

### 3.2 Android

Segundo Monteiro (2012), Android é um sistema operacional baseado no Linux, de código aberto e que utiliza a linguagem de programação Java para o desenvolvimento dos aplicativos. Criado especialmente para dispositivos móveis, o qual começou a ser desenvolvido no ano de 2003 pela então empresa Android Inc, que em 2005 foi agregada ao Google. A partir de 2007 o projeto Android uniu-se a Open Handset Alliance, uma associação de empresas de softwares, hardwares e telecomunicações, que tem por finalidade desenvolver uma plataforma para dispositivos móveis que seja completa, aberta e gratuita.

Krazit (2009) publicou uma entrevista com Rubin, um dos idealizadores do Android, o qual afirma que o sistema pode rodar em equipamentos de diversos fabricantes, evitando assim ficar limitado a poucos dispositivos. Conforme informações do site Android (2015), hoje em dia existe mais de um bilhão de aparelhos espalhados pelo mundo com esse sistema operacional.

De acordo com Monteiro (2012) as aplicações são executadas em uma máquina virtual Java denominada *Dalvik*. Cada aplicativo, usa uma instância dessa máquina virtual tornando-a assim mais segura. Por outro lado os softwares só podem acessar algum recurso do dispositivo, como uma lista de contatos, caso seja formalmente aceito pelo usuário nos termos de uso ao instalá-lo.

Dentre os muitos benefícios do Android, um recurso muito utilizado é a chamada *Intents*, que de acordo com K19 (2012, p.29), “são objetos responsáveis por passar informações, como se fossem mensagens, para os principais componentes da API do Android, como as *Activities*, *Services* e *BroadCast Receivers*.” Monteiro (2012) diz que as *Intents* são criadas quando se tem a intenção de realizar algo como por exemplo compartilhar uma imagem, utilizando os recursos já existentes no dispositivo. Existem dois tipos de *Intents*:

- *Intents* implícitas – Quando não é informada qual *Activity* deve ser chamada, ficando assim por conta do sistema operacional verificar qual a melhor opção.
- *Intents* explícitas – Quando é informada qual *Activity* deve ser chamada. Usada normalmente para chamar *activities* da mesma aplicação.

Segundo K19 (2012) uma aplicação Android pode ser construída com quatro tipos de componentes:

- *Activity* – São as telas com interface gráfica, com capacidade de interações com os usuários.
- *Services* – São serviços executados em segundo plano, com tarefas que levam algum tempo sem comprometer a interação do usuário.
- *Content Providers* – São os provedores de conteúdo que permitem o acesso e a modificação de dados.
- *Broadcast Receivers* – São componentes capazes de receber e responder a eventos do sistema operacional.

Em uma aplicação, um elemento fundamental é a interface gráfica, que deverá ser organizada, simples e elegante. Conforme Monteiro (2012) esses são os principais *Layouts* do sistema operacional Android:

- *LinearLayout* – Permite posicionar os elementos em forma linear, dessa forma quando o dispositivo estiver em forma vertical os itens ficaram um abaixo do outro e quando estiver na posição horizontal eles ficaram um ao lado do outro.
- *RelativeLayout* – Permite posicionar elementos de forma relativa, ou seja um item com relação a outro.
- *TableLayout* – Permite criar *layout* em formato de tabelas. O elemento *TableRow* representa uma linha da tabela e seus filhos as células. Dessa maneira, caso um *TableRow* possua dois itens significa que essa linha tem duas colunas.
- *DatePicker* – *Widget* desenvolvido para a seleção de datas que podem ser usadas diretamente no *layout* ou através de caixas de diálogo.
- *Spinner* – *Widget* que permite a seleção de itens.
- *ListView* – Permite exibir itens em uma listagem. Dessa forma, em uma lista de compras, clicando em uma venda é possível listar os itens dessa venda selecionada.
- *Menus* – Um item muito importante, pois apresenta aos usuários as opções existentes no aplicativo.
- *AlertDialog* – Apresenta informações aos usuários através de uma caixa de diálogo. Comumente utilizado para perguntar ao cliente o que deseja fazer quando seleciona algum elemento.

- *ProgressDialog e ProgressBar* – Utilizado quando uma aplicação necessita de um recurso que será demorado, como por exemplo, fazer um *download*, pode ser feito uma animação informando ao usuário o progresso da operação.

Para uma maior interação, as aplicações normalmente utilizam API's de terceiros, como o Google Maps, quando necessita encontrar alguma localização. Para Monteiro (2012) essa comunicação pode-se utilizar o REST, que envia requisições através da URL. Ao receber informações pedidas a um outro serviço, ela pode estar no padrão XML ou JSON. O REST será detalhado mais adiante.

Outra ferramenta importante e muito utilizada do Android é a Notificação. Segundo Phillips; Hardy (2013) quando uma aplicação está sendo executada em segundo plano e necessita comunicar com o usuário o aplicativo cria uma notificação. Normalmente as notificações aparecem na barra superior, o qual pode ser acessado arrastando para baixo a partir da parte superior da tela. Assim que o usuário clica na notificação ela cria uma *activity* abrindo a aplicação em questão.

Com a ideia de desenvolver um aplicativo para dispositivos móveis, a plataforma Android foi escolhida devido ao seu destaque no mercado e pela facilidade que apresenta aos usuários.

### **3.3 Android Studio**

Umas das ferramentas mais utilizadas para o desenvolvimento em Android é o Eclipse IDE, contudo a Google criou um software especialmente para esse ambiente, chamado Android Studio. Segundo Gusmão (2014), é uma IDE baseado no *IntelliJ Idea* e foi apresentado na Conferência para desenvolvedores I/O de 2013.

De acordo com Hohensee (2013) o Android Studio tem um sistema de construção baseado em *Gradle*, que permite aplicar diferentes configurações no código quando há necessidade de criar mais de uma versão, como por exemplo, um software que terá uma versão gratuita e outra paga, melhorando a reutilização do código. Com o *Gradle* também é possível fazer os *downloads* de todas as dependências de uma forma automática sem a necessidade de importar bibliotecas.

Hohensee (2013) afirma que o Android Studio é um editor de código poderoso, pois tem como característica a edição inteligente, pois ao digitar já completa as palavras reservadas do sistema operacional e fornece uma organização do código mais legível.

Segundo Android (2015) a IDE tem suporte para a edição de interface, o que possibilita ao desenvolvedor arrastar os componentes que deseja. Ao testar o aplicativo permite o monitoramento do consumo de memória e de processador por parte do utilitário.

Gusmão (2014) diz que a plataforma tem uma ótima integração com o GitHub e está disponível para Windows, Mac e Linux. Além disso os programadores terão disponíveis uma versão estável e mais três versões que estarão em teste chamadas de *Beta*, *Dev* e *Canary*.

Devido ao Android Studio ser uma ferramenta de fácil usabilidade e a IDE oficial para o desenvolvimento Android, esta foi escolhida como ambiente de construção do aplicativo.

## REFERENCIA

ANDROID. **A história do Android**. 2015. Disponível em: <https://www.android.com/history/>. Acesso em: 25 fev. 2015.

ANDROID. **Android Studio Overview**. 2015. Disponível em: <http://developer.android.com/tools/studio/index.html>. Acesso em: 12 mar. 2015.

GUSMÃO, G. **Google lança versão 1.0 do IDE de código aberto Android Studio**. 2014. Disponível em: <http://info.abril.com.br/noticias/it-solutions/2014/12/google-lanca-versao-1-0-do-ide-de-codigo-aberto-android-studio.shtml>. Acesso em: 03 mar. 2015.

HOHENSEE, B. **Getting Started with Android Studio**. Gothenburg, 2013.

K19 TREINAMENTOS. **Desenvolvimento mobile com Android**. 2012.

KRAZIT, T. **Google's Rubin: Android 'a revolution'**. 2009. Disponível em: <http://www.cnet.com/news/googles-rubin-android-a-revolution/>. Acesso em: 20 fev. 2015.

MONTEIRO, J.B. **Google Android: crie aplicações para celulares e tablets**. São Paulo: Casa do Código, 2012.

PHILLIPS, B; HARDY, B. **Android Programming: The Big Nerd Ranch Guide**. Atlanta: Big Nerd Ranch, 2013.