Logotipo, nome da empresa

Descrição gerada automaticamente

**UNIVERSIDADE DE UBERABA**

**SAMUEL LUCAS RODRIGUES DE OLIVEIRA E SILVA**

**DESENVOLVIMENTO DE APLICAÇÃO ASSISTENTE NA FORMAÇÃO DE  
HÁBITOS BASEADA EM NEUROCIÊNCIA COMPORTAMENTAL**

UBERLÂNDIA – MG 2024

**SAMUEL LUCAS RODRIGUES DE OLIVEIRA E SILVA**

**DESENVOLVIMENTO DE APLICAÇÃO ASSISTENTE NA FORMAÇÃO DE  
HÁBITOS BASEADA EM NEUROCIÊNCIA COMPORTAMENTAL**

Trabalho apresentado à Universidade de Uberaba, como parte das exigências à conclusão do componente Trabalho de Conclusão de Curso, da 10ª etapa, do curso de graduação em Engenharia da Computação, da UNIUBE, Campus

Uberlândia.

Orientador: Prof. Me. Clênio Eduardo da Silva

UBERLÂNDIA – MG 2024

**SAMUEL LUCAS RODRIGUES DE OLIVEIRA E SILVA**

**DESENVOLVIMENTO DE APLICAÇÃO ASSISTENTE NA FORMAÇÃO DE HÁBITOS BASEADA EM NEUROCIÊNCIA COMPORTAMENTAL**

Trabalho apresentado à Universidade de Uberaba, como parte das exigências à conclusão do componente Trabalho de Conclusão de Curso, da 10ª etapa, do curso de graduação em Engenharia Elétrica, da UNIUBE, Campus Uberlândia.

Aprovado em 13 de dezembro de 2023.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Me. Clênio Eduardo da Silva Universidade de Uberaba

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Me. Stéfano Schwenck Borges Vale Vita Universidade de Uberaba

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Me. Júlio Almeida Borges

Universidade de Uberaba

**DESENVOLVIMENTO DE APLICAÇÃO ASSISTENTE NA FORMAÇÃO DE HÁBITOS BASEADA EM NEUROCIÊNCIA COMPORTAMENTAL**

Samuel Lucas Rodrigues de Oliveira e Silva  
 samuellucasrdg@gmail.com

Clênio Eduardo da Silva clenio.silva@uniube.br

# RESUMO

Atualmente o usuário médio de internet passa cerca de seis horas de seu dia “online”, devido a facilidade de acessar o meio digital e aplicativos que tem como parte de seu “design” a adicção de seus usuários, por meio de técnicas como a seleção de conteúdos baseados no interesse do usuário. No entanto, o ambiente digital não foi criado com o intuito de ser um local maléfico que retira a capacidade produtiva de seus usuários. Esse trabalho tem como objetivo criar uma aplicação *mobile*, com o intuito de auxiliar na formação de novos hábitos utilizando de técnicas baseadas na neurociência comportamental, para facilitar a modulagem de novos comportamentos.

**Palavras-chave:** Dispositivos móveis, aplicativo, hábitos.

# ABSTRACT

Currently, the average internet user spends about six hours a day online, due to the ease of accessing digital media and applications that are designed to be addictive to their users, using techniques such as content selection based on user interest. However, the digital environment was not created with the intention of being a harmful place that diminishes users' productivity. This work aims to create a mobile application to assist in the formation of new habits using techniques based on behavioral neuroscience to facilitate the shaping of new user behaviors.

**Keywords:** Mobile devices, application,habits.

# INTRODUÇÃO

De acordo com o relatório da Global Web Index (2017), usuários da *internet* estimaram passar cerca de seis horas e meia *online*. Além disso, é estimado que o uso de telefones celulares, para acessar a *internet* aumentou 1 hora e 17 minutos para 2 horas e 30 minutos. Durante o mesmo período, computadores e tablets registraram uma queda, mesmo que mínima, em seu tempo de uso.

O Brasil está entre um dos países a qual o tempo de uso da *internet* é maior que o das mídias tradicionais, como rádios e jornais (Global Web Index, 2017). Ademais, de acordo com o Relatório do Acompanhamento do Setor de Telecomunicações da Agência Nacional de Telecomunicações (Anatel), vinculada ao Ministério das Comunicações (MCom), o Brasil registrou 234,07 milhões de acessos móveis em 2020, representando um aumento de 3,26% em relação a 2019 (BRASIL, 2021). Assim, é notável que o acesso aos meios digitais por meio de telefones é algo de suma importância na rotina do brasileiro.

Dito isso, é válido afirmar que, o uso do celular nem sempre é positivo, pois, no ano de 2012, pesquisadores encontraram uma correlação entre o uso intensivo dos celulares e sintomas de depressão e taxa de suícidio em adolescentes. Além disso, um estudo de caso com estudantes de negócios em Bangladesh encontrou que o aumento da impaciência e a perturbação na vida diária, decorrido do uso excessivo de telefones, afetaram negativamente o desempenho acadêmico dos estudantes (SHOUKAT, 2019). Um possível questionamento a ser feito é como um ambiente, mesmo ao trazer tantas adversidades a seus usuários, ainda é utilizado por tantas horas no dia.

De acordo com Mendelsohn (2019), a maioria dos comportamentos realizados em um dia são ditados pelos hábitos. Ademais, essa também descreve o hábito como responsável pela rotina, comportamentos e até mesmo alguns processos cognitivos. Acrescenta-se que, em um estudo realizado em laboratório, quando um rato, treinado para puxar uma alavanca com o objetivo de receber uma recompensa, recebe uma punição, o comportamento habitual se torna menos provável de acontecer uma próxima vez. Dessa maneira, pode-se inferir que o uso intensivo de celular, mesmo ao afetar negativamente o usuário em diversas áreas de sua vida, mantêm-se elevado por se tratar de um hábito que não traz punições imediatas.

De acordo com Ostlund e Balleine (2008), vícios são, de uma maneira simplificada, hábitos, pois a natureza compulsiva da busca e execução do vício é coerente com os estudos e evidências empíricas que se acumularam ao longo de anos. Dito isso, vícios não são punitivos durante a execução desses, pelo contrário, o cérebro entende a ação realizada como positiva devido à alta carga dopaminérgica liberada pelo gatilho responsável pelo impulso inicial. Essa carga é liberada antes de ações que são reconhecidas como positivas pelo cérebro e toda vez que essa ação é repetida, e obtém resultados positivos, ela é reforçada (Schultz et al, 1993).

Para que uma ação seja considerada positiva pelo cérebro, esse tem que ter ciência dos efeitos positivos gerados por tal, um exemplo clássico, é o experimento realizado por Schultz (1993), em que primatas eram inicialmente ensinados que dentro de um recipiente existia suco de maçã, o que fazia com que esse reconhecesse a recompensa inicial. Após isso, uma luz e um botão eram colocados próximos ao recipiente, e esses primatas eram ensinados que sempre que a luz ligasse e o botão fosse apertado mais suco entraria no recipiente, formando assim um reforço para a criação daquele hábito. Mais tarde no teste, houve uma mudança, quando a luz acendesse e o primata apertasse o botão, o suco não seria despejado, devido a isso notou-se que os primatas continuavam repetindo a ação de apertar o botão quando a luz acendia. Porém, quanto mais tempo se passava sem a queda do suco, a ação de apertar o botão era menos recorrente. Ao analisar esse experimento, é notável que caso o macaco não fosse ensinado que o recipiente continhao suco de maçã, não existiria a motivação inicial para o aprendizado das próximas etapas, logo o reconhecimento da recompensa final é necessário para a motivação inicial.

Assim, é possível uma pessoa adicta “quebrar” o vício por meio do reconhecimento da recompensa final, e utilizar esse conhecimento como motivador, o que configura esse como uma comportamentos direcionados a um objetivo. Caso essa pessoa consiga reconhecer ações tomadas por hábito e utilizar dos comportamentos orientados ao objetivo, é possível moldar o comportamento associado ao gatilho por meio da degradação de contingência. No entanto isso é um desafio de ser feito no dia a dia (Mendelsohn, 2019).

Desta forma, esse projeto tem como objetivo geral o desenvolvimento de uma aplicação *mobile* que dê assistência na formação de um hábito escolhido pelo usuário, por meio do reconhecimento da recompensa final, gerado por um modelo de linguagem de grande escala; técnicas de *gameficação* durante o uso da aplicação; e um ambiente social para a formação do hábito, além de buscar um design moderno e uma boa experiência de usabilidade para o usuário.

Para tanto, os objetivos específicos são: avaliação dos estudos sobre hábitos no tópico da neurociência comportamental e desenvolvimento de uma aplicação *mobile* e suas *API*s para formação de hábitos.

# FUNDAMENTAÇÃO TEÓRICA

Nesta sessão será explorado de forma completa os conceitos supracitados, trazendo embasamento teórico e técnico dos recursos explorados para desenvolvimento do projeto. Tais como: estruturas de projetos, usabilidade de tecnologias assistivas e desenvolvimento de código com linguagens nativas *Android*.

## Tecnologia Assistiva

Caracterizada pelo comitê de ajudas técnicas (CTA), realizado pela secretaria dos direitos humanos da Presidência da República “Tecnologia Assistiva é uma área do conhecimento de característica interdisciplinar que engloba produtos, recursos, metodologias, estratégias, práticas e serviços. Tal área objetiva promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social” (NAZARI, 2017).

Dentro do contexto de tecnologias móveis, os recursos assistivos estão presentes no sistema operacional do *hardware*, cada fabricante disponibiliza configurações próprias que interagem com interfaces de aplicativos. Segundo (Hakobyan *et al*, 2013, p.514):

Dispositivos que são discretos ou aplicativos que estão incorporados em um dispositivo convencional, como um celular, podem ajudar as pessoas a se sentirem menos estigmatizadas ou rotuladas. Além disso, os sistemas assistivos geralmente são adaptáveis em várias plataformas móveis e podem oferecer suporte a várias deficiências.

### Acessibilidade em sistemas operacionais android

O *Android* oferece para os desenvolvedores uma documentação vasta sobre recursos nativos disponíveis através de bibliotecas *Java*, linguagem base para o desenvolvimento de implementações gerais de aplicativos móveis. Dentre estes recursos é abrangível a manipulação de interfaces visuais, sonoras e localização.

Desenvolvido inicialmente pela Android Inc, uma empresa fundada por Andy Rubin, Rich Miner, Nick Sears e Chris White. O sistema operacional foi criado com o objetivo de ser uma plataforma aberta e flexível para dispositivos móveis. Em 2005, a Google adquiriu a Android Inc, e continuou a desenvolver e aprimorar o sistema. Atualmente, o *Android* é mantido e gerenciado pela Google, que lança atualizações periódicas do sistema operacional, juntamente com suas versões personalizadas por fabricantes de dispositivos específicos (OLIVEIRA *et al*, 2014).

### Leitor de tela - *Talkback*

O *Talkback* é uma funcionalidade acessível que auxilia pessoas com deficiência visual, analfabetismo ou até dificuldades momentâneas de reconhecer os elementos visuais presentes nas aplicações *mobile*[[1]](#footnote-1). Este é um recurso da Google incluído em dispositivos *android*, tendo sua configuração especificada por cada fabricante, podendo ter alterações de gestos e comportamentos, dependendo da versão do *android* instalada dentro de cada *hardware*.

De acordo com a documentação presente na suportgoogle[[2]](#footnote-2), o leitor de tela após ser ativado oferece uma experiência de leitura dinâmica, sua navegação se dá através de toques na tela. Em versões mais recentes é possível configurar múltiplos toques com até 3 dedos, permitindo que o usuário deslize de cima para baixo ou da esquerda para direita, fazendo assim a movimentação do leitor de forma ordenada pelos elementos de visualização presentes na interface.

Atualmente também é disponibilizado a configuração de atalhos por meio de gestos, customizáveis de acordo com a necessidade do usuário, trazendo a possibilidade de rápido acesso de aplicações e recursos mais utilizados.

Todavia, em versões anteriores do sistema *android*, as configurações do *talkback* permitiam uma movimentação limitada, fazendo a leitura de apenas um dedo para deslizar entre os elementos, com movimentos de cima para baixo, controlando assim a leitura.

### Conversor de texto em voz

Outra funcionalidade a ser explorada no contexto de tecnologias assistivas no sistema *android* é a conversão de voz em texto, podendo ser usada em mídias como vídeos, *podcasts*, chamadas telefônicas e vídeo chamadas. Seu funcionamento consiste em converter dinamicamente o som em legendas que podem ser acompanhadas por usuários que possuem alguma deficiência auditiva ou até indisponibilidade momentânea de escutar o som transmitido por seu aparelho móvel. Sendo um recurso relativamente novo, de acordo com a documentação *android*, seu funcionamento é garantido de forma mais eficaz em *hardwares* com

versões superiores ao *android* 10. Sendo possível a ativação de legendas instantâneas em dispositivos *pixel* 2 configurados com idioma inglês, e dispositivos *pixel* 6 *e pixel* 6 *pro* em inglês, francês, alemão, italiano, japonês e espanhol, podendo também realizar a tradução automática por meio das legendas de mídias destes aparelhos.

Passível de configurações é possível ajustar o local da caixa de texto com as legendas, juntamente com o tamanho da fonte configurada no dispositivo, apresentando conexão com o *talkback,* sendo possível exibir o texto verbalizado por ele.

## *Kotlin*

*Kotlin* pode ser descrito como um projeto de código aberto sem custos, financiado pela Apache 2.0. Seu código é disponibilizado pelo *github* sendo mantido principalmente pelas equipes da JetBrains e Google. Seu desenvolvimento é multiplataforma, sendo disponibilizado para sistemas operacionais como iOS, Android, macOS, Windows, Linux, watchOS, entre outros conforme kotlinlang[[3]](#footnote-3).

Baseada em princípios de orientação a objeto a linguagem de programação *kotlin* é compilada pela máquina virtual  *Java*, apresentando vantagens em sua escrita como maior facilidade de manipulação de recursos, sintaxe mais simples e resumida. De acordo com developer[[4]](#footnote-4) popularizou-se principalmente para desenvolvimento de aplicações nativas *android*, sendo amplamente adotado por empresas como Netflix, Duolingo, Kindle, entre outras.

Portanto, neste projeto o desenvolvimento será voltado para linguagem de programação *kotlin*, sendo uma tecnologia atual e reconhecida por grandes empresas, tendo uma sintaxe simples e direta, com documentações atuais e de fácil acesso.

## Diretrizes para qualidade de código

Para o desenvolvimento de *softwares*, conceitos e diretrizes voltados para qualidade são amplamente utilizados, representando boas práticas aplicáveis

durante o desenvolvimento de um produto digital. Cada código tem suas particularidades, cabendo ao desenvolvedor garantir a qualidade, assim como citado na obra o mítico homem-mês por (Brooks, 1975, p.48), segundo ele:

O programador, como o poeta, trabalha sozinho. Mesmo hoje, cada código é um ato criativo de pensamento solitário. Ninguém pode antecipar com clareza qual será a sua forma definitiva.

Conforme o pensamento de Brooks, pode-se concluir que não existe uma definição única de como de desenvolver uma aplicação, todavia cabe ao programador decidir quais conceitos devem ser adotados de acordo com o escopo de seu projeto.

### Código Limpo – *Clean Code*

O conceito de *clean code* se entende como um conjunto de boas práticas aplicáveis no dia a dia de um desenvolvedor, seu intuito é manter um código limpo, de fácil manutenção. Criado por Robert Martin, a metodologia presente neste padrão de projeto foi descrita em sua obra Código limpo: Habilidades práticas de Agile Software, lançado no ano de 2008.

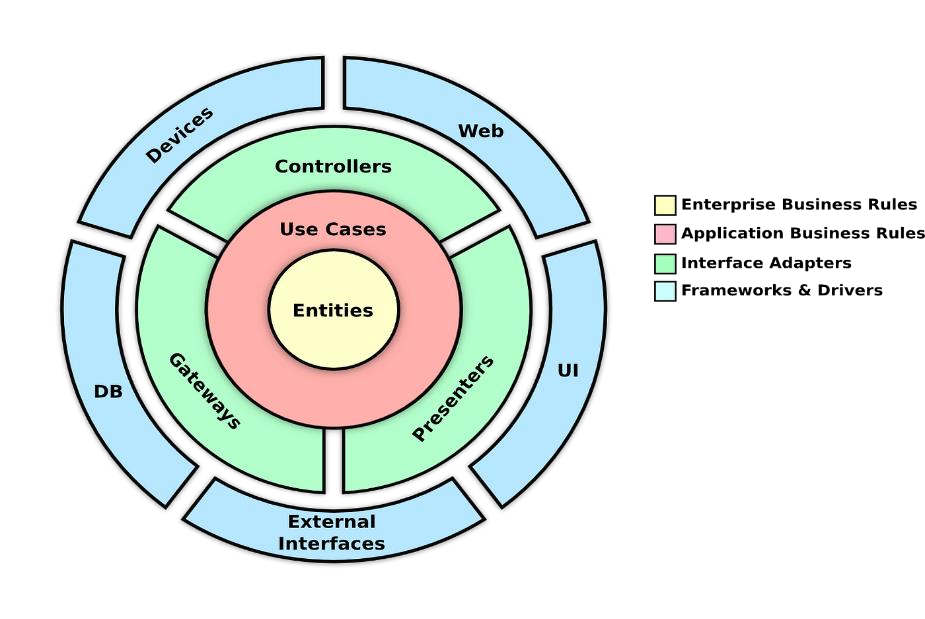
Neste projeto serão aplicados princípios utilizados na obra supracitada, como, nomes significativos, mantendo a nomenclatura de classes, objetos e variáveis de fácil entendimento e compreensão aos demais desenvolvedores. Formatação coerente das estruturas de código, visando manter o alinhamento das linhas para fácil visualização e testes de unidades, criando testes unitários de funções desenvolvidas para garantir seu funcionamento em diferentes cenários.

### Arquitetura Limpa – *Clean Architecture*

Arquitetura limpa se entende como um conjunto de conceitos que tendem a desenvolver camadas de código independentes, viabilizando a recursividade de classes e objetos, permitindo criar segmentações voltadas para cada etapa, desenvolvendo códigos testáveis e de fácil alteração do *software*, assim como mencionado na obra Arquitetura Limpa de Robert Martin.

Pode-se observar as seguintes camadas de desenvolvimento na figura 1:

**Figura 1** – Camadas arquitetura limpa



Fonte: Blog Medium, Disponível em: https://joaogbsczip.medium.com/clean-architecturereactjs-pt-2-fa1a166dcfea. Acesso em: 02 nov. 2023.

* *Entities*, conhecida como camada de entidades, tem como objetivo representar os objetos de negócio presentes na estruturação do projeto, encapsulando o comportamento, relacionado a conceitos específicos do domínio.
* *Use case,* reconhecida como camada de regras de negócio, apresentando os casos de uso presentes na aplicação.
* *Interface Adapters,* a camada de interface representa a programação de telas e funções que serão manipuladas pelo usuário final, realizando o desenvolvimento de futuras ações e comportamentos de *view*[[5]](#footnote-5).
* *Frameworks* e *Drivers*, são responsáveis pelas comunicações mais externas, utilizando recurso como banco de dados e chamadas de *api*, podendo depender de camadas anteriores como interfaces e regras de negócio.

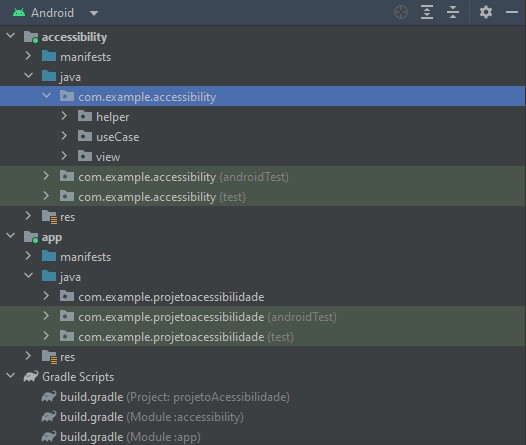
Diante disso, este projeto é baseado nos princípios de desenvolvimento supracitados, apresentando em seu desenvolvimento uma camada de *use case*, contendo regras de funcionamento para as funções implementáveis, exemplos utilizando interfaces interativas e independentes, juntamente com *frameworks* como o *Retrofit*[[6]](#footnote-6), que proporcionam maior facilidade para chamadas de *api*, compondo o conteúdo dos arquivos de *view*.

### Modularização

Componentização, termo análogo a modularização, referem-se ao ato de subdividir um projeto em módulos menores e independentes. Em um contexto de desenvolvimento *android*, ao subdividir um projeto, cada componente apresenta suas próprias configurações de *gradle*8, implementações e serviços (ELY, 2019).

Uma grande vantagem de utilizar essa abordagem é manter um versionamento independente, abordagem utilizada neste projeto para facilitar a implementação de recursos assistivos dentro de exemplos de aplicações *mobile*, conforme a figura 2:

**Figura 2** – Modularização de pacotes e arquivos no Android Studio.



Fonte: Autoria própria, 2023.

Nesta imagem é possível encontrar dois módulos distintos que compartilham o mesmo projeto, sendo *accessibility* o primeiro componente, que é responsável pelas implementações de acessibilidade, o segundo armazena os arquivos de visualização do *app*, sendo executado ao iniciar o projeto.

## *Github*

O *GitHub* é uma plataforma online de renome no universo do desenvolvimento de *software*, largamente adotada para gerenciamento de projetos. Sua funcionalidade central consiste em fornecer um ambiente unificado onde desenvolvedores podem depositar, controlar e colaborar no código-fonte de suas criações. Através do emprego do sistema de controle de versão *Git,* o *GitHub* capacita equipes a monitorarem e gerirem as alterações efetuadas no código ao longo do tempo, o que facilita uma colaboração sinérgica e a manutenção de projetos de *software,* assim como citado em github.com[[7]](#footnote-7).

## Cenários de teste

Existem diferentes metodologias que podem ser aplicadas para o teste de um aplicativo, cada uma contendo suas particularidades, assim são aplicadas de acordo com a necessidade do projeto, garantindo a integridade e usabilidade de um *software* antes de seu lançamento. A seguir estão alguns exemplos de testes que foram aplicados, conforme a figura 3:

**Figura 3** – Tipos de teste de software

Tabela

Descrição gerada automaticamente com confiança média

Fonte: Blog Medium, Disponível em: https://medium.com/@annerocha\_qa/principais-tipos-deteste-de-software-4aeeb7fd23f1. Acesso em: 03 nov. 2023.

Testes estruturais conhecidos também como testes de caixa branca foram realizados através do código, desenvolvidos pela própria equipe de desenvolvimento, geralmente usados em aplicações que possuam o consumo de serviços externos.

Os testes funcionais são realizados pelo profissional de teste, sendo ele um teste da aplicação, que busca evidenciar sua experiência ao utilizar o programa. Um teste não funcional garante a estabilidade da aplicação, podendo observar a quantidade de dados máximos a serem trafegados, a segurança e vazamento de informações sensíveis, entre outros aspectos que englobam a segurança.

Testes relacionados a mudança podem ser utilizados de diferentes formas, geralmente são associados a correções direcionadas ao desenvolvedor, que devem ser corrigidas e testadas novamente (DELMARO, 2013).

# METODOLOGIA

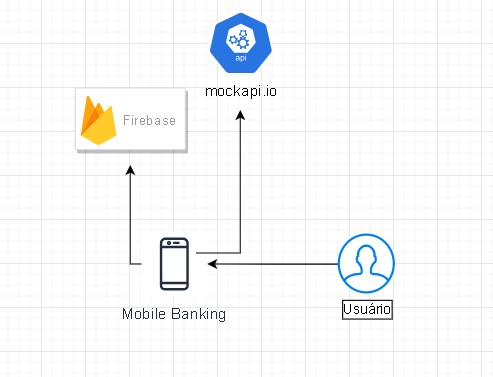
Para desenvolvimento deste projeto foi necessário a criação de uma aplicação *mobile android* base. Durante seu desenvolvimento foram incorporadas funções do módulo *accessibility*, que abriga as ferramentas desenvolvidas para simplificar a integração das tecnologias assistivas previamente estudadas.

Assim, a seguir serão apresentados os caminhos utilizados para desenvolvimento deste projeto, juntamente com as considerações acerca dos resultados desejados.

## Projeto base

Para incorporar as funcionalidades de tecnologia assistiva desenvolvidas neste projeto, foi necessário criar uma aplicação *Android* que oferecesse interfaces interativas, com o objetivo de representar um aplicativo móvel convencional, amplamente utilizado pelos usuários *mobile*. Assim, foi desenvolvido um aplicativo de interface bancária, integrando em sua estrutura a interação com serviços externos, permitindo a exemplificação de cenários realistas enfrentados por usuários cotidianamente. Abaixo, a figura 4 ilustra o fluxo da aplicação:

**Figura 4** – Estrutura da aplicação



Fonte: Autoria própria, 2023.

Acima é possível visualizar a estrutura do projeto base, denominado *Mobile Banking*, sendo assim o serviço disponibilizado ao usuário, o qual representa o público que irá acessar a aplicação fornecida. Pode-se destacar também o uso de dois *frameworks*, que serão utilizados para requerer funcionalidades externas.

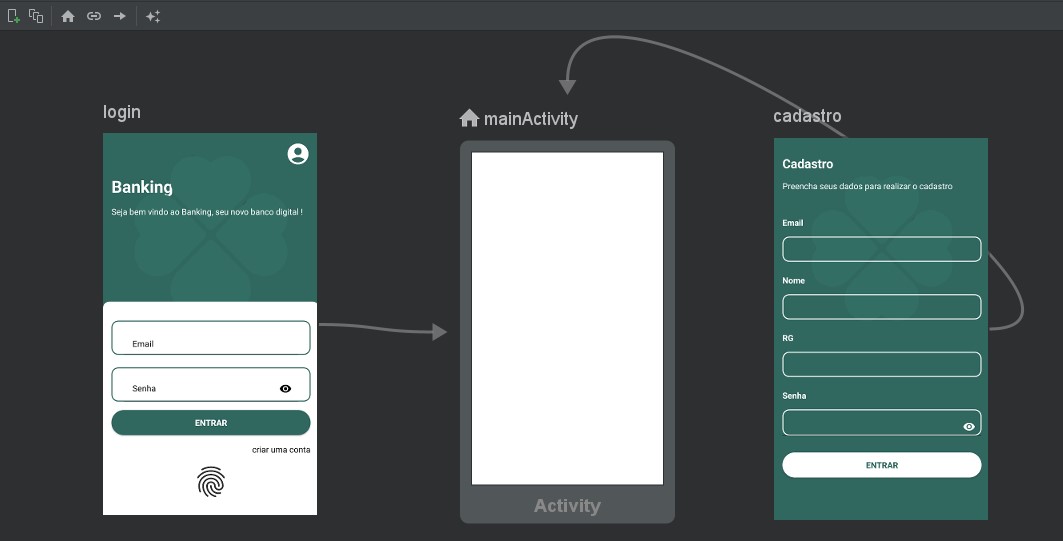
Desenvolvido como um serviço de computação em nuvem voltado para plataformas de desenvolvimento de aplicativos, o *firebase* é um *framework* disponibilizado pela equipe Google. Sendo implementado neste projeto, para consumo de recursos como autenticação de usuários através de chaves *key*, criadas na realização de novos cadastros, juntamente com armazenamento para banco de dados, responsáveis por reunir características de novos usuários.

Outro recurso explorado para criação de interfaces dinâmicas, foi o consumo da *api* *mock.io*, nela é possível gerar *jsons*[[8]](#footnote-8) dinâmicos, trazendo dados mocados conforme a configuração do projeto, desta forma é possível receber listas de dados que serão aplicados durante o desenvolvimento como alternativa para popular objetos criados na interface visual.

## Jornada da aplicação

Para a implementação do módulo de acessibilidade, foi desenvolvida uma jornada padrão que inclui elementos de visualização comumente utilizados na composição do arquivo de *layout XML*11. A atividade principal, inicialmente, abriga dois fragmentos, o primeiro é destinado ao *login*, permitindo aos usuários acessar a aplicação, enquanto o segundo é voltado para o cadastro, possibilitando que novos usuários se inscrevam no sistema, caso ainda não tenham uma conta, conforme a figura 5:

**Figura 5** – Jornada inicial



Fonte: Autoria própria, 2023.

Ao analisar as interfaces fornecidas é possível encontrar componentes visuais que não apresentam semânticas apropriadas para usuários de tecnologia assistiva,

dentre eles é possível destacar a presença de títulos, estes que visualmente se destacam por apresentar características com fontes maiores, e estilo mais robusto em comparação aos demais.

Assim, para um usuário que não necessita de nenhum recurso de acessibilidade é fácil identificar o começo da tela baseado na configuração do título, representando o início das informações fornecidas, todavia para um usuário de acessibilidade é preciso que o título seja verbalizado com a literal apropriada, para identificação de onde a aplicação é iniciada, dando melhor direcionamento entre o começo e o final da leitura de tela.

Neste mesmo fluxo é possível encontrar na tela de *login*, dois objetos clicáveis que fogem da estética padrão de um botão, que seria a alternativa técnica comumente utilizada para direcionar um usuário para um próximo passo. Todavia é normalmente encontrado em *design[[9]](#footnote-9)* de interface, componentes que apresentam a função de um botão, porém precisam ter uma estética diferente para tornar a tela mais harmônica ao *layout* proposto.

Neste caso podemos encontrar esta situação em *links* ou em imagens que remetam a uma ação interativa, a interface fornecida apresenta ambas as situações, é possível encontrar no canto inferior direto abaixo da caixa de edição um *link* que carrega o usuário para realizar um novo cadastro, também é visível a imagem de uma digital que possibilita uma entrada alternativa para usuários que possuam aparelhos com essa tecnologia de autenticação.

Para ambos os casos, não é fornecido pelo sistema operacional uma semântica apropriada, pois ambos os elementos são declarados de forma genérica, então para que seja possível tornar-se acessível, é preciso atribuir as características assistivas de um objeto clicável em ambos os casos.

## Aplicando elementos assistivos

Referenciando as problemáticas supracitadas anteriormente, foram desenvolvidas alternativas técnicas que solucionam os problemas encontrados, podendo destacar inicialmente a verbalização de componentes do tipo título, logo abaixo na figura 6 está disponibilizado uma função aberta que pode ser instanciada

através do objeto *AccessibilityCommons*, presente no módulo *accessibility*, conforme a figura 6:

**Figura 6** – Código de aplicação, função delegada para acessibilidade de títulos



Fonte: Autoria própria, 2023.

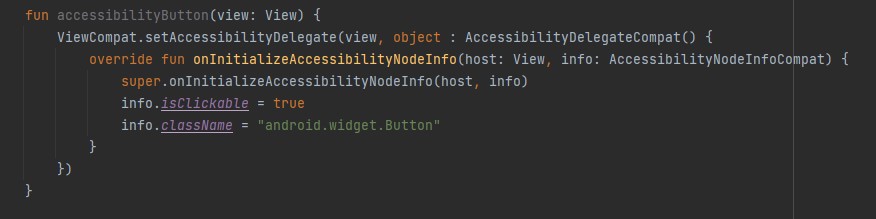
A função *addTitleSemantics* é responsável por configurar informações de acessibilidade para uma determinada *view* em um contexto de desenvolvimento de aplicativos móveis. Ela utiliza a classe *ViewCompat* para atribuir um *AccessibilityDelegate* personalizado. O *AccessibilityDelegate* é um mecanismo que permite a personalização das informações de acessibilidade de uma *view* para melhorar a experiência de usuários que dependem de tecnologias assistivas, como leitores de tela.

Dentro desse *AccessibilityDelegate*, a função substitui o método *onInitializeAccessibilityNodeInfo*, que é chamado durante a inicialização das informações de acessibilidade para a *view*. Ela mantém o comportamento padrão chamando o método correspondente na classe pai por meio de *super.onInitializeAccessibilityNodeInfo(host, info).*

Assim, a função configura a propriedade *isHeading* do objeto *info* como *true*, essa configuração indica que a *view* em questão representa um título ou cabeçalho, fornecendo um contexto semântico importante para os usuários de tecnologia assistiva. Essa ação visa aprimorar a compreensão e a navegabilidade da interface do usuário para esse público-alvo.

Seguindo esse contexto de desenvolvimento para aplicação acima, foi desenvolvido uma função que pode ser aplicada aos elementos de interface que apresentam comportamento de botões, todavia são desenvolvidos com componentes diferentes de acordo com o *layout*, assim o resultado desta codificação se apresenta na figura 7:

**Figura 7** - Código de aplicação, função delegada para acessibilidade em botões

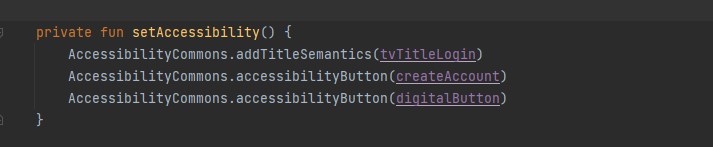


Fonte: Autoria própria, 2023.

Neste caso o objeto *info* recebe a propriedade *isClickable* como *true*, indicando que o elemento de *view* é clicável pelo usuário, na linha subsequente ocorre a atribuição da *className* do objeto para classe pertencente aos botões, garantindo assim a definição correta do contexto.

Por fim para que seja possível utilizar as funções evidenciadas acima, o módulo *accessibility* deve ser importado no projeto desejado, neste caso foi adicionado a referência do arquivo *AAR*[[10]](#footnote-10) no módulo que continha a aplicação *mobile*, posteriormente ao sincronizar o projeto, foi possível instanciar o objeto que contém as implementações desejadas, como podemos ver abaixo na figura 8:

**Figura 8** – Código de implementação das funções de acessibilidade



Fonte: Autoria própria, 2023.

Como é possível observar ao instanciar o objeto *AccessibilityCommons* é necessário apenas encontrar a função desejada e adicionar ao seu construtor a *view* que precisa ser customizada.

# RESULTADOS E DISCUSSÕES

Nesta sessão será discutido os resultados e conclusões acerca dos testes de usabilidade realizados após a implementação do módulo de acessibilidade, este que se encontra na plataforma *github* e pode ser acessado através do *link* github.com*[[11]](#footnote-11)*.

## Configuração do ambiente de testes

Utilizando a *IDE*[[12]](#footnote-12) *Android Studio* para alocação e desenvolvimento do código deste projeto, foi realizado a configuração do ambiente de testes através das ferramentas por ele disponibilizadas. Para a instalação do aplicativo foi necessário habilitar as configurações de desenvolvedor de um *device*[[13]](#footnote-13), sendo ele um *hardware* com sistema operacional e interface *android* 12*.*

Após ocorrer o processo de instalação do *software*, foram mapeados os cenários e eventos que devem ser observados para concluir a efetividade das funções atribuídas.

E por fim foi preciso conferir a instalação dos recursos de acessibilidade inclusos do *hardware* a ser utilizado, sendo ele a versão 14.0.1 lançada pela equipe do sistema operacional *android*. Este projeto pode ser encontrado através do repositório github.com[[14]](#footnote-14), contendo o aplicativo mencionado acima.

## Aplicação de testes de *software*

Os testes manuais atribuídos neste projeto foram divididos e documentados em duas etapas, inicialmente foi evidenciado por meio da gravação de tela do *device*, o aplicativo sem a inserção de nenhum tipo de recurso assistivo aplicado ao código.

O intuito desta etapa inicial é proporcionar evidencias que exemplifiquem uma leitura de *talkback*, cujos elementos de interface não apresentem a semântica apropriada impedindo que o usuário tenha uma experiência satisfatória no uso do

*software*. Esta evidência pode ser acessada através da plataforma *youtube*[[15]](#footnote-15) no *link* youtube.com[[16]](#footnote-16)*.*

Em seguida foram mapeados os problemas encontrados nesta execução, sendo possível destacar a não verbalização dos títulos com a semântica correta, tornando a execução confusa e não padronizada, a semântica de objetos clicáveis sem a verbalização de botão e a não verbalização de ícones que apresentam interatividade com o usuário.

Posterior ao primeiro cenário, a aplicação agora é testada após as implementações do módulo *accessibility*, sua evidência através da gravação da tela foca principalmente na navegação linear da aplicação, após as correções necessárias. Assim, também é disponibilizada através do *link*: youtube.com[[17]](#footnote-17), desta forma é possível comparar as mudanças ocorridas de uma execução para a outra.

# CONCLUSÃO

Ao finalizar a implementação do projeto em um aplicativo nativo *android*, foi possível concluir que o módulo de acessibilidade desenvolvido no decorrer deste artigo proporciona um desenvolvimento facilitado de recursos assistivos.

Em poucos passos foi possível modificar a semântica de objetos fazendo com que as tecnologias assistivas do sistema operacional interpretem os elementos de *view* conforme o esperado. Contudo, é possível considerar não apenas as funcionalidades atribuídas, mas também a economia de linhas de código visando a qualidade e recursividade das funções.

Viabilizando a progressão deste projeto, os próximos passos serão mapear novas funcionalidades que podem ser atribuídas a este módulo, o tornando cada vez mais completo para os mais variados cenários, também buscar atribuir ao projeto uma estrutura que remeta a uma biblioteca *android*, facilitando o uso destes recursos criados a novos aplicativos.

# REFERÊNCIAS

BRASIL. **Brasil tem mais de 234 milhões de acessos móveis em 2020**. Ministério das Comunicações, 2021. Disponível em: https://www.gov.br/pt-br/noticias/transito-e-transportes/2021/04/brasil-tem-mais-de-234-milhoes-de-acessos-moveis-em-2020. Acesso em: 9 jun. 2024.

GLOBAL WEB INDEX**. Digital vs Traditional Media Consumption: analyzing time devoted to online and traditional forms of media at a global level, as well as by age and across countries**. Insight Report, Q1 2017. Disponível em: https://www.gwi.com/hubfs/Digital\_vs\_Traditional\_Media\_Consumption.pdf. Acesso em: 09 jun. 2024.

MENDELSOHN, A. I. **Creatures of habit: the neuroscience of habit and purposeful behavior**. Biological Psychiatry, v. 85, n. 11, p. e49–e51, 2019. Disponível em: https://doi.org/10.1016/j.biopsych.2019.03.978. Acesso em: 9 jun. 2024.

OSTLUND, S. B.; BALLEINE, B. W. **On habits and addiction: an associative analysis of compulsive drug seeking**. Drug discovery today. Disease models, v. 5, n. 4, p. 235–245, 2008. Disponível em: https://doi.org/10.1016/j.ddmod.2009.07.004. Acesso em: 09 jun. 2024.

Schultz, W., Apicella, P., & Ljungberg, T. (1993). **Responses of monkey dopamine neurons to reward and conditioned stimuli during successive steps of learning a delayed response task**. The Journal of neuroscience : the official journal of the Society for Neuroscience, 13(3), 900–913. https://doi.org/10.1523/JNEUROSCI.13-03-00900.1993

SHOUKAT, S. **Cell phone addiction and psychological and physiological health in adolescents**. EXCLI journal, v. 18, p. 47–50, 2019.

1. Tecnologias móveis [↑](#footnote-ref-1)
2. Disponível em <https://www.suportgoogle.com>. Acesso em: 01 nov. 2023. [↑](#footnote-ref-2)
3. Disponível em <http://www kotlinlang.org>. Acesso em: 01 nov. 2023. [↑](#footnote-ref-3)
4. Disponível em <http://www.developer.android.com>. Acesso em: 01 nov. 2023. [↑](#footnote-ref-4)
5. Termo utilizado para elementos de visualização, como layouts e telas. [↑](#footnote-ref-5)
6. Framework utilizado para realizar requisições de serviços Rest. 8 Sistema de automação e compilação de código. [↑](#footnote-ref-6)
7. Disponível em <https://docs.github.com/pt/get-started/quickstart/hello-world>. Acesso em: 03 nov. 2023. [↑](#footnote-ref-7)
8. Formato baseado em texto padrão para representar dados estruturados 11 Estrutura visual para interface. [↑](#footnote-ref-8)
9. Idealização, projeção ou concepção do layout final. [↑](#footnote-ref-9)
10. Archive do Android (AAR) é um formato de arquivo usado para distribuir bibliotecas Android. [↑](#footnote-ref-10)
11. Disponível em <https://github.com/Kauane-SP/model-accessibility>. Acesso em: 03 nov. 2023. [↑](#footnote-ref-11)
12. Ambiente de desenvolvimento integrado. [↑](#footnote-ref-12)
13. Aparelho de hardware utilizado para testes. [↑](#footnote-ref-13)
14. Disponível em <https://github.com/Kauane-SP/app-implementation-accessibility>. Acesso em: 03 nov. 2023. [↑](#footnote-ref-14)
15. Plataforma de vídeos online. [↑](#footnote-ref-15)
16. Disponível em <https://youtube.com/shorts/yHz4QI\_-7\_Q?feature=share>. Acesso em: 03 nov.

    2023. [↑](#footnote-ref-16)
17. Disponível em < https://youtu.be/S0nqoTZuFOo?feature=shared >. Acesso em: 03 nov. 2023. [↑](#footnote-ref-17)