



INFORMAÇÕES GERAIS DO TRABALHO

Título do Trabalho: Recepção de SMS com *FeedBack* Falado usando Serviço de Acessibilidade

Autor (es): Daniela Costa Terra

Palavras-chave: Deficiência Visual, Android, *AccessibilityService*

Campus: Bambuí

Área do Conhecimento (CNPq): **1.00.00.00-3 Ciências Exatas e da Terra, 1.03.00.00-7 Ciência da Computação, 1.03.04.00-2 Sistemas de Computação, 1.03.04.02-9 Arquitetura de Sistemas de Computação**

RESUMO

Este texto apresenta uma proposta de aplicativo acessível para a recepção de mensagens de texto (SMS) para usuários com deficiência visual. O projeto propõe acessibilidade ao disponibilizar acesso às funções telefônicas básicas para esses usuários em aparelhos baseados em Android. O texto menciona sobre o serviço de acessibilidade da plataforma e dá um exemplo de customização do mesmo para oferecer uma interação simplificada com síntese de voz das novas mensagens SMS recebidas. A adesão do leitor de tela *TalkBack*, da Google Inc., ao aplicativo em questão é considerada em comparação ao serviço proposto. Algumas dificuldades encontradas no desenvolvimento do serviço são consideradas ao final.

INTRODUÇÃO:

De acordo com a *The World Bank* (2017) quinze por cento da população mundial tem algum tipo de deficiência, física ou mental. Segundo o Censo do IBGE de 2010 (2010, p. 118) cerca de 7 milhões de pessoas no Brasil possui baixa visão ou cegueira¹. Sonza (2013, p. 42) se refere a proposta de acessibilidade virtual como uma garantia de mobilidade e usabilidade de recursos computacionais a todos, promovendo a inclusão de pessoas com deficiência. A tecnologia, em especial a dos dispositivos móveis, como os celulares, *tablets* e vestíveis precisa ser empregada para produzir serviços e produtos com o objetivo de amenizar ou mesmo solucionar as barreiras encontradas por esses usuários.

Dispositivos móveis promovem a inclusão ao disponibilizar recursos como a conversão de texto para fala e reconhecimento de voz, além da leitura de variáveis a partir dos inúmeros sensores disponíveis, referentes à movimentação, posicionamento, velocidade, *etc.* Esses recursos de *hardware/software* possibilitam a inclusão digital e a mobilidade a usuários deficientes através de dispositivos portáteis equipados com aplicativos (*app*) acessíveis.

O Android, da Google Inc., empregado neste projeto para o desenvolvimento de aplicativos é uma plataforma amplamente adotada. A plataforma inclui um ambiente de execução, um sistema operacional Linux, uma biblioteca de classes e bibliotecas de tempo de execução (*middleware*). Segundo a empresa de

¹ Relativo à deficiência visual mais de 500.000 habitantes da população residente no país responderam que “não conseguem de modo algum” e mais de 6 milhões tem grande dificuldade visual.



pesquisa *Canalys* o Android equipa o maior número de aparelhos com sistemas operacionais móveis no mundo, em 2013 eram mais de 81,3% de *smarthphones* fabricados mundialmente. No Brasil, em outubro de 2016, mais de 90% dos aparelhos vendidos eram baseados em Android (Wikipedia, 2017).

Buscando dar visibilidade à em relação à acessibilidade da plataforma a fabricante vem embutido cada vez mais recursos nas novas versões. Alguns exemplos de serviços pré-instalados que favorecem uma interação do dispositivo para deficientes visuais são: o leitor de tela *TalkBack* que produz *feedback* falado durante a interação do usuário com o dispositivo, o *BrailleBack* para integração com equipamentos braile, e recentemente, o *Voice Access* para o controle por voz do dispositivo, ainda em versão de testes. O *TalkBack* possui similaridade com a solução aqui descrita pois também é um serviço de acessibilidade, conforme é discutido na próxima seção.

Sobre aplicações ou app acessíveis, o *Eyes-free project* é um exemplo de projeto para o desenvolvimento de acessibilidade para Android. Liderado por engenheiro deficiente visual da Google Inc., o *Eyes-free* disponibiliza algumas aplicações tais como o *Talking Dialer*, para serem empregadas *por usuários* que necessitam de uma interação sem visão (*eyes-free*²). O app permite o uso do *touchscreen* para originar chamadas pela entrada de um número de telefone ou a seleção do contato desejado a partir da agenda (GOOGLE OPENSOURCE BLOG, 2017). Este app inspirou o modo de interação por *touchscreen* para discagem e seleção de caracteres alfanuméricos deste projeto (TAVARES, 2016).

Seguindo a mesma linha, o projeto proposto propõe a construção de aplicativo acessível para celulares/tablets, baseados em Android. A motivação é possibilitar ao usuário com deficiência visual uma operação básica do *smartphone* incluindo o envio e o recebimento de chamadas, o envio e o recebimento de mensagens SMS e um gerenciamento básico da agenda de contatos do dispositivo. Outro objetivo importante é pesquisar e aplicar os recursos de acessibilidade disponibilizados pelo Android.

As seções a seguir descrevem a metodologia utilizada e as discussões acerca do desenvolvimento de um serviço de acessibilidade (*AccessibilityService*) específico que busque oferecer uma interação alternativa do usuário com o dispositivo através de telas sensíveis ao toque (*touchscreen*). Um exemplo de implementação de um *AccessibilityService* é apresentado para oferecer o recebimento de mensagens SMS.

METODOLOGIA:

Estudos sobre o desenvolvimento de aplicativos para plataforma Android constituiu a primeira etapa dos trabalhos. Os recursos permitem o tratamento de gestos, reconhecimento de voz, sintetize de voz, *etc.* (MEIER, 2012). A avaliação de aplicativos acessíveis existentes para usuários com deficiência visual foi realizada paralelamente aos estudos. Um exemplo de aplicativo que emprega alguns dos recursos mencionados é o *Talking Dialer*, do *Eyes-Free Project* que serviu de inspiração para o modelo de interação proposto no componente de discagem (TAVARES, 2015).

Seguindo o guia do desenvolvedor Android, o passo seguinte aprofundou os estudos para a utilização e a customização do serviço de acessibilidade oferecido pela plataforma (classe *AccessibilityService*). Um *AccessibilityService* é um componente do tipo serviço que executa em segundo plano para receber os

² O termo usado para descrever uma interação que dispensa o estímulo visual.



eventos (*AccessibilityEvent*) advindos da interação do usuário com o aplicativo em execução (ANDROID, 2017).

Para o desenvolvimento foram empregados um notebook Dell modelo Inspiron 5548, equipado com 8 GB de memória RAM e processador Intel Core i7-5500U. O aplicativo foi desenvolvido no ambiente de desenvolvimento oficial do fabricante, o *Android Studio*, na versão 2.0. Os testes foram realizados utilizando emuladores de celulares oferecidos pela plataforma e um *smarthphone* Samsung S5 equipado com a versão Android 6.0.1.

Inicialmente, um aplicativo para recepção de SMS foi desenvolvido e preparado para operar com o leitor de tela *Talkback* o qual deve reproduzir *feedback* falado usando a síntese de voz da mensagem recebida. Para obter os SMS um componente do tipo receptor de transmissão (*BroadCastReceiver*) foi implementado para capturar as mensagens recebidas. Esse componente é responsável por iniciar interface de usuário (*Activity*) sempre que detectada uma nova recepção. O tratamento de recepção é exibido na Figura 1.

Figura 1 – BroadCastReceiver para novas mensagens

```
@Override
public void onReceive(Context context, Intent intent) {
    Bundle bundle = intent.getExtras();
    String format = bundle.getString("format");
    if (intent.getAction().equals(ACTION)) {
        if (bundle != null) {
            SmsMessage[] messages = Telephony.Sms.Intents.getMessagesFromIntent(intent);
            for (SmsMessage message : messages) {
                String msg = message.getMessageBody();
                long when = message.getTimestampMillis();
                String from = message.getOriginatingAddress();
                Intent i = new Intent(context, 'ActivityReceivesSMS.class');
                i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                i.putExtra("br.ifmg.accesstelsms.RECEIVEDSMSFROM", from);
                i.putExtra("br.ifmg.accesstelsms.RECEIVEDSMS", msg);
                context.startActivity(i);
            }
        }
    }
}
```

Fonte: elaborado pelo autor

O aplicativo foi executado para testes em paralelo com o serviço de acessibilidade pré-instalado do *TalkBack*. Para operação adequada com o *TalkBack*, foram realizadas poucas modificações nas propriedades dos controles de interface da *Activity*. Essas modificações seguiram as diretrizes do guia para incluir, por exemplo, uma descrição de conteúdo para o controle (ANDROID, 2017 b).

Em tela a mensagem recebida é exibida e será ouvida após um duplo *click* no *touchscreen* se a caixa de texto (*TextView*) que exibe a mensagem tiver sido antes selecionada (foco). O *TalkBack* "lê" as informações do controle (ex., *TextView.getText()*) focalizado. A navegação para o próximo controle pode ser feita deslizando o dedo para a esquerda (*swipe left*)

Tendo em vista a ausência de tratamento de gestos que pudessem ser associados a comandos específicos ao *app* descrito foi iniciado o desenvolvimento/customização de um serviço de acessibilidade especializado para implementar uma interação específica para a recepção de mensagens. Uma interação adequada não seria a leitura da tela, executada pelo *TalkBack*, mais adequado fosse inicialmente informar que um novo SMS foi recebido e depois que o usuário sinalizar, executando um gesto ou um toque na tela, a mensagem seria ouvida (síntese de voz).



Para a especialização da classe *AccessibilityService* foi definida uma configuração para o disparo de eventos de exploração por toques (*touschscreen*). Nesse caso, para realizar a ação de “voltar” à tela anterior o usuário deve deslizar o dedo para baixo e para a esquerda (*swipe down left*). O serviço será específico à aplicação pelo filtro do pacote do aplicativo e deverá ser habilitado pelo usuário conforme condicionado pela especificações da plataforma. O trecho de código da Figura 2 ilustra como é possível simular a interação descrita. Um primeiro *click* informa a chegada de um SMS e o *click* seguinte sintetiza a mensagem recebida e seu remetente.

Figura 2 – Um *AccessibilityService* para o *AccessTel* - SMS

```
@Override
public void onAccessibilityEvent(AccessibilityEvent event) {
    if (!mTextToSpeechInitialized) {
        Log.e(LOG_TAG_INI, "Text-To-Speech engine not ready. Bailing out.");
        return;
    }
    AccessibilityNodeInfo inFocus= null, source = null; String formatado;
    if (event.getPackageName().equals(SMS_PACKAGE_APP)) { // obter
        if (event.getEventType() == AccessibilityEvent.TYPE_ANNOUNCEMENT) {
            speakToUser(event.getText().toString());
            return;
        }
        if (mOnGesture) {
            mOnGesture = false;
            mCountHoverEnterExit = 0;
        }
        if (event.getEventType() == AccessibilityEvent.TYPE_VIEW_HOVER_ENTER) {
            inFocus = findFocus(AccessibilityNodeInfo.FOCUS_INPUT);
            if (inFocus == null)
                return;

            if (mCountHoverEnterExit == 0) {
                mCountHoverEnterExit = 2;
                if (inFocus.getContentDescription() != null)
                    speakToUser(inFocus.getContentDescription().toString());
            }
        } else if (event.getEventType() == AccessibilityEvent.TYPE_VIEW_HOVER_EXIT) {
            inFocus = findFocus(AccessibilityNodeInfo.FOCUS_INPUT);
            if (inFocus == null)
                return;
            mCountHoverEnterExit--;
            if (mCountHoverEnterExit == 0) {
                inFocus.performAction(AccessibilityNodeInfo.ACTION_CLICK);
            }
        }
    }
}
```

Fonte: elaborado pelo autor

RESULTADOS E DISCUSSÕES:

Testes preliminares sugerem uma adequação do *AccessibilityService* customizado para execução paralela com o *app* descrito. A interação proposta com o usuário no recebimento de novas mensagens SMS pode ser implementada com o suporte do serviço. Uma dificuldade a ser solucionada é o travamento a partir da tela de configurações do dispositivo após a ativação do serviço, o que não ocorre na tela do aplicativo.

Além disso, observou-se que o emprego desse serviço para implementar a acessibilidade e a interação necessária introduz certa complexidade na codificação. A imprevisibilidade é percebida nos testes pois o serviço executará em paralelo à aplicação estabelecendo com a mesma uma comunicação inter-processo.



Por outro lado, a classe *AccessibilityService* prevê a leitura de um número razoável de gestos (método *onGesture()*) no deslizar do toque na tela em movimentos direcionais: deslizar para cima, para cima e para a direita, para cima e para a esquerda, para baixo, para baixo e para a direita, etc. Esses gestos que o *TalkBack* associa às ações genéricas como voltar à tela anterior, simulação da tecla “home”, etc., poderiam ser associados a ações ou comandos específicos nas funções de envio/recepção de SMS e chamadas e no gerenciamento de contatos.

CONCLUSÕES:

A cada nova versão a plataforma Android vem agregando melhorias em vários recursos, inclusive nos recursos de acessibilidade. Um *AccessibilityService*, por exemplo, a partir do Android 4.1 pode executar ações nos controles de GUI da aplicação cliente, via os objetos representativos desses controles (*AccessibilityNodeInfo*) reconhecidos pelo serviço. Os recursos inseridos às novas versões são normalmente implementados para as versões anteriores em APIs separadas, chamadas API de suporte. Isso permite o uso dos mesmos novos recursos em versões anteriores da plataforma.

Contudo a rapidez com que as evoluções são lançadas acabam gerando a desatualização do referencial teórico pré existente, desafiando o desenvolvedor na busca de soluções. Em consequência disso, uma das dificuldades encontradas neste projeto é lidar com um referencial bibliográfico “desatualizado” ou pouco detalhado para compreender o funcionamento e os recursos para um *AccessibilityService*. Mesmo os guias oficiais fornecem informação superficial ou exemplos de aplicação simplificados.

Dadas as questões de segurança envolvidas, acredita-se que novos e melhores serviços de acessibilidade, fornecidos pela Google, como é o caso do *TalkBack*, estejam disponíveis ao desenvolvedor em breve. Um exemplo é o serviço *Voice Access* que promete realizar controle completo do dispositivo por comandos de voz, incluindo o *feedback* por voz, permitindo auxiliar também usuários com limitações visuais.

Para as próximas etapas pretende-se avaliar a adequação de um serviço de acessibilidade, como o proposto, para oferecer uma interação de usuário adequada às demais funcionalidades de envio de SMS, envio e recepção de chamadas e gerência dos dados de contatos.

REFERÊNCIAS BIBLIOGRÁFICAS:

The World Bank. Disability. Disponível em: <<http://www.worldbank.org/en/topic/disability/overview>> . Acesso em: 14 mai. 2017.

IBGE. Censo demográfico 2010: características gerais da população, religião e pessoas com deficiência. IBGE: Rio de Janeiro, 2010.

Wikipedia. Android. Disponível em: <<https://pt.wikipedia.org/wiki/Android>> . Acesso em: 15 mai. 2017.

Google OpenSource Blog. Announcing Eyes-Free Shell for Android. Disponível em: <<https://opensource.googleblog.com/2009/04/announcing-eyes-free-shell-for-android.html>>. Acesso em: 15 mai. 2017.

TAVARES, Daniele N.; TERRA, Daniela C. Agenda de Contatos Acessível para Dispositivos Móveis para Portadores de Necessidades Visuais. Anais do V Simpósio de Iniciação Científica do IFMG (SIC 2016). Bambuí, 12-13 jul. 2016. Disponível em: <<https://www2.ifmg.edu.br/sic/edicoes-antiores/resumos-2016>>.



MEIER, Reto. Professional Android 4 Application Development: Update for Android 4. Indianapolis: Wrox, 2012.

TAVARES, Daniele N.; TERRA, Daniela C. Acessível: desenvolvimento aplicativos acessíveis para plataforma Android. Anais do IV Simpósio de Iniciação Científica do IFMG (SIC 2015). Congonhas, 13-14 out. 2015. Disponível em: <<https://www2.ifmg.edu.br/sic/edicoes-antiores/resumos-2015>>.

ANDROID DEVELOPERS. Building Accessibility Service. Disponível em: <<https://developer.android.com/guide/topics/ui/accessibility/services.html>>. Acesso em: 15 mai. de 2017.

ANDROID DEVELOPERS. Making Apps More Accessible. Disponível em: <<https://developer.android.com/guide/topics/ui/accessibility/apps.html>>. Acesso em: 15 mai. de 2017 b.

Participação em Congressos, publicações e/ou pedidos de proteção intelectual:

IV Simpósio de Iniciação Científica do IFMG (SIC 2015) . Congonhas, 13-14 out. 2015:

TAVARES, Daniele N.; TERRA, Daniela C. Acessível: desenvolvimento aplicativos acessíveis para plataforma Android. Anais do IV Simpósio de Iniciação Científica do IFMG (SIC 2015). Congonhas, 13-14 out. 2015. Disponível em: <<https://www2.ifmg.edu.br/sic/edicoes-antiores/resumos-2015>>.

V Simpósio de Iniciação Científica do IFMG (SIC 2016). Bambuí, 12-13 jul:

TAVARES, Daniele N.; TERRA, Daniela C. Agenda de Contatos Acessível para Dispositivos Móveis para Portadores de Necessidades Visuais. Anais do V Simpósio de Iniciação Científica do IFMG (SIC 2016). Bambuí, 12-13 jul. 2016. Disponível em: <<https://www2.ifmg.edu.br/sic/edicoes-antiores/resumos-2016>>.