

UNIVERSIDAD DE SALAMANCA
FACULTAD DE TRADUCCIÓN Y DOCUMENTACIÓN

Trabajo de Fin de Grado

**El trabajo de ingeniera de localización: cómo
localizar un JSON y no morir en el proceso**

Elena Landeras Landeras

Tutor: Emilio Rodríguez Vázquez de Aldana

Salamanca, 2016

Resumen

El presente trabajo intenta explicar la labor del ingeniero de localización, más concretamente el proceso necesario para transformar un archivo de localización de una aplicación móvil en formato JSON en un archivo en formato estándar XLIFF que permita al equipo de traductores desarrollar su labor utilizando su herramienta preferida. Para alcanzar esta meta se ha seleccionado una aplicación móvil y se ha llevado a cabo esta transformación, la traducción del texto, el proceso de transformación inversa y la comprobación de que la aplicación funciona correctamente una vez terminado el proceso. Las conclusiones se centran en la conveniencia del proceso y el uso de diferentes herramientas de localización.

Palabras clave

Ingeniería de localización, localización aplicación móvil, localización videojuego, XLIFF, JSON

Abstract

This paper aims to explain the work of the localization engineer, specifically, the process necessary in order to convert a localization file from JSON to the XLIFF standard. A process necessary to help the translation team develop their work using any tool they prefer without worrying about the filters. To achieve this goal, a mobile app was selected to carry out this conversion, the text translation, the reverse transformation and the functionality check, to make sure the app works correctly once the whole process is over. The conclusions focus on the convenience of the conversion process and the advantages on the use of various localization tools.

Keywords

Localization engineering, localization mobile app, localization video game, XLIFF, JSON

Índice de Contenido

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Justificación.....	2
1.3	Objetivos.....	4
1.4	Metodología.....	5
2	Conceptos básicos.....	6
2.1	Localización (I10n).....	6
2.2	Internacionalización (I18n).....	7
2.3	Globalización.....	7
3	El mundo de los videojuegos.....	9
3.1	Definición.....	9
3.2	Breve historia de los videojuegos.....	9
3.3	Breve clasificación por géneros.....	10
4	Localización de videojuegos.....	12
4.1	Historia de la localización de videojuegos.....	12
4.2	Modelo de localización de videojuegos.....	13
4.3	Tipos de localización según Maxwell Chandler (2012:8-10,51).....	14
4.4	Localización de videojuegos para móvil.....	15
4.4.1	Diferentes Sistemas Operativos (SO).....	17
4.5	Localización de juegos para Android.....	17
5	Formatos procesados en la localización de la aplicación Witchy World.....	19
5.1	El formato JSON.....	19
5.2	El Formato bilingüe XLIFF.....	20
5.3	Formato de intercambio de memoria de traducción: TMX:.....	22
6	Justificación del producto elegido.....	24

7	Proceso de descompilación.....	26
8	Archivo descompilado	28
8.1	Estructura	28
8.2	El JSON de la aplicación	29
9	Propuestas técnicas: llevando el archivo de localización a una TAO	31
9.1	De JSON a XLIFF y vuelta, con una herramienta independiente.....	31
9.1.1	Proceso de conversión de JSON a XLIFF con Rainbow.....	32
9.1.2	Proceso de traducción del XLIFF en una TAO: Trados.....	33
9.1.3	Proceso de reconversión de XLIFF a JSON con Rainbow.....	33
9.2	El archivo JSON directamente a una TAO.....	35
10	Recompilación y prueba del juego traducido	40
11	De los JSON paralelos a TMX	41
11.1	Motivación.....	41
11.2	Alineación de XLIFF con Trados	41
11.3	Alineación de JSON con Rainbow	43
12	Comparación de las herramientas y los procesos propuestos en el Trabajo.....	45
13	Conclusiones.....	47
	Bibliografía.....	49
	Software utilizado.....	54
	Anexo	55

Índice de Tablas

Tabla 1: Géneros de videojuegos.....	11
Tabla 2: Diferencias entre SO móviles.....	17
Tabla 3: Comparativa de las herramientas utilizadas	45

Índice de Figuras

Figura 1: Mercado internacional de videojuegos por plataforma.....	2
Figura 2: Datos mercado de videojuegos	3
Figura 3: Ejemplo del sistema de calificación alemán USK	6
Figura 4: Representación de la relación entre internacionalización, localización y globalización.....	8
Figura 5: Línea cronológica de la historia de los videojuegos	10
Figura 6: Ejemplo de traducción del Zero Wing	12
Figura 7: Ejemplo de archivo JSON.....	19
Figura 8: Proceso de localización antes de XLIFF.....	20
Figura 9: Proceso de localización con XLIFF	21
Figura 10: Ejemplo de estructura de XLIFF simple	22
Figura 11: Mercado de SO móviles en 2015	24
Figura 12: Los 10 juegos más descargados de 2015	25
Figura 13: Estructura del programa APKTool de manipulación de aplicaciones	26
Figura 14: Comandos ejecutados por APKTool en CMD.....	27
Figura 15: Estructura de APK descompilada	28
Figura 16: Ruta de archivos JSON en APK descompilada	29
Figura 17: Ejemplo estructura inicial JSON.....	29
Figura 18: Ejemplo de objeto en el JSON de la aplicación.....	29
Figura 19 Ejemplo de “etiqueta” simple.....	30
Figura 20: Ejemplo de codificación de caracteres no ASCII	30
Figura 21 Creación de filtro para reconocimiento de etiquetas internas	32
Figura 22: Comparación de archive JSON original y XLIFF procesado con etiquetas .	33
Figura 23: Elección de paso para recodificar el JSON.....	34
Figura 24: Elección de opciones para la conversión de la codificación del JSON	34
Figura 25: Archivo JSON añadido a TRANSIFEX.....	35
Figura 26: Ejemplo de etiqueta no reconocida	36
Figura 27: Estructura de la consola de traducción.....	36
Figura 28: Cifras de traducción automática con MT del Proyecto.....	37
Figura 29: Opciones de finalización de la consola TRANSIFEX.....	38
Figura 30: Ejemplo de codificación en el JSON saliente	38
Figura 31: Ejemplo de etiqueta reconocida en Transifex	39

Figura 32: Ejemplo de etiqueta copiada en Transifex	39
Figura 33: Captura de pantalla del juego recompilado.....	40
Figura 34: Opciones de manipulación de la alineación.....	42
Fuente: Elaboración propia.....	42
Figura 35: Ejemplo de alineación incorrecta.....	42
Figura 36: Ejemplo de unidad de traducción en archive TMX exportado de Trados	43
Figura 37: Ejemplo de alineación con filtro sin modificar.....	43
Figura 38: Ejemplo de alineación con filtro modificado.....	44

1 Introducción

1.1 Motivación

El presente trabajo se centra en la exploración del proceso más técnico dentro de la actividad de localización. Este proceso, de preparación de los archivos para facilitar el trabajo del equipo de localización, se corresponde con la actividad desempeñada por el ingeniero de localización (Chandler y O'Malley, 2012:59), una figura que no siempre está presente en los procesos de localización, por lo que resulta necesario para cualquier traductor explorar este rol si su meta es dedicarse a esta actividad, como es mi caso.

Inicialmente tenía dos ideas para desarrollar en mi Trabajo de Fin de Grado, pero tras el sorteo de tutores y dado que no me había correspondido ningún tutor relacionado con la traducción literaria, me decidí por desarrollar mis habilidades en localización, ya que es la actividad profesional a la que quiero dedicarme. Por suerte, mis compañeros y yo hicimos varios cambios y pude conseguir un tutor perfecto para mi idea, Emilio.

Al principio pensé que iba a ser muy complicado llegar a un trabajo con el que estuviésemos cómodos ya que durante estos años y por tener créditos de optativas de sobra, no había cursado la asignatura de localización. Pero como le comenté a mi tutor, siempre me ha gustado investigar y trastear con la informática, por lo que me dio el beneficio de la duda para que completase los conocimientos sobre localización que no poseía.

Como ya he dicho, quiero dedicarme a la localización, preferiblemente de videojuegos ya que desde pequeña me gusta jugar. He tenido consolas de todas las generaciones hasta la fecha y he jugado a juegos de todos los géneros, aunque no me gustan demasiado los de deportes. Precisamente por el gusto que tengo por los videojuegos, he de admitir que soy muy crítica con sus traducciones.

Tengo experiencia en localización de videojuegos gracias a un curso y a mi experiencia laboral, por lo que comprendo los problemas de traducción que aparecen en esta actividad, pero hasta desarrollar este trabajo no entendía bien la cantidad de problemas técnicos que pueden aparecer durante el proceso. En cuanto a su traducción, es necesario reconocer que los videojuegos contienen multitud de tipos textuales diferentes, de temas especializados o literarios, con chistes y hechizos que tienen dificultades traductológicas propias. Pero en todo este proceso es también necesario

poseer conocimientos técnicos relacionados con los formatos en los que nos llegan los textos, las herramientas que podemos utilizar, etc. dado que, sobre todo si nos dedicamos a la localización como freelance, los archivos pueden llegarnos en su formato original sin extraer las cadenas traducibles del código y debemos tener los conocimientos necesarios para reconocer cualquier dificultad y subsanarla.

1.2 Justificación

En la actualidad, los videojuegos son el bien cultural más rentable. Según datos de la Asociación Española de Videojuegos (AEVI), en el año 2014, el consumo en el sector de videojuegos en España llegó a los 996 millones de euros entre software (juegos) y hardware (consolas). En el mercado internacional, las ventas de juegos han superado los 81.000 millones de dólares, 11.000 millones más que el año anterior. De estas cifras, los juegos para móviles y tablets no dejan de crecer en número de usuarios, en 2014 representaban el 27% de las ventas totales de videojuegos.



Figura 1: Mercado internacional de videojuegos por plataforma
 Fuente: <http://www.aevi.org.es/anuario2014/#p=71>

En 2015, los videojuegos en dispositivos móviles y tablets sobrepasaron las ventas y el número de jugadores de las consolas portátiles.

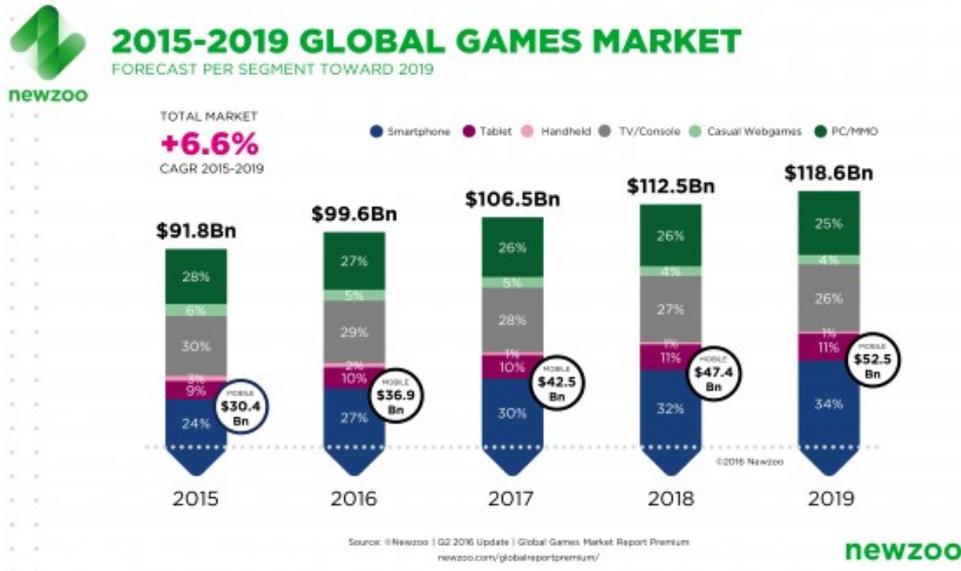


Figura 2: Datos mercado de videojuegos

Fuente: <http://computerhoy.com/noticias/zona-gaming/juegos-moviles-generaran-mas-dinero-que-pc-consolas-43872>

Este aumento del uso de videojuegos ha llevado al sector a depender cada vez más en buenas localizaciones que permitan que el público, sobre todo el público nuevo que espera entender los juegos sin problemas, pueda sumergirse totalmente en la experiencia de juego. Hoy en día, un juego sin localizar o con una mala localización puede suponer un estrepitoso fracaso para su desarrolladora, y en un sector como este con tanta competencia un error así puede ser fatal. Se ha superado el momento en que los usuarios se conformaban con entender lo justo para poder jugar, o en el que los errores graves de localización como en el ejemplo del *Final Fantasy VII*, que aún hoy es famoso por sus problemas de localización, se pasaban por alto en favor de la jugabilidad.

Además de estos datos, como se cita en O'Hagan y Mangiron, se puede distinguir entre la competencia de traducción (centrada en los textos y los idiomas) y la competencia del traductor, que según Chandler (O'Hagan y Mangiron, 2013:252) para los localizadores de videojuegos incluye una larga lista de habilidades necesarias: creatividad, conocimientos culturales, de la industria del videojuego y sus géneros... Pero también subraya la necesidad de que posean conocimientos técnicos en relación a formatos, herramientas especializadas de software y hardware y manipulación de textos con variables. A este respecto, el presente trabajo busca crear una guía del proceso experimental seguido para adquirir algunas de estas habilidades.

1.3 *Objetivos*

El objetivo principal de este trabajo es explorar el rol de ingeniera de localización en el proceso de preparación de los archivos recibidos en un paquete de localización para facilitar el trabajo del equipo de traductores. Para este objetivo se ha decidido utilizar una aplicación móvil, concretamente de Android por ser, como se muestra más adelante en este trabajo, el sistema operativo más sencillo y extendido. En este caso se decidió utilizar un juego cuyos archivos de localización estuviesen en un formato diferente al recomendado por Android (XML). Más concretamente, los objetivos específicos serán:

1. Ofrecer varias opciones de procedimiento para traducir un archivo de localización que recibimos en un formato no estándar. Principalmente explorar el proceso de transformación del formato original de la aplicación seleccionada a un formato estándar (en este contexto XLIFF) que pueda ser procesado con la herramienta de Traducción Asistida por Ordenador (TAO) que el traductor prefiera (o tenga a su alcance), independientemente de los filtros que incluya, mediante un programa auxiliar.
2. Comprobar que, efectivamente, los ficheros XLIFF producidos con dicho software auxiliar se procesan adecuadamente en diferentes TAO. En resumidas cuentas, que puede obtenerse y traducirse el texto en dichas TAO.
3. Realizar, igualmente, el proceso de reconversión o vuelta de XLIFF al formato en el que se encontraban los archivos de localización en origen (en la APK compilada), para asegurarnos de que este paso final puede llevarse a cabo de forma adecuada y sin necesidad de otras herramientas.
4. Explorar el uso de software específico (software de localización en la nube que incluye filtros específicos para este formato) para traducir un archivo JSON.
5. Comprobar que el juego funcione correctamente una vez traducido y recompilado.
6. Crear un archivo TMX mediante la alineación del par de archivos original/traducido (EN/ES) para crear una memoria de traducción que facilite la labor del traductor.

1.4 Metodología

El proceso a seguir para lograr este objetivo principal se desarrollará de esta forma:

- Seleccionar una aplicación móvil que cumpla las condiciones planteadas inicialmente mediante exploración de los archivos internos gracias al proceso de descompilación del archivo APK de la aplicación.
- Explorar el uso de un programa auxiliar de procesamiento y transformación de formatos, Rainbow, para la traducción de los recursos con una herramienta TAO estándar, Trados.
- Explorar la localización de los recursos mediante el uso de una herramienta específica de localización con filtros que reconozcan el formato en el que se encuentran los archivos (JSON) como Transifex.
- Post-procesar los archivos traducidos con Trados mediante la mencionada herramienta auxiliar para comprobar que el proceso es reversible y que el archivo resultante tiene la misma estructura y codificación que el inicial.
- Comprobar el funcionamiento de la aplicación.
- Crear un archivo TMX mediante la alineación de los textos originales (original y traducción) con las herramientas utilizadas para los procesos anteriores¹.

¹ Este proceso podría haberse realizado en primer lugar para facilitar la labor de traducción y reducir el tiempo necesario, pero en este caso no resultaba útil dado que la traducción contenida en los archivos de la aplicación era de muy baja calidad.

2 Conceptos básicos

2.1 Localización (l10n)

El concepto de localización en el entorno informático tiene su raíz etimológica en la palabra localidad, proveniente del inglés *locale* (sitio o lugar o conjunto de información asociada a un lugar o una cultura), que en el caso que nos ocupa, como explican Anastasiou y Schäler (2010), es «the combination of language and culture where the localised digital content, products, and services are used and sold»². Según la definición de Mozilla, la localización «es el proceso de modificar o adaptar un producto de software a los requisitos de un *locale* determinado».

En el contexto de los videojuegos consiste en traducir y adaptar el juego a otros *locales*, prestando atención no sólo al idioma sino a la cultura, las restricciones legales y las particularidades de cada región o país. Por ejemplo, como comenta O'Hagan (2005), el sistema de calificación alemán USK (aquí el equivalente es el PEGI) por el que debe pasar cualquier juego que quiera comercializarse en Alemania (bajo pena de cárcel) incluye restricciones específicas en cuanto a la sangre, los desmembramientos, las vísceras, etc. (los elementos más gore); en cuanto al lenguaje discriminatorio o soez; o los símbolos referentes al racismo, nacionalsocialismo, etc. (hasta el punto de que juegos basados parcialmente en la historia nazi como el Wolfenstein tuvieron que adaptar la versión alemana y cambiar todas las esvásticas por águilas y otros símbolos).



Figura 3: Ejemplo del sistema de calificación alemán USK
Fuente: <http://nintendoenthusiast.com/news/nintendo-eshop-in-germany-to-join-iarc/>

² *Locale* es la combinación de idioma y cultura en la que se utilizan y venden los productos, contenidos y servicios digitales localizados.

La localización implica la traducción de todos los recursos del juego, no se trata únicamente de traducir el texto si no todas las características (texto, menús, imágenes, subtítulos, incluso las voces de los personajes que tendrán que doblarse...) y también los símbolos, monedas, fechas, alfabeto, dirección del texto, atajos de teclado, etc.

La meta final de la localización es que el producto resultante proporcione al usuario una sensación de naturalidad e inmersión; que no sea evidente que el producto se desarrolló originalmente en un *locale* distinto.

2.2 Internacionalización (I18n)

Es el proceso por el cual los productos informáticos se crean facilitando en la medida de lo posible la localización posterior, esto es, que resulte más sencillo adaptarlos a otros *locales*. Este proceso empieza a llevarse a cabo en las fases iniciales del diseño de la aplicación y tiene una importancia capital para triunfo de la distribución global de la misma. De él dependen tanto los plazos como el presupuesto necesario para la localización.

Según Schäler (2003), «internationalisation is the isolation of linguistic and cultural data, so that localisation is performed simply and cheaply»³. Es decir, que en el proceso de desarrollo del software, sitio Web, base de datos o videojuego; el código sea lo más culturalmente neutro posible y que todos los archivos localizables se encuentren aislados del código para hacerlos más fácilmente accesibles y manipulables.

2.3 Globalización

Es el proceso que combina efectivamente la internacionalización y la localización. Según la LISA⁴ “es el proceso del desarrollo de un producto en el que la internacionalización incluye la planificación y preparación, y la localización incluye la adaptación del mismo a un mercado específico.”

³ Traducción: la internacionalización es el aislamiento de los datos lingüísticos y culturales, de modo que la localización se realice fácil y económicamente.

⁴ No es posible acceder al sitio Web de LISA, de donde salían estas referencias en origen así que las he sacado de textos de referencia que a su vez la usaban como fuente propia.

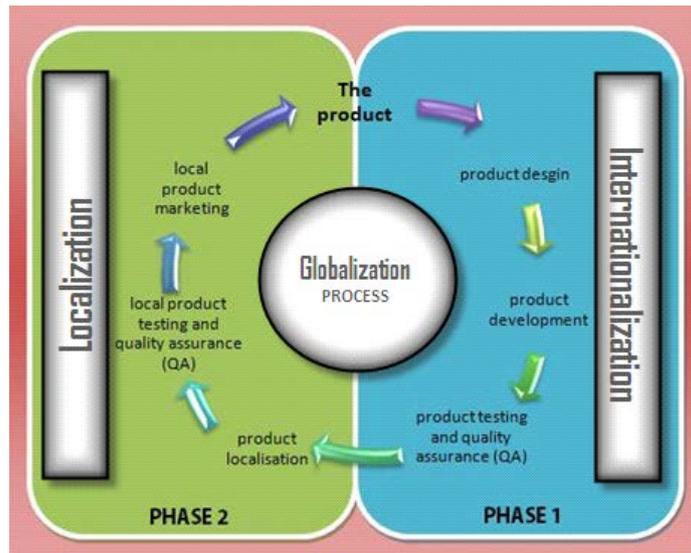


Figura 4: Representación de la relación entre internacionalización, localización y globalización.
 Fuente: <http://blog.belatrixsf.com/globalized-software-a-brief-internationalization-and-localization-overview/>

3 El mundo de los videojuegos

3.1 Definición

En general, se denomina videojuego a cualquier aplicación interactiva orientada al entretenimiento que permite simular experiencias en una pantalla y se controla con mandos. Pueden diferenciarse de otras formas de entretenimiento por su capacidad para involucrar al usuario activamente en el desarrollo del contenido, de la historia.

Según la definición de Frasca, citada por O'Hagan (2005), «any forms of computer-based entertainment software, either textual or image-based, using any electronic platform such as personal computers or consoles and involving one or multiple players in a physical or networked environment»⁵.

Al principio, estos juegos se podían encontrar casi exclusivamente en salones recreativos y se jugaban en una plataforma física diferente constituida por el juego, la pantalla, los controles y los altavoces encajados en un mueble de madera decorado. Estas máquinas se llamaban juegos electrónicos.

Con el avance de la tecnología han aparecido otras variedades de juegos, también en relación con el soporte en el que se reproducen: Juego de ordenador, juego de consola, juego de móvil, etc.

Todos estos tipos de juegos se engloban actualmente bajo el nombre de videojuego, según Bernal (2006), la definición de este concepto es demasiado amplia y habría que variarla: «Video games are an interactive multimedia form of entertainment, powered by computer electronics, controlled by a keyboard, mouse or other peripherals and displayed on some kind of screen »⁶.

3.2 Breve historia de los videojuegos

La historia de los videojuegos se remonta a los años 50, cuando William Higginbotham creó el primer juego interactivo para dos jugadores llamado *Tennis for Two*, que se cree que fue el precursor del *Pong*, aunque no llegó a patentarlo. A

⁵ Cualquier forma de software de entretenimiento, basado en texto o imágenes, que utiliza una plataforma electrónica como un ordenador o una consola y que involucra a uno o más jugadores en un ambiente físico o en red.

⁶ Los videojuegos son una forma interactiva de entretenimiento multimedia, controlada por un teclado, ratón u otros periféricos y visualizada en algún tipo de pantalla.

continuación incluyo una línea cronológica de los hitos más importantes de la historia de los videojuegos:

1962	Se inventa el primer videojuego, <i>Spartan War</i>
1972	-Se comercializa el primer videojuego, Pong -aparece la primera consola doméstica, la <i>Magnavox Odyssey</i>
1980	Se lanza el <i>Pac-Man</i> primer videojuego de éxito mundial
1983	Nace Mario Bros
1984	Aparece la primera consola de Nintendo NES (<i>Nintendo Entertainment System</i>)
1988	Generación de consolas de 16 bits: <i>Mega Drive</i> y SNES
1989	Primera consola portátil, <i>Game Boy</i>
1991	Nace Sonic the Hedgehog
1993	Aparece Doom, primer shooter para PC
1995	Generación de consolas de 32 bits: <i>PlayStation</i> , <i>Saturn</i> y <i>Nintendo 64</i>
2000	Generación de consolas de 128 bits: <i>PS2</i> , <i>Dreamcast</i> y <i>Game Cube</i>
2001	Microsoft entra en la industria de las consolas: Xbox
2002	Sega abandona la producción de consolas y se centra en el desarrollo de juegos
2006	Séptima generación de consolas: <i>PS3</i> , <i>Xbox360</i> y <i>Wii</i>
2011	La aparición de la consola portátil <i>Nintendo 3DS</i> inaugura la octava generación de consolas
2014	Aparecen en el mercado las consolas actuales: <i>PS4</i> y <i>XboxOne</i>

Figura 5: Línea cronológica de la historia de los videojuegos
Fuente: elaboración propia

3.3 Breve clasificación por géneros

Esta tabla está tomada haciendo una comparativa de diferentes fuentes incluyendo autores que tratan el tema (Vela Valido en Reineke, 2005:258-261) (Belli, 2008:13-20) y revistas o páginas web especializadas.

Género	Meristation	Vandal	Hobbyconsolas	3DJuegos	Ultimagame	APK4Fun	Autores
Acción	X	X	X	X	X	X	X
Aventura	X	X	X	X	X	X	X
Aventura gráfica	X	X					X
Deportes	X	X	X	X	X	X	X
Conducción	X	X	X	X	X	X	X

Género	Meristation	Vandal	Hobbyconsolas	3DJuegos	Ultimagame	APK4Fun	Autores
Educativo	X						X
Estrategia	X	X	X	X	X	X	X
Lucha		X	X		X		X
MMO		X	X	X			
Musical		X	X				X
Plataformas	X	X	X				X
Puzle	X	X	X			X	X
Rol	X	X	X	X	X	X	X
Shooter		X	X		X		
Simulación	X	X	X	X			X
Arcade		X	X			X	
Casual				X		X	
Casino						X	
Minijuegos					X		
Habilidad			X				X
Otros		X		X			

Tabla 1: Géneros de videojuegos

Fuentes: www.meristation.com, www.vandal.net, www.hobbyconsolas.com, www.3djuegos.com, www.ultimagame.es, www.apk4fun.com

4 Localización de videojuegos

4.1 Historia de la localización de videojuegos

La localización de videojuegos comienza oficialmente tras el éxito de *Pac-Man* en Japón, cuando los desarrolladores se dieron cuenta de que necesitaban hacer una serie de cambios antes de poder distribuirlo a otros lugares. Los primeros videojuegos carecían casi completamente de texto por lo que la necesidad de la localización no apareció hasta que la industria fue evolucionando.

En los años 80 con la aparición de los juegos como *Mario Bros* se empezó a hacer habitual la localización de manual y caja (Box and Docs). En aquella época la localización del texto interno del juego, si se producía, era únicamente al inglés y no tenía demasiada importancia, como en el ejemplo del juego de 1989 *Zero Wing*.



Figura 6: Ejemplo de traducción del Zero Wing
Fuente: <http://kcng1983.blogspot.com.es/2011/08/all-your-base-are-belong-to-us.html>

En la década de los 90 con el aumento de ventas de videojuegos en Europa se comenzó a desarrollar la localización parcial, es decir, la interfaz de usuario y subtítulos. Con este cambio también se abrió la industria a los usuarios con discapacidad auditiva, ya que ahora podían leer la historia en pantalla.

Para el año 2000, los usuarios empezaron a demandar la localización total de los videojuegos y su lanzamiento simultáneo en todos los países. El modelo de lanzamiento simultáneo ha tenido un gran impacto en la industria de la localización y en el proceso de trabajo de los localizadores.

4.2 Modelo de localización de videojuegos

Este modelo es un híbrido desarrollado en torno a las descripciones del proceso presentes en el trabajo de O'Hagan y Mangiron (2013:128-140), el de Maxwell Chandler (2012:10-13) y el trabajo de Carme Colominas. El proceso de localización de un videojuego tiene varias etapas que pueden dividirse en fases más generales. La primera de estas fases es la planificación, durante esta fase el desarrollador realiza un estudio de mercado, tanto doméstico como internacional para tener la información más amplia sobre la acogida internacional del producto (muy necesario para decidir el tipo de localización que se efectuará en cada *locale*), también tendrá en cuenta las especificaciones del desarrollo del título, las dificultades de localización según esté o no internacionalizado, y el presupuesto del que se dispone. Una vez decididas estas cuestiones, se fijará el calendario del proyecto que estará sujeto a variaciones y retrasos en función del desarrollo del producto.

La segunda gran fase es la fase de producción, en ella el desarrollador entregará al equipo de localización o al ingeniero de localización los archivos necesarios en el paquete de localización y empezará efectivamente la traducción de los recursos internos del juego de acuerdo con el calendario estipulado. Si los archivos contenidos en el paquete de localización deben cambiarse de formato, manipularse o si deben crearse memorias de traducción para facilitar la labor del equipo de traductores, este es el momento en el que el ingeniero de localización entra en el proceso. Según Esselink (2000:16) «Localization engineers are responsible for all technical aspects of localization projects, including Project preparation, software and online help engineering, compiling and testing»⁷. Este mismo autor (2008), considera que la tarea más importante del ingeniero de localización es la preparación de los archivos contenidos en el paquete de traducción para facilitar el trabajo del traductor: localizar y reconocer los archivos traducibles dentro del paquete de traducción y prepararlos para su traducción, a menudo en una herramienta TAO. Una vez traducidos, los recursos se integrarán en el código, es decir, se compilará el juego localizado y se enviará para la etapa de control de calidad.

⁷ Los ingenieros de localización son responsables de todos los aspectos técnicos de los proyectos de localización, incluidos la preparación del proyecto, la ingeniería de software y de la ayuda online, la compilación y el testeo.

Esta es la tercera fase en la que se realizarán pruebas y verificaciones tanto funcionales como lingüísticas por parte de testers nativos de cada idioma y, una vez comprobado el funcionamiento de las versiones localizadas, se pasará a la última fase.

La postproducción es la última de estas fases, en ella se traducen los recursos externos (página web, manual, caja, etc.) y se procede a la fabricación y distribución del título ya terminado. También corresponden a esta última fase las campañas de publicidad y los contenidos adicionales y ediciones especiales.

4.3 Tipos de localización según Maxwell Chandler (2012:8-10,51)

Esta clasificación de los niveles de localización es la más extendida, siendo citada en O'Hagan y Mangiron (2013:141-142). De acuerdo con el presupuesto del juego, la recepción internacional que se espera para él, los recursos de que se dispone (equipos de localización o necesidad de externalizarla, etc.) y el calendario de lanzamiento decidido por la empresa desarrolladora, los videojuegos pueden localizarse de distintos tipos:

- Juegos no localizados: es el tipo de localización más económica ya que no supone gasto alguno, sin embargo, resulta poco rentable porque reduce el mercado internacional del juego casi a cero. Este tipo se reserva casi exclusivamente a juegos de bajo perfil y presupuesto de los que no se esperan grandes ventas. Consiste en enviar un número reducido de copias del juego con su embalaje y su contenido original, sin localizar ningún recurso.
- Juegos con localización de manual y caja (box and docs): Este tipo de localización es relativamente común, principalmente en algunos géneros en los que la mecánica del juego es clara y en juegos de los que se espera vender un número relativamente bajo de copias en el mercado internacional. Consiste en traducir únicamente el embalaje y el manual que lo acompaña, dejando los demás recursos en su versión original.
- Juego parcialmente localizado: Este es el tipo de localización más extendido, resulta relativamente rentable en términos de coste, ya que no es necesario contratar actores de doblaje, etc. y aumenta de forma drástica la previsión de ventas de las copias localizadas. En este modelo se traduce todo el texto interno del juego, incluidos los subtítulos del audio, con lo que la experiencia del jugador, si bien no es la del usuario del idioma original, es mucho más inmersiva. El

jugador dispone aquí de todo el texto de juego traducido a su idioma, incluida la interfaz de usuario, los menús, etc.

- Juego completamente localizado: este último tipo de localización es el que proporciona al jugador una experiencia de juego equivalente a la del usuario original. En este modelo se localizan todos los recursos del juego, incluyendo el doblaje y las imágenes. Es el modelo más costoso por la necesidad de contratar actores, técnicos, etc. para poder integrar el nuevo audio en el producto. Se reserva únicamente para juego con gran proyección internacional, grandes producciones y series ya consolidadas.

4.4 Localización de videojuegos para móvil

Para empezar a hablar de las particularidades de la localización de videojuegos para móviles debemos analizar las principales características específicas de los dispositivos móviles, desarrolladas en base a Maxwell Chandler (2012:138-142). Para el éxito de la programación de aplicaciones, el desarrollador debe tener en cuenta estos obstáculos ya que deberá añadir cambios y opciones para que la aplicación pueda funcionar en el mayor número de dispositivos diferentes.

Primero, debemos atender al gran número de marcas y modelos diferentes de dispositivos móviles, cada uno con sus especificaciones técnicas particulares (memoria, motor gráfico, sistema operativo, etc.). No sólo el interior de los dispositivos es diferente, cada modelo tiene su propia configuración de teclas (aun siendo táctiles en la actualidad pueden cambiar los botones de lugar) y controles. Las dimensiones de la pantalla de los móviles pueden variar en gran medida, por eso el desarrollador debe añadir diferentes configuraciones que varíen el tamaño de fuente (o la posibilidad de alterarlo para adecuarlo a otros dispositivos). Incluso con una misma aplicación, las sucesivas actualizaciones pueden provocar fallos en su funcionamiento con un dispositivo en el que funcionaban anteriormente, ya sea debido a las necesidades técnicas de la nueva versión o al sistema operativo (o la versión de éste) para la que está programada. Esto ha causado recientemente muchos problemas a los usuarios de un popular juego móvil (*Clash of Clans*) produciendo varios días de pérdidas para los ingresos del juego y un gran descenso de su popularidad.

En cuanto a las características particulares de la localización de aplicaciones para móviles, Torres (2007) cita algunas en su artículo. La diferencia más importante es que

en este caso la longitud del texto está más restringida debido a las dimensiones reducidas de la pantalla. Para que la localización sea más sencilla, conviene que la restricción se efectúe por número de caracteres y no por palabras. Debido a la falta de un ingeniero de localización que cree el paquete de traducción, el localizador suele encontrar más dificultades técnicas. Las aplicaciones contienen gran cantidad de cadenas de texto aislado y desprovisto de contexto, por lo que el localizador necesita en ocasiones elementos de apoyo como explicaciones o imágenes que no siempre recibe, ya que los desarrolladores no suelen ser conscientes de las dificultades del proceso de localización. El texto se suele recibir en formato Excel que, si bien es una herramienta útil para ordenar la información, no contiene información textual ni recuento de caracteres lo que la hace menos adecuada para la localización, en muchas ocasiones los textos se convierten a formatos de texto (TXT, DOC, DOCX, etc.) para solventar estas dificultades.

Al contrario que en otros videojuegos, en los juegos de móvil no existen documentos externos (embalaje, guías, manuales, etc.) si no que se traduce únicamente el texto interno, la descripción que lo acompaña en la tienda de aplicaciones y en raras ocasiones una página web. Por esta razón el proceso de localización tiene unos plazos mucho más cortos y necesita mucha más flexibilidad para adaptarse a los ciclos de desarrollo y actualización que son mucho más rápidos.

4.4.1 Diferentes Sistemas Operativos (SO)

Existe un gran número de SO móviles diferentes, que van desde los tres más utilizados, Android 84.7%, IOS 13.1% y Windows Phone 1.7%⁸; hasta sistemas operativos casi obsoletos como Symbian o BlackBerry. También existen SO creados en base a otros como el MIUI, que es una variación de Android con características de IOS, desarrollado por Xiaomi.

	ANDROID	IOS	WINDOWS PHONE
Tipo	Sistema Abierto*	Sistema Cerrado	Sistema abierto
Desarrollador	Open Handset Alliance	Apple	Microsoft
Tienda y aplicaciones	Google Play Store (susceptible a malware) Gran variedad de apps (gratis/de pago)	App Store (muy segura) Gran variedad de apps (de pago)	Microsoft Store (segura) Poca variedad de apps (de pago/gratis)
Dispositivos	Gran variedad de dispositivos	Sólo disponible en dispositivos Apple	Variedad de dispositivos casi nula
Personalización	Muy personalizable	Poco personalizable	Muy personalizable
Rendimiento	Variable en distintos dispositivos	Muy alto	Alto en gama media, medio en gamas superiores

Tabla 2: Diferencias entre SO móviles
Fuente: elaboración propia

*Un sistema abierto es aquel que está disponible para ser utilizado por cualquier marca.

4.5 Localización de juegos para Android

Como ya se ha dicho, Android es el sistema operativo móvil más extendido del planeta, tanto en número de usuarios como en variedad de marcas y terminales. En este apartado, nos ocuparemos con más detenimiento de la estructura de este sistema operativo que se compone de:

- Aplicaciones: componentes básicos encargados de las utilidades (email, navegador, etc.) que pueden estar integrados de origen o descargados por el usuario para cubrir sus necesidades particulares.
- Framework de aplicaciones: facilita el uso de todos los recursos del dispositivo en la misma medida por parte de todas las aplicaciones.
- Runtime: que incluye las bibliotecas de lenguaje Java y la máquina virtual Dalvik que ejecuta los procesos de las aplicaciones en una sola instancia para cada uno de ellos.

⁸ Según datos de Gartner para el último cuatrimestre de 2015. Fuente: <http://www.gartner.com/newsroom/id/3169417>

- Núcleo o kernel de Linux: base del sistema operativo.

Al ser un sistema operativo de código abierto basado en Linux, facilita la creación de aplicaciones y la manipulación de su código para añadir mejoras o arreglar bugs y fallos ya que cualquiera puede hacerlo sin trabas legales ni necesidad de permisos.

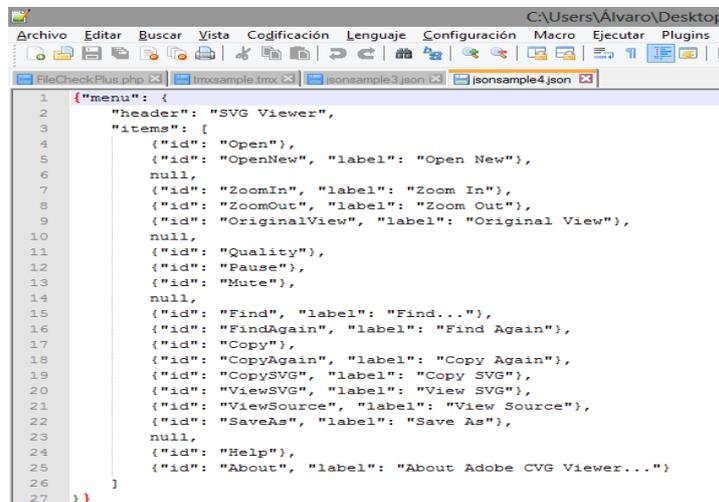
5 Formatos procesados en la localización de la aplicación Witchy World

5.1 El formato JSON

Según su página web⁹, el formato JSON (*JavaScript Object Notation*¹⁰), es un formato de intercambio de datos basado en JavaScript, es un formato de texto independiente del lenguaje JavaScript, aunque utiliza muchas de sus convenciones. Fue desarrollado inicialmente como formato de intercambio de datos en Internet, y es muy popular en la actualidad, gracias a Yahoo! y más tarde a Google, debido a su sencillez.

Un archivo JSON se compone de dos tipos de estructuras:

- Un objeto, que es una colección de pares de nombre/valor. En JSON, todos los objetos comienzan con una llave de apertura `{` y termina con una llave de cierre `}`. Cada par de nombre/valor se separa de los demás por una coma `,` y dentro de los pares el nombre se separa del valor por dos puntos `:`. Una característica que diferencia JSON de JavaScript es que tanto el nombre como el valor deben ir siempre dentro de comillas `"`.



```
1 {"menu": {
2   "header": "SVG Viewer",
3   "items": [
4     {"id": "Open"},
5     {"id": "OpenNew", "label": "Open New"},
6     null,
7     {"id": "ZoomIn", "label": "Zoom In"},
8     {"id": "ZoomOut", "label": "Zoom Out"},
9     {"id": "OriginalView", "label": "Original View"},
10    null,
11    {"id": "Quality"},
12    {"id": "Pause"},
13    {"id": "Mute"},
14    null,
15    {"id": "Find", "label": "Find..."},
16    {"id": "FindAgain", "label": "Find Again"},
17    {"id": "Copy"},
18    {"id": "CopyAgain", "label": "Copy Again"},
19    {"id": "CopySVG", "label": "Copy SVG"},
20    {"id": "ViewSVG", "label": "View SVG"},
21    {"id": "ViewSource", "label": "View Source"},
22    {"id": "SaveAs", "label": "Save As"},
23    null,
24    {"id": "Help"},
25    {"id": "About", "label": "About Adobe CVG Viewer..."}
26  ]
27 }
```

Figura 7: Ejemplo de archivo JSON

Fuente: <https://alvaromira.wordpress.com/2013/10/14/preparacion-de-nuevos-tipos-de-archivos-con-herramientas-cat-con-json-de-ejemplo/>

- Unos *arrays*, o arreglos, que es una lista ordenada de valores, que se colocarán entre corchetes `[]` y en la que cada valor estará separado de los demás por comas.

⁹ <http://www.json.org/json-es.html>

¹⁰ Notación de Objetos de JavaScript.

Un valor puede ser una cadena de caracteres con comillas, un número, true, false, null, un objeto u otro arreglo, por lo que las estructuras pueden anidarse.

5.2 El Formato bilingüe XLIFF

El formato XLIFF (*XML Localisation Interchange File Format*), es, como explica Raya (2004) un tipo especial de archivo XML ideado para ser utilizado durante el proceso de la traducción y localización de documentos. La primera versión fue aceptada por OASIS en 2002 aunque en la actualidad se utilizan la versión 1.2 (2008) y la versión 2.0 (2014) que no son compatibles y que analizaremos más adelante.

El formato XLIFF se creó como respuesta al problema generado por la multitud de formatos diferentes de traducción y localización. Antes, los archivos de traducción y localización podían llegar al traductor en un gran número de formatos diferentes, que hacían necesario crear o disponer de filtros específicos para cada uno o de las herramientas que los incluyeran para poder desarrollar adecuadamente la labor de traducción. La creación de estos filtros suponía un paso extra en el proceso que alargaba los plazos y generaba errores adicionales.

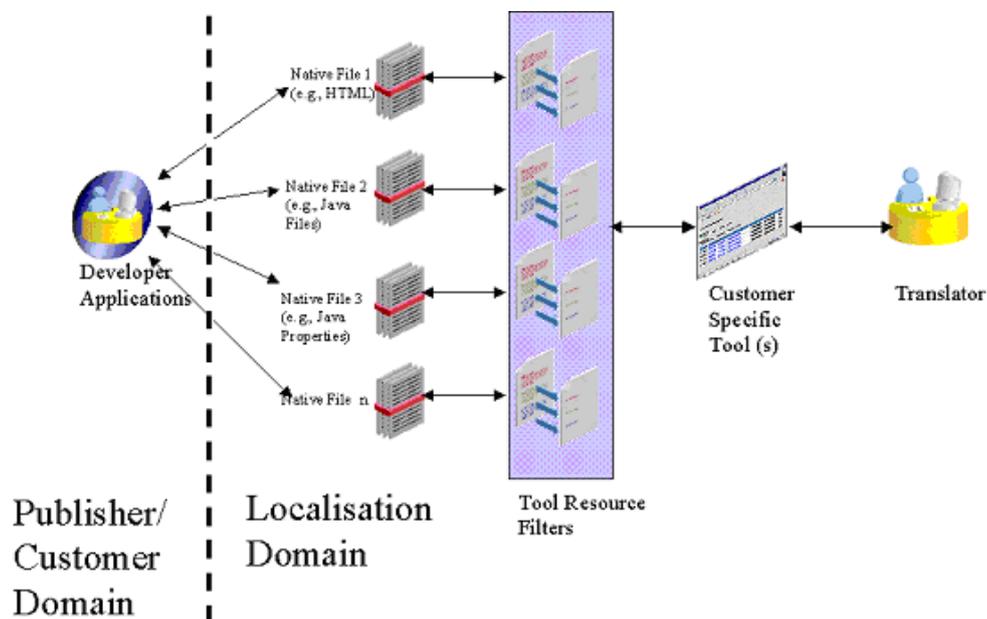


Figura 8: Proceso de localización antes de XLIFF
Fuente: <https://www.oasis-open.org/committees/xliff/faq.php#WhatIsXLIFF>

XLIFF permite mejorar en gran medida este proceso, ya que los diferentes formatos pueden transformarse directamente en un archivo XLIFF que contenga todos los recursos traducibles. De este modo, las herramientas de traducción y/o localización

únicamente necesitan “comprender” (trabajar correctamente) este formato bilingüe, lo que facilita y acorta el proceso y, además, reduce drásticamente el número de errores generados.

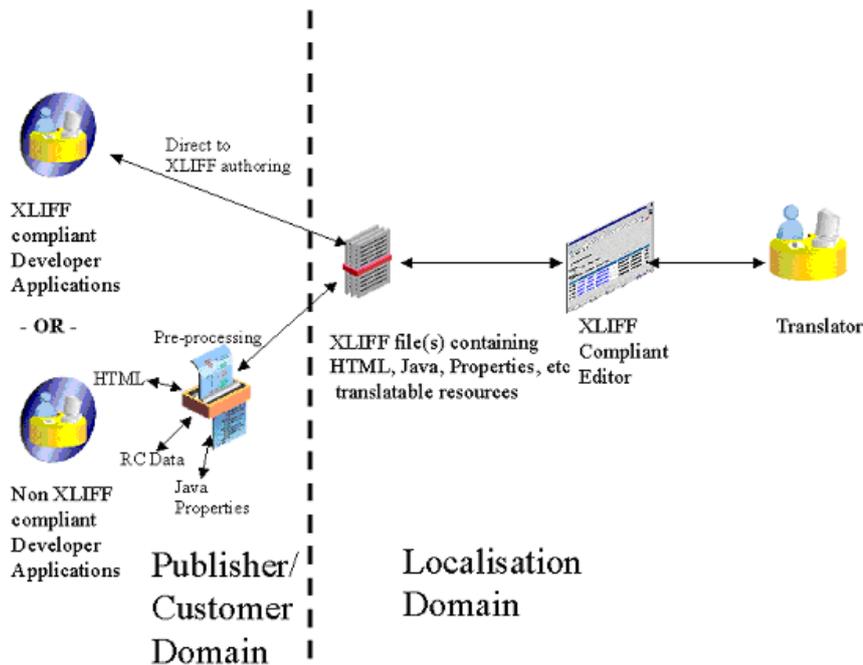


Figura 9: Proceso de localización con XLIFF
 Fuente: <https://www.oasis-open.org/committees/xliff/faq.php#WhatIsXLIFF>

Un archivo XLIFF es una serie de unidades de traducción que contienen una frase o párrafo llamado segmento extraído del documento original con la etiqueta <source> y su traducción correspondiente con la etiqueta <target>, aunque es posible añadir otras traducciones alternativas o utilizar las que ya existieran en la memoria de traducción, éstas llevarían la etiqueta <alt-trans>, generalmente el proceso de añadido lo hacen las herramientas TAO, ya que utilizan este formato.

Al ser un tipo de archivo XML, debe mantener la estructura de éste, es decir, comenzará siempre con la declaración XML, tras la cual comienza el documento XLIFF con la etiqueta <xliff>, la versión utilizada, los idiomas de origen y destino y el elemento <file> dentro del cual se encuentran las unidades de traducción con el segmento original y la traducción o traducciones.

```

- <xliff version="1.1" > Declaración
- <file original="/libs/cq/i18n/ja" source-language="en" target-language="ja" datatype="x-jav > Etiqueta archivo
- <header>
  <tool tool-id="com.day.cq.cq-i18n" tool-name="Adobe Granite I18N Module" tool-vers
</header>
- <body>
  - <trans-unit id="0" > Id. de unidad de traducción
    <source xml:lang="en">fd_Name</source> Segmento original
    <target xml:lang="ja">Name</target> Segmento traducido
  </trans-unit> Fin etiqueta unidad de traducción
  - <trans-unit id="1">
    <source xml:lang="en">fd_<p>Enter your name as First Middle Last</p> </source>
    <target xml:lang="ja"><p>Enter your name as First Middle Last</p> </target>
  </trans-unit>
  - <trans-unit id="2">
    <source xml:lang="en">fd_Reduction of household income</source>
    <target xml:lang="ja">Reduction of household income</target>
  </trans-unit>
  - <trans-unit id="3">
    <source xml:lang="en">fd_Please enter your name.</source>
    <target xml:lang="ja">Please enter your name.</target>
  </trans-unit>

```

Figura 10: Ejemplo de estructura de XLIFF simple
Fuente: Elaboración propia

En cuanto a las diferencias entre XLIFF 1.2 y 2.0, ya hemos comentado que no son compatibles, como explica Savourel (2014), debido a que 2.0 tiene una estructura nueva, en la que se puede crear un esqueleto para cada grupo que contenga un segmento que sirva para recuperar los datos necesarios para reconstruir el documento.

El XLIFF 2.0 tiene una parte central (*Core*) que debe estar presente en todos los documentos y unas partes opcionales o modulares que son especializadas y que proporcionan funciones avanzadas en función de su necesidad. El *Core* representa los elementos y atributos mínimos necesarios para poder extraer el texto traducible, traducirlo y generar la versión traducida del documento original.

La segmentación del texto también es diferente, en estos documentos, la unidad de traducción se segmenta después por lo que cada parte contiene el texto original y la traducción.

5.3 Formato de intercambio de memoria de traducción: TMX:

TMX (*Translation Memory eXchange*) es un formato ideado para el intercambio de las memorias de traducción. Según OSCAR es «a common standard to allow users to

reuse text more efficiently when working with different CAT tools or translation providers»¹¹.

De acuerdo con su página web, el propósito del TMX es facilitar el intercambio de datos de memorias de traducción entre las herramientas y/o los proveedores de traducción con poca o ninguna pérdida de información en el proceso. La versión actual de este formato es la 1.4b (OSCAR, 2005).

¹¹ El estándar común que permite a los usuarios reutilizar el texto de forma más eficiente al trabajar con distintas herramientas TAO o con proveedores de traducción.

6 Justificación del producto elegido

Como ya se dijo en la justificación de este trabajo, el uso de los dispositivos móviles como plataformas de juego está en auge y, como demuestra la tabla siguiente, Android es el SO móvil más extendido del mundo. Por estas razones y como ya se mencionó anteriormente, se decidió realizar el proceso experimental con una aplicación de este SO, concretamente con un juego.

Worldwide Smartphone Sales to End Users by Operating System in 4Q15 (Thousands of Units)

Operating System	4Q15 Units	4Q15 Market Share (%)	4Q14 Units	4Q14 Market Share (%)
Android	325,394.4	80.7	279,057.5	76.0
iOS	71,525.9	17.7	74,831.7	20.4
Windows	4,395.0	1.1	10,424.5	2.8
Blackberry	906.9	0.2	1,733.9	0.5
Others	887.3	0.2	1,286.9	0.4
Total	403,109.4	100.0	367,334.4	100.0

Source: Gartner (February 2016)

Figura 11: Mercado de SO móviles en 2015

Fuente: <http://www.smartphonemarketresearch.com/category/smartphones/>

Para decidir cuál iba a ser la aplicación elegida, propusimos una serie de características que debía cumplir: que el juego en cuestión no tuviese que conectarse a internet para poder utilizarlo en el dispositivo; que los archivos de localización estuviesen integrados en la estructura del archivo instalador y no en un servidor externo, y, por último, que dichos archivos estuviesen en un formato distinto al estándar propuesto por Android para la localización (Android API Guides, [s.f.]) (XML).

Por esta razón, durante la fase inicial del trabajo, mi tarea consistió en descargar los archivos de instalación de las aplicaciones (APK) y comprobar que cumplían estas condiciones mediante el proceso de descompilación que explicaremos más adelante.

Durante esta fase mi idea era buscar un juego del género RPG, ya que contienen mucho texto y podían permitirme desarrollar una tarea de comparación crítica de traducciones. Para esto, intenté ponerme en contacto con los desarrolladores de uno de los juegos que tengo instalados en mi móvil para pedirles permiso para utilizarlo aunque éste no cumplía uno de los requisitos (es necesaria conexión a internet para abrirlo), ya

que cuando empecé a jugar no tenía traducción al español y me parecía interesante poder desarrollar una. Sin embargo, no obtuve ninguna respuesta por su parte por lo que tuve que descartar esta idea.

Tras descargar descompilar y desechar más de 300 juegos de todos los géneros por no cumplir las condiciones pactadas en inicio, encontré un juego arcade (casual) de tipo *Candy Crush Saga*, que las cumplía todas y que, por lo tanto, me permitía desarrollar la labor de localización. Este juego es el *Witchy World*. Sin embargo, su carga textual es bastante reducida, lo que me llevó a desarrollar la parte más técnica del proceso de localización, el papel de ingeniera de localización. Además, según los datos, los juegos casuales son los más populares (número de descargas en Playstore). Como demuestra la siguiente tabla, en la que de los diez juegos más descargados en 2015, cuatro son casuales.

Nombre (Desarrollador)	Descargas	Licencia	País	Género
PlantsVsZombies2 (EA)	371236	Free	US	Strategy
SubwaySurfers(Kiloo)	273128	Free	DK	Platform
GeometryDash(RobTopGames)	266220	Free	SW	Platform
Pou(Zakeh)	236702	Free	LB	Casual
CandyCrushSaga (King)	184351	Free	UK	Casual
ClashofClans(Supercell)	168773	Free	FI	Strategy
FIFA 14 (EA)	162968	Free	US	Sports
MyTalkingAngela(Outfit7)	131413	Free	US	Casual
TriviaCrack (Etermax)	102469	Free	ES	Casual
FIFA 15UltimateTeam(EA)	100272	Free	US	Sports

Figura 12: Los 10 juegos más descargados de 2015

Fuente: <http://elcomercio.pe/visor/1815703/1127553-20-videojuegos-mas-descargados-android-durante-2015-noticia>

7 Proceso de descompilación

Para poder realizar las comprobaciones necesarias en la estructura de los archivos de instalación de las aplicaciones, era necesario disponer de una herramienta que me permitiese descompilar dichos archivos para ver su estructura interna y los archivos de localización contenidos en ellos.

La herramienta que se utilizó para este proceso fue APKTool, una herramienta de manipulación de aplicaciones Android de terceros, que permite hacer cambios en los archivos que las integran para modificarlas (MODs), para localizar los archivos no traducidos, etc. También permite revertir las APKs a su forma original compilada para realizar las comprobaciones necesarias sobre el funcionamiento de los cambios efectuados.

La versión de la que se disponía era una versión simplificada en la que, al contrario que en la versión inicial, no era necesario ejecutar comandos en la consola de comandos de Windows, es decir, en el símbolo de sistema (CMD) ya que el programa los ejecutaba automáticamente al arrastrar hasta el componente adecuado la APK seleccionada:

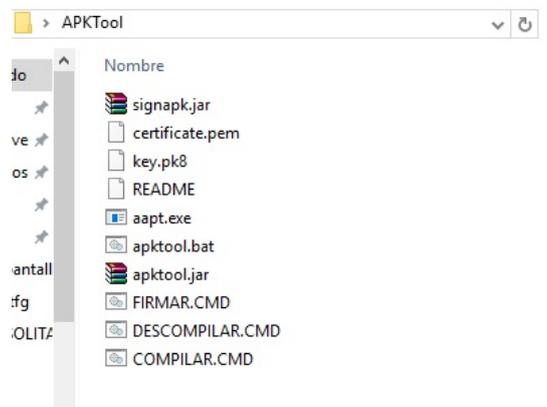


Figura 13: Estructura del programa APKTool de manipulación de aplicaciones
Fuente: Elaboración propia

```
C:\WINDOWS\system32\cmd.exe
C:\Users\HELENA\Desktop\APKTool\air.com.sublinet.witchyworld-5.50-APK4Fun.com_sig.g.apk
air.com.sublinet.witchyworld-5.50-APK4Fun.com_sig
I: Baksmling...
I: Loading resource table...
I: Loaded.
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:\Users\HELENA\apktool\framework\1.apk
I: Loaded.
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Done.
I: Copying assets and libs...
```

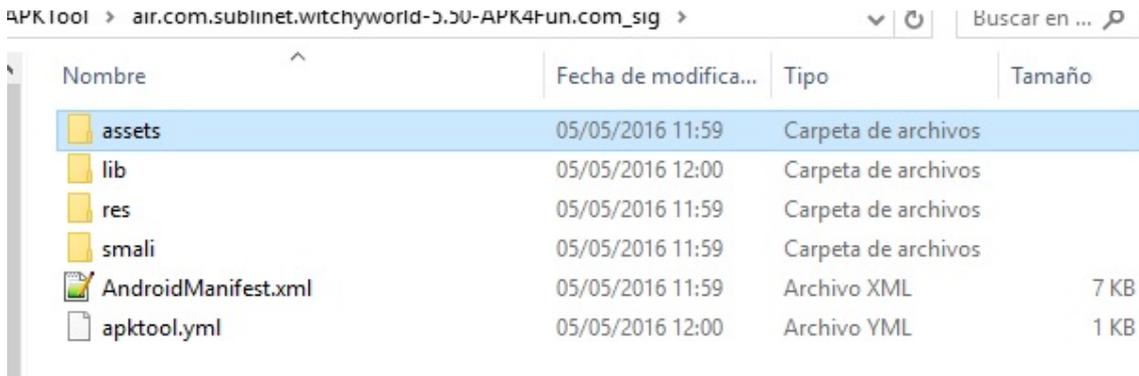
Figura 14: Comandos ejecutados por APKTool en CMD
Fuente; Elaboración propia

El resultado de este proceso es una carpeta con el mismo nombre que la APK del juego que contiene todos los archivos descompilados y listos para su manipulación.

8 Archivo descompilado

8.1 Estructura

La estructura general del archivo descompilado contiene el Manifiesto de Android y cuatro carpetas llamadas *assets*, *lib*, *res* y *smali*.



Nombre	Fecha de modifica...	Tipo	Tamaño
assets	05/05/2016 11:59	Carpeta de archivos	
lib	05/05/2016 12:00	Carpeta de archivos	
res	05/05/2016 11:59	Carpeta de archivos	
smali	05/05/2016 11:59	Carpeta de archivos	
AndroidManifest.xml	05/05/2016 11:59	Archivo XML	7 KB
apktool.yml	05/05/2016 12:00	Archivo YML	1 KB

Figura 15: Estructura de APK descompilada
Fuente: Elaboración propia

En las APKs, los archivos de localización suelen estar contenidos en una subcarpeta dentro de la carpeta "res" llamada "values" (original) o "values-es" (español), "values-en" (inglés), etc., dentro de las cuales se encuentran los archivos XML denominados *strings* que contienen el texto traducible.

Aunque la APK elegida tenía esta estructura y los archivos *strings*, dentro de ellos no se encontraba el texto interno del juego, sino solamente la parte referente a la conexión con Facebook. En esta APK, los archivos de localización del juego se encontraban en una carpeta hija de "assets", denominada a su vez "assets", que contenía varias carpetas, entre ellas una llamada "lang" en la que se encontraban los archivos JSON de cada idioma que contenían el texto traducible.

Nombre	Fecha de modifica...	Tipo
cz.json	05/05/2016 11:59	Archivo JSON
da.json	05/05/2016 11:59	Archivo JSON
de.json	05/05/2016 11:59	Archivo JSON
el.json	05/05/2016 11:59	Archivo JSON
en.json	05/05/2016 11:59	Archivo JSON
es.json	05/05/2016 11:59	Archivo JSON
fr.json	05/05/2016 11:59	Archivo JSON
it.json	05/05/2016 11:59	Archivo JSON
ja.json	05/05/2016 11:59	Archivo JSON

Figura 16: Ruta de archivos JSON en APK descompilada
Fuente: Elaboración propia

8.2 El JSON de la aplicación

Como se explicaba en el marco teórico, los archivos JSON tienen un formato sencillo compuesto por dos tipos de estructuras. En el caso del JSON correspondiente a esta aplicación hay un único tipo de estructura que se compone únicamente de objetos (pares nombre:valor) acotados por llaves.

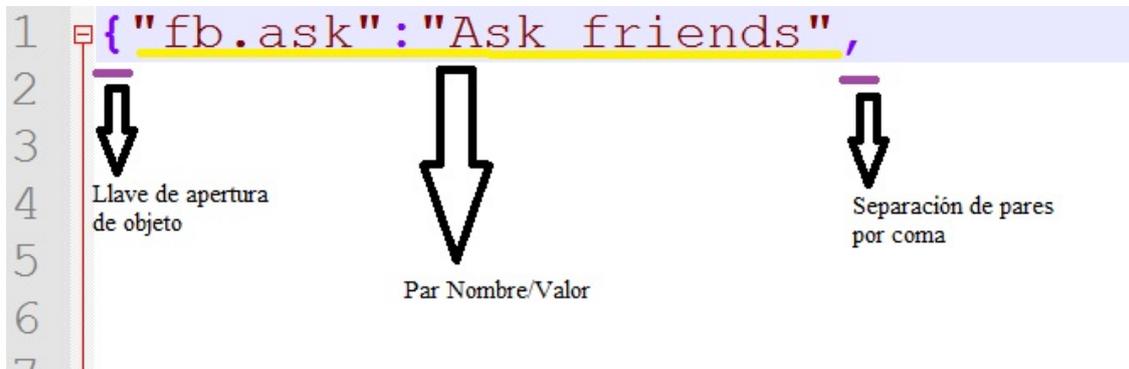


Figura 17: Ejemplo estructura inicial JSON
Fuente: Elaboración propia

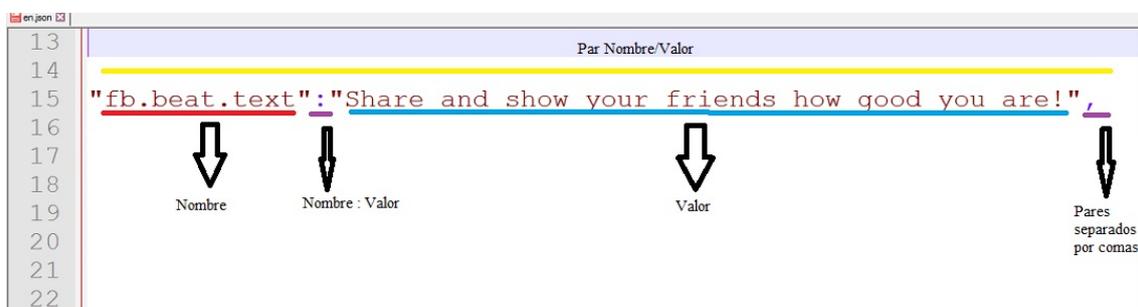


Figura 18: Ejemplo de objeto en el JSON de la aplicación
Fuente: Elaboración propia

Dentro de estos pares se pueden encontrar partes no traducibles, es decir, parámetros dentro del texto que en tiempo de ejecución adquirirán un valor. Estos

parámetros aparecen con diferentes formas y son, en este caso, texto delimitado por llaves, como las de inicio y cierre del archivo pero que se encuentran dentro del par objeto:valor, y deberían capturarse como etiquetas internas para evitar errores de funcionamiento. En el caso de este archivo, estos parámetros aparecen como estructuras simples {name} o compuestas {quantity:Number}.

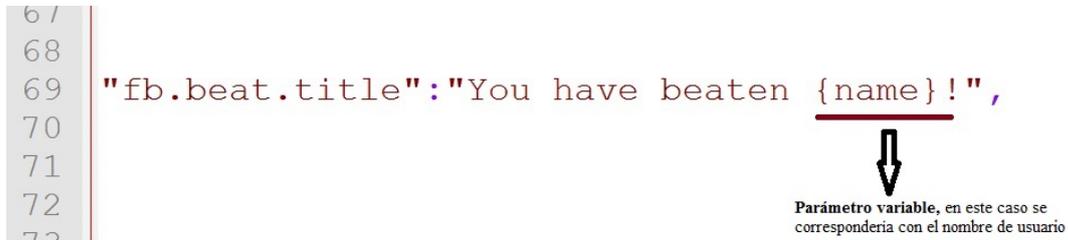


Figura 19 Ejemplo de “etiqueta” simple
Fuente: elaboración propia

Además, el archivo JSON de esta aplicación tenía otra particularidad, los caracteres no ASCII en el texto se codifican como caracteres escapados de acuerdo al número UNICODE¹² en 16 bits (en realidad, cuatro dígitos hexadecimales). Es decir, que los caracteres no americanos se representan en el texto con \uXXXX, donde “\” significa carácter escapado, “u” en forma UNICODE, y XXXX el número en 4 dígitos hexadecimales del carácter a representar en la pantalla del dispositivo. En la siguiente figura puede verse un ejemplo de cómo se codifica la “ó” de la palabra “aplicación”.

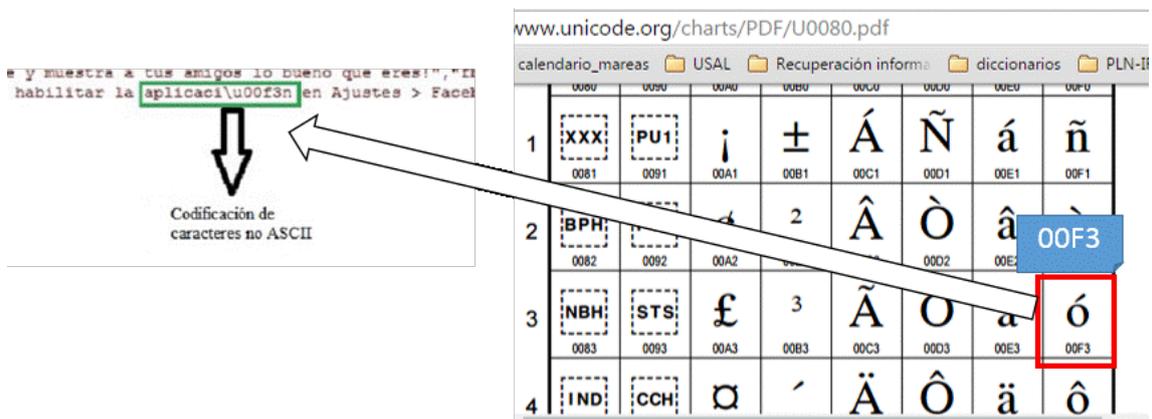


Figura 20: Ejemplo de codificación de caracteres no ASCII
Fuente: Elaboración propia

¹² UNICODE (2015): *Unicode 8.0 Character Code Charts*. <http://www.unicode.org/charts/>

9 Propuestas técnicas: llevando el archivo de localización a una TAO

En lo que se refiere a la meta más práctica del trabajo, es decir, tomar el JSON original no localizado y crear un archivo JSON ya traducido y con la misma codificación y características que el original, se expondrán dos métodos para conseguirlo, que se compararán en las conclusiones.

El primer proceso consistirá en el uso de una herramienta de procesamiento de archivos para transformar el formato original en el formato de compatibilidad de las herramientas TAO (XLIFF) y la posterior traducción del archivo XLIFF resultante mediante una herramienta TAO, en este caso Trados. El segundo consistirá en utilizar una herramienta especializada de localización online, que incluye los filtros necesarios para procesar y traducir los archivos JSON, es decir, Transifex.

Las ventajas e inconvenientes de estos dos procesos se analizarán en las conclusiones de este trabajo.

9.1 *De JSON a XLIFF y vuelta, con una herramienta independiente*

Para poder trasladar el archivo JSON a una herramienta TAO como Trados, debemos procesarlo previamente con una herramienta auxiliar que pueda transformar el formato JSON para el que la herramienta TAO no tiene filtro. En este caso la herramienta que vamos a utilizar es Rainbow, una herramienta que forma parte de *Okapi Framework*, que de acuerdo con su página web «aims at being a crucible where we forge common components that can be used in any localization and translation application, providing faster development time and better interoperability, but still allowing for the diversity of solutions.¹³».

El proceso inicial se llevó a cabo en Transifex, ya que para éste no era necesario realizar ninguna manipulación previa del archivo; pero dado que el núcleo del trabajo es realizar el proceso de conversión del formato, se explicará previamente dicho paso aunque en algunas ocasiones se haga referencia al de Transifex, explicado más adelante.

¹³ Su meta es ser el crisol en el que forjemos componentes comunes que puedan usarse en cualquier aplicación de traducción o localización, procurando un tiempo de desarrollo más rápido y mejor interoperabilidad, pero permitiendo la diversidad de soluciones.

9.1.1 Proceso de conversión de JSON a XLIFF con Rainbow

El primer paso fue descargar Rainbow de la página web de Okapi¹⁴. Una vez descargado y comprobado el funcionamiento de Rainbow, el siguiente paso era añadir a éste el archivo JSON y crear un filtro específico para él. Gracias a la ayuda del tutor y a su explicación sobre lenguaje de Expresiones Regulares (Regular Expressions), fue posible crear un filtro que reconociese las etiquetas internas contenidas en el archivo JSON (tanto las simples como las compuestas). En principio habíamos creado un filtro para las etiquetas del texto, pero al empezar el trabajo individual se descubrió que existía otro tipo de etiquetas compuestas que el filtro creado no reconocía. Por esta razón y utilizando lo explicado por el tutor, se creó un nuevo filtro que incluyera cualquier número de caracteres de texto en mayúscula o minúscula ([a-zA-Z]+) separados por dos puntos y encerrados entre llaves `\{[a-zA-Z]+\};[a-zA-Z]+\}`.

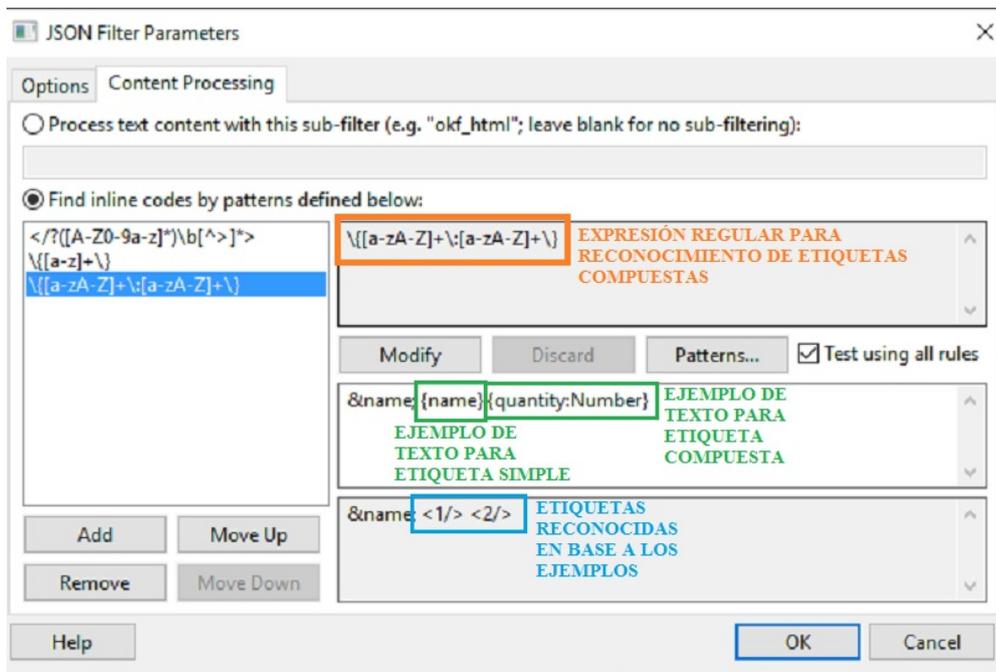


Figura 21 Creación de filtro para reconocimiento de etiquetas internas
Fuente: Elaboración propia

El último paso de este proceso consistió en abrir el archivo XLIFF resultante con el NotePad++ para comprobar que tanto la codificación como las etiquetas se visualizaban correctamente en el archivo bilingüe.

¹⁴ <https://bintray.com/okapi/Distribution>



Figura 22: Comparación de archive JSON original y XLIFF procesado con etiquetas
Fuente: Elaboración propia

9.1.2 Proceso de traducción del XLIFF en una TAO: Trados

Gracias a los procesos desarrollados anteriormente con Rainbow y Transifex, la traducción del archivo XLIFF generado con Rainbow fue muy sencilla.

El proceso comenzó por crear un proyecto de traducción en Trados y añadir como texto para traducir el archivo XLIFF transformado con Rainbow. Después, se creó una memoria de traducción (MT) del proyecto en la que se importó el contenido de la descargada del proyecto de Transifex. Una vez importado el contenido de la memoria se comenzó con la traducción del texto. Este proceso fue muy breve ya que gracias a la MT del proyecto de Transifex sólo fue necesario traducir aquellos segmentos que contenían etiquetas. Además, gracias al filtro y a las características de Trados (quickplace), este reconocía las etiquetas y permitía trasladarlas automáticamente al segmento traducido.

Una vez terminado el proceso de traducción, se procedió a cerrar el proyecto, actualizar la MT con los segmentos que no contenía y generar el archivo de destino traducido.

9.1.3 Proceso de reconversión de XLIFF a JSON con Rainbow

Una vez generado el archivo XLIFF con la traducción, se volvió a colocar en la carpeta de Rainbow en la que se había generado, después se arrastró hasta la consola de Rainbow el documento *Manifest.rkm* y se procedió al post-procesado de los archivos. Así, se creó una carpeta de salida de Rainbow en la que se encontraba el archivo en JSON de nuevo. Sin embargo, la codificación del archivo no era la correcta, así que se debió arrastrar este JSON a Rainbow otra vez y generar un nuevo proceso mediante la opción *Execute Pipeline* que se encuentra en el menú *Utilities*. Aquí se debió añadir un paso nuevo entre las posibilidades que se ofrecían, que a su vez hizo aparecer una nueva pantalla en la que se debía escoger, en la pestaña de *Output*, todos los caracteres extendidos y después *Lowercase Java-style notation*.

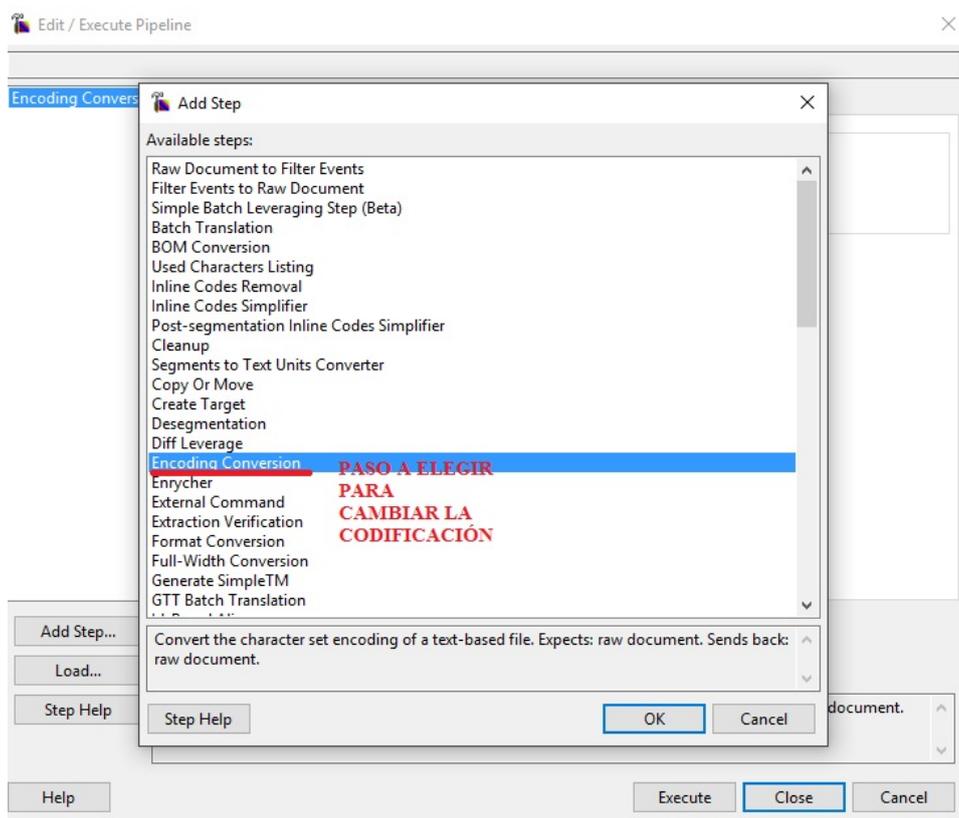


Figura 23: Elección de paso para recodificar el JSON
Fuente: Elaboración propia

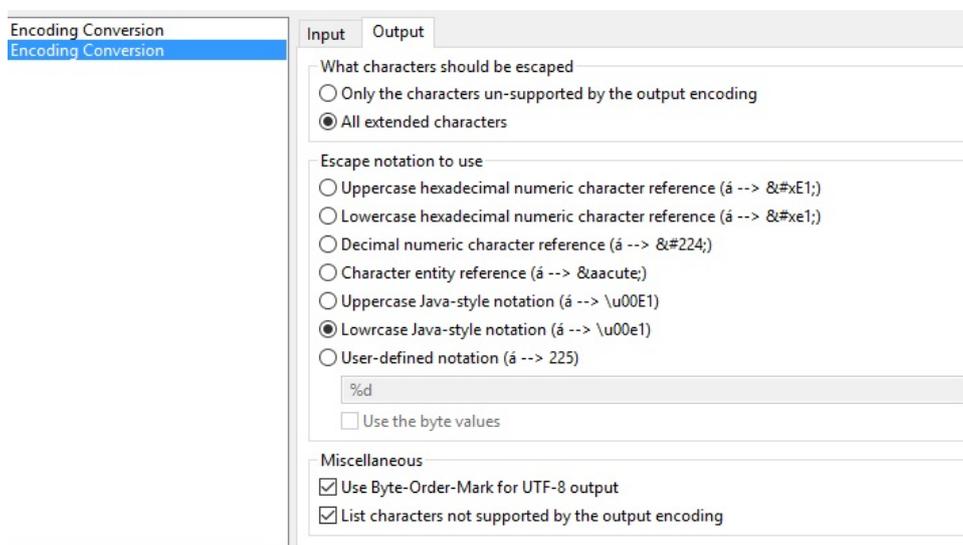


Figura 24: Elección de opciones para la conversión de la codificación del JSON
Fuente: Elaboración propia

Este último proceso podría llevarse a cabo con el archivo de salida generado por Transifex para subsanar el problema de la codificación y generar un archivo JSON con las características del original y preparado para reubicarse en su lugar inicial y comenzar el proceso de recompilación de la APK.

9.2 El archivo JSON directamente a una TAO

Para el proceso directo (sin herramientas auxiliares), la herramienta elegida fue Transifex, que según su página web es una plataforma online de gestión de localización que permite gestionar múltiples proyectos con aplicaciones, páginas web y documentación desde una única plataforma centralizada. Al ser una herramienta de pago, el primer proceso fue crear una cuenta de prueba gratuita (15 días) con la que, durante este periodo de tiempo, se puede acceder a todos los recursos de la plataforma: creación de equipos de traducción, creación de proyectos de localización e integración de los equipos, subida de los archivos, almacenamiento en la nube, integración de una página web para su localización, etc.

Todos estos procesos, así como la gestión de los proyectos y de los integrantes de los equipos son accesibles desde la consola principal de la plataforma. Además permite ver las aportaciones de cada miembro del equipo para facilitar el trabajo del gestor.

En este trabajo, al ser una única persona la encargada de todos los procesos estas funcionalidades no resultaban necesarias pero sí es conveniente señalar que cuenta con ellas, si bien se trata esta cuestión en las partes finales de este trabajo.

El segundo paso de este proceso fue crear un proyecto de localización y añadir el archivo JSON a la plataforma para comprobar si ésta contenía el filtro necesario para traducir este formato.

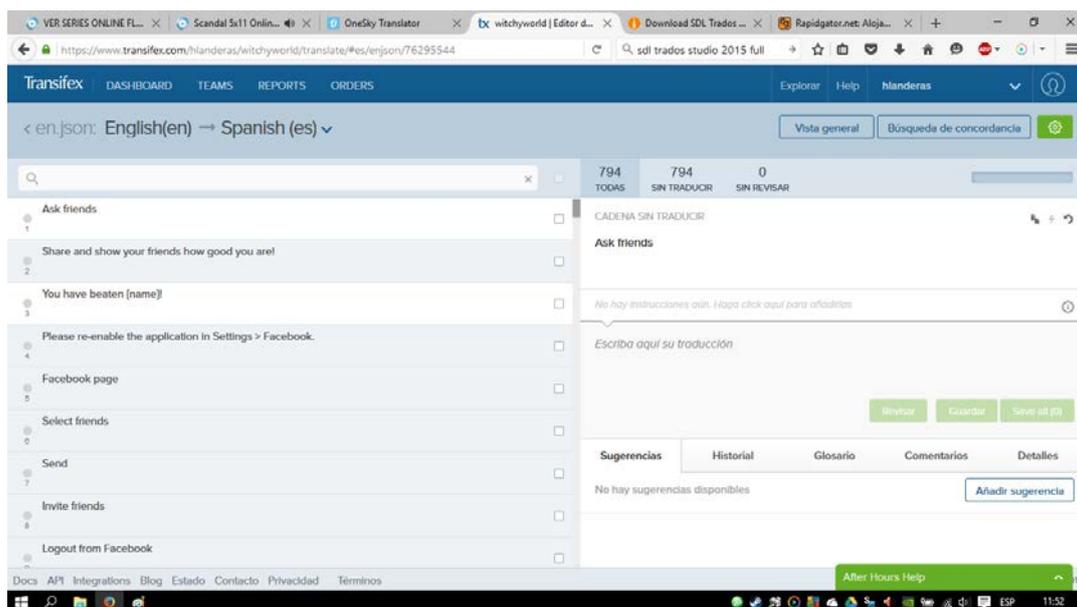


Figura 25: Archivo JSON añadido a TRANSIFEX
Fuente: Elaboración propia

Como puede observarse en esta imagen, la plataforma dispone del filtro necesario para reconocer el formato JSON, aunque las etiquetas internas no se reconocen como tal, por lo que el traductor debe reconocerlas como texto no traducible y copiarlas en los segmentos traducidos.

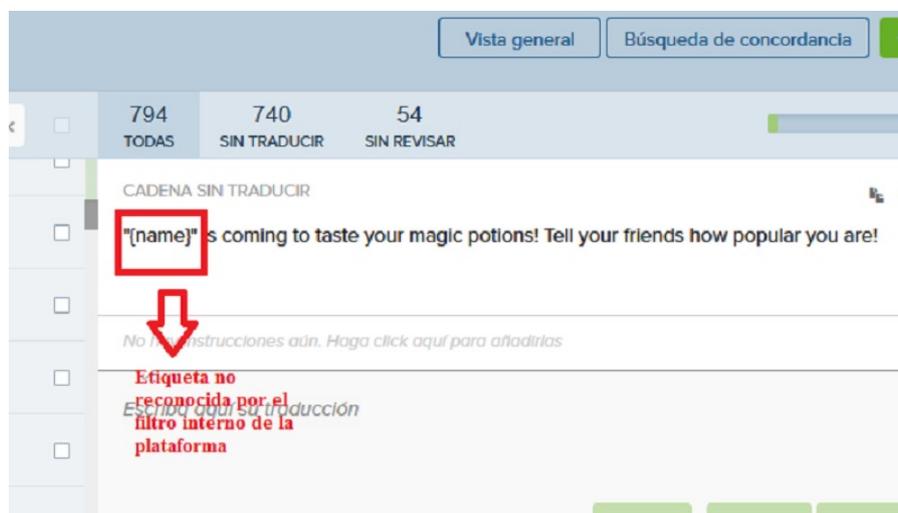


Figura 26: Ejemplo de etiqueta no reconocida
Fuente: Elaboración propia

Dejando de lado el problema de las etiquetas, que en este caso no representaba mayor problema ya que el texto no contenía una gran cantidad, el proceso de traducción del texto se asemeja al de traducción con cualquier TAO, con el texto original a la izquierda de la pantalla y la consola de traducción a la derecha con el segmento original en la parte superior y la traducción en la inferior.

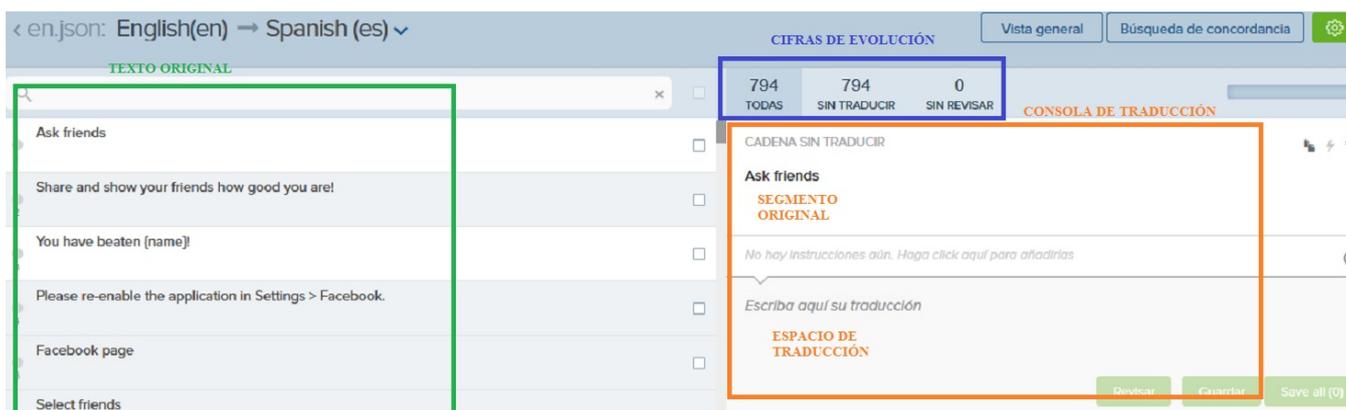


Figura 27: Estructura de la consola de traducción
Fuente: Elaboración propia

Durante la realización del proceso de traducción de la aplicación con Transifex, se descubrió que parte del texto contenido en el archivo JSON de *Witchy World* no se correspondía con el texto que aparece en el juego, sino que era parte del texto de otro juego de la misma compañía, el *Tasty Tale*. Dado este descubrimiento, se procedió a la

descarga de la APK de este juego para añadirlo al proyecto original y comprobar el número de segmentos que se traducirían automáticamente con la memoria de traducción del proyecto, es decir, con la traducción que ya se había realizado de todo el texto del juego.



Figura 28: Cifras de traducción automática con MT del Proyecto
Fuente: Elaboración propia

El resultado fue que casi todo el texto del segundo juego se tradujo automáticamente gracias a la Memoria de Traducción (en adelante MT) del proyecto original (la memoria de traducción del primer proyecto de Transifex, en el que se tradujo por completo el juego *WitchyWorld* que contenía 714 segmentos, un total de 3136 palabras), lo cual demuestra la utilidad de la plataforma o, en este caso, debido al acceso restringido por la versión de prueba, la descarga de la misma o la creación de una MT propia que ahorre tiempo de traducción, sobre todo si los juegos o aplicaciones que se traducen son del mismo desarrollador o actualizaciones de uno en particular.

Una vez traducidos los juegos, se descargó el archivo resultante desde la consola de la plataforma. También se descargó la MT del proyecto en formato TMX y los archivos traducidos en formato XLIFF

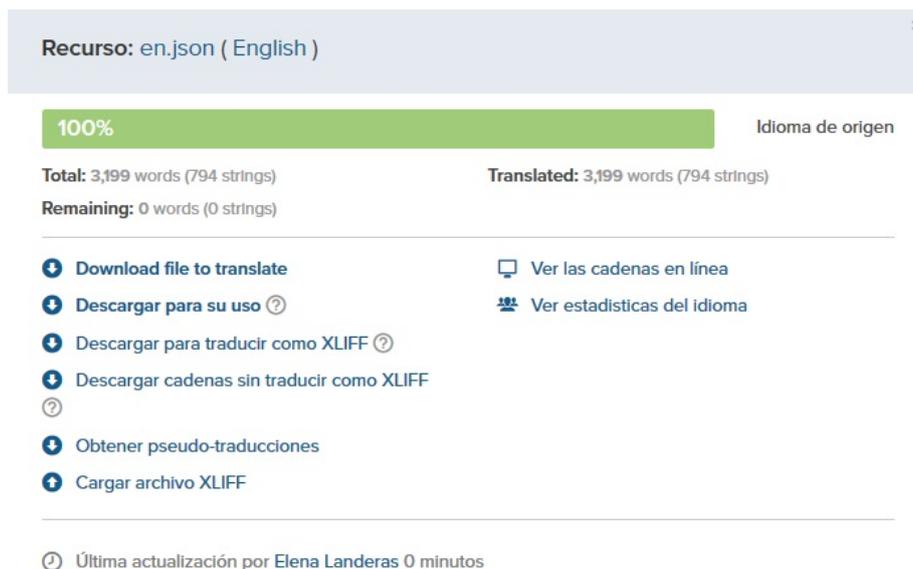


Figura 29: Opciones de finalización de la consola TRANSIFEX
Fuente: Elaboración propia

Una vez descargados los archivos, se procedió a la comprobación de la codificación del archivo JSON resultante. Por desgracia, el archivo JSON descargado no estaba codificado de la misma forma que el original, los caracteres españoles se visualizaban directamente, no en la forma escapada `\uXXXX`, que es la requerida por la aplicación. .

```
"¡Has vencido a {name}!",  
tionTurnedOff": "Por favor, vuelve a habilitar la aplicación en Ajustes > Facebook",
```

CARACTER NO CODIFICADO EN LITTLE ENDIAN

Figura 30: Ejemplo de codificación en el JSON saliente
Fuente: Elaboración propia

Por tanto, era necesario utilizar una herramienta auxiliar para recodificar el texto de modo que se obtuviera la versión del fichero con los caracteres no ASCII escapados según explicamos anteriormente, o lo que es lo mismo, que se visualizase igual que el texto original y no diese problemas al recompilarlo y probarlo.

Como prueba de funcionamiento con texto procesado, se creó un nuevo proyecto en el que se añadió el archivo XLIFF procesado con Rainbow con las etiquetas internas marcadas como tales.

✓ 2	Share and show your friends how good you are!	Comparte y demuestra a tus amigos lo bueno que eres	<input type="checkbox"/>
✓ 3	You have beaten <code><x id="1"/>!</code>	¡Has vencido a <code><x id="1"/>!</code>	<input type="checkbox"/>
✓ 4	Please re-enable the application in Settings > Facebook.	Por favor, vuelve a habilitar la aplicación en Ajustes > Facebook	<input type="checkbox"/>
✓ 5	Facebook page	Página de Facebook	<input type="checkbox"/>
	Select friends	Seleccionar amigos	

Figura 31: Ejemplo de etiqueta reconocida en Transifex
Fuente: Elaboración propia

En el proceso de traducción se observó que Transifex no tiene opción de traslado de etiquetas, por lo que en todo caso debían copiarse manualmente por el traductor. Debido a esta cuestión, al generar el archivo de salida las etiquetas aparecían copiadas como texto y no se reconocían como etiquetas. Esto representaría un problema porque cambia la representación de las etiquetas y podría generar problemas al reconstruir el archivo.

```

</trans-unit>
<trans-unit id="tu3" resname="fb.beat.title" xml:space="preserve">
<source xml:lang="en-us">You have beaten <x id="1"/></source>
<target xml:lang="es-es">¡Has vencido a &lt;x id="1"/>!</target>
</trans-unit>

```

Figura 32: Ejemplo de etiqueta copiada en Transifex
Fuente: Elaboración propia

10 Recompilación y prueba del juego traducido

Con el archivo ya traducido y devuelto a su formato original (JSON) con la codificación correcta, será necesario cambiarle el nombre para que el identificador de idioma coincida con el del idioma al que ha sido traducido. Después, será necesario devolverlo a la carpeta "lang" de la que lo extrajimos en primer lugar para poder recompilar la APK arrastrándola a la opción correspondiente de APKTool, que recompondrá el archivo de instalación del juego. Una vez recompilado será necesario firmarlo, este proceso también lo llevará a cabo el APKTool, y ya será posible transferir la APK a un dispositivo Android en el que se instalará la aplicación y se podrá comprobar que funciona correctamente.



Figura 33: Captura de pantalla del juego recompilado
Fuente: Elaboración propia

En el ejemplo superior, una captura de pantalla de la pantalla de inicio del juego con el botón *Jugar* manipulado para mostrar otro texto.

11 De los JSON paralelos a TMX

Este punto del proceso podría haber sido el primero dado que el juego elegido para el proceso experimental ya incluía el texto traducido al español. La razón por la que se encuentra al final del proceso es que dicho texto parecía provenir de una traducción automática y, por lo tanto, la calidad de la traducción era muy baja, no aportaba ninguna ventaja en el trabajo de traducción.

En una situación de trabajo como ingeniero de traducción y si se dispone de textos originales y traducidos pero no de una MT proporcionada por el desarrollador, el proceso de creación de una MT mediante la alineación sería el primer paso.

11.1 Motivación

Es muy importante incluir este proceso ya que, en los casos de traducción de actualizaciones de aplicaciones o, como sucedió en este caso, de otras aplicaciones del mismo desarrollador que contengan el mismo texto, puede ahorrar mucho tiempo de traducción disponer de una MT creada en base a los textos alineados.

Además, resulta de gran ayuda para mantener el léxico anterior del juego o de sus secuelas, si las hubiere, y así evitar cambiar las traducciones de elementos que ya estaban traducidos previamente. Otra ventaja importante de disponer de una MT como esta es que aquellos segmentos descontextualizados con los que el traductor tiene más problemas pueden solucionarse por estar ya traducidos previamente.

11.2 Alineación de XLIFF con Trados

Una de las ventajas de Trados es que en sus últimas versiones incluye un alineador de textos integrado. En las versiones previas este alineador era una herramienta separada llamada WinAlign que funcionaba bien en textos sencillos pero que no permitía arreglar los problemas detectados en la alineación automática. El nuevo alineador integrado funciona correctamente con los textos sencillos y, además, si se detecta algún fallo de alineación o una descompensación de la cantidad de texto en los segmentos alineados, permite arreglar estos problemas manualmente de forma muy sencilla, modificando las uniones de segmentos incluso cortando y pegando texto entre segmentos.



Figura 34: Opciones de manipulación de la alineación
Fuente: Elaboración propia

Como se comentó en el proceso de traducción del archivo JSON con Trados, esta herramienta no tiene filtro para este formato. Por eso, para poder realizar la alineación con este programa, primero deberemos convertir los archivos JSON de los idiomas que queramos alinear a XLIFF con el proceso ya explicado de Rainbow. Una vez tengamos los archivos convertidos a un formato compatible con Trados, solo deberemos hacer clic en la pestaña *Alinear documentos* del menú *Herramientas*. En el menú desplegable se proporciona tres opciones, en este caso sólo se trataba de alinear una combinación de idiomas (en-es) así que se seleccionó esta opción. El siguiente paso es seleccionar o crear una memoria de traducción para guardar los datos resultantes de la alineación y añadir los archivos de origen y destino para alinear. En este caso se creó una MT de Trados nueva ya que el proceso de traducción estaba hecho. Más tarde se intentó añadir la alineación de los textos tanto de *Witchy World* como de *Tasty Tale* a la memoria descargada de Transifex y a la creada en el proyecto de traducción de Trados para tenerlas completas con todo el texto de ambos juegos.

En principio la alineación parecía ser correcta pero al llegar a algunos segmentos se descompensaba dejando segmentos de inicio alineados con varios de destino o segmentos de destino sin unir con ninguno de origen.

Share the game with a few friends and get 3 more clients immediately!	118	SEGMENTO SIN ALINEAR	118	¡Comparte este juego con algunos amigos y consíguelte ahora mismo 3 clientes más!
More clients, more fun!	119		119	¡Más clientes, más diversión!
Wait for clients (X)	120		120	Espera por clientes (X)
Get 3 more clients immediately!	121		121	¡Consíguelte ahora mismo 3 nuevos clientes!
You just beat (X) Tell everybody that you got a better score!	122		122	¡Acabas de ganarle a (X)! ¡Cuéntale a todos que obtuviste una mejor puntuación!
What a brilliant chef!	123	SEGMENTOS MAL ALINEADOS	123	¡Qué chef más brillante!

Figura 35: Ejemplo de alineación incorrecta
Fuente: Elaboración propia

Una vez arreglados estos problemas y confirmados todos los cambios en los enlaces de alineación, se debe guardar el archivo de alineación e importar los resultados a la MT. Como último paso, se exportó esta MT a un archivo TMX compatible para su posible uso en otras herramientas TAO, y se comprobó mediante su apertura en NotePad++ que la información estaba representada correctamente.

```

373 | </tuv>
374 | <tu creationdate="20160307T115537Z" creationid="HELENA-FC\HELENA" changedate="20160307T115537Z" changeid="HELENA-FC\HELENA"
375 | <prop type="x-Context">-8044167582549841891, 3505856920909</prop>
376 | <prop type="x-Origin">TM</prop>
377 | <prop type="x-OriginalFormat">Unknown</prop>
378 | <tuv xml:lang="en-US">
379 | <seg>Chocolate Coffee Ganache</seg>
380 | </tuv>
381 | <tuv xml:lang="es-ES">
382 | <seg>Ganache de chocolate y café</seg>
383 | </tuv>
384 | </tu>

```

Figura 36: Ejemplo de unidad de traducción en archive TMX exportado de Trados
Fuente: Elaboración propia

11.3 Alineación de JSON con Rainbow

El proceso de alineación con Rainbow de este archivo es muy sencillo, tan solo es necesario arrastrar el archivo de origen a la pestaña *Input List 1* y el de destino a la pestaña *Input List 2*. Después únicamente deberemos elegir en el menú desplegable *Utilities* la opción *ID-Based Alignment*. Este sistema de alineación es la denominada alineación por identificación mediante la cual el programa alinea únicamente la intersección de texto de ambos idiomas, utilizando como método para alinearlos el identificador, es decir, la parte no variable del objeto. Como ya se ha explicado, en la estructura del formato JSON los objetos se componen de pares de nombre:valor, por lo que la alineación se realizará relacionando las partes variables (los valores) que tengan el mismo identificador (nombre) en ambos idiomas. El principal problema de este método de alineación es que, si bien genera menos errores al utilizar solamente el identificador, no segmenta el texto, por lo que podríamos encontrarnos con alineaciones de párrafos enteros. Si queremos evitar este problema podemos marcar la opción de utilizar normas de segmentación, disponible en Rainbow. Dado que el programa tiene un filtro JSON propio, se puede utilizar ese, aunque si lo que queremos es que se reconozcan las etiquetas internas del texto deberemos editar el filtro como hicimos para convertir el texto a XLIFF.

```

]<tu tuid="tu78_s0">
<prop type="Txt::FileName">en.json</prop>
<prop type="Txt::GroupName">fb.share.friendBeaten.title</prop>
<tuv xml:lang="en-us"><seg>You beat {name}!</seg></tuv>
<tuv xml:lang="es-es"><seg>¡Ganaste a {name}!</seg></tuv>
</tu>
]<tu tuid="tu79_s0">
<prop type="Txt::FileName">en.json</prop>
<prop type="Txt::GroupName">fb.share.friendBeaten.twitter</prop>
<tuv xml:lang="en-us"><seg>I beat {name} on level {level}! Who is the best witch now?</seg></tuv>
<tuv xml:lang="es-es"><seg>Le gané a {name} en el nivel {level}! ¿Quién es la mejor bruja ahora?</seg></tuv>
</tu>

```

Figura 37: Ejemplo de alineación con filtro sin modificar
Fuente: Elaboración propia

```

</tu>
<tu tuid="tu31_s0">
<prop type="Txt::FileName">en.json</prop>
<prop type="Txt::GroupName">fb.inbox.ask_episode</prop>
<tuv xml:lang="en-us"><seg><ut>{\cs6\fi\cf6\lang1024 </ut>{name}<ut>}</ut> needs your help to unlock the next episode!</seg></t
<tuv xml:lang="es-es"><seg><ut>{\cs6\fi\cf6\lang1024 </ut>{name}<ut>}</ut> necesita tu ayuda para desbloquear el siguiente epis
</tu>
ETIQUETA RECONOCIDA POR EL FILTRO
<tu tuid="tu32_s0">
<prop type="Txt::FileName">en.json</prop>
<prop type="Txt::GroupName">fb.inbox.ask_life</prop>
<tuv xml:lang="en-us"><seg><ut>{\cs6\fi\cf6\lang1024 </ut>{name}<ut>}</ut> is out of clients. Send a client now!</seg></tuv>
<tuv xml:lang="es-es"><seg><ut>{\cs6\fi\cf6\lang1024 </ut>{name}<ut>}</ut> le faltan clientes. ¡Envíale un cliente!</seg></t

```

Figura 38: Ejemplo de alineación con filtro modificado
Fuente: Elaboración propia

12 Comparación de las herramientas y los procesos propuestos en el Trabajo.

En la época en que vivimos, y en particular en el ámbito de la localización, se hace cada vez más necesario disponer de las mejores herramientas para el desarrollo de nuestra actividad profesional. A este respecto, es imperativo que los profesionales de la localización posean los conocimientos técnicos necesarios para sacar el máximo provecho de estas herramientas.

La labor del localizador se ve facilitada por los recursos informáticos de los que dispone (herramientas TAO y de localización, herramientas de gestión online que ponen en contacto equipos de traductores de todo el mundo, herramientas de gestión de MT, etc.), tanto en el tiempo y el esfuerzo que invierte en cada proyecto como en el mantenimiento de la coherencia léxica entre versiones o entregas de un mismo producto o serie.

En base a esta necesidad, se comparan aquí las herramientas utilizadas para la realización del experimento a fin de ofrecer, a modo de conclusión, una propuesta de uso de cada una.

TRANSIFEX	TRADOS	TRANSIFEX	TRADOS
VENTAJAS	VENTAJAS	INCONVENIENTES	INCONVENIENTES
Gestión de múltiples proyectos	Gestión de múltiples proyectos	No reconoce las etiquetas internas	No permite creación o gestión de equipos y roles
Plataforma centralizada	Menús intuitivos y sencillos de utilizar	Cambio de codificación de documentos de salida	Cambio de codificación de documentos de salida
Creación y gestión de equipos y participantes	Quickplace para traslado de etiquetas	No quickplace o métodos similares de traslado de etiquetas	Almacenamiento interno
Creación y actualización de MTs	Creación y actualización de MTs	Herramienta online (necesaria conexión a internet)	No incluye filtros para archivos de localización
Almacenamiento en la nube	Reconoce etiquetas internas	Herramienta de pago	Herramienta de pago
Monitorización del trabajo del equipo	Uso de archivos en formato XLIFF compatible		
Descarga de MT en formato TMX compatible	Exportación de MT en formato TMX compatible		
Descarga de archivos en formato XLIFF compatible	Alineador de textos interno		
Filtros específicos de localización			

Tabla 3: Comparativa de las herramientas utilizadas
Fuente: Elaboración propia

Como se ha ido explicando, Rainbow es una herramienta muy útil para la labor de localización. Al facilitar al localizador la conversión de los archivos al formato XLIFF, le permite liberarse de las restricciones de las herramientas TAO o de localización. El localizador deja de ser esclavo de las herramientas y no necesita buscar para cada encargo aquella que incluya el filtro específico para el formato en el que esté el texto. Además, la creación y edición de los filtros facilita la labor del traductor ya que pueden reconocer las etiquetas internas, eliminando los problemas de funcionamiento derivados de la traducción del texto contenido en ellas.

En los casos en los que, como en el proceso experimental con Transifex, se cambie la codificación del lenguaje, Rainbow permite transformarlo para devolver el archivo a las condiciones de origen. Por último, al igual que Trados, esta herramienta permite también la alineación de textos.

13 Conclusiones

La industria de la localización funciona cada vez con plazos más reducidos, se hace necesario tener los máximos recursos posibles para ser capaces de cumplir estos plazos y ahorrarnos tiempo y esfuerzo reutilizando los recursos que ya teníamos. Además, tanto la industria del software como la de los videojuegos avanzan a tal velocidad que las actualizaciones, nuevas versiones, contenidos añadidos o descargables, incluso los parches de arreglo de problemas aparecen en un ritmo vertiginoso, existen productos de software, aplicaciones y juegos que publican actualizaciones prácticamente cada semana.

No es la meta de este trabajo elegir una única herramienta de trabajo como mejor que las demás. Por este motivo, no se propone una decisión sino más bien una recomendación de uso en base a lo experimentado durante la realización de este trabajo. Como se explicaba en los objetivos, la meta de este trabajo era explicar el proceso para convertir los diferentes formatos con los que podemos toparnos en nuestra actividad de localización en el estándar más sencillo de trabajar con cualquier herramienta TAO, precisamente para poder liberarnos de las limitaciones de filtro de éstas y elegir la que más nos guste o convenga. Por esta razón podemos concluir que el experimento ha sido un éxito ya que, no solo hemos logrado transformar el formato del archivo a XLIFF si no que no hemos tenido problema al hacer el proceso inverso una vez traducido el juego, y al tiempo hemos podido desarrollar una suerte de guía práctica sobre la transformación de formatos con Rainbow y la alineación de los textos para tener una primera memoria de traducción. Podría decirse que este trabajo serviría, si así lo quisiéramos, de tutorial para usuarios familiarizados con el tema y, con algunas explicaciones más, para cualquier usuario con necesidad de convertir un archivo de localización a formato XLIFF. Somos, pues, conscientes de las dificultades de los temas desarrollados y, con menores restricciones de espacio, podríamos haber creado una auténtica guía práctica para principiantes.

Con respecto a la comparativa entre herramientas, como se puede ver en la tabla, tanto Trados como Transifex tienen ventajas e inconvenientes. En el caso del último, se presenta como más útil para las labores de gestión de proyectos de localización, especialmente la coordinación de equipos de traductores freelance, ya que al ser una herramienta online facilita el acceso al proyecto y los recursos del mismo desde cualquier lugar y a cualquier hora. Además, gracias a la monitorización que lleva a cabo dentro de

los equipos, el gestor o el cliente puede saber exactamente cuánto trabajo ha desarrollado cada integrante (palabras traducidas, revisadas, etc.). Por supuesto, el hecho de que no reconozca las etiquetas hace más complicada la labor de traducción, pero si el recurso a traducir tiene pocas o si el gestor, ingeniero de localización o el cliente facilitan a los traductores la información necesaria para reconocerlas y las instrucciones de no traducirlas, puede resultar la mejor herramienta del localizador.

Por otro lado, las herramientas TAO son sencillas de utilizar pero no facilitan la gestión de proyectos, por esta razón son más recomendables para traductores freelance o para trabajos individuales en los que no se deban compartir los recursos. Esto no quiere decir que no puedan utilizarse para este tipo de trabajos, sino que necesitarían de más pasos intermedios (exportación de MT y copia de archivos) y además los archivos tendrían que dividirse y enviarse a los integrantes del equipo por separado y luego volver a unirse una vez traducidos.

Bibliografía

- aDeSe. 2014. *Balance económico de la industria del videojuego 2013*. 24 de marzo. <http://www.aevi.org.es/web/wp-content/uploads/2015/12/Balance2013.pdf>
- AEVI. 2015. *Balance económico 2014 de la industria en España*. 60-75. http://www.aevi.org.es/web/wp-content/uploads/2015/12/Anuario_AEVI_2014.pdf
- Álvarez-Bolado Sánchez, Carola María y Almoguera Navarrete, Gema (2010). *La localización de videojuegos: un caso de colaboración interdisciplinaria*. En: "XXVIII Congreso Internacional de la Asociación Española de Lingüística Aplicada (AESLA)", 15/04/2010 - 17/04/2010, Vigo, España.
- Anastasiou, Dimitra y Reinhard Schäler. 2010. *Traslating Vital Information: Localisation, Internationalisation and Globalisation*. 28 de enero de 2016. <http://www.d-anastasiou.com/Publications/Syntheses.pdf>
- Android. 2016. *String Resources*. <https://developer.android.com/guide/topics/resources/string-resource.html?hl=es#String>
- Bernal-Merino, Miguel Á. 2006. "On the Translation of Video Games". *JoSTrans: The Journal of Specialised Translation* 6. http://www.jostrans.org/issue06/art_bernal.php
- Bernal-Merino, Miguel Á. 2007. "Challenges in the translation of video games". *Tradumática* 5. <http://www.fti.uab.cat/tradumatica/revista/num5/articles/02/02art.htm>
- Bernal-Merino, Miguel Á. 2008. "What's in a "Game"?". *The International Journal of Localisation* 6: 29-38. https://www.localisation.ie/sites/default/files/resources/locfocus/issues/LocalisationFocusVol6Issue1_online.pdf
- Bernal-Merino, Miguel Á. 2011. "A Brief History of Game Localisation". *TRANS: Revista de Traductología* 15: 11-17. http://www.trans.uma.es/pdf/Trans_15/11-17.pdf
- Chandler, Heather Maxwell y Stephanie O'Malley. 2012. *The Game Localization Handbook*. Estados Unidos: Jones & Bartlett Learning.
- Chandler, Heather Maxwell. 2008. "Practical skills for video game translators". *MultiLingual* 19: 34-37

- Colominas, Carme y Cristina Varga. 2010. *Un módulo de aprendizaje para la localización de videojuegos*. 21 de diciembre de 2015.
https://www.academia.edu/1022519/Un_m%C3%B3dulo_de_aprendizaje_para_la_localizaci%C3%B3n_de_videojuegos
- Corless, Peter. “Mobile Game Localization”. *e2f* (página web). 13 de enero.
<https://e2f.com/5607/>
- Cruz Facundo, Edwin. Huarcaya Quilla, Armando Cristian. Lipa Choque, Diego Frank. 2014. “Sistema Operativo Android”. *Monografías.com*. Julio 2014.
<http://www.monografias.com/trabajos101/sistema-operativo-android/sistema-operativo-android.shtml>
- Czech, Dawid. 2013. “Challenges in video game localization: An integrated perspective”. *Exploration: A Journal of Language and Literature* 1: 3-25
- Dietz, Frank. 2007. “How Difficult can that be? – The Work of Computer and Video Game Localization”. *Tradumática* 5.
<http://www.fti.uab.es/tradumatica/revista/num5/articles/04/04art.htm>
- Donoso, Feliciano. 2002. “Internationalisation”. *Tradumática* 1.
<http://www.fti.uab.es/tradumatica/revista/articles/fdonoso/art.htm>
- Esselink, B. (2003). “The evolution of localization”. *The Guide from Multilingual Computing & Technology: Localization*. 14(5), 4-7.
http://www.intercultural.urv.cat/media/upload/domain_317/arxiu/Technology/Esselink_Evolution.pdf
- Esselink, Bert. 2000. *A practical Guide to Localization*. Amsterdam/Philadelphia: John Benjamins.
- Esselink, Bert. 2002. “The Dream Job?” *Tradumática* 1.
<http://www.fti.uab.es/tradumatica/revista/articles/besselink/art.htm>
- Hernández, David. 2016. “Juegos para móviles generarán más dinero que los de pc y consolas”. *ComputerHoy.com*. 22 de abril.
<http://computerhoy.com/noticias/zona-gaming/juegos-moviles-generaran-mas-dinero-que-pc-consolas-43872>
- Icanlocalize. 2011. *Android Localization Tutorial*. 12 de abril de 2011
- Invarato Menéndez, Ramón. 2014. *Android 100%. Versión 1*.
<http://jarroba.com/libro-android-100-gratis/>

- Ishida, Richard, W3C, Susan K. Miller, Boeing. 2010. “Diferencias entre Localización e Internacionalización”. Traducido por Spanish Translation Team. W3C. 7 de enero. <https://www.w3.org/International/questions/qa-i18n>
- Jiménez-Crespo, Miguel A. 2013. *Translation and Web Localization*. Estados Unidos/Canadá: Routledge: Taylor & Francis Group.
- Lieske, Christian; Sebastian Rhatz y Feliz Sasaki. “Internationalization and Localization of XML: Introducing “ITS””. *Internationalization Tag Set (ITS) Version 2.0*. 7 de diciembre de 2015. <https://www.w3.org/People/fsasaki/docs/xtech06-sasakietal.pdf>
- Lutsuk, Andriy. 2014. “Getting started with Game Localization”. *MultiLingual* 26: 27-30
- Mangiron, Carmen y Minako O’Hagan. 2006. “Game Localisation: Unleashing Imagination with “Restricted” Translation”. *The Journal of Specialised Translation* 6: 10-21
- Méndez González, Ramón. 2015. “La influencia de la localización de videojuegos en la percepción del productor por parte de los usuarios”. *Translation Journal* April 2015. <http://www.translationjournal.net/April-2015/la-influencia-de-la-localizacion-de-videojuegos-en-la-percepcion-del-producto-por-parte-de-los-usuarios.html>
- Mozilla *Internationalization & Localization Guidelines*. 22 de marzo de 2005. <http://www-archive.mozilla.org/docs/refList/i18n/>
- O’Hagan, Minako (2005). *Multidimensional Translation: A Game Plan for Audiovisual Translation in the Age of GILT*. En: “MuTra 2005 – Challenges of Multidimensional Translation: Conference Proceedings”, 76-78. 4 de mayo de 2007. http://www.translationconcepts.org/pdf/MuTra_2005_Proceedings.pdf
- O’Hagan, Minako y Carmen Mangiron. 2013. *Game Localization: Translating for the Global Digital Entertainment Industry*. Amsterdam: John Benjamins.
- O’Hagan, Minako. 2007. “Video games as a new domain for translation research: From translating text to translating experience”. *Tradumática* 5. <http://www.fti.uab.cat/tradumatica/revista/num5/articles/09/09art.htm>
- Oasis. 2008. *XLIFF Version 1.2 Oasis Standard*. 1 de febrero 2008. <http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html>

- Oasis. 2008. *OASIS XML Localisation Interchange File Format TC*.
<https://www.oasis-open.org/committees/xliff/faq.php#WhatIsXLIFF>
- Oasis. 2014. *XLIFF Version 2.0 Oasis Standard*. 5 de agosto de 2014.
<http://docs.oasis-open.org/xliff/xliff-core/v2.0/xliff-core-v2.0.html>
- Okapi Framework. <http://okapi.sourceforge.net/>
- Oliver, Antoni; Joaquim Moré y Salvador Climent. 2008. *Traducción y tecnologías*. Barcelona: UOC
- Raya, Rodolfo. 2004. "XML in localization: A practical analysis; An overview of the most relevant XML standards used in the localization industry". *ibm.com*. 20 de agosto. <http://www.ibm.com/developerworks/library/x-localis/>
- Raya, Rodolfo. 2004. "XML in localization: Use XLIFF to translate documents; XML Localisation Interchange File Format as an intermediate file format". *ibm.com*. 22 de octubre. <http://www.ibm.com/developerworks/library/x-localis2/index.html>
- Samora, Filipe. 2010. "Successfully localizing mobile games". *Tcworld. Magazine for international information management*. 25 de enero de 2016.
<http://www.tcworld.info/e-magazine/translation-and-localization/article/successfully-localizing-mobile-games/>
- Sandrini, Peter (2005). *Website Localization and Translation*. En: "MuTra 2005 – Challenges of Multidimensional Translation: Conference Proceedings", 131-138. 4 de mayo de 2007.
http://www.translationconcepts.org/pdf/MuTra_2005_Proceedings.pdf
- Savourel, Yves. 2014. "An introduction to XLIFF 2.0". *MultiLingual* 25: 42-47.
- Scholand, Michael. 2002. "Localización de videojuegos". *Tradumática* 1.
<http://www.fti.uab.es/tradumatica/revista/articles/mscholand/art.htm>
- *Servicios de internacionalización (i18n)*. 7 de enero. <https://spanish.net-translators.com/internationalization>
- Torres del Rey, Jesús. 2014. "Compte rendu – Book Review". *Paralèlles* 26.
http://www.paralleles.unige.ch/anciens-numeros/numero-26/torresdelrey/Paralleles_26_dec2014_torresdelrey.pdf
- Torres Molina, Yolanda. 2007. "Localización de juegos para móvil". *Tradumática* 5. <http://www.fti.uab.cat/tradumatica/revista/num5/articles/05/05art.htm>

- Transifex documentation. *Getting Started with Transifex*. 3 de febrero.
<http://docs.transifex.com/introduction/>
- Vela Valido, Jennifer. 2005. “La Localización de Videojuegos”. En *Traducción y localización. Mercado, gestión y tecnologías*. Islas Canarias: Anroart Ediciones.

Software utilizado

- APK Tool. Disponible en: <http://ibotpeaches.github.io/Apktool/>
- NotePad++, versión 6.9.1. Disponible en: <https://notepad-plus-plus.org/>
- Rainbow, versión 6.0.29. Disponible en:
<http://okapi.sourceforge.net/downloads.html>
- Transifex, software online. Versión de prueba disponible en:
<https://www.transifex.com/about/>
- SDL Trados, versión 12.0.4809.0. Versión de prueba disponible en:
<http://www.sdl.com/es/cxc/language/translation-productivity/trados-studio/trial.html>

Anexo

Enlaces a los archivos de Google Drive del trabajo:

1. APK original de WitchyWorld:
<https://drive.google.com/open?id=0B44E701b9HpcUnZ6b2lfT1VDVDg>
2. APK descompilada de WitchyWorld:
<https://drive.google.com/open?id=0B44E701b9HpcbldZUUVENXhtUE0>
3. Archivo XLIFF generado con Rainbow del JSON original:
<https://drive.google.com/open?id=0B44E701b9HpcTEhuZWt1Y292WE0>
4. Archivo XLIFF traducido con Trados:
<https://drive.google.com/open?id=0B44E701b9HpcNm94eDlwY196NTQ>
5. Archivo JSON traducido con Transifex:
<https://drive.google.com/open?id=0B44E701b9HpcNDFBVFVSaHZ1YVU>
6. Archivo JSON de vuelta de XLIFF de Rainbow:
<https://drive.google.com/open?id=0B44E701b9HpcbHBpLTVpOEhBc2s>
7. APK descompilada y traducida de WitchyWorld:
<https://drive.google.com/open?id=0B44E701b9HpcVIIBSzNuNUVFYVU>
8. APK firmada y funcional de WitchyWorld:
<https://drive.google.com/open?id=0B44E701b9HpcVW9hWVowUUZCWUE>
9. TMX de Trados con mis traducciones de WitchyWorld y TastyTale:
<https://drive.google.com/open?id=0B44E701b9HpceU16bmJmRzVpSHM>
10. TMX de alineación con Rainbow:
<https://drive.google.com/open?id=0B44E701b9HpcTXE1UVZzdUZvTjQ>
11. TMX de alineación con Trados:
<https://drive.google.com/open?id=0B44E701b9HpcbEVsMkJPZEJyaEk>