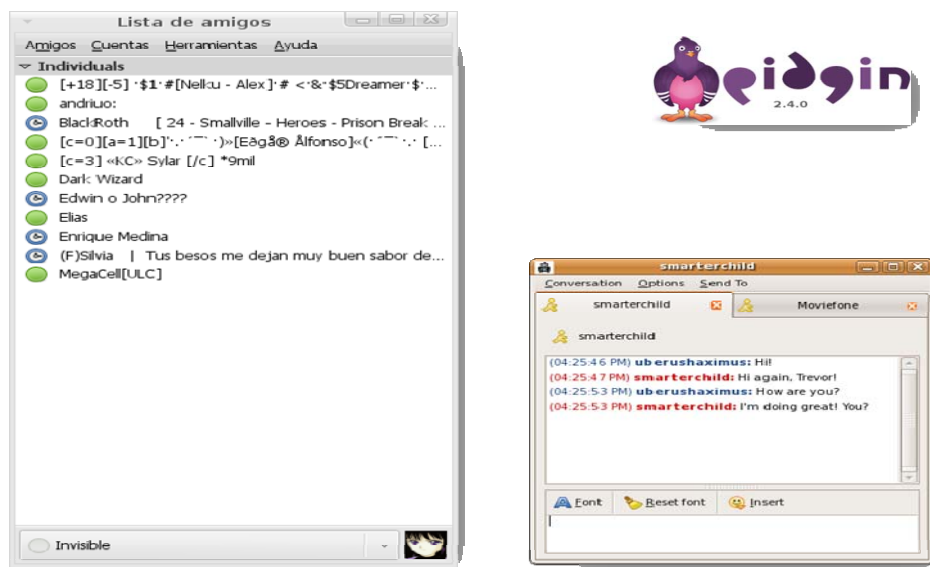


Figura 3.2 Imágenes y características de Pidgin

❖ Google Talk

Google Talk es un programa de mensajería instantánea y VoIP de protocolo Jabber, desarrollado por Google, que funciona bajo sistemas Windows. La versión beta de Google Talk fue lanzada el 24 de agosto de 2005.

Google Talk se sustenta bajo el protocolo de interoperabilidad de Jabber y XMPP, siendo configurable en programas como Psi, Miranda IM, iChat y Pidgin (anteriormente llamado Gaim), entre otros. Para que un cliente Jabber se pueda conectar necesita cifrado TLS y

autenticación SASL PLAIN a través del puerto 5222. El servicio está disponible para los usuarios de Gmail. Actualmente el registro es abierto, y se puede conseguir una cuenta entrando a Gmail.com.

El protocolo que Google ha desarrollado para la comunicación por voz, Jingle, ha sido liberado desde el primer momento, de tal forma que cualquier cliente lo puede incluir. PSi y Gaim planean incluirlo.

El programa que existe hoy en día es la versión beta, y se encuentra aún en etapas de desarrollo, sin presentar todas las características y funciones de la versión definitiva. Además de errores de probable aparición en las últimas características que puedan ser añadidas.

Una de las alternativas al Google Talk es el GMail Chat que usa la misma red pero este no necesita la instalación o descarga de ningún programa ya que el usuario habla a través del navegador mientras observa su correo. Este original webchat ha convencido a todos los usuarios y ya se está probando en otros correos como Yahoo! Mail y Windows Live Mail está barajando la posibilidad de añadir esta opción [16].



Figura 3.3 Imágenes y características de Google Talk

❖ ICQ

ICQ ("*I seek you*", en castellano *te busco*) es un cliente de mensajería instantánea y el primero de su tipo en ser ampliamente utilizado en Internet, mediante el cual es posible chatear y enviar mensajes instantáneos a otros usuarios conectados a la red de ICQ. También permite el envío de archivos, videoconferencias y charlas de voz [13].

ICQ fue creado por la empresa de software israelí Mirabilis a finales de los años 1990. El 8 de junio de 1998 la compañía fue adquirida por AOL por 407 millones de dolares. Hoy en día ICQ es usado por más de 38 millones de usuarios por todo el mundo. Según Time Warner, ICQ tiene más de 50 millones de cuentas registradas.

El protocolo de comunicaciones utilizado por ICQ es conocido como OSCAR, utilizado también por AIM. Los usuarios de la red ICQ son identificados con un número, el cual es asignado al momento de registrar un nuevo usuario, llamado UIN ("*Universal Internet Number*" o "número universal de Internet"). Debido al gran número de usuarios de ICQ, las identificaciones de usuario más recientes se encuentran por encima del número 100.000.000. En algunos casos, los números más simples y fáciles de recordar son vendidos en subastas por Internet o incluso secuestrados por otros usuarios.



Figura 3.4 Imágenes y características de Miranda IM

❖ SKYPE

Skype es un software para realizar llamadas sobre Internet (VoIP), fundada en 2003 por el sueco Niklas Zennström y el danés Janus Friis, creadores de Kazaa. El código y protocolo de Skype permanecen cerrados, pero los usuarios interesados pueden descargar gratuitamente la aplicación del sitio oficial. Los usuarios de Skype pueden hablar entre ellos gratuitamente [21].

La aplicación también incluye una característica denominada SkypeOut[] que permite a los usuarios llamar a teléfonos convencionales, cobrándoseles diversas tarifas según el país de destino. Otra opción que brinda Skype es SkypeIn, gracias al cual ellos otorgan un número de teléfono para que desde un aparato telefónico en cualquier parte del mundo puedan contactarte a tu

ordenador. Además, proveen de un servicio de buzón de voz gratuito. La interfaz de Skype es muy parecida a otro software de mensajería instantánea tales como MSN Messenger o Yahoo! Messenger, y de igual forma que en éstos es posible entablar una conversación de mensajes instantáneos con los usuarios del mismo software. Este programa también te permite enviar mensajes de voz en caso de que el usuario no se encuentre disponible.

Cuando se inicia una llamada se establece una conexión directa con la persona, eliminando así el consumo de ancho de banda utilizado por la voz en los servidores de Skype e incrementando la seguridad, al ser una conexión directa.

Y a parte irán apareciendo para esta versión plugins y widgets que harán más fácil su manejo.



Figura 3.5 Imágenes y características de Skype

❖ AOL (AIM)

AIM (America-On-Line Instant Messenger) es un programa de mensajería instantánea de America On Line denominada habitualmente Instant Messenger. La popularidad de la herramienta varía, y suele ubicarse en el tercer lugar de uso, después del Windows Live Messenger de Microsoft y del Yahoo! Messenger de Yahoo!.

Además de los servicios tradicionales de mensajería, también ofrece conexión a telefonía y chat con voz, grupos de usuarios filtrados por tus propios gustos dentro de la red de AOL, juegos en línea, noticias y contenido propio de AOL. A pesar de todo, no se considera tan vistoso u operativo como ICQ.

Tiene versiones para diversos sistemas operativos y son muy ligeros, para Windows 2 MB, Mac OS X 3 MB y Linux 741 KB.

Desde una única interfaz obtendrás acceso al sistema de mensajería instantánea de AOL, correo AOL, libreta de direcciones, explorador AOL y total acceso a entretenimiento multimedia en línea. AOL OpenRide está dividido en varias ventanas desde las que ofrece todas sus funciones siendo cómodo y fácil de usar en su totalidad [17].

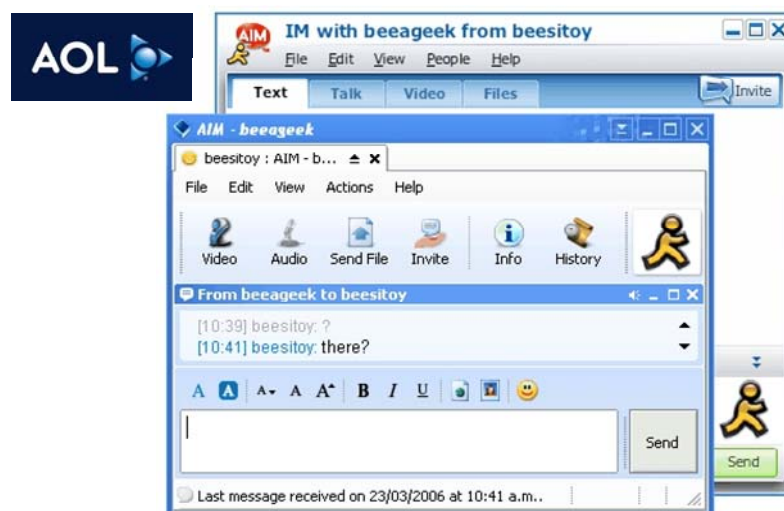


Figura 3.6 Imágenes y características de AOL

❖ YAHOO MESSENGER

Yahoo! Messenger, también conocido como Yahoo! Instant Messaging y abreviado como YIM, es una aplicación que permite el intercambio en tiempo real de mensajes entre dos o más usuarios en forma de texto [14].

Es uno de los sistemas de mensajería instantánea más comunes y populares por la facilidad de comunicación que permite entre dos personas sin importar la distancia entre las mismas, dado que el servicio es completamente gratuito para cualquier persona con acceso a Internet.

Hoy en día, al igual que otros programas en esta categoría, extiende sus capacidades mucho más allá de un simple intercambio de mensajes de texto en tiempo real, dos usuarios pueden intercambiar también archivos de cualquier tipo por medio del programa, escuchar música mientras conversan e incluso utilizar una cámara web para ver en tiempo real a la persona con quien se conversa, entre otras utilidades.

El mensajero de Yahoo! permite mantener una lista de contactos de las personas que se han agregado a ella, haciendo de esta manera una fácil visualización de quien se encuentra conectado al sistema de mensajería en determinado momento. Además de una conversación usuario-usuario, Yahoo! Messenger permite crear una sala de conversación comúnmente conocida como chat donde se mantendrá una conversación instantánea entre más de dos usuarios.

En su última versión, permite la llamada a teléfonos fijos y móviles de todo el mundo, con una calidad de voz muy similar a la de Skype.



Figura 3.7 Imágenes y características de Yahoo Messenger

3.2.3.2 Ventajas

- Intercambio de mensajes en tiempo real.
- Envío de mensajes offline.
- Posibilidad de intercambio de imágenes y archivos de todo tipo.
- Reproducción de música mientras se lleva a cabo la conversación.
- Soporte para cámara web.
- Videoconferencias.
- Posibilidad de compartir videos.
- Posibilidad de chatear por voz con un micrófono.
- Hacer llamadas internacionales gratis de PC a PC.
- Chat de conversación instantánea entre más de dos usuarios.
- Mensajes en desconexión.
- Conexión a telefonía.
- Soporte para grupos de usuarios.

-

- Juegos en línea.
- Permite la utilización de emoticonos.
- Posee interfaz con posibilidad de ampliarse sin deformarse.

3.2.3.3 *Inconvenientes*

- No tiene botones grandes ni accesibles.
- No tiene botones virtuales de acceso rápido para los símbolos que precisen la pulsación de dos teclas simultáneamente (@, €, !, ?).
- Carece de teclado virtual.
- No tiene diccionario ni frases predictivas.
- Carece de soporte icónico como apoyo a la palabra escrita en los botones de función.
- No permite configuración de formato y letra con plantillas especiales para las diferentes deficiencias visuales.
- Carece de soporte función como traducción automática.

3.3 **Arquitectura Orientada a Servicios (SOA)**

La Arquitectura Orientada a Servicios (SOA) [22] supone la utilización de servicios débilmente acoplados, y altamente inoperables para dar soporte a los requisitos de los procesos de negocio y los usuarios. En un entorno SOA, los recursos de la red están disponibles a través de servicios independientes, a los que se tiene acceso a través de métodos estándar sin necesidad de conocer su funcionamiento interno. Una definición de SOA propuesta por el consorcio OASIS es [23]:

“Un paradigma para organizar y utilizar capacidades distribuidas que pueden encontrarse bajo el control de diferentes dominios de propiedad. Proporciona un mecanismo uniforme para ofrecer, descubrir, interaccionar con y usar las

capacidades para producir los efectos deseados de forma consistente con precondiciones y expectativas medibles.”

SOA es una arquitectura de sistemas distribuidos que está caracterizada por [24]:

- Vista lógica. El servicio es una vista abstracta y lógica de programas, bases de datos, etc. definidas simplemente por su función, resultando transparente para el cliente el proceso que realiza.
- Orientado a mensajes. El servicio está definido como un intercambio de mensajes entre entidades suministradoras (servidores), y entidades solicitantes (clientes), omitiendo las propiedades internas entre sí.
- Orientado a la descripción. Los servicios se describen mediante el uso de meta-datos, que pueden ser procesados por el ordenador.
- Granularidad. Los mensajes suelen ser cortos y sencillos.
- Orientado a la red. Los servicios están orientados a su uso a través de redes de comunicaciones.
- Independiente de la plataforma. Los mensajes son enviados en formatos estándar, e independientes de la plataforma, como XML.

Normalmente se utiliza la tecnología SOA, cuando la aplicación a desarrollar cumple los siguientes requisitos [22]:

1. Funcionar sobre Internet, donde fiabilidad y velocidad no están garantizadas.
2. No existe la manera de gestionar al despliegue de la aplicación de manera que todos los proveedores y solicitantes sean mejorados de forma instantánea.
3. Donde los componentes del sistema distribuido, se ejecutan en diferentes plataformas, y sistemas propietarios.

4. Donde una aplicación existente necesite ser expuesta para su uso a través de una red de comunicaciones, y pueda ser encapsulada en un Servicio Web.

3.3.1 Los Servicios Web

Los Servicios Web [25] proporcionan un mecanismo estándar para que diferentes aplicaciones software que estén siendo ejecutadas en diferentes plataformas y/o marcos de trabajo puedan interoperar [24]. En [24] se propone la siguiente definición de Servicio Web, que coincide con la proporcionada por el glosario de la actividad “Web Service Activity” del W3C.

“Un servicio Web es un sistema software designado para soportar interacciones interoperables de máquina a máquina sobre una red de comunicaciones. Un Servicio Web tiene una interfaz descrita en un formato procesable por el ordenador (en particular WSDL). Otros sistemas puede interaccionar con el Servicio Web de la manera prescrita en su descripción usando mensajes SOAP, generalmente transmitidos mediante el uso del protocolo HTTP con una serialización XML en conjunción con otros estándares relacionados con la Web”.

Es importante señalar que la tecnología de los Servicios Web no es totalmente novedosa, ya que su origen se suele situar alrededor de los años 70, con el surgimiento de Internet, y de la computación distribuida.

En esta sección, se estudian los antecedentes de los actuales Servicios Web, y los distintos componentes que forman la arquitectura actual de los mismos, también se realizará un repaso por los fundamentos de la Arquitectura Orientada a Servicios (SOA), terminando con una descripción de las ventajas y los inconvenientes de esta tecnología.

3.3.2 Cronología

El origen de los Servicios Web se asocia, con el inicio de dos tecnologías [26]:

- Minicomputadores (Estaciones de trabajo, y ordenadores personales).
- Redes de ordenadores (Ethernet e Internet).

Un sistema distribuido está compuesto por entidades software heterogéneas y discretas que deben trabajar de forma conjunta para llevar a cabo tareas [24]. Las entidades software en un sistema distribuido no operan en el mismo entorno de procesamiento, por lo que deben comunicarse a través de protocolos, tanto software como hardware, a través de una red de comunicaciones. Existen muchas tecnologías que permiten la construcción de sistemas distribuidos, las más importantes y que más han influido en la elaboración de Servicios Web son: RPC, DCOM, RMI y CORBA [27].

Llamada a Procedimiento Remoto (RPC). Este protocolo permite a un programa ejecutar una subrutina o procedimiento en una máquina remota sin necesidad, por parte del programador, de codificar los detalles de interacción remota [28].

Modelo de Objetos de Componentes Distribuidos (DCOM). Esta tecnología es propietaria de Microsoft para desarrollar componentes distribuidos sobre varios ordenadores y que se comuniquen entre sí [29].

Invocación de Métodos Remotos (RMI). Es un mecanismo ofrecido en Java para invocar un método remotamente, su arquitectura se puede ver como un modelo de cuatro capas; una primera capa de aplicación (aplicaciones cliente-servidor), una segunda capa Proxy, que permite a los desarrolladores elaborar aplicaciones sin tener en cuenta los detalles referentes a la interacción, una tercera capa de referencia

remota, y una cuarta capa de transporte, que se encarga de realizar las conexiones necesarias, y de la gestión del transporte de los datos [30].

Un grave problema de las arquitecturas anteriores RMI (Sun Microsystems), DCOM (Microsoft), etc. es que se basan en estándares propietarios, de manera que la interoperabilidad entre ellas se reduce al máximo.

CORBA es una arquitectura estándar que establece una plataforma de desarrollo de sistemas distribuidos, facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos. Esta arquitectura fue definida, y es controlada por el Object Management Group (OMG), que es un consorcio formado por las empresas más relevantes del sector. A pesar de iniciativas como ésta, hay que destacar que no se han conseguido los niveles de interoperabilidad deseado [31].

Después de lo dicho anteriormente, se demostró que para desarrollar estándares aceptados globalmente, era necesario establecer un conjunto mínimo de elementos comunes. El protocolo HTTP, y el lenguaje XML han establecido este conjunto mínimo tecnológico para el éxito de la computación distribuida, constituyendo la base de los Servicios Web [32].

3.3.3 Componentes de la arquitectura

Como se señaló en el apartado anterior, el problema de la interoperabilidad entre sistemas suponía un problema muy importante, y su solución radicaba en encontrar un conjunto de estándares que fueran aceptados de manera general para así poder construir una tecnología más acorde a la visión de Internet, donde XML y HTTP constituyeron esa base.

La tecnología de los Servicios Web se fundamenta en tres estándares principalmente [33]:

- El Web Services Description Language (WSDL – Lenguaje de Descripción de Servicios Web). Es un lenguaje basado en XML que permite describir las características de los Servicios Web [34].
- El Simple Object Access Protocol (SOAP, Protocolo Simple de Acceso a Objetos). Es un protocolo estándar que define como dos objetos localizados remotamente pueden comunicarse por medio de intercambio de datos XML, y que funciona sobre HTTP [35].
- El Universal Description, Discovery, and Integration (UDDI, Descripción, Descubrimiento e Integración Universales). Es un registro que proporciona mecanismos estándares para publicar documentos WSDL que contienen descripciones de Servicios Web, y que permite realizar búsquedas sobre los mismos [36].

3.3.4 Funcionamiento de los Servicios Web

A continuación se va describir el mecanismo de funcionamiento de los Servicios Web, que se puede desglosar en los siguientes pasos principalmente:

1º Paso. Es necesario que la organización que desee ofrecer un servicio, añada una descripción del mismo (archivo WSDL) en un registro UDDI, es decir, que registre el servicio para su posterior uso.

2º Paso. Una vez que la descripción del servicio se encuentra almacenada en un registro UDDI accesible por Internet, los clientes que utilizan los servicios deberán de buscar aquellos que les resulten necesarios para alcanzar su objetivo (este proceso lo realiza el propio registro UDDI, que recibirá como entrada la funcionalidad esperada del servicio, y devolverá la lista de los servicios que cumplan esa funcionalidad, archivos WSDL).

3º Paso. Una vez recibidos los archivos WSDL con todos los servicios posibles, el cliente debe determinar cuál de ellos es el más apropiado.

4° Paso. Cuando el cliente ha seleccionado el archivo más apropiado a ejecutar, realiza la petición al proveedor de servicio a través de un mensaje SOAP.

5° Paso. El servicio es ejecutado, utilizando los parámetros indicados en el mensaje SOAP, y el resultado de la ejecución se devuelve al cliente en forma de otro mensaje SOAP, terminando así el proceso de invocación de un Servicio Web.

3.3.5 Extensible Markup Language (XML)

De acuerdo con [37] Markup se define como “sintaxis extra textual que puede ser usada para describir formatos, acciones, estructuras, atributos, etc.”. Un ejemplo de Markup pueden ser los formatos de los comandos del popular procesador de texto LaTeX [38].

A finales de los años 70 fue definido Standard Generalized Markup Language (SGML) [39], que es un meta-lenguaje para etiquetado de texto, desarrollado por Charles F. Goldfarb y su grupo, y basado en un trabajo previo desarrollado en IBM. En 1996, el SGML Editorial Review Board se convirtió en el Grupo de Trabajo de XML, bajo el control del Consorcio del World Wide Web (W3C). Este grupo de trabajo desarrollo el XML [40], como un subconjunto de SGML, cuyo principal objetivo es permitir un SGML genérico [41].

XML, al igual que SGML, no son exactamente lenguajes de etiquetado, sino que son lenguajes que pueden ser utilizados para definir lenguajes de etiquetado o marcado específicos, como por ejemplo XHTML [42], MathML [43], es decir, XML permite a los usuarios definir nuevas etiquetas y estructuras para sus propios lenguajes.

En una sociedad interconectada y global, el intercambio de datos a través de una sintaxis estándar se ha convertido en una cuestión fundamental, y aquí es donde encaja perfectamente XML.

3.3.6 XML Schema

XMLSchema es la síntesis o norma de esquema basado en XML. XML Schema sirve para definir y describir la estructura y modelo de contenido de XML documentos. Puede definir las relaciones entre los elementos de documentos XML, también se puede definir tipos de datos y atributos. XML Schema es un documento XML, corresponde a la estructura de XML síntesis y que puede analizarlo por XML analizador general [44].

XML Schema tiene las ventajas siguientes:

- XML Schema es basado en XML, no tiene la síntesis especial.
- Se puede analizar y corregir al igual que los documentos XML.
- XML Schema soporta un serie de tipos de datos (int, float, boolean, date y etc.).
- XML Schema permite ampliar los tipos de datos.
- XML Schema soporta los espacios de nombres integrados.
- XML Schema soporta tipo atributo.

3.3.7 Web Services Description Language (WSDL)

El lenguaje de descripción de servicios proporciona un modelo y un formato XML para describir Servicios Web [45]. WSDL permite separar la descripción de la funcionalidad abstracta ofrecida por el servicio, de los detalles más concretos relativos al “como” y “donde” obtener dicha funcionalidad.

Los mensajes están descritos independientemente del formato utilizado, empleando un sistema de tipos como XMLSchema. En la (Figura 3.8) se muestra un ejemplo de un archivo WSDL.

```
<?xml version="1.0"?>  
<definitions name="StockQuote"  
targetNamespace="http://example.com/stockquote.wsdl"
```

```

xmlns:tns="http://example.com/stockquote.wsdl"
xmlns:xsd1="http://example.com/stockquote.xsd"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
<types>
  <schema targetNamespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="TradePriceRequest">
      <complexType>
        <all>
          <element name="tickerSymbol" type="string"/>
        </all>
      </complexType>
    </element>
    <element name="TradePrice">
      <complexType>
        <all>
          <element name="price" type="float"/>
        </all>
      </complexType>
    </element>
  </schema>
</types>
<message name="GetLastTradePriceInput">
  <part name="body" element="xsd1:TradePriceRequest"/>
</message>
<message name="GetLastTradePriceOutput">
  <part name="body" element="xsd1:TradePrice"/>
</message>
<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>
<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
</service>

```

</definitions>

Figura 3.8 Ejemplo de un archivo WSDL

En la (Figura 3.9) se muestra un ejemplo de un archivo XMLSchema [46].

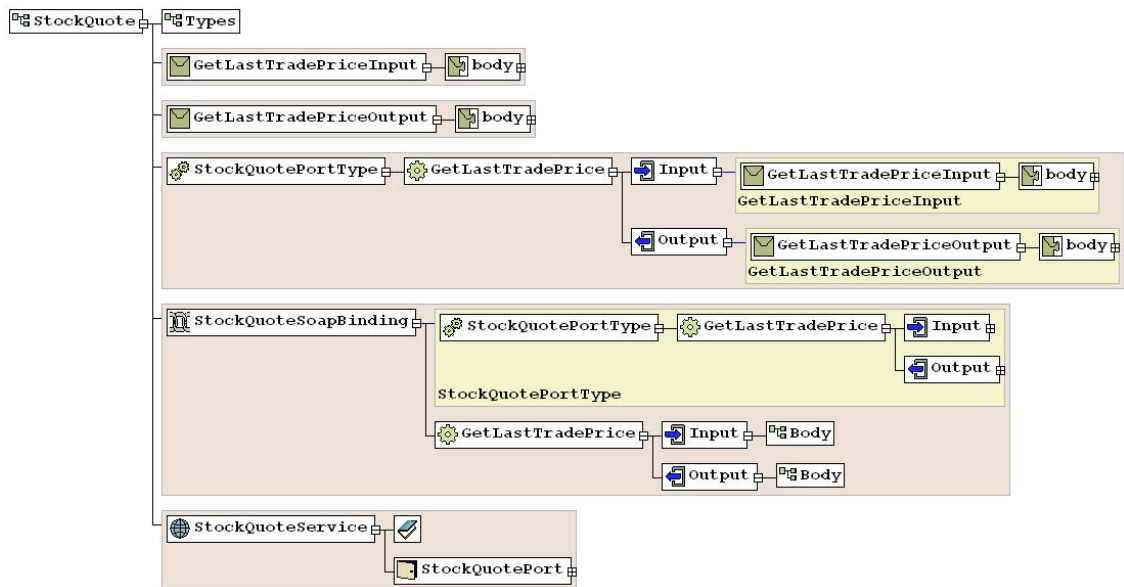


Figura 3.9 Representación gráfica del XMLSchema de un archivo WSDL

3.3.8 Universal Description, Discovery and Integration (UDDI)

Como se señaló anteriormente, UDDI es un protocolo que constituye uno de los bloques fundamentales de la arquitectura de Servicios Web [47]. UDDI proporciona una plataforma estándar e interoperable que permite a las compañías y a las aplicaciones, encontrar y utilizar Servicios Web a través de Internet. UDDI, es una iniciativa de OASIS¹ (Organization for the Advancement of Structured Information Standards).

¹ OASIS – Es un consorcio internacional formado por las organizaciones más importantes a nivel mundial, que está orientado al desarrollo y la adopción de estándares para negocio electrónico y Servicios Web.

La información que forma parte de un registro UDDI está compuesta por cuatro tipos de estructuras de datos; la entidad de negocio (businessEntity), el servicio de negocio (businessService), la plantilla de vinculación (bindingTemplate), y el modelo (tModel). En la (Figura 3.10) se puede ver un ejemplo de la especificación de la estructura de datos de una entidad de negocio (businessEntity) [48].

```
<element name="businessEntity" type="uddi:businessEntity" />
<complexType name="businessEntity">
  <sequence>
    <element ref="uddi:discoveryURLs" minOccurs="0" />
    <element ref="uddi:name" maxOccurs="unbounded" />
    <element ref="uddi:description" minOccurs="0" maxOccurs="unbounded" />
    <element ref="uddi:contacts" minOccurs="0" />
    <element ref="uddi:businessServices" minOccurs="0" />
    <element ref="uddi:identifierBag" minOccurs="0" />
    <element ref="uddi:categoryBag" minOccurs="0" />
  </sequence>
  <attribute name="businessKey" type="uddi:businessKey" use="required" />
  <attribute name="operator" type="string" use="optional" />
  <attribute name="authorizedName" type="string" use="optional" />
</complexType>
```

Figura 3.10 Ejemplo de un archivo UDDI

Los registros UDDI se basan fundamentalmente en estándares como HTML y XML, y tienen dos funciones principalmente; primero facilitar la descripción de Servicios Web de forma que sean suficientemente expresivos para ser útiles en el proceso de búsqueda, y segundo suministrar las herramientas necesarias para hacer las descripciones lo suficientemente completas para que tanto las personas, como las máquinas puedan interpretar el funcionamiento de los servicios, y por tanto interactuar con ellos [49].

3.3.9 SOAP

SOAP es una especificación de protocolo para la prestación de servicios utilizando XML sobre HTTP para codificar y transportar tanto la petición como la respuesta. [50] . SOAP utiliza XML, para definir el marco de intercambio de mensajes extensible para así poder

trabajar sobre múltiples protocolos, por lo que es independiente de cualquier modelo de programación, o semántica específica de implementación [51].

En la (Figura 3.11), y en la (Figura 3.12) se pueden observar dos esquemas que muestran, y detallan la arquitectura de los Servicios Web.

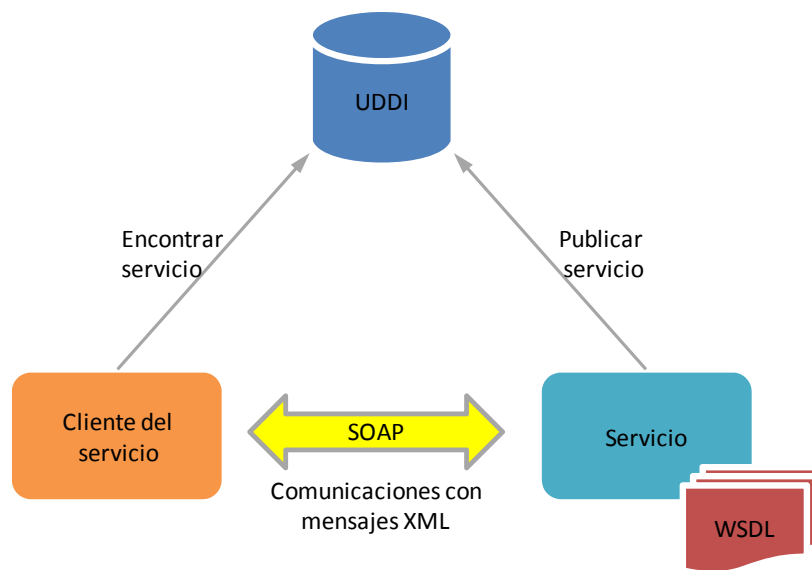


Figura 3.11 Arquitectura de Servicios Web

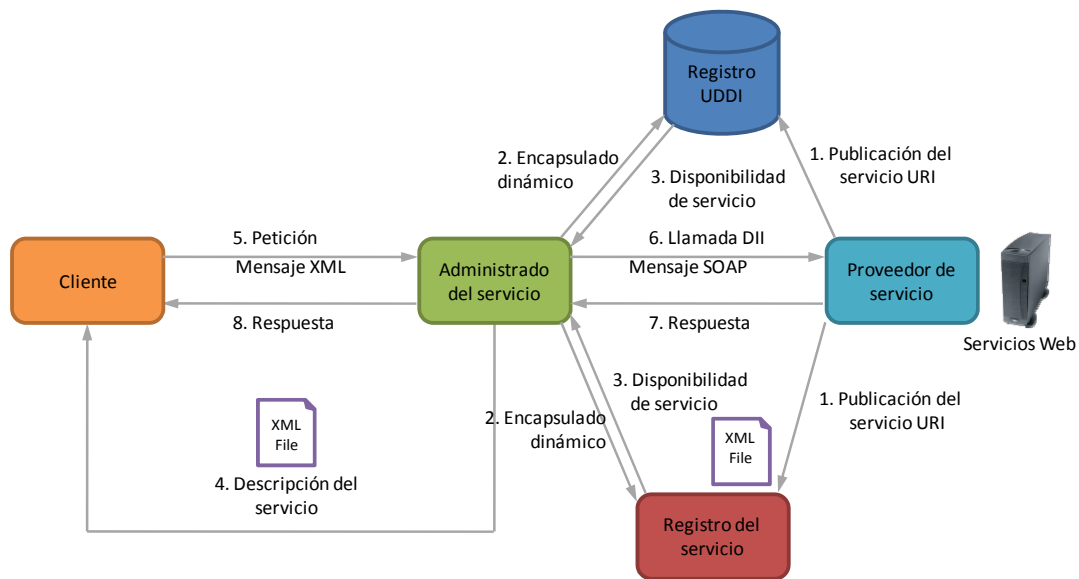


Figura 3.12 Arquitectura detallada de Servicios Web

Los mensajes SOAP consisten fundamentalmente, en la transmisión entre emisores, y receptores SOAP, por lo que los mensajes resultan bastante simples, aunque se pueden crear patrones de interacción más complejos. Los mensajes SOAP están compuestos por un elemento contenedor “*envelope*”, que además del nombre, y los atributos contiene de forma opcional un elemento “*header*” (cabecera del mensaje), y obligatoriamente un elemento “*body*” (cuerpo del mensaje) [52].

En la (Figura 3.13) se muestra un ejemplo sencillo de una reserva de un viaje expresado en un mensaje SOAP [53].

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
<env:Header>
<m:reserva xmlns:m="http://empresaviajes.example.org/reserva"
env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
env:mustUnderstand="true">
<m:referencia>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</m:referencia>
<m:fechaYHora>2001-11-29T13:20:00.000-05:00</m:fechaYHora>
</m:reserva>
<n:pasajero xmlns:n="http://miempresa.example.com/empleados"
env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
```

```

    env:mustUnderstand="true">
      <n:nombre>Felipe</n:nombre>
      </n:pasajero>
    </env:Header>
    <env:Body>
      <p:itinerario
        xmlns:p="http://empresaviajes.example.org/reserva/viaje">
        <p:ida>
          <p:salida>Nueva York</p:salida>
          <p:llegada>Los Ángeles</p:llegada>
          <p:fechaSalida>2001-12-14</p:fechaLlegada>
          <p:horaSalida>Última hora de la tarde</p:horaSalida>
          <p:preferenciaAsiento>pasillo</p:preferenciaAsiento>
        </p:ida>
        <p:vuelta>
          <p:salida>Los Ángeles</p:salida>
          <p:llegada>Nueva York</p:llegada>
          <p:fechaSalida>2001-12-20</p:fechaSalida>
          <p:horaSalida>media-mañana</p:horaSalida>
          <p:preferenciaAsiento/>
        </p:vuelta>
      </p:itinerario>
      <q:alojamiento
        xmlns:q="http://empresaviajes.example.org/reserva/hoteles">
        <q:preferencia>ninguna</q:preferencia>
      </q:alojamiento>
    </env:Body>
  </env:Envelope>

```

Figura 3.13 Ejemplo de un mensaje SOAP

3.4 Conclusiones

Las principales ventajas de las arquitecturas orientadas a Servicios Web están relacionadas con la utilización de protocolos de transporte estándar, como HTTP, que permite el intercambio de mensajes sin necesidad de preocuparse por los sistemas de seguridad, como los cortafuegos, etc. y que facilita la interoperabilidad entre plataformas, aunque esto se está convirtiendo también en una desventaja debido a la existencia de ataques informáticos basados en Servicios Web, y por tanto a la necesidad de comprobar la integridad de los archivos XML.

Algunas de las desventajas de los Servicios Web tienen que ver con el rendimiento de los sistemas SOA, comparándolos con otros modelos de computación distribuida como RMI, DCOM o CORBA que

-
obtienen unos rendimientos superiores, esto es debido fundamentalmente al uso de XML en la codificación de los mensajes, ya que es necesario procesar y comprobar la integridad de los mismos. En la actualidad, las aplicaciones de mensajería instantánea tienen sus funciones cada vez más avanzadas, pero todavía no hay ninguna que soporte la función de traducción automática, es decir, la capacidad de entendimiento entre dos usuarios que no tienen un lenguaje común. Ahora que las comunicaciones entre diferentes países o culturas son cada día más importante.

La aplicación de mensajería instantánea presentada en este trabajo utiliza una arquitectura SOA con el fin de poder implementar las estructuras de VILA_1 para realizar óptimamente el proceso de traducción-comunicación.

CAPÍTULO 4

Diseño de la Aplicación de intercambio de mensajes con VILA_1



Este capítulo es la especificación de un diseño de un sistema de mensajería instantánea que permita editar y utilizar el lenguaje `VIL_A_1` en Internet. Y también introduce como usar el sistema.



4.1 Introducción

En esta tesis doctoral, se diseña y define una aplicación de mensajería instantánea basada en el lenguaje VILA_1. Esta aplicación tiene las funciones comunes similares a las que existen en el mercado. Además realiza la comunicación entre dos usuarios que tienen idiomas diferentes. Es decir, esta aplicación tiene una función de traducción automática. Como se mencionó anteriormente, esta aplicación está basada en el lenguaje VILA_1, que aporta un novedoso método de traducción basado en VILA_1. VILA_1, utiliza una sintaxis visual común a los diferentes idiomas. A través de esta forma visual, tiene las siguientes ventajas:

1. Fácil y cómodo. En este sistema de mensajería instantánea se aplica un grupo de símbolos para enlazar los elementos de lenguaje natural. Los usuarios pueden leer los mensajes de una forma más visual, para editar y entender mensajes fácilmente.
2. Puede esquivar los problemas de la gramática del lenguaje natural. Los símbolos ayudan a traducir los mensajes de elemento a elemento, no hace falta pensar mucho en la gramática de lenguaje natural.

Así para realizar este sistema de mensajería instantánea, Primero hay que saber qué significa el sistema, y luego sobre los elementos del sistema a analizar y diseñar.

4.2 Análisis y diseño del sistema

El diseño de este sistema de mensajería instantánea es igual que los principales aplicaciones de mensajería instantánea, pero tiene algunas funciones especiales para soportar la aplicación de traducción los mensajes por VILA_1.

4.2.1 Diseño total del sistema de mensajería instantánea

Este sistema de mensajería instantánea aplica el diseño de modelo C/S (Cliente y Servidor), en una estructura que tiene 3 capas (*Figura 4.1*):

- El servidor de la base de datos
- El terminal servidor de la aplicación
- El terminal cliente de aplicación

El sistema utiliza la estructura C/S, así se pueden distribuir los trabajos entre terminal servidor y el terminal cliente, entonces se pueden disminuir las comunicaciones entre las 3 capas en el sistema, así el sistema funciona más rápido y obtiene una buena estabilidad.

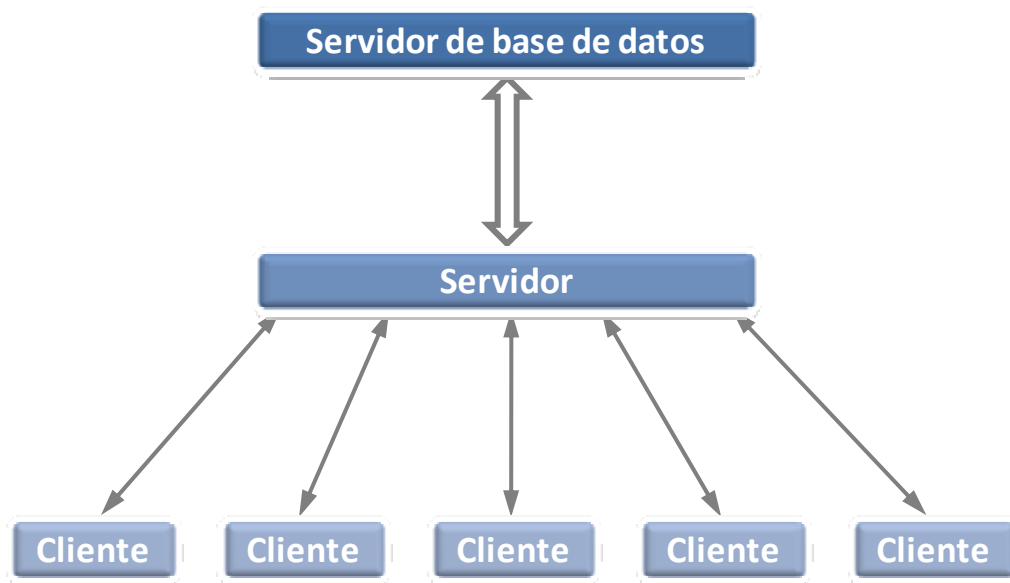


Figura 4.1 Estructura de C/S de 3 capas

El sistema de mensajería instantánea utiliza una especificación de protocolo de tipo SOAP, utilizando documentos XML (archivos VILA_1), sobre HTTP, su funcionamiento se puede resumir en el siguiente procedimiento:

1. El usuario debe de estar dentro de la aplicación, a la que accede a través de su “dirección de correo electrónico”, y de su

- “contraseña”, una vez dentro, haciendo clic en el icono de “chat” se iniciará sesión en el servicio de mensajería instantánea.
2. El terminal servidor enviará al usuario una lista con todos los usuarios que se encuentran conectados al servicio de mensajería instantánea de VILA_1 en ese momento.
3. El usuario solicitará la apertura de una sesión de chat haciendo doble clic en un contacto de la lista, en ese momento se enviará a dicho contacto una solicitud de comunicación.
4. Si el destinatario acepta la solicitud de comunicación, el terminal servidor VILA_1 establecerá una conexión entre ambos usuarios, permitiendo a los mismos el intercambio de información utilizando el lenguaje VILA_1.
5. Toda la información que intercambien ambos usuarios, será filtrada a través de un firewall (cortafuegos) localizado en el terminal servidor VILA_1, para comprobar la validez, y la consistencia de los datos enviados. Si los datos enviados son válidos el mensaje llegará al destinatario.
6. Si el mensaje llega al destinatario se envía un mensaje de recibido. El terminal servidor VILA_1 es el encargado de mantener la información, y el estado de los participantes constantemente actualizada.

En la (Figura 4.2) se muestra la estructura de funcionamiento del sistema de mensajería instantánea.

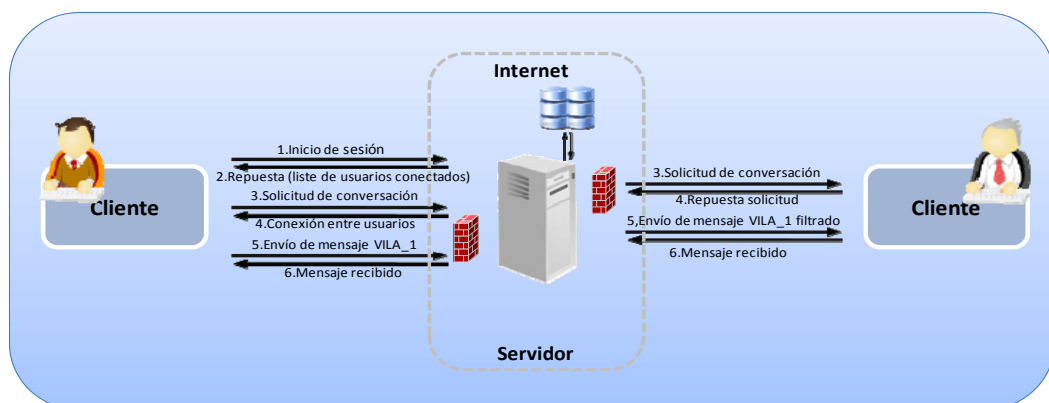


Figura 4.2 Estructura de funcionamiento del sistema de mensajería instantánea

4.2.2 La definición de los módulos funcionales

Este sistema se puede dividir en tres partes sobre los requisitos de los usuarios a analizar, incluyen las funciones del terminal servidor, el terminal cliente y la base de datos, como la figura siguiente: (*Figura 4.3*):

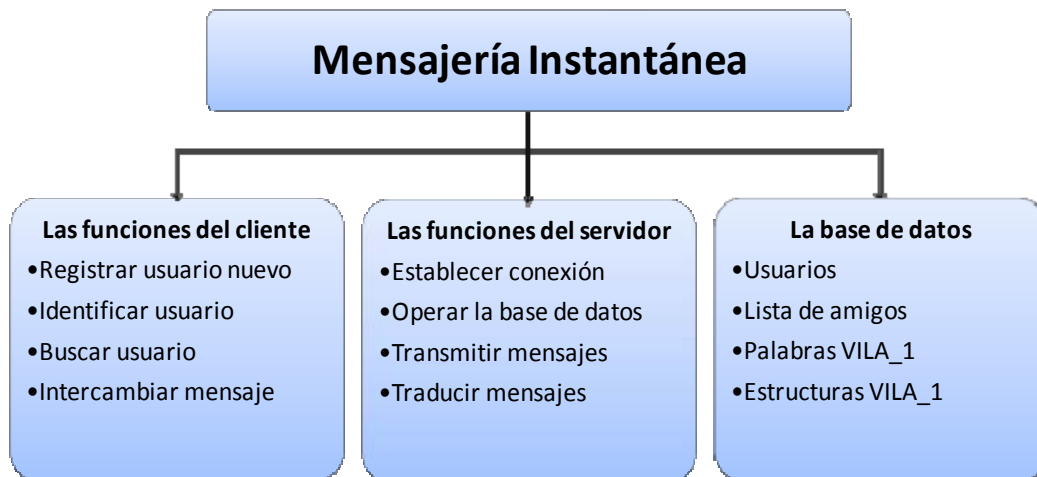


Figura 4.3 Los Módulos funcionales

➤ El terminal cliente

El terminal cliente a través de establecer una comunicación con el terminal servidor, envía las informaciones del usuario al terminal servidor y recibe las informaciones del terminal servidor para realizar los registros de usuarios nuevos.

Cuando el usuario quiere identificarse, envía el nombre de usuario y contraseña al servidor, el terminal servidor las verifica y realiza la identificación.

La función de buscar usuarios es una función básica e importante, que se realiza en el cliente. Los usuarios a través de esta función buscan a otros usuarios para añadirlos a su lista de amigos. Además, los usuarios pueden ver la información pública de otros usuarios.

Otra función básica, es el intercambio de mensajes, donde los usuarios pueden editar, enviar y recibir los mensajes por el terminal cliente.

➤ El terminal servidor

El servidor siempre está escuchando, por sí el cliente se conecta o desconecta con el servidor.

El terminal servidor también es un centro de tratamiento de las informaciones. Todas las informaciones de los terminales clientes se tienen que enviar al terminal servidor, y el servidor según diferentes requisitos las envía a sus destinos.

Las operaciones de la base de datos incluyen registrar y modificar las informaciones de usuario, buscar y añadir las informaciones de la lista de amigos.

La función de traducir mensajes, según las demandas de usuarios, traduce los mensajes de idioma original al idioma de destino, y los envía.

➤ La base de datos

En este sistema de mensajería instantánea necesita cuatro bases de datos.

La base de datos de usuarios sirve para guardar las informaciones de registro de usuarios

La base de datos de lista de amigos sirve para guardar las informaciones de los amigos de usuarios.

La base de datos de palabras VILA_1 sirve para guardar las palabras y sus traducciones.

La base de datos de estructuras VILA_1 sirve para guardar las estructuras VILA_1.

4.3 El diseño del terminal cliente

Ya que el objetivo de esta tesis doctoral es definir las especificaciones de cómo el lenguaje VILA_1 realiza los intercambios de mensajes entre diferentes idiomas, por eso destaca el diseño y uso de interfaz de la aplicación de mensajería instantánea.

4.3.1 El diseño de la interfaz de inicio sesión

La interfaz de inicio sesión del sistema se presenta abajo (*Figura 4.4*). El usuario nuevo tiene que registrarse. Los datos de usuario para registrar como mínimo tienen que incluir dos cosas: la dirección de correo electrónico y la contraseña, los usuarios pueden llenar los datos opcionales o no. Después del registro, ya se puede chatear por este sistema. Antes de iniciar sesión, los usuarios pueden elegir cómo recordar la cuenta, recordar la contraseña o iniciar la sesión automáticamente. También se pueden elegir los estados de usuarios, por ejemplo disponible, ocupado, ausente, desconectado. El sistema recuerda las operaciones, para facilitar el uso del sistema la próxima vez.

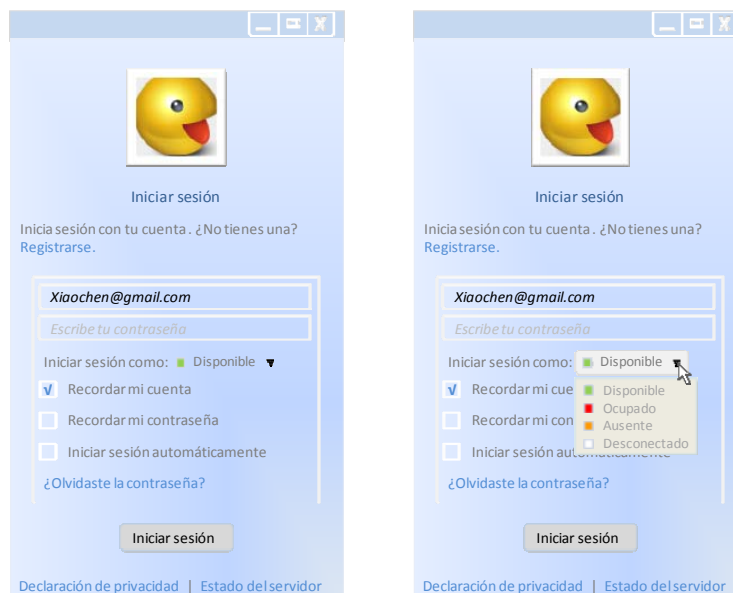


Figura 4.4 Interfaz de inicio sesión

4.3.2 El diseño de la interfaz principal del sistema

La interfaz principal (*Figura 4.5*) sirve para buscar otros usuarios, para realizar la conversación entre los usuarios. Se pueden dividir en varios grupos los requisitos de usuarios, como favoritos, familias, compañeros, compañeros, amigos, y otros contactos. Hay simbolo con distintos colores para distinguir sus diferentes estados, esta a frente a los nombres de usuarios. Si quiere chatear con alguien, solo necesita pinchar su nombre de usuario y sale la ventana de chatear.

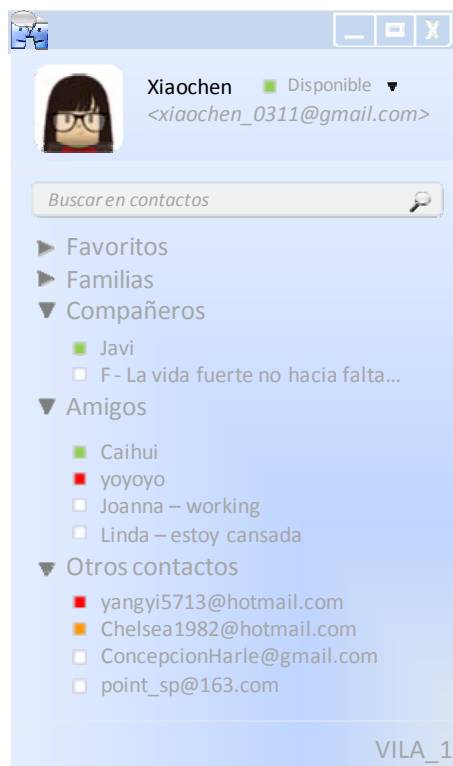


Figura 4.5 Interfaz principal del sistema

4.3.3 El diseño de la interfaz de chatea

La interfaz de chatea está formado por 5 secciones, como se puede observar en la (*Figura 4.6*).

- La parte referenciada con número 1, es la barra de tareas de la aplicación, esta incluye los botones de funciones básicas del

sistema de mensajería instantánea.

- La parte referenciada con número 2, es el campo de los mensajes. En este campo se presentan todos los mensajes de chat que se intercambian por los usuarios.
- La parte referenciada con número 3, es la barra de tipo de mensajes. Los usuarios pueden elegir los modos de mensajes según diferentes tipos.
- La parte referenciada con número 4, es el campo de edición de mensajes.
- La parte referenciada con número 5, es la barra de información de usuarios. Esta barra contiene las informaciones de los usuarios que están charlando.



Figura 4.6 Interfaz de chat

En la interfaz de chat, cuando se pincha en las imágenes de los usuarios, sale una ventana con los datos del usuario, se pueden ver los datos de usuarios amigos, también se pueden editar y modificar los datos de nuestro usuario.



Figura 4.7 Ventana de los datos del usuario

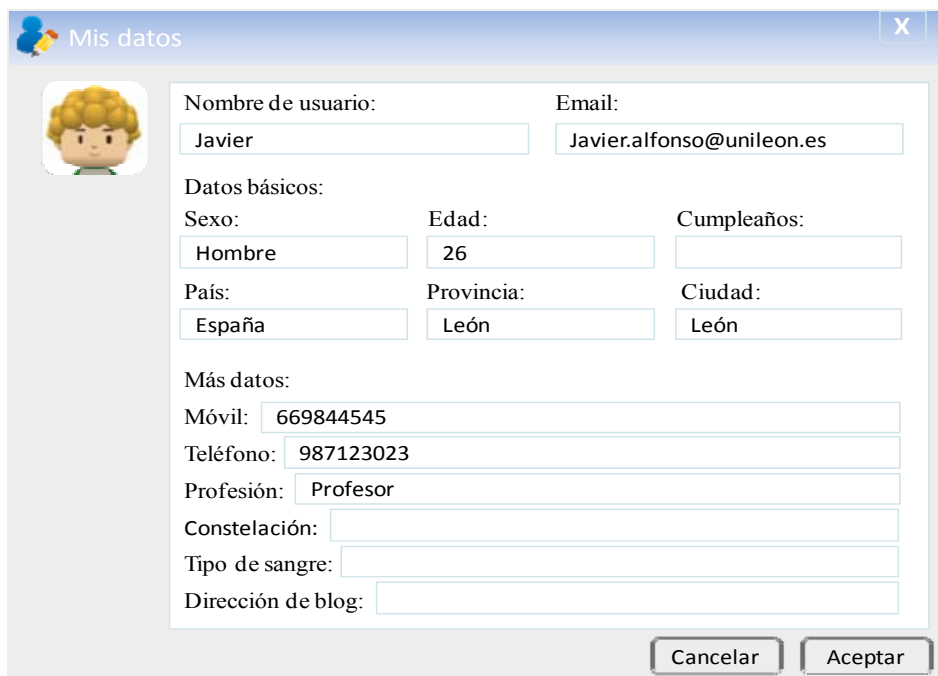


Figura 4.8 Ventana de editar los datos del usuario

En la barra de tareas de la aplicación, hay dos botones, son “Idioma de origen” y “Idioma de destino”, al pinchar en las flechas desplegables aparecen dos menús para elegir (*Figura 4.9*):

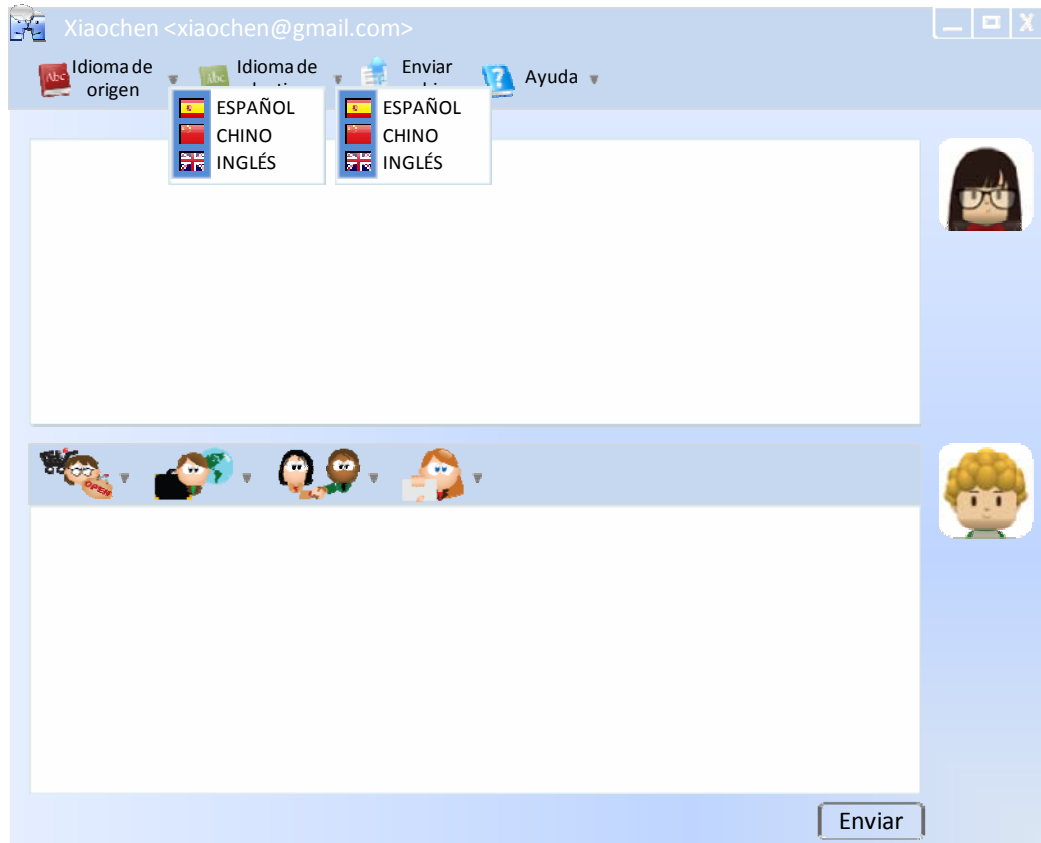


Figura 4.9 Captura de pantalla de la interfaz de seleccionar idioma.

Los menús tienen tres idiomas, son español, chino e inglés. Según la encuesta de *Miniwatts Marketing Group* [54], estos tres idiomas son los primeros tres idiomas más utilizados en Internet (*Figura 4.9*), y sin duda esta aplicación de mensajería instantánea es una aplicación de Internet, por eso se eligen estos tres idiomas, ya que incluye a la mayoría.

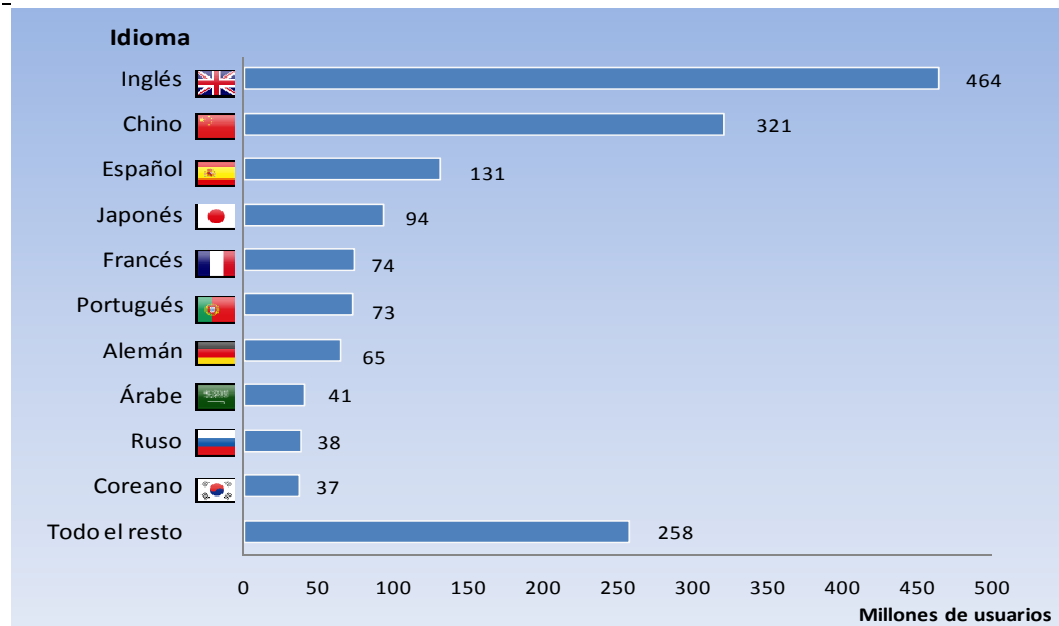


Figura 4.10 Los diez idiomas más utilizados en el Internet

Aunque ahora sólo se tienen 3 idiomas, que incluyen al 57.4% de los usuarios del Internet en todo el mundo (*Tabla 4.1*), en futuro añadir a cualquier idioma al sistema, sí es necesario, lo único tiene que hacer es añadir los vocabularios a la base de datos.

Los tres idiomas más utilizados en la Web					
Los tres idiomas más utilizados en la Web	% de todo los usuarios de Internet	Los usuarios de Internet por idioma	La penetración de Internet por idioma	El idioma en el crecimiento de Internet (2000-2008)	2008 Estima la población mundial de Internet
Inglés	29.1%	463,790,410	37.2%	226.7%	1,247,862,351
Chino	20.1%	321,361,613	23.5%	894.8%	1,365,138,028
Español	8.2%	130,775,144	32.0%	619.3%	408,760,807
Los tres idiomas más utilizados	57.4%	915,927,167	30.3%	375.9%	3,021,761,186
Resto de las lenguas	42.6%	680,342,941	18.4%	303.7%	3,688,267,884
Total mundial	100.0%	1,596,270,108	23.8%	342.2%	6,710,029,070

Tabla 4.1 Los tres idiomas más utilizados en la web

Debido a que VILA_1 es un lenguaje semiformal y visual, utiliza muchos símbolos especiales, estos símbolos tienen su significado, por eso en la barra de tareas de la aplicación hay un botón “Ayuda”. Al pinchar en él aparece una lista de los símbolos con sus significados, ayudando a los usuarios a conocer y entender bien cuando se están leyendo o editando los mensajes, así se puede utilizar este sistema más fácilmente (*Figura 4.11*).

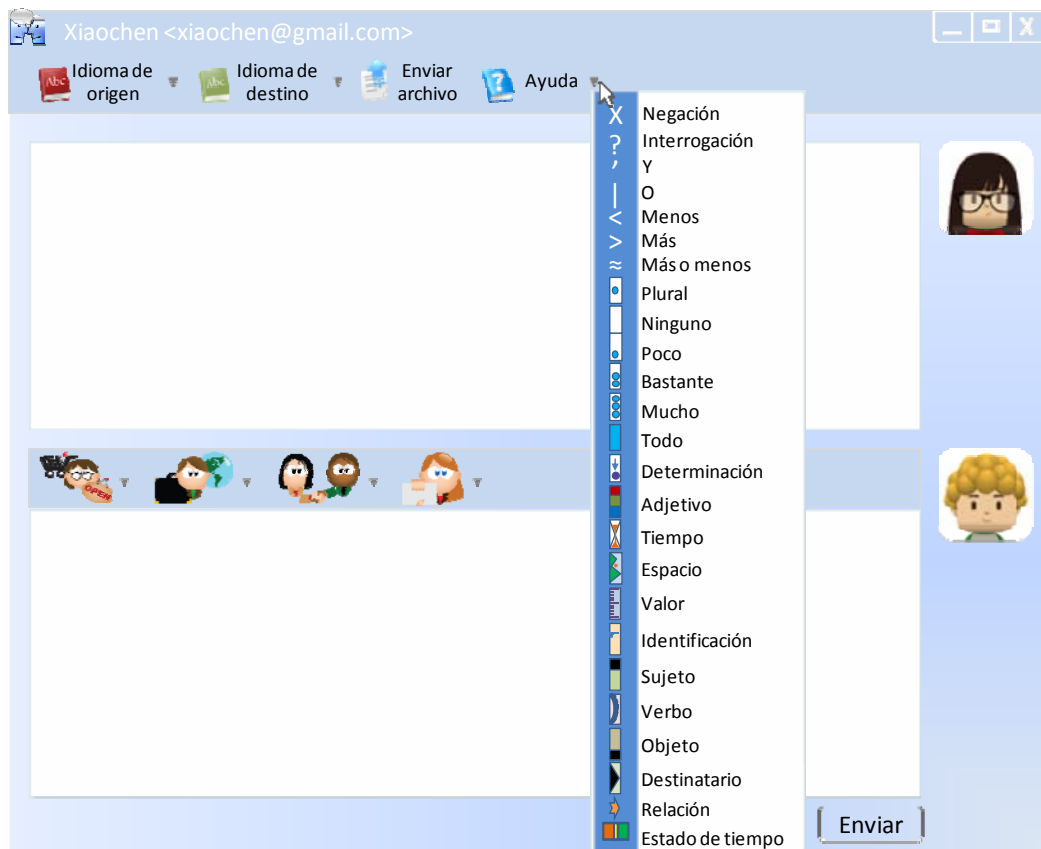


Figura 4.11 Captura de pantalla de la interfaz de ayuda

En la barra tipo de mensajes, hay dos partes, una los mensajes comunes, y otra los mensajes que editan los usuarios.

En los mensajes comunes, se ofrecen tres temas comunes de comunicación en Internet a los usuarios, negocios electrónicos, viaje y amistad (*Figura 4.12*). Sobre estos tres temas, se hacen algunos ejemplos, de algunas frases que siempre utilizan en la vida cotidiana,

y los elementos de la frase, los deja en blanco, se pueden rellenar según los deseos de usuario, por ejemplo “Tiempo”, “Lugar”, y “Objeto”. Así, los usuarios solo necesitan escribir estos elementos en los campos en blanco para generar los mensajes. Los diferentes niveles tienen diferentes mensajes de mas sencillo a complejo, los usuarios pueden elegir lo que necesitan según sus deseos. Así, no solo ahorra tiempo para editar los mensajes, de manera mas fácil y cómoda, también reducen la posibilidad de errores de traducción automática.



Figura 4.12 Captura de pantalla de la interfaz de los elementos de tres temas comunes de comunicación

Los mensajes editados por los usuarios, sirven para cuando los mensajes comunes no cubren lo que quiere, o porque quiere describir unas ideas más complejas o especiales. Es imposible ofrecer todos los mensajes a los usuarios, cada uno tiene su costumbre y situación especial, por eso estos tipos de mensajes permiten a solucionar este problema.

Como todos saben, los lenguajes naturales tienen elementos básicos parecidos, los tres idiomas que se han elegido, español, chino, inglés son tipo de SVO, por eso se pueden hacer los mensajes en la misma forma, es fácil de realizar la traducción entre ellos. En el capítulo 1 ya se ha introducido la gramática de lenguaje VILA_1, hay cuatro tipos de expresiones lingüísticas de identificación, y otras cuatro expresiones extras (*Figura 4.13*), los usuarios a través de elegir estas expresiones, organizan las frases y generan los mensajes.

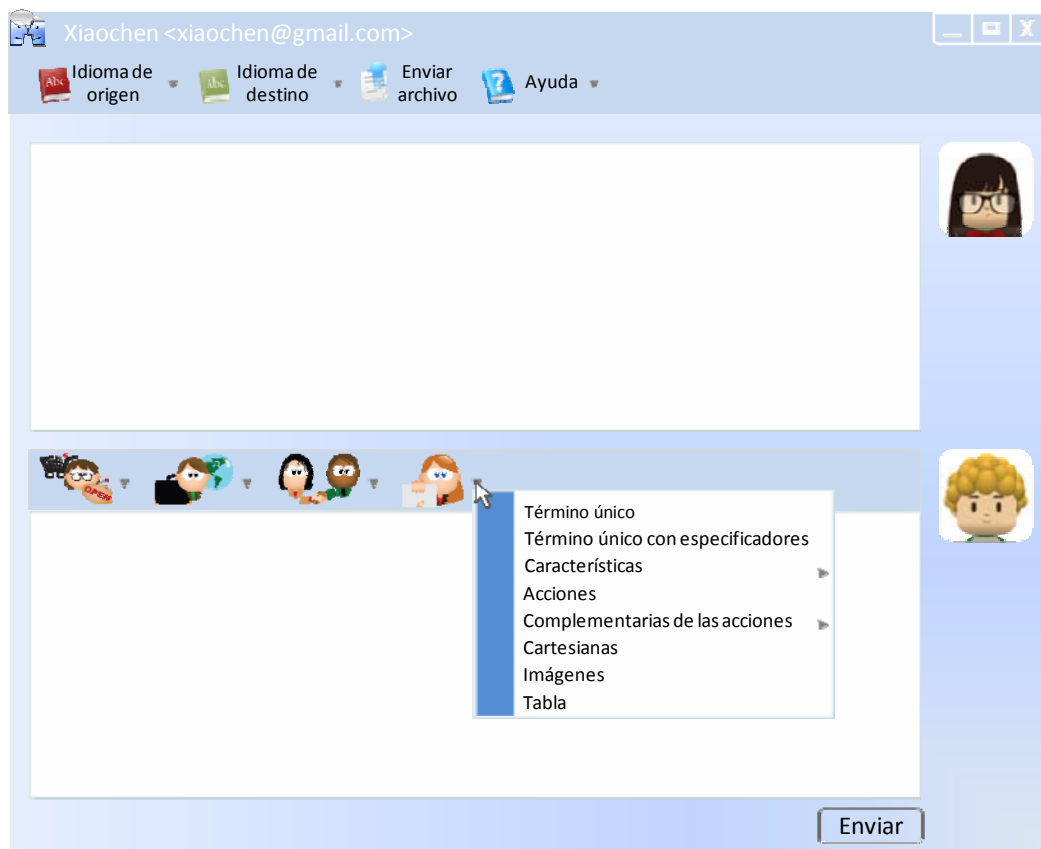


Figura 4.13 Captura de pantalla de la interfaz de los elementos para editar mensaje propio

También detrás de algunos elementos hay flecha, porque estos elementos se dividen en partes, como la figura (Figura 4.14):

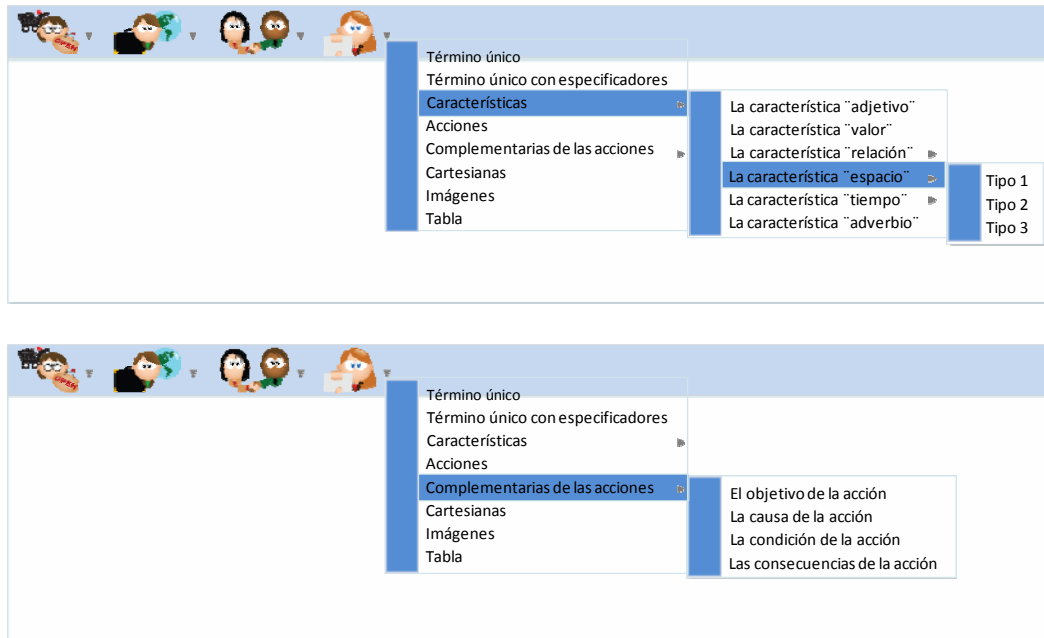


Figura 4.14 Captura de pantalla de la interfaz de los partes de un elemento

Los usuarios si encuentran algunos problemas en el proceso de editar los mensajes, o tienen algunas dudas de uso de VILA_1, sólo necesitan pinchar F1. Sale una ventana de ayuda (Figura 4.15) que tiene el manual del lenguaje VILA_1 y el sistema de mensajería instantánea.

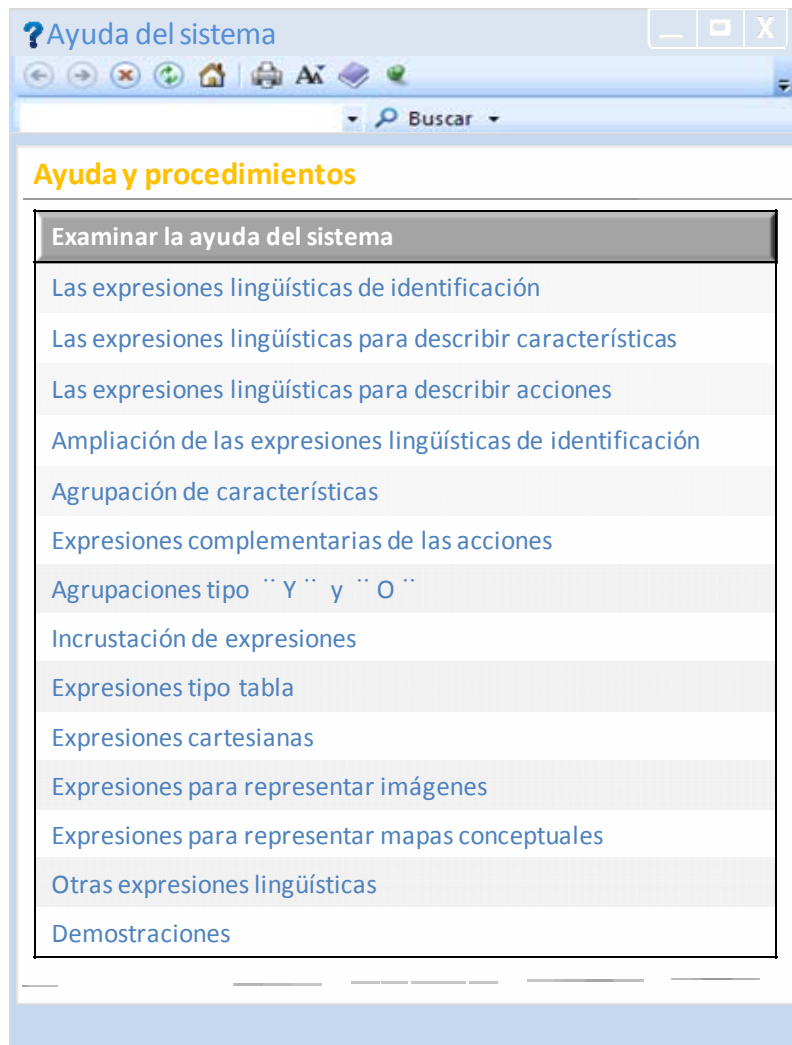


Figura 4.15 Ventana de ayuda del sistema

4.3.4 El uso de la aplicación de mensajería instantánea

Se mencionaron anteriormente las funciones básicas de este software de mensajería instantánea, ahora se ofrecen algunos ejemplos para explicar cómo editar, enviar y recibir los mensajes.

Porque en esta tesis doctoral en principio se explica cómo utiliza el software de mensajería instantánea basado en lenguaje VILA_1 para realizar el intercambio de mensajes, usando el vocabulario chino y el español, por eso sólo se pone los ejemplos entre chino y español.

4.3.4.1 El uso de tipo de mensajes comunes

Se muestra hace un ejemplo para explicar cómo se editan y envían los mensajes comunes. Suponemos que un usuario chino, Xiaochen, está hablando con un usuario español, Javier, sobre el tema de negocio, por eso Xiaochen tiene que elegir chino como el idioma de origen y el español como el idioma de destino, y Javier tiene que elegir español como el idioma de origen y chino como el idioma de destino, y luego en el tema de negocio electrónico elige los mensajes correspondientes para realizar la comunicación

Por ejemplo, Xiaochen quiere consultar el precio de la camiseta blanca de Real Madrid, es una consulta sencilla, así en la lista de “Preguntar precio” elige “Nivel 1” (Figura 4.16):

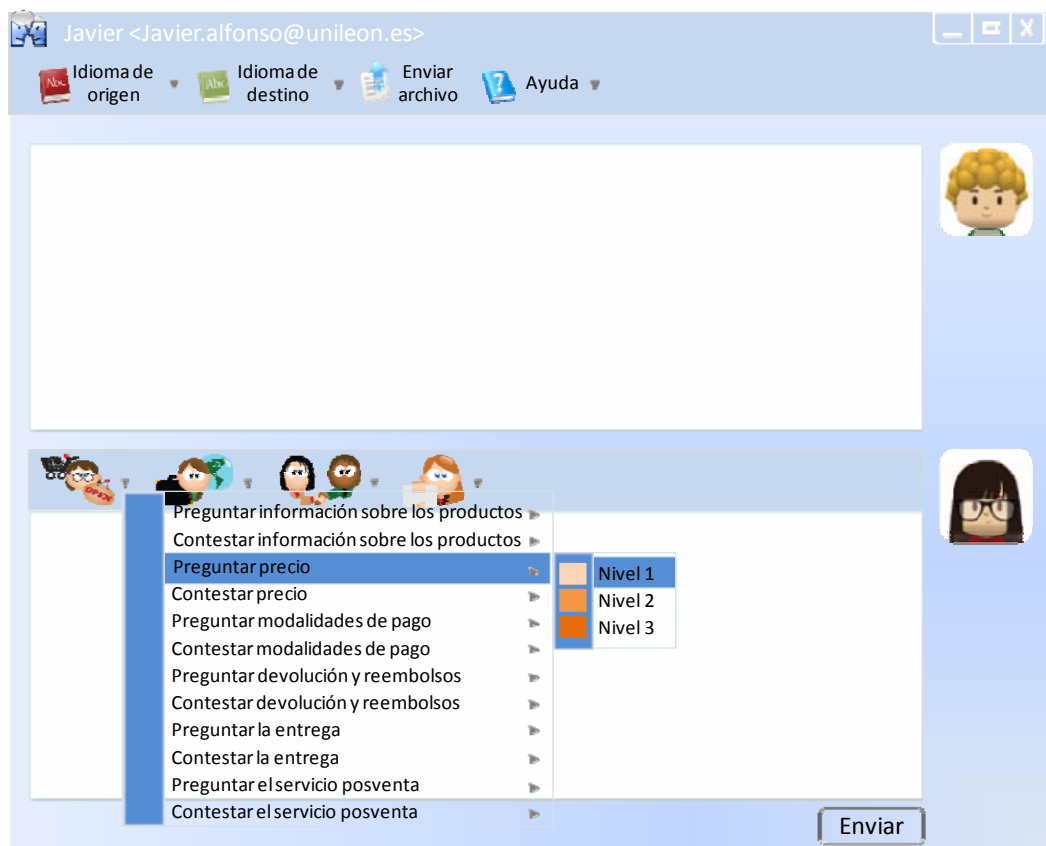


Figura 4.16 Captura de pantalla de la interfaz de elige tema de preguntar precio

Como se presenta en la figura (*Figura 4.17*), sale una ventana del lenguaje VILA_1 en el campo de editar mensajes.

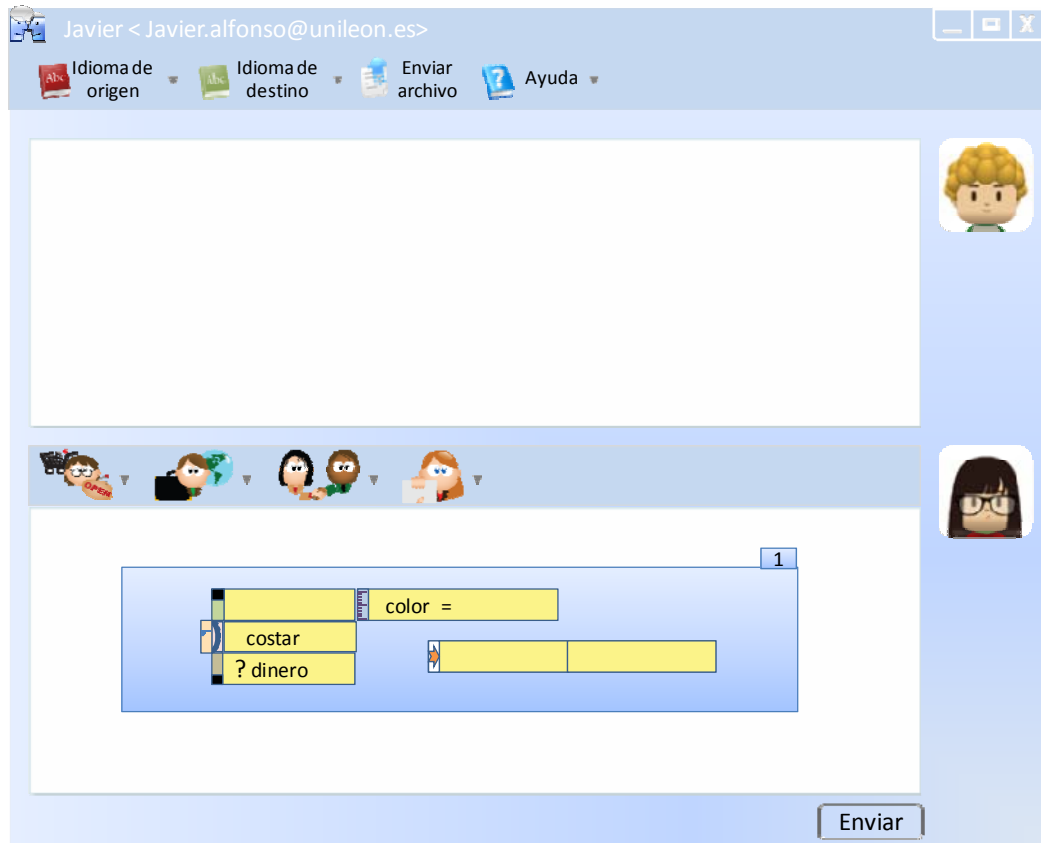


Figura 4.17 Captura de pantalla de la interfaz de editar mensaje

El usuario Xiaochen sólo necesita llenar las informaciones (*Figura 4.18*), y pinchar el botón “Enviar”, y el mensaje se envía al servidor, y el servidor según el vocabulario de la base de datos, lo traduce a español, y luego envía a Javier, se presenta en el campo de los mensajes de Javier. Para que sea más fácil de entender, aquí se explica en español, en realidad en el sistema de usuario Xiaochen, la interfaz lo presenta en chino.

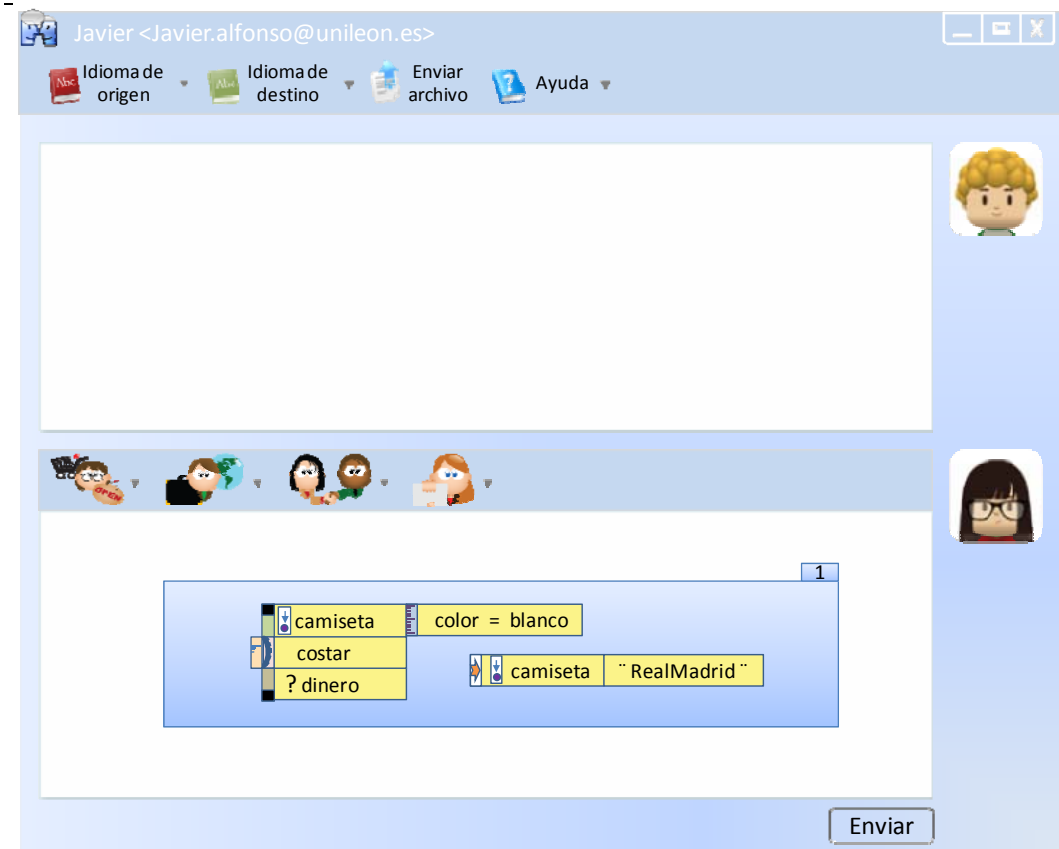


Figura 4.18 Captura de pantalla de la interfaz de generar un mensaje

Aquí hay una cosa que debe explicarse bien. Delante de todos los términos, existen una serie de especificadores gráficos. Se dividen en dos grupos, es decir delante de los términos hay dos opciones a elegir, el primer grupo incluye “X”, “?” y “↓”, El segundo grupo incluye “ ”, “□”, “<”, “>”, “≈”, “□”, “□”, “□”, “□” y “□”. Sólo necesita pinchar los botones a elegir, cada vez que se pincha cambia la circulación de los especificadores gráficos. Además delante de Adjetivo y Adverbio también hay dos grupos de especificadores gráficos, primer grupo son “X” y “?”, el segundo son “□”, “□”, y “□”, la forma de uso igual que los de término.

Es importante destacar que durante la instalación del sistema, se instala junto con el cliente un paquete, que contiene un archivo con

- todas las palabras que se pueden insertar a la hora de editar VILA_1. El archivo de palabras se descarga solamente en el idioma que el usuario del sistema seleccione para el uso de VILA_1. Este procedimiento resuelve dos problemas:

- Evita errores sintácticos, ya que se controlan las palabras que se insertan. Dicho control se realiza mediante un sistema de autocompletado, en el que a medida que se va escribiendo una determinada palabra, el cliente muestra las distintas opciones que tiene. Por ejemplo, si se teclea la letra J, y el idioma en el que se está escribiendo es el castellano, se mostrarán todas las palabras que empiecen por J en español.
- Reduce la carga de procesamiento del servidor, y más concretamente de las bases de datos, debido a que no se realizan consultas al servidor cada vez que se inserta una palabra, sino que todo ese proceso es realizado por el cliente, mediante consultas al archivo de palabras mencionado anteriormente.

Y supone después de leer el mensaje de Xiaochen, el usuario Javier quiere contestar lo que la camiseta blanca de Real Madrid cuesta 65 euros, si una vez comprara más de cinco, no cuesta el gasto de envío. Así en la lista de “negocio electrónico”, elige “nivel 2” de “contestar precio”, sale la ventana del lenguaje VILA_1 (Figura 4.19):

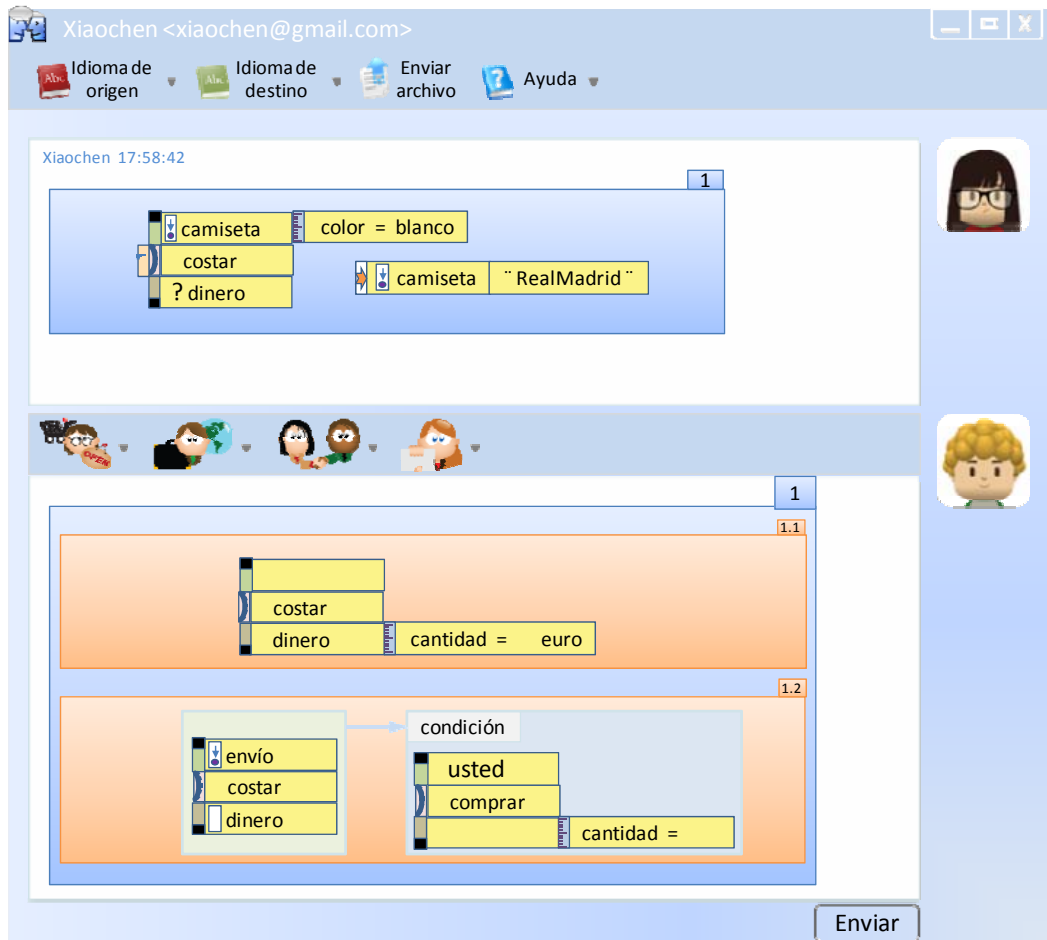


Figura 4.19 Captura de pantalla de la interfaz de editar mensaje

El usuario Javier llena los blancos de la ventana y ya esta, como se ve en la figura (Figura 4.20) siguiente:

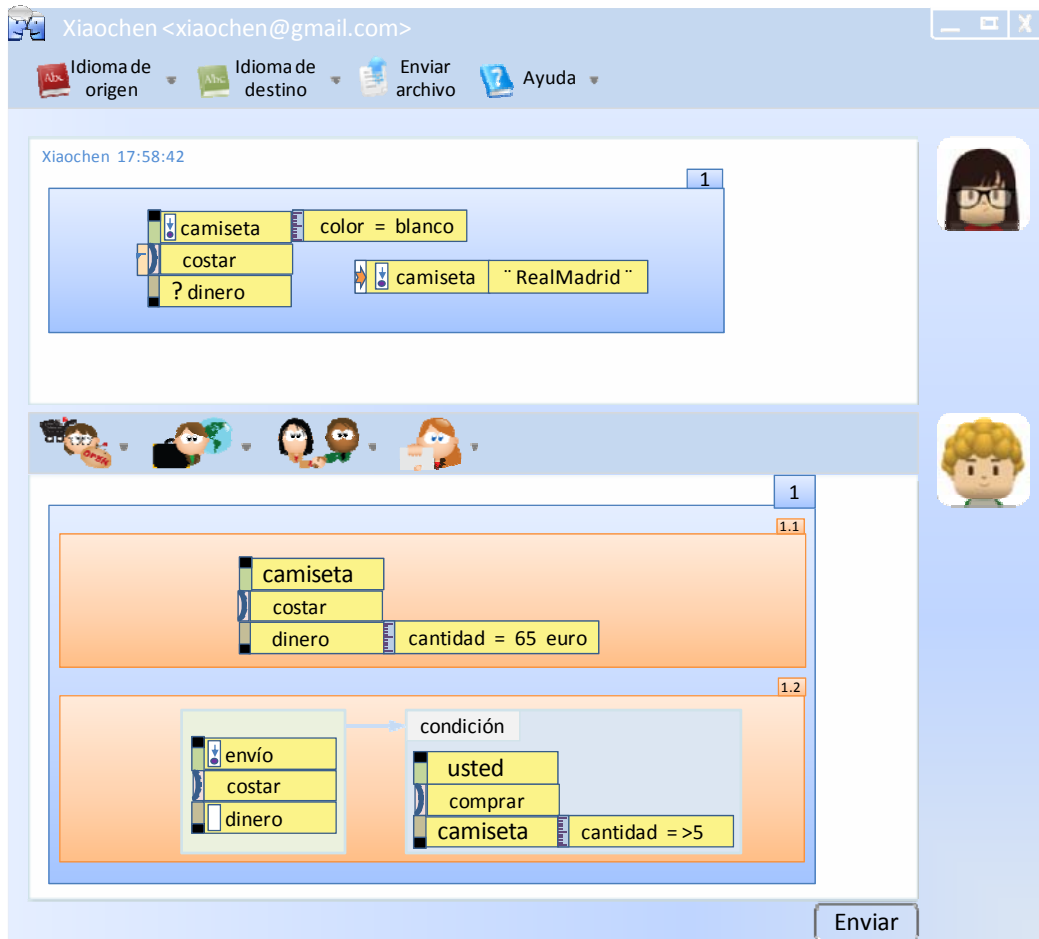


Figura 4.20 Captura de pantalla de la interfaz de generar un mensaje

Y luego pincha el botón “Enviar”, este mensaje a través de los procesamientos anteriores y manda a Xiaochen, la información sale en el campo de los mensajes (Figura 4.21).

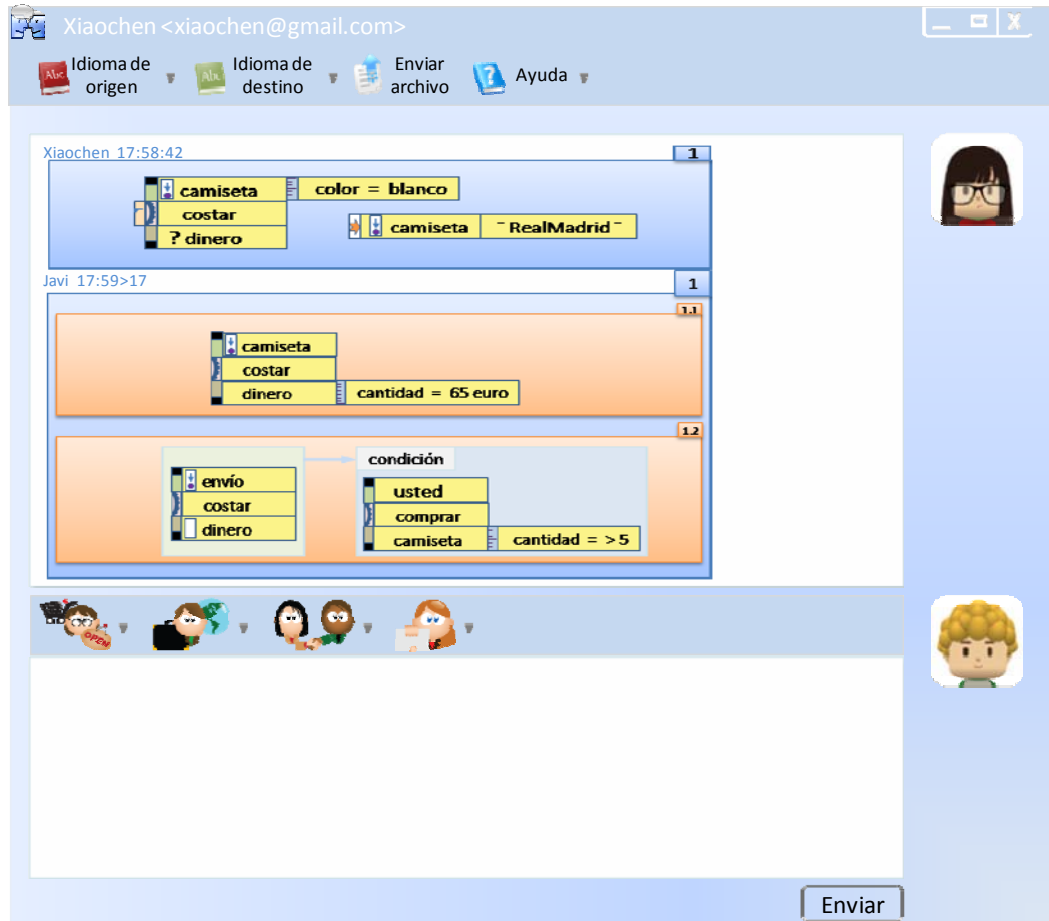


Figura 4.21 Captura de pantalla de la interfaz de los mensajes se sale en el campo de los mensajes

La interfaz de chat de usuario chino Xiaochen es el siguiente:



Figura 4.22 Interfaz de chat de usuario chino Xiaochen

4.3.4.2 El uso de tipo de mensajes propios

A continuación se muestra un ejemplo para explicar la edición de mensajes propios. Suponemos que Javier (usuario de España) quiere mandar un mensaje a Xiaochen (usuario de China) con contenido “Yo vivo en León, es una ciudad tranquila y bonita, está en el norte de España.”, y adjunta una foto de León y un mapa que indica en qué parte de España está León.

Según la sintaxis del lenguaje VILA_1, este mensaje incluye el elemento acción, el elemento adjetivo, el elemento espacio, el

elemento imagen y el elemento mapa conceptual. Hay que elegir todas estas expresiones en el tema de los mensajes propios, y en el campo de editar mensajes salen los rectángulos, como la figura (Figura 4.23):

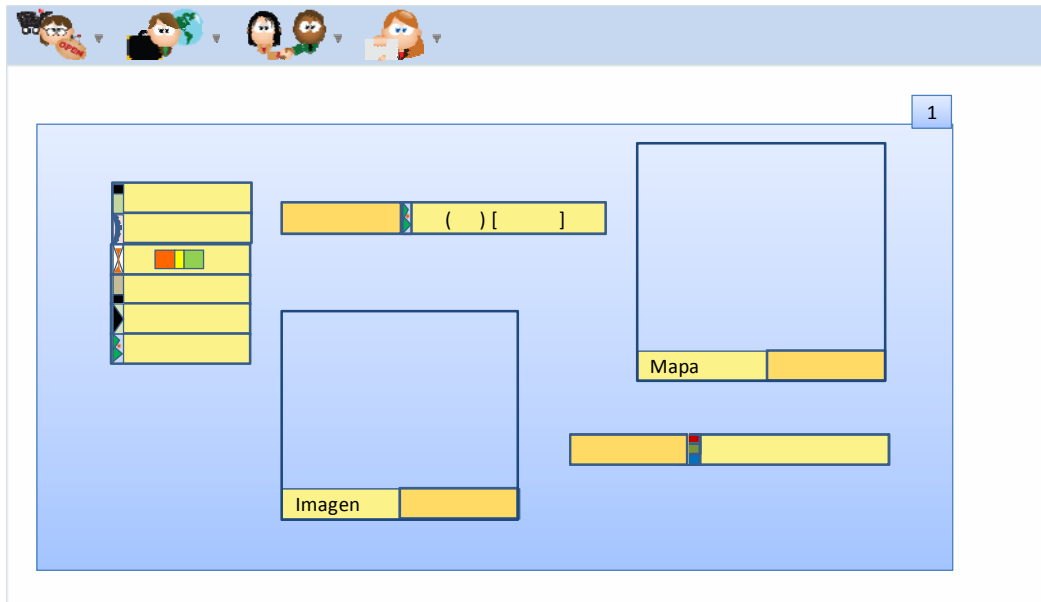


Figura 4.23 Captura de pantalla de la interfaz de editar un mensaje propio

Ahora llena los blancos de cada expresión, genera el mensaje deseado por Javier en la figura (Figura 4.24) siguiente:

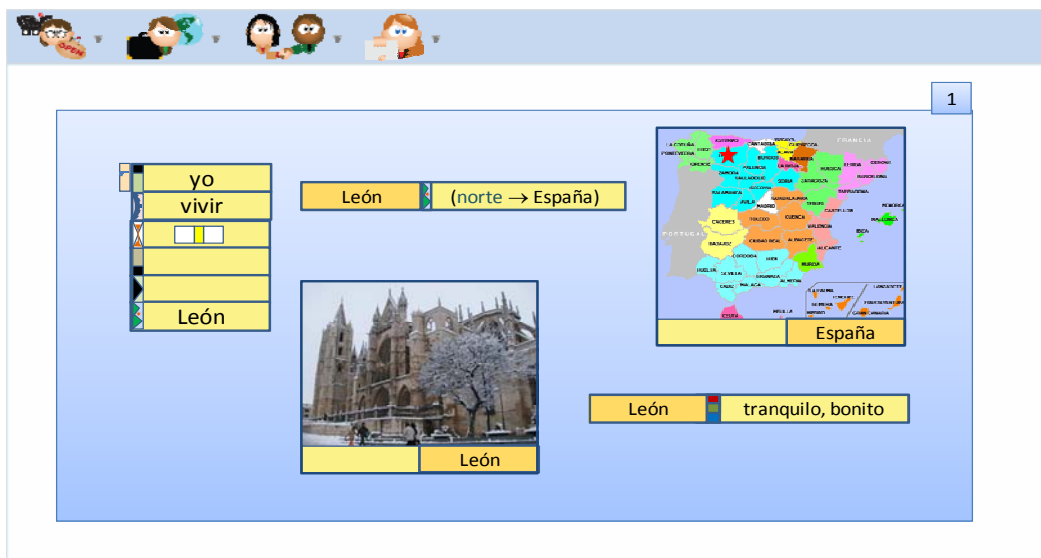


Figura 4.24 Captura de pantalla de la interfaz de generar un mensaje propio

En la figura se puede ver, delante de la expresión de “Acción” hay un símbolo identificador, significa entre estas 5 expresiones, él es el núcleo, otras expresiones son para describirlo. Cuando asegura el núcleo del mensaje, sólo necesita pinchar la expresión, se sale este símbolo identificador.

4.4 El diseño del servidor

Para realizar la investigación de esta tesis, hay varios aspectos importantes que hay que considerar. Por ejemplo cómo realiza la conexión entre el cliente y el servidor, cómo conecta con la base de datos, en qué forma puede realizar la transmisión de los mensajes seguro y rápido, cómo realiza la función de traducción automática.

4.4.1 Establecer la conexión

Como se mencionó anterior, para realizar el intercambio de mensajes en tiempo real, primero hay que conectar al servidor, y para conectar al servidor, un procesamiento obligatorio es que el cliente tiene que identificarse en el servidor. Está basado en la protección de acceso al sistema mediante una dirección de correo electrónico, y una contraseña, por lo que sólo pueden acceder al sistema usuarios registrados. La dirección de correo electrónico, la contraseña, el usuario son almacenados en la base de datos de usuarios. El proceso de autenticación de usuarios se resume en la (Figura 4.25).

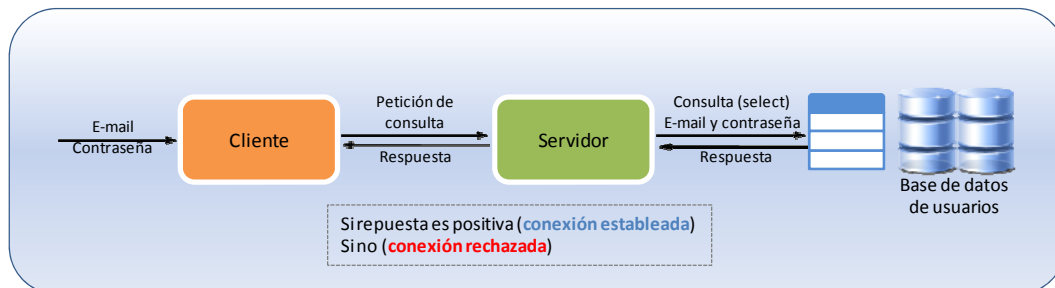


Figura 4.25 Esquema de funcionamiento del proceso de autenticación de usuarios

4.4.2 Conexión de las bases de datos

Las bases de datos han sido desarrolladas con MySQL. Las conexiones, las consultas, y las actualizaciones de la base de datos se realizan mediante el uso de la tecnología JDBC². La conexión con las base de datos se realiza según el esquema que se muestra en la (Figura 4.26).

Existen distintos tipos de controladores JDBC, en este sistema se ha utilizado un controlador de tipo 3, debido a que se ha considerado el controlador más eficiente. Este tipo de controlador está formado por bibliotecas escritas en Java puro, que traducen directamente solicitudes JDBC a un protocolo propio de la base de datos MySQL.

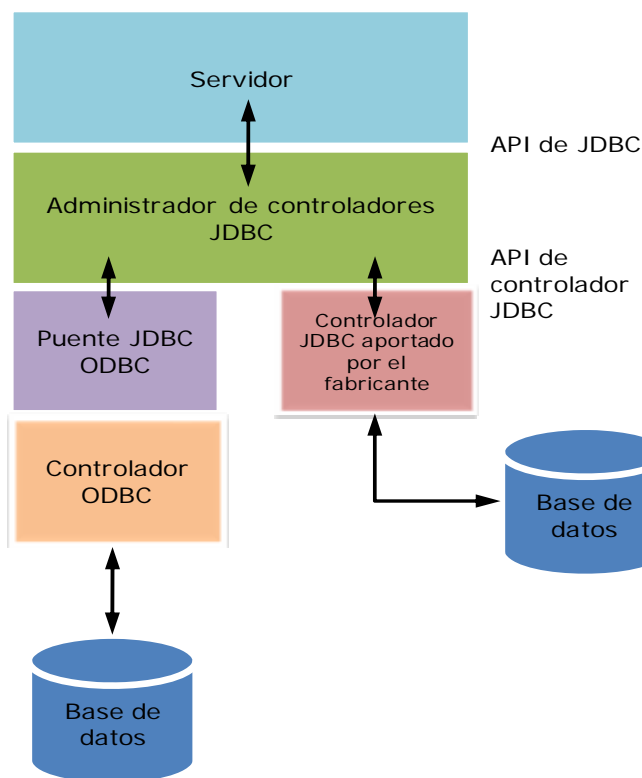


Figura 4.26 Ruta de comunicaciones JDBC con las bases de datos

² JDBC (Java Database Connectivity), es un API que permite la ejecución de operaciones sobre bases de datos, desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute, o de la base de datos a la cual se pretenda acceder.

La ruta de comunicaciones representada en la (*Figura 4.26*), sigue el muy exitoso modelo del ODBC de Microsoft [55], que se basa en la idea, de que todos los programas escritos respetando la API se comunican con el administrador de controladores, que a su vez hace uso de los controladores conectados a él para comunicarse con la base de datos en sí.

4.4.3 Formato de transmisión de mensajes

Según la introducción del capítulo 3, se utiliza el protocolo SOAP para realizar la transmisión de los mensajes, por eso en el procesamiento de intercambio de mensajes, todos los archivos generados con este sistema, son codificados en formato XML, siguiendo los XML Schema de las estructuras VILA_1 almacenados en las bases de datos.

El sistema genera un archivo XML por cada expresión VILA_1, y un archivo “Documento” por cada documento, en el que se almacenan las diferentes estructuras, e información sobre la versión, la fecha, el idioma, y número de documentos de las que está formado. Cada documento incluye también una carpeta denominada “Resources”, que se destina al almacenamiento de todos los recursos del documento, y que forman parte de alguna de las estructuras del lenguaje VILA_1. Debido a que no resulta funcional el uso y manejo de todos los archivos mencionados anteriormente, se ha implementado una función que comprime todo el contenido del documento, en un único archivo con la extensión “.vla”.

Para realizar la compresión de los distintos archivos que componen el documento, y su encapsulamiento, se ha aplicado un algoritmo de compresión ZIP, se tomó la decisión de utilizar el formato ZIP porque es un formato de archivo simple, que comprime cada uno de los archivos que contiene de manera individual, permitiendo recuperar

cada uno de los archivos independientemente, sin necesidad de leer todo el documento, proporciona unos rendimientos elevados.

No se ha implementado ningún método de encriptación ya que el único fin de la compresión es facilitar el manejo de los archivos, permitiendo así que puedan ser convertidos a otros estándares, permitiendo así que otros grupos de investigación utilicen y desarrollen contenidos en el lenguaje VILA_1. En la (Figura 4.27) se puede observar un esquema con la estructura de un archivo “.vla”.

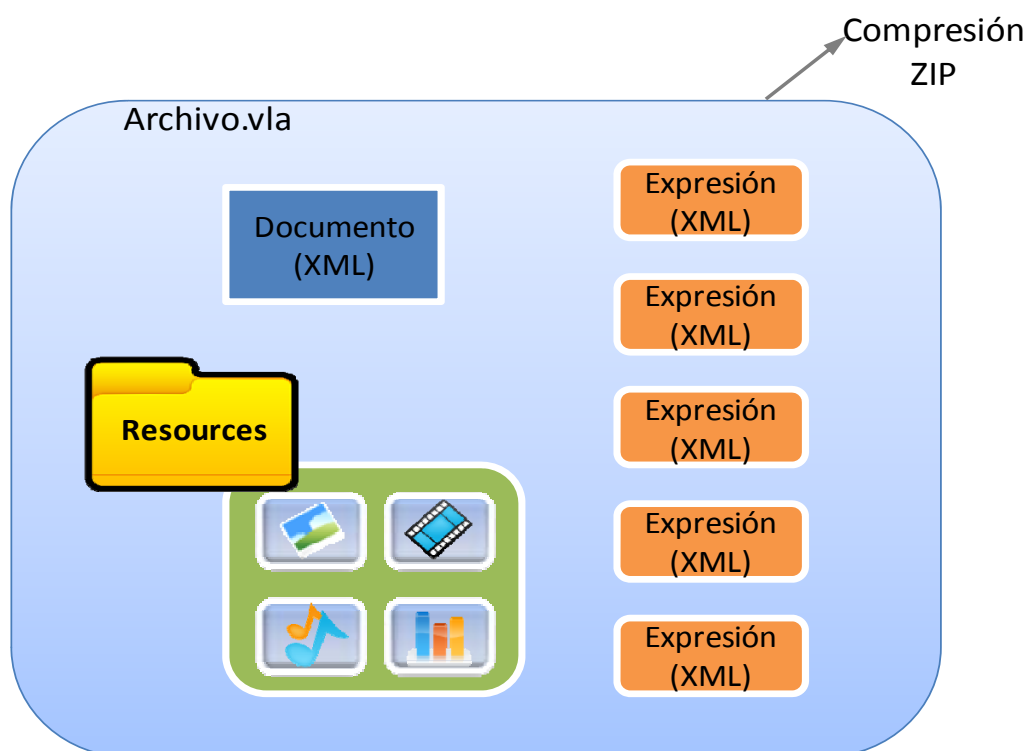


Figura 4.27 Estructura de un archivo “.vla”

Los archivos desarrollados con este sistema, tienen la extensión “.vla”, y su tabla MIME se puede observar en la (Tabla 4.2).

Tipo MIME	Extensión	Descripción
application/vila_1	.vla	Documento de VILA_1

Tabla 4.2 Tabla MIME de un archivo “.vla”

El sistema realiza dos controles sobre los archivos contenidos en la carpeta “Resources”. El primer control es de nombres, comprobando que no exista más de un archivo con el mismo nombre, y si lo hubiera se ejecuta automáticamente un algoritmo que lo cambia, resultando el proceso totalmente transparente para el usuario final. El segundo control tiene que ver con las extensiones de los archivos, asignando según su tipo, al paquete que debe tratarlo y visualizarlo (*Tabla 4.3*).

Tipo de Archivo	Extensiones	Módulo
Gráfico	.JPG .GIF .BMP .PNG	Visualización – Clase: Gráfica
Audio	.MP3 .WAV	Visualización – Clase: Audio
Video	.AVI .MP4 .MPEG .WMV	Visualización – Clase: AudioVideo

Tabla 4.3 Control de extensiones del sistema.

Antes de procesar un archivo “.vla”, el prototipo lo analiza, comprobando su integridad, y su correcta formación, realizando una validación para cada uno de los documentos XML por los que está compuesto.

A continuación se muestra un documento completo en VILA_1 (*Figura 4.28*), y las estructuras que son utilizadas para representarlo y transmitirlo. El documento VILA_1 es el ejemplo anterior “Yo vivo en León, es una ciudad tranquila y bonita, está en el norte de España.”

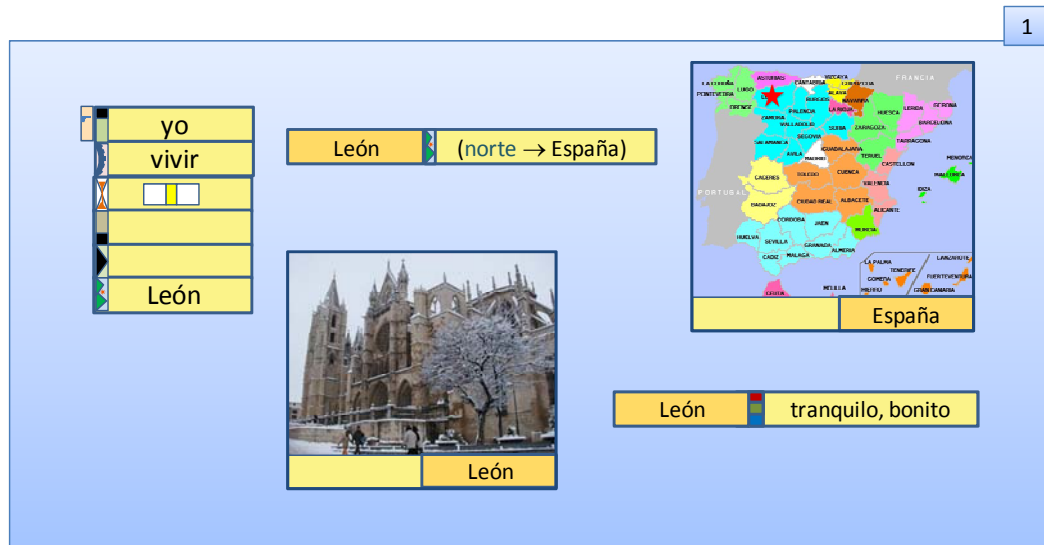


Figura 4.28 Un documento completo en VILA_1

Este documento está compuesto por cinco tipos de estructuras VILA_1 diferentes; “Acción”, “Adjetivo”, “EspacioA”, “Imagen” y “Mapa”, que se detallan a continuación.

Una estructura VILA_1 de tipo “Acción” se utiliza para describir acciones, sirven para describir los cambios de características de un concepto o entidad. Estos cambios se producen por la acción de fuerzas y siempre están asociados a un intervalo temporal. Este tipo de estructura está formada por los campos de tipo “Identificación”, tipo “Sujeto”, tipo “Verbo”, tipo “Tiempo”, tipo “Objeto”, tipo “Destinatario”, tipo “Espacio”, y tipo “Documento”.

En la figura se puede ver, delante de la expresión de “Acción” hay un símbolo identificador, significa entre estas 5 expresiones, él es el núcleo, otras expresiones son para describirlo. Si se quiere que una expresión sea el núcleo, solo se necesita poner “true” en el valor de Identificación.

El campo de tipo “Sujeto” contiene los siguientes elementos:

- “EspecificadoresA” (*Complex*). Almacena los especificadores como “Negación” (*booleano*), “Interrogación” (*booleano*) y

- “Determinación” (*booleano*), si un especificador está insertado de manera aseverativa (*true*), o de manera negativa (*false*).
- “EspecificadoresB” (*Complex*). Almacena los especificadores como “Singular” (*booleano*), “Plural” (*booleano*), “MenosDe” (*booleano*), “MasDe” (*booleano*), “MasOMenos” (*booleano*), “Ninguno” (*booleano*), “Poco” (*booleano*), “Bastante” (*booleano*), “Mucho” (*booleano*), “Todo” (*booleano*), si un especificador está insertado de manera aseverativa (*true*), o de manera negativa (*false*).
- “Palabra” (*string*). Almacena los sujetos.

El campo de tipo “Verbo” está compuesto por los siguientes elementos:

- “EspecificadoresC” (*Complex*). Almacena los especificadores como “Negación” (*booleano*) y “Interrogación” (*booleano*), si un especificador está insertado de manera aseverativa (*true*), o de manera negativa (*false*).
- “Palabra” (*string*). Almacena los verbos.

El campo de tipo “Tiempo” está formado por los siguientes elementos:

- “Pasado” (*boolean*). Almacena el tiempo pasado, si está elegido de manera aseverativa (*ture*), o de manera negativa (*false*).
- “Presente” (*boolean*). Almacena el tiempo presente, si está elegido de manera aseverativa (*ture*), o de manera negativa (*false*).
- “Futuro” (*boolean*). Almacena el tiempo futuro, si está elegido de manera aseverativa (*ture*), o de manera negativa (*false*).

El campo de tipo “Objeto” está compuesto por los siguientes elementos:

- “EspecificadoresA” (*Complex*).
- “EspecificadoresB” (*Complex*).
- “Palabra” (*string*). Almacena los objetos.

El campo de tipo “Destinatario” está compuesto por los siguientes elementos:

- “EspecificadoresA” (*Complex*).
- “EspecificadoresB” (*Complex*).
- “Palabra” (*string*). Almacena los destinatarios.

El campo de tipo “Espacio” está compuesto por los siguientes elementos:

- “EspecificadoresA” (*Complex*).
- “EspecificadoresB” (*Complex*).
- “Palabra” (*string*). Almacena los espacios.

Los campos de tipo “Documento” son iguales en todas las estructuras VILA_1. Si el valor de la “Documento” es 1, las estructuras están en el mismo documento. En la (Figura 4.29) el XML que se genera en el ejemplo de la estructura de “Acción”.

```
<?xml version="1.0"?>
<Accion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///e:/XML%20Bueno/accion.xsd">
  <Identificacion>true</Identificacion>
  <Sujeto>
    <EspecificadoresA>
      <Negacion>>false</Negacion>
      <Interrogacion>>false</Interrogacion>
      <Determinacion>>false</Determinacion>
    </EspecificadoresA>
    <EspecificadoresB>
      <Singular>>false</Singular>
      <Plural>>false</Plural>
      <MenosDe>>false</MenosDe>
      <MasDe>>false</MasDe>
      <MasOMenos>>false</MasOMenos>
      <Ninguno>>false</Ninguno>
      <Poco>>false</Poco>
      <Bastante>>false</Bastante>
      <Mucho>>false</Mucho>
  </Sujeto>
</Accion>
```

```
<Todo>>false</Todo>
</EspecificadoresB>
<Palabra>yo</Palabra>
</Sujeto>
<Verbo>
  <EspecificadoresC>
    <Negacion>>false</Negacion>
    <Interrogacion>>false</Interrogacion>
  </EspecificadoresC>
  <Palabra>vivir</Palabra>
</Verbo>
<Tiempo>
  <Pasado>>false</Pasado>
  <Presente>>true</Presente>
  <Futuro>>false</Futuro>
</Tiempo>
<Objeto>
  <EspecificadoresA>
    <Negacion>>false</Negacion>
    <Interrogacion>>false</Interrogacion>
    <Determinacion>>false</Determinacion>
  </EspecificadoresA>
  <EspecificadoresB>
    <Singular>>false</Singular>
    <Plural>>false</Plural>
    <MenosDe>>false</MenosDe>
    <MasDe>>false</MasDe>
    <MasOMenos>>false</MasOMenos>
    <Ninguno>>false</Ninguno>
    <Poco>>false</Poco>
    <Bastante>>false</Bastante>
    <Mucho>>false</Mucho>
    <Todo>>false</Todo>
  </EspecificadoresB>
  <Palabra>    </Palabra>
</Objeto>
<Destinatario>
  <EspecificadoresA>
    <Negacion>>false</Negacion>
    <Interrogacion>>false</Interrogacion>
```



```

    <Determinacion>>false</Determinacion>
</EspecificadoresA>
<EspecificadoresB>
    <Singular>>false</Singular>
    <Plural>>false</Plural>
    <MenosDe>>false</MenosDe>
    <MasDe>>false</MasDe>
    <MasOMenos>>false</MasOMenos>
    <Ninguno>>false</Ninguno>
    <Poco>>false</Poco>
    <Bastante>>false</Bastante>
    <Mucho>>false</Mucho>
    <Todo>>false</Todo>
</EspecificadoresB>
<Palabra>    </Palabra>
</Destinatario>
<Espacio>
    <EspecificadoresA>
        <Negacion>>false</Negacion>
        <Interrogacion>>false</Interrogacion>
        <Determinacion>>false</Determinacion>
    </EspecificadoresA>
    <EspecificadoresB>
        <Singular>>false</Singular>
        <Plural>>false</Plural>
        <MenosDe>>false</MenosDe>
        <MasDe>>false</MasDe>
        <MasOMenos>>false</MasOMenos>
        <Ninguno>>false</Ninguno>
        <Poco>>false</Poco>
        <Bastante>>false</Bastante>
        <Mucho>>false</Mucho>
        <Todo>>false</Todo>
    </EspecificadoresB>
    <Palabra>León</Palabra>
</Espacio>
<Documento>1</Documento>
</Accion>

```

Figura 4.29 XML del archivo "Acción" del ejemplo

Una estructura VILA_1 de tipo “Adjetivo” sirven para darle valor cualitativo a magnitudes y otros elementos de valoración, y está formada por los campos de tipo “Identificación”, tipo “Identificador”, tipo “Adjetivo”, y tipo “Documento”.

El campo de tipo “Identificador” contiene los siguientes elementos:

- “EspecificadoresA” (*Complex*). Almacena los especificadores como “Negación” (*booleano*), “Interrogación” (*booleano*) y “Determinación” (*booleano*), si un especificador está insertado de manera aseverativa (*true*), o de manera negativa (*false*).
- “EspecificadoresB” (*Complex*). Almacena los especificadores como “Singular” (*booleano*), “Plural” (*booleano*), “MenosDe” (*booleano*), “MasDe” (*booleano*), “MasOMenos” (*booleano*), “Ninguno” (*booleano*), “Poco” (*booleano*), “Bastante” (*booleano*), “Mucho” (*booleano*), “Todo” (*booleano*), si un especificador está insertado de manera aseverativa (*true*), o de manera negativa (*false*).
- “Palabra” (*string*). Almacena los identificadores.

El campo de tipo “Adjetivo” está compuesto por los siguientes elementos:

- “EspecificadoresD” (*Complex*). Almacena los especificadores como “Negación” (*booleano*) y “Interrogación” (*booleano*), si un especificador está insertado de manera aseverativa (*true*), o de manera negativa (*false*).
- “EspecificadoresE” (*Complex*). Almacena los especificadores como “Poco” (*booleano*), “Bastante” (*booleano*) y “Muy” (*booleano*), si un especificador está insertado de manera aseverativa (*true*), o de manera negativa (*false*).
- “Palabra” (*string*). Almacena los adjetivos.

En la (Figura 4.30) el XML que se genera en el ejemplo de la estructura de “Adjetivo”.

```

<?xml version="1.0"?>
<Adjetivo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///e:/XML%20Bueno/adjetivo.xsd"
  <Identificacion>>false</Identificacion>
  <Identificador>
    <EspecificadoresA>
      <Negacion>>false</Negacion>
      <Interrogacion>>false</Interrogacion>
      <Determinacion>>false</Determinacion>
    </EspecificadoresA>
    <EspecificadoresB>
      <Singular>>false</Singular>
      <Plural>>false</Plural>
      <MenosDe>>false</MenosDe>
      <MasDe>>false</MasDe>
      <MasOMenos>>false</MasOMenos>
      <Ninguno>>false</Ninguno>
      <Poco>>false</Poco>
      <Bastante>>false</Bastante>
      <Mucho>>false</Mucho>
      <Todo>>false</Todo>
    </EspecificadoresB>
    <Palabra>León</Palabra>
  </Identificador>
  <Adjetivo>
    <EspecificadoresD>
      <Negacion>>false</Negacion>
      <Interrogacion>>false</Interrogacion>
    </EspecificadoresD>
    <EspecificadoresE>
      <Poco>>false</Poco>
      <Bastante>>false</Bastante>
      <Muy>>false</Muu>
    </EspecificadoresE>
    <Palabra>tranquilo</Palabra>
    <Palabra>bonito</Palabra>
  </Adjetivo>
  <Documento>1</Documento>
</Adjetivo>

```

Figura 4.30 XML del archivo "Adjetivo" del ejemplo

Una estructura VILA_1 de tipo “EspacioA” puede especificarse el espacio físico o virtual en el que se encuentra ubicada la entidad o el concepto que se está describiendo o donde sucede algo, y está formada por los campos de tipo “Identificación”, tipo “Espacio”, tipo “Medida”, y tipo “Documento”.

El campo de tipo “Espacio” contiene los siguientes elementos:

- “TerminoRelativo” (*string*). Almacena los puntos de partida para especificar el espacio.
- “Identificador” (*Complex*). Almacena tanto los conceptos como las entidades.

El campo de tipo “Medida” está compuesto por los siguientes elementos:

- “Palabra” (*string*). Almacena los espacios figuras geométricas o términos.
- “Dato” (*string*). Almacena los datos de identificación.

En la (Figura 4.31) el XML que se genera en el ejemplo de la estructura de “EspacioA”.

```
<?xml version="1.0"?>
<EspacioA xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///e:/XML%20Bueno/espacio1mi.xsd">
  <Identificacion>>false</Identificacion>
  <Espacio>
    <TerminoRelativo>norte</TerminoRelativo>
    <Identificador>
      <EspecificadoresA>
        <Negacion>>false</Negacion>
        <Interrogacion>>false</Interrogacion>
        <Determinacion>>false</Determinacion>
      </EspecificadoresA>
      <EspecificadoresB>
        <Singular>>false</Singular>
        <Plural>>false</Plural>
        <MenosDe>>false</MenosDe>
      </EspecificadoresB>
    </Identificador>
  </Espacio>
</EspacioA>
```

```

        <MasDe>>false</MasDe>
        <MasOMenos>>false</MasOMenos>
        <Ninguno>>false</Ninguno>
        <Poco>>false</Poco>
        <Bastante>>false</Bastante>
        <Mucho>>false</Mucho>
        <Todo>>false</Todo>
    </EspecificadoresB>
    <Palabra>España</Palabra>
</Identificador>
</Espacio>
<Medida>
    <Palabra>    </Palabra>
    <Dato>        </Dato>
</Medida>
<Documento>1</Documento>
</EspacioA>

```

Figura 4.31 XML del archivo "EspacioA" del ejemplo

Una estructura VILA_1 de tipo "Imagen" se utiliza para insertar conocimiento directo en la aplicación a través de una imagen o de un gráfico, esta estructura está compuesta por los campos de tipo "Identificación", tipo "Identificador", tipo "Imagen", y tipo "Documento". El campo de tipo "Identificador" (*complex*) contiene el nombre, o el título de la imagen que se desea mostrar, y el campo de tipo "Imagen" (string) contiene la ruta de la imagen en el directorio "Resources".

En la (Figura 4.32) el XML que se genera en el ejemplo de la estructura de "Imagen".

```

<?xml version="1.0"?>
<Imagen xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///e:/XML%20Bueno/imagen.xsd">
    <Identificacion>>false</Identificacion>
    <Identificador>
        <EspecificadoresA>
            <Negacion>>false</Negacion>
            <Interrogacion>>false</Interrogacion>

```

```

    <Determinacion>>false</Determinacion>
  </EspecificadoresA>
  <EspecificadoresB>
    <Singular>>false</Singular>
    <Plural>>false</Plural>
    <MenosDe>>false</MenosDe>
    <MasDe>>false</MasDe>
    <MasOMenos>>false</MasOMenos>
    <Ninguno>>false</Ninguno>
    <Poco>>false</Poco>
    <Bastante>>false</Bastante>
    <Mucho>>false</Mucho>
    <Todo>>false</Todo>
  </EspecificadoresB>
  <Palabra>León</Palabra>
</Identificador>
<Imagen>string</Imagen>
<Documento>1</Documento>
</Imagen>

```

Figura 4.32 XML del archivo "Imagen" del ejemplo

Una estructura VILA_1 de tipo "Mapa" sirven para representar lo que se conoce mapas conceptuales, esta estructura está compuesta por los campos de tipo "Identificación", tipo "Identificador", tipo "Mapa", y tipo "Documento". El campo de tipo "Identificador" (*complex*) contiene el nombre, o el título de la imagen que se desea mostrar, y el campo de tipo "Mapa" (string) contiene la ruta de la Mapa en el directorio "Resources".

En la (Figura 4.33) el XML que se genera en el ejemplo de la estructura de "Mapa".

```

<?xml version="1.0"?>
<Mapa xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///e:/XML%20Bueno/mapa.xsd">
  <Identificacion>>false</Identificacion>
  <Identificador>
    <EspecificadoresA>
      <Negacion>>false</Negacion>

```

```

    <Interrogacion>>false</Interrogacion>
    <Determinacion>>false</Determinacion>
</EspecificadoresA>
<EspecificadoresB>
    <Singular>>false</Singular>
    <Plural>>false</Plural>
    <MenosDe>>false</MenosDe>
    <MasDe>>false</MasDe>
    <MasOMenos>>false</MasOMenos>
    <Ninguno>>false</Ninguno>
    <Poco>>false</Poco>
    <Bastante>>false</Bastante>
    <Mucho>>false</Mucho>
    <Todo>>false</Todo>
</EspecificadoresB>
    <Palabra>España</Palabra>
</Identificador>
    <Imagen>string</Imagen>
    <Documento>1</Documento>
</Imagen>

```

Figura 4.33 XML del archivo "Mapa" del ejemplo

A continuación en la (Figura 4.34) el XML que se genera para el ejemplo del formato del archivo "Documento".

```

<?xml version="1.0"?>
<Documento xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///e:/XML%20Bueno/documento.xsd">
    <Version>1.0</Version>
    <Idioma>español</Idioma>
    <Documento Orden="1">
        <Estructura>
            <Accion>Accion1.xml</Accion>
            <Adjetivo>Adjetivo1.xml</Adjetivo>
            <EspacioA>EspacioA1.xml</EspacioA>
            <Imagen>Imagen1.xml</Imagen>
            <Mapa>Mapa1.xml</Mapa>
        </Estructura>
    </Documento>
</Documento>

```

Figura 4.34 XML del archivo "Documento" del ejemplo

4.4.4 La función de traducción

La función se realiza en el terminal servidor, utilizando el vocabulario de la base de datos, traduce los contenidos de los documentos “.vla” que se han generado por el sistema entre diferentes idiomas.

Por ejemplo, los archivos de XML en español que se mencionaron anterior, después de traducir, generan los archivos nuevos en chino como **sique**:

```
<?xml version="1.0"?>
<动作 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///e:/XML%20Bueno/accion.xsd">
  <指定>true</指定>
  <名词>
    <说明 A>
      <否定>false</否定>
      <疑问>false</疑问>
      <特指>false</特指>
    </说明 A>
    <说明 B>
      <单数>false</单数>
      <复数>false</复数>
      <少于>false</少于>
      <多于>false</多于>
      <大约>false</大约>
      <没有>false</没有>
      <少>false</少>
      <足够>false</足够>
      <多>false</多>
      <全部>false</全部>
    </说明 B>
    <单词>我</单词>
  </名词>
  <动词>
    <说明 C>
      <否定>false</否定>
      <疑问>false</疑问>
    </说明 C>
    <单词>住在</单词>
```



```

</动词>
<时间>
  <过去>false</过去>
  <现在>true</现在>
  <将来>false</将来>
</时间>
<宾语>
  <说明 A>
    <否定>false</否定>
    <疑问>false</疑问>
    <特指>false</特指>
  </说明 A>
  <说明 B>
    <单数>false</单数>
    <复数>false</复数>
    <少于>false</少于>
    <多于>false</多于>
    <大约>false</大约>
    <没有>false</没有>
    <少>false</少>
    <足够>false</足够>
    <多>false</多>
    <全部>false</全部>
  </说明 B>
  <单词>    </单词>
</宾语>
<目标>
  <说明 A>
    <否定>false</否定>
    <疑问>false</疑问>
    <特指>false</特指>
  </说明 A>
  <说明 B>
    <单数>false</单数>
    <复数>false</复数>
    <少于>false</少于>
    <多于>false</多于>
    <大约>false</大约>
    <没有>false</没有>
    <少>false</少>
    <足够>false</足够>

```

```

    <多>false</多>
    <全部>false</全部>
  </说明 B>
  <单词>    </单词>
</目标>
<空间>
  <说明 A>
    <否定>false</否定>
    <疑问>false</疑问>
    <特指>false</特指>
  </说明 A>
  <说明 B>
    <单数>false</单数>
    <复数>false</复数>
    <少于>false</少于>
    <多于>false</多于>
    <大约>false</大约>
    <没有>false</没有>
    <少>false</少>
    <足够>false</足够>
    <多>false</多>
    <全部>false</全部>
  </说明 B>
  <单词>莱昂</单词>
</空间>
<文件>1</文件>
</动作>

```

Figura 4.35 XML del archivo "Acción" del ejemplo en chino

```

<?xml version="1.0"?>
<形容词 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///e:/XML%20Bueno/adjetivo.xsd"
  <指定>false</指定>
  <本体>
    <说明 A>
      <否定>false</否定>
      <疑问>false</疑问>
      <特指>false</特指>
    </说明 A>
    <说明 B>

```

```

    <单数>>false</单数>
    <复数>>false</复数>
    <少于>>false</少于>
    <多于>>false</多于>
    <大约>>false</大约>
    <没有>>false</没有>
    <少>>false</少>
    <足够>>false</足够>
    <多>>false</多>
    <全部>>false</全部>
  </说明 B>
  <单词>莱昂</单词>
</本体>
<形容词>
  <说明 D>
    <否定>>false</否定>
    <疑问>>false</疑问>
  </说明 D>
  <说明 E>
    <少>>false</少>
    <足够>>false</足够>
    <非常>>false</非常>
  </说明 E>
  <单词>安静</单词>
  <单词>漂亮</单词>
</形容词>
<文件>1</文件>
</形容词>

```

Figura 4.36 XML del archivo "Adjetivo" del ejemplo en chino

```

<?xml version="1.0"?>
<空间 A xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///e:/XML%20Bueno/espacio1mi.xsd">
  <指定>>false</指定>
  <空间>
    <相关术语>北</相关术语>
    <说明 A>
      <否定>>false</否定>
      <疑问>>false</疑问>
      <特指>>false</特指>

```

```

</说明 A>
<说明 B>
  <单数>false</单数>
  <复数>false</复数>
  <少于>false</少于>
  <多于>false</多于>
  <大约>false</大约>
  <没有>false</没有>
  <少>false</少>
  <足够>false</足够>
  <多>false</多>
  <全部>false</全部>
</说明 B>
<单词>西班牙</单词>
</本体>
</空间>
<计量>
  <单词>  </单词>
  <数据>  </数据>
</计量>
<文件>1</文件>
</空间 A>

```

Figura 4.37 XML del archivo "EspacioA" del ejemplo en chino

```

<?xml version="1.0"?>
<图片 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///e:/XML%20Bueno/imagen.xsd">
  <指定>false</指定>
  <本体>
    <说明 A>
      <否定>false</否定>
      <疑问>false</疑问>
      <特指>false</特指>
    </说明 A>
    <说明 B>
      <单数>false</单数>
      <复数>false</复数>
      <少于>false</少于>
      <多于>false</多于>
      <大约>false</大约>

```

```

<没有>false</没有>
<少>false</少>
<足够>false</足够>
<多>false</多>
<全部>false</全部>
</说明 B>
<单词>莱昂</单词>
</本体>
<图片>string</图片>
<文件>1</文件>
</图片>

```

Figura 4.38 XML del archivo "Imagen" del ejemplo en chino

```

<?xml version="1.0"?>
<地图 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:///e:/XML%20Bueno/mapa.xsd">
  <指定 >false</指定>
  <本体>
    <说明 A>
      <否定>false</否定>
      <疑问>false</疑问>
      <特指>false</特指>
    </说明 A>
    <说明 B>
      <单数>false</单数>
      <复数>false</复数>
      <少于>false</少于>
      <多于>false</多于>
      <大约>false</大约>
      <没有>false</没有>
      <少>false</少>
      <足够>false</足够>
      <多>false</多>
      <全部>false</全部>
    </说明 B>
    <单词>西班牙</单词>
  </本体>
  <地图>string</地图>
  <文件>1</文件>
</地图>

```

Figura 4.39 XML del archivo "Mapa" del ejemplo en chino

4.5 El diseño de la base de datos

La base de datos de los sistemas de mensajería instantánea normalmente sirve para guardar las informaciones de registro de usuarios y las informaciones de los amigos de usuarios, sólo necesita dos bases de datos para guardarlos. Pero el sistema de mensajería instantánea que he diseñado en esta tesis doctoral tiene la función especial de traducción automática, por eso en el servidor existen cuatro bases de datos diferentes:

- La base de datos de usuarios, que es donde se almacena toda la información relativa a los mismos.
- La base de datos de lista de amigos, que es donde se almacena la lista de amigos de los usuarios.
- La base de datos de palabras VILA_1, donde se almacena las bases de datos de palabras, y sus traducciones.
- La base de datos de estructuras VILA_1, donde se almacena las estructuras VILA_1, definidas con el estándar XML Schema.

En la (Figura 4.40) se puede observar un esquema genérico de las bases de datos.

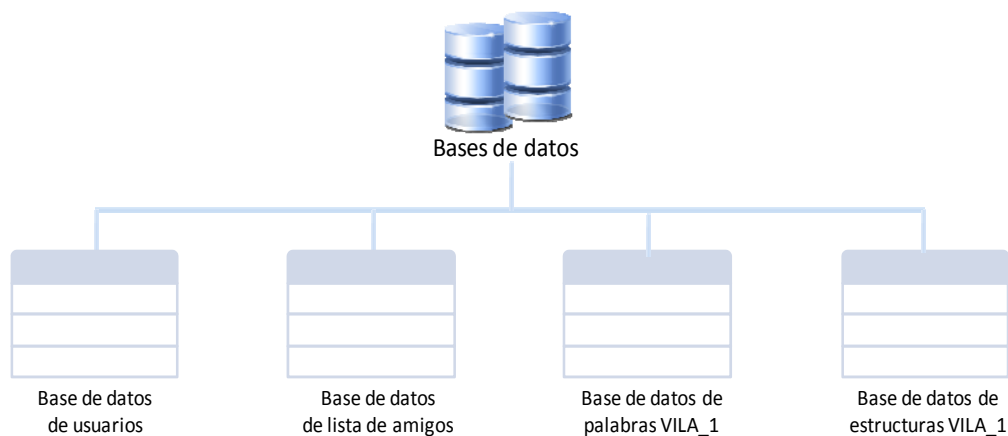


Figura 4.40 Esquema genérico de las bases de datos del sistema

4.5.1 La base de datos de usuarios

Las informaciones de usuarios incluyen: la dirección de correo electrónico de usuario, nombre de usuario, contraseña, estado de conexión, dirección de IP, imagen, sexo y etc. Entre ellos, la dirección de correo electrónico de usuario, nombre de usuario y contraseña son campos obligatorios, estado de conexión y dirección de IP los establecen automáticamente el sistema, y el resto son campos opcionales (*Figura 4.41*).

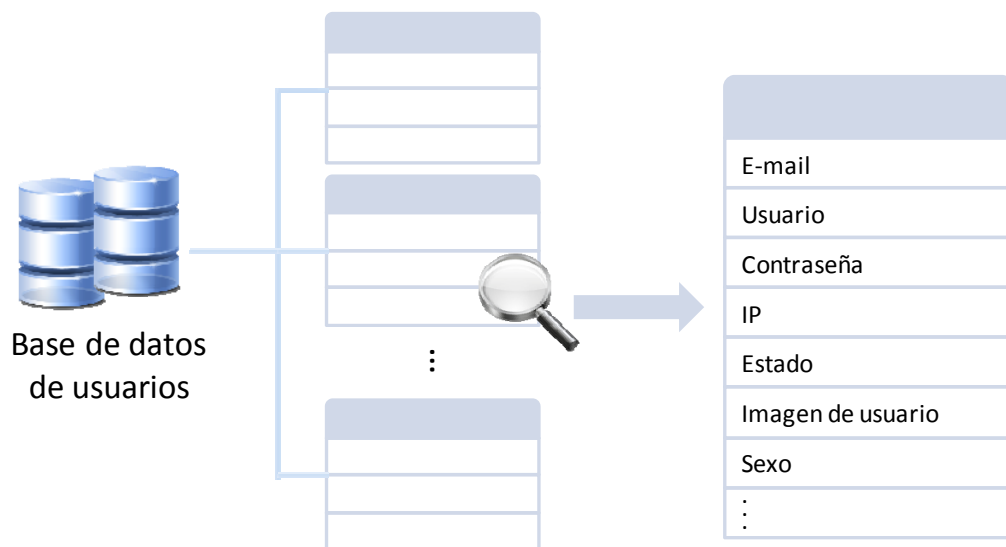


Figura 4.41 Estructura de una tabla de la base de datos de usuarios.

4.5.2 La base de datos de lista de amigos

La base de datos de lista de amigos, que es donde se almacena la lista de amigos de los usuarios. Debe incluir la dirección de correo electrónico de usuario y las direcciones de correo electrónico de los amigos, son campos obligatorios.

4.5.3 La base de datos de palabras VILA_1

La base de datos de VILA_1, es la encargada de almacenar las palabras que se van a utilizar en VILA_1, ya que la semántica de este

lenguaje es el lenguaje natural, como se explicó en el capítulo de introducción de esta tesis. En la tesis de Jia Fu hay detalles sobre esto. Estas bases de datos contienen los siguientes campos; (Figura 4.42); “ID”, “Español”, “Dominio”, “English”, “Domain”, “中文”, “领域”, y “Pinyin”.

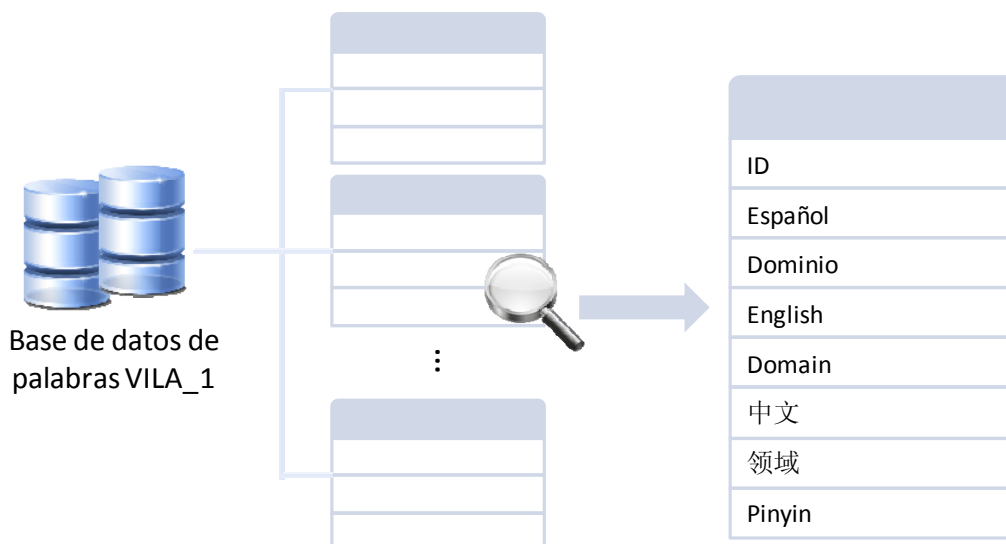


Figura 4.42 Estructura de una tabla de la base de datos de palabras VILA_1

El campo “ID”, es la clave principal de cada una de las tablas. Es un identificador exclusivo para cada fila, lo que garantiza la identificación exacta de cada una de las palabras, y de sus distintos significados.

Los campos “Español”, “English”, y “中文”, contienen las palabras en español, inglés, y chino, que tienen un mismo significado. Los campos “Dominio”, “Domain”, y “领域”, indican el dominio específico al que se refieren las palabras en cada momento, lo que permite resolver un gran número de problemas, ya que la polisemia se reduce de manera importante en palabras de un mismo dominio.

El campo “Pinyin” o “Hanyu Pinyin” [56], como se denomina correctamente es el “deletreo por sonido”, y el sistema de transcripción oficial del chino mandarín, en lugar de utilizar los

-
símbolos chinos, se usan letras del alfabeto latino para escribir el “sonido” de las palabras chinas. Se ha incorporado a las bases de datos porque es el sistema que permite escribir “chino mandarín” en el ordenador, y ordenar sus palabras alfabéticamente.

4.5.4 La base de datos de estructuras VILA_1

La base de datos de estructuras VILA_1, como se dijo anteriormente están constituidas por archivos XML Schema, que permiten definir las estructuras de los XML que definen cada componente del lenguaje, almacenándose por tanto un XML Schema por cada tipo de componente VILA_1, y un XML Schema “Documento”, como se explicará en la sección 4.4.3, almacena los distintos componentes, y la información del documento.

En el XML Schema a parte de definir la estructura del XML que tenga asignado, define los tipos de datos válidos para cada elemento y atributo, por lo que en este sentido las posibilidades de control sobre la estructura y los tipos de datos son muy amplias.

4.6 Conclusiones

En este capítulo se especifica la estructura del sistema de mensajería instantánea y el diseño de la interfaz de chat. Es que este sistema de mensajería instantánea tiene la función especial que es la traducción automática, así para satisfacer los requisitos hay unos diseños especiales. También por unos ejemplos, explica el uso del sistema. En resumen, soportar una manera que se puede permitir desarrollar una aplicación de tipo de mensajería instantánea que basado en el lenguaje VILA_1, y realizar la función de intercambio de mensajes entre los usuarios que utilizan diferentes idiomas.

CAPÍTULO 5

Validación empírica



*En este capítulo se
presentará el proceso de
evaluación del sistema,
lo que puede ayudar a
obtener una
retroalimentación para
mejorar el sistema.*



5.1 Introducción

A la hora de realizar la validación del sistema de mensajería instantánea basada en el lenguaje visual VILA_1 se ha llevado a cabo una validación orientada a usuario, donde se intenta recabar información sobre la validez de la aplicación desarrollada para comunicarse a través de Internet entre usuarios chinos y españoles.

Para ello se ha diseñado una metodología de prueba y validación que a continuación se explica.

Participantes Chinos				
Nombre	Sexo	Edad	Estudio de español(años)	Experiencia de internet (año)
Participante 1	Hombre	26	2	8
Participante 2	Hombre	24	2	5
Participante 3	Mujer	21	2	3
Participante 4	Mujer	23	2	4
Participante 5	Mujer	19	1	1
Participante 6	Mujer	20	1	2
Participante 7	Hombre	23	2	4
Participante 8	Mujer	23	2	5
Participante 9	Hombre	21	2	3
Participante 10	Mujer	22	2	3
Participante 11	Hombre	24	2	6
Participante 12	Hombre	21	2	2
Participante 13	Mujer	21	2	1
Participante 14	Hombre	23	2	4
Participante 15	Mujer	20	1	2
Participante 16	Hombre	24	2	8
Participante 17	Hombre	25	2	7
Participante 18	Mujer	18	1	1
Participante 19	Mujer	19	1	2
Participante 20	Hombre	19	1	3

Tabla 5.1 Participantes chinos

Participantes Españoles				
Nombre	Sexo	Edad	Estudio de chino (años)	Experiencia de internet (año)
Participante 1	Hombre	22	0	3
Participante 2	Hombre	28	0	5
Participante 3	Mujer	26	0	2
Participante 4	Hombre	25	0	3
Participante 5	Mujer	21	0	3
Participante 6	Hombre	25	0	6
Participante 7	Mujer	24	0	3
Participante 8	Mujer	23	0	3
Participante 9	Hombre	26	0	5
Participante 10	Mujer	21	0	2
Participante 11	Hombre	21	0	2
Participante 12	Mujer	20	0	1
Participante 13	Hombre	20	0	2
Participante 14	Mujer	19	0	1
Participante 15	Mujer	26	0	7
Participante 16	Mujer	21	0	2
Participante 17	Mujer	24	0	3
Participante 18	Hombre	20	0	3
Participante 19	Hombre	27	0	5
Participante 20	Mujer	23	0	2

Tabla 5.2 Participantes españoles

5.2 Metodología

La metodología de la validación se ha desglosado en una serie de etapas que a continuación de describen.

1. Selección de la población.
2. Diseño y ejecución dinámica de grupo.
3. Diseño, realización de encuestas para evaluación de la prueba.
4. Análisis de resultados y conclusiones.

5.2.1 Selección de la población

Para poder asegurar el correcto proceso de validación se debe partir de una serie de requisitos de los usuarios tales como:

- Los participantes de nacionalidad china tienen que entender los dos idiomas, chino y español. Estos requisitos son básicos, porque la traducción lleva implícita la lengua china y las tareas a realizar a la hora de resolver formularios se expresan en español.
- Los participantes tienen que tener los conocimientos sobre lingüística. Ya que la división de dominios del vocabulario y las tablas de la base de datos se refieren a los problemas del lenguaje natural.
- Los participantes tienen que tener experiencia de uso de las aplicaciones de Internet. Debido al hecho que las aplicaciones basadas en lenguaje VILA_1, están orientadas a Internet, los participantes deben conocer palabras del vocabulario habitual que se utilizan en la comunicación de Internet.

Según los requisitos, para realizar esta evaluación, se han creado dos grupos. Para el primero se han seleccionado 20 alumnos chinos y para el segundo 20 alumnos españoles de la Universidad de León. Cabe destacar que no se ha escogido ningún elemento de población relacionado con este trabajo de investigación.

Los miembros chinos de la población tienen como idioma nativo el chino, estos han estado establecidos en España de uno a tres años y han cursado cursos regulares de lenguaje español durante uno o dos años.

Los participantes españoles tienen como idioma nativo el español mientras que tienen un conocimiento nulo del chino.

Ambos grupos de participantes tienen suficiente experiencia en el uso de Internet y nuevas tecnologías.

5.2.2 Diseño y ejecución de la dinámica de grupo

El uso de una dinámica de grupo, parte de la eficacia de los resultados alcanzados por los psicólogos y psiquiatras en los métodos de terapias de grupo, ya que se entiende que el comportamiento del individuo se realiza dentro de un grupo, en la sociedad, en la familia, etc. e interactúa en él. Por lo tanto, para estudiar los actos de los potenciales usuarios del lenguaje, y del prototipo desarrollado no se debe aislar a éste, sino que deben desarrollarse en ambientes naturales en los que los usuarios pueden reflejar abiertamente su comportamiento. De acuerdo con la corriente germánica Gestalt [57], al grupo se le entiende como un todo, constituido por un conjunto de individuos que tienen necesidades y comportamientos comunes, el propio grupo es dinámico y por lo tanto evoluciona en el tiempo, sin un carácter estable [58].

Planificación de la dinámica de grupo

Una vez escogida la población dedicada a realizar la dinámica de grupo se ha llevado a cabo el diseño, planificación y ejecución de la misma.

La dinámica de grupo esta dividida en dos sesiones.

- Presentación herramienta software protagonista de esta dinámica: en esta fase se realiza una presentación y mini-tutorial del software, poniéndose en común las distintas cuestiones o dudas sobre el manejo de la misma que pueden aparecer.
- Prueba de funcionamiento entre los usuarios: se propone el intercambio de mensajes preestablecidos típicos de los sistemas de mensajería instantánea mediante la aplicación, entre usuarios de los dos grupos creados (grupo de estudiantes chinos y grupo de estudiantes españoles). De manera que para cada prueba se

escoge un usuario de cada grupo para que se realiza una comunicación típica basada en los entes emisor-receptor.

Respecto al moderador en ambas fases, se tomo la decisión de que no fuera ni el investigador que ha desarrollado esta tesis ni ninguno de los tutores u colaboradores, ya que podría condicionar los comentarios y observaciones de los participantes en la misma, generando un ambiente hostil.

Prueba de funcionamiento

La prueba de funcionamiento se diseñó de manera que individualmente cada uno de los participantes en la dinámica de grupo, se enfrentara al prototipo, e interactuara con él, intercambio mensajes con otro usuario del grupo complementario (diferente idioma). Para influenciar e interferir lo menos posible, y tratar de ser imparcial en la prueba, se tomó la decisión de elegir el contenido que se iba que se iba a enviar en VILA_1, así se ha propuesto 10 expresiones de las mas utilizadas en sistemas de mensajería instantánea (5 de español a chino y 5 de chino a español) a enviar mediante el sistema de mensajería basado en VILA_1, de manera que un interlocutor envíe un mensaje a otro con un idioma diferente y viceversa.

El proceso se resume en los siguientes pasos durante 5 iteraciones (una por cada expresión tipo):

1. Propuesta de expresión tipo a enviar.
2. Construcción y envío de la expresión mediante el sistema de mensajería.
3. Recepción de la expresión en VILA_1 por parte del interlocutor.
4. Puesta en común de la expresión y valoración por parte del modelador de la comprensión de la expresión (significado correcto de la expresión VILA_1 recibida) y la capacidad del emisor para la construcción de la misma.

En el último punto, el moderador evalúa si la expresión ha podido ser construida mediante el software y que dicha expresión ha sido entendida por el receptor, el resultado de la evaluación ha sido puntuado con 0 o 1 (bien o mal), para así evitar ambigüedades en los resultados.

Las pruebas se realizaron por parejas, con una duración máxima de 15 minutos, en el laboratorio F1, de la Escuela de Ingenierías de la Universidad de León, ya que éste aula cumplía con todos los requisitos necesarios para su correcto desarrollo.

El conjunto de expresiones escogidas para la realización de los 20 experimentos con los 40 usuarios fueron:

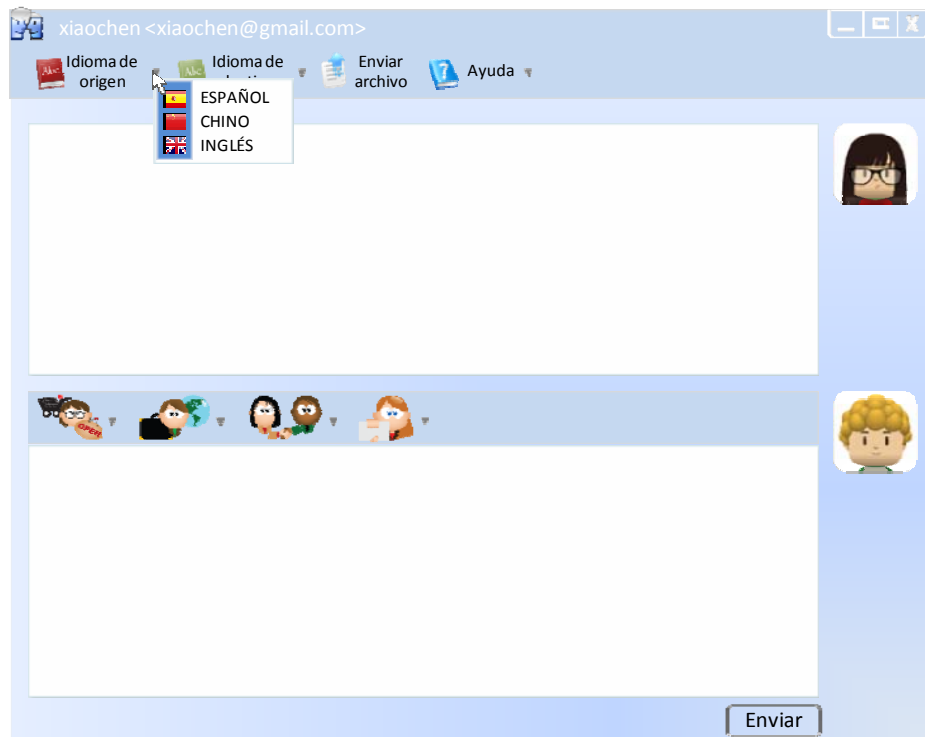
1. Mañana vamos a tomar café a las 7 por la tarde.
2. El billete de Madrid a León cuesta 26 euros.
3. Esta semana en León hace buen tiempo.
4. Soy profesor de informática.
5. Si el producto tiene problemas, el vendedor paga los gastos de devolución.
6. 预定一个双人间从 25 号开始持续 3 天。(Reservo una habitación doble desde el día 25 y durante 3 días.)
7. 圣周期间有什么活动吗?
8. 您从事什么职业?
9. 我已婚, 有两个小孩。
10. 牛仔裤多少钱一条?

A continuación se muestran dos ejemplos, el primero muestra el proceso de construcción y envío de la expresión “Mañana vamos a tomar café a las 7 por la tarde” llevado a cabo por un usuario español. El segundo ejemplo muestra el proceso de construcción y envío de una expresión “预定一个双人间从 25 号开始持续 3 天。”.

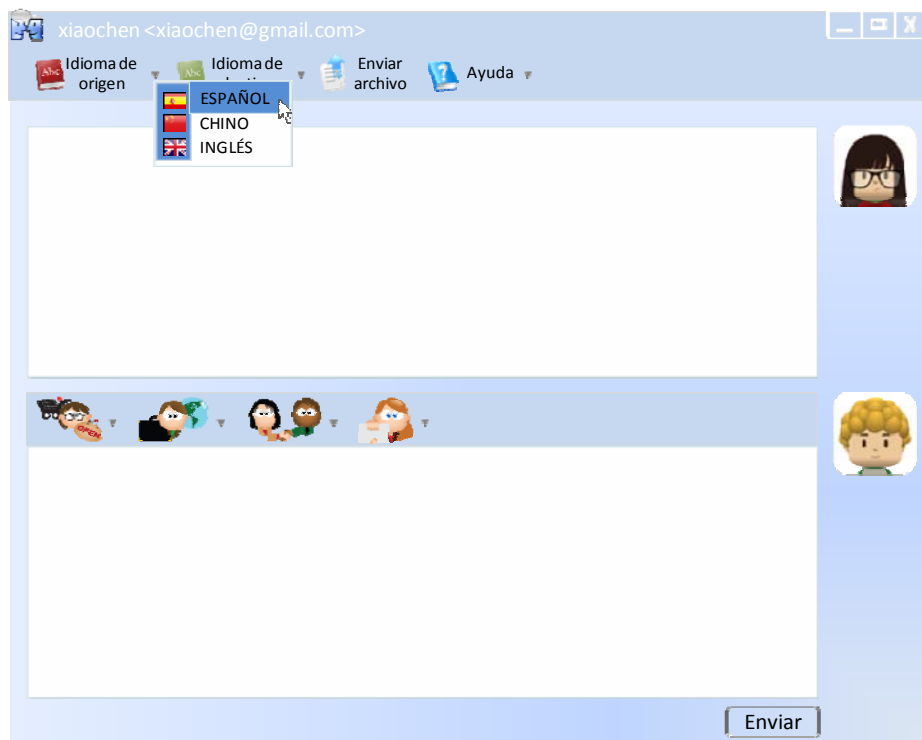
En el pie de cada captura se estable la acción asociada a cada un de las 11 fases que estable un proceso de construcción-traducción y envío.

Ejemplo 1. Usuario español enviado expresión a usuario chino:

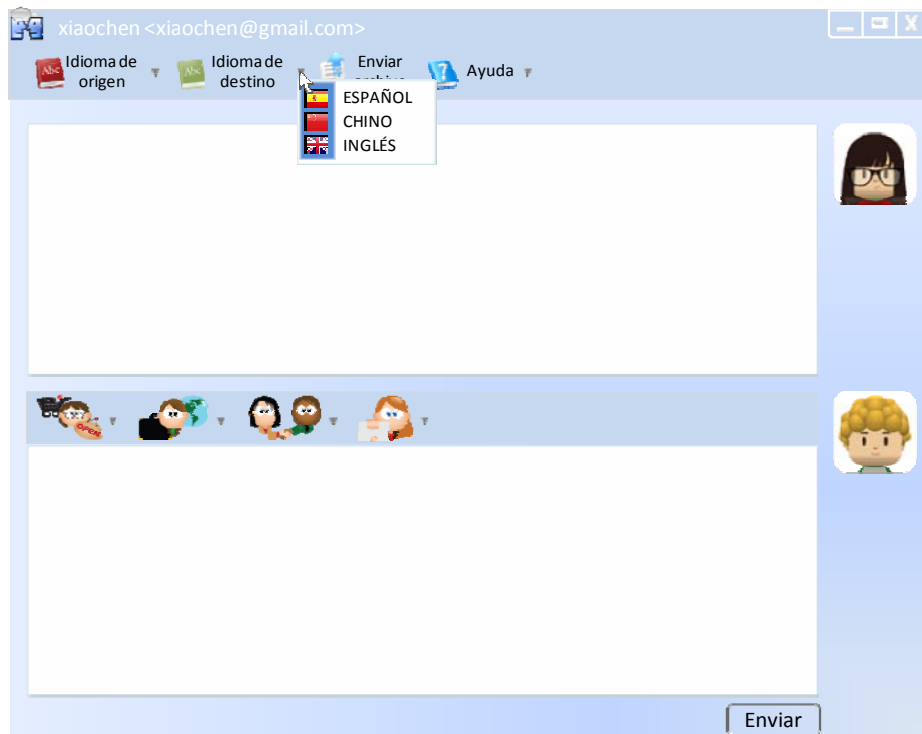
Frase: “Mañana vamos a tomar café a las 7 por la tarde”.



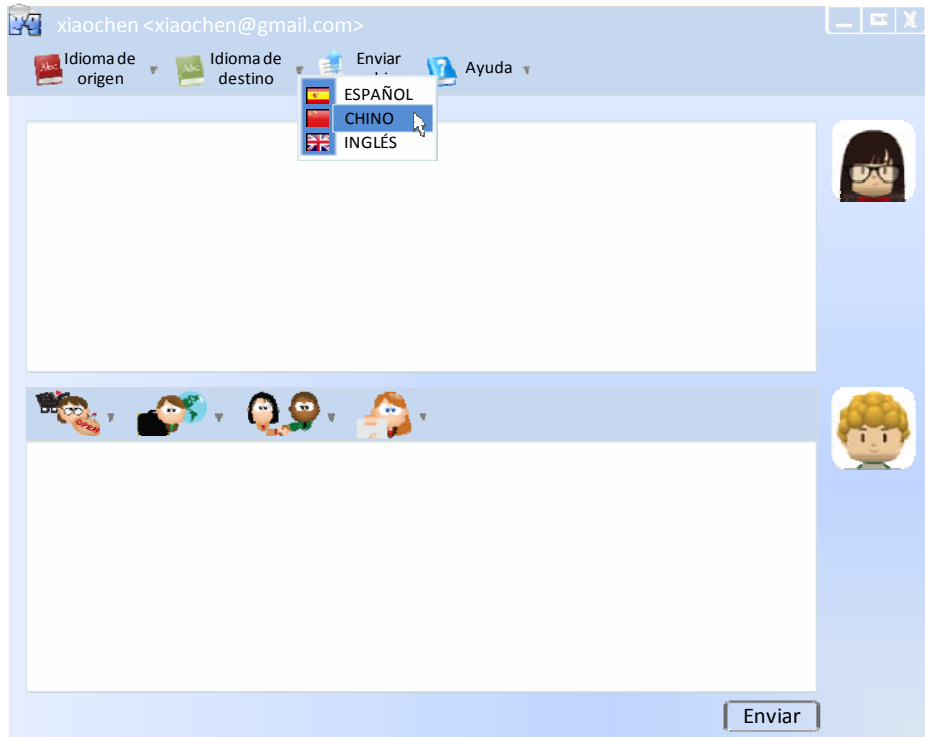
Fase 1: Seleccionar el menú de idioma de origen



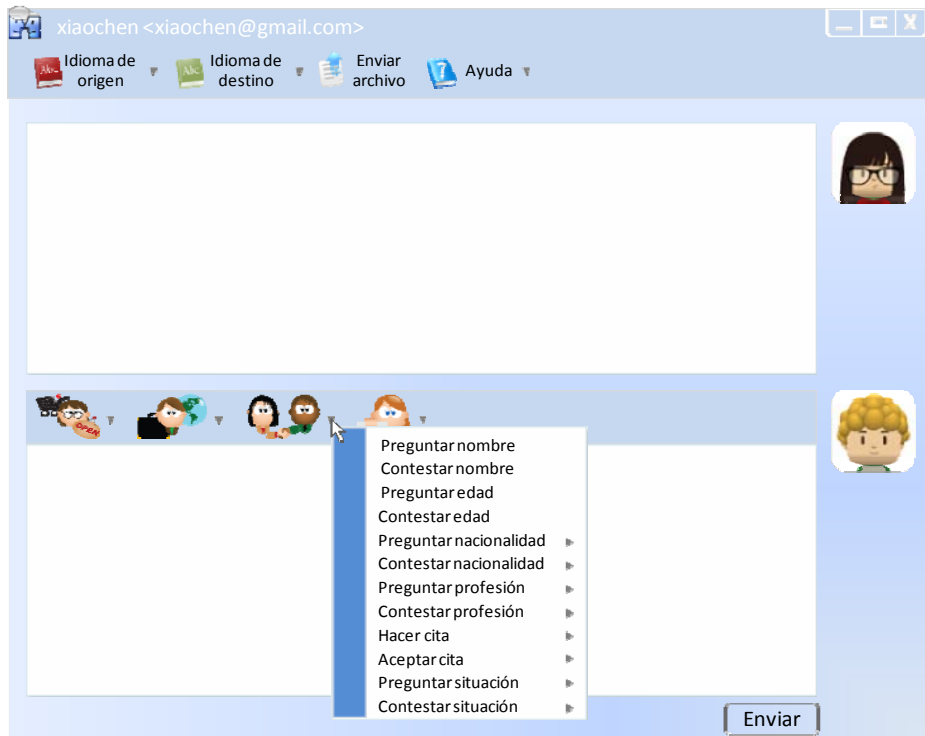
Fase 2: Elegir el español como el idioma de origen



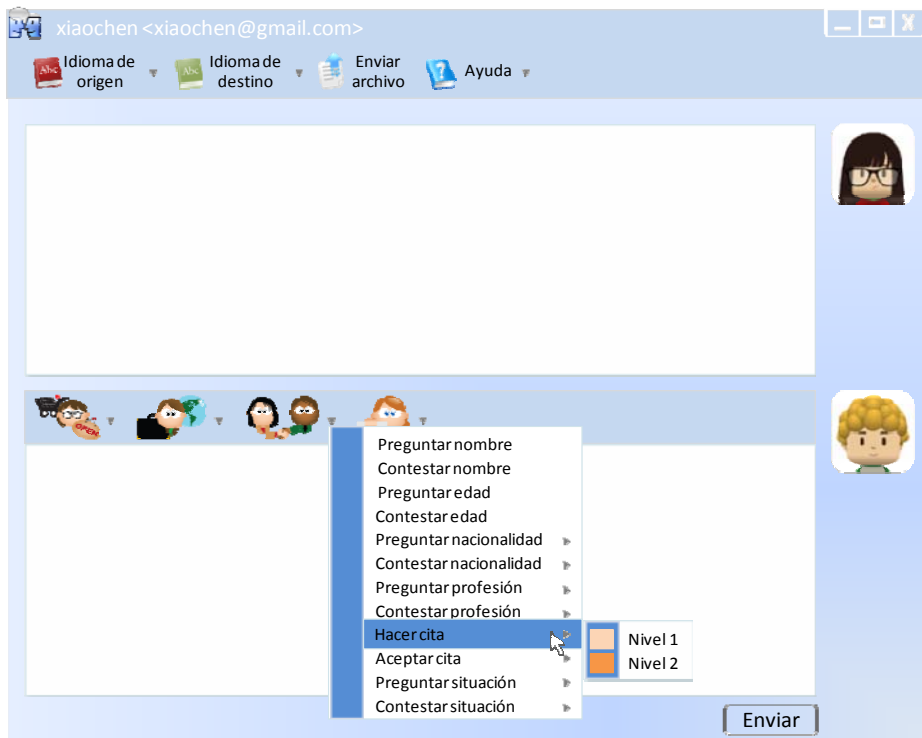
Fase 3: Seleccionar el menú de idioma de destino



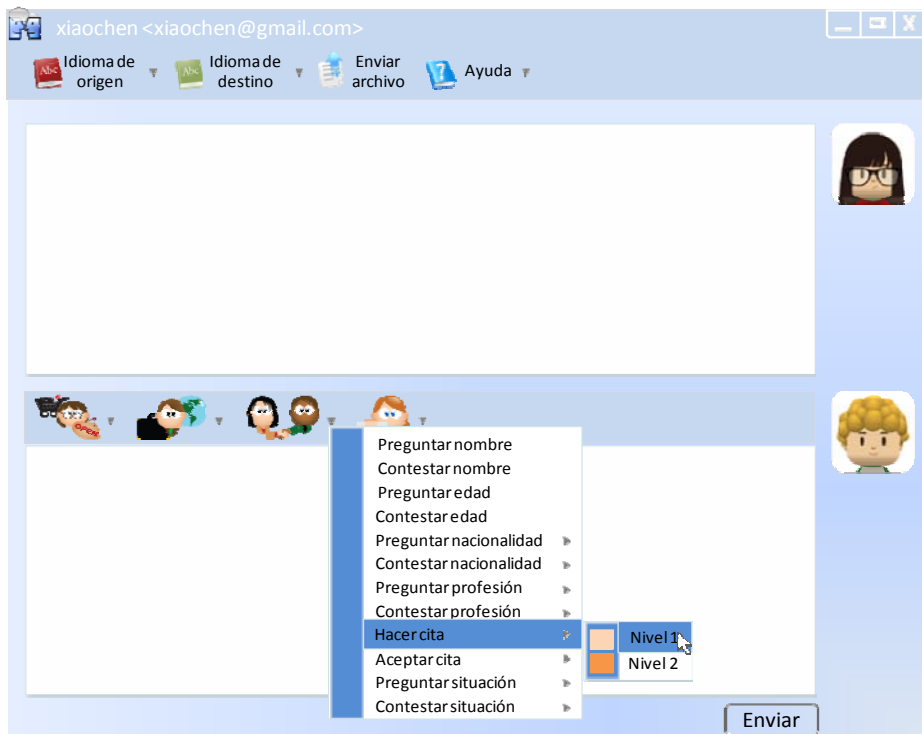
Fase 4: Elegir el chino como idioma destino



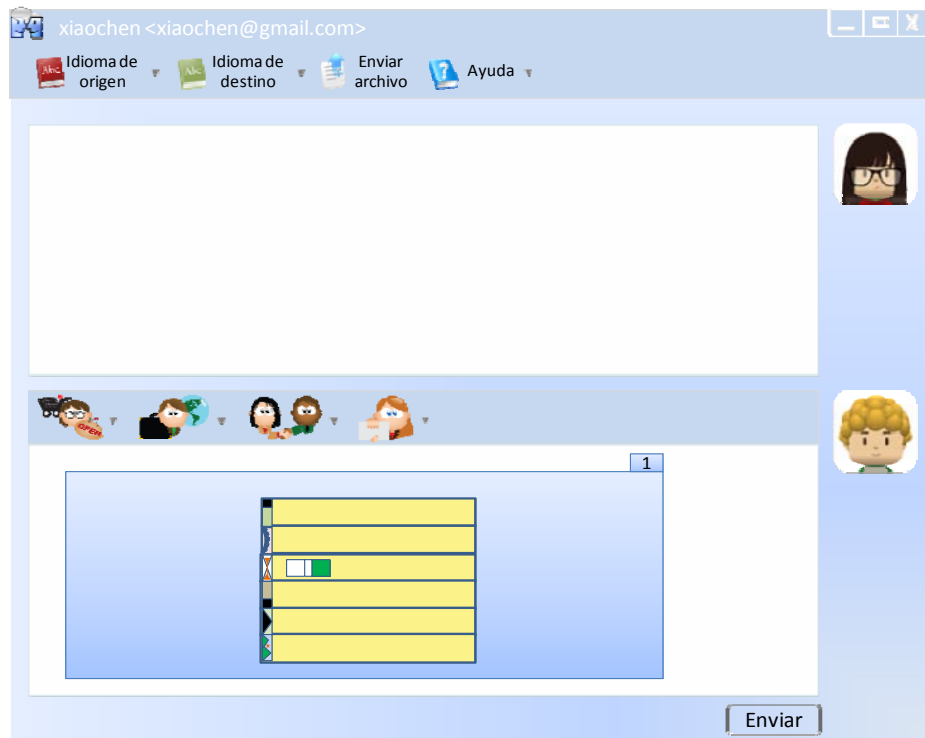
Fase 5: Elegir el menú de amistad de los mensajes comunes



Fase 6: Elegir el tipo de mensaje “hacer cita”



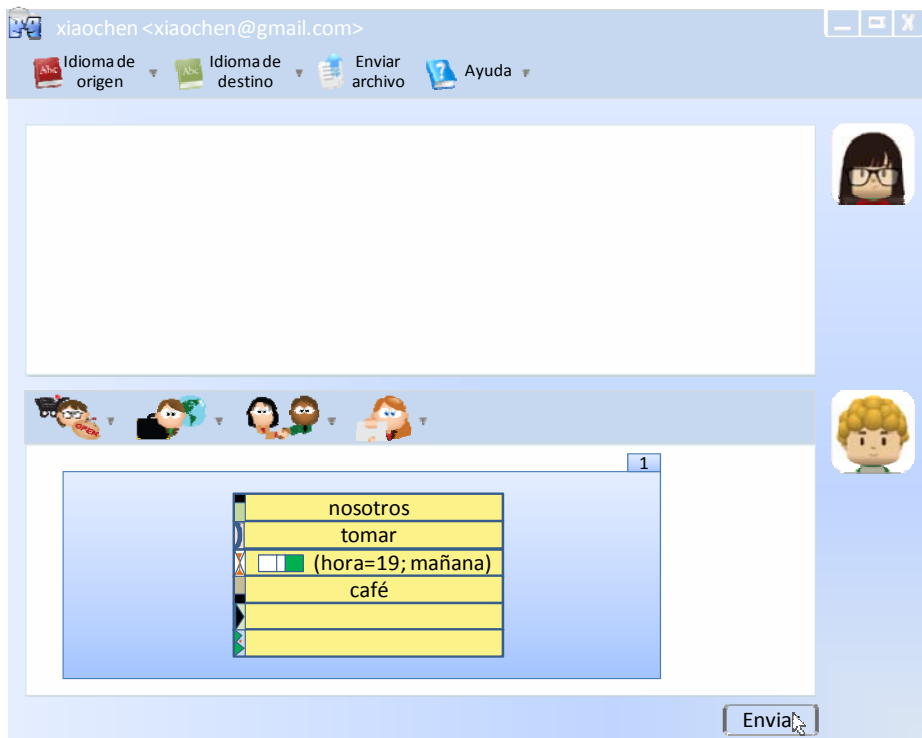
Fase 7: Elige el nivel 1 del tipo de mensaje “hacer cita”



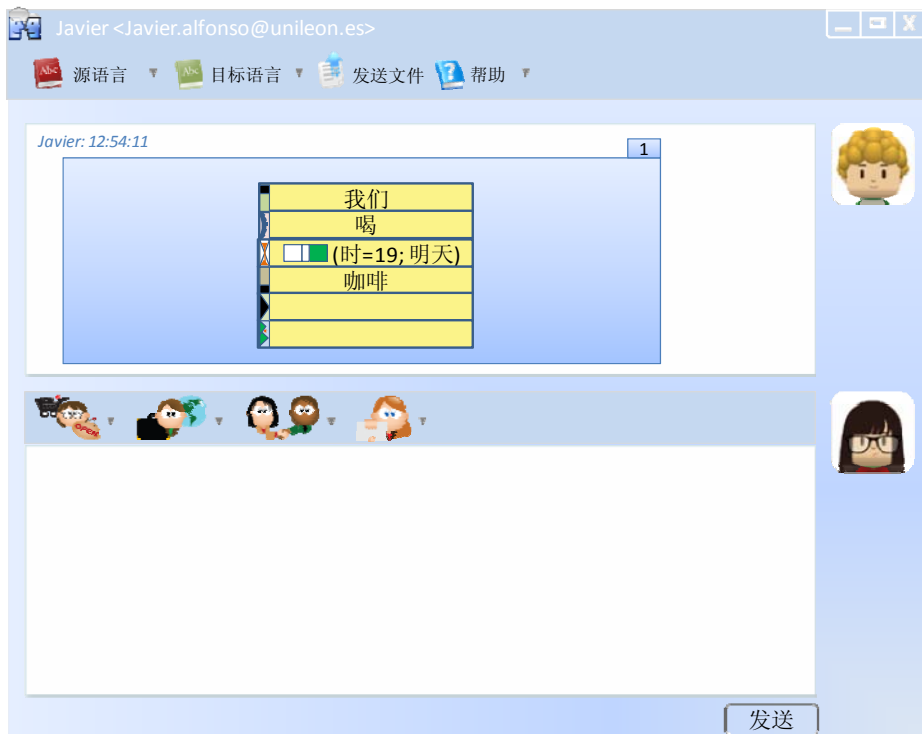
Fase 8: Salir de la ventana de mensaje



Fase 9: El usuario rellena los elementos del mensaje



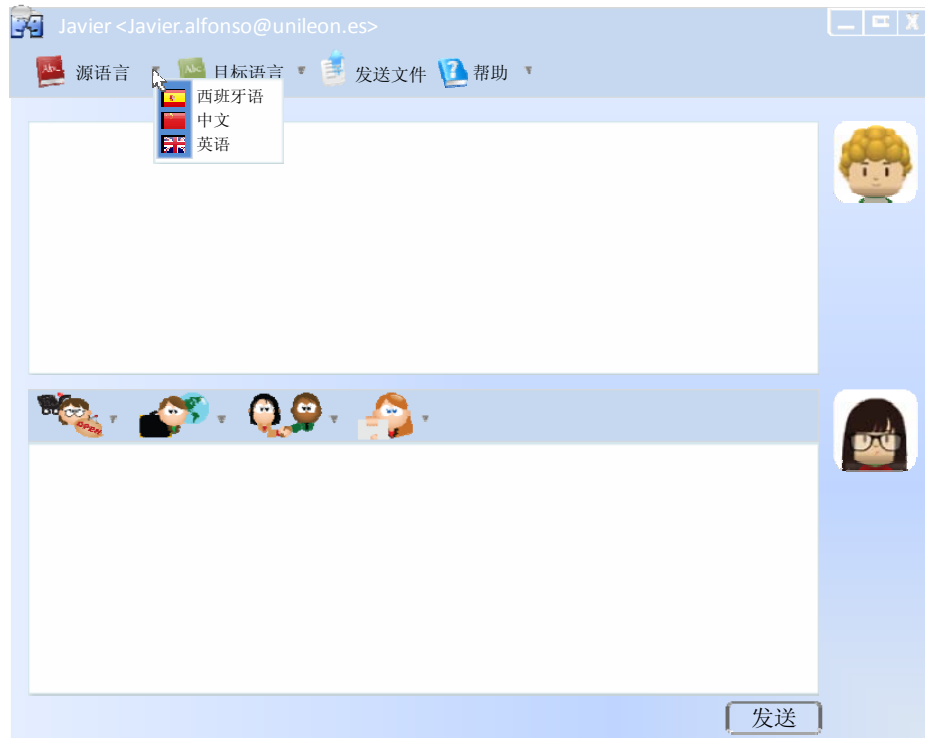
Fase 10: Clic en el botón para enviar el mensaje



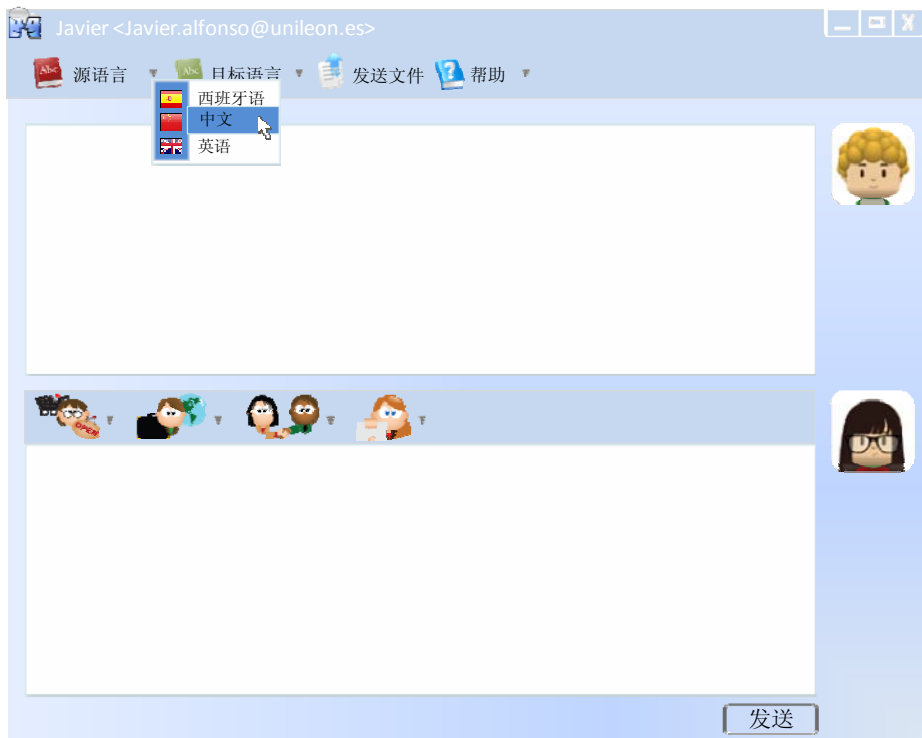
Fase 11: La frase se presenta en chino en la interfaz en el usuario destino chino

Ejemplo 2. Usuario chino enviando expresión a usuario español:

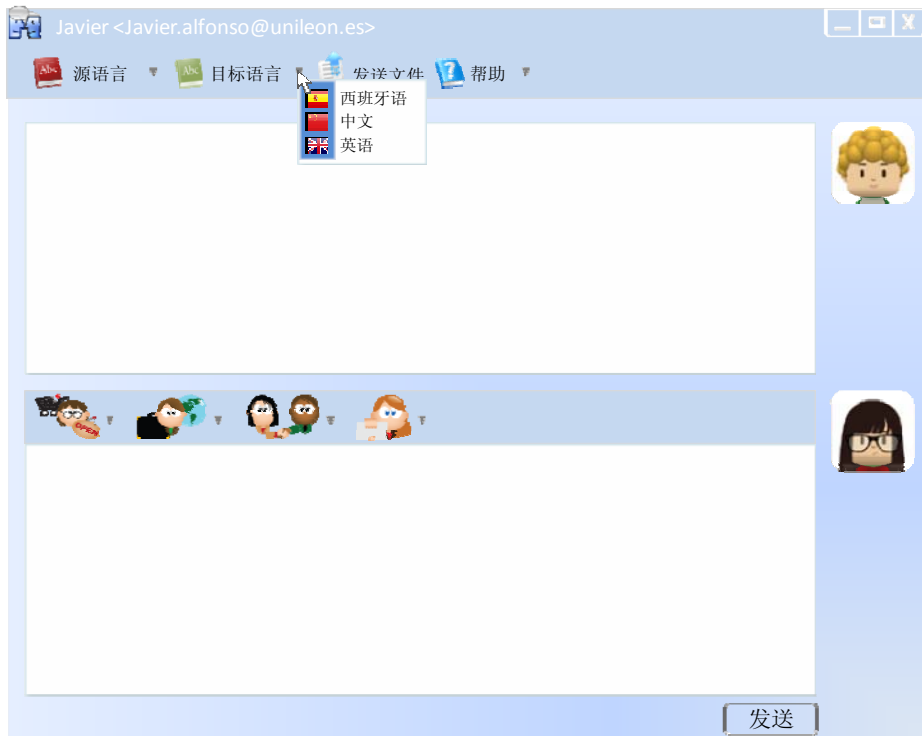
Frase 6: 预定一个双人间从 25 号开始持续 3 天。



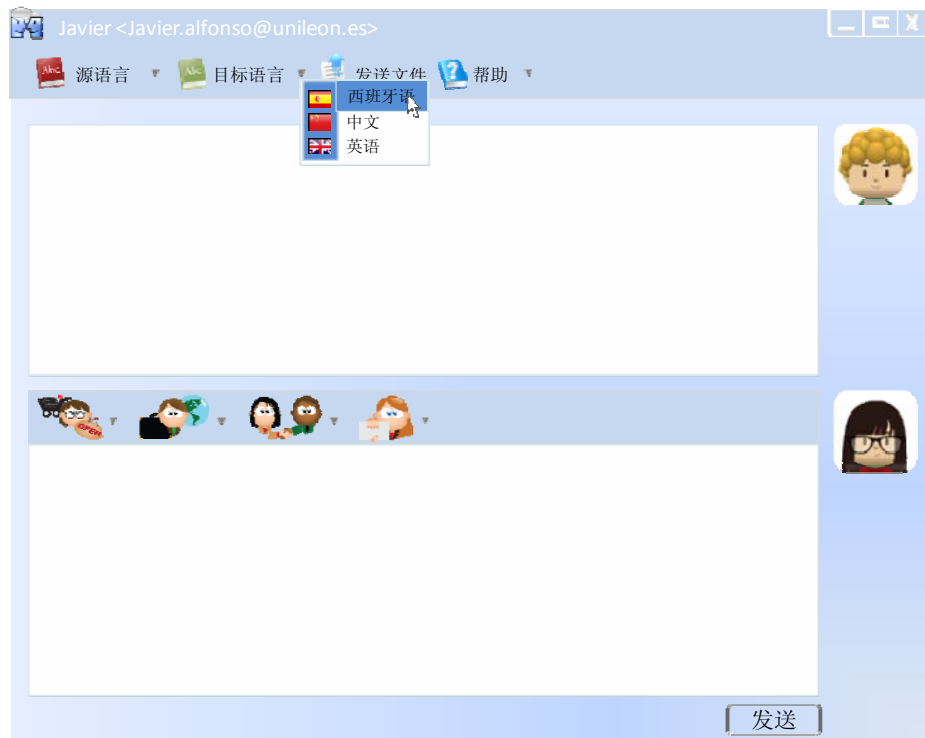
Fase 1: Seleccionar el menú de idioma de origen



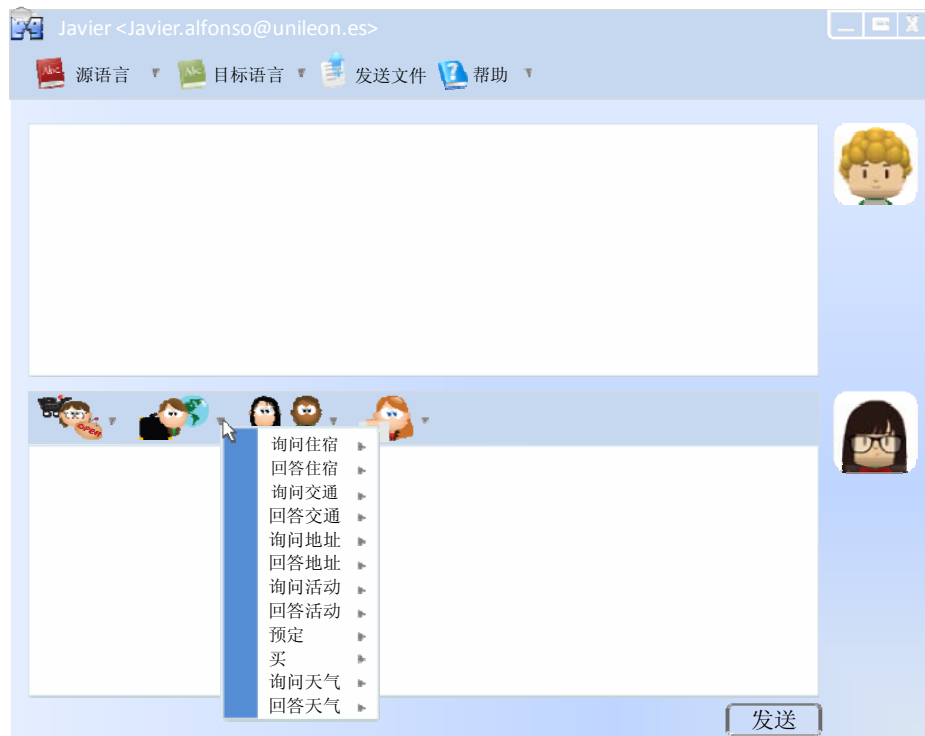
Fase 2: Seleccionar el chino como el idioma de origen



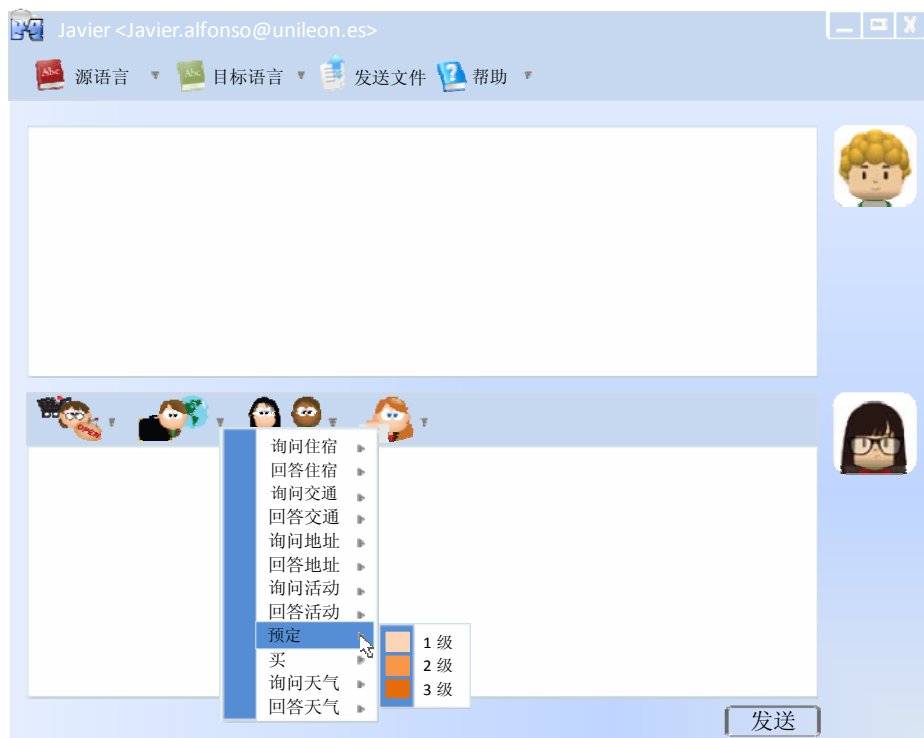
Fase 3: Seleccionar el menú de idioma de destino



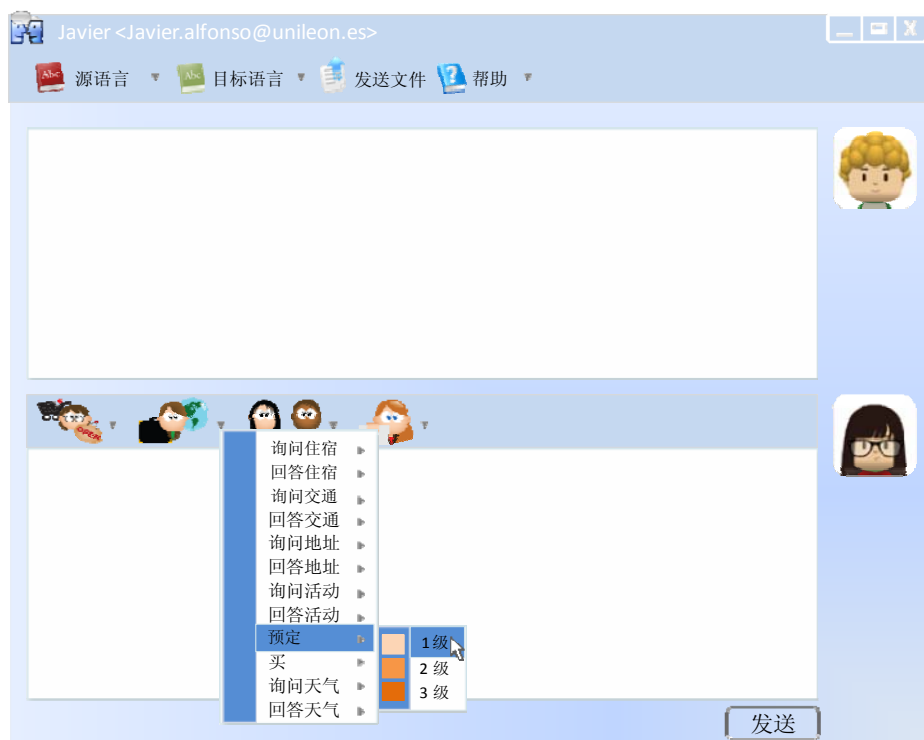
Fase 4: Elegir el español como idioma de destino



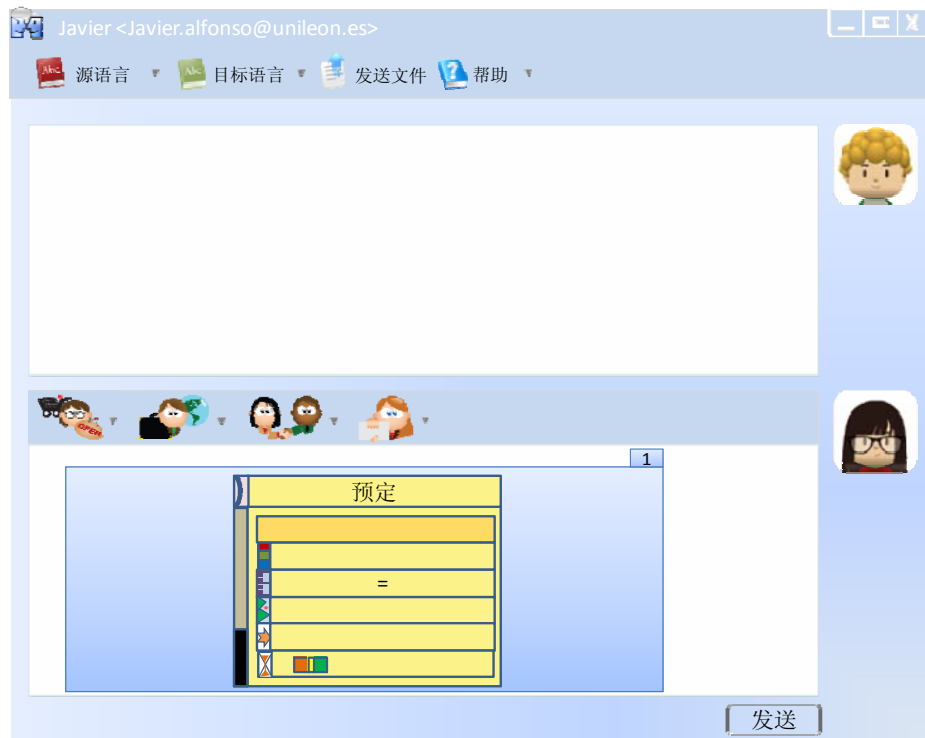
Fase 5: Clic el menú de viaje de los mensajes comunes



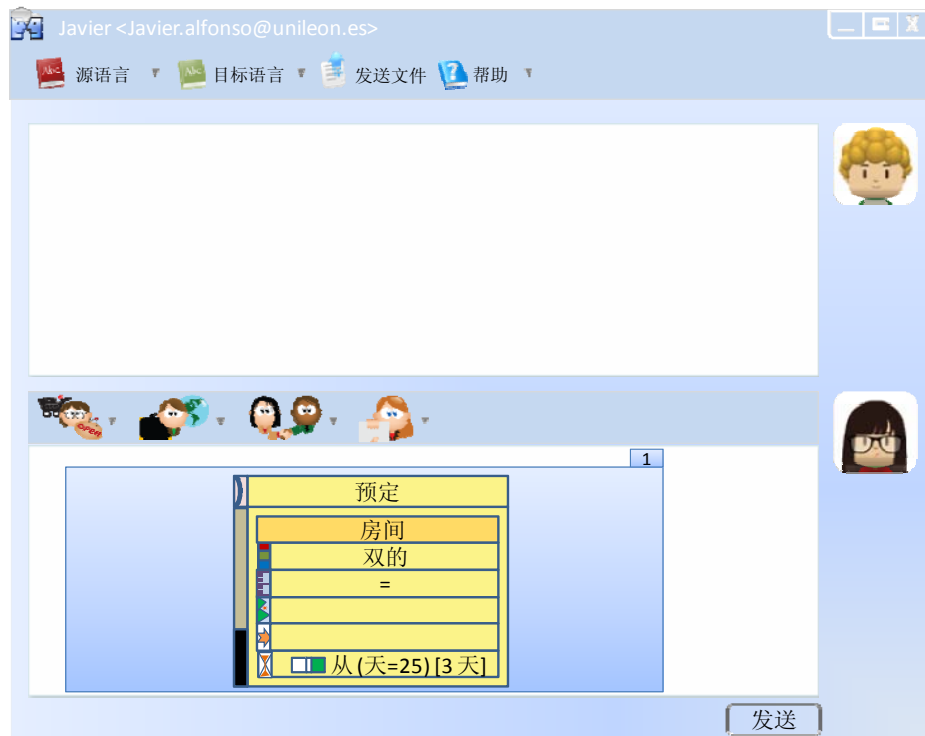
Fase 6: Elegir el tipo de mensaje “Reservar”



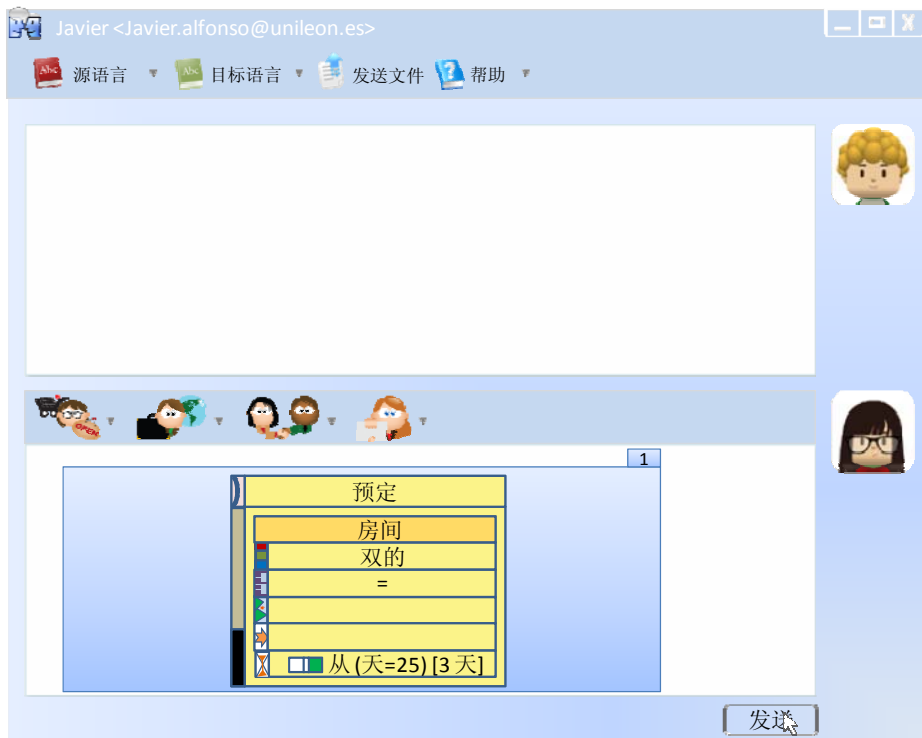
Fase 7: Elegir el nivel 1 del tipo de mensaje “Reservar”



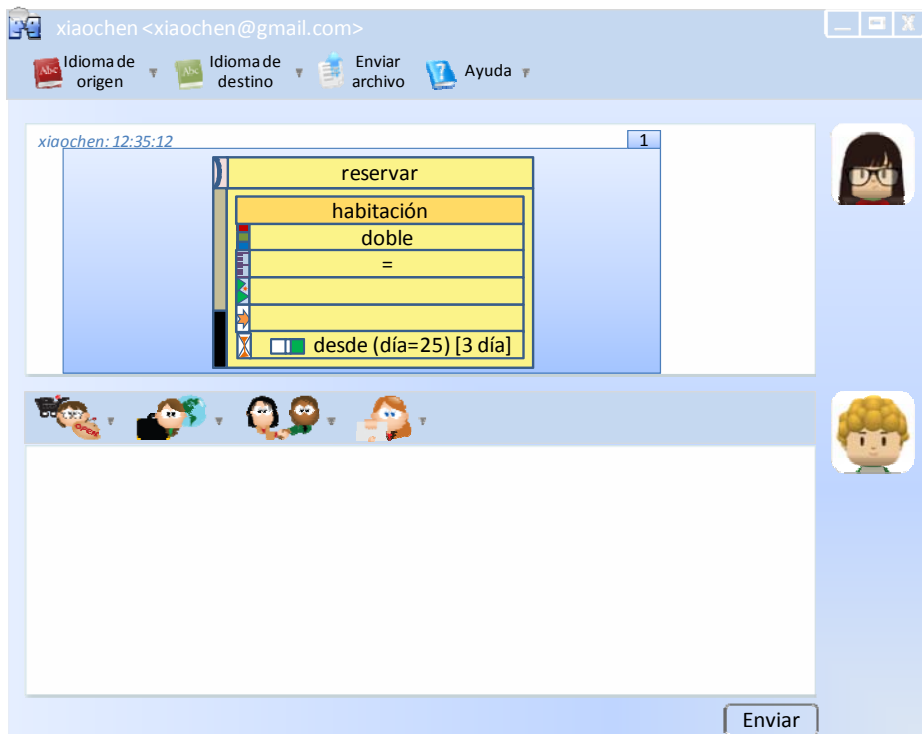
Fase 8: Aparición de la ventana de mensaje



Fase 9: El usuario rellena los elementos del mensaje



Fase 10: Clic el botón para enviar el mensaje



Fase 11: La frase se presenta en español en la interfaz en el usuario destino español

5.3 Diseño, realización de encuestas para evaluación de la prueba

Para evaluar las pruebas de funcionamiento, se decidió por una parte realizar dos encuestas y por otra analizar los datos de la dinámica de grupo con respecto a la evaluación de comprensión de cada expresión.

Encuestas

Con respecto a las encuestas se necesitaba obtener información cuantitativa, que permitiera establecer conclusiones robustas, y de la forma más instantánea posible. El método elegido para realizar las encuestas, fue la entrevista personal, debido principalmente a las siguientes ventajas; el porcentaje de respuesta es alto, no influyen terceros en las respuestas, las preguntas se pueden adaptar al nivel cultural del entrevistado, se conoce el hábitat del encuestado; se puede emplear material auxiliar, y existe la posibilidad de obtener un mayor número de información.

Las encuestas fueron desarrolladas de manera personal a cada participante, después de terminar la prueba de funcionamiento.

Las pautas seguidas a la hora de diseñar el cuestionario, fueron las siguientes: realizar preguntas concretas, evitar términos ambiguos que puedan inducir a interpretaciones erróneas, evitar cálculos, no realizar preguntas que contengan potencialmente la respuesta, evitar preguntas embarazosas o difíciles, y seguir un orden en las preguntas.

Las preguntas elegidas para la primera encuesta (valoración del funcionamiento), se presentan en la (Tabla 5.3).

EVALUCIÓN DEL FUNCIONAMIENTO DEL PROTOTIPO	1	2	3	4	5	6	7	8	9	10
1. El software resulta de gran utilidad										
2. Aprender el uso de este software es fácil										
3. El software responde de manera eficiente										
4. Los mensajes comunes son útiles										
5. Trabajar con este software es satisfactorio										
6. Los mensajes de error de este software son adecuados										
7. El software funciona correctamente										
8. Recomendaría este software a otras personas										
9. ¿ Desea realizar alguna observación ?										
Valora de 1 a 10 las siguientes afirmaciones (1: Totalmente en desacuerdo – 10: Totalmente de acuerdo)										

Tabla 5.3 Encuesta para la evaluación del funcionamiento del prototipo.

Las preguntas elegidas para la segunda encuesta (valoración del diseño de la interfaz de usuario), se presentan en la (Tabla 5.4).

EVALUCIÓN DEL INTERFAZ USUARIO DEL PROTOTIPO	1	2	3	4	5	6	7	8	9	10
1. El interfaz resulta atractivo visualmente										
2. El interfaz de usuario es intuitivo										
3. El interfaz responde de manera rápida										
4. Resulta difícil encontrar algunas opciones										
5. El interfaz de ayuda resulta útil										
6. El interfaz de usuario es personalizable										
7. Los componentes gráficos resultan vistosos										
8. Los componentes gráficos resultan intuitivo										
9. ¿ Desea realizar alguna observación ?										
Valora de 1 a 10 las siguientes afirmaciones (1: Totalmente en desacuerdo – 10: Totalmente de acuerdo)										

Tabla 5.5 Encuesta para la evaluación de la interfaz de usuario del prototipo.

Cada una de las encuestas anteriores está formada por nueve preguntas, ocho de escala unipolar, en las que se pide cuantificar de 1 a 10 el nivel de acuerdo o de desacuerdo con cada una, y una pregunta abierta, con la que se pretende recopilar información adicional.

En ambas encuestas se ha incluido un mecanismo de control, con el que se tratará de evitar respuestas automáticas, y poco razonadas, este mecanismo consiste en intercalar afirmaciones positivas, y afirmaciones negativas en las preguntas, es decir, en unas preguntas estar de acuerdo se considerará positivo, y en otras será considerado negativo.

Este sistema de control es reflejado en la metodología de evaluación de resultados, considerándose en unas preguntas el 10 como el mejor resultado, y el 1 como el peor, y en otras el 1 como el mejor resultado, y el 10 como el peor.

5.4 Análisis de resultados y conclusiones

Encuestas

Una vez terminado el proceso de evaluación, se procedió al tratamiento de la información, para lo cual se utilizó el software estadístico SPSS [59], ya que permite un análisis exhaustivo, y pormenorizado de la información.

Las pruebas de funcionamiento del prototipo de mensajería han obtenido una calificación media de 8.0125 puntos, lo que supone un resultado muy positivo, pero también significa que es necesario realizar mejoras, y replantear algunos aspectos del prototipo que un principio se habían considerado óptimos, como la satisfacción al trabajar con el software, que obtuvo la calificación más baja con 6.1 puntos, esta baja puntuación en este apartado puede justificar porque la inmensa mayoría de usuarios de prueba están acostumbrados a

trabajar con MSM Windows Live. En la (Figura 5.1) se puede observar un gráfico con los resultados completos de la prueba de evaluación de funcionamiento.

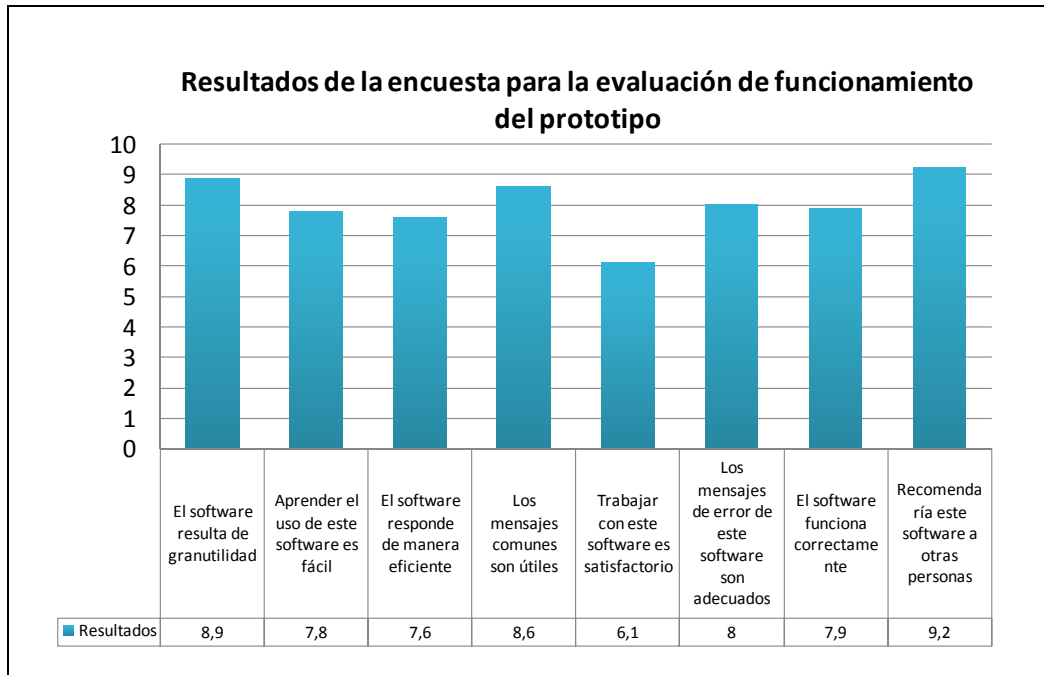


Figura 5.1 Gráfico de resultados de la evaluación del funcionamiento del prototipo

Los resultados de las pruebas de evaluación de la interfaz de usuario (Figura 5.2), al igual que los de la prueba de evaluación de funcionamiento son muy positivos, con una calificación media de 7.875 puntos, obteniendo como conclusión que aunque la puntuación es alta, es necesario considerar algunos factores, como por ejemplo, los niveles de personalización del prototipo. Una vez más, la expansión del MSN Windows Live, hace que otros sistemas de mensajería resulten más difíciles utilizar a la hora de encontrar las opciones de usuario.

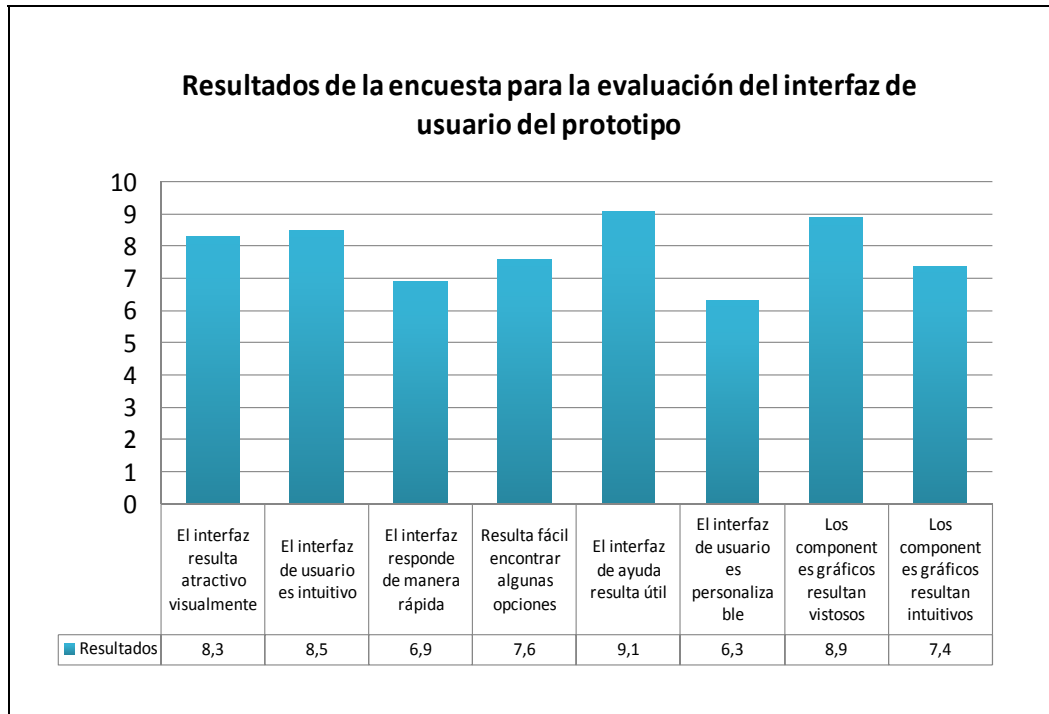


Figura 5.2 Gráfico de resultados de la evaluación de la interfaz de usuario del prototipo

Respecto a la pregunta, de si deseaban realizar alguna observación, las respuestas no han aportado datos nuevos, pero sí que han permitido matizar los resultados del estudio.

Comprensión de expresiones

A continuación se muestra la evaluación que define si los usuarios han podido enviar y recibir de manera correcta el conjunto de expresiones de tipo preestablecidas en sistemas de mensajería instantánea. Recordar que estos datos fueron tomados por parte del moderador en al prueba de funcionamiento.

La puntuación será de 1 si se realizan con éxito las siguientes tareas:

- Construcción y envío de la expresión VILA_1 español/VILA_1 chino mediante la interfaz de la aplicación.
- Comprensión de la expresión VILA_1 español/VILA_1 chino recibida.

Los siguientes gráficos muestran el numero de comunicaciones realizadas con éxito en entre cada parejas de usarlos a partir del conjunto de frases prediseñadas.

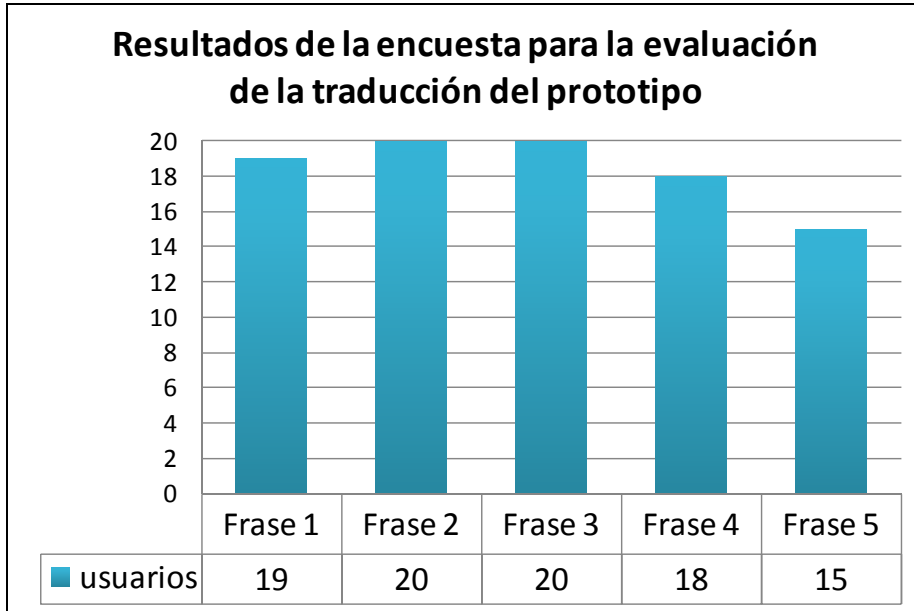


Figura 5.3 Gráfico de resultados de la evaluación de la traducción de las frases desde español a chino

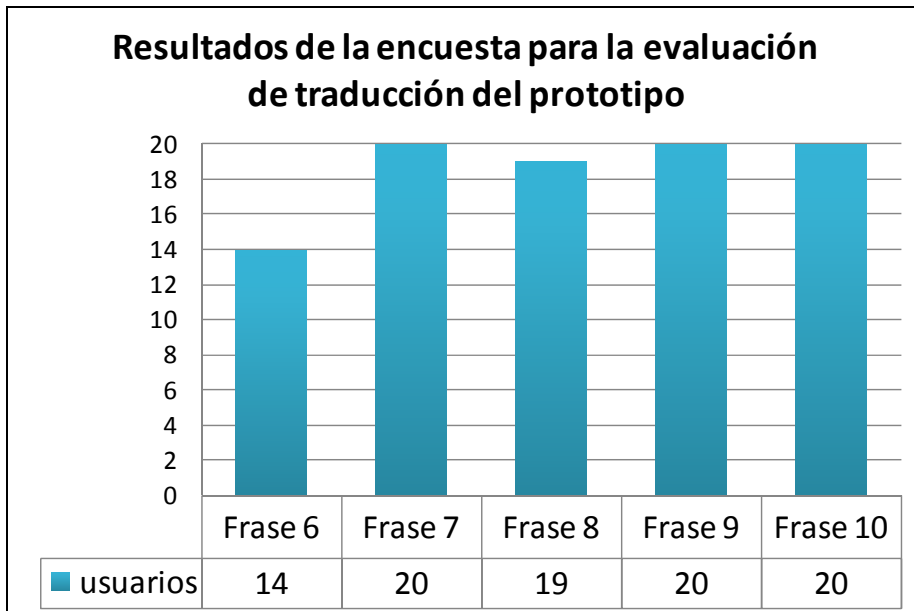


Figura 5.4 Gráfico de resultados de la evaluación de la traducción de las frases desde chino a español

Como se puede observar existen dos frases que coinciden con las más complejas a priori, semánticamente hablando, frases 5 y 6 respectivamente, donde se han producido los mayores problemas de comprensión. Sin embargo, analizando el conjunto global de resultados, se percibe un funcionamiento altamente positivo del proceso de construcción y envío de cada expresión, ya que en 85% de las expresiones enviadas son comprendidas en un porcentaje alto por parte de los 20 clientes que reciben la información en el experimento.

CAPÍTULO 6

Conclusiones finales y trabajos futuros



En este capítulo se detallan las conclusiones finales y las principales aportaciones realizadas en el desarrollo de la presente tesis doctoral. Se indican también algunas de las posibles líneas futuras de investigación que pueden llevarse a cabo.



6.1 Conclusiones finales

Las conclusiones de esta tesis son las siguientes:

- Se ha comprobado que VILA_1 es un lenguaje muy apropiado para expresar mensajes cortos.
- Dado que los mensajes cortos pueden expresarse con poca complejidad lingüística, el intercambio de mensajes entre las versiones china y española de VILA_1 resulta muy eficiente.
- Si en los mensajes cortos se usan expresiones idiomáticas o de uso coloquial se plantean problemas que dificultan la comprensión del mensaje al trasladarlo a otro idioma.
- Se ha comprobado que, dado que la tasa de éxito en mensajes sin complejidad lingüística es alto, el sistema resulta útil para ser usado en aplicaciones de comercio electrónico, mensajería personal, redes sociales, etc.

6.2 Trabajos futuros

Sobre las ideas y experiencias obtenidas en esta presente tesis doctoral, hay gran variedad de trabajos que pueden desarrollarse. Por ejemplo se citan siguientes:

- Desarrollar una aplicación especial para los negocios electrónicos, de forma que los clientes se pueden comunicar más fácil y cómodo, evitando los problemas de idioma.
- Se puede ampliar la plataforma de uso. Por ejemplo trasplantar unas aplicaciones a móvil, PDA, GPS y etc. En este caso los usuarios podrían utilizar esta aplicación en cualquier sitio y tiempo.
- Explorar y comprobar todas las posibles aplicaciones que el lenguaje VILA_1 puede tener, entre las que se pueden citar la traducción automática, mejorar la generación automática de

resúmenes, la creación de interfaces de usuario para personas con algún tipo de diversidad funcional.

- Se pueden añadir más idiomas. Hacer investigación en otros idiomas, mejorar la gramática del lenguaje VILA_1, desarrollar esta aplicación con nivel más alto de internacionalización.
- Se puede promover el uso de esta aplicación, y también en la aplicación incluir las reacciones de los usuarios, o hacer encuestas regularmente, para recoger las informaciones y mejorarla.

Abreviaturas

-

AIM – America-On-Line Instant Messenger.
API – Application Programming Interface.
CORBA – Common Object Request Broker Architecture.
DCOM – Distributed Component Object Model.
GAIM – GTK+ AOL Instant Messenger.
GPL – General Public License.
GUI – Graphical User Interface.
HTTP – Hypertext Transfer Protocol.
IA – Inteligencia Artificial.
ICQ – I seek you.
IM – Instant Messenger.
ISO – International Organization for Standardization.
ISST – Italian Syntactic-Semantic Treebank.
JDBC – Java Database Connectivity.
JNLP – Java Networking Launching Protocol.
KPI – Key Performance Indicator.
MIME – Multipurpose Internet Mail Extensions.
MSN – MicroSoft Network.
OASIS – Organization for the Advancement of Structured Information Standards.
ODBC – Open DataBase Connectivity.
OMG – Object Management Group.
OWL – Ontology Web Language.
P2P – Point to Point.
PC – Personal Computer.
PDA – Personal Digital Assistant.
PLATO – Programmed Logic for Automated Teaching Operations.
RMI – Java Remote Method Invocation.
RPC – Remote Procedure Call.
SASL – Simple Authentication and Security Layer.
SGML – Standard Generalized Markup Language.
SMTP – Simple Mail Transfer Protocol.
SOA – Service Oriented Architecture.
SOAP – Simple Object Access Protocol.
SQL – Structured Query Language.
SVO – Sujeto Verbo Objeto.

-

TLS – Transport Layer Security.

UDDI – Universal Description Discovery and Integration.

UIN – Universal Internet Numver.

UML – Unified Modeling Language.

USB – Universal Serial Bus.

UTF-8 – 8 Bit Unicode Transformation Format.

VILA – Visual Language.

VILA_1 – Visual Language 1.

VoIP– Voz sobre Protocolo de Internet.

WSDL – Web Services Description Language.

WWW – World Wide Web.

W3C – World Wide Web Consortium.

XHTML – Extensible Hypertext Markup Language.

XML – Extensible Markup Language.

XML Schema – Extensible Markup Language Schema.

XMPP– Protocolo extensible de mensajería y comunicación de presencia.

Referencias

Nota: Todos los enlaces a los documentos electrónicos que aparecen en las siguientes referencias han sido comprobados con fecha 12 de abril de 2010

1. **Alonso Álvarez, Ángel.** *Introducción a VILA_1. El lenguaje de la accesibilidad. Hacia un mundo sin barreras lingüísticas.* León : Instituto de Automática y Fabricación (Universidad de León), 2009.
2. **Alonso Álvarez, Ángel and Fernández Moro, Mari Paz.** *Manual de Técnicas de Estudio.* León : Everest, 1991. ISBN: 84-241-2716-1.
3. *Una propuesta de lenguaje exclusivamente gráfica para la comunicación humana.* **Alonso Álvarez, Ángel.** 14, 1992, Comunicación y Lenguaje.
4. *Hacia un lenguaje gráfico en la enseñanza.* **Alonso Álvarez, Ángel, Foces Morán, Jose María and Fernández Moro, Mari Paz.** 18, 1993, Comunicación y Lenguaje.
5. *Transparencias-resumen en la enseñanza de electrónica.* **Alonso, Ángel and Ferrero, Miguel.** Madrid : Universidad Politécnica de Madrid, 1994. Actas del primer congreso sobre tecnologías aplicadas a la enseñanza de la electrónica.
6. *Un lenguaje visual en la enseñanza técnica.* **Ferrero, Miguel and Alonso, Ángel.** Las Palmas de Gran Canaria : Universidad de Las Palmas, 1995. Actas de las 6ª Jornadas de Tecnología Electrónica.
7. *Estructura gráfica de la electrónica mediante sistemas multimedia.* **Alonso, Ángel, et al.** Sevilla : Universidad de Sevilla, 1996. Actas del 2º congreso de tecnologías aplicadas a la enseñanza de la electrónica.
8. *Estructuración gráfica de la electrónica mediante sistemas multimedia.* **Alonso, Ángel, et al.** 3, Chile : s.n., 1998, Información Tecnológica, Vol. 9.
9. **Sierra, R.** *Tesis doctorales y trabajos de investigación científica.* Madrid : Paraninfo, 1994.
10. NetLingo. [Online] 2009. [Cited: julio 28, 2009.] <http://www.netlingo.com/>.

-
11. *Interaction and outeraction: instant messaging in action*. **Nardi, Bonnie A, Whittacker, Steve and Bradner, Erin**. Philadelphia, Pennsylvania, United States : s.n., 2000. Proceedings of the 2000 ACM conference on Computer supported cooperative work. pp. 79-88.
 12. ICQ.COM. [Online] 2010. [Cited: abril 2, 2010.] <http://www.mirabilis.com/>.
 13. ICQ. [Online] 2010. [Cited: abril 1, 2010.] <http://www.icq.com/>.
 14. Yahoo!Messenger. [Online] 2010. [Cited: abril 2, 2010.] <http://messenger.yahoo.com/>.
 15. Windows Live Messenger. [Online] 2010. [Cited: abril 2, 2010.] <http://download.live.com/messenger>.
 16. Google Talk. [Online] 2010. [Cited: abril 2, 2010.] <http://www.google.com/talk/>.
 17. AIM. [Online] 2010. [Cited: abril 5, 2010.] <http://dashboard.aim.com/aim>.
 18. I'm QQ. [Online] 2010. [Cited: abril 2, 2010.] <http://im.qq.com/>.
 19. Microsoft. [Online] 2010. [Cited: marzo 31, 2010.] <http://www.microsoft.com/en/us/default.aspx>.
 20. Pidgin. [Online] 2010. [Cited: abril 1, 2010.] <http://www.pidgin.im/>.
 21. Skype. [Online] 2010. [Cited: abril 1, 2010.] <http://www.skype.com/intl/es/>.
 22. **Erl, T.** *Service-Oriented Architecture (SOA). Concepts, Technology, and Design*. s.l. : Prentice Hall PTR, 2005.
 23. **MacKenzie, C. Matthew, et al.** Reference Model for Service Oriented Architecture 1.0. *OASIS*. [Online] Agosto 2, 2006. [Cited: Junio 1, 2009.] <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>.
 24. **Booth, David, et al.** Web Services Architecture. *W3C Working Group*. [Online] Febrero 11, 2004. [Cited: Junio 1, 2009.] <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>.
 25. *Semantic Web Services*. **McIlraith, Sheila A., Son, Tran Cao and Zeng, Honglei**. 2001, Intelligent Systems, IEEE, pp. 46- 53.
 26. *Entropia: architecture and performance of an enterprise desktop grid system*. **Chien, Andrew, et al.** s.l. : Elsevier, 2003, Journal of Parallel and Distributed Computing, pp. 597-610.
 27. **McGovern, James, et al.** *Java Web Services Architecture*. s.l. : Elsevier Science Ltd , 2005.
 28. **Srinivasan, R.** *RPC: Remote Procedure Call Protocol Specification Version 2*. s.l. : DDN Network Information Center, 1995.
 29. **Brown, N. and Kindel, C.** *Distributed Component Object Model Protocol -- DCOM/1.0*. Redmond : Microsoft Corporation, 1996.

-
30. **Sun Microsystems, Inc.** Java™ Remote Method Invocation Specification. [Online] 2006. <http://java.sun.com/javase/6/docs/platform/rmi/spec/rmi-title.html>.
 31. **OMG.** *CORBA Services: Common Object Services Specification*. s.l. : Object Management Group, 1997.
 32. *Performance comparison of CORBA and RMI.* **Juric, M. B., Rozman, I. and Hericko, M.** 2000, Information and Software Technology, pp. 915-933.
 33. **Newcomer, Eric.** *Understanding Web services: XML, WSDL, SOAP, and UDDI*. s.l. : Addison-Wesley, 2002.
 34. **McGovern, James, et al.** WSDL. *Java Web Services Architecture*. s.l. : Elsevier Science Ltd , 2003, pp. 133-176.
 35. **Gurugé, Anura.** SOAP. *Web Services*. s.l. : Digital Press, 2003, pp. 189-226.
 36. **UDDI.** The UDDI Technical White Paper. [Online] 2000. <http://uddi.xml.org/>.
 37. **Baeza-Yates, R and Ribeiro-Neto, B.** *Modern Information Retrieval*. s.l. : Addison Wesley - Longman Publishing co., 1999.
 38. **Lamport, Leslie.** *LATEX: A document preparation system: user's guide and reference manual*. Segunda edición. Boston : Addison-Wesley Longman Publishing Co., Inc, 1994.
 39. **Goldfarb, Charles F. and Rubinsky, Yuri.** *The SGML handbook*. s.l. : Oxford University Press, 1990.
 40. Extensible Markup Language (XML). *W3C*. [Online] Junio 1, 2009. [Cited: Junio 1, 2009.] <http://www.w3.org/xml/>.
 41. **Bray, Tim, et al.** Extensible Markup Language (XML) 1.0 (Fifth Edition). [Online] 2008. <http://www.w3.org/TR/REC-xml/>.
 42. **Pemberton, Steven.** XHTML 1.0: The Extensible HyperText Markup Language. [Online] 2000. [Cited: Julio 1, 2009.] <http://www.w3.org/TR/xhtml1>.
 43. **Ion, Patrick and Miner, Robert.** Mathematical markup language (MathML). [Online] 1999. [Cited: Julio 1, 2009.] <http://www.w3.org/TR/REC-MathML/>.
 44. **Brown, Allen, et al.** XML Schema: Formal Description. [Online] 2001. [Cited: agosto 31, 2009.] <http://www.w3.org/TR/xmlschema-formal/>.
 45. **Chinnici, Roberto, et al.** Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. *W3C*. [Online] Junio 26, 2007. [Cited: Junio 1, 2009.] <http://www.w3.org/TR/wsdl20/>.
 46. **Christensen, E., et al.** Web Services Description Language (WSDL) 1.1. [Online] 2001. <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>.

-
47. **Clement, Luc, et al.** UDDI Version 3.0.2. *UDDI Spec Technical Committee Draft*. [Online] Octubre 19, 2004. [Cited: Junio 1, 2009.] http://www.uddi.org/pubs/uddi_v3.htm.
 48. **Von Riegen, Claus, et al.** UDDI Version 2.03 Data Structure Reference. *UDDI*. [Online] Julio 19, 2002. [Cited: Junio 1, 2009.] <http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm>.
 49. **Paolucci, Massimo, et al.** Importing the Semantic Web in UDDI. *Web Services, E-Business, and the Semantic Web*. Berlin / Heidelberg : Springer, 2002, pp. 815-821.
 50. **Gudgin, M., et al.** SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). *W3C*. [Online] 2007. [Cited: Junio 1, 2009.] <http://www.w3.org/TR/soap12-part1>.
 51. *Investigating the limits of SOAP performance for scientific computing*. **Chiu, K., Govindaraju, M and Bramley, R.** 2002, High Performance Distributed Computing, 2002. HPDC-11 2002. Proceedings. 11th IEEE International Symposium on, pp. 246 - 254.
 52. **Don Box, D. E., et al.** Simple Object Access Protocol. [Online] 2000. <http://www.w3.org/TR/SOAP>.
 53. **Mitra, Nilo.** SOAP Versión 1.2 Parte 0: Fundamentos. *W3C*. [Online] Junio 24, 2003. [Cited: Junio 1, 2009.] <http://www.w3c.es/Traducciones/es/TR/2003/REC-soap12-part0-20030624/>.
 54. Internet World Stats. [Online] 2010. [Cited: enero 27, 2010.] <http://www.internetworldstats.com/stats7.htm>.
 55. **Geiger, K.** *inside ODBC*. Washington : Microsoft Press Redmond, 1995.
 56. *Recombinant Chinese Pinyin system for efficient processing of information in Chinese*. **Tong, Harry and Jin, Linda.** November 16, 1998, Computer Standards & Interfaces, Vol. 20, pp. 25-29.
 57. **Koffka, Kurt.** *Principles of Gestalt Psychology*. s.l. : Routledge, 1999.
 58. **Gordon, Wendy and Langmaid, Roy.** *Qualitative market research: A practitioner's and buyer's guida*. s.l. : Gower , 1988.
 59. **Cronk, B. C. and Inc, S.** *How to use SPSS: A step-by-step guide to analysis and interpretation*. s.l. : Pyrczak, 2008.
 60. **Berns, M. and Matsuda, K.** Applied Linguistics: Overview and History. *Encyclopedia of Language & Linguistics*. s.l. : Elsevier, 2006, Vol. II, pp. 394-405.
 61. **Chomsky, Noam.** *Syntactic Structures*. Paris : The Hague/Paris: Mouton, 1957.
 62. **Dorr, Bonnie J., Jordan, Pamela W. and Benoit, John W.** A Survey of Current Paradigms in Machine Translation. *Advances in Computers*. s.l. : Elsevier, 1999, Vol. 49, pp. 1-68.
 63. **Rudolf, Arnheim.** *Visual Thinking*. Berkeley : University of California Press, 1969.

-
64. *Identification of ambiguous queries in web search*. **Song, Ruihua, et al.** s.l. : Elsevier, 2009, Information Processing & Management, pp. 216-229.
65. *The organization and use of information: Contributions of information science, computational linguistics and artificial intelligence*. **Walker, Donald E.** 2007, Journal of the American Society for Information Science, pp. 347 - 363.
66. **Woods, William.** *Transition Networks for Natural Language Analysis*. Cambridge, Mass. : Harvard Computer Laboratory Report, 1969.
67. **ISO.** The iso 9126 standard. [Online] 1991.
<http://www.issco.unige.ch/en/research/projects/ewg96/node13.html>.
68. *Corpus Linguistics*. **Hunston, S.** 2006, Encyclopedia of Language & Linguistics, pp. 234-248.
69. **Mitkov, Ruslan.** *The Oxford Handbook of Computational Linguistics*. Oxford : Oxford University Press, 2003.
70. **Sperberg-McQueen, C. M. and Burnard, Lou.** *Guidelines for Electronic Text Encoding and Interchange*. s.l. : ACH, ACL, ALLC, 1993.
71. *Intervals*. **Date, C. J., Darwen, Hugh and Lorentzos, Nikos A.** 2003, Temporal Data & the Relational Model,, pp. 77-88.
72. *Developing a Natural Language Interface to ComplexData*. **Hendrix, G.G., et al.** 1978, ACM Transactions on Database Systems, pp. 105-147.
73. *Progress in natural language understanding--- an application in lunar geology*. **Woods, W.** s.l. : AFIPS Conference Proceedings, 1973. Proceedings of the 1973 National Computer Conference. pp. 441-450.
74. *SpeechDat Experiences in Creating Large Multilingual*. **Draxler, Christoph, Heuvel, Henk van den and Tropf, Herbert S.** 1998, Proc. of the First International Conference on Language Resources and Evaluation, pp. 361–366.
75. **Vossen, P.** *EuroWordNet: A multilingual database with lexical semantic networks*. Dordrecht : Kluwer, 1998.
76. **Marriott, Kim and Meyer, Bernd E.** *Visual language thoery*. s.l. : Springer, 1998.
77. *A domain-specific visual language for domain model evolution*. **Sprinkle, Jonathan and Karsai, G. Gabor.** 2004, Journal of Visual Languages & Computing, pp. 291-307.
78. *The Web Service Modeling Frameworj WSMF*. **Fensel, D. and Bussler, C.** s.l. : Elsevier, 2002, Electronic Commerce Research and Applications, pp. 113-137.
79. *Panorama de la investigación en lingüística informática*. **Gómez Guinovart, J., et al.** 1999, Revista Española de Lingüística Aplicada, pp. 11-24.

-
80. **Winograd, T.** *Understanding natural language*. Orlando : Academic Press, 1972.
81. **Abeillé, Anne.** *Treebanks: building and using parsed corpora*. s.l. : Springer, 2003.
82. *A preliminary study on various implementation approaches of domain-specific language.*
Kosar, Tomaž, et al. 2008, *Information and Software Technology*, pp. 390-405.
83. **Narayanan, N. Hari and Hübscher, Roland.** *Visual Language Theory: Towards a Human-Computer Interaction Perspective*. [book auth.] Kim Marriott and Bernd Meyer. *Visual Language Theory*. New York : Springer, 1998, pp. 87-128.
84. **Harrison, M. A.** *Introduction to Formal Language Theory*. Reading : Addison-Wesley, 1977.
85. *Principles for encoding machine readable dictionaries.* **Jean Véronis, Nancy Ide, Warwick-Armstrong, Susan and Calzolari, Nicoletta.** Finland : University of Tampere, 1992. Fifth Euralex International Congress,.
86. **RAE.** *Diccionario de la Lengua Española*. Vigésima segunda edición. Madrid : Real Academia Española, 2001.
87. **Rozenberg, G. and Salomaa, A.** *Handbook of Formal Language Theory*. s.l. : Springer Verlag, 1997. Vols. 1-3.
88. **Gibbon, Dafydd, Mertins, Inge and Moore, Roger K.** *Computational Linguistics. Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation*. s.l. : MIT Press, 2001, Vol. 27, pp. 149-150.
89. **Arnheim, R.** *Visual Thinking*. Berkeley : University of California Press, 1969.
90. **Saint-Martin, F.** *Semiotics of Visual Language*. Bloomington : Indiana University Press, 1990.
91. **Tufte, E. R.** *Envisioning Information*. Cheshire : Graphics Press, 1990.
92. *A principled taxonomy of software visualization.* **Price, B. A., Baecker, R. M. and Small, I. S.** 1993, *Journal of Visual Language and Computing*, pp. 211-266.
93. *Why a diagram is (sometimes) worth ten thousand words.* **Larkin, J. H. and Simon, H. A.** 1987, *Cognitive Science*, pp. 65-99.
94. **CACM.** Special section on educational technology. *Communications of the ACM*. Abril 1996.
95. *Formal methods: The very idea, some thoughts about why they work when they work.* **Berry, D. M.** 2002, *Science of Computer Programming*, pp. 11-27.
96. **Kerievsky, J.** *Refactoring to patterns*. s.l. : Addison-Wesley, 2004.
97. **Kohavi, Z., Hamming, R. W . and Feigenbaum, E. A.** *Switching and Finite Automata Theory: Computer Science Series*. s.l. : McGraw-Hill Higher Education, 1990.

-
98. **Salomaa, A and Sneddon, IN.** *Theory of automata*. s.l. : Pergamon Press Reprint, 1969.
 99. **Myhill, J.** *Linear bounded automata*. Ohio : Wright Air Development Division, 1960.
 100. *On computable numbers, with an application to the Entscheidungsproblem.* **Turing, A. M.** 1936, Proc. of London Mathematical Society 2, no. 42 and no. 43, pp. 230-236,544-546.
 101. *Domain-specific language design requires feature descriptions.* **van Deursen, A. and Klint, P.** 2002, Journal of Computing and Information Technology, pp. 1-17.
 102. *Visual Language and Converging Technologies in the Next 10-15 Years (and Beyond).* **Horn, Robert E.** 2001. National Science Foundation Conference on Converging Technologies (Nano-Bio-Info-Cogno) for Improving Human Performance.
 103. **Antonietti, A.** Why does mental visualization facilitate problem solving? [book auth.] R. H. Logie and M. Denis. *Mental Images in Human Cognition*. Amsterdam : Elsevier Science Publishers, 1991, pp. 211-227.
 104. **Merriam-Webster.** *Merriam-Webster's Collegiate Dictionary* . s.l. : Merriam-Webster, 2003.
 105. **Horn, Robert E.** *Visual Language: Global Communication for the 21st Century*. Primera edición. s.l. : Macrovu Inc., 1999.
 106. *A chomsky hierarchy of isotonic array grammars and languages.* **Cook, Curtis. R and Wang, Patrick Shen-Pei.** 1978, Computer Graphics and Image Processing, pp. 144-152.
 107. *Backus Normal form vs. Backus Naur form.* **Knuth, Donald E.** s.l. : ACM, 1964, Communications of the ACM, pp. 735 - 736.
 108. **Manning, Christopher D. and Schütze, Hinrich.** *Foundations of Statistical Natural Language Processing*. Cambridge : The MIT Press, 1999.
 109. **Fowler, M.** *Refactoring: Improving the design of existing code*. s.l. : Addison-Wesley, 2000.
 110. **McGuinness, Deborah L. and Van Harmelen, Frank.** OWL Web Ontology Language. [Online] 10 2004. [Cited: Julio 1, 2009.] <http://www.w3.org/TR/owl-features/>.
 111. **Klyne, Graham, Carroll, Jeremy J. and McBride, Brian.** Resource Description Framework (RDF):Concepts and Abstract Syntax. [Online] 2004. [Cited: Julio 1, 2009.] <http://www.w3.org/TR/rdf-concepts/>.
 112. **Houde, Stephanie and Hill, Charles.** What do Prototypes Prototype. [book auth.] M. G. Helander, T. K. Landauer and P. V. Prabhu. *Handbook of human-computer interaction*. New York : Elsevier Science Inc, 1997, pp. 367-381.
 113. **Suehring, Steve.** *Mysql bible*. New York : Wiley, 2002.

-
114. **Gosling, James, et al.** *Java Language Specification: The Java Series*. 2nd edition. Boston : Addison-Wesley Longman Publishing Co., Inc, 2000.
115. **Horstmann, Cay S. and Cornell, Gary.** *Core Java 2 Volument II Características avanzadas*. Madrid : Pearson Educacuón,S.A., 2006.
116. **Abbott, Doug.** *Introducing Eclipse. Embedded Linux Development Using Eclipse*. s.l. : Newnes, 2008, pp. 1-12.
117. **Hanselman, Duane and Littlefield, Bruce R.** *Mastering MATLAB 6*. s.l. : Prentice Hall, 2000.
118. **ISO.** *Codes for the representation of languages Part 6: Alpha-4 representation for comprehensive coverage of language variation*. Switzerland : International Organization for Standardization, 2007.
119. —. *Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes*. Switzerland : International Organization for Standardization, 2006.
120. *Toward a better understanding of PowerPoint deck design*. **Farkas, David K.** 2006, Information Design Journal + Document Design, pp. 162–171 .
121. *Instant Messaging: between the messages*. **Campbell, J. D., Stanziola, E. and Feng, Jinjuan.** 2003 . Systems, Man and Cybernetics, 2003. IEEE International Conference on. pp. 2193 - 2198 vol.3 .
122. *Messaging platform and services migration*. **Carvalho, T. and Carter, R. T.** 2005, BT Technology Journal, pp. 90-97.
123. *IM means business*. **Cherry, S. M.** 2002, Spectrum, IEEE , pp. 28 - 32.
124. **Dietzold, Sebastian, Unbehauen, Jörg and Auer, Sören.** xOperator – Interconnecting the Semantic Web and Instant Messaging Networks. *The Semantic Web: Research and Applications*. s.l. : Springer Berlin / Heidelberg, 2008, pp. 19-33.
125. *An Exploratory Investigation into Instant Messaging Preferences in Two Distinct*. **Guo, Z., et al.** 2008. Professional Communication, IEEE Transactions on. pp. 396 - 415.
126. *Instant messaging puts on a business suit*. **Lawton, G.** 2003, Computer , pp. 14 - 16 .
127. *Language translation module for instant messaging systems*. **Rutaks, A.** 2008. Modern Problems of Radio Engineering, Telecommunications and Computer Science, 2008 Proceedings of International Conference on . pp. 630 - 631.

