



CAMPUS
DE EXCELENCIA
INTERNACIONAL



Graduado en Ingeniería Informática
Universidad Politécnica de Madrid
Facultad de Informática
TRABAJO FIN DE GRADO

Diccionario multilingüe para lenguas africanas

Autor: Javier Extremera Aguado
Director: Susana Muñoz Hernández

MADRID, ENERO DE 2013

Agradecimientos

Agradezco a mis padres Vicente y Concepción, por el esfuerzo que han hecho para que yo pudiera estudiar en esta Universidad, a mi hermano David que también será un futuro ingeniero.


A mi tutora Susana por asignarme el proyecto y a Dave también.

A mis amigos de siempre, Juan, Elena, Martin, Porres, Luis, Blanca y Verónica. Por estar siempre ahí cuando los necesito

A mis amigos hechos en la universidad, Rafa, Pablo, Chema, Nacho, Javi, Bailón, Jesús, Petacas y Gepeto. Compañeros de prácticas y de fatigas, y sobre todo por los buenos ratos y esas cenas que hemos hecho y a nuestro equipo *League of Legends*, *The True Judas Team* en el que estamos todos y que tantas úlceras me ha producido, nuestras sesiones de Raidcall que tantas risas me ha proporcionado, que han hecho que este último año sea el mejor de mi vida. A Hugo aunque no es de la universidad pero es del grupo también.


A mi compañero de piso Alejandro, por soportar mis gritos y mi desorden durante estos años. Y a mí excompañero de piso por irse y dejarme acabar la carrera tranquilamente.

ÍNDICE

RESUMEN.....	III
SUMARIO 	IV
INTRODUCCIÓN	1
MOTIVACIÓN	1
OBJETIVOS	1
ANTECEDENTES.....	1
TÉRMINOS Y TECNOLOGÍAS USADOS.....	3
TEDECO	3
ORIGEN DE TEDECO.....	3
UNL (UNIVERSAL NETWORKING LANGUAGE)	4
ANDROID	4
HISTORIA DE ANDROID	4
ARQUITECTURA DEL SISTEMA OPERATIVO	5
JAVA.....	5
HISTORIA DE JAVA.....	5
LINUX	6
HISTORIA DE LINUX	6
MYSQL.....	7
HISTORIA DE MYSQL.....	7
SQLITE	7
HISTORIA DE SQLITE	8
CARACTERÍSTICAS SQLITE.....	8
SMARTPHONES	8
ESTADO DEL ARTE.....	9
GOOGLE TRANSLATE	9
OROMODICTIONARY	9
TRADUCTOR PARA KIRUNDI.....	10
ENGLISH SWAHILI DICTIONARY FR	10
AMHARICDICTIONARY	11
DESARROLLO	12
REQUISITOS	12

PRIMEROS PASOS	12
MODELO ENTIDAD RELACIÓN DE LA APLICACIÓN	13
ENTIDADES	13
PROTOTIPOS EN PAPEL	14
DISEÑOS FINALES DE LA INTERFAZ.....	16
DIGRAMA DE FLUJO DE LA APLICACIÓN.....	20
DESARROLLO PARTE DE PROGRAMACION	20
HISTORIA DEL DESARROLLO	20
DESARROLLO DE LA APLICACIÓN PARA ANDROID.....	21
FUNCIONAMIENTO DE LAS APLICACIONES EN ANDROID	21
CLASES Y PANTALLAS DE LA APLICACIÓN	23
FUNCIONAMIENTO DE LA APLICACIÓN	24
DATOS DE LA APLICACIÓN.....	25
REPOSITORIO DE DATOS	26
CASOS DE PRUEBA.....	27
RESULTADOS DE LAS PRUEBAS DE USUARIO	27
CONCLUSIONES	28
LÍNEAS FUTURAS	29
BIBLIOGRAFÍA.....	30

RESUMEN

El objetivo de este proyecto es el desarrollo de un diccionario multilingüe para lenguas minoritarias. El proyecto surge de uno de los grupos de cooperación de la UPM, el grupo TEDECO, el cual tiene una línea de investigación sobre cómo las lenguas minoritarias sirven para acercar la tecnología a los colectivos más aislados. Esta línea se inició hace unos años cuando se publicó un artículo en AGIS en el cual se proponía la utilización de una lengua abstracta, UNL  como pivote entre lenguas minoritarias. Esta es una gran oportunidad para crear diccionarios bilingües no ambiguos. En el marco de esta línea de trabajo se encuentra el desarrollo de una aplicación informática para dispositivos móviles.

Actualmente el diccionario multilingüe desarrollado cuenta con unas 4000 palabras UNL. Se trata de un vocabulario básico en UNL donado por el grupo de investigación de validación y aplicaciones de negocio (Validation and Business Applications Research Group). El diccionario dispone actualmente de la traducción de este vocabulario a 5 idiomas: español, inglés, francés, kirundi (de Burundi) y swahili (de Tanzania), y tiene pendientes la próxima incorporación de otros dos lenguajes (amárico y oromo ambos de Etiopia). En el uso de UNL es donde radica la mayor diferencia entre este diccionario y cualquier otro de este tipo que se pueda encontrar en internet. En general muchos diccionarios se suelen basar en el inglés para traducir entre lenguajes, después de todo es el lenguaje más extendido en el mundo digital, pero hay un problema y es que cuando se usa como inter-lingua se pueden perder significados o generar ambigüedad, por eso este diccionario usa UNL que al ser abstracto se ha diseñado específicamente para resolver ese problema. Con UNL conseguimos que desaparezca la ambigüedad, puesto que no es un idioma real, es la representación abstracta de un idioma, para expresarlo mejor, los esquimales tienen muchos términos para nuestro término “hielo”, después de todo el hielo es lo que más tienen a su alrededor, sin embargo el inglés solo tiene un término para describirlo “ice”, si se usa el inglés como lengua base para realizar las traducciones, los diferentes términos pierden su significado y al final solo queda un significado. Con UNL esto no ocurre, porque cada palabra es diferente de las anteriores, de esta manera si buscas hielo del inglés a un idioma con más términos, aparecerán todos los términos para evitar que ocurra ambigüedad y pérdida de significados o información.

El objetivo implícito es que de forma sencilla se puedan incorporar nuevas lenguas a este diccionario. Para ello se ha creado una plantilla en la cual se realiza la traducción de esos nuevos idiomas. A raíz de esto surge otra de las partes del proyecto, para que la aplicación se pueda ir actualizando en este proyecto se incluye una herramienta para administradores que permite introducir nuevos lenguajes a través de la plantilla de manera rápida y fácil.

Por motivos de libertad de desarrollo, libertad de licencias y de publicación se ha desarrollado una aplicación para Android, con licencia GPL, de código abierto y gratuito que permite buscar la traducción de una palabra y ver su significado sin perder información en los diferentes idiomas así como conocer su palabra en UNL.

SUMMARY

The project involves the creation of a multilingual dictionary for minority languages. The project comes from one of the groups cooperation of UPM TEDECO group, which has a line of research on minority languages and how to bring technology to the most isolated. This line was launched a few years ago when **he** published an article in AGIS in which proposed the creation of a mobile application, this application would be a multilingual dictionary.

Currently the dictionary has about 4000 words UNL donated by the research group validation and business applications (Validation and Business Applications Research Group), has five languages, English, Spanish, French, Kirundi (Burundi) and Swahili (Tanzania), and has pending the incorporation of two languages (Amharic and Oromo of Ethiopia both). In using UNL is where the biggest difference between this dictionary and any other of this type can be found on the internet. Generally many dictionaries are usually based on English to translate between languages, after all, is the most widespread language, but there is a problem is that when using a common language meanings may be lost or generate ambiguity, so this dictionary uses UNL. With UNL get ambiguity disappears, since it is not a real language, is an abstract representation of a language, to put it better, the Eskimos have many words for our word "ice", after all the ice is what most have around, but English has only one word to describe it "ice", if used English as the language basis for translations, the different terms lose their meaning and the end there is only one meaning. With UNL does not, because every word is different from previous ones, so you're ice from English into a language with more terms, all terms appear to occur to avoid ambiguity and loss of meaning or information.

Although there are few languages **now** the idea is that users worldwide or institutions cooperate and send their own translations of languages. This has created a template which performs the translation of these new languages. Following this comes another of the parts of the project, so that the application can be updated on this project includes a tool that allows administrators to introduce new languages through that template quickly and easily.

For reasons of freedom of development, freedom of publication licenses and has developed an Android application, GPL, open source and free for searching the translation of a word and see its meaning without losing information in different languages and as knowing his word in UNL.


INTRODUCCIÓN

Esta parte de la memoria sirve para explicar las motivaciones y objetivos del proyecto y lo más importante los antecedentes para que se vea que se lleva mucho tiempo planeándose el desarrollo de esta aplicación y comprobar que es una necesidad real.

MOTIVACIÓN

Este proyecto tiene como motivación la creación de una herramienta de ayuda para dispositivos móviles Android. Esta herramienta debe ser capaz de encontrar las palabras que el usuario busca para que pueda encontrar la traducción a otro idioma. Se trata de ayudar a las lenguas minoritarias que no poseen recursos informáticos en su lengua para que al menos puedan usar las aplicaciones que estén en otros idiomas. Esta herramienta debe ser fácilmente actualizable y su contenido debe ser fácilmente expandido.

OBJETIVOS

El objetivo es  el diccionario tenga una interfaz clara limpia y que no sea confusa y de hacerla lo más **universal** posible. Que no tenga errores y que no asuste a los usuarios porque la puedan ver muy complicada.

Los usuarios deben de ser capaces de seleccionar fácilmente los idiomas en la aplicación y de entender que es lo que les está mostrando el resultado de la búsqueda.

También está la posibilidad de ver la palabra en lenguaje UNL. En esta pantalla se trata de enseñar la palabra en UNL de la manera más simple posible y de que no asuste a las personas que no necesiten esa información.

La aplicación debe ser sencilla de instalar, para ello se instalara desde “Google Play”, la tienda de aplicaciones de Google.

ANTECEDENTES

En el año 2011 tuvo lugar el congreso: AGIS 2011. En dicho congreso participó el grupo TEDECO de la facultad de informática. El grupo TEDECO es un grupo de cooperación para el desarrollo de la UPM. La mayoría de sus integrantes son alumnos o profesores de la Facultad de Informática.

En la ponencia de TEDECO “Automatic Generation of Multiple Bilingual dictionaries for Minority Languages” presentada en AGIS 2011 se propone la creación de un diccionario electrónico que pueda ser usado por usuarios de lenguas minoritarias. La idea es poder acercar la tecnología a los colectivos más aislados.

La idea de que se genere automáticamente es que los propios usuarios puedan aportar nuevos lenguajes o correcciones a los lenguajes existentes o incluso nuevas palabras, siempre que exista en UNL. De esta manera la comunidad de internet podrá cooperar y hacer que los usuarios de lenguas minoritarias puedan tener en su mano una herramienta gratuita y fácil de

usar para poder buscar términos rápidamente, lo cual puede ayudarles a buscar información o acercarse a la tecnología, además de ser un primer paso para la localización de aplicaciones traduciéndolas a lenguas minoritarias.

Las grandes compañías solo tienen en mente los **beneficios** nunca pensarán en traducir un programa a una lengua hablada por ejemplo por 10000 personas, por ello el grupo TEDECO se decidió a crear una herramienta que pudiera ser fácilmente actualizada por los usuarios o por grupos cooperantes. Hay unas 6000 lenguas en el mundo, pero solo se traducen textos y aplicaciones a un máximo de 300, que en general son las que más dinero dan. Lo cual es normal las empresas buscan beneficios, pero se les podrían proporcionar herramientas para que la traducción sea barata y viable.

El uso de UNL se debe a la cultura de las lenguas minoritarias, no puedes crear un diccionario que elimine parte de su lenguaje, por este motivo se decidió que como UNL es un éxito al ser usado en diccionarios multilingües, al usarlo para traducir palabras de lenguas minoritarias no se perdería información.

TÉRMINOS Y TECNOLOGÍAS USADOS

Esta sección es para hablar sobre las distintas tecnologías y los distintos términos que se usan en esta memoria.

- TEDECO (TEcnología para el DEsarrollo y la COoperación)
- UNL (Universal Networking Language)
- Android
- Linux
- Java
- Smartphones
- Base de datos

TEDECO

TEDECO (TEcnología para el DEsarrollo y la COoperación) es un grupo de cooperación al desarrollo de la Universidad Politécnica de Madrid (UPM), compuesto principalmente por profesores y alumnos de la Facultad de Informática de la UPM.

TEDECO es un grupo especializado en las tecnologías de la información y las comunicaciones (las conocidas como TIC), su objetivo es utilizar las nuevas tecnologías para dotar a los centros de países en vías de desarrollo de la infraestructura necesaria para poder desarrollarse y mejorar su situación.

ORIGEN DE TEDECO

En enero de 2006 la Universidad Politécnica de Madrid, recibió un llamamiento urgente de la universidad de Ngozi (en Burundi): “Se necesita URGENTEMENTE la ayuda de profesores cooperantes en una universidad de Burundi”.

A este llamamiento respondieron dos profesores de la UPM que durante los meses de febrero, marzo y abril de 2006 realizaron sendas estancias en la Universidad de Ngozi.

Tras esto decidieron enfocar la ayuda de forma global y no puntual y se creó el grupo TEDECO, como grupo de cooperación de la UPM.

Un año después TEDECO contaba con más de 25 integrantes entre profesores y alumnos de la UPM y colaboradores externos.

UNL (UNIVERSAL NETWORKING LANGUAGE)

Es un lenguaje formal declarativo, diseñado para representar información semántica extraída de textos de lenguaje natural, puede ser usado como lengua pivote para realizar traducciones entre lenguajes.

La idea es que UNL evita la ambigüedad de significados entre palabras. Lo hace aplicándole a cada palabra una serie de atributos. Por ejemplo, “cocinar “ esta varias veces en UNL pero unas veces se refiere a la acción de por ejemplo meter algo en una sartén, otras veces se refiere a la acción de preparar los alimentos, etc.

UNL usa “Universal Words”, que son las palabras en inglés, por ejemplo “cook”, pero cada palabra tiene diferentes atributos que hacen que aunque el mismo término se encuentre repetido varias veces, sus atributos marquen la diferencia de significado. Lo más útil de UNL ha sido cuando se usa en diccionarios, puesto que el proceso de traducción usándolo como lengua pivote todavía no ha dado los resultados deseados.

ANDROID

Android es un sistema operativo ideado inicialmente para teléfonos móviles, al igual que otros muchos, pero tiene algo que lo hace diferente y es que está basado en Linux por lo que su núcleo es libre gratuito y multiplataforma.

Por encima del núcleo se encuentra una variación de la máquina virtual de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (GPS, llamadas, agenda, etc.). El lenguaje en el que se desarrollan las mayorías de las aplicaciones es java, aunque se puede programar en C, evitando la máquina virtual y trabajando directamente contra el teléfono.

Es la sencillez de programar en java y la existencia de programación gratuitas, así como el coste de la licencia para publicar aplicaciones en Android (18€) lo que ha permitido que se desarrollen miles de aplicaciones para Android.

También al ser un sistema operativo gratuito muchos fabricantes se han decantado por Android para lanzar teléfonos con ese sistema operativo.

HISTORIA DE ANDROID

Android fue desarrollado inicialmente por la compañía Android Inc. Era una compañía de software ubicada en Palo Alto, California. Fue comprada en 2005 por Google. En 2007 se lanzó la “Open Handset Alliance”, que agrupaba a la mayoría de fabricantes de teléfonos móviles y Google. Se proporcionó la primera versión de Android, junto con el SDK para que los programadores empezaran a crear sus aplicaciones para este sistema.

Empezó vendiendo muy lentamente pero en el último trimestre de 2010 se situó como el sistema operativo móvil más vendido del mundo.

ARQUITECTURA DEL SISTEMA OPERATIVO

Aplicaciones: las aplicaciones base incluyen cliente de correo, calendario, mapas, navegador de internet, contactos, alarmas...

Marco de trabajo de aplicaciones: los desarrolladores tienen acceso completo a la API del framework con la que se crean las aplicaciones base. La arquitectura está diseñada para permitir la reutilización de componentes fácilmente.

Bibliotecas: Android incluye bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características están disponibles para los desarrolladores a través del marco de trabajo de aplicaciones de Android.

Runtime de Android: Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación ejecuta su propio proceso, con su propia instancia de la máquina virtual.

Núcleo de Android: Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo actúa también como capa de abstracción entre el hardware y el resto de la pila de software

JAVA

Java es un lenguaje de programación y la primera plataforma informática creada por Sun Microsystems en 1995. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como móviles y televisiones.

Las aplicaciones en java se ejecutan sobre una máquina virtual de java (JVM) sin importar la arquitectura del sistema, por eso Java es tan utilizado, no depende ni del sistema operativo ni de la arquitectura. Java es un lenguaje orientado a objetos, de propósito general y concurrente. Fue diseñado para tener tan pocas dependencias de implementación como fuera posible la intención es que el programa se escriba una vez y se ejecute en cualquier dispositivo.

HISTORIA DE JAVA

Se inició su desarrollo en 1991. En teoría su nombre proviene de una lista de nombres al azar, pero hay indicios de que su nombre proviene de una marca de café que bebían en Sun, por eso su logo es una taza con café. También refuerza la teoría que los 4 primeros bytes del número mágico de cada archivo en java sea en hexadecimal "0xCAFEBAE".

Se intentó crear una máquina virtual y un lenguaje con una estructura y una sintaxis similar a C++. En julio de 1994 se reorientó la plataforma hacia la Web, sintieron que internet se convertiría en un medio interactivo.

En 1994 pudo descargarse por primera vez Java 1.0a. Pero hasta el 23 de mayo de 1995 no se hizo pública la existencia de Java y HotJava el navegador Web, también se anunció que

Java estaría soportado por el navegador Web Netscape. En 1996 se publicó la primera versión de Java.

Actualmente Java pertenece a la empresa Oracle que compro Sun Microsistem. Actualmente Java está en su versión 1.7.

LINUX

Es un núcleo libre de sistema operativo basado en Unix, es uno de los principales ejemplos de software libre. Puesto que Android tiene Linux como núcleo veo conveniente comentar que es Linux, así como su historia.

El creador de Linux es Linus Torvalds, en aquel momento era un estudiante finlandés, lo creo en 1991, al principio solo era un terminal, hasta que se dio cuenta de que había creado el núcleo de un sistema operativo. Linux consiguió rápidamente desarrolladores de todo el mundo y multitud de personas aportan soluciones a bugs, o modifican el código fuente para realizar sus propias distribuciones.

HISTORIA DE LINUX

En 1991 en Helsinki Linus Torvalds comenzó un proyecto que al principio era un emulador terminal. Después de un tiempo comprendió que estaba realizando el núcleo de un sistema operativo. Publicó el 25 de agosto de 1991 en la red “usenet” en el “newsgroup” (grupo de noticias): comp.os.minix, que estaba haciendo un sistema operativo por hobby para procesadores 386, pedía sugerencias puesto que estaba basándose en minix para hacerlo y quería saber que les gustaba a los usuarios de esa máquina y que no.

El proyecto rápidamente empezó a crecer y se empezó a usar junto a otro software especialmente con software del proyecto GNU. Richard Stallman creador del movimiento GNU se enfadó bastante porque se usaba el nombre de “Linux” para referirse al núcleo y al software que venía junto al núcleo, lo que solía ser software de GNU. Por eso pidió que se nombrara al proyecto GNU/Linux, también pidió que se llamara “Lignux” pero no tuvo éxito.

La primera versión de Linux tenía 10000 líneas de código, la versión 1.0 tenía 170000 líneas de código.

MYSQL

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

HISTORIA DE MYSQL

MySQL surgió alrededor de la década del 90, Michael Widenis comenzó a usar mSQL para conectar tablas usando sus propias rutinas de bajo nivel (ISAM). Tras unas primeras pruebas, llegó a la conclusión de que mSQL no era lo bastante flexible ni rápido para lo que necesitaba, por lo que tuvo que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, totalmente compatible a mSQL.

El origen del nombre MySQL no se sabe con certeza de donde proviene, por una lado se dice que en sus librerías han llevado el prefijo “my” durante los diez últimos años, por otra parte, la hija de uno de los desarrolladores se llama My. Así que no está claramente definido cuál de estas dos causas han dado lugar al nombre de este conocido gestor de bases de datos.

SQLITE

SQLite es un sistema de gestión de bases de datos relacional. Está contenida en una biblioteca escrita en C. Es un proyecto libre creado por D. Richard Hipp, a diferencia de otros sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. Aquí la biblioteca entera se enlaza con el programa pasando a ser parte del mismo. Esto hace que sea mucho más rápido que usando otros gestores de bases de datos. Puesto que en vez de tener que realizar comunicación entre procesos, lo que hace es llamar a funciones y subrutinas, lo cual es mucho más rápido. El conjunto de la base de datos (tablas, datos, definiciones, índices) es guardado como un solo fichero. Para evitar problemas se bloquea la base de datos en cada transacción. La versión 3 de SQLite permite bases de datos de hasta 2 Terabytes de tamaño.

HISTORIA DE SQLITE

D. Richard Hipp trabajaba desarrollando software para la fuerza naval de los Estados Unidos cuando comenzó a desarrollar SQLite, surgió porque él lo necesitaba. En enero del año 2000 estaba trabajando en un proyecto que conectaba con una base de datos. Todo funcionaba bien pero estaban teniendo problemas cuando el sistema se reiniciaba, cambio de base de datos a PostgreSQL, pero estaban teniendo problemas al administrar la base de datos. Y ahí empezó a surgir la idea de escribir un motor de bases de datos simple que permitiera leer archivos directamente del disco duro. Unos meses más tarde empezó a escribir SQLite pensando que sería útil más adelante.

CARACTERÍSTICAS SQLITE

La biblioteca implementa transacciones de base de datos atómicas, consistencia de base de datos, aislamiento, y durabilidad (ACID), “triggers” y la mayor parte de las consultas complejas.

SQLite usa un sistema de tipos diferente a lo habitual, cuando le das un tipo a una columna, SQLite no te obliga a que todos los elementos de esa columna sean de ese tipo, esto es porque SQLite da los tipos de forma individual a cada elemento de la columna. Esto es que se puede tener una columna de tipo entero y meter como dato una cadena de caracteres. Independientemente de eso SQLite intentara convertirlo al tipo de la columna.

Para esta aplicación he usado SQLite. El porqué está claro Android usa nativamente SQLite y proporciona métodos para poder usarlo fácilmente.

SMARTPHONES

Desde 2007 ha habido un gran crecimiento del número de Smartphone, un Smartphone o teléfono inteligente es un teléfono que tiene potencial para sustituir al ordenador. Antes los Smartphone eran un mercado limitado a la gente de negocios, pero con la llegada IOS (sistema operativo de los móviles de Apple, el iPhone) y de Android eso cambio. Prácticamente ahora todos los teléfonos son Smartphone, pueden navegar por internet y recibir correo, también tienen funciones de GPS.

ESTADO DEL ARTE

En esta parte comentaremos los distintos programas encontrados en internet que realicen un funcionamiento similar a la aplicación que he desarrollado.

GOOGLE TRANSLATE

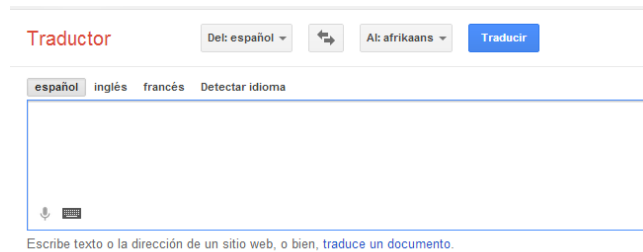


Figura 1 Google Translate

Una interfaz muy sencilla y muy usable, tiene la posibilidad de traducir webs enteras, cada día mejor funcionamiento. El traductor tiene un algoritmo mediante el cual compara millones de textos de fuentes oficiales que estén en varios idiomas y busca patrones, una vez encuentra un patrón lo añade a la base de datos, si vuelve a encontrar el patrón ya sabe cómo traducirlo. Esto es así porque si se intenta hacer una traducción en base a reglas semánticas ocurre que las lenguas tienen excepciones a las reglas e incluso esas excepciones tienen excepciones, lo bueno es que este traductor tiene el swahili, lo malo es que no tiene ninguna de las demás lenguas.

OROMODICTIONARY

Existen pocos diccionarios en la web que sirvan para traducir el lenguaje Oromico. Uno de ellos es este diccionario.

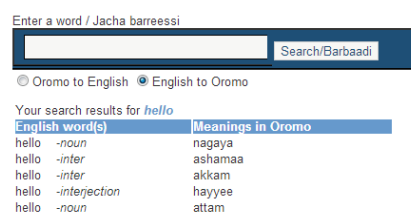


Figura 2 Captura OromoDitctiorany

Es un diccionario independiente creado por hablantes del lenguaje Oromo para ayudar a la gente para que pueda comunicarse en inglés.

TRADUCTOR PARA KIRUNDI

La búsqueda de un traductor de este idioma ha sido decepcionante, la mayoría de los traductores dicen que tienen este lenguaje, pero cuando escribes una palabra no te da ningún resultado. Solo he encontrado este diccionario.



Figura 3 Captura Amajambo

ENGLISH SWAHILI DICTIONARY FR

En mi búsqueda por aplicaciones similares he encontrado una aplicación de Android que es un diccionario de Ingles a Swahili. Este programa tiene una interfaz bastante confusa, los resultados no tienen mucho orden y es muy poco limpia también tiene publicidad. Esta poco optimizada, mi aplicación tiene unas 20000 palabras en total y ocupa 4 MB, esta aplicación tiene 32000 términos y ocupa 26MB solo la base de datos.

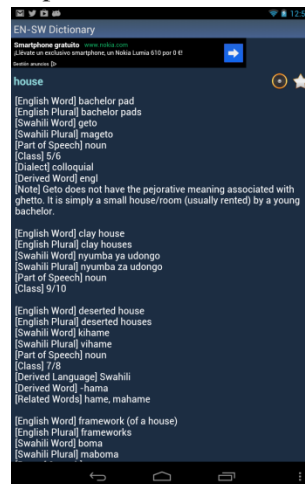


Figura 4 Captura Aplicación para Android

AMHARICDICTIONARY

The screenshot shows the AmharicDictionary.com website. At the top, there is a search bar with the text 'AmharicDictionary.com' and 'አማርኛ-እንግሊዝኛ መዝገበ ቃላት'. Below the search bar, there are two input fields and a 'Search' button. To the right of the search bar, there is a 'Login' link. Below the search bar, there is a table with three columns: 'Word', 'Meaning', and 'Votes'. The table contains three entries for the word 'hello' in Amharic. Each entry has a 'Show Comments' button and a 'Downloads' button. The first entry has 0 votes up and 1 down. The second entry has 4 votes up and 0 down. The third entry has 1 vote up and 0 down. The table also includes links for 'Edit / ማረጋገጫ' and 'Order on CD'.

Word	Meaning	Votes	Downloads
ታዲያስ ~ hello (n.)	/tadiyahas/	0 votes up, 1 down	Download Amharic Audio Basic Version English Audio
ሰላም ~ hello (intj.)	/salam/	4 votes up, 0 down	Download Amharic Audio Basic Version
ጤና ይስጥልኝ ~ hello (plur.)	/T'ena ysT'lny/	1 votes up, 0 down	Download Amharic Audio Basic Version

Figura 5 Amharicdictionary

Buscando diccionarios en internet he encontrado este diccionario de Amárico. Está bastante bien te salen los resultados coincidentes y muestra en inglés y en Amárico la palabra buscar. En la aplicación en vez de un botón, hay un icono de búsqueda que es más intuitivo y no depende del idioma.

DESARROLLO

En este apartado se documenta el desarrollo de la aplicación, desde la especificación de requisitos, los primeros pasos donde se pensaron las ventanas principales, la base de datos, los diferentes prototipos y diseños de la interfaz, el diseño final de cada una de las secciones de la aplicación, y el funcionamiento detallado de cada método de la aplicación. También se documenta el desarrollo de una aplicación para futuros administradores para poder insertar fácilmente nuevos lenguajes en la aplicación.

REQUISITOS

- La aplicación debe ser estable.
- La aplicación debe permitir elegir fácilmente los lenguajes.
- Debe de ser sencillo entender cuál es el lenguaje de origen y cual el lenguaje de destino.
- Debe de haber un botón de ayuda que muestre algo parecido a un manual de usuario.
- Debe de tener un botón de borrar el texto.

METODOLOGÍA

Durante el desarrollo se siguió una metodología ágil, cada dos semanas había una reunión en la que se fijaban unos objetivos a cumplir en las dos semanas siguientes y se corregían los errores que se iban encontrando en cada iteración. Es un buen método de desarrollo puesto que no esperas al final del desarrollo para ver si se está logrando el resultado final esperado.

PRIMEROS PASOS

Al inicio del desarrollo se pensó en un esbozo de cómo debería ser la lógica de la interfaz.

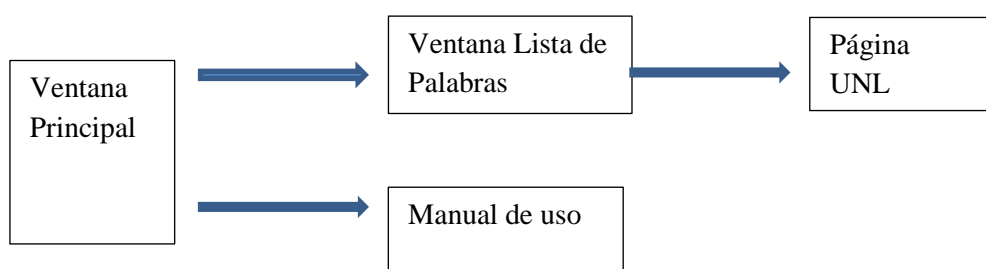


Figura 6 Diagrama de flujo

Basándonos en uno de los traductores más utilizados, vimos que la interfaz de “Google Translate” nos servía bastante bien para nuestro objetivo. Por lo que basándonos en ella elegimos los componentes que debería tener la aplicación.

MODELO ENTIDAD RELACIÓN DE LA APLICACIÓN

Una vez ideado el diagrama de flujo de la aplicación se desarrolló el modelo entidad relación de la aplicación.

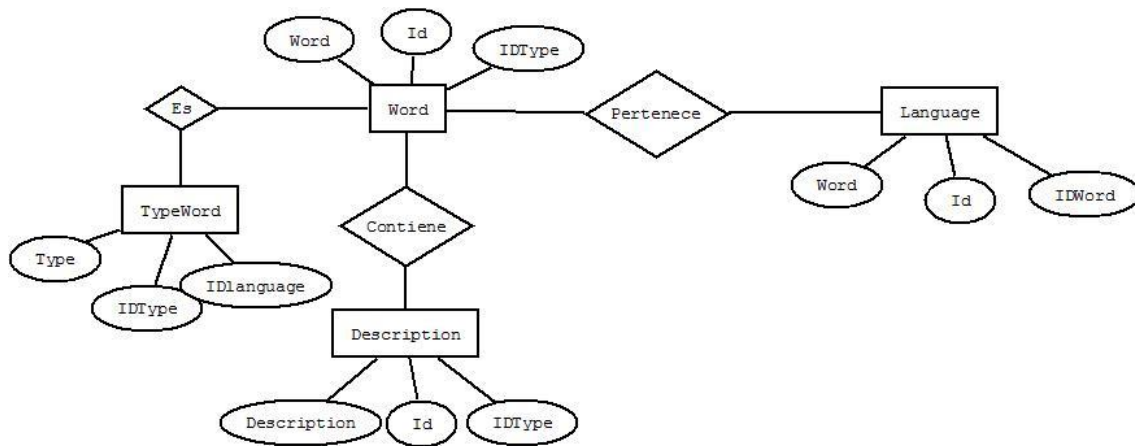


Figura 7 Modelo E/R

ENTIDADES

Word, es la entidad que almacena las palabras, está relacionada con todas las tablas, es el núcleo de la base de datos.

Atributos:

- Word: es la palabra en sí misma.
- Id: es el identificador único de la palabra.
- IDType: es el identificador del tipo de la palabra.
- IDLanguage: identificador del lenguaje al que pertenece la palabra.

Description, es la tabla donde se almacenaran las descripciones de cada palabra.

Atributos

- IDType, indica el tipo de cada descripción (se usa para realizar las búsquedas más sencillamente).
- Description, es la descripción en sí.
- IDLanguage, identificador de la palabra a la que pertenece la descripción.
- Id, es el identificador de la descripción.

TypeWord, en esta tabla se almacenan los tipos de palabras

Atributos

- IDType, identificador del tipo.

- IDLanguage, identificador del lenguaje.
- Type, nombre del tipo (sustantivo, adjetivo, adverbio, nombre).

Language, es la tabla donde se guardaran los diferentes lenguajes.

Atributos

- ID, es el identificador del lenguaje
- Name, es el nombre del lenguaje, será en inglés.

PROTOTIPOS EN PAPEL

Al empezar el proyecto una vez tenía clara como iba a ser la base de datos, realice un prototipo en papel de cómo iba a ser la aplicación. Como desde el primer momento tuve en mente realizar un prototipo para pc, mi primer diseño en papel fue para realizar la aplicación para pc.

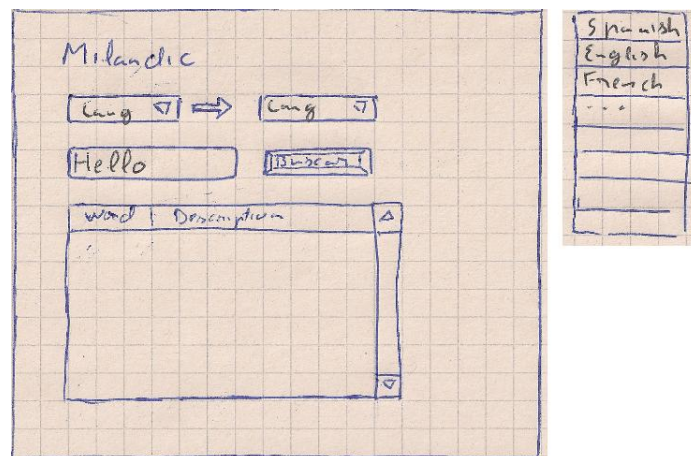


Figura 8 prototipo en papel

Como se ve tiene lo básico, dos desplegables para seleccionar los idiomas, un cuadro de texto para escribir la palabra y un botón para buscar, también tiene un espacio donde aparecerá la búsqueda de la palabra.

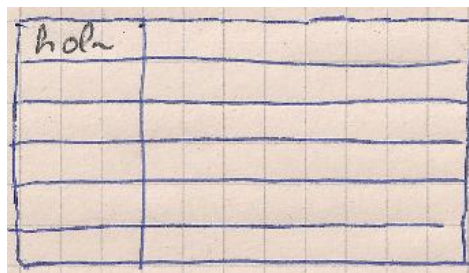


Figura 9 prototipo en papel

El diseño fue en papel porque antes de empezar a codificar es mejor pensar que es lo que se va a codificar.

Una vez tuve el diseño realizado, me dispuse a realizar la aplicación de concepto en Java. Era una aplicación muy sencilla y se realizó relativamente poco tiempo, un par de semanas. Mi primera versión uso MySQL. Mi primera base de datos era mucho más sencilla que la final, la primera solo tenía tres tablas, Idioma, Palabra y Descripción.

En esta etapa se desarrolló una aplicación para poder cargar la base de datos, puesto que las diferentes palabras y lenguajes se encontraban en una hoja de cálculo. Este programa ha ido evolucionando a lo largo del proyecto y se habla de él más adelante.

Una vez enseñada esta versión a mi tutora y habiéndole dado el visto bueno me dispuse a realizar la versión para Android. El desarrollo de la aplicación en Android se explica más adelante, aquí solo se explica el diseño de la interfaz en papel.

Teniendo en cuenta el tipo de aplicación desarrollada, la interfaz de una aplicación de escritorio es muy válida, siempre que se tengan en cuenta las limitaciones de pantalla y el lugar donde se mostraran los resultados.

Al inicio se diseñó este prototipo en papel de como sería la pantalla principal de la aplicación.



Figura 10 prototipo en papel

Como se puede ver página principal mantiene lo esencial, dos desplegables para seleccionar idioma, un indicador del sentido de los idiomas para indicar origen y destino. Un cuadro de texto para escribir y un botón de búsqueda. También pensé en cómo debería ser la lista de palabras encontradas, basándome en la agenda de contactos de Android



Figura 11 prototipo en papel aplicación para Android

Más tarde ya programando se creó la interfaz del diseño pero pensamos que había que añadir un botón de información. En un principio puse el botón de información abajo a la derecha por motivos de usabilidad se modificó, puesto que al iniciarse la aplicación la mayoría de las veces aparece el teclado y nunca se veía el botón por lo se cambió la zona del botón de información y se colocó en la zona de los desplegables. En esa zona es muy visible y no afecta al diseño.

DISEÑOS FINALES DE LA INTERFAZ

Esta es la ventana donde se buscara las palabras, se elegirán los idiomas y se pulsara sobre el botón de información para ver el video de ayuda, este es el diseño definitivo de la ventana principal.

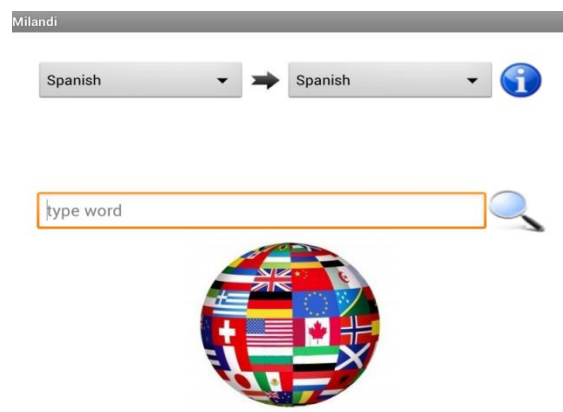


Figura 12 Pantalla de inicio

Primero debería tener dos desplegables para seleccionar los idiomas. Elegimos lo que se llama en algunos lenguajes “combobox” o en Android “Spinner”.

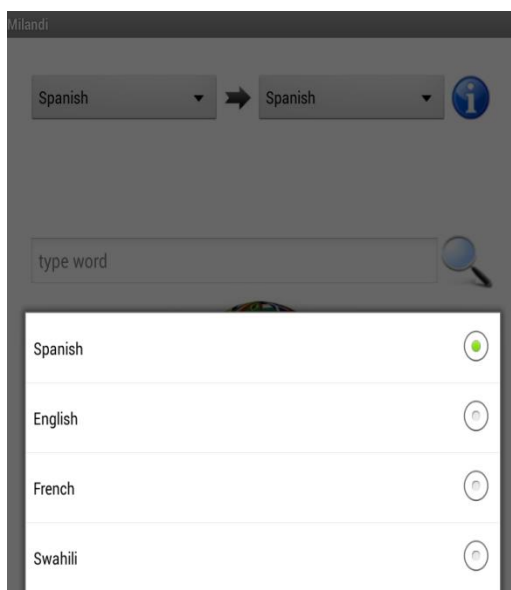


Figura 13 al pulsar el desplegable



Figura 14 Desplegables (Spinners)

También es necesario tener un elemento en el cual escribir la palabra que buscamos. Para ello usamos un cuadro de texto “TextBox” y un botón para poder realizar la búsqueda y un botón con icono de lupa para que al pulsar sobre él manda a otra ventana con los resultados.



Figura 15 Cuadro de Texto

El botón después de pensarlo decidimos ponerle el icono de una lupa en vez de un botón esto es porque con una lupa intentamos que para alguien que use la aplicación sea más sencillo que piense que tiene que pulsar el botón para buscar la palabra en otro idioma.

Para borrar el texto que se introduzca en el cuadro de texto se añadió una funcionalidad y al escribir texto aparece dentro del cuadro un botón para poder borrar el texto.



Figura 16 Botón de Borrar

Adicionalmente se introdujo un botón de información.



Figura 17 Botón de información

También se añadió para darle algo de estética una imagen con las banderas de los diferentes países del mundo.



Figura 18 Imagen de banderas

Al pulsar sobre la lupa para buscar las palabras te mandara a una página nueva donde saldrán las palabras. Se **mostrarán** las palabras que empiecen por las letras que has introducido en el texto. Esto es porque si el usuario no sabe exactamente la palabra es más sencillo buscar un poco de la palabra y que la aplicación se encargue de buscarla por él.

Una vez buscada la palabra aparecerá en pantalla una lista de palabras con una descripción de ejemplo de uso de la palabra. También aparecerá si la palabra es un sustantivo (“noun”), un adjetivo (“adjective”), un adverbio (“adverb”) o un verbo (“verb”).

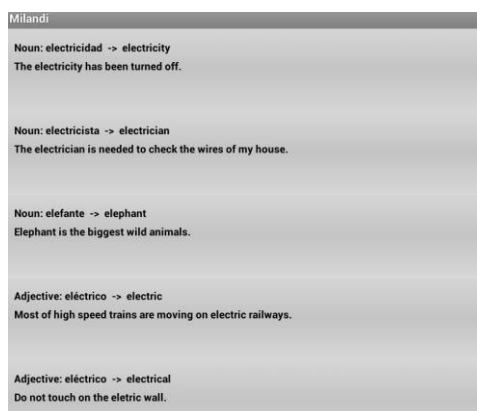


Figura 19 Lista de palabras

Su el usuario selecciona una palabra de la lista aparecerá una nueva página en la que aparecerá el UNL de esa palabra. Aparecerá tanto la “Universal Word” como los atributos de la palabra. También aparecerá una pequeña explicación de que es cada cosa, esta pantalla ha sufrido numerosas modificaciones, en un principio ofrecía mucha información, pero ha sido finalmente eliminada porque en las pruebas que he realizado con usuarios cuando veían esa pantalla pensaban que había habido algún fallo en la aplicación, se decidió ofrecer menos información pero más concisa para que fuera efectivo.

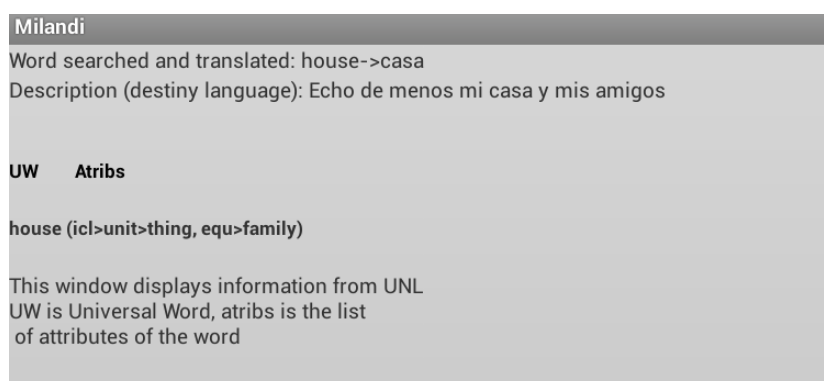


Figura 20 Pantalla UNL simplificada

Si en la página principal el usuario pulsa sobre el botón de información aparecerá una página nueva. En esta página solo habrá un video explicando el funcionamiento de la aplicación. Nos decidimos por un video porque pensamos que para un usuario extranjero sería más sencillo ver un **video** en el que se vea cómo funciona la aplicación, que con texto en el que explique cómo funciona.

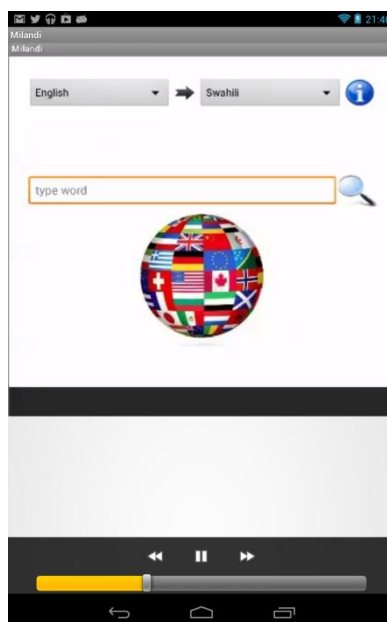


Figura 21 Video de uso

DIGRAMA DE FLUJO DE LA APLICACIÓN



Figura 22 diagrama de flujo

Esta imagen muestra los dos diferentes diagramas de flujo que tiene la aplicación. Uno muestra el video de cómo funciona la aplicación y el otro flujo es al buscar una palabra.

DESARROLLO PARTE DE PROGRAMACION

Desde el primer momento quedo claro que la aplicación debía de ser de libre distribución y de libre modificación y gratuita. Con esas condiciones y viendo que el sistema operativo móvil mayoritario es Android, estuvo claro que la aplicación debía desarrollarse para Android, que es un abanderado del software libre.

HISTORIA DEL DESARROLLO

La aplicación tuvo un inicio duro puesto que algunos idiomas se perdieron y hasta unos meses después solo teníamos los idiomas mayoritarios, español, inglés y francés.

Al inicio del desarrollo se ideó la interfaz y se planifico el funcionamiento de la aplicación y la estructura de datos. El funcionamiento de las búsquedas esta explicado en el apartado de la base de datos. Se crearon tablas extra porque durante el desarrollo pensé que en un futuro cuando el diccionario este más completo se hará que en vez de aparecer los tipos de palabra en inglés como aparecen ahora en el futuro aparecerán en sus respectivos idiomas. Por el momento esa función no está habilitada porque no hemos podido conseguir esas palabras en otros idiomas.

Como era mi primera aplicación para Android, y para entender mejor lo que tenía que hacer decidí hacer una versión de prueba del programa en java. Al principio use MySQL (un motor de base de datos) porque era una base de datos que había usado anteriormente y ya tenía hecho algún programa en java que lo usaba por lo que hacer el primer prototipo fue bastante sencillo.

Consistía en una sola pantalla en la que había un cuadro de texto un botón para buscar y dos menús desplegables (o “combobox”) para seleccionar los diferentes idiomas. El desarrollo de esta pequeña demostración o prueba sí que tenía una parte importante, principalmente por la forma en la que se me suministraron los lenguajes, lo cual explicare en otro apartado. Después de enseñar la aplicación y ver que era exactamente lo que se pedía para el proyecto, ya pude empezar la aplicación para Android.

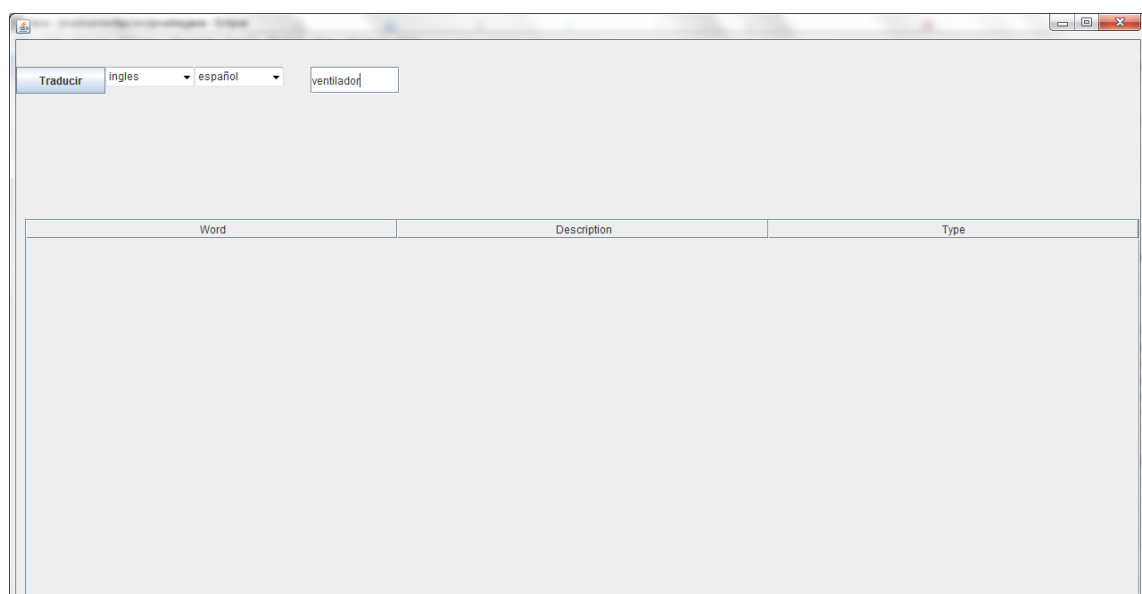


Figura 23 ventana de la aplicación para sistemas de escritorio

DESARROLLO DE LA APLICACIÓN PARA ANDROID

Descripción de cómo se desarrolló la aplicación en la plataforma Android, se detalla la explicación de cómo funcionan las distintas pantallas de Android, como fue el desarrollo de la aplicación, explicación de las clases desarrolladas.

El inicio del desarrollo de la aplicación en Android fue bastante lento al principio. Lo que en otros lenguajes se hace de manera muy sencilla en Android es bastante complicado. Primero explicare cómo funcionan las aplicaciones en Android.

FUNCIONAMIENTO DE LAS APLICACIONES EN ANDROID

Cada pantalla que vemos en Android es un archivo XML, es como una página web, cada una de estas páginas tiene varios elementos que comentare después (puede tener botones, **fotos**, etc.). Cada uno de estos archivos se asocia a una “Activity” (Actividad). Esta actividad es

la que controla el funcionamiento y la lógica de negocio de la aplicación, ahí se programa. El XML no hace nada, la actividad tiene los métodos o las llamadas a métodos que hacen que el programa funcione. También es importante recalcar que los nombres de las paginas no pueden tener caracteres en mayúscula, si no da un error.

En cada página creas una interfaz, por suerte cuenta con un editor visual muy completo que suele facilitar en algunos casos la adición de componentes.

El elemento más importante dentro de cada página son los “Layouts” (capas), son las superficies donde se van a incrustar los componentes. Los “Layouts” más importantes son “Relative Layout” y “Linear Layout”.

“Relative Layout” son capas libres, en teoría puedes insertar los componentes donde quieras por la pantalla, pero no da muy buen resultado, tiende a quedar demasiado descuadrado, la primera versión del programa usaba esta capa, pero tuve que removerla porque no podía conseguir que quedara como yo quería.

“Linear Layout”, en esta capa los elementos se colocan en fila, podemos indicar si será vertical u horizontal. Use este porque era mucho más sencillo colocar los elementos con esta capa. La idea es poner varias de estas capas en horizontal de esa manera en cada una de las capas puedo organizar los elementos como yo quiera. Por ejemplo en la primera capa coloque dos menus desplegables una imagen y el botón de información. En la siguiente capa coloque un botón para borrar el texto, un cuadro de texto y un botón para buscar palabras. Y en la última capa coloque una imagen, es meramente estético.

Pero en Android no es tan sencillo como poner una capa y ya pones los elementos como quieras, además los elementos tienen una propiedad muy importante, el atributo “Peso”, este atributo indica la cantidad de espacio que puede tomar un elemento. Por ejemplo si en la primera capa ponemos a cada elemento como valor 1 en el atributo “Peso”, lo que ocurrirá es que cada elemento tomara un 25% del espacio disponible en la pantalla. Pero como queremos que los desplegables tengan más importancia en la interfaz, los desplegables tendrán valor 2 y el resto 1. Es una manera muy buena de asignar tamaños, porque lo hace de manera relativa a la pantalla del dispositivo.

Una vez hecha la parte visual de una página necesitamos relacionarla con una “Activity” que hará que funcione. Pero antes de eso necesitamos hacer una cosa mucho más importante y que prácticamente en ningún tutorial se da la importancia necesaria. En Android al crear una aplicación se genera un archivo llamado “Manifest.xml”. Este archivo es el que indica que “Activity” existe en la aplicación, es muy importante tener en cuenta este archivo puesto que si no añadimos las nuevas “Activity” que creemos no funcionara. Me ha llevado muchas horas el ver que algo no funcionaba y no saber porque era, y que al final fuera por ese motivo.

CLASES Y PANTALLAS DE LA APLICACIÓN

“MainActivity”

Es la clase principal por donde empieza la aplicación. Esta clase controla la página principal y desde ella se llega al resto de páginas de la aplicación. Tiene asociada la página “activity_main.xml” que tiene la interfaz principal de la aplicación.

“ListController”

Esta clase es la clase que se encarga de generar la lista de palabras e invocar los métodos de gestión de la base de datos para obtener las palabras. Tiene asociada la página “list.xml”, esta página tiene un funcionamiento peculiar puesto que usa elementos individuales para formar la lista, por lo que se ayuda de la página “item_list.xml”, que define cada uno de los elementos de la lista (esto es, las palabras y sus descripciones). Una vez que esta actividad recibe la lista de palabras, rellena la página “list.xml” con varios elementos “item_list.xml”. Esta clase comprueba mediante métodos de la clase “DataBaseHelper” la existencia de la base de datos y le pide los datos necesarios.

También posee un método mediante el cual, cuando seleccionas una palabra de la lista entiende que elemento has seleccionado y activa la actividad “unlhandler” que mostrara la información UNL de la palabra seleccionada.

Cada elemento “item_list.xml” tiene su propio formato en el que usa un degradado, simplemente es estético.

“DataBaseHelper”

Esta clase se encarga de controlar y gestionar la base de datos. Android tiene una forma peculiar de controlar su base de datos. No se puede acceder a la base de datos de manera directa, Android te proporciona herramientas para acceder a ella, y tampoco se puede acceder a la ruta directamente, porque instala la base de datos en la aplicación. El método para conseguir que tu propia base de datos este en la aplicación es haciendo que busque tu base de datos, si no la encuentra que borre lo que había antes y meta la nueva (la que has creado). De esta manera introduce tu base de datos en la aplicación. Por lo cual es muy necesario realizar una clase auxiliar para obtener datos y trabajar con una base de datos.

Esta clase se encarga de comprobar la existencia de la base de datos. Si no existe la copia. También se encarga de obtener las palabras que necesita el diccionario, para ello busca primero las palabras parecidas en el idioma original, una vez tiene todas las palabras, realiza una nueva búsqueda por cada una de ellas y obtiene la traducción de cada palabra (mediante diferentes identificadores).

También posee el método de búsqueda “UNL” por el cual obtiene el “UNL” de cualquier palabra en cualquier idioma.

“WordList”

Esta clase es la que define como es una palabra. En este caso una palabra tiene una descripción un tipo, un identificador, pero dos palabras, esto es porque en una palabra esta la palabra en el lenguaje original y en la otra se guarda la palabra traducida.

“info”

Es la actividad que define el funcionamiento de la página de información, solamente ejecuta el video de explicación de la aplicación. Tiene asociada la página “info.xml” la cual solo tiene un video incrustado.

“unlhandler”

Es la clase que muestra la información UNL de la palabra seleccionada y una explicación simple del significado de cada parte de la palabra. Tiene asociada la página “unlview.xml”.

Para pasar la información necesaria entre las paginas cada actividad tiene un objeto “Bundle” en este objeto se insertan las palabras que quieres enviar entre actividades. De esta forma se consigue que de una actividad a otra se mande la información necesaria para que funcione correctamente y que sea sencillo de realizar. Se explicara su funcionamiento en el apartado siguiente.

FUNCIONAMIENTO DE LA APLICACIÓN

La aplicación empieza en la página “activity_main.xml”. Primero elegimos el idioma en el que está escrita la palabra que buscamos y después elegimos el idioma al que queremos traducir. Después escribimos la palabra o parte de la palabra que queremos traducir y pulsamos sobre la lupa.

Internamente la actividad “MainActivity” recoge los una vez pulsamos sobre la lupa y se activa un evento. Este evento genera un objeto de tipo Bundle. Este objeto es el que usaremos para pasar información de una actividad a otra, de esta manera cada elemento que queremos pasar a la siguiente actividad (la palabra, y los dos lenguajes) se introducen en el Bundle la manera de introducirlo es asignándole a cada dato una clave y desde la otra actividad cuando queremos recuperar el dato lo obtenemos usando esa clave. Después se llama a la actividad que controla la lista de palabras.

“ListController” obtiene los lenguajes y la palabra que se quiere traducir. Realiza las llamadas a la base de datos mediante la clase “DataBaseHelper” explicada anteriormente. Obtiene la lista y las muestra en la página “list.xml”. Si se pincha sobre algún elemento de la lista se activa un evento nuevo que vuelve a usar la clase “DataBaseHelper” y obtiene la información UNL correspondiente a esa palabra. Guarda los datos en un **Bundle** y activa la siguiente actividad, “unlhelper”.

La clase “unlhelper” muestra los datos UNL relacionados con cada palabra. No realiza ninguna función más, esta página es en general solo para los que quieran ver que es UNL no para un usuario normal.

DATOS DE LA APLICACIÓN

Los datos más importantes de la aplicación son sin duda las palabras, necesito palabras en todos los idiomas que sea posible. Se me suministraron unas 4000 palabras por lenguaje divididas en varios tipos: Sustantivos, Adjetivos, Adverbios y Verbos. Se eligieron estas 4000 palabras porque son las palabras que sabe aproximadamente un niño pequeño.

Estas palabras se encuentran en un archivo de Excel que a su vez está dividido en varias hojas, una hoja por cada tipo de palabra. Cada palabra tiene asociada una frase en la que esa palabra aparece, por ejemplo con la palabra “Abanico” aparece la frase “Oculto su rostro tras el abanico”. Puesto que las palabras estaban en un archivo de Excel (u hoja de cálculo) tuve que realizar un programa que insertara esas palabras en una base de datos.

La primera versión del programa trabajaba con “MySQL”, cuando empecé a trabajar en Android lo tuve que cambiar por “SQLite” puesto que Android usa ese motor de base de datos. Tuve algunos pequeños problemas puesto que al insertar ocurrían errores con las comillas simples “” así que tuve que realizar funciones para tratar caracteres extraños.

Este programa ha evolucionado puesto que la idea del proyecto es que en un futuro la aplicación siga creciendo y que los voluntarios que lo deseen puedan seguir mejorando y añadiendo lenguajes a la aplicación. Para facilitar el trabajo al futuro administrador el programa ha evolucionado lo suficiente como para tener interfaz gráfica lo suficientemente sencilla. Al mismo tiempo se le suministrara un manual en el que se explique cómo funciona la aplicación.

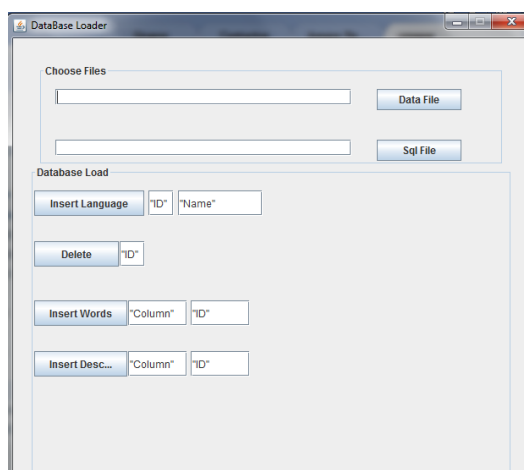


Figura 24 Botón de Borrar

Los botones “Data File” y “Sql File” sirven para seleccionar el archivo con las palabras y el archivo de Sql respectivamente.

“Insert Language” y el campo “ID” y “Name” sirven para crear un nuevo lenguaje en la base de datos. Se le asigna a cada lenguaje un Id. Como es el administrador el que va a insertar el lenguaje debe de saber que Id’s puede poner y que Id’s están en uso.

“Delete” al introducir un Id de algún lenguaje, borrara todo el contenido y el propio lenguaje de la base de datos.

“Insert Words” en el campo “Column” se introduce el número de columna de la hoja de cálculo donde están las palabras del idioma. En el campo “ID” indicas el identificador del lenguaje.

“Insert Descriptions” es exactamente igual que el botón anterior solo que sirve para insertar las descripciones de las palabras del idioma. “Column” indica la columna de la hoja de cálculo e “ID” indica el identificador del lenguaje.

El formato de archivo de la hoja de cálculo es el siguiente

NOUN		
Univ. Word	Word in new language	Description in the new Language
fan		
alphabet		
bee		
lawyer		
attorney		
fertilizer		
subscription		
payment		
hug		
can opener		
bottle opener		

Figura 25 Hoja de cálculo para nuevos lenguajes

Tiene 4 hojas en la primera hoja van los sustantivos, en la segunda los adjetivos, en la tercera los adverbios y en la cuarta los verbos.

El funcionamiento de esta aplicación es sencillo.

Tiene un método para leer todas las filas de palabras dependiendo de la columna que se quiera leer de la hoja de cálculo. Almacena todas las palabras de la columna en una lista y las inserta una por una en la base de datos asignándole sus correspondientes identificadores. Para las descripciones es exactamente igual.

REPOSITORIO DE DATOS

Este es un proyecto de código abierto, por lo que cualquier usuario puede modificar el código fuente y distribuirlo como desee, siempre y cuando siga siendo código abierto y cualquiera pueda seguir modificándolo. Por ese motivo el proyecto tenía que estar en algún sitio en el que los usuarios puedan descargarse el código y estudiarlo o modificarlo.

El mejor servicio que hemos encontrado es el servicio de Google, conocido como Google Code. El servicio también sirve para tener un control de versiones y poder tener un respaldo de los datos.

Es en Google Code donde también se guardara la documentación del proyecto (este documento). Y se mantendrá actualizado, cualquiera podrá descargarlo.

El enlace es el siguiente: <http://code.google.com/p/milandi/>

CASOS DE PRUEBA

En esta sección se detallara los diferentes casos de prueba realizados para probar el correcto funcionamiento de la aplicación.

- Correcta búsqueda de palabras: Al introducir un usuario una palabra que existe, la palabra se encuentra sin ningún problema.
- Correcto funcionamiento al introducir una palabra que no existe: Al introducir el usuario una palabra que no existe se muestra un mensaje informando de ello.
- Botón de borrar texto: Al introducir texto en el cuadro de texto aparece un botón y al pulsarlo se borra el texto.
- Botón Información: Al pulsar sobre la aplicación muestra una página nueva en la que se reproduce automáticamente un video de uso de la aplicación.
- Visualizar UNL: Al pulsar sobre una palabra aparece una página nueva en la que se muestra el UNL de esa palabra.

RESULTADOS DE LAS PRUEBAS DE USUARIO

He probado la aplicación con 10 personas, después de ver sus reacciones he encontrado las siguientes consideraciones y se han realizado algunos cambios en la interfaz.

- Botón información: Al principio al estar en la esquina inferior derecha ningún usuario logro verlo, por lo que se cambió a la esquina superior derecha, que es mucho más accesible.
- Márgenes: Después de recomendaciones de algunos usuarios se han puesto márgenes a la aplicación que hacen que sea más fluida y da una mejor sensación a la vista.
- Botón de buscar: Al principio era un botón con el texto “Translate”. Después de probarlo, se cambió por un icono, puesto que es más universal.
- Botón de borrado: en las primeras versiones a la izquierda del cuadro de texto había un botón rojo para borrar con el icono de “X” en rojo, se ha sustituido y ahora aparece un icono con una “X” en gris dentro del cuadro de texto, que da mejor apariencia y no da sensación de peligro.
- Se ha dotado de colores grisáceos la página de las listas de palabras y de UNL, cambio meramente estético porque quedaba demasiado estéril todo en blanco.
- En la página de información aparece un video porque asi no es necesario mostrar ningún texto y es más universal.

CONCLUSIONES

La idea del proyecto no me convencía al principio, porque pensaba que ya existían aplicaciones que hicieran **realizaran** el mismo trabajo que esto, al principio pensaba que esta aplicación nunca tendría ningún uso real, hasta que vi que una aplicación que realiza exactamente lo mismo que está pero más pesada y con un solo idioma tenía más de 10000 descargas.

He podido ver que en la red al buscar traductores para estos idiomas minoritarios, muchos de los traductores no traducen nada, y no solo traductores, los propios diccionarios electrónicos, algunos no tenían ningún termino, simplemente era para mostrar publicidad.

Cuando vi eso comprendí que esta aplicación de verdad podría tener mucho futuro, hay un público real que demanda este tipo de aplicaciones, y para los que son útiles. Me gustaría que la aplicación tuviera por lo menos 20000 palabras por cada lenguaje sería totalmente increíble y muy útil. Si la aplicación llegara a un público masivo la gente empezaría a colaborar con el proyecto y no estaría mal poder decir he hecho una aplicación que se han descargado más de 10000 personas.

El desarrollo de la aplicación a veces se me ha hecho cuesta arriba. Android tiene algunos requisitos muy difíciles de ver. Por ejemplo para insertar el video que muestra cómo se usa la aplicación, solamente por tener la “m” de milandi en mayúsculas, no me funcionaba el proyecto. Pienso que esas restricciones son un incordio, no tiene mucho sentido que actualmente haya restricciones por caracteres de ese tipo.

Lo bueno es que he aprendido a programar para este entorno, que es algo que siempre había querido pero que por falta de tiempo nunca había podido hacer en serio. Muchas cosas me han gustado, todo es muy gráfico, el desarrollar la parte grafica es muy sencillo, aunque a veces falle.

Gracias a este proyecto he aprendido cómo funciona el traductor de Google, he aprendido la historia de las traducciones automáticas.

He descubierto grupos de trabajo que desconocía de la Facultad de Informática, como el grupo TEDECO, me parece admirable su labor altruista, también he conocido a alumnos que cooperan con ese grupo y se lo toman muy en serio y van a muchas conferencias en las que se tratan temas de cooperación.

En general estoy contento con el resultado final de la aplicación, sobretodo comparando con las primeras versiones que desarrolle de la aplicación que tenían una interfaz muy desordenada y poco cuidada. La versión final ofrece un acabado convincente y simple, lo suficiente para una aplicación de este tipo.

LÍNEAS FUTURAS

De este proyecto se pueden ver fácilmente algunas líneas futuras:

Una vez aprobado el trabajo lo primero será comprar una licencia de publicación en Google Play para poder publicar la aplicación.

Otra línea también evidente es la de añadir más vocabulario a la aplicación. Esta era una primera versión pero se deberían introducir por lo menos unas 20000 palabras por lenguaje. Siempre que estén en UNL. También introducir más idiomas y cambiar la descripción y en vez de descripciones poner las definiciones. Ahora mismo está bien usando frases de ejemplo de uso pero en el futuro sería mejor usar las definiciones.

Otra línea de futuro sería usar el trabajo iniciado en este proyecto y crear una API que pueda traducir instantáneamente los menús de cualquier aplicación. Por ejemplo se podría poner en la aplicación los menús en UNL, que después de todo las Universal Words están escritas en inglés y que la API dependiendo del idioma del sistema operativo traduzca la aplicación.

También se podría intentar realizar un traductor, pero no es muy viable. De hecho Google dice que es imposible realizar actualmente una traducción usando un lenguaje intermedio. Por eso ellos usan estadística para las traducciones. Se supone que los lenguajes tienen reglas para la semántica, el problema es que también tienen contra reglas y a veces las contra reglas tienen contra reglas por lo cual al final es imposible realizar la traducción así. Puede que se pueda intentar usando estadística y reglas, pero sería un proyecto muy grande Google usa miles de traducciones de libros para generar la estadística.

BIBLIOGRAFÍA

Carla Parra Escartín, *Traducciones automáticas*. Enlace: <http://www.lalinternadeltraductor.org/n6/traduccion-automatica.html>

Susana Muñoz Hernández, *artículo donde se propone la creación de un diccionario multilingüe para dispositivos móviles*. Automatic Generation of Multiple Bilingual dictionaries for Minority Languages

Enríquez Toledo Alma, Maldonado Ayala Jesús , Nakamura Ortega Yunko , Nogueron Toledo Goretty, *información MySQL*. Enlace: <http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf>

Wikipedia, *información MySQL*. Enlace: http://es.wikipedia.org/wiki/Base_de_datos

UNLD Foundation, *información sobre UNL*. Enlace: <http://www.undl.org/>

Aitor Santana, *historia de Android*. Enlace: <http://www.elandroidelibre.com/2012/10/la-historia-de-android-ordenada-cronologicamente.html>

Wikipedia, *historia de Android y especificaciones*. Enlace: <http://es.wikipedia.org/wiki/Android>

Alejandro Nieto Gonzalez, *Información sobre Android*. Enlace: <http://www.xatakandroid.com/sistema-operativo/que-es-android>

Alex Chitu, *Como funciona Google Translate*. Enlace: <http://googlesystem.blogspot.com.es/2010/08/how-google-translate-works.html>

Información sobre diversos traductores:

Traductor de Google: www.translate.google.com

Diccionario de Kirundi: <http://amajambo.ijuru.com/>

Diccionario Oromo: <http://oromodictionary.com/>