

# Um Estereótipo de Interface de Usuário para Sistemas de Informação em Android

Isabella Melo Freitas<sup>1</sup>, Valdemar Vicente Graciano Neto<sup>3</sup>, Sofia L. Costa Paiva<sup>2</sup>

<sup>1</sup> Universidade Federal de Minas Gerais

<sup>2</sup> Departamento de Ciências da Computação, Universidade Federal de São João Del Rei

<sup>3</sup> Instituto de Informática, Universidade Federal de Goiás

isabellamfreitas2@gmail.com, vgracianoneto@gmail.com, sofialarissa@gmail.com

**Abstract.** *The User Interface (UI) is one of the most complex components of modern Information Systems (IS), as it is necessary to achieve satisfactory communication between the UI and other parts of the code, in addition to offering acceptable consistency and performance. Model-based development approaches have aided in UI productivity and standardization. This paper proposes a Presentation Metamodel for UI in Android enabling the production of model-based UI. From an exploratory study with IS for Android devices, the lack of standardization was identified due to the publication policy on this platform. A Presentation Metamodel and a UI Stereotype were developed for the Android platform bringing guidelines and standards from the iOS platform, since this platform has more solid guidelines. An implementation of the UI Stereotype for Android was made showing the feasibility of using the Metamodel as an executable model. The benefits of Stereotype UI for Android include not only building and standardizing, but also reusing SI for the Android platform.*

**Resumo.** *A Interface com Usuário (IU) é um dos componentes mais complexos dos Sistemas de Informação (SI) modernos, já que é necessário alcançar uma comunicação satisfatória entre a IU e outras partes do código, além de oferecer consistência e desempenho aceitáveis [Bowen et al. 2020]. Abordagens de desenvolvimento baseado em modelos tem auxiliado na produtividade e padronização de IU. Este artigo propõe um Metamodelo de Apresentação para IU em Android possibilitando a produção de IU baseada em modelos. A partir de um estudo exploratório com SI para dispositivos Android, foi identificada a falta de padronização devido à política de publicação nessa plataforma. Um Metamodelo de Apresentação e um Estereótipo de IU foram desenvolvidos para a plataforma Android trazendo diretrizes e padrões da plataforma iOS, uma vez que tal plataforma possui diretrizes mais sólidas. Uma implementação do Estereótipo de IU para Android foi feita mostrando a viabilidade de utilizar o Metamodelo como um modelo executável. Os benefícios do Estereótipo de IU para Android incluem não apenas a construção e padronização, mas também a reutilização de SI para a plataforma Android.*

## 1. Introdução

A consolidação das Tecnologias de Informação e Comunicação (TICs) tem levado as organizações públicas e privadas a prestarem serviços e informações de modo ágil e

ubíquo. Assim, o desenvolvimento de aplicações web e móveis ainda traz complexidade devido às diferentes tecnologias envolvidas e se torna um desafio para sua implantação. Por esse motivo, cada vez mais tem sido investigada a usabilidade em aplicações web, em especial para dispositivos móveis, sendo um desafio a construção de normas e padrões para estas questões em aplicações para dispositivos móveis [Bowen et al. 2020].

O processo de construção de IU é um processo caro e propenso a erros. Isso ocorre principalmente por lidar com diferentes questões, desde aspectos cognitivos e psicológicos da modelagem de usuário até variações tecnológicas de renderização gráfica em diferentes plataformas [Costa et al. 2014a, Marin et al. 2015].

A geração de IU para aplicações em Sistemas de Informação (SI) móveis engloba várias etapas, como a modelagem de tarefas e dados que o usuário necessita manipular, dos relacionamentos entre tarefas e informações, e da forma de interação entre usuário e sistema. A complexidade dessa modelagem e da implementação dos modelos resultantes torna o processo de construção de IU para SI móveis difícil, caro e propenso a erros. Estudos mostram que questões relativas a IU podem consumir cerca de 50% do tempo de desenvolvimento do software [Lee and Kim 2009].

As abordagens para desenvolvimento de IU baseada em modelos promovem um processo de geração de código automatizado por meio de modelos e costumam ser genéricas para diversos tipos de plataformas computacionais [Marin et al. 2015]. No entanto, uma parte do processo de modelagem precisa tratar questões específicas da plataforma na qual o código-fonte será gerado. Nesse sentido, há abordagens que tratam questões importantes de padronização tanto de aparência quanto de comportamento de IUs em SI, como [Costa et al. 2014a]. Porém, tal abordagem foi aplicada apenas para plataformas convencionais WIMP (*windows, icons, menus and pointers*).

Dessa forma, o aumento no uso de dispositivos móveis e a necessidade de desenvolver SI móveis com IU adequadas às necessidades dos usuários fomentam a extensão da abordagem de construção de IU proposta em [Costa et al. 2014a] para a plataforma móvel, visando a padronização de SI para esta plataforma.

Este trabalho propõe um Metamodelo de Apresentação para Android, que deve ser utilizado como uma das etapas do processo proposto por [Costa et al. 2014a] para a geração de IU. Um estudo exploratório para compreender a apresentação e comportamento dos aplicativos para Android foi realizado, além de diretrizes para geração de IU para dispositivos móveis foram analisadas no processo de construção do metamodelo. Também foi proposto um Estereótipo de IU para Android visando promover a padronização e consistência dos SI para esta plataforma, em semelhança à plataforma iOS. Dessa forma, a aplicação do estereótipo proposto juntamente com o restante do metamodelo possibilita a padronização da geração de IU para dispositivos móveis tratando de especificidades dessa plataforma.

O artigo está organizado da forma a seguir. A Seção 2 apresenta os principais conceitos empregados neste estudo e os trabalhos relacionados. A Seção 3 descreve uma análise com os aplicativos atuais. A Seção 4 apresenta o metamodelo desenvolvido neste estudo. A Seção 5 mostra uma prova de conceito com a definição do Estereótipo de IU para Android. Por fim, a Seção 6 traz as considerações finais deste trabalho.

## 2. Estereótipo de Interface de Usuário

O desenvolvimento de IU compreende, minimamente, a modelagem de aspectos de apresentação e de comportamento da IU (*look and feel*). A apresentação corresponde à descrição da aparência e organização das informações ao serem mostradas para os usuários, conforme as informações que serão manipuladas por meio da IU. Já o comportamento descreve a forma pela qual a IU responde às interações do usuário com a aplicação. O comportamento da IU é influenciado pelo tipo de aplicação manipulado. Assim, espera-se que aplicações do mesmo tipo tenham comportamentos similares [Costa et al. 2014a].

O desenvolvimento de IU baseado em modelos (do inglês, *Model-Driven User Interface Development*) [Szekely 1996, Delgado et al. 2016] é uma abordagem que descreve a IU por meio de uma coleção de modelos que guiam o processo de desenvolvimento. Apesar do uso de modelos aumentar o nível de abstração e os modelos servirem como entrada para ferramentas de geração de código, o principal benefício é o auxílio na compreensão de como os requisitos são transformados em IU além de facilitar a reusabilidade de código e de conceitos de IU em outros projetos [Bowen et al. 2020].

Uma abordagem baseada em Estereótipos de IU foi apresentada em [Costa et al. 2014a]. Essa abordagem inclui os seguintes passos de geração de IU: i) geração da IU abstrata (independente de aspectos tecnológicos) com conceitos e serviços a serem disponibilizados, ii) geração da IU concreta, com conceitos específicos de uma plataforma computacional, e iii) geração do código-fonte final. A base desta abordagem é o conceito de Estereótipo de IU, na qual permite a definição de um conjunto de metamodelos para construção de IU para SI. Esta abordagem também possui um mecanismo para interpretar o modelo da IU concreta a fim de gerar o código-fonte final para uma plataforma específica e está integrada com um framework de aplicação para SI, envolvendo também outros aspectos de SI como modelo de negócios. Tais estudos tem usado a UML como modelo executável a fim de gerar o SI com os modelos propostos [Loja et al. 2020].

O conceito de Estereótipo de IU baseia-se no pressuposto de que SI possuem características e requisitos comuns para apresentação de informações [Costa et al. 2014a]. Um Estereótipo de IU deve capturar similaridades na forma de apresentação de informações, modelando a apresentação de IU e seu respectivo comportamento subjacente que aplicações exibem recorrentemente. Desse modo, este conceito possibilita abstrair um padrão de interações e tarefas do usuário de uma aplicação subjacente e a forma de apresentação desta tarefa. Cada Estereótipo de IU determina a forma de apresentação da informação para uma dada necessidade do SI e o comportamento da IU para cada interação com esta apresentação [Graciano Neto et al. 2017, Costa et al. 2014a].

Além disso, um trabalho relacionado é o framework CAMELEON, que foi evoluído para LIZARD [Marin et al. 2015]. Tal abordagem divide os modelos em: IU de alto nível, IU da plataforma e IU final. Porém, a abordagem de [Costa et al. 2014a] apresenta metamodelos relacionados ao primeiro passo do processo de geração de IU (IU abstrata) e o segundo passo está relacionado apenas a plataformas WIMP tradicionais.

Deste modo, um estereótipo de IU é um elemento dominante que organiza a apresentação de elementos subordinados e seus comportamentos. Um exemplo de um Estereótipo de IU possui no mínimo um elemento de IU de acordo com a taxonomia do Metamodelo de IHC. Além disso, as ações de IU são predefinidas pelo Estereótipo

de IU, fazendo com que todas as instâncias executem o mesmo comportamento. Ações que dependem da aplicação ou de sistemas externos são especificadas como ações externas e são associadas a uma instância de Estereótipo de IU no momento de instanciação [Costa et al. 2014a]. Exemplos de Estereótipos de IU, além de exemplos de modelagem, foram apresentados: Questionário [Costa et al. 2014a] e Portal Web [Costa et al. 2014b], evidenciando os benefícios em termos de reutilização e padronização de IU. Porém, o estudo não inclui as plataformas móveis.

### 3. Estudo exploratório sobre aplicativos Android

Esta seção apresenta o estudo exploratório com aplicativos Android em comparação com aplicativos para a plataforma iOS, que possuem diretrizes bem definidas de IU.

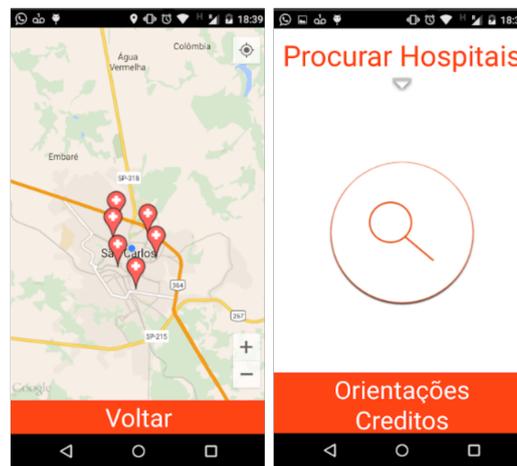
Baseado em sistemas de código aberto (do inglês *open source softwares*), o sistema operacional Android traz consigo os mesmos princípios desta comunidade de programadores e entusiastas. Desta forma, qualquer pessoa pode produzir código para a plataforma, contribuindo para o desenvolvimento constante deste sistema operacional e auxiliando no tratamento de defeitos. Como consequência, há uma sensação de liberdade nos projetos, popularizando a produção de aplicativos e acelerando a modernização de conteúdos. Entretanto, a falta de padrões de design leva a projetos com padrão estético desagradável ao usuário e excesso de informações.

Exemplos de aplicativos Android mostram a necessidade de refinamento para prover melhor experiência ao usuário, seguindo os padrões propostos em [Oliveira et al. 2012]. A Figura 1 apresenta um aplicativo governamental que permite ao usuário encontrar UBS (Unidades Básicas de Saúde) próximas de sua localização. Além de exibir as UBS mais próximas, ele fornece uma pontuação (de 1 a 3) sobre a estrutura física do hospital, seus equipamentos, medicamentos e o nível acessibilidade para deficiente físico e idoso. Com um botão central sem qualquer descrição e que pode ser confundido com uma imagem qualquer, usuários dependentes de aplicativos de leitura de tela (i.e pessoas com deficiência visual ou baixa visão) podem ter dificuldade em distinguir o que deve ser feito nesta tela. Também não há hierarquia de objetos, fator importante para ajudar os observadores na navegação e atrair o usuário. O menu inferior não separa os botões, causando uma falta de direcionamento à IU - o usuário não sabe onde deve clicar. A tela de resultados não apresenta qualquer descrição textual dos mesmos, falhando novamente em quesitos de acessibilidade da aplicação.

Como forma de prevenir este problema, o sistema operacional iOS (concorrente do Android) criou um rígido conjunto de regras de design, nomeadas como Diretrizes de Interface Humana (<sup>2</sup>, em português). Há uma extensa avaliação para que um aplicativo seja colocado à disposição do público, como medida para preservar a imagem do produto.

Conforme as Diretrizes de IU para iOS, são, portanto, boas práticas presentes em IUs para aplicações iOS (i) *bottom navigation*, ou navegação inferior - os menus são situados na parte inferior da IU, garantindo consistência e fácil alcance; (ii) mais textos e menos *icons*, garantindo intuitividade à aplicação (i.e substitui-se um símbolo para a opção "Voltar" pela palavra Voltar); (iii) padrões de cores mais claras, seguindo uma tendência de cores pastéis para garantir melhor fluidez à aplicação; e (iv) menos informações, garantindo uma navegação centralizada e sem distrações.

<sup>2</sup>Disponível em: <https://developer.apple.com/design/human-interface-guidelines/ios/>



**Figura 1. Aplicativo Android para encontrar UBS (Unidades Básicas de Saúde).**

É oportuno lembrar que além de questões de design de IU, deve-se levar em consideração a usabilidade das aplicações [Kronbauer et al. 2012]. Neste âmbito, eis outra preocupação para desenvolvedores iOS, uma vez que estes devem seguir estritamente a ISO 9241-11<sup>1</sup>, responsável pela definição deste termo como um meio no qual o produto pode ser utilizado pelos usuários, realizando a tarefa que o produto tem como objetivo.

Como apresenta [Vaupel et al. 2018], outro aspecto importante entre sistemas iOS e Android é o mapeamento de processos invocados, ou seja, serviços externos à aplicação. A plataforma Android fornece um construtor *Intent*<sup>2</sup> para chamar um componente *Service*<sup>3</sup>, enquanto o iOS é mapeado com uma instanciação da classe de serviço correspondente. A prática Android promove segurança à aplicação, em razão da possibilidade de determinar qual serviço responderá ao *Intent* (através da classe de mesmo nome) e a inicialização do *Service* é invisível ao usuário. Tal controle não é possível pela abordagem utilizada pelo iOS, promovendo certo risco à aplicação.

#### **4. Metamodelo de Apresentação para Android**

Esta seção apresenta o Metamodelo de Apresentação para Android, visto na Figura 2, que é uma evolução de [Freitas and Paiva 2018]. Além de apresentar os principais elementos que compõem uma IU de usuário para Android, ele também apresenta como estes elementos são integrados ao tratamento do comportamento da IU.

Uma IU em Android é composta por *Views* na qual representam uma composição de retângulos que apresentam conteúdo na tela do dispositivo móvel. Uma composição de *Views* organiza o *layout* da IU, e o *layout* possui atributos – características que determinam o comportamento ou a aparência desta *View* – específicos para cada aplicação [Freitas and Paiva 2018].

<sup>1</sup>Disponível em: <https://www.en-standard.eu/bs-en-iso-9241-11-2018-ergonomics-of-human-system-interaction-usability-definitions-and-concepts/>

<sup>2</sup>Objeto de mensagem usado para solicitar uma ação de outro componente de aplicativo. Disponível em: <https://developer.android.com/guide/components/intents-filters?hl=pt-br>

<sup>3</sup>Realiza operações em segundo plano sem IU. Disponível em: <https://developer.android.com/guide/components/services?hl=pt-br>

Para determinar a interação entre o usuário e a aplicação, existe uma tendência crescente para que o usuário faça uso máximo da aplicação com o mínimo de interação física possível com a mesma [Emmanouilidis et al. 2013]. Se estiverem presentes recursos de toque ao dispositivo e o usuário começar a interagir com a interface ao tocá-la, torna-se desnecessário sempre realçar ou dar foco a uma determinada *View*. Motiva-se assim um modo de interação Android chamado "*touch mode*", ativo através do método *Activity*. *Call isInTouchMode()*, presente no referido metamodelo.

Nesse sentido, cada método de uma IU pertence a classe *Activity*, algo único que o usuário pode fazer. Quase todas elas interagem com o usuário, portanto a *Activity* cuida da criação de uma janela na qual pode-se colocar uma IU com o método *setContentViews* (*View*). Para o tratamento destes métodos, uma atualização de [Freitas and Paiva 2018] implementa o pacote "Eventos" que trata as classes: (i) Métodos, responsável por *onCreate* - na qual inicializa-se a *Activity* e *setOnClickListener* - registra uma chamada a ser executada quando esta visualização for clicada. Se essa visualização não for clicável, ela se tornará clicável; (ii) *Intent*, um objeto de mensagem usado para solicitar uma ação de outro componente de aplicativo; e (iii) *Activity*, com todos os métodos possíveis desta classe, cada qual podendo executar algo para o usuário.

Sobre o *design* de IU em Android, pesquisadores concordam que não há um processo único de criação [Wang and Shan 2005]. Como forma de auxiliar o processo, este metamodelo apresenta uma visão de todo o espaço de *design*, descrevendo a IU concreta da aplicação e aprimorando seu entendimento, incluindo a visualização da hierarquia de componentes (os elementos deste estão ordenados em uma árvore de precedência), e definindo as regras para estruturação da IU, mantendo a mesma reutilizável e padronizada.

O metamodelo proposto não trata regras de domínio da aplicação, que já foram tratadas pela abordagem [Costa et al. 2014a], mas mostra-se completo para modelagem de IU. A composição e hierarquia de uma IU para Android foi tratada de forma integral, evidenciando os possíveis relacionamentos entre os elementos. A seguir, é apresentado um Estereótipo de IU baseado na ideia de padronização trazida por este metamodelo.

## 5. Prova de Conceito: Estereótipo de IU para Android

Tendo como base o Metamodelo de Apresentação para Android e as Diretrizes para IU de aplicações iOS, foi possível identificar elementos a serem padronizados para SI móveis da plataforma Android. Assim, é proposta a seguinte padronização sumarizada em um Estereótipo de IU. A Figura 3 apresenta uma abstração da forma de apresentação de uma aplicação Android genérica. Mesclando características das plataformas abordadas neste trabalho, temos as seguintes partes:

- **Barra de ações:** Um dos elementos de *design* mais importantes das aplicações Android, fornece uma estrutura visual e elementos interativos que são familiares para os usuários. O uso da barra de ações torna a aplicação consistente com outros aplicativos, permitindo que os usuários entendam rapidamente como operá-lo e ter uma ótima experiência.
- **Conteúdo:** Parte do estereótipo que possibilita apresentar informações ou agregar outros estereótipos. Como padrão é exibido um estereótipo de tela inicial [Costa et al. 2014b] e com o decorrer da navegação esta parte é atualizada com os estereótipos solicitados.

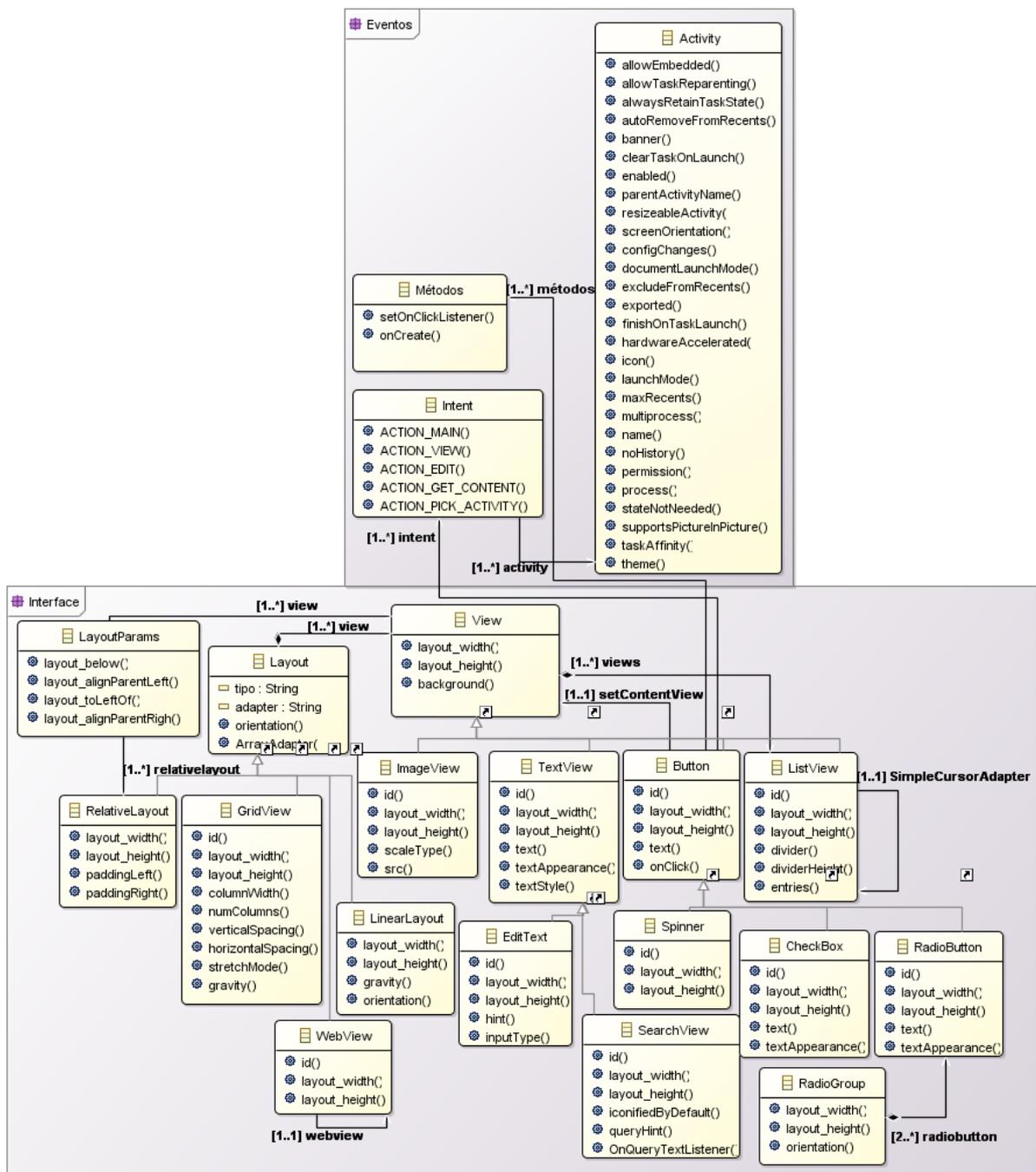
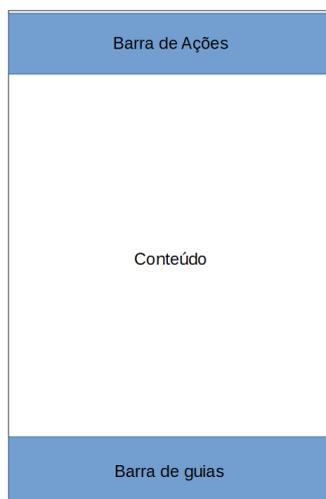


Figura 2. Metamodelo de Apresentação para Android

- **Barra de guias:** Derivada de aplicações iOS, aparece na parte inferior de uma tela de aplicativo e oferece a capacidade de alternar rapidamente entre diferentes seções de um aplicativo, além de centralizar a navegação pela aplicação. Pode conter qualquer número de guias, mas o número de guias visíveis varia de acordo com o tamanho e a orientação do dispositivo.

Cada barra deve disparar uma ação interna do estereótipo cuja responsabilidade é atualizar a IU de acordo com a solicitação realizada pelo usuário. Na parte de conteúdo, não há ações predefinidas, a não ser que seja agregado outro estereótipo que contenha



**Figura 3. Apresentação do Estereótipo de IU para Android**

outras ações. Como o conteúdo da aplicação pode ser composto por outros estereótipos, este deve ser capaz de carregar uma IU solicitada de forma correta, podendo utilizar o conhecimento do estado corrente de cada um desses estereótipos [Costa et al. 2014a].

Como pode ser constatado no estereótipo apresentado, foi necessária a agregação das barras de guias e ações na mesma IU. Esta escolha ocorre por dois motivos, sendo eles (i) a barra de ações tem uso encorajado pelo material de design para desenvolvedores Google <sup>4</sup>, fazendo parte da identidade visual do sistema Android. A criação de uma identidade visual consolidada promove reconhecimento e visibilidade organizacional, além de ligações emocionais; (ii) a barra de guias organiza as informações no aplicativo. Esta é uma boa maneira de nivelar a hierarquia de informações e fornecer acesso a várias categorias de mesmo nível de uma só vez. Além disso, toda atenção do usuário é voltada ao centro da aplicação, onde fica o conteúdo, reduzindo a distração pela quantidade de menus e informações na parte superior.

Embora as convenções de *design* possam diferir de plataforma para plataforma, há um esforço para que este estereótipo preserve as características essenciais do sistema Android. O mesmo foi construído sobre uma base de melhores práticas para desenvolvimento de IU Android e iOS, esperando que as diretrizes de design desenvolvidas a partir dessas descobertas possam ser aplicadas à qualquer aplicativo Android, maximizando a experiência positiva do usuário com a IU.

Para demonstrar o uso deste Estereótipo de IU na plataforma Android, foi feita uma prova de conceito. De forma semelhante ao estudo de caso de [Costa et al. 2014a] com a tecnologia JSF <sup>5</sup>, foi gerado o código deste modelo de Estereótipo de IU usando o conceito de *template* com a estrutura proposta na Figura 3. Além disso, para prover a parte de comportamento do *template*, classes abstratas em Java, correspondendo às classes do Metamodelo, foram criadas a partir do Ecore <sup>6</sup> na IDE Eclipse <sup>7</sup>. Porém, não apenas a

<sup>4</sup>Disponível em: <https://developer.android.com/design/material/?hl=pt-br>

<sup>5</sup><https://javaee.github.io/javaxserverfaces-spec/>

<sup>6</sup><https://projects.eclipse.org/projects/modeling.emf.ecoretools>

<sup>7</sup><https://www.eclipse.org>

aparência foi colocada como um template, como também classes em Java foram trazidas do Ecore para a IDE Android Studio <sup>8</sup> e os devidos métodos foram implementados. Uma classe foi criada para um aplicativo de teste e o template foi importado. No layout.xml foi colocado o componente de template criado. Assim, podemos observar a viabilidade de implementação desses conceitos usando a plataforma Android.

## 6. Considerações Finais

Este trabalho apresentou o Metamodelo de Apresentação para Android, correspondendo ao segundo passo para a geração de IU da abordagem baseada em modelos para geração de IU proposta por [Costa et al. 2014a]. O estudo exploratório com aplicativos da plataforma Android identificou problemas de padronização devido a política de publicação de aplicativos desta plataforma. Analisando-se as diretrizes de aplicativos para a plataforma iOS, foi identificado que há um padrão para aplicações desta plataforma móvel. Tais padrões foram trazidos para um Estereótipo de IU para Android, que não só foi projetado como também demonstrada sua viabilidade de implementação por meio de templates em Android. Tal prova de conceito mostrou a viabilidade de tornar esse modelo executável por meio de um mecanismo de geração de código-fonte, nos mesmos moldes que os trabalhos anteriores desta mesma abordagem baseada em modelos [Loja et al. 2020]. Essa abordagem possibilita que aplicativos com os problemas apresentados na Figura ?? sejam evitados, por preocupar-se com a padronização de elementos nessa plataforma. Além disso, o uso do Estereótipo de IU para Android possibilita a automatização da geração de IU com esse formato e que seguem o metamodelo proposto.

O Estereótipo de IU para Android promove o desenvolvimento de aplicativos padronizados e consistentes, além de possibilitar a reusabilidade de IU com comportamento similar. O metamodelo aqui proposto é aderente às diretrizes para a plataforma iOS. Próximos passos incluem desenvolver o mecanismo de geração de código-fonte a partir deste metamodelo, realizar estudos de caso que mostrem a geração de SI móveis a partir deste mecanismo, além de possibilitar a manutenção de forma visual por meio do desenvolvimento de ferramentas de apoio a essa abordagem. Outro ponto importante é alinhar este mecanismo de geração com os demais aspectos inerentes ao desenvolvimento de SI – modelos de negócio e domínio – de modo que seja integrado ao processo completo de geração de SI baseada em modelos. Uma comparação do estereótipo proposto com os aplicativos da plataforma iOS poderiam mostrar se há aderência dessa abordagem para a plataforma iOS, uma vez que para a geração do metamodelo proposto foram usadas as diretrizes da plataforma iOS.

## Agradecimentos

Os autores agradecem o apoio financeiro do CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) com projeto No. GS501100001809 e da UFSJ.

## Referências

- [Bowen et al. 2020] Bowen, J., Winckler, M., and Vanderdonckt, J. (2020). A glimpse into the past, present, and future of engineering interactive computing systems. *Proc. ACM Hum.-Comput. Interact.*, 4(EICS).

---

<sup>8</sup><https://developer.android.com/studio>

- [Costa et al. 2014a] Costa, S. L., Graciano Neto, V. V., Oliveira, J. L., and Calçado, B. R. (2014a). User interface stereotypes: A model-based approach for information systems user interfaces. In *Proc. of the SBSI '14*, pages 113–124, Londrina, Brazil. SBC.
- [Costa et al. 2014b] Costa, S. L., Neto, V. V. G., and Oliveira, J. L. (2014b). A user interface stereotype to build web portals. In *9th Latin American Web Cong.*, pages 10–18.
- [Delgado et al. 2016] Delgado, A., Estepa, A., Troyano, J., and Estepa, R. (2016). Reusing ui elements with model-based user interface development. *International Journal of Human-Computer Studies*, 86:48–62.
- [Emmanouilidis et al. 2013] Emmanouilidis, C., Koutsiamanis, R.-A., and Tasidou, A. (2013). Mobile guides: Taxonomy of architectures, context awareness, technologies and applications. *Journal of network and computer applications*, 36(1):103–125.
- [Freitas and Paiva 2018] Freitas, I. and Paiva, S. L. C. (2018). *Anais I Workshop Lat.-Amer. Trab. em Andam. em Comput.*, volume 1, chapter Definição de um Metamodelo para geração de Interfaces de Usuário para aplicativos móveis, pages 56–61. SBC.
- [Graciano Neto et al. 2017] Graciano Neto, V. V., Paiva, S. L. C., Loja, L. F. B., and Oliveira, J. L. (2017). *Compêndio sobre Pesquisa e Desenv. de Plataformas Computacionais Centradas na Web: Uma visão da Eng. de Soft.*, volume 1, chapter Web Enterprise Information Systems Engineering: A Path and the Road Aheads, pages 33–42. UFSJ.
- [Kronbauer et al. 2012] Kronbauer, A. H., Santos, C. A., and Vieira, V. (2012). Smartphone applications usability evaluation: a hybrid model and its implementation. In *International Conference on Human-Centred Software Engineering*, pages 146–163. Springer.
- [Lee and Kim 2009] Lee, C.-M. and Kim, Y.-W. (2009). User interface prototype generation technique supporting usage-centered design. *International Journal of Software Engineering and Knowledge Engineering*, 19(01):23–46.
- [Loja et al. 2020] Loja, L., Paiva, S., and Oliveira, J. (2020). Towards bpm@runtime. In *Anais do II Workshop em Modelagem e Simulação de Sistemas Intensivos em Software*, pages 16–25, Porto Alegre, RS, Brasil. SBC.
- [Marin et al. 2015] Marin, I., Ortin, F., Pedrosa, G., and Rodriguez, J. (2015). Generating native user interfaces for multiple devices by means of model transformation. *Frontiers of Information Technology & Electronic Engineering*, 16(995–1017).
- [Oliveira et al. 2012] Oliveira, D. H. D., Miranda, L. C., Miranda, E. E. C., and Silva, L. F. (2012). Prototipação de interfaces de aplicativos para dispositivos móveis: Estado da arte e desafios de IHC. In *Proc. of IHC'12*, pages 315–324, Porto Alegre, Brazil. SBC.
- [Szekely 1996] Szekely, P. (1996). Retrospective and challenges for model-based interface development. In *Design, Specification and Verification of Interactive Systems '96*, pages 1–27, Namur, Belgium. Springer-Verlag.
- [Vaupel et al. 2018] Vaupel, S., Taentzer, G., Gerlach, R., and Guckert, M. (2018). Model-driven development of mobile applications for android and ios supporting role-based app variability. *Software & Systems Modeling*, 17(1):35–63.
- [Wang and Shan 2005] Wang, G. G. and Shan, S. (2005). Review of metamodeling techniques for product design with computation-intensive processes. *Proceedings of the Canadian Engineering Education Association (CEEA)*.