Corrección de errores de accesibilidad en una aplicación móvil nativa: Caso de estudio

Francisco J. Estrada-Martinez[1], Juan Aguado-Delgado[1], José R. Hilera[1], Salvador Otón[1], Jose-Maria Gutierrez-Martinez[1]

[1]Departamento de Ciencias de la Computación
Escuela Politécnica Superior - Universidad de Alcalá
28871 Alcalá de Henares (España)
{francisco.estrada, j.aguado}@edu.uah.es, {jose.hilera, salvador.oton, josem.gutierrez}@uah.es

Resumen. La accesibilidad para personas con discapacidad es una característica cada vez más demandada en las TIC. El estudio de su aplicación en la web ha sido el más extendido en el ámbito del software. Con la aparición de nuevas tecnologías como los smartphone, el foco de la accesibilidad debe desplazarse para cubrir estas necesidades en aplicaciones restringidas a cada plataforma. El software no web cobra cada vez más importancia. Para ello, se han adaptado estándares y legislaciones, y se han elaborado algunas guías de ayuda a los desarrolladores. Este trabajo estudia un caso de ejemplo presentado como parte de la Guía de accesibilidad de aplicaciones móviles, publicado por el Ministerio de Hacienda y Administraciones Públicas de España.

Palabras clave: Accesibilidad, Android, Aplicaciones móviles nativas, iOS.

1. Introducción

La accesibilidad es una característica de los productos, servicios, infraestructuras y sistemas que permite que puedan ser usados por personas con cualquier rango de capacidades. Se compone de una combinación de diseño universal y asistencia personal. Esta última puede ser proporcionada por una tercera persona o una herramienta de apoyo [1].

Los fabricantes y proveedores de productos TIC deben tener en cuenta el diseño universal para que sus productos puedan ser utilizados por personas con discapacidad. Para ello se han creado diversos estándares que afectan a diferentes clases de productos. El más representativo son las Web Content Accessibility Guidelines (WCAG) [2], reflejadas en la norma ISO 40500 [3], que se refieren a los requisitos que deben cumplir las aplicaciones web. No obstante, existen otros estándares como las Authoring Tools Accessibility Guidelines (ATAG) [4], User Agent Accessibility Guidelines (UAAG) [5], la norma europea EN 301 549 [6] o la Section 508 of the Rehabilitation Act [7] estadounidense. Estos estándares pueden ser tomados o adaptados por las legislaciones de cada región para que su cumplimiento sea obligatorio.

La norma EN 301 549 surge en el ámbito de la Unión Europea y abarca un abanico muy amplio de las TIC, tanto software como hardware. Su aplicación será progresiva y, por el momento, solo afectará a las administraciones públicas. Sin embargo, cada

estado miembro es libre de extender este cumplimiento como le parezca oportuno, tomando siempre como mínimo las administraciones públicas. En el caso de España, ya se ha publicado un real decreto que exige su cumplimiento para garantizar la accesibilidad de sitios web y de aplicaciones para dispositivos móviles [8].

Las aplicaciones móviles (apps) han cobrado cada vez más relevancia. Actualmente, se estima que los ciudadanos acceden a internet en su mayor parte a través de un smartphone. Las administraciones públicas, igual que el resto de entidades, han desarrollado sus propias apps para ofrecer un gran número de servicios al ciudadano. No obstante, en muchas de ellas no se han tenido en cuenta los requisitos de accesibilidad.

Este panorama hizo necesaria la elaboración de una Guía de accesibilidad de aplicaciones móviles para funcionarios públicos y desarrolladores [9]. El objetivo es que tanto los desarrollos propios como las aplicaciones compradas a terceros cumplan con los requisitos exigidos por la nueva legislación. La guía fue elaborada en la Universidad de Alcalá y publicada por el Ministerio de Hacienda y Administraciones Públicas de España.

Acompañando a la guía se han preparado ejemplos prácticos de una aplicación móvil accesible corrigiendo posibles errores. La app de ejemplo se ha desarrollado para las dos principales plataformas móviles: Android e iOS. El código fuente está disponible en dos repositorios de GitHub, en ambos casos cada repositorio está dividido en dos ramas: “master”, con fallos de accesibilidad, y “accesible”, con dichos fallos corregidos:

- Código Android: https://github.com/ctt-gob-es/Ejemplo-App-Accesible-Android
- Código iOS: https://github.com/ctt-gob-es/Ejemplo-App-Accesible-iOS.

En las figuras 1 y 2 se muestra una copia de pantalla de la app para Android, tanto de la versión original no accesible (Fig. 1), como la versión corregida accesible (Fig.2).

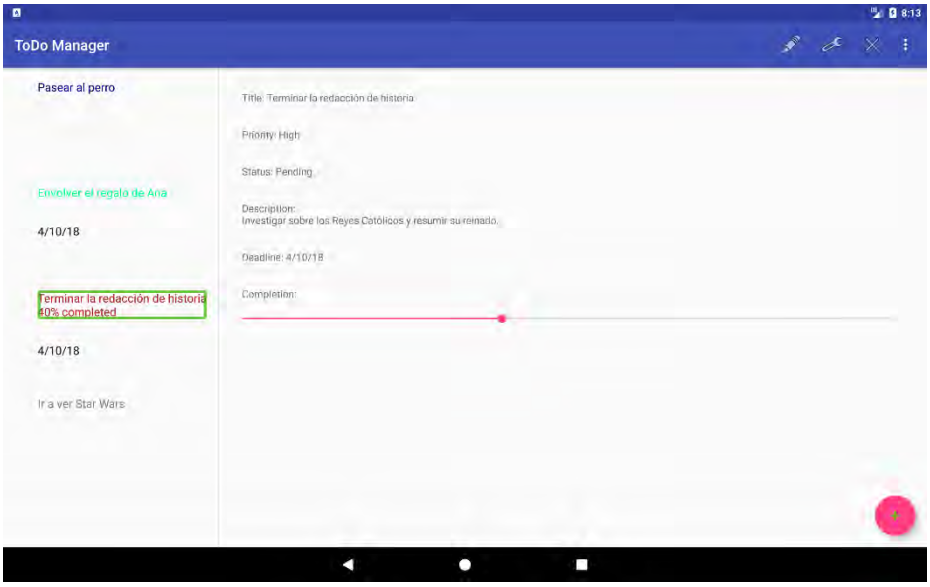

Fig. 1. App de ejemplo desarrollada para Android (Versión original no accesible)

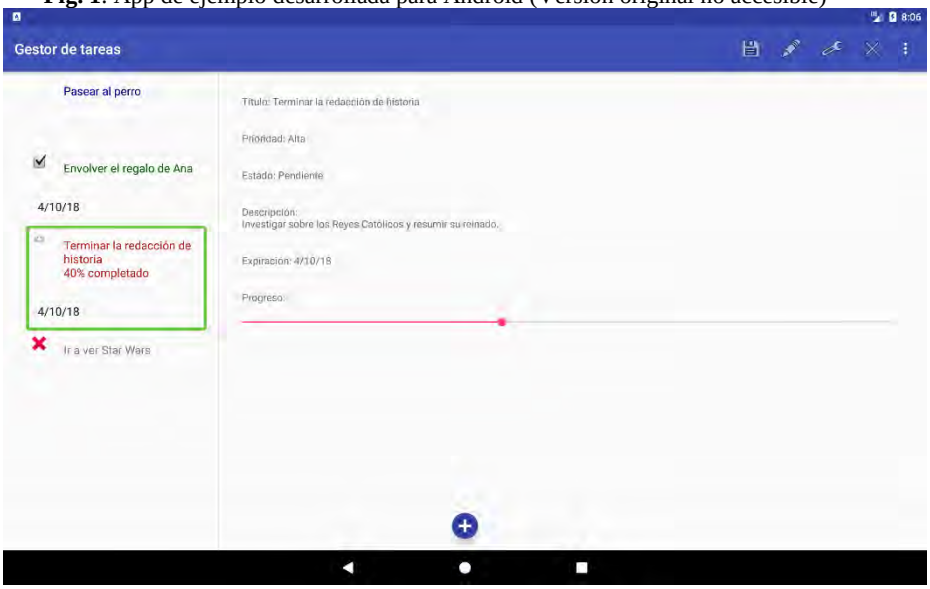

Fig. 2. App de ejemplo desarrollada para Android (Versión corregida accesible).

El objetivo de este trabajo es enumerar los fallos de accesibilidad que presenta la aplicación de ejemplo original y detallar cómo se han solventado en cada caso. Para ello se tomará como referencia el código de ejemplo para Android. En la sección 2 se listan los errores de accesibilidad incluidos, dónde se encuentran y dónde se hallan las soluciones en la versión accesible. En la sección 3 se detalla uno de los errores de accesibilidad y su solución como ejemplo. En la sección 4 se discuten algunos aspectos a tener en cuenta extraídos de la elaboración de los ejemplos. Por último, en la sección 5 se exponen las conclusiones y posibles trabajos futuros.

2. Errores de accesibilidad

El ejemplo para la guía proporciona una versión no accesible y otra accesible de la misma aplicación. La
Tabla 1 muestra la ubicación de estos errores en el código de la versión no accesible y su solución en la accesible. Los códigos de los criterios indicados provienen de la norma EN 301 549 [6].
La detección de los errores de accesibilidad se ha sustentado en las herramientas propias del IDE Android Studio, herramientas de análisis de contraste y una revisión manual utilizando lectores de pantalla. Estos métodos pueden encontrarse con más detalle en la guía para la que se ha elaborado el ejemplo [9]. A continuación, se describen los errores encontrados y la solución adoptada:

1. Criterio 11.2.1.1 – Contenido no textual: Los botones de completar y cancelar tarea no tienen texto alternativo. Se agrega.

2. Criterio 11.2.1.1 – Contenido no textual: El botón de nueva tarea no tiene texto alternativo. Se agrega.
3. Criterio 11.2.1.10 – Uso del color: La prioridad y estado de las tareas se identifica mediante color. Se introduce un icono para cada prioridad y estado, y se le da un texto alternativo.
4. Criterio 11.2.1.12 – Contraste: El contraste de algunas prioridades y estados no es adecuado. Se modifica la paleta de color.
5. Criterio 11.2.1.12 – Contraste: El contraste del botón de nueva tarea no es adecuado. Se modifica el atributo de estilo.
6. Criterio 11.2.1.22 – Orden del foco: El botón de nueva tarea se alcanza después de pasar por toda la lista. Se modifica el orden para que se alcance antes de entrar en la lista.
7. Criterio 11.2.1.27 – Lenguaje del software: La aplicación está en inglés. Puesto que el ámbito de uso es el español, se añade la localización al castellano con el recurso strings.xml
8. Criterio 11.2.1.27 – Lenguaje del software: La ayuda de la aplicación está en inglés. Se crea un archivo HTML localizado al castellano y se añade soporte de localización en la carga.
9. Criterio 111.2.1.25 – Encabezado y etiquetas: Los encabezados de las secciones de la ayuda no están correctamente marcados. Se marcan con las etiquetas adecuadas.
10. Criterio 11.2.1.33, 11.2.1.35: En el formulario de contacto no se identifican ni se sugieren correcciones para los errores. Se añade un diálogo para identificar errores y sugerir correcciones.
11. Criterio 11.2.1.33, 11.2.1.35: En el formulario de edición/nueva tarea no se identifican ni se sugieren correcciones para los errores. Se añade un diálogo para identificar errores y sugerir correcciones.
12. Criterio 11.2.1.36 – Prevención de errores: En el formulario de contacto no existe ningún tipo de prevención de errores. Se añaden los atributos necesarios a los campos para ayudar en la inserción de texto.
13. Criterio 11.2.1.36 – Prevención de errores: En el formulario de edición/nueva tarea no existe ningún tipo de prevención de errores. Se añaden los atributos necesarios a los campos para ayudar en la inserción de texto.

Tabla 1. Listado de errores de accesibilidad corregidos y sus soluciones.

Id	Criterio	Localización	Corrección
1	11.2.1.1	task_list_content.xml: line 30-42	task_list_content.xml: line 44-59
2	11.2.1.1	activity_task_list.xml: line 30-40	activity_task_list.xml: line 34-45
3	11.2.1.10	TaskListActivity.java: line 168-183	TaskListActivity.java: line 169-201 task_list_content.xml: line 12-19
4	11.2.1.12	colors.xml: line 8-9	colors.xml: line 8-9
5	11.2.1.12	activity_task_list.xml: line 30-40	activity_task_list.xml: line 42-43
6	11.2.1.22	task_list.xml: line 2-13	task_list.xml: line 14
7	11.2.1.27	values/strings.xml	values-es/strings.xml values-en/strings.xml
8	11.2.1.27	help.html https://github.com/ctt-gob-es/Ejemplo-App-Accesible-	help-en.html help-es.html HelpActivity: line 25-26

		Android/blob/e082b742bcc3d06f046cc9453ed7c76af3690ee8/app/src/main/assets/html/help-en.html HelpActivity.java: line 18	values-es/strings.xml: line 95 values-en/strings.xml: line 95
9	11.2.1.25	help.html: line 10 help.html: line 16 help.html: line 38 help.html: line 49	help.html: line 10 help.html: line 16 help.html: line 38 help.html: line 49
10	11.2.1.33 11.2.1.35	ContactActivityFragment.java: line 38-42	ContactActivityFragment.java: line 25 ContactActivityFragment.java: line 34 ContactActivityFragment.java: line 46-57 errorContactMessageDialog.java
11	11.2.1.33 11.2.1.35	EditTask1Fragment.java: line 185-188 EditTask1Fragment.java: line 202-205	EditTask1Fragment.java: line 185-193 EditTask1Fragment.java: line 207-214 ErrorTaskDialog.java
12	11.2.1.36	fragment_contact.xml: line 16-20 fragment_contact.xml: line 27-31 fragment_contact.xml: line 38-43	fragment_contact.xml: line 21 fragment_contact.xml: line 33 fragment_contact.xml: line 46
13	11.2.1.36	fragment_edit_task1.xml: line 15-21 fragment_edit_task1.xml: line 43-49	fragment_edit_task1.xml: line 22 fragment_edit_task1.xml: line 51

3. Ejemplo de solución

En este apartado se detalla la solución ofrecida para el error de accesibilidad número 6. Este error se refiere al orden del foco. Para que la aplicación sea accesible, el foco debería visitar antes el botón flotante de añadir tarea que la lista de tareas. El **Código 1** muestra la declaración que debe ser modificada. El
Código 2 muestra el atributo que se inserta en la versión accesible para corregir el error.

Código 1. Fragmento de task_list.xml: líneas 2-13

```
<android.support.v7.widget.RecyclerView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/task_list"
    android:name="cc.uah.es.todomanager.TaskListFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    app:layoutManager="LinearLayoutManager"
    tools:context="es.uah.cc.todomanager.TaskListActivity"
    tools:listitem="@layout/task_list_content" />
```

Código 2. Fragmento de task_list.xml: línea 14

```
android:accessibilityTraversalAfter="@id/fab"/>
```

4. Discusión

La elaboración de estos ejemplos nos ha servido para ilustrar cómo pueden hacerse accesibles aplicaciones típicas que no lo son. Sin embargo, también nos da una idea del esfuerzo necesario que hay que invertir para lograrlo. En muchos casos, los cambios se reducen a la inserción o modificación de una o unas pocas líneas de código. En los casos más extremos, los cambios han conllevado un trabajo aproximado de una hora. Estos cambios, sin embargo, son poco numerosos. De hecho, al tratarse de cambios tan pequeños en su mayor parte, podemos afirmar que el esfuerzo se minimiza hasta casi ser nulo si los requisitos se tienen en cuenta desde las fases iniciales del desarrollo. Esto es debido a que lleva más tiempo buscar dónde se deben realizar los cambios una vez terminada la aplicación que introducir los propios cambios.

Por último, podemos decir que cumplir con algunos requisitos de accesibilidad, como la prevención o detección de errores, ayudan a crear una experiencia más agradable para cualquier tipo de usuario. La accesibilidad, por tanto, redunda en una mejor calidad del producto final.

5. Conclusiones

Como se puede apreciar en los ejemplos, hacer que una app típica sea accesible no requiere un gran esfuerzo. La mayoría de los errores pueden corregirse añadiendo o modificando unas pocas líneas de código. Si se tiene en cuenta en las fases iniciales del desarrollo, el coste incluso puede ser despreciable. Además, el resultado gozará de una mayor calidad.

Los errores más típicos en este tipo de aplicaciones son la ausencia de texto alternativo en imágenes y botones, un orden del foco inadecuado, un contraste insuficiente y el uso del color para transmitir información. Los elementos interactivos también adolecen con frecuencia de falta de etiquetado, de ayudas a la entrada, ausencia o insuficiencia de la detección y soporte para corrección de errores.

Como trabajo futuro queda la adaptación de los ejemplos para la inclusión de criterios de las WCAG 2.1 [2] en la EN 301 549 [6]. También queda pendiente la elaboración de ejemplos para otras plataformas móviles secundarias como Universal Windows Platform.

6. Agradecimientos

Este estudio ha sido financiado con el apoyo de la Comunidad de Madrid a través del Sistema Regional de Ciencia y Tecnología, Investigación y Desarrollo, Información y Promoción Tecnológica madri+d.

7. Referencias

1. ISO 26800:2011 Ergonomics -- General approach, principles and concepts. International Organization for Standardization, 2011.
2. Kirkpatrick, A., O'Connor, J., Campbell, A., Cooper, M. (eds.) (2018) Web Content Accessibility Guidelines (WCAG) 2.1. W3C Recommendation. World Wide Web Consortium. https://www.w3.org/TR/WCAG21/. (Consultado el 1 de octubre de 2018).
3. ISO/IEC 40500:2012 (W3C) Information technology -- W3C Web Content Accessibility Guidelines (WCAG) 2.0. International Organization for Standardization, 2012.
4. Richards, J., Spellman, J., Treviranus, J. (eds.) (2015) Authoring Tool Accessibility Guidelines (ATAG) 2.0. W3C Recommendation. World Wide Web Consortium. https://www.w3.org/TR/ATAG20/. (Consultado el 1 de octubre de 2018).
5. Allan, J., Lowney, G., Patch, K., Spellman, J. (eds.) (2015) User Agent Accessibility Guidelines (UAAG) 2.0. W3C Working Group Note. World Wide Web Consortium. https://www.w3.org/TR/UAAG20/. (Consultado el 1 de octubre de 2018).
6. UNE-EN 301549 V1.1.2:2015 Requisitos de accesibilidad de productos y servicios TIC aplicables a la contratación pública en Europa. UNE, 2015.
7. About the ICT Refresh. United States Access Board, 2017. https://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-ict-refresh. (Consultado el 1 de octubre de 2018).
8. Real Decreto 1112/2018, de 7 de septiembre, sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público. Boletín Oficial del Estado, 19 de septiembre de 2018, 90533-90549. https://www.boe.es/boe/dias/2018/09/19/pdfs/BOE-A-2018-12699.pdf.
9. Aguado-Delgado, J., Estrada-Martínez, F.J. (2017). Guía de accesibilidad de aplicaciones móviles (APPS). Ministerio de Hacienda y Función Pública de España. https://administracionelectronica.gob.es/pae_Home/pae_Estrategias/pae_Accesibilidad/pae_documentacion/pae_eInclusion_Accesibilidad_de_apps.html. (Consultado el 1 de octubre de 2018).