

Accesibilidad de aplicaciones móviles para discapacitados visuales: problemas y estrategias de solución

Pablo Santana-Mansilla^{1,2}, Germán Lescano^{1,2}, Rosanna Costaguta¹

¹Instituto de Investigación en Informática y Sistemas de Información (IIISI)
Facultad de Ciencias Exactas y Tecnologías (FCEyT)
Universidad Nacional de Santiago del Estero (UNSE)

²Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
{psantana, gelescano, rosanna}@unse.edu.ar

Resumen. La popularidad de las tecnologías móviles continúa en crecimiento a medida que mejoran su fidelidad, incrementan su capacidad de procesamiento y reducen sus costos. Sin embargo, la tendencia creciente de reemplazar teclados físicos con pantallas táctiles no solo ha provocado cambios en las modalidades de interacción sino que, para los discapacitados visuales ha significado un incrementado de la complejidad y la dificultad de utilización de aplicaciones móviles. Dado que los desarrolladores de software son en parte responsables de garantizar la accesibilidad de las aplicaciones móviles, en este documento se describen las barreras de accesibilidad que típicamente enfrentan los discapacitados visuales al utilizar aplicaciones de software mediante dispositivos móviles. Para cada problema de accesibilidad se indican posibles estrategias de solución y se describen aplicaciones móviles implementadas con software libre para aquellos problemas cuya solución va más allá de la aplicación de buenas prácticas de diseño.

Palabras Claves: Discapacidad visual, aplicaciones móviles, problemas de accesibilidad, estrategias de solución, software libre.

1. Introducción

La popularidad de las tecnologías móviles continúa en crecimiento a medida que mejoran su fidelidad, incrementan su capacidad de procesamiento y reducen sus costos. Con el desarrollo de *smartphones* los dispositivos móviles no solo proveen funcionalidades (editores de texto, hoja de cálculo, e-mail, y navegadores web) comparables a las de una PC sino que también se ha producido un cambio en las modalidades de interacción debido a la tendencia creciente de emplear pantallas táctiles y descartar el uso de teclas de hardware [4,11,20].

Si bien las tecnologías móviles ofrecen los beneficios de ubicuidad y portabilidad, la ausencia de teclas de hardware provoca que la interacción con dispositivos táctiles sea más difícil y compleja para los usuarios con discapacidad visual (no videntes, con visión reducida, o daltónicos) [5]. La complejidad de las interfaces de usuario sumada al hecho de que las pantallas táctiles son completamente lisas y no proveen

distinciones entre controles y espacio de visualización que se puedan percibir mediante el tacto, provocan que la detección de elementos de la Interfaz de Usuario (IU) sea sumamente difícil para personas que no pueden ver la pantalla y los controles de las aplicaciones móviles [2,4,23]. En consecuencia, determinar el comienzo de un mensaje cuando se lee un e-mail, encontrar el área de resultados de un motor de búsqueda, o leer el contenido de un campo de edición mientras se completa un formulario, es muy difícil para los no videntes y se requiere que alguien les muestre la localización de los objetos en pantalla o que se utilice algún tipo de tecnología de asistencia [2,11,8]. Esta situación puede provocar en las personas con problemas de visión desde frustración y temor a cometer errores, hasta resistencia a usar dispositivos móviles con pantallas táctiles [11,12,8].

Las tecnologías de asistencia usadas para traducir el contenido visual de las pantallas táctiles en un formato accesible a no videntes se basan principalmente en el canal auditivo, a través de la conversión de texto a palabra (*Text-to-Speech* o TTS) [4]. Sin embargo, esto no siempre es suficiente para una interacción completamente accesible ya que el feedback mediante voz puede no funcionar bien en ambientes ruidosos y en algunas situaciones (por ejemplo durante clases, reuniones, o discursos) los usuarios pueden preferir algo más silencioso [4]. Si además se considera que la mayoría de las aplicaciones para dispositivos móviles se diseñan para personas que pueden ver, las características de accesibilidad no siempre son adecuadas para obtener un resultado fiable [21]. Esto significa que la tecnología de asistencia no resuelve los problemas de accesibilidad por su propia cuenta y funciona satisfactoriamente en la medida que las aplicaciones móviles estén bien diseñadas [7].

En [10] se estima que la principal causa de las barreras de accesibilidad que afectan a los discapacitados visuales es que los fabricantes de hardware y desarrolladores de software usualmente marginan a este grupo de usuarios. Como consecuencia a los usuarios no videntes se les brindan (mayormente de forma retrasada) solo herramientas accesibles (tales como lectores de pantalla) que adaptan los métodos para operar las interfaces gráficas táctiles en lugar de proveer otros métodos apropiados. La actitud de los diseñadores y desarrolladores de software es hasta cierto punto comprensible ya que al ser en su mayoría personas que no sufren de problemas de visión, tienen un entendimiento limitado de como las personas no videntes experimentan la tecnología [9].

Considerando que los desarrolladores son en parte responsables de la accesibilidad de las aplicaciones móviles, y que por lo general no conocen las dificultades por las que atraviesan los usuarios con discapacidad visual, en este documento se hace una revisión de las barreras de accesibilidad que enfrentan día a día los discapacitados visuales así como también de las estrategias de solución que podrían implementarse mediante software libre. Las cuestiones abordadas en este documento podrían ser de utilidad tanto para programadores interesados en desarrollar aplicaciones accesibles o en mejorar la accesibilidad de aplicaciones existentes, así como también para organizaciones que buscan cumplir con su responsabilidad legal, mejorar su imagen corporativa y expandir su base de consumidores [23].

El presente trabajo se organiza de la siguiente manera: en primer lugar se abordan los conceptos de accesibilidad, barrera de accesibilidad, tecnología de asistencia y

consecuencias de las barreras de accesibilidad (sección 2). A continuación, en la sección 3, se brinda una descripción de las barreras de accesibilidad que típicamente encuentran los discapacitados visuales al utilizar aplicaciones móviles. En la sección 4 se plantean estrategias que utilizando software libre pueden contribuir a solucionar algunas de las barreras de accesibilidad que enfrentan los discapacitados visuales. Finalmente, la sección 5 contiene algunas conclusiones.

2. ¿Qué es una barrera de accesibilidad?

El término accesibilidad se usa para describir el grado hasta el cual un producto, dispositivo, servicio o ambiente está disponible para tantas personas como sea posible [21]. Los siguientes cuatro principios tomados del área de accesibilidad web son el fundamento necesario para que una persona tenga acceso a las aplicaciones móviles:

- **Perceptible:** la información y los componentes de la IU deben presentarse a los usuarios en manera que los puedan percibir [24].
- **Operable:** los componentes de la IU y la navegación se deben poder utilizar [24].
- **Entendible:** la información y operación de la IU deben ser comprensibles [24].
- **Robusto:** el contenido debe ser lo suficientemente robusto para que pueda ser interpretado de manera confiable por una amplia variedad de agentes de usuario [24]. El término agentes de usuario hace referencia a las herramientas utilizadas por los usuarios finales para visualizar e interactuar con el contenido incluyendo a los navegadores web, reproductores multimedia, y tecnologías de asistencia [25].

Tomando como base los cuatro principios antes mencionados se puede decir que una aplicación móvil es accesible para una persona con discapacidad visual si le permite percibir, entender, y navegar tanto el contenido como los componentes de la IU con independencia de la tecnología de asistencia que utilice.

Para [25] las evaluaciones de accesibilidad deben hacerse situando al usuario en un contexto determinado, por lo cual una barrera o problema de accesibilidad puede ser entendido como cualquier condición que hace difícil para el usuario (en este caso una persona con discapacidad visual) la consecución de un objetivo cuando usa la aplicación móvil en un contexto específico.

2.1. Tecnología de asistencia

Se estima que en el mundo existen al menos 6 millones de personas con algún tipo de discapacidad visual que les impide utilizar pantallas de computadoras tradicionales [12]. Muchos de los usuarios con discapacidad visual utilizan tecnologías de asistencia, un término usado para hacer referencia a hardware y software diseñado para facilitar el uso de computadoras por parte de personas con discapacidades [7]. La tecnología de asistencia para discapacitados visuales incluye dispositivos tales como:

- **Lector de pantalla:** es un software que traduce y convierte la información mostrada en pantalla en voz sintetizada por computadora, Braille para una pantalla Braille, y sonidos [16]. Los lectores de pantalla con características basadas en gestos

permiten a las personas con discapacidad visual escuchar descripciones de las funciones en sus dispositivos y operar las funciones mediante el tacto y gestos [23].

- Magnificador de pantalla: es una aplicación de software usada para aumentar el tamaño del texto o las imágenes en una pantalla [18,23].
- Tamaños de fuente ajustables: algunos dispositivos móviles tienen incorporada esta característica para agrandar el tamaño de la fuente de las aplicaciones [23].
- Reconocimiento de voz: una aplicación de software que combinada con un dispositivo de entrada de voz (usualmente un micrófono incorporado o separado) permite a los usuarios completar una función o control del dispositivo móvil a través de comandos de voz [17,23].
- Controles de contraste/brillo: permiten modificar el color de primer plano/ fondo de la pantalla y alterar el brillo para satisfacer las necesidades individuales [23].
- Pantalla Braille: dispositivo de hardware que lee, traduce y subsecuentemente provee al usuario en formato Braille información electrónica desde una computadora [17].

Se debe tener en cuenta que estas tecnologías trabajan de manera satisfactoria en la medida en que las aplicaciones móviles están bien diseñadas [7].

2.2. Consecuencias de la existencia de barreras de accesibilidad

Entre las consecuencias que tienen aparejadas las barreras de accesibilidad a las que se enfrentan las personas con discapacidad visual al usar dispositivos móviles táctiles se pueden mencionar las siguientes:

- Transportar un dispositivo secundario: según lo reportado [11] muchos usuarios no videntes llevan consigo una laptop además del celular. Esto se debe a que sienten que no pueden confiar en su dispositivo móvil y lector de pantalla para realizar tareas basadas en palabra, como ser el inspeccionar y escribir documentos. Aunque estas personas reconocen que es una carga transportar un dispositivo secundario con ellos, este les provee un respaldo efectivo ante la imposibilidad de acceder al contenido usando un lector de pantalla móvil.
- Frustración: las múltiples dificultades que enfrentan los usuarios con discapacidad visual durante la utilización de un dispositivo móvil no solo producen enojo sino también pérdida de tiempo [12]. En algunos casos, la frustración de los usuarios con la tecnología puede incrementar la tensión muscular y la presión arterial [12]. Incluso existe un impacto potencial en el humor y las relaciones con otros [12].
- Temor a cometer errores: de acuerdo con [11] la ansiedad subyacente a cometer un error disuade a muchos discapacitados visuales a entrenarse en el uso de dispositivos táctiles.
- Resistencia a usar pantallas táctiles: para los usuarios no videntes que formaron parte de la investigación conducida por [8] la incerteza involucrada en el uso de dispositivos táctiles, debido a la falta de un referente tangible, es el motivo por el cual las pantallas táctiles nunca serán realmente accesibles. Si este prejuicio no se combate o trata apropiadamente, no importa cuánto esfuerzo se destine a eliminar

los problemas de accesibilidad o si se resuelven totalmente porque, los discapacitados visuales se van a seguir negando a emplear los dispositivos móviles táctiles.

Dado que la falta de confianza, la frustración, el temor a equivocarse y la resistencia usar dispositivos móviles táctiles pueden ser tratados o al menos aliviados, es importante conocer cuáles son los problemas de accesibilidad que aquejan a los usuarios con discapacidad visual, al igual que las estrategias o técnicas que pueden emplearse para resolverlos.

3. Problemas de accesibilidad para usuarios no videntes

En [10] estiman que la principal causa de las barreras de accesibilidad que afectan a los discapacitados visuales es que los fabricantes de hardware y desarrolladores de software usualmente marginan a este grupo de usuarios. Como consecuencia a los usuarios no videntes se les brindan (mayormente de forma retrasada) solo herramientas accesibles (tales como lectores de pantalla) que adaptan los métodos para operar las interfaces gráficas táctiles en lugar de proveer otros apropiados. La actitud de los diseñadores y desarrolladores de software es hasta cierto punto comprensible ya que en su mayoría son personas sin problemas de visión y por consiguiente tienen un entendimiento limitado de como las personas no videntes experimentan la tecnología [9]. Por consiguiente, es de esperar que la accesibilidad de las interfaces de los dispositivos móviles táctiles se pueda mejorar sustancialmente cuando los desarrolladores y diseñadores comprendan de qué forma los discapacitados visuales utilizan las pantallas táctiles y las dificultades que tienen que enfrentar [9].

Los problemas de accesibilidad típicos a los cuales se enfrentan los discapacitados visuales están relacionados con: enlaces, imágenes, estructuración de las páginas, tablas, uso de color, videos, formularios, navegación y empleo de zoom [2, 3, 4, 5, 11, 12, 19, 23, 24, 25]. Problemas ligeramente sencillos de resolver siguiendo buenas prácticas de diseño y programación, como ser: uso de CSS (siglas en inglés de *cascading style sheets*); empleo de unidades relativas para que la IU se ajuste a diversos tamaños de pantalla [25]; enlaces claros, descriptivos, concisos y canónicos [23, 25]; sustitución de imágenes de texto por su representación textual [23]; asociación de descripción textual a las imágenes y videos [3, 23, 25]; uso de etiquetas de encabezado [23, 25], entre otros. Sin embargo, existen otros problemas de accesibilidad vinculados con cuestiones tales como síntesis de voz, uso de mapas, ausencia de teclado físico y reconocimiento de voz que requieren de mayores esfuerzos para resolverlos ya que están más relacionados con el software del sistema.

3.1. Problemas de accesibilidad con síntesis de voz

Para las personas no videntes el acceso al contenido de la IU de las aplicaciones móviles depende considerablemente del uso de mecanismos de síntesis de voz. Si se pretende evitar que la realimentación basada en voz le cause dificultades a los no

videntes al momento de reconocer cambios en el estado y los eventos de las aplicaciones es preciso que se tengan en cuenta los dos siguientes factores:

- Ruido del ambiente: en ambientes ruidosos (como ser un shopping o el transporte público) el escuchar la realimentación por voz provista por los dispositivos móviles o el software de lectura de pantalla puede ser difícil y hasta imposible [2,11,13].
- Interferencia con sonidos generados por aplicaciones: para los usuarios es irritante que se produzca interferencia entre los sonidos generados por el mecanismo de síntesis de voz y los sonidos de otras aplicaciones móviles [10,15].

De manera adicional a la interferencia que pueden ejercer el ruido ambiental y otras aplicaciones sobre el mecanismo de síntesis de voz, debe considerarse la protección de la privacidad ya que en lugares públicos las personas que rodean al usuario con deficiencia visual también escuchan (de manera intencional o no) los sonidos producidos por el dispositivo móvil [11].

3.2. Problemas de accesibilidad con el uso de mapas

Las aplicaciones de mapas para los usuarios con discapacidad visual se basan principalmente en la conversión de texto a palabra, donde la lista de direcciones o ítems se enumeran para los usuarios. Desafortunadamente no existe una buena correspondencia entre TTS y la exploración geométrica y de mapas debido a la dificultad de producir descripciones claras y breves de tanto formas complejas como de las posiciones de los objetos [22].

Una segunda limitación de las aplicaciones de mapas existentes es que soportan principalmente ambientes exteriores públicos [22]. No existen sistemas de localización para ambientes interiores (tales como oficinas o centros comerciales) que ayuden en la navegación o búsqueda [22]. Por consiguiente los ambientes interiores son difíciles de explorar para las personas con discapacidad visual.

3.3. Problemas de accesibilidad con el teclado

Las teclas físicas al igual que todos los elementos claramente palpables con los dedos (por ejemplo botones, puntos, bordes, etc.) son cruciales para las personas con discapacidad visual. Uno de los principales problemas encontrados por los usuarios no videntes en los dispositivos móviles con pantallas táctiles se relaciona con la ausencia de teclas físicas que sirvan como puntos de referencia [5,10,13].

El uso del teclado numérico en pantallas táctiles es un asunto angustiante para los no videntes porque están acostumbrados a usar la tecla número "5" de los teclados físicos para orientarse al momento de editar los números o realizar llamadas [13]. En la Figura 1 se muestra un ejemplo de celular con teclado físico donde la tecla "5" se señaló con dos puntos sobre relieve para distinguirla del resto. En el teclado de la Figura 1 una vez que el usuario no vidente detecta la tecla "5" puede marcar con facilidad un número de teléfono porque sabe que: si se desplaza una posición para arriba

encuentra la tecla “2”, si se desplaza una posición a la derecha encuentra la tecla “6”, si se desplaza una posición en forma diagonal hacia abajo encuentra la tecla “7”, etc.



Figura 1. Teclado físico con la tecla “5” señalada mediante dos puntos sobre relieve.

La falta de referencias físicas para localizar y seleccionar ítems visualizados en pantalla genera obstáculos a las personas no videntes cuando utilizan un teclado virtual del tipo QWERTY. Una queja común es la dificultad para escribir signos de puntuación (tales como: . , ?!) y otros caracteres especiales puesto que para usarlos es preciso cambiar el modo de operación del teclado QWERTY [10,13].

En cuanto a la escritura en movimiento, mientras es factible que un no vidente pueda emplear un teléfono con teclas físicas a medida que se desplaza, esto no siempre es posible en los teléfonos con pantalla táctil. En situaciones en movimiento se necesitaría de la interacción con las dos manos (una para sostener el teléfono y la otra para hacer los gestos) pero dado que la mayoría de los usuarios no videntes tiene una mano ocupada usando un bastón, es difícil que puedan mantenerse deambulando mientras usan el teléfono y al mismo tiempo perciben señales del ambiente que los rodea (por ejemplo obstáculos en la calle) [11].

3.4. Problemas de accesibilidad con reconocimiento de voz

Las personas no videntes suelen utilizar sistemas de reconocimiento automático de voz (*Automatic Speech Recognition- ASR*) para introducir información con mayor frecuencia y para mensajes más largos en relación a personas sin problemas de visión [1]. Los no videntes en general suelen estar más satisfechos que las personas videntes con el uso de sistemas ASR ya que la introducción de información mediante la voz puede ser hasta 5 veces más rápida que el uso de teclados virtuales en pantalla [1].

Mientras que para hablarle a un sistema ASR una persona no necesita ver, el proceso de dictado, revisión y edición del texto es complejo y requiere de operaciones de entrada adicionales. Tanto la revisión como la edición pueden ser desafiantes para los usuarios no videntes ya que al no poder leer en pantalla la salida del reconocedor de voz (como lo haría normalmente una persona vidente), tienen que basarse en

lectores de pantalla [1]. El reconocimiento de errores en base la salida de audio del lector de pantalla puede ser difícil puesto que, es probable que el reconocedor de voz produzca como salida palabras que suenen similares a las que el usuario pronunció, pero que en realidad sean incorrectas.

Una vez identificados los errores de reconocimiento, la corrección de los mismos puede no ser fácil para los usuarios no videntes. Mientras las personas sin ningún tipo de impedimento visual gastan alrededor del 66% de su tiempo corrigiendo errores de las salidas de ASR en sistemas de escritorio de dictado de texto [9], en un estudio realizado por [1] se comprobó que las personas con impedimentos visuales invierten en promedio el 80% del tiempo de composición para revisar y editar el texto.

Por último, la privacidad y pertinencia social suele ser una preocupación para muchos usuarios no videntes cuando usan un sistema de ASR en un contexto público (un tren, un colectivo, la calle, etc.) puesto que las otras personas pueden escuchar lo que dicen cuando le hablan a su dispositivo [1,5]. El contexto no solo afecta la privacidad de los usuarios sino que el ruido ambiental puede degradar drásticamente el desempeño del sistema de ASR provocando que el texto generado sea erróneo [6].

4. Soluciones a problemas de accesibilidad de discapacitados visuales en dispositivos móviles

En esta sección se describen estrategias que pueden implementarse empleando software libre para resolver problemas de accesibilidad en dispositivos móviles vinculados con síntesis de voz, uso de mapas, uso del teclado y reconocimiento de voz.

4.1. Solución a problemas de accesibilidad con síntesis de voz

Cuando a los usuarios con discapacidad visual se les suministra información mediante síntesis de voz se debe prestar atención al efecto distorsivo de los sonidos emitidos por otras aplicaciones móviles y el ruido ambiental. En primer lugar, el sonido de fondo generado por las aplicaciones móviles puede ser molesto para los usuarios no videntes cuando interfiere con el mecanismo de síntesis de voz por lo cual, debería haber una función de navegación para desactivar la reproducción automática del audio o bien se debería proveer una función de configuración para que el usuario controle el audio [15, 23]. En segundo lugar, el efecto distorsivo del ruido ambiental sobre las salidas generadas por el sintetizador de voz podría evitarse usando los dispositivos móviles en ambientes que no sean ruidosos. De más está decir que esta solución no es aceptable porque estaría restringiendo la libertad de desplazamiento de los usuarios con discapacidad visual. Ninguna de las fuentes bibliográficas consultadas propone alguna estrategia para evitar las complicaciones que produce el ruido ambiental o lograr que la privacidad de los no videntes sea respetada. Pero, en un estudio de [11] sobre la accesibilidad de los lectores de pantalla se menciona que, para lograr privacidad los usuarios suelen modificar la presentación de las salidas del sintetizador de voz de modo que los sonidos sean imposibles de comprender para otras personas.

4.2. Solución a problemas de accesibilidad con el uso de mapas

Dos aplicaciones de mapas para Android que asisten a los discapacitados visuales en la exploración de ambientes exteriores son: TouchOver Map y PointNav.

Teniendo en cuenta que los individuos con impedimentos visuales no pueden percibir los mapas mostrados en la pantalla de dispositivos sensibles al tacto, la estrategia usada en TouchOver Map [19] consistió en usar modalidades de representación no visuales (realimentación auditiva y táctil) para presentar la información geoespacial. TouchOver Map usa la realimentación mediante audio y vibración lograr que los mapas de ruta sean accesibles a los discapacitados visuales. En TouchOver Map cada vez que un dedo del usuario toca una calle en el mapa se produce una vibración en el celular y el nombre de la calle se anuncia continuamente. Si el dedo se aleja de la calle, ambas señales se detienen. Sin embargo, se continúa leyendo el nombre de la calle hasta el final porque se asume que de esta manera se ayuda al usuario a seguir la ruta en el mapa y a formarse una idea de la localización de la calle dentro de la pantalla.

El aspecto más desafiante en lo que hace al desarrollo de TouchOver Map estuvo dado por decidir cuan anchas deberían ser las líneas que representan las calles o rutas en los mapas. Una línea se puede descubrir con facilidad si es ancha pero, se pierde precisión espacial si las líneas son demasiado anchas. Luego de varias pruebas experimentales en diversos dispositivos en [19] concluyeron que una línea ideal debería tener aproximadamente 48 píxeles.

Para evaluar la estrategia propuesta por TouchOver Map se pidió a 8 personas que utilicen TouchOver Map para explorar un mapa y luego dibujen el mapa en una hoja. A pesar de que ninguno de los usuarios dibujo una réplica exacta del mapa original, se comprobó que lograron formarse una clara idea de los objetos básicos y sus relaciones. Por lo tanto, el uso de la realimentación de audio y vibración a medida que los dedos se desplazan por los objetos de un mapa es una estrategia efectiva para ayudar a los usuarios con impedimentos visuales a percibir el contenido del mapa.

De manera análoga a TouchOver Map en PointNav [14] la realimentación dada al usuario es en forma de vibraciones y de voz. PointNav es una aplicación en la cual las personas con discapacidad visual puedan escanear información de un mapa usando sus teléfonos celulares como un dispositivo apuntador para orientarse y para seleccionar objetivos de navegación hacia los cuales ser guiados. La interfaz de PointNav contiene los 9 botones mostrados en la Figura 2a. El usuario percibe una pequeña vibración a medida que se desplaza de un botón al otro. Cuando el usuario se detiene en un botón concreto el mecanismo de realimentación por voz indica el nombre del botón. La selección de un botón se realiza levantando el dedo de la pantalla.

En el modo escáner el usuario apunta el celular en la dirección deseada y cuando el dispositivo señala un Punto de Interés (POI) dentro de cierto rango de distancia, se genera una vibración corta seguida del nombre del POI y la distancia. El ángulo de escaneo es de 30° (ver Figura 2b), y en caso de que varios POIs caigan dentro del mismo sector, se muestra el más cercano al ángulo de 0°. Presionando el botón "Add" el usuario puede seleccionar el último POI reportado y también tiene la posibilidad de solicitar más información. En el mundo real por lo general hay varios POIs y por

consiguiendo el usuario puede seleccionar que el escaneo se aplique para puntos cercanos (0-50 m), puntos intermedios (50-200 m) y puntos lejanos (200-500m).

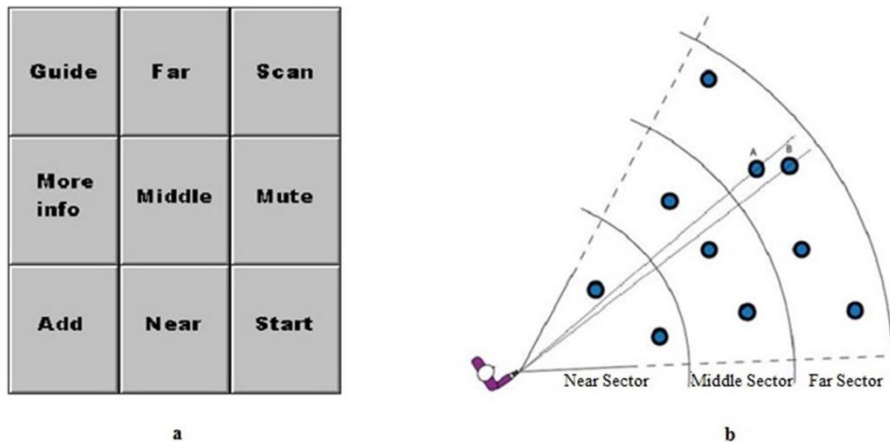


Figura 2. a) Diseño de la pantalla de interacción de PointNav b) Angulas de escaneo y rangos de sectores empleados en PointNav [14].

En el modo de guía o “Guide” el usuario es justamente guiado a un punto seleccionado previamente. Para dirigir al usuario hacia un POI en PointNav se utiliza el diseño mostrado en la Figura 3. Para “straight ahead” el ángulo es 46°, para “turn around” el ángulo es de 60° y tanto para “keep right” como para “keep left” el ángulo empleado es de 124°.

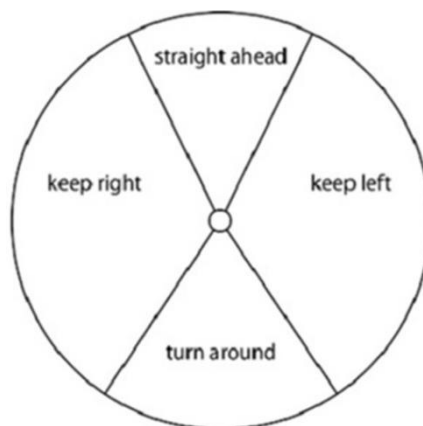


Figura 3. Diseño del mecanismo de guía de PointNav [14].

El diseño de la Figura 3 sirve de base para proveer no solamente información sobre la cercanía del dispositivo al ángulo de 0°, sino también sobre la dirección en la cual girar para apuntar más derecho hacia él. Esto significa que PointNav usa las direccio-

nes de desplazamiento para no tener que afrontar el problema de describir la forma y disposición de los objetos que aparecen en los mapas. A medida que el usuario se desplaza hacia el POI la aplicación anuncia el nombre del objetivo, la distancia y los textos de dirección de la Figura 3. La realimentación por voz se complementa con los siguientes patrones de vibración: larga corta para “keep right”, corta larga para “keep left”, 3 pulsos de vibración cortos para “straight ahead”, y una secuencia de vibraciones largas para “turn around”. La guía proporcionada al usuario se detiene cuando está ubicado a 15 metros o menos de su objetivo y la realimentación por voz señala la situación diciendo “Arrivando al <nombre del POI>. No más guía”. Además se produce una secuencia de 5 pulsos de vibración cortos.

Durante la evaluación del funcionamiento de PointNav se solicitó a 6 personas (5 no videntes y 1 sin problemas de visión) que utilicen la aplicación para desplazarse hacia 3 POIs en el mapa de un parque. La aproximación empleada en PointNav resultó ser efectiva ya que los 6 participantes pudieron localizar todos los POIs.

4.3. Solución problemas de accesibilidad con el teclado

Para las personas que no pueden ver los caracteres mostrados en pantalla el ingreso de texto es una necesidad apremiante sobre todo cuando usan dispositivos móviles. En la revisión bibliográfica realizada se pudo establecer que, para simplificar la tarea de escritura de los discapacitados visuales, dos estrategias frecuentemente empleadas son: realimentación vibro-táctil y el uso de marcadores táctiles.

El trabajo de [13] se centró por un lado en cómo organizar funciones y botones en la interfaz, y por el otro en cómo combinar mensajes de voz con feedback vibro-táctil para mejorar la percepción de teclas por parte de personas con discapacidad visual. En lo que respecta a la organización de la IU, la aplicación desarrollada para Android propone utilizar 4 puntos de referencias que estén siempre presentes en los *smartphones* para que el usuario pueda obtener información básica en cualquier momento. Estos 4 botones deberían estar localizados en las 4 esquinas de la pantalla como puede apreciarse en la Figura 4 y tienen la siguiente finalidad:

- 1- esquina superior izquierda: es el botón “Exit/Back” para volver al paso previo.
- 2- esquina superior derecha: es el botón “Position” para conocer el estado de la posición actual.
- 3- esquina inferior izquierda: botón “Repeat” que puede ser usado para leer un número de teléfono que fue editado de modo de chequear si hay errores.
- 4- esquina inferior derecha: botón “Done” que puede ser usado como un botón “OK” para confirmar una acción.

En particular las teclas “Position” y “Repeat” son útiles para obtener el estado actual en cualquier momento y así evitar la necesidad de tener que explorar la pantalla entera. En base a la estrategia de [13] los usuarios no videntes deberían utilizar gestos para navegar entre ítems (por ejemplo menús y sub menús, celdas de una tabla, etc.) mientras que, los 4 botones especiales pueden utilizarse en todo momento para solicitar información (teclas 2 y 3), retroceder (tecla 1) o confirmar la tarea actual (tecla 4).



Figura 4. Teclado numérico complementado con puntos de referencia [13].

Para ayudar a las personas con discapacidad visual a percibir los números en un teclado virtual [13] decidieron combinar vocalización de los números con vibración de las teclas. La vibración de las teclas es necesaria dado que en ambientes ruidosos los usuarios no siempre pueden oír las teclas anunciadas por el sintetizador de voz. En la primera versión prototipo del teclado numérico la vibración se utilizó solamente para marcar el número “5” como sucede comúnmente en los celulares con teclados físicos. Sin embargo, a partir de los comentarios de 4 usuarios que formaron parte el proceso de diseño, se decidió crear una nueva versión del teclado numérico que suministre diferentes tipos de feedback vibro táctil:

- Vibración simple para los números pares: 0, 2, 4, 6, y 8.
- Vibración doble para los números impares con excepción del 5: 1, 3, 7 y 9.
- Vibración triple para el número 5.

La solución propuesta por [4] para mejorar la accesibilidad de los no videntes, cuando usan teclados táctiles virtuales, se basa en usar combinaciones de mensajes de voz, audios y vibraciones que pueden ser insertadas en código fuente de la IU. El feedback de voz se complementó con las vibraciones puesto que los sonidos pueden no ser efectivos en entornos ruidosos y pueden ser mal vistos cuando se los usa durante clases, reuniones, o discursos. El objetivo que persiguen [4] con el uso de combinaciones personalizables de audio y vibración, un concepto que ha sido denominado *Customizable Accessibility Cue* (CAC), es asistir a los no videntes en la exploración de contenido y reconocimiento de áreas de interés en las pantallas táctiles pero sin que ello implique modificar la organización original de la IU.

En [4] plantearon a las CACs como objetos gráficos con comportamiento personalizable en términos de accesibilidad. Atributos XML se utilizan para definir la propiedad de accesibilidad de cada CAC y personalizarla a través de mensajes de voz, audios y vibraciones. Esta mejoría que puede lograrse en la accesibilidad de la IU es

casi transparente porque solo debe modificarse un archivo XML para configurar los parámetros de la CAC. El fragmento de código XML que se muestra debajo representa una CAC típica provista de un mensaje de voz, un patrón de vibración y un sonido.

```
<org.cnr.iit.accessible.CustomAccessibleButton
android:id="@+id/ballBtn01"
android:layout_width="fill_parent"
android:layout_height="20dp"
android:layout_gravity="top"
android:layout_margin="5dp"
android:paddingTop="10dp"
cnr:customMsg="custom spoken message"
cnr.vibPattern="0,100,100,100"
cnr:customSound="mySound" />
```

Una vez que el fragmento de código se inserta en el archivo XML, es traducido por Android en una bola coloreada reactiva tal como se muestra en la Figura 5.



Figura 5. CAC traducida por la IU como una bola coloreada [4].

Para seleccionar las mejores localizaciones de las CACs en [4] primero se identificaron las principales áreas interactivas en la aplicación de teclado numérico: barra de estado (indicador de batería, red móvil, tiempo, etc.), área de visualización numérica (donde aparecen los números), teclado numérico (para discar el número), botones de llamar y borrar, y barra de navegación (botones de inicio y atrás). Establecidas estas 5 áreas interactivas, se procedió a separarlas mediante 4 CACs. A las CACs que aparecen en la parte superior e inferior de la Figura 6 se les asocio un sonido tanto para resaltar el límite de la parte de la interfaz relacionada al teclado numérico, como para marcar la barra de estado (en la parte superior) y la barra de navegación (en la parte inferior). Estas señales ayudan a evitar que accidentalmente se presione alguna de las teclas programables incorporadas en los *smartphones* que componen las barras de navegación y de estado. Las otras dos CACs corresponden a patrones de vibración usados para señalar el comienzo y el final del teclado numérico. Con estas CACs intermedias también es fácil detectar el área de visualización numérica y los botones de llamada y borrar.

Durante la evaluación experimental de la aplicación de teclado numérico a dos usuarios ciegos se les pidió que: 1) marcaran un número a llamar y 2) chequearan el

número de teléfono en el área de visualización numérica. Ambos usuarios fueron capaces de completar las tareas y localizar las áreas deseadas con facilidad y rapidez.



Figura 6. Ejemplo de aplicación con CACs usadas para mejorar la accesibilidad [4]

Al igual que los dos trabajos anteriormente descriptos en esta sub sección, la aproximación de [2] se basó en aprovechar el canal táctil de los usuarios mediante marcadores táctiles que proveen puntos de referencia. Sin embargo, la idea de [2] involucró el desarrollo tanto de software como de hardware. Originalmente los marcadores táctiles flexibles se iban a implementar mediante actuadores basados en polímeros electro activos cuyo tamaño y forma varié en respuesta a estimulación eléctrica pero, los actuadores (construidos en la Universidad de Pisa) aún no estaban disponibles debido a que faltaba resolver cuestiones de diseño tales como tamaño y fuente de energía. Para superar este obstáculo y progresar en la construcción de un prototipo (específicamente en el componente de software para controlar la interfaz eléctrica a ser aplicada al *smartphone*), en [2] optaron por simular mediante LEDs los actuadores táctiles basados en polímeros electro activos.

Para investigar cómo manejar dinámicamente los puntos táctiles (es decir cuándo y cuáles activar) en [2] desarrollaron un controlador de hardware que coordina la activación y desactivación. En la construcción del controlador en [2] utilizaron un tablero de Arduino¹. Debido a que los puntos de referencia táctiles podían tener solo dos estados (“on” o bien “off”) se trabajó con un arreglo de LEDs para simular los actuadores. Así entonces, el primer prototipo consistió de un tablero de Arduino con 6 LEDs colocadas arriba y controladas por una aplicación de Android (‘BTBubbles’) a través de la librería Amarino² (Ver Figura 7).

¹ <http://www.arduino.cc/>

² <http://www.amarino-toolkit.net/>



Figura 7. Esquema del prototipo con puntos táctiles dinámicos [2].

Amarino es un framework que brinda métodos y herramientas para simplificar la comunicación entre Android y Arduino a través de una conexión Bluetooth. En el lado de Android, Amarino provee un servicio de background basado en APIs Bluetooth nativas. En el lado del tablero, existe una librería Amarino que permite al microcontrolador Arduino reconocer eventos que provienen de Android y enviar datos de regreso si es necesario. Cada vez que un cierto patrón de puntos necesita ser activado, la aplicación de Android envía una señal al microcontrolador de Arduino con la configuración de los puntos a ser activados o desactivados.

Para [2] la disposición de los marcadores táctiles no era algo arbitrario si no que debía efectuarse siguiendo el principio de partición lógico planteado por W3C WAI-ARIA³ (*Web Accessibility Initiative - Accessible Rich Internet Applications*). El principio de partición lógica sugiere utilizar roles de región y puntos de referencia (*landmarks*) para descomponer el contenido de una página web en varias partes que permiten al usuario obtener en forma rápida una visión global de la estructura de la página. Puesto que un *landmark* se coloca al comienzo de cada porción de la página, un bloque de contenido o marcador táctil debe utilizarse para señalar el principio de cada área de la IU.

Para ilustrar de qué manera el principio de partición lógica puede utilizarse para lograr que la estructura de una IU táctil en dispositivos móviles sea perceptible por usuarios no videntes considérese la aplicación para realizar llamadas cuya IU se muestra en Figura 8. Con el fin de simplificar la localización de elementos en la interfaz en [2] decidieron dividirla en 4 partes mediante 3 puntos de referencia. Dos de los 3 puntos coinciden con los bordes superior e inferior del área del teclado numérico (número 3 y número 4) y el tercer punto (número 5) junto con el número 4 ayudan al usuario a localizar con rapidez la barra de aplicación del teléfono para realizar una

³ <http://www.w3.org/WAI/intro/aria.php>

llamada y revisar el registro de llamadas o la lista de contactos. Una cuestión de suma importancia cuando se realiza una llamada es chequear si el número de teléfono marcado es correcto. Si bien esta operación es algo tan simple como leer el campo de edición que muestra el número ingresado, puede ser difícil para un discapacitado visual cuando no es ayudado por el lector de pantalla al momento de chequear el número (algo típico en ambientes ruidosos). Para permitir una mejor identificación del área con el número discado se agregó el punto de referencia número 2 (ver Figura 8).

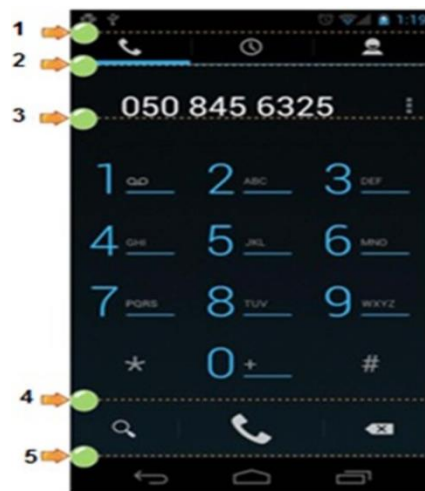


Figura 8. Disposición de las referencias táctiles en un teclado numérico [2].

Para evitar la pérdida del control sobre la aplicación al presionar uno de los botones de función de las barras de sistema, se utilizó los puntos de referencia número 1 y número 5 de la Figura 8 que delimitan las barras de navegación y de estado.

Si bien la propuesta de [2] se basa en el uso de marcadores táctiles, los propios autores reconocen que cada componente de la IU debería ser provisto simultáneamente con feedback táctil y oral.

4.4. Solución problemas de accesibilidad con reconocimiento de voz

El trabajo de [6] en torno a VoiceMail, una aplicación que permite a personas no videntes enviar y recibir e-mails, muestra una posible solución a los problemas típicos de los sistemas de ASR: precisión del reconocimiento de voz, revisión/edición del texto reconocido, e interferencia del ruido ambiental.

VoiceMail permite a los usuarios no videntes grabar su voz pero, en lugar de convertirla a texto el sistema envía directamente el mensaje de voz grabado a la dirección de e-mail del destinatario como un archivo adjunto. De esta manera los usuarios ya no tienen que preocuparse por los errores de reconocimiento que pudiera cometer el sistema de ASR ni mucho menos por corregir tales errores. El sistema también ofrece la

opción de acceder al buzón de entrada del e-mail en representación del usuario y leer los mensajes de voz recibidos.

Dado que la aplicación VoiceMail fue inicialmente pensada para funcionar en computadoras de escritorio, [6] crearon una nueva versión para ejecutarla en una plataforma embebida basada en Android. Debido a las restricciones de hardware de los dispositivos móviles se tuvo que rediseñar la GUI entera de VoiceMail de modo tal de optimizarla para pantallas de tamaño reducido. El proceso de rediseño de la GUI se hizo con la premisa de que una persona no vidente tuviera que realizar la mínima cantidad de operaciones para enviar o recibir un e-mail.

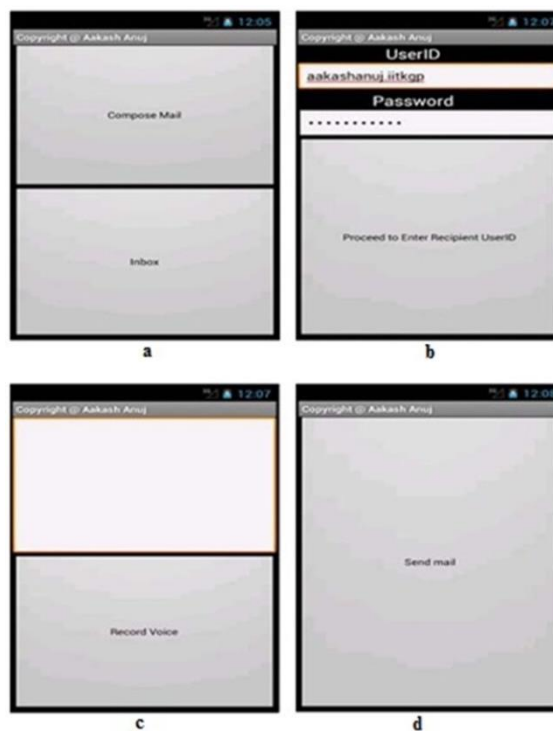


Figura 9. Capturas de pantalla de la IU de la versión móvil de VoiceMail [6].

Una captura de pantalla de la IU de la versión embebida de VoiceMail puede verse en la Figura 9. De manera similar a la versión de escritorio, la versión móvil de VoiceMail le da a los usuarios dos opciones: *Compose Mail* (redactar) y *Inbox* (chequear buzón de entrada). Sin embargo, mientras en la interfaz de escritorio las opciones pueden seleccionarse mediante clic del mouse, la interfaz móvil divide la pantalla en las dos secciones que pueden verse en la Figura 9a. Cuando se toca la sección superior se selecciona la opción “*Compose Mail*” y cuando se toca la sección inferior se selecciona la opción “*Inbox*”. Ante la selección de la opción “*Compose Mail*” lo primero que se hace es solicitarle al usuario que ingrese la dirección de e-mail de los destinatarios (Ver Figura 9b) y luego se le da la opción de escribir un mensaje de

texto o grabar un mensaje de voz (Ver Figura 9c). A la pantalla de selección del tipo de mensaje se la divide en 2 partes mediante un separador horizontal al igual que la pantalla inicial. El contacto con la mitad superior de la pantalla le permite al usuario el escribir un mensaje de texto mientras que el contacto con la mitad inferior de la pantalla activa el modo mensaje de voz. Una vez que la grabación finaliza VoiceMail guarda automáticamente el mensaje de voz grabado y lo envía a las direcciones de los destinatarios (Ver Figura 9d).

Si bien VoiceMail evita las dificultades asociadas con el reconocimiento de voz y la edición del texto reconocido, no resuelve todos los problemas ya que durante el proceso de redacción de los e-mails los usuarios tienen que introducir la dirección e-mail de los destinatarios mediante el teclado. Esta operación requiere conocimiento de la organización del teclado.

5. Conclusiones

Los dispositivos móviles crecen día a día en su capacidad de procesamiento y en la cantidad de aplicaciones originalmente pensadas para PC que pueden ejecutar, pero al mismo tiempo para las personas con discapacidad visual la interacción con estos dispositivos móviles es fuente de múltiples barreras o dificultades de accesibilidad que pueden causar desde frustración hasta resistencia a usar dispositivos móviles. El uso de recursos tales como lectores de pantalla o reconocedores de voz mejora la accesibilidad de las aplicaciones móviles para las personas con discapacidad visual pero, no basta con la tecnología de asistencia, gran parte de la responsabilidad por la existencia y eliminación de los problemas de accesibilidad recae sobre los desarrolladores de aplicaciones. Teniendo en cuenta que los desarrolladores por lo general no conocen de qué manera las personas con discapacidad visual interactúan con los dispositivos móviles, en la presente monografía se describieron los problemas de accesibilidad que enfrentan los usuarios con discapacidad visual junto con estrategias de solución.

Si bien parte de las barreras de accesibilidad se solucionan aplicando buenas prácticas de diseño y desarrollo de software, existe una serie de problemas (síntesis de voz, uso de mapas, ausencia de teclado físico y reconocimiento de voz) cuya solución exige estrategias más complejas. Estas estrategias de solución pueden mapearse a dos acciones básicas: inclusión de feedback en forma de voz o sonidos y uso de feedback en forma de vibraciones. Por lo tanto, existe una tendencia a explotar los sentidos del tacto y el oído de las personas con discapacidad visual para compensar la falta o reducción del sentido de la visión. Cada estrategia de accesibilidad podría explotar exclusivamente el canal táctil o auditivo de los no videntes pero como sostiene [4] en las aplicaciones móviles conviene que se utilice simultáneamente sonidos, síntesis de voz y vibraciones.

Las estrategias de solución a los problemas de accesibilidad que se describieron a lo largo de la monografía son bastante promisorias a pesar de que se necesiten algunos ajustes. De hecho, se estima que la accesibilidad de las aplicaciones móviles para discapacitados visuales mejoraría considerablemente si se abordaran dos cuestiones: precisión del reconocimiento de voz y accesibilidad en dispositivos de gama baja. Si

se tiene en cuenta que una persona puede llegar a pronunciar hasta 150 palabras por minuto [1], es claro que conviene invertir tiempo y esfuerzo tratando de mejorar la precisión del reconocimiento de voz así como también el tipo de asistencia que se le brinda al usuario para editar y corregir los errores de reconocimiento.

En lo que respecta a la accesibilidad en los dispositivos de gama baja, parecería existir el supuesto subyacente de que todos los individuos con discapacidades visuales pueden acceder a dispositivos móviles de gama alta o media ya que las aplicaciones descritas en el apartado 4 que se desarrollaron para probar las soluciones a los problemas de accesibilidad fueron implementadas en *smartphones*. Este supuesto deja a varios dispositivos básicos o de gama baja sin ningún tipo de característica de accesibilidad. Esto significa que las personas ciegas o visualmente impedidas solo reciben o realizan llamadas en teléfonos que no están equipados con ASR, TTS o pantalla Braille. Incluso la posibilidad de realizar llamadas depende de la capacidad de los usuarios con discapacidad visual de memorizar los números del teléfono y secuencias específicas de teclas. Características adicionales tales como estado de la batería, indicadores de mensajes, identificador de llamada, y envío o recepción de mensajes de texto no están disponibles para usuarios con discapacidad visual que utilizan teléfonos básicos.

Referencias

1. Azenkot, S., Lee, N.B.: Exploring the Use of Speech Input by Blind People on Mobile Devices. En: 15th International ACM SIGACCESS Conference on Computers and Accessibility. ACM, Nueva York (2013)
2. Buzzi, M.C., Buzzi, M., Donini, F., Leporini, B., Paratore, M.T.: Haptic Reference Cues to Support the Exploration of Touchscreen Mobile Devices by Blind Users. En: Biannual Conference of the Italian Chapter of SIGCHI. ACM, Nueva York (2013)
3. Buzzi, M. C., Buzzi, M., Leporini, B., Akhter, F.: Usability and Accessibility of eBay by Screen Reader. En: Holzinger, A., Miesenberger, K. (eds.) HCI and Usability for e-Inclusion. LNCS, vol. 5889, pp. 500–510. Springer-Verlag Berlin Heidelberg (2009)
4. Buzzi, M.C., Buzzi, M., Leporini, B., Paratore, M.T.: Vibro-Tactile Enrichment Improves Blind User Interaction with Mobile Touchscreens. En: Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (eds.) Human-Computer Interaction –INTERACT. LNCS, vol. 8117, pp. 641-648. Springer Berlin Heidelberg (2013)
5. Chiti, S., Leporini, B.: Accessibility of Android-Based Mobile Devices: A Prototype to Investigate Interaction with Blind Users. En: Miesenberger, K., Karshmer, A., Penaz, P., Zagler, W. (eds.) Computers Helping People with Special Needs. LNCS, vol. 7383, pp. 607-614. Springer-Verlag Berlin Heidelberg (2012)
6. Dasgupta, T., Anuj, A., Sinha, M., Ghose, R., Basu, A.: VoiceMail Architecture in Desktop and Mobile Devices for the Blind People. En: 4th International Conference on Intelligent Human Computer Interaction, pp. 1-6. IEEE Press (2012)
7. Harper, S., Yesilada, Y.: Web Accessibility and Guidelines. En: Harper, S., Yesilada, Y. (eds.) Web Accessibility: A Foundation for Research, pp.61- 78. Springer-Verlag (2008)
8. Kane, S.K., Bigham, J.P., Wobbrock, J.O.: Slide Rule: Making Mobile Touch Screens Accessible to Blind People Using Multi-Touch Interaction Techniques. En: 10th International ACM SIGACCESS conference on Computers and accessibility, pp.73-80. ACM, Nueva York (2008)

9. Kane, S.K., Wobbrock, J.O., Ladner, R. E.: Usable Gestures for Blind People: Understanding Preference and Performance. En: SIGCHI Conference on Human Factors in Computing Systems, pp. 413-422. ACM, Nueva York (2011)
10. Kocieliński, D., Brzostek-Pawłowska, J.: Improving the accessibility of touchscreen-based mobile devices: Integrating Android based devices and Braille notetakers. En: 2013 Federated Conference on Computer Science and Information Systems, pp. 655-658. IEEE Press, Nueva York (2013)
11. Kuber, R., Hastings, A., Tretter, M., Fitzpatrick, D.: Determining the Accessibility of Mobile Screen Readers for Blind Users. En: IASTED Conference on Human-Computer Interaction, pp. 182-189 (2012)
12. Lazar, J., Allen, A., Kleinman, J., Malarkey, C. :What Frustrates Screen Reader Users on the Web: A Study of 100 Blind Users. *International Journal of Human-Computer Interaction* 22 (3), 247-269 (2007)
13. Leporini, B., Chiti, S.: Investigating the Use of Vibro-tactile Feedback for Mobile Interaction by Blind Users: the Numeric Keypad Case. En: CHI2013 Mobile Accessibility Workshop (2013)
14. Magnusson, C., Molina, M., Rasmus-Gröhn, K., Szymczak, D.: Pointing for non-visual orientation and navigation. En: 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, pp.735-738. ACM, Nueva York (2010)
15. McGookin, D., Brewster, S., Jiang, W.: Investigating Touchscreen Accessibility for People with Visual Impairments. En: 5th Nordic conference on Human-computer interaction: building bridges, pp.298-307. ACM, Nueva York (2008)
16. G3ict – the Global Initiative for Inclusive Information and Communication Technologies, <http://goo.gl/0wpDCz>
17. Paciello, M.G.: *Web Accessibility for People with Disabilities*. CMP Books, Kansas (2000)
18. Pernice, K, Nielsen, J.: *Beyond ALT Text: Making the Web Easy to Use for Users with Disabilities*. Nielsen Norman Group, Fremont (2001)
19. Poppinga, B., Magnusson, C., Pielot, M., Rasmus-Gröhn, K.: TouchOver Map: Audio-Tactile Exploration of Interactive Maps. En: 13th International Conference on Human Computer Interaction with Mobile Devices and Services, pp.545-550. ACM (2011)
20. Qian, H., Kuber, R., Sears, A.: Towards developing perceivable tactile feedback for mobile devices. *International Journal of Human-Computer Studies* 69,705-719 (2011)
21. Sánchez Sierra, J., Selva Roca de Togores, J.: Designing Mobile Apps for Visually Impaired and Blind Users. En: *The Fifth International Conference on Advances in Computer-Human Interactions (ACHI 2012)*, pp. 47-52 (2012)
22. Su, J., Rosenzweig, A., Goel, A., de Lara, E., Truong, K.N.: Timbremap: Enabling the Visually-Impaired to Use Maps on Touch-Enabled Devices. En: 12th international conference on Human computer interaction with mobile devices and services, pp.17-26. ACM, Nueva York (2010)
23. The Government of the Hong Kong Special Administrative Region, <http://goo.gl/AMEqLX>
24. Yesilada, Y., Brajnik, G., Harper, S.: Barriers common to mobile and disabled web users. *Interacting with Computers* 23, 525-542 (2011)
25. Yesilada, Y., Chen, T., Harper, S.: *Mobile Web Barriers for the Barrier Walkthrough*. Human Centred Web Lab, School of Computer Science, University of Manchester (2008)