



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES
ACATLÁN

**SISTEMA DE AUTO-ENSEÑANZA DEL IDIOMA
JAPONÉS (SAIJA)**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN MATEMÁTICAS APLICADAS
Y COMPUTACIÓN

PRESENTA:

Marco Antonio Curiel Fragoso

DIRECTOR DE TESIS:
Ing. Francisco Javier Patiño Donnadieu



Acatlán, México

Septiembre 2004



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ESTA TESIS NO SALE
DE LA BIBLIOTECA



Sistema de Auto-enseñanza del Idioma Japonés (SAIJA).

Objetivo

Facilitar el aprendizaje del idioma japonés a los alumnos que cursan el primer nivel en el CEIA a través de un programa de cómputo.

Hipótesis

Si a los alumnos del primer nivel del idioma japonés del CEIA se les brinda un material didáctico adicional, como es un programa de cómputo con ejercicios prácticos que refuercen los conocimientos adquiridos en cada lección y, mecanismos de evaluación que permitan la retroalimentación del estudiante, entonces el aprendizaje de este idioma les será más fácil y sólido.

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: Curiel Fragoso

Marco Antonio

FECHA: 2004. 10 / 07

FIRMA: 



Agradecimientos

A mi madre:

Por haberme dado la vida y porque luchaste contra viento y marea por darme todo lo necesario para vivir como un ser humano de bien y de utilidad para la sociedad.

A mi esposa:

Porque gracias a tu amor incondicional y a tu apoyo, he logrado alcanzar esta meta, ya que siempre luchaste a mi lado, codo con codo, dándome la fuerza necesaria para llegar hasta el final. Ahora sé que más que una pareja, somos un gran equipo.

A mi padre:

Por tu apoyo moral, tus sabios consejos y por alentarme siempre a poner empeño en todo lo que me proponga, trabajando arduamente hasta lograrlo sin desesperarme.

A toda mi familia:

Por ser yo la muestra de su forma y existencia, y porque con su cariño y su apoyo he logrado salir adelante, pues la familia es la base de todo individuo.

A todos mis profesores:

Porque en cada una de sus enseñanzas me mostraron que no basta estudiar para el próximo examen, sino que hay que estudiar para toda la vida y que las cosas aprendidas en las aulas, no sirven de nada si no se aplican día con día.

A todos mis amigos:

Porque son parte muy importante de mi vida, ya que gracias a su apoyo he aprendido que dos cabezas piensan mejor que una y el trabajo en equipo siempre es mejor que trabajar de manera aislada.

A mí:

Con toda humildad y a la vez con orgullo, debo levantar mi mano en alto y decir: "Si se puede y voy a seguir adelante, luchando como hasta ahora y no conformarme con los resultados obtenidos, sino esforzarme por ser mejor cada día.

A mi universidad:

Porque en ella encontré un segundo hogar en el que viví momentos alegres y tristes, aprendí muchas cosas y conocí a personas maravillosas que influyeron directa o indirectamente en mi desarrollo personal.

「井の中の蛙大海を知らず。」

井の中に住む蛙はそれ以上に広い世界があることを知らず。そこが一番だと思っている。

"La rana de la cascada no conoce el océano."

Las ranas que viven en las cascadas no conocen la existencia de un mundo más amplio. Piensan que ese mundo es el mayor.

Kotowasa

「七転び八起き。」

七回失敗してもまた八回目がある。何回でも挑戦しつづける。

"Siete caídas, ocho alzas."

A pesar de fracasar siete veces, siempre queda un octavo intento. Hay que seguir perseverando las veces que sean necesarias.

Kotowasa

「あるしかしながら方法が目標が、進化決して終わったり、それ住んでいる存在のイメージにならない。」

"No hay fin, sólo existe el camino, la evolución no se termina nunca, se transforma a la imagen del ser que la vive."

Nakamura



Contenido

<u>Contenido</u>	4
<u>Introducción</u>	7
<u>Capítulo Primero - Generalidades.</u>	9
1.1 <u>Características generales del idioma japonés.</u>	9
1.2 <u>Sistemas de escritura.</u>	10
1.2.1 <u>El silabario hiragana.</u>	11
1.2.1.1 <u>Descripción del silabario.</u>	11
1.2.1.2 <u>Pronunciación.</u>	12
1.2.2 <u>El silabario katakana.</u>	12
1.2.2.1 <u>Palabras extranjeras.</u>	12
1.2.2.2 <u>Nombres de persona.</u>	14
1.2.3 <u>Los kanji.</u>	14
1.2.3.1 <u>Relación con el japonés.</u>	14
1.2.3.2 <u>On'yomi y kun'yomi.</u>	15
1.2.3.3 <u>Nihongo wa nan desu ka (¿qué es el japonés?).</u>	15
1.2.3.4 <u>Los kanji son complejos.</u>	16
1.2.3.5 <u>¿Cuántos kanji hay?</u>	16
1.3 <u>Aspectos gramaticales.</u>	16
1.3.1 <u>Acentuación.</u>	16
1.3.2 <u>Fonética.</u>	17
1.3.2.1 <u>Sílabas y sonidos.</u>	18
1.3.2.2 <u>Debilitamiento o desaparición de las vocales 'i' y 'u'.</u>	19
1.3.2.3 <u>Acento.</u>	19
1.3.3 <u>Sintaxis.</u>	19
1.3.3.1 <u>Verbos y partículas.</u>	20
1.3.3.2 <u>Adjetivos.</u>	21
1.4 <u>Principales dificultades de aprendizaje.</u>	22
1.4.1 <u>Letras de espejo y letras similares.</u>	22
1.4.2 <u>Diversidad de cuantificadores.</u>	23
1.4.2.1 <u>Cuantificadores genéricos.</u>	23
1.4.2.2 <u>Cuantificadores para personas.</u>	24
<u>Capítulo Segundo - Análisis De La Aplicación</u>	25
2.1 <u>Reconocimiento del problema.</u>	25
2.1.1 <u>Solución.</u>	26
2.2 <u>Estudio de factibilidad.</u>	26





2.3	Determinación de requerimientos de información.	26
2.4	Estructura lógica del sistema.	27
2.4.1	Módulo Principal.	27
2.4.1.1	Diálogo.	31
2.4.1.2	Análisis gramatical.	28
2.4.1.3	Práctica 1.	28
2.4.1.4	Lecto-escritura.	29
2.4.1.5	Vocabulario.	29
2.4.1.6	Práctica 2.	30
2.4.1.7	Evaluación.	30
2.4.2	Módulo de Diccionario.	30
2.4.3	Módulo de Configuración.	31
2.4.3.1	Diálogo.	31
2.4.3.2	Gramática.	31
2.4.3.3	Lecto-escritura.	31
2.4.3.4	Vocabulario.	32
2.4.3.5	Evaluación.	32
2.4.3.6	Alumnos.	32
2.5	Especificaciones técnicas del sistema.	32
 Capítulo Tercero - Diseño De La Aplicación		34
3.1	Diagramas.	34
3.1.1	Diagrama jerárquico.	34
3.1.2	Diagrama de flujo de datos.	35
3.1.3	Diagrama de entidad-relación.	38
3.2	Diccionario de datos.	39
3.3	Interfaz de usuario.	43
3.3.1	Pantalla de Inicio.	43
3.3.2	Pantalla del módulo Principal.	43
3.3.3	Pantalla del módulo de Configuración.	46
3.3.4	Pantalla del módulo de Diccionario.	50
 Capítulo Cuarto - Desarrollo E Implementación De La Aplicación		51
4.1	Breves nociones de programación.	51
4.1.1	El cambio de concepto.	51
4.1.2	La importancia de comprender qué son los eventos del sistema.	52
4.1.3	Convenciones de codificación.	53
4.1.4	Creación de nombres básicos para los objetos.	54
4.1.4.1	Nombres básicos para variables y constantes.	54
4.1.4.2	Nombres básicos de tablas.	54
4.1.4.3	Nombres básicos para consultas.	55
4.1.4.4	Nombres básicos para módulos de formulario y de clase.	55



4.1.4.5	Nombres básicos para los controles.	55
4.1.4.6	Nombres básicos para los módulos de código.	56
4.1.4.7	Nombres básicos para los procedimientos y funciones.	56
4.2	Codificación de Saija.	56
4.2.1	Módulo Saija.	56
4.2.2	Formulario Carátula.	58
4.2.3	Formulario Principal.	58
4.2.4	Formulario de Configuración.	63
4.2.5	Formulario de Diccionario.	63
	Conclusiones	64
	Bibliografía	66
	Anexos	68
	Anexo 1. El silabario hiragana.	68
	Anexo 2. El silabario katakana.	69
	Anexo 3. Los kanji.	70
	Anexo 4. Notación de Yourdon para diagramas de flujo de datos.	71
	Anexo 5. Nomenclatura de objetos y entidades empleados en SAIJA.	72



Introducción

El presente trabajo de tesis está basado en la problemática de aprendizaje existente entre los alumnos del Centro de Enseñanza de Idiomas de la FES Acatlán (CEI) que se encuentran estudiando el primer nivel de japonés. A pesar de que el número de estudiantes de este idioma es aún bastante menor al de los que estudian, por ejemplo, francés o inglés, día con día va aumentando considerablemente, ya que cada vez más y más personas se interesan en el estudio de esta lengua.

Mi intención es crear una herramienta que sirva de apoyo a los alumnos que inician el primer curso de japonés, por lo cual, no abarcaré más allá de los aspectos básicos que necesita comprender el estudiante cuando comienza sus estudios en este idioma. En ningún momento pretendo que este programa venga a remplazar la cátedra del profesor, únicamente deseo que sea útil como apoyo al aprendizaje de la lengua japonesa.

En el capítulo primero, doy un panorama general de lo que es el idioma japonés así como la influencia que ha recibido de otras lenguas, principalmente del inglés. También abundo sobre las características gramaticales más relevantes de esta lengua; los sistemas de escritura que emplean los japoneses: hiragana, katakana, kanji y roomaji (alfabeto latino), fonética, entre otras cosas. Además de, en este capítulo también he recopilado las principales dificultades a las que se enfrenta el estudiante mexicano de japonés.

El capítulo número dos está dedicado a presentar un minucioso análisis del problema en cuestión; se realiza un listado de todos los requerimientos de información, así como de las especificaciones técnicas que debe reunir el equipo en el que se pretende tanto desarrollar como implementar el software. Asimismo, en este capítulo se plantea la estructura lógica que habrá de tener el sistema, es decir, los módulos de los que va a estar compuesto así como el objetivo que deberá de cumplir cada uno de ellos.

En el tercer capítulo se habla del diseño tanto lógico como físico del sistema, haciendo uso de diagramas de flujo de datos, entidad-relación, entre otros, para facilitar la abstracción del problema planteado en el capítulo anterior. De esta manera, se puede tener una visión esquemática del sistema así como de la forma en la que interactúan sus componentes. Al igual que el capítulo previo, éste es uno de los más importantes de todo el proyecto, ya que, sin un buen análisis ni un buen diseño, resulta mucho más difícil emprender el desarrollo de algo que no tiene unas buenas bases sólidas. El diseño de un sistema de cómputo es lo mismo que sería una maqueta para una casa o un conjunto habitacional.

En el capítulo número cuatro, adjunto los algoritmos más importantes de cada uno de los módulos del sistema, pues es aquí donde los problemas reales ya quedan codificados en un idioma más fácil de entender para la computadora (lenguaje máquina); también cabe mencionar



que el lenguaje que aquí empleo, Visual Basic 6.0, es de muy alto nivel y es relativamente sencillo de usar.

Al término del cuarto capítulo, vienen las conclusiones del presente trabajo y, ya al final, se encuentran algunos apéndices que consisten en tablas de símbolos japoneses como son los silabarios hiragana y katakana, algunos ejemplos de símbolos kanji, la notación utilizada en algunos diagramas, así como los estándares aplicados en la nomenclatura de identificadores, objetos y entidades dentro de cada uno de los módulos con conforman el sistema.

Capítulo Primero - Generalidades

El japonés es un idioma muy importante en el mundo actual, pero lamentablemente, aún no puede decirse que algún día llegue a convertirse en lengua universal, desplazando totalmente al inglés, pues existen motivos estructurales del idioma que lo impiden; no obstante, la lengua japonesa es importante y prueba de ello es que en los últimos 10 años se ha incrementado el número de gente que habla a la perfección este idioma, pese a que no es su lengua nativa.





El japonés es una lengua distinta, no sólo en la estructura o en la escritura, sino también por cómo se construye mentalmente una frase y por el modo en el cual se codifica la realidad, así que evidentemente se ve reflejada su cultura, y es que en toda sociedad resulta difícil pensar en una civilización sin tomar en cuenta cómo su lengua la representa; por ejemplo, en México, si a alguien se le cae una cosa por accidente, y otro la levanta, generalmente se dan las gracias, pero en Japón se ofrece una disculpa por el hecho (algunas veces se entiende también como un agradecimiento). "Es una especie de filosofía de interpretación de la realidad que es completamente diferente".

Actualmente el japonés ha adquirido una creciente importancia en todo el mundo, tal vez por motivos políticos, económicos, culturales, o hasta porque se convirtió en el idioma de moda; como quiera que sea, hoy en día es un hecho que miles de personas quieren aprenderlo.

Para usar el idioma en la comunicación existen normas tanto culturales como sociales, por lo que ni el inglés, ni el español, ni otra lengua son "estándar", ya que cada cultura es relativa y debemos estar abiertos al conocimiento, hay que aprender a pensar de diferentes maneras.

1.1 Características generales del idioma japonés.

Las características más importantes que podemos apreciar en el idioma japonés son las siguientes:

-  No hay diferencia entre el singular y el plural en general.
-  No hay género, excepto la diferencia de sexo que existe en la naturaleza biológica.
-  El japonés se puede escribir tanto vertical como horizontalmente.
-  El silabario japonés se compone de 47 signos fonéticos.

📖 El acento japonés no es de intensidad sino de altura. Tiene muchas palabras del mismo sonido que se distinguen, de acuerdo con su acento. Sin embargo, hay también palabras cuyo acento y voz son iguales. En estos casos, se pueden distinguir según el contexto o por la diferencia de los **kanji** (caracteres chinos).

📖 Existen muchos homófonos, porque cuando los **kanji** fueron introducidos en Japón, los japoneses los aceptaron como letras ideográficas; es decir, tomaron principalmente su significado más que su sonido. Ellos creían que aunque hubiese tantos homófonos, esto no ocasionaría inconvenientes si los caracteres eran diferentes. Por eso, se han formado muchas palabras del mismo sonido con diferentes significados.

📖 En japonés, se pueden construir oraciones sin utilizar el sujeto. En este punto, es algo parecido al español, pero su omisión es mucho más amplia. A diferencia del español, los verbos japoneses no cambian su inflexión según las personas, lo que al principio parece difícil, para los que inician sus estudios, pero juzgando por el contexto, se puede entender en la mayoría de los casos.

1.2 Sistemas de escritura.

El sistema de escritura japonés es mixto (**ideográfico** y **silábico**); en su variante *kana-majiri* procede de los siglos X-XII d. C. El sentido de la escritura es de arriba hacia abajo comenzando por la derecha.

El sistema *hiragana* nace en el siglo VIII d. C. de partes aisladas de los caracteres chinos.

El sistema *katakana* nace en el siglo IX d. C. de formas cursivas de los caracteres chinos.

El japonés moderno hunde sus raíces en el **chino**, el cual fue transmitido a través de Corea en el siglo IV después de Cristo. Los caracteres chinos escogidos por su significado para expresar las raíces de las palabras japonesas se denominan *kanji*.

Sin embargo, las palabras japonesas no pueden funcionar con sentido sin sus afijos gramaticales. Por lo tanto escogieron otros caracteres chinos por su sonido y los abreviaron; se les dio el nombre de *kana*. Éste último, a su vez se divide en los llamados *hiragana* y *katakana*. Ambos cuentan con el mismo número de signos, que son equivalentes (hay dos signos diferentes para

cada combinación silábica). Por ejemplo, la sílaba **ko** puede escribirse tanto こ (en hiragana) como コ (en katakana). Además, los llamados *kanji* son utilizados en conjunto con el hiragana para formar palabras e ideas.

Ante todo hay que señalar que el japonés se puede escribir de la manera occidental, es decir, horizontalmente y de izquierda a derecha, y también de la manera tradicional, verticalmente y de derecha a izquierda, como podemos ver en todos los *manga*. Actualmente en Japón se utilizan ambas modalidades habitualmente. Los signos de puntuación japoneses son diferentes de los occidentales. El punto se escribe <。 > y la coma se escribe al revés <、 >. No existe alfabeto propiamente dicho, sino que se utilizan signos que representan sílabas –combinaciones de consonante más vocal– con una sola excepción: la letra *n* es la única consonante que puede ir sola.

1.2.1 El silabario hiragana.

Este silabario es quizás el más importante y utilizado de los dos que existen, ya que se utiliza para escribir las palabras propiamente japonesas. Se dice que fue inventado por las antiguas mujeres japonesas, dado que sus formas son más suaves y redondeadas que las del otro silabario, el katakana, el cual seguramente fue inventado por los hombres.

Si una palabra no tiene kanji o si el autor no recuerda el kanji correspondiente a una palabra se utiliza el hiragana. Asimismo, las partículas y las desinencias verbales se escriben en hiragana (*Véase Anexo 1*).

El *hiragana* o *hirakana* “kana simple”, derivado de la escritura cursiva china (*ts’ao-shu*) es empleado profusamente en periódicos, literatura y, en general, en la vida diaria. Por esta razón, es lo primero que aprenden los niños japoneses, ya que todos los libros infantiles de lectura están escritos en este silabario. Más tarde se van introduciendo el katakana y los kanji a medida que el niño va ampliando su conocimiento.

1.2.1.1 Descripción del silabario.

Existen 46 sonidos básicos, que son los que se pueden ver en la primera columna del silabario que se muestra en la tabla. En la segunda columna se puede observar la lista de sonidos impuros (derivados del otro sonido). Cabe mencionar que la sílaba **ka** es la misma que **ga**, pero que **ga** tiene dos rayitas –llamadas “ten ten”– en la parte superior derecha, lo mismo sucede cuando pasamos de la fila **s** a la **z**, de la **t** a la **d** y de la **h** a la **b** (*Véase Anexo 3*).

Para pasar de la **h** a la **p** hay que poner un circulito –llamado “maru”– en la parte superior del carácter.

En la décima primera columna podemos ver las combinaciones de los caracteres de la columna *i* (ki, gi, shi, etc.) con los de la fila *y* (ya, yu, yo), éstos últimos escritos en menor tamaño. Estas combinaciones se utilizan para representar sonidos más complejos como **cha**, **hyo** o **gyu**. No existe la letra *l* en japonés. Cuando tengamos que escribir una palabra extranjera que lleve una *l*, tendremos que sustituirla por una *r*. Por ejemplo “Laura” se pronunciará “Raura”. Esto es completamente distinto al chino, lo que provoca unos malentendidos muy importantes. (¿Cuántas veces habremos oído a algún gracioso imitando a un japonés y hablando con la “l”?).

1.2.1.2 Pronunciación.

La pronunciación del japonés es muy fácil para los hispanohablantes. Vamos a ver:

- ✓ La **r** se pronuncia como en cara y nunca como en carro.
- ✓ La **sh** como en “Sharon”.
- ✓ La **j** como en James y no como en Juan.
- ✓ La **z** como en “Zelda” y no como en zorro.
- ✓ La **h** es un poco aspirada, como en “Harry”.
- ✓ Todo lo demás suena igual que en español, incluso las vocales, aunque, en realidad la **u** suena más cerrada que en español.

1.2.2 El silabario katakana.

El *katakana* “kana adjunto”, también conocido como *Yamatogana* (“kana japonés”) (*Yamato* = Japón, *gana* = *kana*), derivado de algunos caracteres de la escritura china normal (*k'ai-shu*), y empleado principalmente en la literatura científica y los documentos públicos. Éste es el segundo silabario, cuyo aprendizaje es básico para leer en japonés. Sobre todo si se tiene un gran gusto por el **manga** porque este silabario se utiliza con profusión en él.

1.2.2.1 Palabras extranjeras.

Básicamente, el *katakana* se utiliza para escribir palabras extranjeras que han sido adaptadas previamente a la pobre fonética japonesa. Así, “ordenador” se convierte en “konpyuutaa” コンピューター, “trabajo a tiempo parcial” se convierte en “arubaito” アルバイト, y “prêt-à-porter” en “puretaporute” プレタポルテ. Los ejemplos anteriores provienen del inglés (computer), del alemán (Arbeit) y del francés (prêt-à-porter), aunque tampoco faltan ejemplos del portugués como “pan” (パン) y del español “capa” (カツパ). Es importante constatar que el 11% de las palabras japonesas son extranjerismos, viniendo la gran mayoría del inglés. Ésta es una proporción descomunal de extranjerismos que no se encuentra en ninguna otra lengua del mundo. Además,



el katakana también se utiliza mucho en las onomatopeyas de los manga, es extremadamente frecuente ver palabras escritas en katakana en las páginas de cualquier cómic (*Véase Anexo 2*).

1.2.2.2 Nombres de persona.

Todos los nombres de personas y topónimos -nombres de ciudades o lugares geográficos- que no sean japoneses puros se deben escribir en katakana. Primero que nada, hay que convertir los sonidos de la palabra que queramos escribir en japonés a su propia fonética. El principal problema es que el japonés no tiene sonidos consonantes individuales -salvo la **n**-. Así, para cada sonido consonante que vaya solo, tendremos que escribir una **u** detrás, excepto en el caso de la **t** y **d**, en cuyo caso habrá que poner una **o**. Así, el nombre "Marc" quedaría マルク (Maruku), "Sandra" quedaría サンドラ (Sandora), y curiosamente "Olga" quedaría オルガ (Oruga).

1.2.3 Los kanji.

La natural inquietud humana por que "las palabras no se las lleve el viento" conduce a inventar un tipo de escritura basado en dibujos. Se trata de dibujos que representan diferentes conceptos materiales o abstractos. Esto no es raro en la historia humana, no hay más que ver la escritura jeroglífica de los egipcios para comprenderlo. Lo peculiar en el caso de los caracteres chinos es que la escritura no se simplificó para acabar formando un alfabeto que representase meramente los sonidos, como en el caso de la escritura romana, que se originó a partir del fenicio, pasando por el tamiz del griego. En China la función de los ideogramas es la de expresar a la vez sonido y significado, no como en la escritura occidental, que expresa simplemente sonido. Evidentemente, los pictogramas empezaron como dibujos más o menos realistas de las cosas y con el uso se empezaron a estilizar y simplificar hasta llegar a las formas actuales.

1.2.3.1 Relación con el japonés.

Pues bien, transcurría el siglo IV de nuestra era y los habitantes del archipiélago japonés no tenían todavía escritura. A través de la península de Corea se introdujo la escritura china, en principio sólo alguna gente culta sabía leer el chino y únicamente lo hacían para leer tratados sobre budismo y filosofía. Poco a poco se fueron utilizando los ideogramas chinos para escribir el japonés. Pero aquí hubo un problema. El idioma japonés ya existía, pero no tenía escritura, y al importar los caracteres chinos también se importó la pronunciación china (con cambios sustanciales a causa de la pobre fonética japonesa). Así que un mismo carácter se podía leer de dos o más maneras distintas. Por ejemplo, el carácter que representa montaña, 山, se puede pronunciar "a la japonesa", es decir "yama" o bien al estilo chino, "san". Y aquí encontramos uno de los errores de interpretación más garrafales del japonés, ya que esta palabra, 富士山, que significa "monte Fuji", se pronuncia "Fuji-san" y no "Fuji-yama" como nos han hecho creer siempre.

1.2.3.2 On'yomi y kun'yomi.

Estas diferentes maneras de pronunciar un carácter se llaman on'yomi (lectura "on", derivada del chino) y kun'yomi (lectura "kun", palabra japonesa). ¿Cómo podemos saber que la palabra 富士山 se pronuncia "Fuji-san" y no, por ejemplo "Fuji-yama" o "Tomishi-san" o "Tomishi-sen" o "Fuushi-yama" u otras combinaciones?

Pues la respuesta es: no se sabe. Pero se tienen pistas: Normalmente si un carácter va solo en la frase, se suele leer en kun'yomi y si va acompañado de otros kanji se suele leer en on'yomi. Ésta es una norma que funciona en el 90% de las veces, pero se debe ser muy cuidadoso con el 10% restante.

Ejemplo: El carácter 新 (nuevo)

その新しい新聞はおもしろいです

"Sono atarashii shinbun wa omoshiroi desu."

"Ese periódico nuevo es interesante."

Sono = ese / atarashii = nuevo / shinbun = periódico / wa = (partícula de sujeto) / omoshiroi = interesante / desu = (verbo ser).

Vemos que el mismo carácter aparece dos veces en la misma frase pero que se pronuncia diferente. La primera vez se pronuncia en kun'yomi, "atara". La palabra "atarashii" significa "nuevo".

La segunda vez va acompañado de otro kanji, que significa "oír", y se lee "shin", es decir, en on'yomi, "shin" = "nuevo", "bun" = "oír", juntos forman la palabra "periódico", que recoge "sucesos (cosas que se han oído) nuevos" (*Véase Anexo 3*).

1.2.3.3 Nihongo wa nan desu ka (¿qué es el japonés?).

Analicemos la palabra Nihongo, que es el nombre que se le da al idioma japonés.

Vemos que se escribe 日本語.

El primer kanji, (ni), significa "día", "sol". El segundo carácter (hon) significa "origen", "raíz" y el tercero (go) "idioma". El nombre de Japón es "Nihon" o "Nippon" y se escribe 日本. ¿Y qué tiene que ver "sol y origen", con el Japón? Quizá, alguna vez habremos escuchado la expresión "el país del sol naciente".

Por tanto, “Nihongo” significa “el idioma del país del sol naciente”, es decir “japonés”. En el caso de “español” se escribe スペイン語 (Supein-go), “Supein” = “España”, del inglés “Spain”.

1.2.3.4 Los kanji son complejos.

Efectivamente, dominar la escritura kanji resulta un desafío importante, ya que existen muchos kanji parecidos, existe un orden de trazos que hay que seguir para escribir los kanji correctamente y además está el tema de las lecturas **on** y **kun**. Existen kanji muy fáciles como por ejemplo el de persona 人 (hito, nin o jin - 2 trazos) pero también kanji complicados como el de máquina 機 (ki - 16 trazos).

1.2.3.5 ¿Cuántos kanji hay?

¡Buena pregunta! Técnicamente hay alrededor de 45,000 kanji, pero “solamente” se utilizan habitualmente unos 3,000 en el japonés. Existe una lista de 1,945 kanji llamada “Jôyô kanji” o kanji de uso común que son los que se pueden utilizar en la prensa escrita. Si se utiliza algún kanji de fuera de esa lista, hay que dar la lectura en *furigana*. Por lo tanto, hay que tener mucha paciencia con el estudio de este idioma.

1.3 Aspectos gramaticales.

La gramática es la rama de la lingüística que tiene por objeto el estudio de la forma y composición de las palabras (morfología), así como de su interrelación dentro de la oración o de la frase (sintaxis). El estudio de la gramática muestra el funcionamiento de las palabras en una lengua.

La primera vez que casi todo el mundo establece contacto con la gramática es en la escuela cuando estudia su propia lengua o al aprender otra, como segunda lengua. Se denomina normativa porque dice cuál es el funcionamiento de las diversas partes de la oración según la norma de cada idioma. Dictamina qué palabras son compatibles entre sí y qué oraciones están bien formadas, de manera que cualquier hablante a través de las reglas gramaticales perciba si emplea bien o mal esa lengua.

1.3.1 Acentuación.

El acento del japonés es de tono y no de intensidad.

Hay cuatro tipos de acento en el japonés:



- 1) Sólo la primera sílaba de la palabra está en tono bajo y todas las otras que siguen, en tono alto. Cuando la palabra va seguida de una partícula, como 'ga', el tono alto se extiende también sobre la partícula.

Ejemplos:

u- shi	(vaca / toro)	u- shi-ga
su- i-ka	(sandía)	su- i-ka-ga
chi- ka-te -tsu	(metro)	chi- ka-te -tsu- ga

- 2) La primera sílaba de la primera palabra está en tono bajo y las que siguen en tono alto, pero al agregar alguna partícula, como 'ga', ésta se pronuncia en tono bajo.

Ejemplos:

na- tsu	(verano)	na- tsu-ga
a- ta-ma	(cabeza)	a- ta-ma-ga

- 3) La primera sílaba se pronuncia en tono bajo, las intermedias en tono alto y las siguientes en tono bajo.

Ejemplos:

ko- ko -ro	(corazón)
ku- da -mo-no	(fruta)
o- to -to-i	(anteayer)

- 4) Sólo la primera sílaba está en tono alto y todas las otras que siguen en tono bajo.

Ejemplos:

go- go	(tarde)
ki- n -ko	(caja fuerte)
ro- p -pu-n	(seis-minutos)

1.3.2 Fonética.

La *fonética* es la rama de la lingüística que estudia la producción, naturaleza física y percepción de los sonidos de una lengua. Existen también otras dos ramas de la lingüística que están estrechamente relacionadas con la fonética: la fonemática y la fonología.

Este otro fonema constituye una sílaba independiente que repite el sonido vocálico de la sílaba anterior.

1.3.2.2 Debilitamiento o desaparición de las vocales 'i' y 'u'.

Las vocales 'i' y 'u', cuando ocurren entre dos consonantes sordas, se debilitan o desaparecen.

Ej.	sukoshi	s(u)koshi
	Supein	s(u)pein
	kiku	k(i)ku

En los primeros dos ejemplos, cuídese de no agregar una 'e' al comienzo de la palabra.

El debilitamiento (o desaparición) de la 'u' ocurre también frecuentemente al final de la oración.

Ej.	ohayoo gozaimas(u)
-----	--------------------

1.3.2.3 Acento.

La acentuación de las palabras japonesas, a diferencia del español, que puede ser tónica (fuerte) o átona (débil), se clasifica en alta y baja. Además, no hay reglas de acentuación.

Ej.	i ma	ku ni	
	mi ka n	a ta ma	ta ka i
	ra i ge tsu	ra i ne n	
	ke i za i	ji do o sha	
	se n se i	ta te mo no	
	mi ji ka i	su pe i n	

Tanto las sílabas de acentuación alta como baja deben pronunciarse con la misma duración.

1.3.3 Sintaxis.

La sintaxis de las frases japonesas es muy peculiar y difiere un tanto de la del español. Para empezar, mencionaré que en el japonés no existen artículos ni hay género ni número. No hay preposiciones, pero en cambio existen ciertos elementos llamados **partículas** que acompañan al sustantivo. Además, se puede seguir prácticamente cualquier orden con la condición de que las partículas van siempre detrás del sustantivo al que acompañan y el verbo principal de la oración,

siempre debe ir hasta el final. No obstante, hay algunos casos en los que después del verbo se coloca alguna palabra ya sea interrogativa o para imprimir cierto énfasis al enunciado en cuestión.

1.3.3.1 Verbos y partículas.

Los verbos no tienen desinencias para cada persona. Existen dos formas básicas de un verbo japonés: la forma de diccionario (jishokei) y la forma “masu” (masukei). Hay un verbo que es más o menos equivalente a “ser” o “estar” en español: **desu**. Este verbo se utiliza para formular oraciones afirmativas y muchas veces no es necesario traducirlo.

Ej. 1 Watashi-wa Antonio desu.

En este caso, **watashi** es la primera persona del singular y, por tanto equivale al **yo** español. La partícula **wa** identifica el tópico o tema de lo que se está hablando y no tiene un equivalente directo en español, pero sería más o menos como decir: “a propósito de...”. Entonces, una traducción muy literal sería: “A propósito de mí, soy Antonio”.

Ej. 2 Naomi-san-ga konpyuutaa-o tsukaimasu.

En este segundo ejemplo, **Naomi** es un nombre propio y va sucedido por la partícula **ga**, la cual funciona de manera similar a **wa**, sólo que esta vez, más que identificar al tema, identifica al sujeto; la partícula **o** sirve para señalar el objeto, que en este caso es **konpyuutaa** (ordenador) y **tsukaimasu** (usar) es el verbo. En ese sentido, la frase querría decir algo así como: “Naomi usa el ordenador”.

Lo que resulta especial del uso de partículas en japonés, es que permiten muchas ordenaciones de las palabras en la frase, y el significado sigue siendo el mismo (respetando la regla de que el verbo vaya siempre al final). Retomando el ejemplo anterior, si escribimos “konpyuutaa-o Naomi-ga tsukaimasu.”, la frase sigue significando lo mismo aunque hayamos cambiado el orden de las palabras (en español no se puede hacer esto: “una computadora usa a Naomi”, ya que significa algo completamente diferente).

En frases más complejas ocurre lo mismo. Veamos otro ejemplo, ahora añadiendo además un objeto indirecto. La frase: “Naomi dio un ordenador a Taro”, en el orden básico es: Naomi-san-ga (sujeto: Naomi-san) Taro-san-ni (objeto indirecto: a Taro-san) konpyuutaa-o (objeto: una computadora) ageta (verbo: dio). Pues bien, todas estas frases significan lo mismo y son correctas:

- ✓ Naomi-san-ga Taro-san-ni konpyuutaa-o ageta.
- ✓ Naomi-san-ga konpyuutaa-o Taro-san-ni ageta.
- ✓ Taro-san-ni Naomi-san-ga konpyuutaa-o ageta.
- ✓ Konpyuutaa-o Naomi-san-ga Taro-san-ni ageta.
- ✓ Taro-san-ni konpyuutaa-o Naomi-san-ga ageta.



✓ Konpyuutaa-o Taro-san-ni Naomi-san-ga ageta.

Para formar oraciones negativas, simplemente se quita la terminación **masu** del verbo principal y se reemplaza por **masen**.

Ej. Naomi-san-ga konpyuutaa-wa tsukaimasen.
(Naomi no usa el ordenador.)

Kyoo, gakkoo-niwa ikimasen.
(Hoy no iré a la escuela.)

En el caso del verbo **desu** la negación quedaría así:

Watashi-wa Antonio dewa arimasen.
(Yo no soy Antonio.)

1.3.3.2 Adjetivos.

Los adjetivos calificativos se colocan siempre antes de la palabra en cuestión, por ejemplo: "computadora cara" sería: "takai (caro) konpyuutaa", y las dos palabras se colocarían juntas dentro de la frase, como una sola unidad. La frase "Naomi usa una computadora cara" quedaría:

Naomi-ga takai konpyuutaa-o tsukaimasu.

Existen dos clases de adjetivos: los terminados en **i** (ikeiyooshi) y los terminados en **na** (nakeiyooshi).

Los adjetivos-*i*, en la oración afirmativa, simplemente se colocan tales cuales delante del sustantivo al que van a calificar:

Inu wa ookii desu.
(El perro es grande.)

Neko wa chiisai desu.
(El gato es pequeño.)

Y en la forma negativa, se reemplaza la terminación **i** por **ku**, quedando:

Inu wa ookiku arimasen.
(El perro no es grande.)

Neko wa chiisaku arimasen.

(El gato no es pequeño.)

Los adjetivos-*na* en la forma afirmativa pierden la terminación **na** cuando no van delante del sustantivo. Por ejemplo:

Onnanohito wa kirei desu.
(La mujer es bonita.)

En la forma negativa, también se toma el adjetivo sin la terminación **na**, pero la oración queda de la siguiente forma:

Onnanohito wa kirei dewa arimasen.
(La mujer no es bonita.)

1.4 Principales dificultades de aprendizaje.

Cuando estudiamos algo, sea un curso de Mercadotecnia, de Contabilidad o algún idioma generalmente nos encontramos con una serie de dificultades propias del aprendizaje de dichos estudios. Finalmente, se nos ha dicho que “la práctica hace al maestro” y no es sino hasta que se ponen en práctica todos los conocimientos adquiridos durante la teoría cuando encontramos un sinnúmero de dudas e inquietudes que en su momento no fueron plenamente identificadas.

En el caso del japonés, como en el de cualquier otro idioma extranjero –incluso en el mismo español–, existen ciertos temas que traen consigo algunas peculiaridades las cuales dificultan el aprendizaje, haciéndolo a veces un tanto lento y pesado. Por ello, surge la necesidad de crear un método de enseñanza extra que refuerce los conocimientos adquiridos en los cursos normales y que esclarezca aquellos conceptos que en un principio eran difíciles de comprender, con ejemplos sencillos y, ciertamente, divertidos.

Algunas de estas dificultades son: las letras de espejo, la diversidad de cuantificadores para personas y objetos, las respuestas a las expresiones interro-negativas, entre otras.

1.4.1 Letras de espejo y letras similares.

Ya he hablado anteriormente sobre la escritura japonesa. En el caso del hiragana, se trata de 46 signos básicos que representan sílabas con las que se puede escribir cualquier palabra del japonés. El katakana, al igual que el hiragana consta de otros 46 signos elementales con los que se pueden escribir palabras de origen extranjero.

En hiragana existen algunas letras que presentan ciertas similitudes en su escritura. Cuando se escribe una letra de una forma e invirtiéndola da origen a otra se la conoce con el nombre de: *letra de espejo* y si se trata de la misma letra que con sólo agregarle o quitarle algo se convierte en otra, recibe el nombre de *letra similar*.

Ej. de letras de espejo:

La **sa** y la **chi**: さ ち

Ej. de letras similares:

La **sa** y la **ki**: さ き
La **ku** y la **he**: く へ
La **ha** y la **ho**: は ほ
La **me**, la **nu** y la **no**: め ぬ の
La **ne** y la **wa**: ね わ

Estas dificultades existen también en el idioma español, tal es el caso de las siguientes letras:

d y b **p y q** **a y e** **t y f** **u y n**

1.4.2 Diversidad de cuantificadores.

Otra de las dificultades del idioma japonés se presenta al momento de cuantificar personas u objetos tales como sillas, mesas, libros, hamburguesas, etc.

En español, es muy sencillo, ya que basta con agregar el número delante del objeto que se desea cuantificar, con la correspondiente terminación del plural.

Ej. 1 pelota
 2 sillas
 3 mexicanos

En el caso del japonés, sucede algo parecido. Sin embargo, cada tipo de objetos tiene una serie de cuantificadores con ciertas irregularidades.

1.4.2.1 Cuantificadores genéricos.

Para contar objetos en general, se utilizan los siguientes cuantificadores:

1 hitotsu 2 futatsu



3 mittsu	4 yottsu
5 itsutsu	6 muttsu
7 nanatsu	8 yattsu
9 kokonotsu	10 too

Ej.	mittsu hanbaagaa.	(tres hamburguesas)
	too kutsu.	(diez zapatos)

1.4.2.2 Cuantificadores para personas.

Siempre que se hable de seres humanos, se usarán:

1 hitori	2 futari
3 sannin	4 yonin
5 gonin	6 rokunin
7 shichinin	8 hachinin
9 kyuunin	10 juunin

Ej.	hitori Mekishiko-jin	(un mexicano)
	hachinin Amerika-jin	(ocho estadounidenses)

Y existen muchos otros cuantificadores para cosas alargadas (plumas, dedos, paraguas, etc.), planas y delgadas (hojas de papel, camisetas, etc.), máquinas y aparatos (bicicletas, coches, ordenadores, etc.). Por tal motivo, aprender todos estos cuantificadores resulta ser muy complicado, pues no sólo hay que aprendérselos sino que también se debe saber en qué circunstancias emplear cada uno de ellos.



Capítulo Segundo - Análisis De La Aplicación

Un proyecto de sistema es el proceso de gestión para la creación de un sistema o aplicación, el cual encierra un conjunto de actividades, una de las cuales es el reconocimiento del problema. Al momento de reconocer el problema ya se está obteniendo gran parte de la solución del mismo, pues al momento de plantearlo se pueden conocer las posibles soluciones y sus alcances, lo cual permite formular un procedimiento (o conjunto de procedimientos) para conseguir la resolución del problema en cuestión.

Además del procedimiento técnico, se deben de tomar en cuenta los recursos (económicos, técnicos y humanos), los costos y la planificación al momento de dar solución a un problema. El tamaño del problema es otro factor importante que puede afectar la precisión de las estimaciones. Cuando se tiene que el problema a resolver es demasiado complejo, lo ideal es formar particiones, cada una de las cuales viene a ser un nuevo problema, pero más simple de resolver y así sucesivamente hasta que obtengamos problemas con solución inmediata.

2.1 Reconocimiento del problema.

El problema que da origen al presente trabajo es que el aprendizaje del idioma japonés en el Centro de Enseñanza de Idiomas es difícil para los alumnos del primer nivel debido a ciertas características especiales –como escritura compleja y extensa, gramática un tanto diferente a la del español, etc.- que posee esta lengua y que representan un obstáculo para quienes empiezan a estudiarla.

A pesar de que el Departamento de Japonés del C. E. I. tiene materiales didácticos tales como: audiocasetes, videocasetes, diccionarios, libros, etc., no cuenta con ningún programa multimedia que apoye la enseñanza de este idioma. Por esta razón, considero que el material existente a la fecha en dicho departamento es insuficiente.

El beneficio directo para los alumnos será proporcionarles una aplicación de apoyo didáctico la cual reforzará, mediante ejercicios específicos, todos aquellos temas que en las clases tradicionales se tornan difíciles tanto para los maestros al enseñarlos como para los estudiantes a la hora de aprenderlos. En otras palabras, el software que se pretende crear servirá como un instrumento auxiliar en el proceso de enseñanza-aprendizaje tanto a los profesores como a los alumnos.

2.1.1 Solución.

Para dar solución a esta cuestión, será necesario construir una aplicación de software en ambiente visual de fácil manejo, que sirva como material de apoyo didáctico a los alumnos de primer nivel del idioma japonés y que facilite su comprensión con lecciones prácticas. El programa estará dividido en niveles, unidades y lecciones; estas últimas a su vez contarán con las secciones de: *diálogo, análisis gramatical, práctica 1, lecto-escritura, vocabulario, práctica 2 y evaluación.*

2.2 Estudio de factibilidad.

La realización de este material es factible tanto técnica como económicamente, ya que cuento con herramientas visuales de desarrollo de software, las cuales son accesibles y, con el equipo de cómputo adecuado para poder llevarlo a cabo, además de que la creación de este software forma parte del presente trabajo y el costo no es considerable si lo comparamos con la adquisición de un producto ya terminado de una empresa de las que existen en el mercado.

2.3 Determinación de requerimientos de información.

Esta fase consiste en identificar las necesidades de información a partir de los usuarios finales del sistema.

Los usuarios finales, en este caso serán los alumnos del primer nivel del idioma japonés del Centro de Enseñanza de Idiomas y los profesores, usuarios intermedios, quienes podrán acceder a la configuración del sistema para proveerlo de nuevas palabras y ejercicios en cada lección, a fin de tener un material actualizado.

Para el aprendizaje del idioma, el alumno necesita desarrollar cuatro habilidades fundamentales:

- Lectura.
- Escritura.
- Comprensión auditiva.
- Producción oral.

En ese sentido, para la lectura y la escritura –en adelante denominada simplemente como *lecto-escritura*– será necesario formar un catálogo de símbolos (hiragana, katakana y kanji) con los que construir palabras y oraciones.

Para la comprensión auditiva será necesaria la inclusión de diálogos que puedan ser escuchados por el usuario (a través de las bocinas), así como algunas canciones para poder sensibilizar su oído.

Y para la producción oral, los mismos diálogos empleados en la comprensión auditiva serán utilizados de manera que el usuario pueda tomar parte en la conversación, asumiendo el papel de alguno de los personajes y haciendo uso de un micrófono para captar su voz, con el fin de simular que está charlando con alguien.

2.4 Estructura lógica del sistema.

Ante todo, el sistema deberá estar basado en un programa de estudios organizado por unidades y lecciones dentro de cada nivel de estudio, con el fin de alcanzar los objetivos marcados. En este caso, me ocuparé únicamente del primer nivel. Las lecciones tendrán un número consecutivo único e irrepetible por nivel, sin importar las unidades, es decir, estará dividido de la siguiente manera:

NIVEL I

Unidad 1

Lección 1

Lección 2

Lección 3

...

Unidad 2

Lección 8

Lección 9

...

A su vez, el sistema estará compuesto de los siguientes módulos:

- Módulo Principal
- Módulo de Diccionario
- Módulo de Configuración

2.4.1 Módulo principal.

Todas las actividades que el usuario estará desarrollando a lo largo de cada lección estarán concentradas en un módulo único, llamado módulo principal, con el fin de que el usuario no tenga la necesidad de estar navegando entre pantallas y le sea más cómodo el uso del sistema.

Además de, las cuatro habilidades que deberá desarrollar el alumno se encuentran estrechamente relacionadas entre sí y, por tal motivo, no podrían estar separadas cada una en un módulo distinto, pues de ser así, sólo entorpecerían el aprendizaje del idioma.

Como había mencionado anteriormente, los contenidos del curso multimedia estarán distribuidos por niveles, unidades y lecciones. Dentro de cada lección, el alumno encontrará las siguientes secciones:

- 📖 Diálogo
- 📖 Análisis gramatical
- 📖 Práctica 1
- 📖 Lecto-escritura
- 📖 Vocabulario
- 📖 Práctica 2
- 📖 Evaluación

2.4.1.1. Diálogo.

Mediante el uso de diálogos en cada lección, el alumno tendrá contacto desde el inicio con la ideología de los japoneses y verá el contexto bajo el cual se usan determinadas expresiones entre los nativos de este idioma. Al mismo tiempo, será capaz de apreciar el acento y la entonación del japonés hablado, ya que las voces serán de auténticos nativos a una velocidad de conversación notablemente lenta para facilitar la comprensión y que gradualmente irá siendo más rápida hasta llegar a ser la velocidad normal con la que los japoneses hablan a diario.

2.4.1.2. Análisis gramatical.

Todo lo que aparezca en el apartado anterior será estudiado a fondo en este apartado, en el cual, la idea es enseñar al alumno las reglas del juego, esto es, la forma en la que debe de colocar cada elemento en una oración, la función que cumple dentro de ella y el sentido de ésta de acuerdo con el orden en el que se han escrito las palabras. Lo anterior será posible gracias al empleo de modelos de estructura de frase, los cuales irán apareciendo en cada lección, aumentando gradualmente su complejidad dado que cada vez incorporarán más elementos, tales como sustantivos, adjetivos, partículas, etc. Cada modelo será ilustrado mediante al menos un ejemplo para que con esto, el alumno sea capaz de resolver los ejercicios prácticos de la lección en curso.

Cabe hacer hincapié en que ésta es una de las secciones más importantes de todo idioma, ya que es la que nos da las bases para poder construir frases que den vida a nuestra conversación. Como ya lo había mencionado, son las “reglas del juego”, puesto que si vemos esta actividad como un juego didáctico, será divertido para el estudiante y no se le hará pesado.

2.4.1.3. Práctica 1.

En esta sección, el alumno encontrará diferentes tipos de ejercicios que pondrán a prueba sus conocimientos adquiridos en la sección precedente, que es de gramática. Algunos de ellos le pedirán que complete frases con ciertas palabras, donde tendrá forzosamente que escribirlas y el programa evaluará la ortografía de las mismas.

2.4.1.4. Lecto-escritura.

El propósito de este apartado es el de enseñar al usuario (alumno) los sistemas de escritura del idioma japonés, con la finalidad de que sea capaz de identificar cada uno de los símbolos al momento de leer algún texto y los sonidos que tienen éstos.

De inicio, el alumno deberá aprender el silabario hiragana, desde los sonidos simples hasta los sonidos compuestos, pues como ya mencioné en el capítulo anterior, es la base de la escritura puramente japonesa, es decir, todas las palabras que son auténticamente japonesas se deben de escribir en hiragana. Después, el alumno aprenderá el silabario katakana y su utilidad en la escritura de palabras de origen extranjero. Y finalmente, el usuario aprenderá kanji, desde la forma en cómo se escriben –en qué orden, cuántos trazos lleva cada uno, etc.– hasta las lecturas kun’yomi (lectura cuando el símbolo se encuentra solo) y on’yomi (lectura cuando el símbolo se encuentra mezclado con uno o más símbolos), y cómo se combinan unos con otros para formar palabras o ideas.

Este apartado constará de tres secciones: una en la que se presenta el símbolo con una animación que mostrará claramente cómo debe de realizarse el trazado del carácter en cuestión; su nombre y su pronunciación (mediante la grabación de una voz nativa), y otra en la que el alumno practicará los símbolos aprendidos haciendo uso del ratón. En la última sección habrá una serie de ejercicios que pondrán a prueba la destreza del alumno para identificar los caracteres japoneses, su nombre y el sonido que tiene cada uno. Al término del ejercicio, el alumno sabrá los resultados que obtuvo y conocerá las respuestas correctas para cada reactivo, permitiéndole de esta manera la retroalimentación de sus conocimientos.

Adicionalmente, el sistema tendrá la opción en este apartado de imprimir hojas con los caracteres que fueron revisados en cada lección con el fin de que el alumno practique con lápiz o bolígrafo fuera del sistema. Este tipo de ejercicio es complementario y muy recomendable para lograr una mejor caligrafía.

2.4.1.5. Vocabulario.

Además de que el alumno debe de aprender reglas gramaticales, necesita de materia prima para trabajar, que en este caso son las palabras y las frases. Para tal fin, será creado este apartado, en

el cual, en una primera sección se le enseñarán palabras y frases que utilicen las letras y símbolos aprendidos en el apartado anterior; para ello se hará uso de imágenes, ya que pedagógicamente se dice que es la mejor manera de retener el significado de una palabra o frase. En otra sección existirá una serie de ejercicios (tanto de escritura como de lectura, pronunciación y comprensión auditiva) que servirán para ratificar que efectivamente el alumno aprendió el vocabulario presentado en la lección en curso.

2.4.1.6. Práctica 2.

Esta segunda práctica tiene como fin poner a prueba la habilidad del alumno para escribir con los símbolos japoneses recién aprendidos, con actividades tales como juegos de memoria, lectura y comprensión de texto, entre otras. De alguna manera, y retomando el contenido de la práctica uno, esta parte irá preparando al alumno para lo que será el examen que deberá de realizar al final de cada lección, sólo que aquí tiene mayores libertades ya que en todo momento podrá hacer uso de sugerencias que el sistema le proponga cuando tenga dudas en lo que está desarrollando.

2.4.1.7. Evaluación.

Los exámenes que se apliquen en este apartado consistirán en grupos de ejercicios con los que el alumno ya ha trabajado a lo largo de cada lección. La medición irá desde 0 hasta 100 puntos, permitiendo así establecer un porcentaje de dominio de los elementos que ha estudiado. Al término de cada examen, si el alumno llegase a tener alguna equivocación, el programa mostrará el lugar donde la tuvo y la manera correcta en que debió de contestar determinado ejercicio. Estas pruebas podrán repetirse, si el alumno así lo desea, debiendo ocuparse el sistema de no volver a trabajar con exactamente los mismos reactivos que en un principio.

Si el examen es aprobado con 100 puntos, el alumno podrá pasar a la siguiente lección, de lo contrario tendrá que repetir la lección actual. Cuando obtenga menos de 100 puntos, el sistema le indicará al alumno qué es lo que debe de repasar para que la siguiente vez que presente la prueba alcance un mayor puntaje.

Es importante que el alumno obtenga los 100 puntos en cada lección, pues esto demostrará que domina plenamente los contenidos aprendidos en dicha lección. En caso de que no logre esa cantidad de puntos, el sistema le indicará al alumno los tópicos que debe de repasar para poder aprobar la lección en curso y pasar a la siguiente.

2.4.2 Módulo de Diccionario.

Naturalmente, el sistema contará con un módulo de diccionario para buscar palabras del español al japonés y viceversa. Cada palabra mostrará su equivalente en la otra lengua, según sea el

caso, y su función gramatical, es decir, si es un verbo, un sustantivo, adverbio, etc. En cualquier momento podrán ser agregadas nuevas palabras desde el módulo de Configuración. Para comodidad del usuario, las palabras japonesas aparecerán tanto en kana como en roomaji, facilitando así su lectura.

2.4.3 Módulo de Configuración.

Este módulo obedece a un solo propósito que es administrar al sistema y facilitar el mantenimiento del mismo. Estará organizado de las siguientes secciones: Alumnos, Símbolos, Palabras, Frases y Modelos.

2.4.3.1 Diálogo.

En esta sección, se podrán configurar los diálogos por lección, permitiendo escribir línea por línea de cada personaje así como asignar un archivo de audio donde se encuentre la grabación del diálogo en cuestión, con voces nativas. Además, se permitirá la escritura del diálogo completo en un lugar especial que brinde diferentes características de formato de texto.

2.4.3.2 Gramática.

El propósito de esta sección será el de gestionar modelos de estructura de frase que servirán para facilitar al alumno el aprendizaje de las reglas que posee el idioma japonés para formar enunciados. Se podrá capturar una explicación alusiva a cada modelo en un cuadro de texto con formato enriquecido.

2.4.3.3 Lecto-Escritura.

En esta sección se podrán configurar nuevos caracteres, en este caso, nuevos kanji, ya que de inicio, la base de datos contendrá las letras correspondientes a los silabarios *hiragana* y *katakana*, y alguno que otro kanji básico.

2.4.3.4 Vocabulario.

Este apartado tiene por objeto registrar palabras, las cuales estarán clasificadas de acuerdo a su función gramatical (sustantivo, verbo, adjetivo, etc.) y al número de lección en la que deba de ser tratada. Desde aquí también podrá asignarse alguna imagen a cada palabra, con el fin de reforzar el significado de la misma. Además, en esta sección será posible registrar nuevas frases de gran

utilidad en la conversación, las cuales estarán clasificadas según sea la situación en la que deban de ser usadas (saludos, tiempo, viajes, etc.).

2.4.3.5 Evaluación.

Esta sección servirá para configurar diferentes tipos de ejercicios, dependiendo de la lección de que se trate. Estará dividido en dos apartados, uno para los ejercicios del tipo I que se refiere más a la gramática y la sintaxis y el otro para los ejercicios del tipo II que se refiere más a vocabulario y a la escritura.

2.4.3.6 Alumnos.

Este apartado tiene como fin registrar algunos datos personales de cada alumno, pues de esta manera, el programa podrá evaluar a cada uno en forma individual. Para que todo lo anterior sea posible, es necesario llevar también un registro de todos los elementos que cada alumno va estudiando: símbolos, frases, etc.

2.6 Especificaciones técnicas del sistema.

Para poder crear el programa de multimedia SAIJA, será necesario hacer uso de una herramienta visual llamada Microsoft Visual Basic, versión 6. Su elección ha sido en base a todas las ventajas que ofrece este lenguaje con respecto a algunos otros de los que hoy en día existen en el mercado, tales como Borland Delphi o Microsoft Visual Fox Pro. Visual Basic es un lenguaje de fácil manejo y muy potente; permite crear desde un programa simple hasta las más sofisticadas aplicaciones.

Para crear y gestionar la base de datos se hará uso de Microsoft Access, dado que es una herramienta fácil de emplear y no se requieren grandes niveles de seguridad como en el caso de los sistemas de información de las grandes empresas que manejan un volumen de transacciones superior y, en donde la seguridad de los datos es vital, por ser en muchos casos de carácter confidencial.

La aplicación será desarrollada para ser ejecutada en ambiente Windows, a partir de la versión XP, ya que las herramientas anteriores requieren de este sistema operativo y por ser éste uno de los sistemas más populares y actuales en México, el cual proporciona una interfaz suficientemente amigable para el usuario y, por tal motivo, es más fácil de usar.



Para un adecuado funcionamiento de SAIJA, es indispensable una computadora personal con las siguientes características, como mínimo:

- Procesador Pentium III a 800 Mhz.
- 128 MB en RAM.
- 30 MB de espacio en disco duro.
- Ratón.
- Micrófono.
- Bocinas.
- Monitor SVGA que soporte una resolución de 1024 x 768 píxeles.
- Lápiz óptico (opcional).

Capítulo Tercero - Diseño De La Aplicación

El *Diseño de Sistemas* se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física.

El diseño del software es un proceso y un modelado a la vez. El proceso de Diseño es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del Sistema a construir. A lo largo del diseño se evalúa la calidad del desarrollo del proyecto con un conjunto de revisiones técnicas.

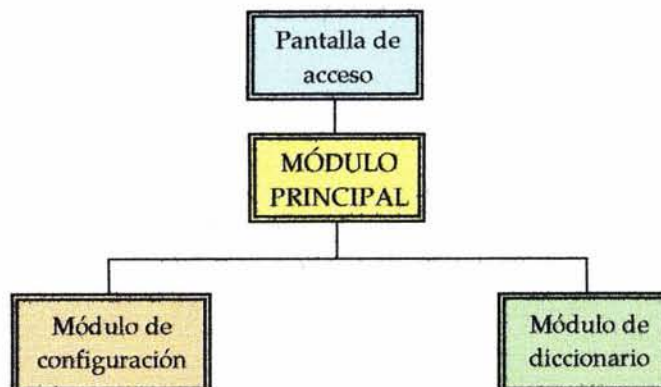
3.1 Diagramas.

Los diagramas son una representación gráfica de la estructura interna de un sistema. Gracias a ellos es posible abstraer ideas, con lo cual, se pueden comprender mejor tanto la composición como la forma de operación de un sistema informático.

Existen muchos tipos de diagramas. En este caso, sólo me ocuparé de tres de ellos por ser los más representativos: diagrama jerárquico de módulos y de procesos; diagrama de flujo de datos, y diagrama de entidad-relación.

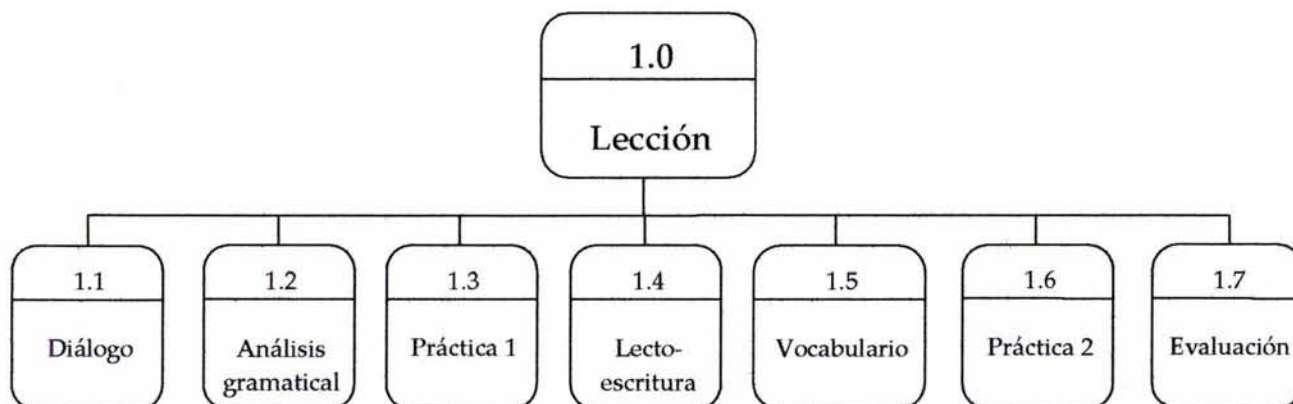
3.1.1. Diagramas jerárquico y funcional.

En un diagrama jerárquico se pueden ver claramente las relaciones de jerarquía que existen entre un módulo y otro. Es similar a un organigrama, ya que utiliza rectángulos para representar a cada módulo y líneas para marcar dichas jerarquías.



El diagrama anterior muestra que sólo existirán tres módulos en el sistema, el módulo principal en el que se encontrarán las lecciones de estudio organizadas por unidades y niveles. De este módulo se desprenderán otros dos: el módulo de configuración, en donde se podrá dar mantenimiento al programa y el módulo de diccionario, en el que se podrán realizar búsquedas de palabras o frases del japonés al español y viceversa.

Dentro del módulo principal será posible estudiar las lecciones que estarán organizadas como se ilustra en el siguiente diagrama de procesos:



En el gráfico anterior se pueden ver las secciones que el alumno deberá estudiar dentro de cada lección. Podemos decir entonces que cada sección equivale a un proceso, ya que cada uno cuenta con un conjunto de instrucciones y subprogramas independientes de los demás que, como tales, le dan una identidad dentro del sistema. Sin embargo, es necesaria la interacción de un proceso con otro para que el sistema pueda funcionar como es debido. En ese sentido, todo lo que se vea dentro del proceso denominado *Diálogo* será retomado en *Análisis gramatical* para explicarle al alumno la manera en la que se deben de estructurar las frases utilizadas dentro de una conversación.

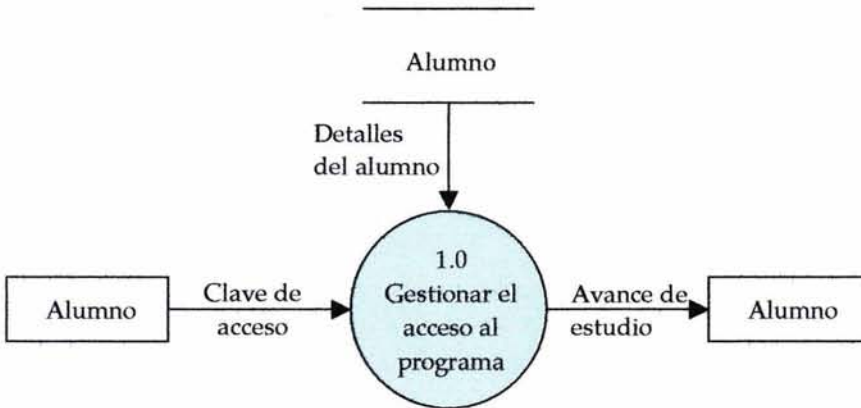
Por otra parte, las lecciones estarán agrupadas dentro de unidades de acuerdo a sus contenidos temáticos. Y a su vez, grupos de unidades constituirán cada nivel de estudio. Como ya lo había mencionado anteriormente, en este trabajo, sólo me ocuparé del primer nivel, que es considerado el más básico y en el que se espera que el principiante se familiarice con esta lengua.

3.1.2. Diagrama de flujo de datos.

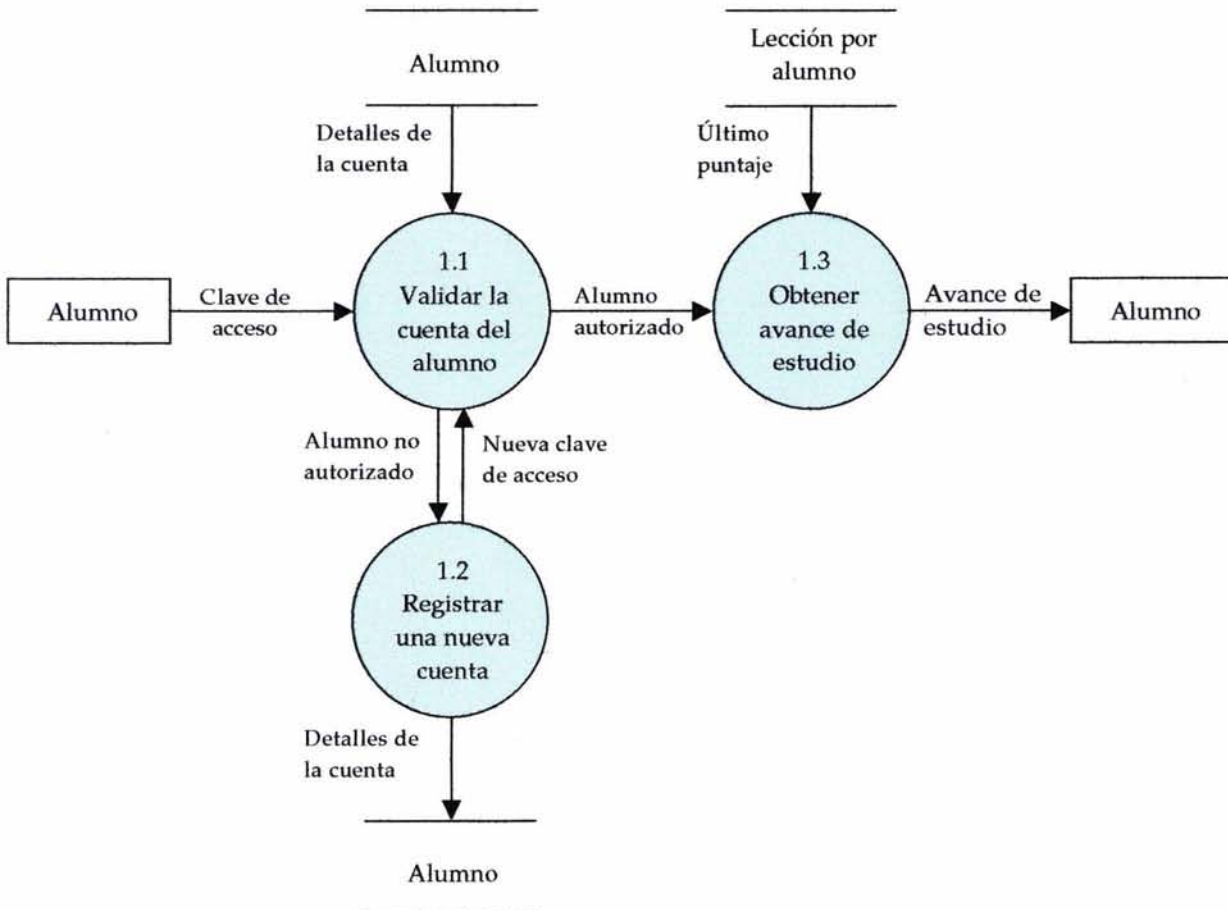
Un diagrama de flujo de datos nos permite conocer el movimiento que tendrá toda la información con la que se estará trabajando a lo largo de cada lección en del sistema. Por razones de simplicidad, utilizaré la notación de Yourdon para trazar el diagrama (Véase Anexo 4).

El primer paso para utilizar el programa es identificarse. Los siguientes diagramas muestran el primero y segundo niveles del proceso de Acceso al programa.

Primer nivel

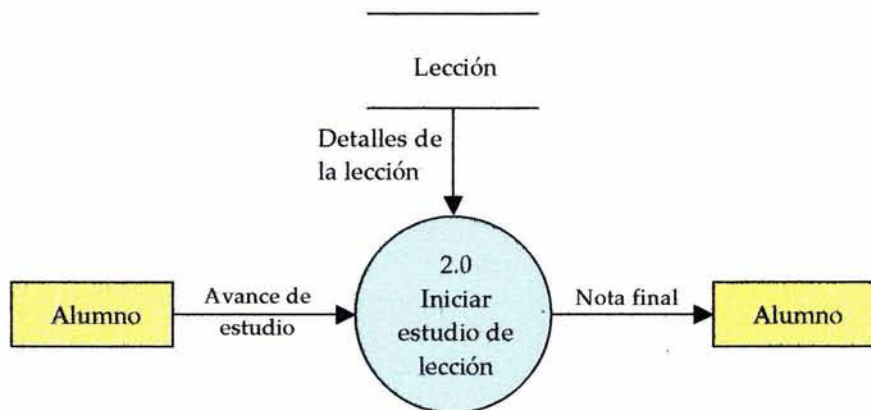


Segundo nivel

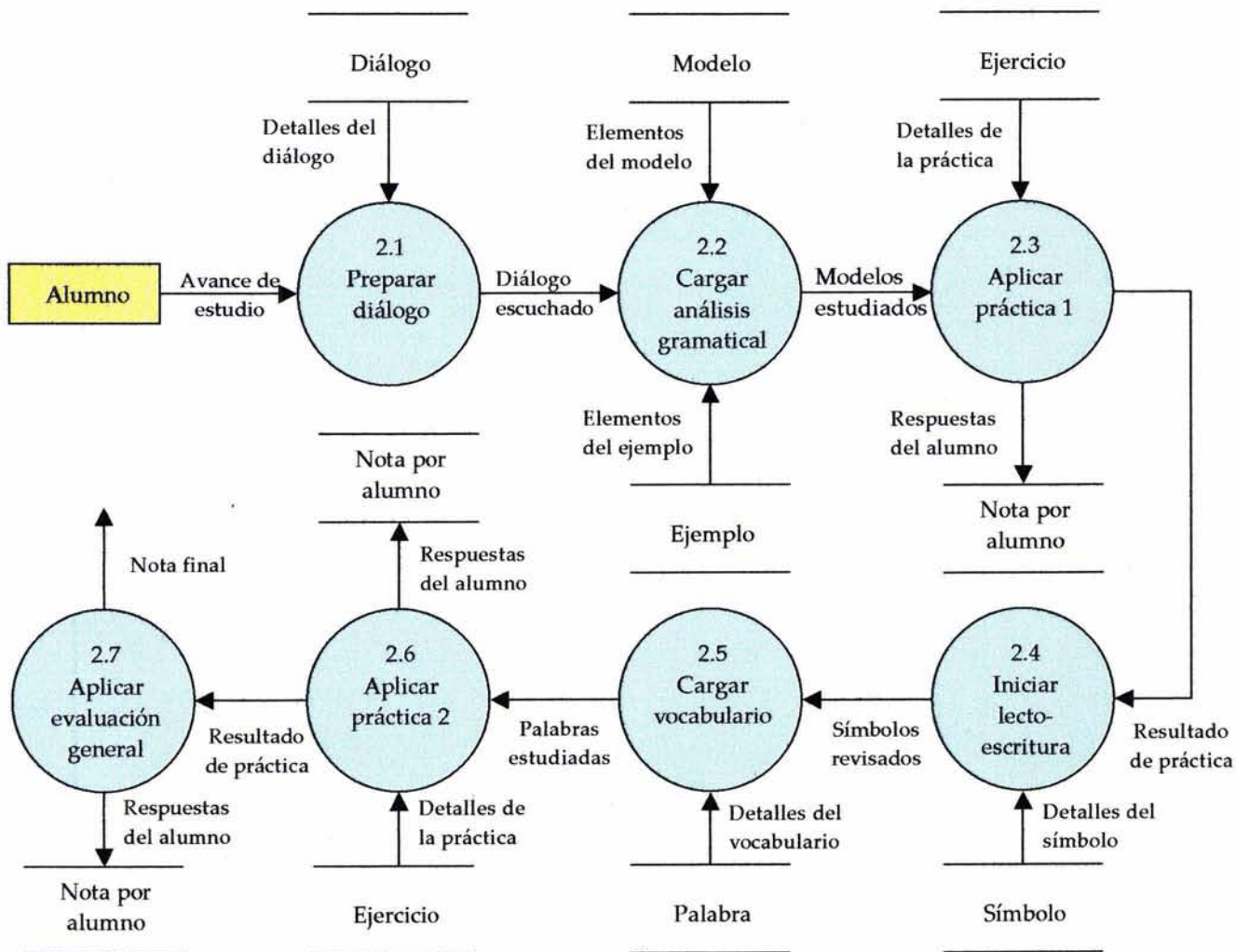


Una vez que el alumno es identificado puede comenzar ya con el estudio de la lección correspondiente (o con cualquiera anterior a ésta), la cual será determinada por el puntaje obtenido hasta el momento.

Primer nivel



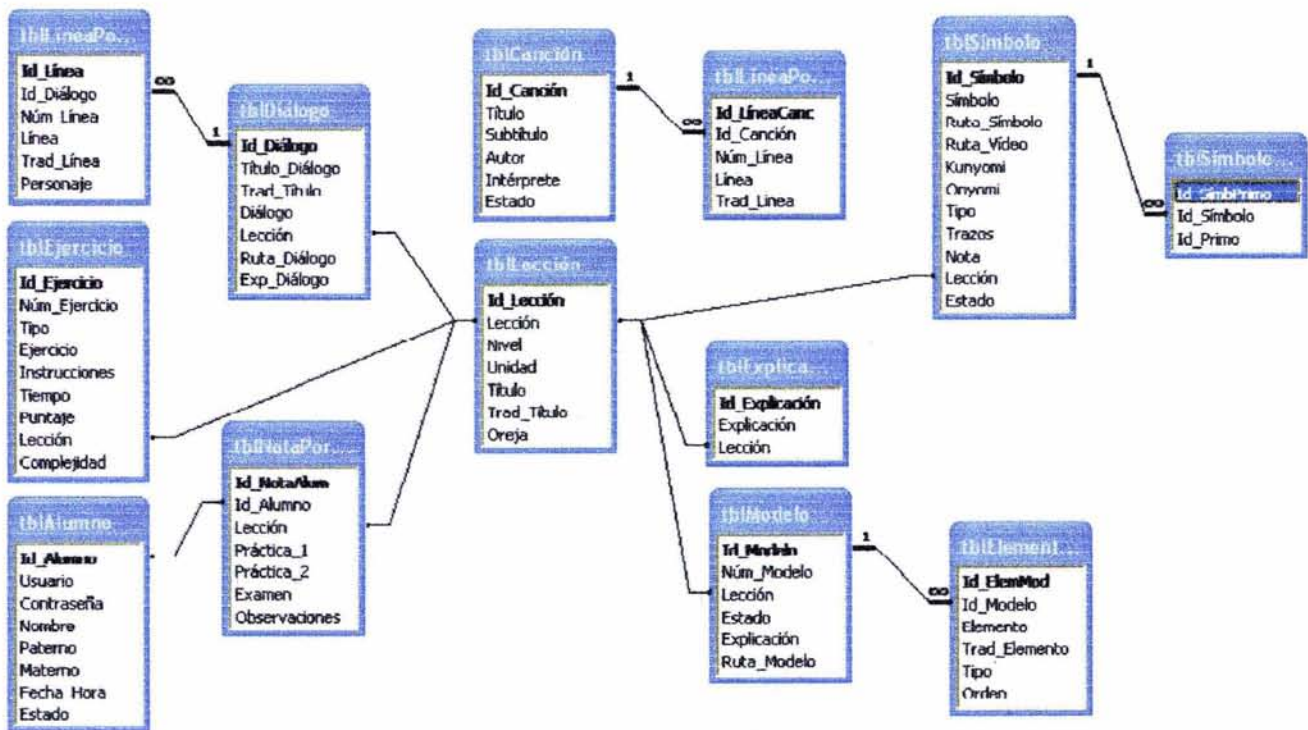
Segundo nivel



El diagrama anterior se ha construido a un segundo nivel, en el cual se pueden apreciar mejor algunas de las fases de las que estará compuesta cada lección, ya que contempla el diálogo, el análisis gramatical, la práctica I, la lecto-escritura, el vocabulario, la práctica II y la evaluación.

Es necesario mencionar que el flujo del sistema es progresista, o sea, generalmente va de menos a más (sin dejar fuera la posibilidad de regresar a lecciones anteriores con fines de repaso) y es una serie de ciclos, pues al término de la evaluación, si es aprobada al 100%, se da comienzo a la siguiente lección comenzando nuevamente por la sección del diálogo, que es el principio de toda lección.

3.1.3. Diagrama de entidad-relación.



Se puede decir que el diagrama de entidad-relación ofrece un panorama de la estructura interna del sistema. Es como si mirásemos un esqueleto humano, que es lo que da soporte a nuestro cuerpo. Entonces, cada una de las entidades de la base de datos son algo así como una vértebra o un hueso que está estrechamente vinculado a los demás mediante articulaciones, en este caso, esos vínculos reciben el nombre de relaciones.

El anterior diagrama muestra la forma en la que interactúan las diferentes tablas que componen una base de datos; las relaciones pueden ser de uno a uno, de uno a muchos o de muchos a muchos.

3.1 Diccionario de datos.

El diccionario de datos es una sección de la documentación de todo proyecto en donde se encuentra la descripción de cada una de las tablas que conforman la base de datos, especificando nombre de la tabla, campo, tipo de dato, longitud, etc.

tblAlumno				
Campo	Tipo	Longitud	Requerido	Descripción
Id_Alumno	Auto		Sí	(CP) Identificador de alumno.
Usuario	Texto	8	Sí	Nombre de inicio de sesión.
Contraseña	Texto	72	No	Clave de acceso.
Nombre	Texto	30	No	Nombre o nombres del alumno.
Paterno	Texto	30	No	Apellido paterno del alumno.
Materno	Texto	30	No	Apellido materno del alumno.
Fecha_Hora	Fecha/Hora		No	Fecha y hora en la que fue dado de alta el alumno.
Estado	Sí/No		No	Estado de actividad del registro.

tblCanción				
Campo	Tipo	Longitud	Requerido	Descripción
Id_Canción	Auto		Sí	(CP) Identificador de canción.
Título	Texto	50	No	Título de la canción.
Subtítulo	Texto	50	No	Título en español de la canción.
Autor	Texto	30	No	Nombre del autor de la canción.
Intérprete	Texto	30	No	Nombre del intérprete de la canción.
Estado	Sí/No		No	Estado de actividad del registro.

tblDiálogo				
Campo	Tipo	Longitud	Requerido	Descripción
Id_Diálogo	Auto		Sí	(CP) Identificador de diálogo.
Título_Diálogo	Texto	50	No	Título del diálogo.
Trad_Diálogo	Texto	50	No	Traducción del título del diálogo.
Diálogo	Memo		No	Desarrollo del diálogo.
Lección	Numérico		No	(CF) Número de lección.
Ruta_Diálogo	Texto	255	No	Ruta del archivo de audio que contiene el diálogo.
Exp_Diálogo	Memo		No	Explicación del diálogo.

tblEjercicio				
Campo	Tipo	Longitud	Requerido	Descripción
Id_Ejercicio	Auto		Sí	(CP) Identificador de ejercicio.
Núm_Ejercicio	Numérico		No	Número de ejercicio.
Tipo	Numérico		No	Tipo de ejercicio.
Ejercicio	Memo		No	Nombre del título del ejercicio.
Instrucciones	Memo		No	Instrucciones para resolver el ejercicio.
Tiempo	Numérico		No	Tiempo máximo para resolver el ejercicio.
Puntaje	Numérico		No	Puntaje por reactivo.
Lección	Numérico		No	(CF) Número de lección.
Complejidad	Numérico		No	Nivel de complejidad del ejercicio.

tblElementoDeModelo				
Campo	Tipo	Longitud	Requerido	Descripción
Id_ElemMod	Auto		Sí	(CP) Identificador de elemento de modelo
Id_Modelo	Numérico		No	(CF) Identificador de modelo.
Elemento	Texto	50	No	Elemento del modelo.
Trad_Elemento	Texto	50	No	Traducción del elemento del modelo.
Tipo	Texto	10	No	Tipo de elemento: fijo o variable.
Orden	Numérico		No	Número ordinal del elemento.

tblExplicación				
Campo	Tipo	Longitud	Requerido	Descripción
Id_Explicación	Auto		Sí	(CP) Identificador de explicación.
Explicación	Memo		No	Explicación.
Lección	Numérico		No	(CF) Número de lección.

tblLección				
Campo	Tipo	Longitud	Requerido	Descripción
Id_Lección	Auto		Sí	(CP) Identificador de lección.
Lección	Numérico		Sí	Número de lección.
Nivel	Numérico		No	Nivel en el que se encuentra la lección.
Unidad	Numérico		No	Unidad de estudio que abarca la lección.
Título	Texto	255	No	Título de la lección.
Trad_Título	Texto	255	No	Traducción del título de la lección.
Oreja	Memo		No	Rótulo que muestra nivel, unidad y lección de estudio.

tblLíneaPorCanción				
Campo	Tipo	Longitud	Requerido	Descripción
Id_LíneaCanc	Numérico		Sí	(CP) Identificador de línea de canción.
Id_Canción	Numérico		Sí	(CF) Identificador de canción.
Núm_Línea	Numérico		No	Número de línea.
Línea	Memo		No	Contenido de la línea.
Trad_Línea	Memo		No	Traducción de la línea.

tblLíneaPorDiálogo				
Campo	Tipo	Longitud	Requerido	Descripción
Id_Línea	Auto		Sí	(CP) Identificador de línea.
Id_Diálogo	Numérico		Sí	(CF) Identificador de diálogo.
Núm_Línea	Numérico		No	Número de línea.
Línea	Texto	255	No	Contenido de la línea.
Trad_Línea	Texto	255	No	Traducción del contenido de la línea.
Personaje	Texto	20	No	Nombre del personaje

tblModelo				
Campo	Tipo	Longitud	Requerido	Descripción
Id_Modelo	Auto		Sí	(CP) Identificador de modelo.
Núm_Modelo	Numérico		No	Número de modelo.
Lección	Numérico		Sí	(CF) Número de lección.
Estado	Sí/No		No	Estado de actividad del registro.
Explicación	Memo		No	Explicación del modelo.
Ruta_Modelo	Memo		No	Ruta donde reside el archivo que contiene ejemplos.

tblNotaPorAlumno				
Campo	Tipo	Longitud	Requerido	Descripción
Id_NotaAlum	Auto		Sí	(CP) Identificador de modelo.
Id_Alumno	Numérico		Sí	(CF) Identificador del alumno.
Lección	Numérico		Sí	(CF) Número de lección.
Práctica_1	Moneda		No	Calificación obtenida en un ejercicio de práctica 1.
Práctica_2	Moneda		No	Calificación obtenida en un ejercicio de práctica 2.
Examen	Moneda		No	Calificación de la evaluación por lección.
Observaciones	Memo		No	Observaciones sobre el desempeño del alumno.



tblPalabra				
Campo	Tipo	Longitud	Requerido	Descripción
Id_Palabra	Auto		Sí	(CP) Identificador de modelo.
Palabra	Texto	255	No	Palabra.
Palabra_Romaji	Memo		No	Palabra en roomaji.
Palabra_Kana	Memo		No	Palabra en kana.
Palabra_Kanji	Memo		No	Palabra en kanji.
Palabra_Español	Texto	255	No	Palabra en español.
Cód_Palabra	Texto	255	No	Código de palabra.
Trad_Palabra	Memo		No	Traducción de palabra.
Categoría	Memo		No	Categoría de la palabra.
Trad_Categoría	Memo		No	Traducción de la categoría.
Tipo	Numérico		No	Tipo de registro: Palabra o Frase.
Lección	Numérico		No	(CF) Número de lección.
Nota	Memo		No	Observaciones acerca del uso de la palabra o frase.
Ruta_Audio	Texto	255	No	Ruta donde reside el archivo de sonido.
Ruta_Imagen	Texto	255	No	Ruta donde reside el archivo de imagen.
Estado	Sí/No		No	Estado de actividad del registro.

tblSímbolo				
Campo	Tipo	Longitud	Requerido	Descripción
Id_Símbolo	Auto		Sí	(CP) Identificador de símbolo.
Símbolo	Texto	10	No	Nombre del símbolo.
Ruta_Símbolo	Texto	255	No	Ruta del archivo de imagen del símbolo.
Ruta_Vídeo	Texto	255	No	Ruta del archivo de vídeo del símbolo.
Kunyomi	Memo		No	Lectura kun'yomi.
Onyomi	Memo		No	Lectura on'yomi.
Tipo	Numérico		No	Tipo de símbolo: Hiragana, katakana o kanji.
Trazos	Numérico		No	Número de trazos.
Nota	Memo		No	Nota sobre el trazado del símbolo.
Lección	Numérico		Sí	(CF) Número de lección.
Estado	Sí/No		No	Estado de actividad del registro.

tblSímboloPrimo				
Campo	Tipo	Longitud	Requerido	Descripción
Id_SímPrimo	Auto		Sí	(CP) Identificador de pareja de símbolos primos.
Id_Símbolo	Numérico		Sí	(CF) Identificador de símbolo.
Id_Primo	Numérico		No	Identificador de símbolo parecido.

3.2 Interfaz de usuario.

La interfaz de usuario es lo que ve la persona que está frente al monitor. Debe de ser amigable y accesible para evitar causarle aversión a quien usará el programa. Por el contrario, el usuario necesita convencerse de que el sistema le servirá como herramienta de trabajo o estudio.

3.2.1 Pantalla de Inicio.

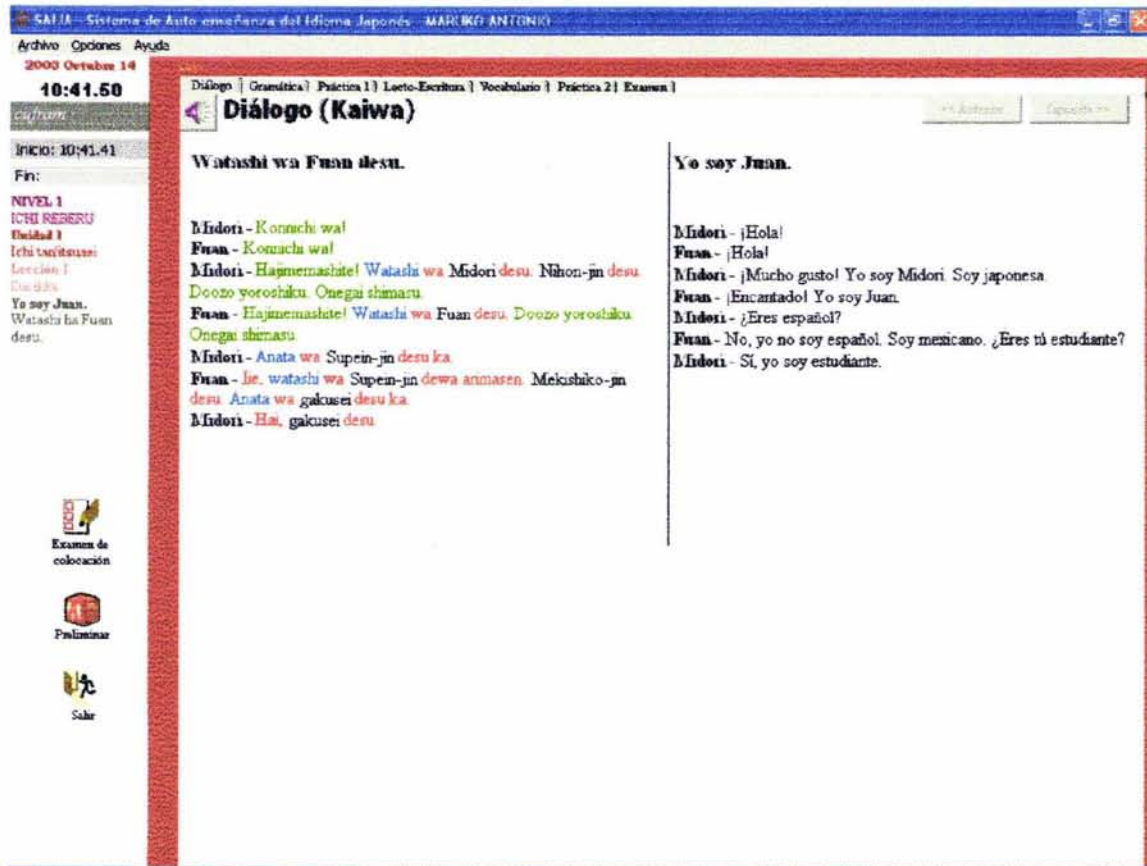
Al iniciar la sesión en Saija, aparecerá una ventana que solicitará el nombre de usuario y una contraseña para identificar a la persona que está frente al monitor con el sistema. Ahí mismo, habrá tres opciones a la derecha: *Iniciar* (comienza la sesión), *Registrarse* (abre un formulario para registrarse en caso de ser un nuevo usuario y *Salir* (termina la ejecución del programa).



3.2.2 Pantalla del módulo Principal.

La pantalla principal es donde se exponen los contenidos de cada una de las lecciones para ser estudiados por los alumnos. Está dividido en: Diálogo, Gramática, Práctica 1, Lecto-Escritura, Vocabulario, Práctica II y Evaluación.

La primera sección ofrece al alumno un diálogo con voces nativas para que pueda establecer un contacto con la ideología japonesa desde el principio, además de escuchar la entonación del idioma.



The screenshot displays the SAJJA software interface. At the top, the title bar reads "SAJJA - Sistema de Auto-enseñanza del Idioma Japonés - MARIBO ANTONIO". Below the title bar, there are menu options: "Archivo", "Opciones", and "Ayuda". The date "2003 Octubre 14" and time "10:41:50" are shown. The main content area is titled "Diálogo (Kaiwa)" and contains a dialogue between two characters, Midori and Fuan. The dialogue is presented in a two-column format. On the left, Midori's lines are shown, and on the right, Fuan's lines are shown. The dialogue includes greetings, introductions, and a conversation about nationality and being a student. The text is color-coded, with green for Midori's lines and red for Fuan's lines. The dialogue is as follows:

Watashi wa Fuan desu.

Midori - Konnichi wa!
Fuan - Konnichi wa!
Midori - Hajimemashite! Watashi wa Midori desu. Nihon-jin desu.
Doozo yoroshiku. Onegaishimasu.
Fuan - Hajimemashite! Watashi wa Fuan desu. Doozo yoroshiku.
Onegaishimasu.
Midori - Anata wa Supain-jin desu ka.
Fuan - Ie, watashi wa Supain-jin dewa arimasen. Meikushiko-jin desu. Anata wa gakusei desu ka.
Midori - Hai, gakusei desu.

Yo soy Juan.

Midori - ¡Hola!
Fuan - ¡Hola!
Midori - ¡Mucho gusto! Yo soy Midori. Soy japonesa.
Fuan - ¡Encantado! Yo soy Juan.
Midori - ¿Eres español?
Fuan - No, yo no soy español. Soy mexicano. ¿Eres tú estudiante?
Midori - Sí, yo soy estudiante.

Después, en la segunda sección, se realiza un análisis gramatical mediante modelos de estructura de frase. Cada modelo se ilustra con ejemplos claros, pues es muy importante que el alumno aprenda las reglas específicas que se emplean para cada diferente tipo de oración. Lo cual servirá de base para una buena conversación, más adelante.

The screenshot shows the SAJA application window. The title bar reads "SAJA - Sistema de Auto-enseñanza del Idioma Japonés MARLEJO ANTONIO". The menu bar includes "Archivo", "Opciones", and "Ayuda". The main content area is titled "Gramática (Bunpoo)" and "Modelo 1 (Bunkei 1) pronombre wa sustantivo desu." It contains three numbered lessons explaining the particle "wa" and the copula "desu" with examples and translations. A sidebar on the left shows navigation options like "Examen de colocación", "Problemas", and "Salir".

SAJA - Sistema de Auto-enseñanza del Idioma Japonés MARLEJO ANTONIO

Archivo Opciones Ayuda

2002 Octubre 14

18:43:35

Inicio: 10:41:41

Fin:

NIVEL 1
ICHI FUDZERI
Daijisho 1
Ichi taritamae
Lectura 1
Daijisho
Yo soy Juan.
Watashi ha Juan
desu

Examen de colocación

Problemas

Salir

Díptico [Gramática] Práctica 1 | Lecto-Escritura | Vocabulario | Práctica 2 | Examen 1

Gramática (Bunpoo)

Modelo 1 (Bunkei 1) *pronombre wa sustantivo desu.*

- Un **pronombre** (da-meishi), como ya se había visto, es aquella palabra que está en lugar del nombre. Yo, Tú, Él, etc.
Ejemplo 1 (rei ichi): **Watashi wa** Midori desu.
Yo soy Midori.
- La partícula **wa** (que se pronuncia [ua]) marca el tema de la oración y sería algo así como *A propósito de...* en español.
Ejemplo 2 (rei ni): **Watashi wa** Juan desu.
(A propósito de mí,) yo soy Juan.
- Un **nombre** o **sustantivo** (meishi) puede ser un *nombre de persona, cosa, animal, habitante de un país*, etc. La palabra **desu** no se traduce en español, pero más o menos significa *ser*.
Ejemplo 3 (rei san): **Anata wa** Nihon-jin desu.
Tú eres japonesa.
Ejemplo 4 (rei yon): **Watashi wa** gakusei desu.
Yo soy estudiante.

Anata normalmente significa *Usted*, pero a veces podría significar *Tú*.

* En japonés, no hay diferencia de género (masculino o femenino) ni de número (singular o plural) en la mayoría de las palabras. Esto es, da lo mismo emplear *Nihon-jin* para decir *japonés, japonesa, japoneses o japonezas*.

[Anterior modelo](#) [Siguiente modelo >>](#)

Otra de las secciones que merece ser mencionada aquí es la de Lecto-escritura. Aquí, se le presenta al alumno un conjunto de símbolos, uno a la vez, con vídeos que le muestran cómo realizar cada uno de los trazos que conforman la letra en cuestión. También, se le presenta información acerca de sus lecturas kun'yomi (lectura donde el símbolo va solo) y on'yomi (lectura donde el símbolo va acompañado), la cantidad de trazos, así como las letras que se le parecen en su escritura, si las hubiese, con la finalidad de que las distinga muy bien y no las vaya a confundir en un futuro, ya que desde ahorita debe de ser capaz de identificarlas plenamente. En la parte de abajo, a la izquierda, habrá una breve explicación sobre el trazado del símbolo y algunas palabras de ejemplo que utilizan en su escritura el símbolo que se está estudiando. A la derecha, se encuentra un panel llamado pizarra virtual en la que el alumno puede practicar la escritura del símbolo en cuestión.

The screenshot shows the SAIJA software interface. At the top, the title bar reads 'SAIJA - Sistema de Auto-enseñanza del Idioma Japonés - MARUKO ANTONIO'. The main window has a menu bar with 'Archivo', 'Opciones', and 'Ayuda'. Below the menu, the date is '2003 Octubre 14' and the time is '11:19.44'. A sidebar on the left contains a 'Inicio' section with the time '10:41.41' and a 'Fin:' section. Below this is a 'NIVEL 1' section with 'ICHI FEBERU', 'Eitōdō 1', 'Ichi tsūjūsanen', 'Lectura 1', and 'Un dōshi'. A small text box says 'Yo soy Juan. Watashi ha Juan desu.' Below the sidebar are icons for 'Examen de colocación', 'Preleminar', and 'Salir'. The main content area is titled 'Lecto-escritura (Kakiyomimono)'. It features a video player with a play button and a 'Siguiente >>' button. To the right of the video, it says 'Tipo de símbolo: Hiragana' and 'Letra: e'. Below that, it says 'Número de trazos: 2'. There are 'Anterior <<' and 'Siguiente >>' buttons. Below the video, there are 'Observaciones:' and 'Algunas palabras que comienzan con esta letra:'. The words are listed in a table:

え [e] pintura	えいこ [eiko] gloria	えいご [eigo] película
えいせい [eisei] higiene	えいゆう [eiyū] héroe	えいよう [eiyō] nutrición
えき [eki] estación	えだ [eda] rama	えび [ebi] camarón

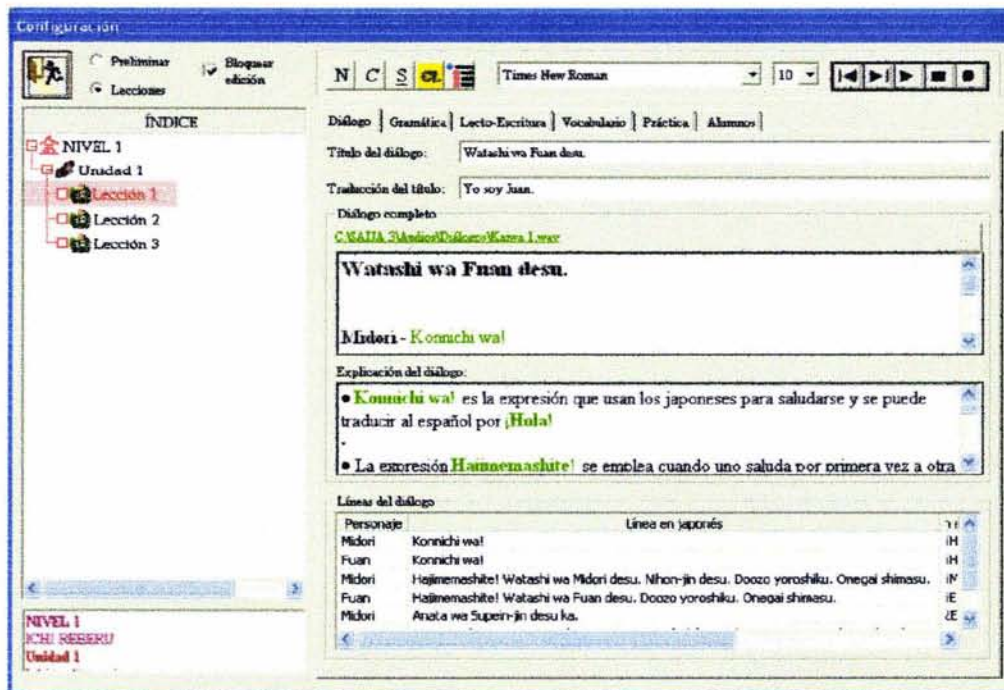
To the right of the table, there is a section titled 'Prueba a escribir la letra de arriba a la izquierda' with a large image of the character 'エ' showing its stroke order with arrows.

3.2.3 Pantalla del módulo de Configuración.

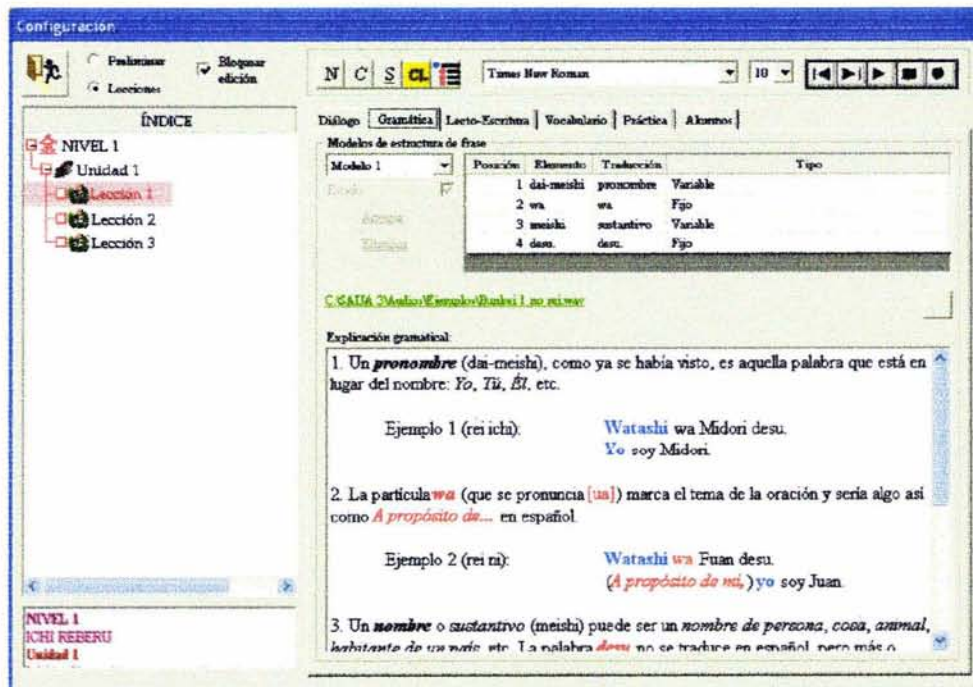
Mediante esta pantalla, el usuario (alumno o profesor) podrá configurar todas y cada una de las lecciones. Cuenta con seis apartados: *Diálogo*, *Gramática*, *Lecto-Escritura*, *Vocabulario*, *Práctica* y *Alumnos*. Este módulo fue diseñado para guardar los datos de manera automática, es decir, el menor cambio que realice el usuario será guardado en la base de datos inmediatamente, con el fin de facilitar la configuración del programa.

En la pestaña de *Diálogo* existen dos recuadros en la parte superior, en los cuales deberán ser tecleados tanto el título del diálogo como la traducción de dicho título. En la parte de en medio existen dos espacios para capturar el diálogo, con formato enriquecido de texto y una explicación del mismo. En la parte de abajo hay una cuadrícula tipo hoja de Excel que permite teclear línea por línea del diálogo, esto es para poder tratarlas a cada una con particularidad.

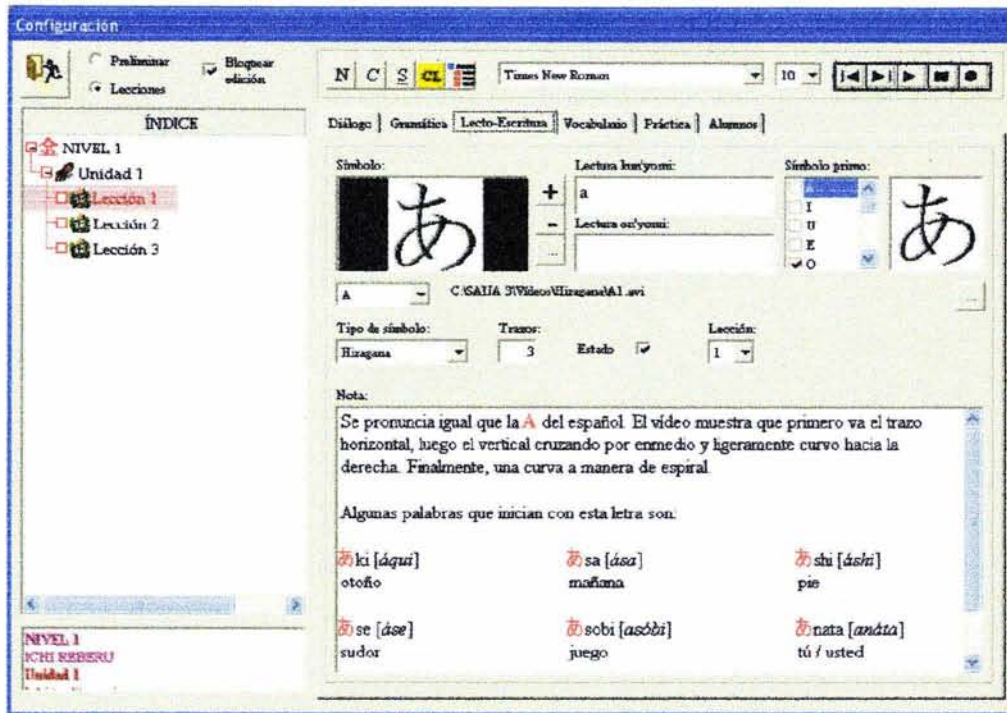
Es importante señalar que por omisión, la pantalla se encontrará bloqueada, es decir, sólo se podrá consultar la información existente en cada sección. Sin embargo, si se desea realizar alguna modificación en cualquier parte, hasta arriba se encuentra una casilla de verificación con una palomita, lo cual indica bloqueo de sólo lectura; al desactivarla, el programa permitirá realizar modificaciones, grabándose instantáneamente en la base de datos el menor cambio detectado.



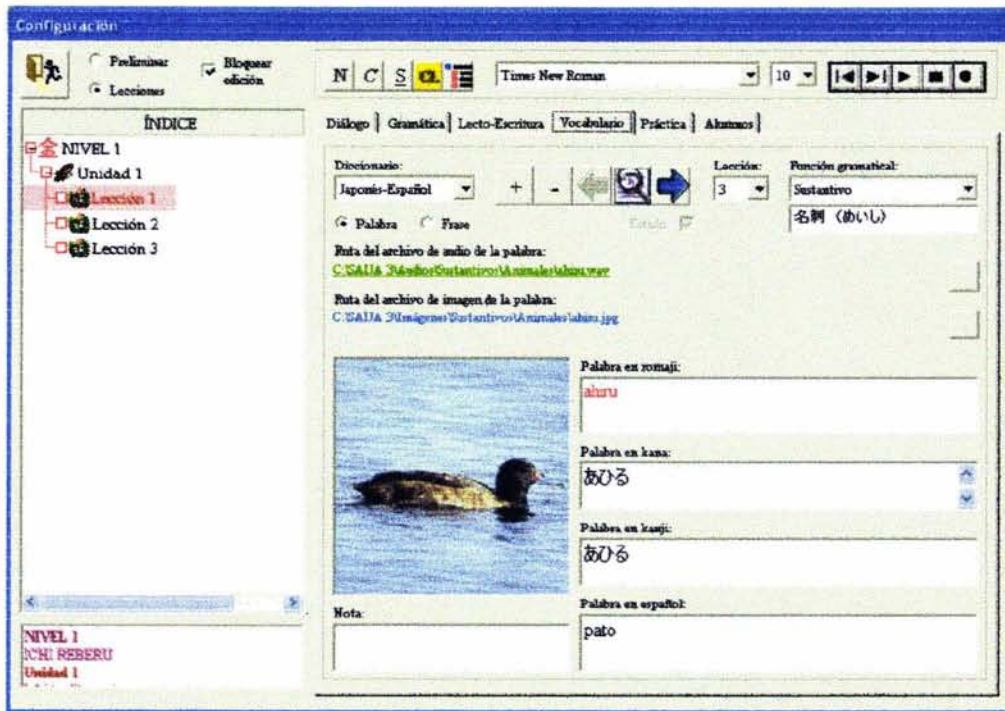
En la segunda pestaña, la de Gramática, se encuentra a arriba a la derecha una cuadrícula en la que se deben de capturar todos y cada uno de los elementos de que estará compuesto cada modelo, mientras que en la parte de abajo hay un cuadro de texto con formato enriquecido que permite registrar el modelo con algunos ejemplos, con la capacidad de cambiar la letra y el color de ésta. En la parte de en medio a la derecha se encuentra un botón que permite asociar un archivo de audio, que contenga ejemplos, al modelo en cuestión.



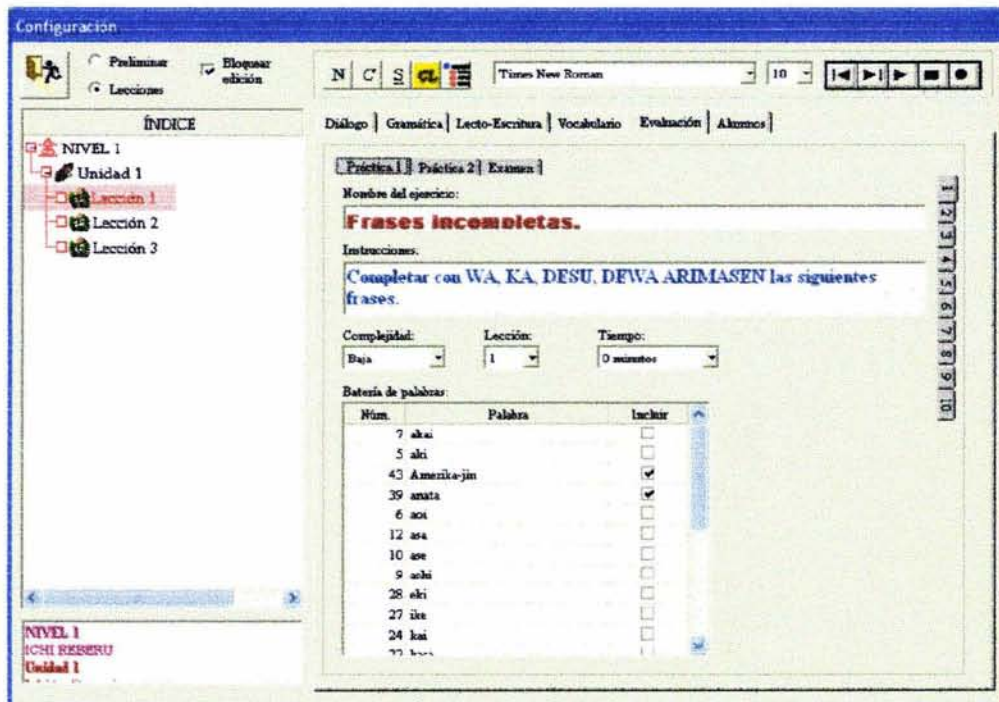
La tercera pestaña está dedicada a la Lecto-escritura y es aquí donde el usuario administrador del programa puede configurar los símbolos que se enseñarán en cada lección. En la parte de arriba, a la derecha está un cuadro que presenta un vídeo que ilustra el trazado de la letra en cuestión. En medio se encuentran unos recuadros que registrarán sus respectivas lecturas kun'yomi y on'yomi. Más a la derecha, existe una lista con todos los símbolos, en la cual, se pueden seleccionar aquellos que presenten una escritura similar a la del símbolo que se encuentra a la izquierda. Más abajo, se encuentran algunos datos del símbolo como son su tipo, número de trazos, ruta del archivo de vídeo, etc. Hasta abajo, se encuentra un gran cuadro de texto con formato enriquecido que permite capturar algunas observaciones acerca de cómo trazar cada símbolo, así como algunos ejemplos de palabras que se escriben con la letra en cuestión.



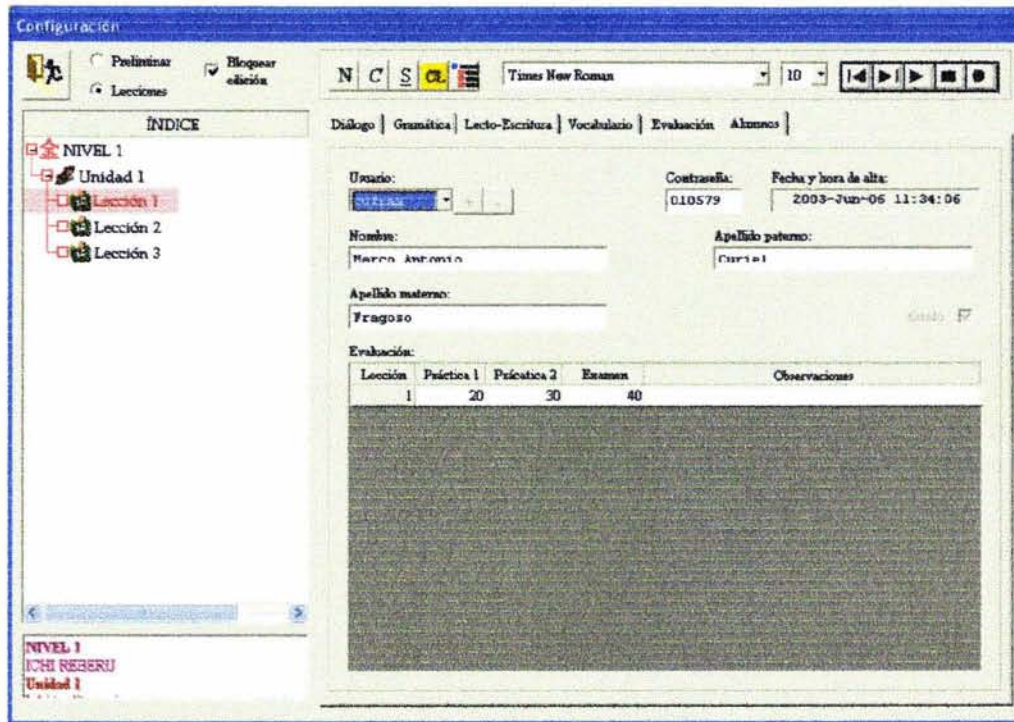
En la cuarta pestaña, encontramos la sección de Vocabulario, en donde se pueden gestionar las palabras y frases que serán mostradas en cada lección. Se pueden organizar en dos sentidos: del japonés al español y viceversa. En ambos casos, se requiere la captura de la palabra o frase en versión roomaji o alfabeto; la versión en hiragana o katakana sin el uso de kanji, la versión completa (con hiragana y kanji) y la equivalencia en español. También, se pueden asociar un archivo de sonido con la pronunciación y una imagen; se puede clasificar según la categoría a la que pertenezca donde, si es palabra podría por ejemplo quedar en Sustantivo, Adjetivo y, si es frase podría ser incluida en Saludos, Viajes, etc. Hay una pequeña caja situada justo debajo de la imagen en la que se puede escribir alguna observación en la que se desee hacer hincapié al usuario acerca del uso de la palabra o la frase en determinada situación o bajo ciertas condiciones.



La quinta pestaña se ocupará de administrar los juegos en las prácticas 1 y 2. Aquí se establecerán algunos patrones como son el nivel de dificultad, el tiempo, etc. En ella se encuentran dos conjuntos de pestañas cada uno con diez elementos que corresponden a las diez opciones de juegos o ejercicios para las prácticas 1 y 2, respectivamente. Además de, es necesario especificar el nombre del juego y las instrucciones que deberá seguir el usuario antes de empezar a practicar.

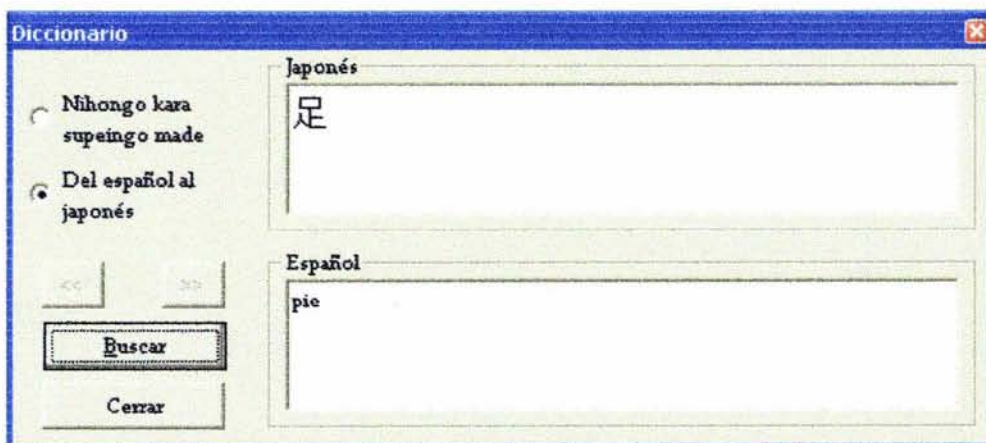


Por último, viene la sección de Alumnos que no tendrá otro fin más que el de administrar la lista de alumnos que se han matriculado en el sistema.



3.2.4 Pantalla del módulo de Diccionario.

Esta pantalla ilustra el módulo de Diccionario, herramienta que será muy útil al alumno cuando tenga dudas sobre cómo se escribe o pronuncia una palabra así como su significado.





Capítulo Cuarto - Desarrollo e implementación de la aplicación

El desarrollo de software consiste en traducir la solución al problema real a un lenguaje con el que el ser humano pueda comunicarse con la computadora. En este caso, hablamos del empleo de un lenguaje de alto nivel como Microsoft Visual Basic para dicho fin.

La implementación del software viene una vez que se ha terminado de programar cada rutina de cada módulo de que estará compuesto el sistema. Sin embargo, antes de implementar el sistema, se debe realizar una serie de pruebas de funcionamiento para asegurar que no surgirá ningún error al momento de estar trabajando con él.

4.1 Breves nociones de programación.

No cabe duda que el mundo de la programación visual y orientada a objetos es muy diferente a la programación procedural tradicional. Tan es así que aquellos que habíamos aprendido a trabajar con la programación procedural, al principio llegamos a tener muchos problemas para entender la nueva ideología, pero una vez entendido este nuevo concepto, es fácil adoptarla como propia y ser productivo de forma inmediata.

4.1.1 El cambio de concepto.

En realidad este cambio no debería ser tan complicado. Básicamente todo se sigue manejando por procedimientos, lo que cambia es la manera en que éstos son llamados. Abundo: En general la programación visual está orientada a lo que sucede en la interfaz. Esta interfaz debió haber sido previamente diseñada por algún programa de diseño de interfaces que por lo general viene incluido en el entorno de desarrollo del lenguaje elegido. En el caso de Visual Basic para Windows, este programa de diseño de la interfaz es lo que primero salta a la vista, lo mismo que en Delphi. En el caso de Visual Basic para DOS el programa de diseño de interfaz no salta a la vista de buenas a primeras, sino que debe ser ejecutado al indicarle al entorno de programación que se añadirá una nueva ventana de interfaz.

Aunque en los diversos entornos de programación a las áreas que se usan para diseñar la interfaz se les llaman de diferentes maneras, la mayoría coinciden en darle el calificativo de FORM (Formulario en Visual Basic, Forma en Delphi). En realidad el famoso *Form* es lo que el usuario ve en su pantalla como una ventana con diversas opciones que le muestran o le solicitan información. La mayoría de estas opciones son objetos intrínsecos que se llaman "Controles". En general estos

controles son los que tienen que ver con la manera en que funciona la aplicación visual. Al momento de diseñar la interfaz se colocan los controles deseados en el formulario. Así pues, suponiendo que crea un formulario para que capture el nombre del usuario y luego el programa le salude, lo que necesitará será colocar un control que permita al usuario capturar texto (*TextBox* en el caso de VB, *Edit* en Delphi) y un botón que ordene a la aplicación que salude al usuario (*CommandButton* en VB, *Button* en Delphi) sólo si el usuario ya tecleó algún texto.

4.1.2 La importancia de comprender qué son los eventos del sistema.

Los dos controles que han sido colocados van a responder a lo que el usuario haga. Si aún la idea no nos ha quedado clara, es aquí donde empieza el problema para los que estábamos acostumbrados a programar de la forma tradicional. Esto es, en la programación tradicional uno como programador le iba indicando al usuario en qué orden capturar la información necesaria, en la programación visual es el usuario el que elige el orden en el que hará las cosas. Usted no puede adivinar si el usuario primero capturar el texto y luego oprimirá el botón o viceversa, por lo tanto la programación visual podría exigir un pequeño esfuerzo extra para validar la información suministrada por el usuario.

Sin embargo, ¿cómo hacer para saber si el usuario está tecleando texto en el área destinada para ello? ¿O cómo hacer para determinar si el usuario ha oprimido el botón de comando? Para esto existen los denominados *Eventos*. Así es, la programación visual se centra principalmente en la determinación de aquellos sucesos que están ocurriendo en torno a la aplicación. El entorno que manipula la aplicación visual tiene la característica de estar monitoreando constantemente (y de forma transparente para el usuario y el programador) lo que está sucediendo, ya sean sucesos provocados por el usuario o por el propio sistema.

Así pues, el entorno es capaz de detectar si se está alterando el texto en el cuadro correspondiente o si el usuario ha oprimido (ha hecho clic) en el botón de comando. Cada uno de estos controles y todos los que se incluyen tienen, predefinidos, una serie de eventos que puede detectar y que, por lo tanto, se pueden codificar. Este código es encapsulado en lo que se denomina *Procedimientos de Evento*. Aunque cada control puede "sentir" *n* cantidad de eventos, sólo aquellos eventos a los que se les haya añadido código serán ejecutados. Ahora bien, de vuelta a nuestro ejemplo el usuario deberá poder oprimir el botón para saludar sólo si hay texto en el cuadro correspondiente. Esto podría hacerse de muchas maneras y una de ellas es la siguiente:

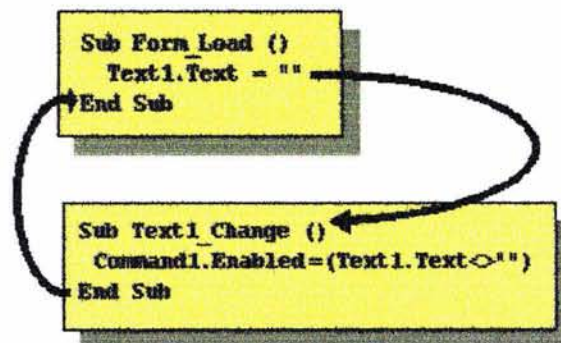
```
Sub Text1_Change()  
    Command1.Enabled = (Text1.Text <> "")  
End Sub  
  
Sub Command1_Click()  
    MsgBox "Hola " & Text1.Text  
End Sub
```

Como se puede ver, los procedimientos de evento se encuentran delimitados por las palabras `sub` y `End Sub`. Cada bloque de instrucciones entre esas dos palabras recibe el nombre de subprograma,

con lo cual se le dice a la computadora qué debe de hacer al momento en que se genere el evento al que pertenece el código del subprograma en cuestión.

Pero no todo termina allí. Como ya se había explicado, el sistema está verificando constantemente lo que está ocurriendo ya sea dentro del propio sistema o mediante el usuario. Así, el código de un evento puede ser provocado por una acción directa del usuario, del propio código del programa o del sistema. A esto se le llama *Evento en cascada*. Es decir, si el código asignado a un evento provoca otro evento, esto llevará a que el compilador ejecute el código del evento provocado; si éste a su vez provoca otro evento, pues el compilador ejecutará ese código, y así sucesivamente hasta que no se provoquen más eventos por el código o se acabe el espacio de la pila (stack).

Esto pudiese tomarse como ventaja. Si cuando se carga el formulario que sirve de ejemplo borramos el contenido de nuestras áreas de texto, automáticamente el botón de comando se establecerá a "Inhabilitado". Considérese la figura que ilustra esta acción: Cuando se ejecuta el código que establece a *vacío* el contenido del área de texto, ocurre el evento "Change" de este control, de tal forma que se ejecuta el código acorde que deshabilita el botón de comando y, luego de terminar la ejecución del código, se regresa a la siguiente línea de código que provocó el evento y continúa con el flujo normal del programa. Así el evento Load podría quedar de esta manera:



Se dice entonces que desde el evento Load se salta al evento Change y luego se regresa a aquel evento para terminar el flujo ahí mismo.

4.1.3 Convenciones de codificación.

En los primeros días de desarrollo en Visual Basic (1 y 2), se utilizaban convenciones de denominación que se derivaban de los estilos creados a través de los años al trabajar con otras herramientas de desarrollo. Al derivar las convenciones de VB desde el mundo de DOS, con las conocidas limitantes en la denominación de archivos y los nombres de los objetos, lo que se obtuvo fueron muchas denominaciones que, en la mayor cantidad de casos, eran incomprensibles para cualquiera de nosotros.

Así, en 1993 el propio Microsoft coadyuvó en dispersar la niebla al publicar la sección "Convenciones de denominación para los objetos" que se encuentra en el capítulo 3 de la documentación de VB3. Aquí recomendaron asignar un prefijo de tres letras para controlar los nombres. Con esto se pudo establecer todo un conglomerado de estándares para la denominación de objetos y variables que no se limitan a 3 letras, sino que permiten ubicar el contexto de una entidad en sí.

Pero, ¿para qué utilizar convenciones de denominación? La respuesta radica en la necesidad de estandarizar la estructura y el estilo de codificación de una aplicación de forma que el autor y otras personas puedan leer y entender el código fácilmente.

Las convenciones de codificación correctas dan como resultado un código fuente preciso, legible y sin ambigüedad, que es coherente con otras convenciones del lenguaje y lo más intuitivo posible.

4.1.4 Creación de nombres básicos para los objetos.

Si no se utiliza ninguna convención de denominación, cada entidad en su código podría tener un nombre que se le ocurrió al programador en ese momento y punto. Para los formularios y otros objetos, estos nombres podrían ser descriptivos como "vendedores" o "mis_amigos". En lo que toca a las variables, los programadores somos más economizadores (como con el clásico *Dim I As Integer*). Podríamos llamar a este punto inicial como "nombre inicial" o "nombre básico".

4.1.4.1 Nombres básicos para variables y constantes.

Tanto las variables como las constantes pueden tener tipos y ámbitos (o alcance). Esto es, puede existir alguna variable de tipo entero dentro de una rutina (local), la cual podría denominarse como *VLintContador*, donde "VL" indica que es una variable local e "int" que es entera; Si hay otra variable de tipo precisión simple que es global para todo un módulo, podría ser *VMsngTotal*, donde "VM" denota a una variable modular y "sng" que es de precisión simple; cuando se trate de una variable global a todo el proyecto podría usarse algo como *VGstrNombre*, donde "VG" indica variable global y "str" que es de tipo cadena. En cuanto a las constantes, se seguirá el mismo criterio, sólo que en vez de una "V" se usará una "C": *CMLngColorDeLetra*. Los parámetros utilizarán una "P" al inicio y tres letras que identifiquen el tipo requerido: *PblnBandera*, donde "bln" indica que es de tipo lógico.

4.1.4.2 Nombres básicos de tablas.

Al denominar las tablas de datos, como en la mayoría de las cosas en la vida, la moderación podría ser la mejor solución. Ni el abuso de nombres cortos en extremo, ni de aquellos cuya

longitud haga que se pase todo el día con la captura del puro nombre. Si se crea un grupo de abreviaciones estandarizadas para el desarrollo y se eliminan palabras raras, podría decirse que la creación y lectura posterior de nombres de tablas podrá ser un aspecto por demás sencillo y fácil de leer. Los nombres de los campos de una tabla simplemente deberán estar apegados a tres aspectos:

- Mantener los nombres cortos pero legibles.
- Utilizar abreviaturas estándar.
- Ser consistente al aplicar nombres en singular y plural.

4.1.4.3 Nombres básicos para consultas.

Los nombres de las consultas (query) son, por lo general, más largos que el promedio de los nombres de las tablas. Una de las razones por las que se sugieren nombres de tablas de hasta 15 caracteres es que los nombres de las consultas podrían apoyarse en el nombre de la tabla y, así, obtener nombres legibles de una longitud razonable. Sin embargo, por salud mental, es conveniente no sobrepasar 30 caracteres en la denominación de consultas, de modo que se continúe conservando esa combinación de simplicidad con legibilidad.

4.1.4.4 Nombres básicos para módulos de formulario y de clase.

Dado que estos módulos son de los más utilizados en una aplicación, podría preferirse ser descriptivo, pero sin exagerar. VB permite nombres de entidades de hasta 40 caracteres, pero pocos encontrarán útiles los nombres con estas longitudes.

Podría preferirse el establecer una longitud en la denominación de un formulario o clase. En lugar de utilizarse tan sólo un nombre, como Clientes, podría preferirse el identificar al formulario como frmClientes o a la clase como clsClientes. Por añadidura, tal como se ve, la denominación incluye el uso de altas y bajas (mayúsculas y minúsculas) para identificar las diferentes palabras que conforman el nombre (en lo que se ha dado en llamar: notación camello). Es decir, es más fácil leer: frmVentasEnTotal, que: FRMVENTASENTOTAL o frmventasentotal. Algunos prefieren utilizar el guión bajo (_) para separar las partes de la oración, como: frm_ventas_en_total. Esto último, a pesar de ser válido, está cayendo en desuso.

4.1.4.5 Nombres básicos para los controles.

Es sano utilizar el nombre del resultado o dato que se desplegará en un determinado control. Es decir, si al leer alguna información de una base de datos ésta se desplegará en algún cuadro de texto (en la suposición de que el dato sea un apellido paterno), el cuadro se podría llamar apellidoPaterno, apPaterno, o, como indica el prefijo, txtApellidoPaterno. (Véase Anexo 3).

4.1.4.6 Nombres básicos para los módulos de código.

Los módulos de código (también conocidos como módulos BAS, módulos de código plano, módulos Basic o tan sólo módulos) son muy subjetivos y siguen la convención que sea más conveniente para el(los) desarrollador(es) de la(s) aplicación(es). Tales convenciones podrían dictar el tipo de procedimientos que están relacionados con el módulo. Así, podrían usarse nombres como: modRutinasAPI_DelTechNet y modRutinasAPI_Biblioteca.

4.1.4.7 Nombres básicos para los procedimientos y funciones.

Este es un punto álgido en las tareas de definición de nombres. Sin embargo poco a poco se ha convertido en una buena norma el indicar una descripción precisa (y sin embargo corta) del procedimiento que se refiera. Normalmente, un procedimiento (Sub) se denomina con un verbo en infinitivo (o con una acción sobre un objeto): CalcularTotal. En cambio, una función, dado que devuelve un valor, podría ser denominada con un sustantivo y un adjetivo: CadenaInvertida().

4.2 Codificación de Saija.

A continuación, enlistaré el código de cada una de las funciones y procedimientos de los módulos de que está compuesto el sistema Saija. Como ya se había mencionado, un proyecto puede constar de módulos de formulario, módulos planos (sólo código), módulos de clase, diseñadores de ambiente de datos, etc. En este caso, el sistema Saija está constituido de únicamente módulos planos, formularios y diseñadores de ambiente de datos.

4.2.1 Módulo Saija.

En el módulo Saija se encuentran las siguientes declaraciones como encabezados:

```
Option Explicit      'Todas las variables deben declararse.  
'Base de datos -----  
Global VGdbsSaija As Database  
Global VGstrNombre As String  
Global VGstrRutaDiálogo As String  
Global VGstrRutaEjemplosModelo As String
```

La primera sentencia provoca que Visual Basic exija al usuario siempre declarar las variables que vaya a utilizar en el módulo en el que se encuentre trabajando.

Las siguientes instrucciones únicamente sirven como declaración de variables globales, es decir, variables que podrán ser manipuladas en cualquier parte del sistema.

Después de los encabezados, vienen algunos procedimientos y funciones que cuentan con un encabezado que incluye Autor, Última actualización (fecha), Descripción y Parámetros (si es necesario) con el fin de que la persona que lo mire el código se forme una idea general de lo que hace la rutina en cuestión. Esto lo he aplicado para todos y cada uno de los formularios que conforman el proyecto, por lo cual, volver a mencionar esta breve nota sale sobrando en los demás. Olvidaba mencionar que, con el propósito de identificar errores a la hora de realizar las pruebas de funcionamiento, he insertado antes de cada línea un "L" igualada a un valor numérico que corresponde a la línea que antecede. Es decir, si yo tengo "L = 5" significa que me encuentro en la línea 5 y por lo tanto, si aparece este número al momento de ocurrir un error, será más fácil ir directamente a la línea 5 a corregir la falla, en vez de estar buscando por todo el código. A continuación listaré los algoritmos más relevantes de este módulo:

FUNCIÓN Nombre(NomEsp)

// Función que obtiene el equivalente en japonés de un nombre en español.

```
Nomb = Minúsc(NomEsp)
A = 1
SI Izq(Nomb, 1) = 'h' ENTONCES Nomb = SubCad(Nomb, 2)
HACER HASTA QUE A > Long(Nomb)
  A = SubCad(Nomb, A, 1)
  SELECCIONAR CASOS DE A
  CASO 'c', 'f', 'l', 'r', 's', 'z'
    B = SubCad(Nomb, A + 1, 1)
    SI SubCad(Nomb, A, 2) <> 'll' Y B <> ('a', 'á', 'i', 'í', 'u', 'ú', 'e', 'é', 'o', 'ó') ENTONCES
      SI SubCad(Nomb, A, 2) <> ('ch', 'sh') O A = Long(Nomb) ENTONCES
        Nomb = SubCad(Nomb, 1, A) + 'u' + SubCad(Nomb, A + 1)
        A = A - 1
      FIN SI
    FIN SI
  CASO 'm'
    B = SubCad(Nomb, A + 1, 1)
    SI A = Long(Nomb) ENTONCES
      Nomb = SubCad(Nomb, 1, A) + 'u'
    SI NO SI B <> ('a', 'á', 'i', 'í', 'u', 'ú', 'e', 'é', 'o', 'ó', 'm') ENTONCES
      Nomb = SubCad(Nomb, 1, A - 1) + 'n' + SubCad(Nomb, A + 1)
      A = A - 1
    FIN SI
  CASO 'd', 't'
    B = SubCad(Nomb, A + 1, 1)
    SI B <> ('a', 'i', 'u', 'e', 'o') Y B <> A ENTONCES
      SI B <> ('s', 'z') ENTONCES
        Nomb = SubCad(Nomb, 1, A) + 'o' + SubCad(Nomb, A + 1)
        A = A - 1
      FIN SI
    FIN SI
  CASO 'h'
    SI SubCad(Nomb, A - 1, 1) <> 'c', 's' ENTONCES Nomb = SubCad(Nomb, 1, A - 1) + SubCad(Nomb, A + 1)
  FIN SELECCIONAR CASOS
  A = A + 1
FIN HACER
Sustituir(Nomb, 'á', 'a')
Sustituir(Nomb, 'é', 'e')
Sustituir(Nomb, 'í', 'i')
Sustituir(Nomb, 'ó', 'o')
Sustituir(Nomb, 'ú', 'u')
Sustituir(Nomb, 'x', 'kus')
Sustituir(Nomb, 'cc', 'kus')
Sustituir(Nomb, 'z', 's')
Sustituir(Nomb, 'ca', 'ka')
Sustituir(Nomb, 'cu', 'ku')
Sustituir(Nomb, 'qu', 'k')
Sustituir(Nomb, 'co', 'ko')
Sustituir(Nomb, 'ci', 'shi')
Sustituir(Nomb, 'si', 'shi')
Sustituir(Nomb, 'ja', 'ha')
```



```
Sustituir(Nomb, 'ji', 'hi')
Sustituir(Nomb, 'gi', 'hi')
Sustituir(Nomb, 'ju', 'fu')
Sustituir(Nomb, 'je', 'he')
Sustituir(Nomb, 'ge', 'he')
Sustituir(Nomb, 'jo', 'ho')
Sustituir(Nomb, 'du', 'do')
Sustituir(Nomb, 'tu', 'to')
Sustituir(Nomb, 'v', 'b')
Sustituir(Nomb, 'gue', 'ge')
Sustituir(Nomb, 'gui', 'gi')
Sustituir(Nomb, 'ñ', 'ny')
Sustituir(Nomb, 'mp', 'np')
Sustituir(Nomb, 'mb', 'nb')
Sustituir(Nomb, 'tz', 'ts')
Sustituir(Nomb, 'll', 'j')
Sustituir(Nomb, 'l', 'r')
RETORNAR Nomb
```

FIN FUNCIÓN

PROCEDIMIENTO SepararEnSilabas(Palabra, Silabas[])
//Procedimiento que separa una palabra japonesa en sílabas kana.

```
Aux = Palabra
A = 1
HACER HASTA QUE A = Long(Aux) + 1
  A = A + Subcad(Aux, A, 1)
  Prueba = Mayúsc(Subcad (Aux, A, 1))
  SALTAR A EsVocal
  SI EsVocal = CIERTO ENTONCES
#1  REDIMENSIONAR Silabas[LímSup(Silabas[]) + 1]
    Silabas[LímSup (Silabas[])] = A
    A = ''
  SI NO SI A = ('N', '') ENTONCES
    Prueba = Mayúsc (Subcad(Aux, A + 1, 1))
    SALTAR A EsVocal
    SI EsVocal = FALSO ENTONCES REGRESAR A #1
  FIN SI
  A = A + 1
FIN HACER
IR AL FIN
```

```
EsVocal:
  EsVocal = FALSO
  SELECCIONAR CASOS DE Prueba
    CASO 'A', 'I', 'U', 'E', 'O', ''
    EsVocal = CIERTO
  FIN SELECCIONAR CASOS
  VOLVER DE SALTAR A #1
FIN PROCEDIMIENTO
```

4.2.2 Formulario Carátula.

El formulario Carátula tiene como finalidad precisamente mostrar una carátula o pantalla de inicio toda vez que es arrancado el programa, además de mostrar unos campos que solicitan al usuario su identificación con el sistema, esto es, su *nombre de usuario* y su *contraseña*. En dicha pantalla se puede registrar a un nuevo alumno, haciendo clic en la opción correspondiente, lo cual tendrá como efecto mostrar otra pantalla en la que se pueden grabar los datos de un nuevo usuario del sistema.

4.2.3 Formulario Principal.

Ésta es la pantalla principal del sistema, ya que en ella se encuentra todo el material audiovisual con el que estará trabajando el alumno, a lo largo de cada una de las lecciones que conforman el curso. A la izquierda se encuentra la fecha y la hora hasta arriba y más abajo, el nombre del usuario activo e información sobre la lección en curso. Del lado derecho se encuentra un conjunto de fichas en el siguiente orden: diálogo, gramática, ejercicios I, lecto-escritura, vocabulario, ejercicios II y evaluación. Se van activando en orden progresivo, conforme el alumno vaya realizando cada una de las actividades que el sistema le vaya pidiendo.

Enseguida, escribiré los principales algoritmos de este módulo:

```

PROCEDIMIENTO CompletarEnunciados(Cant)
// Procedimiento que prepara el ejercicio de Completar enunciados, dejando un espacio en blanco.
SQL = 'SELECT * FROM tblModelo WHERE Lección = ' + Lección + 'ORDER BY Id_Modelo'
ConjDatos.AbrirConsulta(SQL)
SI ConjDatos->NúmRegs ENTONCES
  REDIMENSIONAR Mod[ConjDatos->NúmRegs]
  REDIMENSIONAR ContMod[ConjDatos->NúmRegs]
SI NO
  SALIR DE PROCEDIMIENTO
FIN SI
//Llena el arreglo con los identificadores de modelos
HACER HASTA QUE ConjDatos->FinDeArchivo
  Mod[ConjDatos->PosActual] = ConjDatos->Campos->Id_Modelo
  ConjDatos.SigReg
FIN HASTA
PARA A = 1 HASTA Cant
  CREAR OBJETO rtxFrase[A & 0 & 0]
  CREAR OBJETO imgCalif[A]
  rtxFrase[A + 0 + 0]->Visible = CIERTO
  imgCalif[A]->Visible = CIERTO
  rtxFrase[A + 0 + 0]->CoordIzq = imgCalif[A]->Anchura
  rtxFrase[A + 0 + 0]->CoordSup = (rtxFrase[A + 0 + 0]->Altura + 50) * (A - 1)
  imgCalif[A]->CoordSup = rtxFrase[A + 0 + 0]->CoordSup
  rtxFrase[A + 0 + 0] = A + '.'
  // Toma aleatoriamente un número de modelo
  HACER
    HACER
      B = Entero((LímSup(Mod[]) * NúmAleat()) + 1)
    FIN HASTA QUE (B ≠ Ant)
  FIN HASTA QUE (ContMod[B] ≤ (Cant \ LímSup(Mod[]))) O (A = Cant)
  ContMod[B] = ContMod[B] + 1
  Ant = B
  SQL = 'SELECT * FROM tblElementoDeModelo WHERE Id_Modelo = ' + Mod[B] + 'ORDER BY Orden'
  ConjDatos.AbrirConsulta(SQL)
  YaHayVacio = FALSO
  HACER //Toma aleatoriamente el lugar de la frase que irá en blanco
    HACER
      D = Entero((NúmAleat() * ConjDatos->NúmRegs) + 1)
    FIN HACER HASTA QUE D ≠ Ant2
  ConjDatos.PrimReg
  HACER HASTA QUE ConjDatos->FinDeArchivo
    SI D = ConjDatos->PosActual Y ConjDatos->Campos->Tipo = 'Fijo' ENTONCES
      YaHayVacio = CIERTO
    SALIR DE HACER
  FIN SIN
  ConjDatos.SigReg
  FIN HACER
  ConjDatos.PrimReg
FIN HACER HASTA QUE YaHayVacio = CIERTO
  Ant2 = D
HACER HASTA QUE ConjDatos->FinDeArchivo
  C = A + Formato(ConjDatos->PosActual, '00')
  CREAR OBJETO rtxFrase[C]
  rtxFrase[C]->Visible = CIERTO
  rtxFrase[C]->CoordIzq = rtxFrase[C - 1]->CoordIzq + rtxFrase[C - 1]->Anchura
  SI ConjDatos->PosAct ≠ D ENTONCES
    SI ConjDatos->Campos->Tipo = 'variable' ENTONCES
      SQL = 'SELECT Palabra FROM tblPalabra WHERE Trad_Categoría = ''

```



```
SQL = SQL + ConjDatos->Campos->Trad_Elemento + '' AND Id_Palabra IN (SELECT DISTINCT Id_Palabra '
SQL = SQL + 'FROM tblBateriaDePalabras)'
ConjDatos.AbrirConsulta(SQL)
SI ConjDatos->NúmRegs ENTONCES
    ConjDatos.Mover(Entero(NúmAleat() * ConjDatos->NúmRegs)
    rtxFrase[C] = ConjDatos->Campos->Palabra
SI NO
    rtxFrase[C] = ConjDatos->Campos->Elemento
FIN SI
SI SI
SI NO
    rtxFrase[C] = ConjDatos->Campos->Elemento
FIN SI
    rtxFrase[C]->Anchura = AnchoDeTexto(rtxFrase[C]->Texto) + 102
SI NO
    rtxFrase[C] = ConjDatos->Campos->Elemento
    rtxFrase[C]->Anchura = AnchoDeTexto(rtxFrase[C]->Texto) + 102
    rtxFrase[C] = ''
FIN SI
    rtxFrase[C]->CoordSup = rtxFrase[C - 1]->CoordSup
    ConjDatos.SigReg
FIN HACER
FIN PARA
FIN PROCEDIMIENTO
```

PROCEDIMIENTO Memorama(Tamaño)

// Procedimiento que genera un tablero con símbolos japoneses para jugar al memorama.

```
vgdTablero->Rengs = 0
vgdTablero->Cols = 0
vgdTablero->Rengs = 6
vgdTablero->Cols = 5
SQL = 'SELECT * FROM tblSímbolo WHERE Lección = ' + Lección
ConjDatos.AbrirConsulta(SQL)
REDIMENSIONAR Símbola[1]
// Recupera los símbolos de la lección en curso
HACER HASTA QUE ConjDatos->FinDeArchivo
    PARA B = 1 HASTA 2
        HACER
            R = Entero(NúmAleat() * 5)
            C = Entero(NúmAleat() * 4)
            FIN HACER HASTA QUE vgdTablero ->Casilla[R, C] = ''
            vgdTablero ->Casilla[R, C] = ConjDatos->Campos->Símbolo
            SI HayDir(ConjDatos->Campos->Ruta_Símbolo) ≠ '' ENTONCES vgdTablero->Casilla[R, C] =
                CargarImagen(ConjDatos->Campos->Ruta_Símbolo)
            vgdTablero->AltReng[R] = 1200
            vgdTablero->AnchCol[C] = 1200
            vgdTablero->Casilla[R, C].EstirarImagen
            vgdTablero->Casilla[R, C]->Relleno = 100
            vgdTablero->Reng = R
            vgdTablero->Col = C
            vgdTablero->Casilla[R, C]->Alineación = CentralCentral
        FIN PARA
        ConjDatos.SigReg
    FIN HACER
FIN PROCEDIMIENTO
```

PROCEDIMIENTO SopaDeLetras(Tamaño)

// Procedimiento que forma un tablero con palabras ocultas entre varias letras dispuestas aleatoriamente.

```
SI Tamaño = 0 ENTONCES Salir()
vgdTablero->Rengs = 0
vgdTablero->Cols = 0
vgdTablero->Rengs = Tamaño
vgdTablero->Cols = Tamaño
PalabrasSopa->Rengs = 0
PARA A = 0 HASTA Tamaño - 1
    vgdTablero->AlturaReng[A] = 400
    vgdTablero->AnchuraCol[A] = 400
SIGUIENTE A
```



```
// Recupera las palabras de esta lección
Lección = Valor(&vgdÍndice->Casilla[vgdÍndice->Reng, 2])
SQL = 'SELECT * '
SQL = SQL + 'FROM tblPalabra '
SQL = SQL + 'WHERE Lección = ' + Cadena(Lección)
ConjDatos.AbrirConsulta(SQL)
// Comienza la colocación de palabras
HACER HASTA QUE ConjDatos->FinArchivo = CIERTO
Palabra = ConjDatos->Campos->Cód_Palabra)
PalabrasSopa.Añadir(ConjDatos->Campos->Palabra)
&PalabrasSopa->Casilla[vgdPalabrasSopa->Rengs - 1, 0] = ConjDatos->Campos->Cód_Palabra
SepararEnSilabas(Palabra, Silabas[])
B = LimSup(Silabas[])
// Sitúa al azar las palabras
HACER
#1 SI MalSentido = CIERTO ENTONCES
    Sentido = 0
    SI R1 - B ≤ 0 O C1 + B > (vgdTablero->Cols - 1) ENTONCES
        Sentido = 1
        SI C1 + B > (vgdTablero->Cols - 1) ENTONCES
            Sentido = 2
            SI R1 + B > (vgdTablero->Rengs - 1) O C1 + B > (vgdTablero->Cols - 1) ENTONCES
                Sentido = 3
                SI R1 + B > (vgdTablero->Rengs - 1) ENTONCES
                    Sentido = 4
                    SI (R1 + B) > (vgdTablero->Rengs - 1) O (C1 - B ≤ 0) ENTONCES
                        Sentido = 5
                        SI R1 - B ≤ 0 ENTONCES
                            Sentido = 6
                            SI R1 - B ≤ 0 Y C1 - B ≤ 0 ENTONCES Sentido = 7
                        FIN SI
                    FIN SI
                FIN SI
            FIN SI
        FIN SI
    Sentido = Sentido + 1
SI NO
    Sentido = Entero(Aleatorio() * 8)
    HACER
        R1 = ValorAbs(Entero(Aleatorio() * (Tamaño - 1))) + 1
        C1 = ValorAbs(Entero(Aleatorio() * (Tamaño - 1))) + 1
    FIN HACER HASTA QUE vgdTablero->Casilla[R1, C1] = ''
FIN SI
MalSentido = FALSO
SI Sentido = 1 ENTONCES // Dirección Noreste (NE)
    SI R1 - B ≤ 0 O C1 + B > (vgdTablero->Cols - 1) ENTONCES
        MalSentido = CIERTO
    SI NO
        MR = -1
        MC = 1
    FIN SI
SI NO SI Sentido = 2 ENTONCES //Dirección Este (E)
    SI C1 + B > (vgdTablero->Cols - 1) ENTONCES
        MalSentido = CIERTO
    SI NO
        MR = 0
        MC = 1
    FIN SI
SI NO SI Sentido = 3 ENTONCES //Dirección Sureste (SE)
    SI R1 + B > (vgdTablero->Rengs - 1) O C1 + B > (vgdTablero->Cols - 1) ENTONCES
        MalSentido = CIERTO
    SI NO
        MR = 1
        MC = 1
    FIN SI
SI NO SI Sentido = 4 ENTONCES //Dirección Sur (S)
    SI R1 + B > (vgdTablero->Rengs - 1) ENTONCES
        MalSentido = CIERTO
    SI NO
        MR = 1
        MC = 0
    FIN SI
SI NO SI Sentido = 5 ENTONCES //Dirección Suroeste (SO)
    SI (R1 + B) > (vgdTablero->Rengs - 1) O (C1 - B ≤ 0) ENTONCES
        MalSentido = CIERTO
    SI NO
        MR = 1
```




```
MC = -1
FIN SI
SI NO SI VLintSentido = 6 ENTONCES //Dirección Oeste (O)
SI R1 - B ≤ 0 ENTONCES
  MalSentido = True
SI NO
  MR = 0
  MC = -1
FIN SI
SI NO SI Sentido = 7 ENTONCES //Dirección Noroeste (NO)
SI R1 - B ≤ 0 Y C1 - B ≤ 0 ENTONCES
  MalSentido = CIERTO
SI NO
  MR = -1
  MC = -1
FIN SI
SI NO //Dirección Norte (N)
SI C1 - B ≤ 0 ENTONCES
  MalSentido = CIERTO
SI NO
  MR = -1
  MC = 0
FIN SI
FIN SI
FIN HACER MIENTRAS MalSentido
SI R1 < 0 O C1 < 0 ENTONCES SALTAR A #1
vgdTablero->Casilla[R1, C1] = Silabas[1]
R = R1
C = C1
Redimensionar(Rx[LímSup[Silabas]])
Redimensionar(Cx[LímSup[Silabas]])
Rx[1] = R1
Cx[1] = C1
PARA A = 1 HASTA B - 1
  R = R + MR
  C = C + MC
  SI R < 0 O C < 0 ENTONCES //Si no se sale de la cuadrícula
    vgdTablero->Casilla[R1, C1] = ''
    SALTAR A BorrarPalabra
  SI NO
    SI vgdTablero->Casilla[R, C] = '' ENTONCES //Si está disponible la casilla
      vgdTablero->Casilla[R, C] = Silabas(A + 1)
      Rx[A + 1] = R
      Cx[A + 1] = C
    SI NO
      SALTAR A BorrarPalabra
  FIN SI
FIN SI
FIN PARA A
CongDatos.SigReg
FIN HACER
SQL = 'SELECT * '
SQL = SQL + "FROM tblSímbolo "
SQL = SQL + "WHERE Lección = " + Cadena(Lección)
CongDatos.AbrirConsulta(SQL)
SI CongDatos->TotRegs > 0 ENTONCES Redimensionar(Símbolos[1 HASTA CongDatos->TotRegs])
HACER HASTA QUE CongDatos ->FinArchivo
  Símbolos(CongDatos->RegActual) = Trim(!Símbolo)
  CongDatos.SigReg
FIN HACER
PARA R = 0 HASTA Tamaño - 1
  PARA C = 0 HASTA Tamaño - 1
    SI vgdTablero->Casilla[R, C] = '' ENTONCES
      vgdTablero->Casilla[R, C] = Símbolos[Entero(Aleatorio() * (LímSup(Símbolos[]) - 1)) + 1]
    FIN SI
    SQL = 'SELECT * '
    SQL = SQL + 'FROM tblSímbolo '
    SQL = SQL + 'WHERE Símbolo = ' + CadenaSinSubcadena(Minúsc(Tablero->Casilla[R, C]), '') + ''
    ConjDatos.AbrirConsulta(SQL)
    SI ConjDatos->TotRegs ENTONCES
      SI Dir(ConjDatos->Campos->Ruta_Símbolo) ≠ '' ENTONCES Tablero->Casilla[R, C] = _
      CargarImagen(ConjDatos->Campos->Ruta_Símbolo)
    FIN SI
  FIN PARA C
FIN PARA R
FIN PARA R
lblNúmPalabras->Texto = 0
Salir()
```



BorrarPalabra:

```
PARA X = 1 HASTA LimSup{Rx[]}  
  vgdTablero->Casilla[Rx[X], Cx[X]] = ''  
FIN PARA X  
SALATAR A #1  
FIN PROCEDIMIENTO
```

4.2.4 Formulario de Configuración.

El módulo de Configuración es una pantalla en la cual se puede dar mantenimiento al sistema, es decir, administrar los catálogos de usuarios, símbolos, palabras, etc. Asimismo, en esta ventana se puede configurar el contenido de cada lección, o sea, el texto del diálogo, los modelos de estructura gramatical de frase, los archivos de audio e imágenes relacionados con las actividades que así lo requieran, entre otros elementos.

Cada cambio que se realice será guardado en automático, sin necesidad de oprimir algún botón de “guardado” ni mucho menos. Además, el módulo cuenta con un seguro que activa el modo de sólo lectura para evitar hacer modificaciones involuntarias a la información que existe en la base de datos.

La intención de este módulo es permitir que un usuario avanzado, como puede ser un profesor de japonés, programe determinada lección, agregándole o quitándole información. Lo anterior da como resultado una aplicación suficientemente abierta y actualizable.

4.2.5 Formulario de Diccionario.

El formulario de Diccionario, como su nombre lo indica, consiste en un diccionario bilingüe japonés-español, español-japonés que sirva como herramienta de consulta al alumno, cuando se encuentre estudiando determinada lección.

En esta pantalla, el alumno puede escribir una palabra dada para que el sistema inicie una búsqueda a lo largo del catálogo de palabras y, después puede mirar su traducción ya sea del español al japonés o viceversa, y también puede escuchar la pronunciación de dicha palabra. Con todo esto, le estaremos dando al alumno la “materia prima” con la que pueda producir japonés, ya sea escrito u oral, pues la manera de formar las oraciones, de acuerdo con ciertos modelos de estructura de frase, ya la habrá estudiado en la sección de gramática.

Conclusiones

El programa SAIJA es una potente herramienta de cómputo que apoyará al estudiante de japonés para reforzar sus conocimientos en este idioma de manera interactiva.

En México hay muy pocas herramientas informáticas relacionadas con la enseñanza del idioma japonés. En Internet, la gran mayoría de los sitios que tratan de esta lengua son de origen español o estadounidense. Las expresiones idiomáticas del japonés que ahí aparecen son traducidas de acuerdo a la ideología de los países en cuestión y por lo tanto, carecen de sentido para el alumno mexicano. Por esto, SAIJA es un software que verdaderamente inicia al alumno mexicano en el estudio del japonés, mostrándole en cada una de las lecciones las equivalencias gramaticales que tiene este idioma con respecto al español que se habla en nuestro país.

SAIJA, al ser un método audiovisual ofrece muchas ventajas, unas de ellas son las que se enumeran a continuación:

1. Incluye un apartado que contiene algunas definiciones básicas de gramática que nos enseñaron durante el período de nuestra educación elemental, por si el alumno ya no las recuerda.
2. La pronunciación de los diálogos es clara y la velocidad va aumentando de acuerdo con el nivel de las lecciones.
3. Las explicaciones de gramática se ilustran con modelos de estructura de frase y con ejemplos simples y claros.
4. El primer apartado de ejercicios pone a prueba al alumno en su habilidad para construir frases según el orden de las palabras (sintaxis) que aprendió en la sección de gramática.
5. Cuenta con un apartado de lecto-escritura que muestra al alumno símbolo por símbolo, desde el silabario hiragana hasta los kanji, indicándole claramente mediante un vídeo la cantidad de trazos, el orden, el tipo y el sentido de éstos, con la posibilidad de que pueda practicar dibujándolos en una pizarra virtual, mediante el uso del ratón o un lápiz óptico. También se le dan algunos ejemplos de palabras que se escriben con el símbolo que se esté estudiando, así como su pronunciación.
6. Incluye una sección de vocabulario en la que se enseña al alumno la pronunciación de cada palabra, la forma de escribirlo en roomaji (letras de nuestro alfabeto), hiragana puro, hiragana combinado con kanji y su traducción al español, incluyendo una imagen alusiva a la palabra o expresión que se esté estudiando.



7. Existe un apartado de ejercicios que miden la destreza que el alumno ha ido desarrollando para memorizar e identificar los símbolos aprendidos en la lección en curso.
8. Cuenta con una sección de evaluación al final de cada lección que tiene la finalidad de probar los conocimientos globales que ha adquirido el alumno en la lección actual, permitiéndole saber el nivel que va alcanzando a lo largo del curso.

Con todo lo anterior, puedo concluir que SAIJA es un programa que vincula todos los elementos necesarios para apoyar didácticamente al alumno que esté estudiando japonés, de una forma accesible, sin demasiada teoría pero sí marcando algunas notas importantes que debe tomar en cuenta en la gramática japonesa, además de que este software cuenta con mecanismos de evaluación que le permiten conocer al estudiante el avance que ha obtenido durante el proceso de auto-enseñanza. Por esta razón, el sistema SAIJA es bastante redituable para todo aquel que desee iniciarse en el estudio de este idioma o simplemente reforzar los conocimientos que ya de por sí posee del mismo.

Bibliografía

Referencias bibliográficas

- *Nihongo with the compliments of the Japan Foundation.* Japanese Language, Urawa. Edit. Bunka Institute of Language, Japan, 2000.
- *Guía básica para principiantes en el estudio del idioma japonés.* Edit. The Japan Foundation, Japanese Language Institute.
- *Basic kanji book, vol. I.* Kano, Chieko, Yuri Shimizu, et. al. Edit. Bonjinsha Co., Ltd. Japan, 2000.
- *El japonés escrito.* Francisco De la Corte. Edit. Hiperión, España, 1998.
- *Análisis y diseño de sistemas de información.* James Senn. 2ª. edición, Edit. McGraw-Hill, México, 1990.
- *Aprenda Visual Basic 6.0 ya.* Michael Halvorson. Edit. McGraw-Hill, España, 1999.
- *Nihongo, japonés para hispanohablantes. Libro de texto/1.* Junichi Matsuura y Lourdes Porta Fuentes. 2ª. edición, Edit Herder, España, 2000.
- *Nihongo, japonés para hispanohablantes. Cuaderno de ejercicios complementarios/1.* Junichi Matsuura y Lourdes Porta Fuentes. 2ª. edición, Edit Herder, España, 2000.

Referencias en Internet

- <http://www.pue.udlap.mx/boletin/bbs/b312.html>
- http://www.lsi.ugr.es/~gunkan/html/gunkan_apendice/gunkan_Nihongo_apendice_006.html
- <http://www.proel.org/alfabetos.html>
- <http://www.nipoweb.com/>

- <http://www.lawebdelprogramador.com/>
- http://www.salohogar.com/espanol/lenguaje/lengua/que_gramatica.htm
- http://rose.ruru.ne.jp/multiplication/index_es.html
- http://ourworld.compuserve.com/homepages/Miyuki_Eusebio/nsindex.htm
- <http://www.dreamers.com/nihongo/>
- <http://www.publilatin.com/nihongo1.html>
- <http://www.spain-japan.com/languages/japanese.html>
- <http://www.japanese-online.com/>
- <http://www.timonel.com.ar/idiomas6.html>
- <http://www.hikyaku.com/fljap/fljapg.html#part1>
- <http://asakiyumemishi.com/gunkan/cjapo2/>
- <https://maktub.eps.ua.es/aeja/clases.html>
- http://www.emagister.com/frame.cfm?id_centro=25022080010157686969496653564555&id_curso=39768030032648517070546953674553&url_frame=http://www.solotutoriales.com/visitar.asp?id=390
- <http://www.pro-3.com.mx/ab10.htm>
- <http://pronad.uson.mx/docs/acapulco/unam.htm>

Anexos

Anexo 1. El silabario hiragana.

日本語全部平仮名表

	a	i	u	e	o		ya	yu	yo
	あ	い	う	え	お		や	ゆ	よ
k	ka	ki	ku	ke	ko	ki	kya	kyu	kyo
	か	き	く	け	こ		きや	きゆ	きよ
g	ga	gi	gu	ge	go	gi	gya	gyu	gyo
	が	ぎ	ぐ	げ	ご		ぎや	ぎゆ	ぎよ
s	sa	shi	su	se	so	shi	sha	shu	sho
	さ	し	す	せ	そ		しゃ	しゆ	しよ
z	za	ji	zu	ze	zo	ji	ja	ju	jo
	ざ	じ	ず	ぜ	ぞ		じゃ	じゆ	じよ
t	ta	chi	tsu	te	to	chi	cha	chu	cho
	た	ち	つ	て	と		ちや	ちゆ	ちよ
d	da	ji	zu	de	do				
	だ	ぢ	づ	で	ど				
n	na	ni	nu	ne	no	ni	nya	nyu	nyo
	な	に	ぬ	ね	の	ん	にや	にゆ	によ
h	ha	hi	fu	he	ho	hi	hya	hyu	hyo
	は	ひ	ふ	へ	ほ		ひや	ひゆ	ひよ
b	ba	bi	bu	be	bo	bi	bya	byu	byo
	ば	び	ぶ	べ	ぼ		びや	びゆ	びよ
p	pa	pi	pu	pe	po	pi	pya	pyu	pyo
	ぱ	ぴ	ぷ	ぺ	ぽ		ぴや	ぴゆ	ぴよ
m	ma	mi	mu	me	mo	mi	mya	myu	myo
	ま	み	む	め	も		みや	みゆ	みよ
y	ya		yu		yo				
	や		ゆ		よ				
r	ra	ri	ru	re	ro	ri	rya	ryu	ryo
	ら	り	る	れ	ろ		りや	りゆ	りよ
w	wa				wo				
	わ				を				

Anexo 2. El silabario katakana.

日本語全部片仮名表

	a	i	u	e	o		ya	yu	yo		a	i	u	e	o	
	ア	イ	ウ	エ	オ		ヤ	ユ	ヨ							
k	ka	ki	ku	ke	ko		ki	kya	kyu	kyo						
	カ	キ	ク	ケ	コ			キヤ	キユ	キョ						
g	ga	gi	gu	ge	go		gi	gya	gyu	gyo						
	ガ	ギ	グ	ゲ	ゴ			ギヤ	ギユ	ギョ						
s	sa	shi	su	se	so		shi	sha	shu	sho				she		
	サ	シ	ス	セ	ソ			シャ	シュ	ショ				シェ		
z	za	ji	zu	ze	zo		ji	ja	ju	jo				je		
	ザ	ジ	ズ	ゼ	ゾ			ジャ	ジュ	ジョ				ジェ		
t	ta	chi	tsu	te	to		chi	cha	chu	cho				che		
	タ	チ	ツ	テ	ト			チャ	チュ	チョ				チェ		
d	da	ji	zu	de	do						te		ti			
	ダ	ヂ	ヅ	デ	ド								テイ			
n	na	ni	nu	ne	no	ン	ni	nya	nyu	nyo	de		di			
	ナ	ニ	ヌ	ネ	ノ			ニヤ	ニユ	ニョ			ディ			
h	ha	hi	fu	he	ho		hi	hya	hyu	hyo	fu	fa	fi		fe	fo
	ハ	ヒ	フ	ヘ	ホ			ヒヤ	ヒユ	ヒョ		ファ	フィ		フェ	フォ
b	ba	bi	bu	be	bo		bi	bya	byu	byo	v	va	vi	vu	ve	vo
	バ	ビ	ブ	ベ	ボ			ビヤ	ビユ	ビョ		ヴァ	ヴィ	ヴ	ヴェ	ヴォ
p	pa	pi	pu	pe	po		pi	pya	pyu	pyo						
	パ	ピ	プ	ペ	ポ			ピヤ	ピユ	ピョ						
m	ma	mi	mu	me	mo		mi	mya	myu	myo						
	マ	ミ	ム	メ	モ			ミヤ	ミュ	ミョ						
y	ya		yu		yo											
	ヤ		ユ		ヨ											
r	ra	ri	ru	re	ro		ri	rya	ryu	ryo						
	ラ	リ	ル	レ	ロ			リヤ	リュ	リョ						
w	wa				wo						u		wi		we	
	ワ				ヲ								ウイ		ウエ	

Anexo 3. Los kanji.

Formas originales	Carácter moderno	Significado
	木	árbol
	林	bosque
	森	selva
	本	origen / raíz / libro
	日	sol / día
	月	luna / mes
	明	brillante
	山	montaña
	鳥	pájaro
	島	isla



Este kanji significa "oro" y el gráfico muestra en qué orden y en qué sentido debe trazarse.

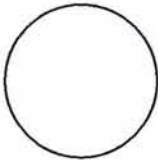
Anexo 4. Notación de Yourdon para diagramas de flujo de datos.

Los diagramas lógicos de flujo de datos se pueden dibujar con sólo cuatro notaciones sencillas, es decir, con símbolos especiales o iconos y anotaciones que los asocian con un sistema específico.

1. **Flujo de datos.** Movimiento de los datos en determinada dirección desde un origen hacia un destino en forma de documentos, cartas, etc. El flujo de datos es un “paquete” de datos.



2. **Procesos.** Personas, procedimientos o dispositivos que utilizan o producen (transforman) datos. No se identifica el componente físico.



3. **Fuente o destino de datos.** Fuentes o destinos externos de datos que pueden ser personas, programas, organizaciones u otras entidades que interactúan con el sistema pero se encuentran fuera de su frontera. Los términos *fuelle* y *pozo* son intercambiables con origen y destino.



4. **Almacenamiento de datos.** Es el lugar donde se guardan los datos o al que hacen referencia los procesos en el sistema. El almacenamiento de datos puede representar dispositivos tanto computarizados como no computarizados.



Anexo 5. Nomenclatura de objetos y entidades empleadas en SAIJA.

En los nombres de identificadores, "V" al inicio significa "variable", mientras que "C" denota "constante". La "L" que le sigue significa "Local" al procedimiento; la "M" es de "Modular" y la "G" es de "Global".

Objeto o entidad	Prefijo	Ejemplo	Variables	Prefijo	Ejemplo
Combo box	cbo	cboSímbolo	String	str	CGstNombre
Command button	cmd	cmdAceptar	Database	db	VGdbSaija
Common dialog	cdl	cdlRuta	Integer	int	CLintF
Cufram ticker	tck	tckAnuncio	Long	lng	VLlngNúmero
Check box	chk	chkBloquear	Single	sng	VLlngTotal
Dir list box	dir	dirCarpeta	Double	dbl	VLlngPorcent
Drive list box	drv	drvUnidad	Currency	cur	CMcurImporte
File list box	fil	filArchivo	Boolean	bln	VLblnBandera
Form	frm	frmConfiguración	Byte	byt	VLbytCantidad
Frame	fra	fraOpciones	Variant	var	VMvarAuxiliar
Horizontal scroll bar	hsb	hsbÁrea	Recordset	rst	VLrstSímbolos
Image	img	imgFoto	Query	qry	VLqryConsulta
Image list	iml	imgDibujos	Date	dte	VLdteFecha
Label	lbl	lblNombre	Object	obj	VLobjObjeto
Line	lin	linLínea			
List box	lst	lstLección			
MDI form	mdf	mdfPrincipal			
Media player	mpy	mpyDiálogo			
Menu	mnu	mnuArchivo			
Module	mod	modSaija			
Option button	opt	optPalabra			
Picture box	pic	picDibujo			
Progress bar	pbr	pbrEstado			
Project	prj	prjProgreso			
Query	qry	qryAltas			
Rich edit text box	rtx	rtxPalabra			
Shape	shp	shpContorno			
Slider	sld	sldVolumen			
Status bar	sbr	sbrEstado			
Tab strip	tbs	tbsOpciones			
Table	tbl	tblUsuario			
Text box	txt	txtNombre			
Timer	tmr	tmrEjercicio			
Tool bar	tbr	tbrHerramientas			
Vertical scroll bar	vsb	vsbÁrea			
VsFlex grid	vgd	vgdMemorama			