

Reading the News on Android TV: framework para exibição de notícias em smart TVs

Trabalho de Conclusão do Curso de
Tecnologia em Sistemas Para Internet

Antonio Paulo Serpa Antunes
Orientador: Fabio Yoshimitsu Okuyama
Coorientadora: Silvia de Castro Bertagnolli

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Porto Alegre
Av Cel Vicente, 281, Porto Alegre – RS – Brasil

pauloserpaantunes@gmail.com, fabio.okuyama@poa.ifrs.edu.br,
silvia.bertagnolli@poa.ifrs.edu.br

Resumo. *Este TCC descreve a trajetória para a construção de um aplicativo para smart TVs. Este processo incluiu uma pesquisa sobre este mercado e suas principais plataformas, um levantamento de sistemas acadêmicos desenvolvidos para aparelhos de TV e pesquisas sobre o design para telas de grandes dimensões. A partir destes estudos foi projetado um framework de código-fonte aberto em linguagem Java que permite a criação de apps que exibam resumos de notícias para dispositivos Android TV.*

1. Introdução

A edição de setembro de 2022 da Pesquisa Nacional por Amostra de Domicílios (PNAD) Contínua do Instituto Brasileiro de Geografia e Estatística (IBGE) apurou que em 2021 o acesso à internet chegou a 90% dos lares do país e que houve uma transformação significativa nos dispositivos usados para este acesso: o segundo dispositivo mais usado no acesso à internet passou a ser a televisão, com 44,4% de presença nas residências, ultrapassando o computador, com 42,2% (IBGE, 2022). Através das *smart TVs*, *streaming devices* ou *set-top boxes*¹, nossos televisores nos últimos anos passaram a se conectar à internet, consumindo conteúdo multimídia.

A partir desta observação, buscou-se investigar como se configura o mercado de *smart TVs* e o ecossistema de desenvolvimento de software para estes equipamentos, em uma pesquisa exploratória em bases de dados, livros técnicos, sites de notícias e de fabricantes, juntando informações técnicas e de mercado. Em um segundo momento, através de pesquisas em bases de dados acadêmicas, buscou-se identificar trabalhos de desenvolvimento de software para estes dispositivos feitos por pesquisadores da área da Ciências da Computação. O pequeno número de projetos encontrados nos motivou a desenvolver uma aplicação para estes aparelhos, de forma a compreender esta tecnologia e explorar novos usos para ela.

Este artigo descreve o caminho traçado para o desenvolvimento de um *framework* chamado *Reading the News on Android TV*, que permita a construção de aplicações para

¹ O significado de termos técnicos como *streaming device* e *set-top box* foram inseridos em um glossário, disponível no Apêndice A.

a visualização de resumos de notícias na tela de um aparelho de televisão conectado à internet. Entendemos *framework* como “uma estrutura genérica estendida para criar um subsistema ou aplicação mais específicos” (Sommerville, 2018, p. 414). A construção do *framework* objetiva ainda:

- Identificar os desafios de programar para *smart TVs* e em particular estudar as peculiaridades do ponto de vista da experiência de usuário.
- Extrapolar a principal forma de uso do dispositivo, basicamente usado com um *media center* para consumo de *streaming* de áudio e vídeo.
- Criar um sistema que permita navegar e ler resumos de notícias, apresentando assim uma contribuição para a área da pesquisa aplicada ao jornalismo digital.
- Contribuir com a comunidade de software livre, disponibilizando o código-fonte da aplicação².
- Compreender particularidades do desenvolvimento de software livre, como versionamento de código e a documentação do sistema.

As principais informações coletadas nas etapas de pesquisa e as especificações técnicas e de design da aplicação estão descritas a seguir, bem como as lições aprendidas neste processo.

2. Referencial teórico: Smart TVs

A *smart TV* é uma evolução dos tradicionais aparelhos de televisão que, com o processo de digitalização e de conectividade, ganhou novas funcionalidades. Segundo Alam, Khusro e Naeem (2017, p. 35, tradução nossa), “a *smart TV* mudou o sistema legado de TV, ao fornecer recursos adicionais de processamento e conectividade com a internet. A *smart TV* está se tornando um *hub* central para entretenimento doméstico tanto em países desenvolvidos como em desenvolvimento”³.

As primeiras referências a estes novos usos para os aparelhos de televisão datam da década de 90 do século XX. Em 1994 foi feito o primeiro registro, na França, de um “Sistema de processamento de dados para televisão em rede digital ou analógica” (GATTO; BERTRAND, 1994). A tecnologia chegaria ao mercado na década seguinte, quando a Hewlett-Packard Company lançou em 2007 o HP MediaSmart TV, um televisor capaz de acessar conteúdos transmitidos pela internet ou através de uma rede local de computadores rodando o sistema operacional Windows Vista (REID, 2007).

O segmento ganharia tração na década seguinte, com a entrada de novos fabricantes no mercado e a adoção de novas tecnologias. Entre elas, se destaca o lançamento em 2010 do Google TV (GOOGLE NEWS FROM GOOGLE, 2010). Ao portar o sistema operacional Android para televisores, o Google tornou as *smart TVs* mais parecidas com *smartphones*: no lugar de rodar apenas aplicativos pré-instalados, elas passaram a rodar *apps* escolhidos pelo usuário, entre as opções disponíveis em um repositório.

² Durante todas as etapas de construção deste aplicativo, ele esteve disponível para download. O endereço do repositório é: https://github.com/psantunes/Reading_the_News_on_Android_TV.

³ “The *smart TV* has changed the legacy TV system by providing additional features of processing and internet connectivity. *Smart TV* is becoming a central hub for home entertainment in both developed and developing countries”.

Dados coletados pelo site Statista (2023) mostram que o mercado mundial de *streaming* para *Smart TV* em 2020 estava distribuído como mostra a Figura 1.

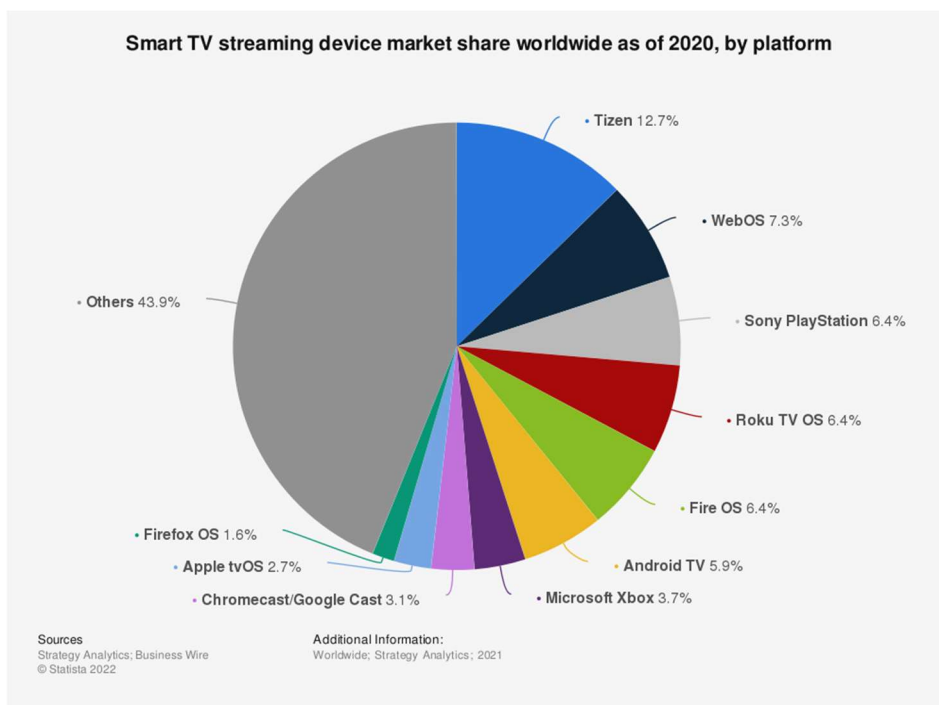


Figura 1. Participação de mercado de dispositivos de streaming para Smart TV no mundo, em 2020, por plataforma

Observa-se um mercado extremamente fragmentado, liderado pelos sistemas de duas das principais fabricantes de TV do mundo, a Samsung (Tizen) e a LG (WebOS). A Figura 1 mostra ainda que 10 empresas dividem 55,1% do mercado, mas ainda existem outros 43,9% divididos em fatias ainda menores por outros concorrentes. A segmentação do mercado lembra bastante o mercado de telefonia móvel antes de sua concentração entre Android e iOS, quando cada fabricante de telefone tentava impor o seu padrão de sistema operacional.

A seguir destacamos as plataformas com maior participação de mercado, seu histórico e suas características:

- **Tizen:** Em 2011, a Linux Foundation propôs a criação de um sistema operacional *open source* para dispositivos móveis. O projeto vinha para suceder outros projetos baseados em Linux que não vingaram e recebeu o apoio da Intel e da Samsung (JAYGARL; LUO; KIM; CHOI; BRADWICK; LANSDELL, 2014). O Tizen acabou preterido na disputa de mercado para *smartphones* e *tablets*, mas virou líder de mercado para *smart TVs* (ver figura 1), embarcado nos televisores da Samsung desde 2015 (SAMSUNG NEWSROOM, 2015). O Tizen roda *web apps* em HTML5, aplicações .NET e aplicações nativas na linguagem C (TIZEN PROJECT, 2012).
- **WebOs:** O webOS é o sistema operacional embarcado nos televisores de empresa sul-coreana LG desde 2014, substituindo sua antiga plataforma, chamada NetCast (LG ELECTRONICS, 2023a). O sistema, desenvolvido sobre Linux, foi construído originalmente pela Palm e vendido para a LG em 2013 por sua controladora, a HP (HP..., 2013). Em 2018, a LG lançou uma versão *open source*

da plataforma (LG ELECTRONICS, 2023b) e em 2021 a liberou para uso por fabricantes concorrentes (KLEINA, 2021). É possível construir aplicações para o sistema usando HTML e JavaScript e incorporar *frameworks* como Enact, jQuery e Node no desenvolvimento.

- **Roku OS:** É a plataforma da companhia de hardware homônima norte-americana, nascida em 2002, inicialmente como um sistema de DVR. Em 2007, a Roku se associou com a Netflix, passando a produzir *streaming devices* e se tornando um dos principais nomes do mercado. Na década seguinte, a empresa licenciou o Roku OS, baseado em Linux, também para *smart TVs* (BUTLER, 2020). No Brasil, encontramos televisores rodando o Roku OS das marcas Philco, Semp, AOC e TCL (ROKU INC, 2023a). Para desenvolver aplicações para o sistema é usada a linguagem interpretada BrightScript (ROKU INC, 2023b).
- **Fire TV:** A marca reúne uma série de dispositivos licenciados pela gigante de e-commerce Amazon desde 2014, como *streaming devices*, *set-top boxes* e uma linha de *smart TVs* produzida por empresas parceiras, integrando tecnologias populares da companhia, como o assistente de voz Alexa (AMAZON Fire TV, 2023). O sistema operacional Fire OS é baseado em Android, permitindo à plataforma rodar aplicativos feitos para Android TV, apenas demandando algumas modificações. A Amazon também oferece um *framework* para desenvolvimento de aplicações em Java e permite também a execução de *web apps* com HTML e Cordova (AMAZON.COM INC, 2023).
- **Android TV:** Apresentado pela primeira vez em 2014, o Android TV foi lançado pela Google em substituição a plataforma da Google TV. O sistema é uma versão do sistema operacional Android otimizada para uso em televisores (TREBILCOX-RUIZ, 2016). Desde então, o sistema passou a ser adotado por fabricantes como Panasonic, Philips, Sharp, Sony, TCL e Toshiba, entre outros, e é encontrado ainda em *set-top boxes* de empresas como Xiaomi e Intelbras. É possível desenvolver aplicações para a plataforma nas linguagens Java e Kotlin.
- **Google Cast:** O Google Cast é um protocolo desenvolvido pelo Google que permite a conexão de dispositivos como *smartphones* à *players* de mídia ou televisores. A tecnologia pode ser incorporada a aplicativos Android, iOS, Windows e macOS para a transmissão, e incorporada a televisores e *set-top boxes* para a recepção de dados (GOOGLE Cast, 2023). O dispositivo mais popular a adotar a tecnologia é o Chromecast, da própria Google, que permitiu que milhões de televisores antigos passassem a ser usados como *smart TVs*. Entre 2013 e 2017, foram vendidos mais de 55 milhões de aparelhos Chromecast ou outros dispositivos com a tecnologia embarcada (COLLINS, 2017).
- **tvOS:** O tvOS é o sistema operacional usado nos *set-top boxes* Apple TV, da empresa norte-americana Apple Inc. Apesar do aparelho ser comercializado desde 2007, o sistema atual data de 2015, quando a empresa apresentou uma adaptação do iOS para televisores e liberou sua plataforma para rodar software de terceiros. Assim como os demais produtos da Apple, o tvOS roda aplicações escritas em Objective-C, Swift e AppleScript (TVOS, 2023).
- **Outras plataformas:** Além das plataformas citadas acima, diversas outras ainda são usadas e estão embarcadas em televisores ou outros dispositivos. A companhia

japonesa Panasonic na década passada usava em seus televisores o descontinuado sistema Firefox OS, da Mozilla (MOZILLA, 2015). É importante destacar ainda que alguns consoles de videogame também agregam serviços de *streaming*, como o Xbox, da Microsoft, e o PlayStation, da Sony, disputando parte deste mercado. Existem ainda plataformas licenciadas para operadoras de TV por assinatura, soluções de código-fonte aberto e soluções de outras empresas que não têm penetração no Brasil.

Este cenário de divisão é um grande entrave para quem quer produzir programas para *smart TVs*. Em artigo publicado em 2022, Goetzenberger, Mooney e Riefler observam que a fragmentação deste mercado é um empecilho para produtores de conteúdo que desejam portar seus serviços para estas plataformas. Os autores conduziram uma série de entrevistas com representantes de empresas que atuam neste segmento (como canais de TV por assinatura e serviços de mídia online) nos Estados Unidos e na União Europeia para entender os critérios que os levam a escolher entre adotar um sistema ou outro, no que chamaram de “dilema dos provedores de conteúdo”⁴. A ampla maioria dos entrevistados entende que o mercado de sistemas operacionais para *smart TVs* é relevante para seus negócios, mas apontam três características negativas nelas: baixa usabilidade (no sentido de que cada dispositivo tem sua própria lógica de operação), baixa performance e interface do usuário ruim. Perguntados sobre os fatores que os levariam a escolher entre investir em um sistema e não em outro aparecem, empatados, o alcance da plataforma e motivos comerciais. Em terceiro lugar, os empresários destacaram o custo de desenvolvimento e manutenção de um *app* (GOETZENBERGER; MOONEY; RIEFLER, 2022, p. 54).

Falando especificamente em plataformas, os entrevistados mostraram interesse no Tizen, por ser o sistema com maior participação do mercado. Um entrevistado elogiou o Fire OS, por sua performance. Outros dois entrevistados demonstraram interesse no Android TV, por acreditarem a plataforma do Google ganhará terreno nos próximos anos em razão de muitas empresas estarem lançando *set-top boxes* que rodam o sistema e por acreditar que os consumidores gostam de acessar a Google Play Store – a loja de aplicativos que já é amplamente conhecida dos usuários de *smartphone* (GOETZENBERGER; MOONEY; RIEFLER, 2022, p. 55-57).

3. Pesquisas acadêmicas sobre aplicações para smart TVs

Complementando o levantamento bibliográfico, foi feita uma pesquisa exploratória em cinco bases de dados acadêmicas por projetos de software para *smart TVs*. A busca se concentrou em trabalhos produzidos a partir de 2015 em língua portuguesa, espanhola e inglesa, através de pesquisa por palavras-chave como “aplicativo para *smart TV*”, “*smart TV apps*” e suas derivações.

É importante destacar que no Brasil ocorreu um grande debate acadêmico e científico referente a criação do Sistema Brasileiro da Televisão Digital (SBTD) e do padrão Ginga, que permite exploração e desenvolvimento de aplicações interativas usando o sinal de TV digital. Nesta tecnologia, o software é enviado pelos transmissores de televisão e, portanto, seu sucesso depende de investimentos em conteúdo multimídia

⁴ Tradução nossa para “*The Content Providers’ Dilemma*” (GOETZENBERGER; MOONEY; RIEFLER, 2022).

por parte das empresas de radiodifusão, o que até o momento não se efetivou (LEAL, 2019). As pesquisas que exploram esta tecnologia (bem como outros protocolos híbridos) não são objetos desta pesquisa e foram descartadas.

A Tabela 1 reúne apenas trabalhos localizados relacionados ao desenvolvimento de aplicações para *smart TVs* – outros tipos de pesquisa foram aproveitados no referencial teórico ou descartados (por exemplo, pesquisas sobre segurança do dispositivo e projetos de *games*).

Tabela 1. Trabalhos sobre aplicativos para *smart TVs* por repositório

Repositório	Resultados
Google Acadêmico	10
Biblioteca Digital Brasileira de Teses e Dissertações (BDTD)	1
Biblioteca do IFRS	1
Lume (UFRGS)	0
SciELO	0

Classificando as pesquisas por país, observou-se que três delas foram produzidas por brasileiros. Os demais trabalhos são de pesquisadores da Grécia, Índia, Chile, Equador, Espanha e Estados Unidos. Os trabalhos são de diversos tipos: cinco são TCCs, cinco são artigos e dois são dissertações. Um dos TCCs foi desenvolvido no Campus Rio Grande do IFRS.

Das 12 pesquisas selecionadas, quatro descrevem *apps* criados para o sistema Android TV, quatro para Tizen, três para o descontinuado sistema Samsung Smart TV e um dos trabalhos descreve o desenvolvimento de uma aplicação para Chromecast. Os projetos acadêmicos localizados concentram em sua maioria projetos para a área de educação e saúde, compreendendo que o dispositivo é mais fácil de ser usado por idosos e pessoas com necessidades especiais, como crianças com síndrome de Down. Apenas um dos softwares desenvolvidos está disponível em um repositório público: é um sistema de mensageria chamado Chime (LEEF; NAGREE, 2016).

4. Proposta de aplicativo para leitura de notícias

Levando em consideração o pequeno número de trabalhos acadêmicos descrevendo o desenvolvimento de aplicativos para *smart TVs*, propomos a construção de um software específico para este tipo de dispositivo, que não seja simplesmente um reproduzidor de arquivos de mídia, apresentando um uso distinto para o dispositivo.

Após buscas em repositórios como a Google Play e a Aptoide TV por aplicativos para *smart TV*, observou-se ainda que não existem produtos noticiosos pensados para o dispositivo. O que se encontrou foram aplicativos de grandes empresas de mídia que funcionam basicamente como galerias de vídeos.

Em razão disto, imaginou-se um *app* que exibisse na tela resumos de notícias – compostas por informações como título, resumo e foto (os campos definidos estão descritos na seção 5, mais especificamente na Tabela 2). O aplicativo teria duas formas distintas de exibição: uma ativa, onde o usuário navega pelo controle remoto, e outra passiva, onde a transição de uma tela para outra é automática e ocorre em *loop*, para

exibição em lugares públicos. No modo ativo, o usuário poderia ainda salvar suas notícias preferidas, mais ou menos como faz ao guardar os favoritos do seu navegador.

O projeto objetiva ainda explorar as possibilidades e relatar os desafios para o desenvolvimento de um sistema para telas grandes, com especial atenção para a construção de sua interface gráfica.

Não sendo o objetivo criar um *app* para ser publicado – até porque entendemos que a adoção de um aplicativo com este fim pressupõe a criação de um novo hábito de consumo, que seria o uso do televisor como um dispositivo de leitura, o que é bastante difícil – o sistema aqui apresentado foi construído como um *framework* de código-fonte aberto, denominado *Reading the News on Android TV*.

A proposta do *framework* contempla oito requisitos funcionais e os quatro requisitos não funcionais, detalhados no Apêndice B. O projeto inclui ainda um tutorial⁵, com as instruções para quem deseja criar e publicar o seu próprio *app*.

O desenvolvimento do protótipo foi iniciado em 2022, usando o método de desenvolvimento incremental, onde o objetivo é “o software evoluir através de várias versões, até alcançar o sistema necessário” (SOMMERVILLE, 2019). Nesta forma de desenvolvimento, cada nova versão introduz uma ou mais funcionalidades ao sistema.

Definiu-se então um cronograma com quatro entregas, apresentado na Tabela 2.

Tabela 2. Fases do desenvolvimento da aplicação

Versão	Recurso	Data
0.1	O protótipo permite login anônimo (RF001), lê notícias de fonte externa (RF002), permite a navegação entre notícias (RF003), oferece links (RF005), acessa notícias favoritas (RF006) e permite ao usuário sair da aplicação (RF008)	7/2022
0.2	O protótipo acessa notícias de um arquivo JSON externo (RNF003) e grava notícias favoritas localmente (RNF004).	5/2023
0.5	O protótipo permite navegação em modo quiosque (RF0007), teve a interface da tela de notícia customizada (RNF001) e exibe múltiplos <i>layouts</i> (RNF002).	6/2023
1.0	O sistema foi testado em uma <i>smart TV</i> para adequação da interface e foi documentado para adoção por terceiros.	7/2023

5. Especificações técnicas, requisitos e interface gráfica do sistema

O sistema proposto usa a Internet para baixar um arquivo com resumos de notícias e a URL de imagens para a exibição da tela da TV. A conexão com a rede também é usada para autenticação do usuário. Uma vez exibidas na TV, as notícias podem ser salvas localmente. Estas conexões estão apresentadas graficamente na Figura 2.

⁵ Disponível em https://github.com/psantunes/Reading_the_News_on_Android_TV#readme

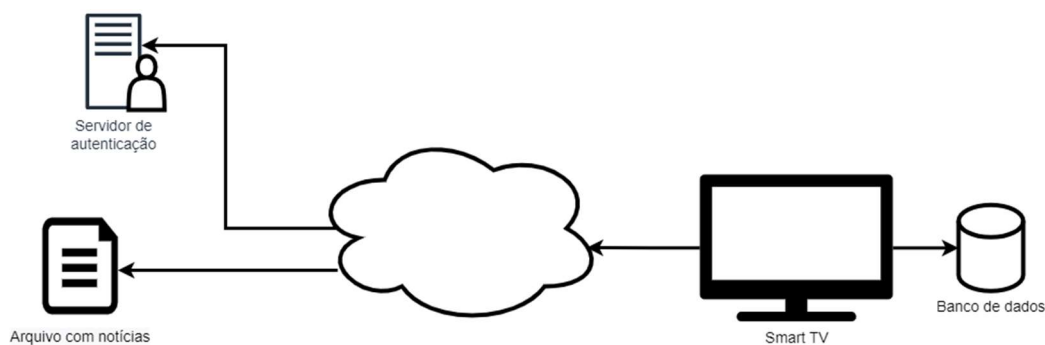


Figura 2. Diagrama da arquitetura do sistema

Para o desenvolvimento da aplicação foi escolhida a plataforma Android TV, em razão da familiaridade do autor com a linguagem Java e com a IDE do Android Studio e pela facilidade de testar as telas de demonstração do *framework* com um *set-top box* de baixo custo. Como observado na seção 2, o Android TV tem boa penetração de mercado (ver Figura 1) e é uma plataforma que os executivos que atuam no setor acreditam que ganhará mercado nos próximos anos (GOETZENBERGER; MOONEY; RIEFLER, 2022, p. 55).

Referente a autenticação do usuário, se decidiu por implantar um sistema de *login* anônimo. No primeiro acesso um identificador único é atribuído ao usuário, que fica armazenado num servidor externo e é validado a cada acesso – desta forma, os dados de navegação de usuários podem ser coletados para análise e melhorias na aplicação. A escolha pelo *login* anônimo se deu em razão da grande dificuldade de autenticar um usuário através da tela da TV – hoje a maioria dos sistemas de *streaming* oferece algum tipo de autenticação usando um segundo dispositivo justamente em razão da dificuldade para se digitar um *login* e senha usando um teclado virtual com o controle remoto. Para este sistema, optou-se pelo registro do usuário com o Firebase Authentication, solução do Google fácil de integrar com múltiplas plataformas (FIREBASE, 2023).

A principal definição técnica foi a escolha do formato de entrada dos textos das notícias na aplicação. O primeiro protótipo da aplicação consumia dados de um banco de dados NoSQL, o que tornava custosa a inserção de dados – basicamente exigindo a criação de um sistema de gerenciamento de conteúdo (*Content Management System*) apenas para alimentar a aplicação. Optou-se então pela leitura de um documento disponibilizado na Internet, a ser configurado no *setup* do aplicativo, o que facilitaria a integração com sistemas já existentes através de uma API.

A primeira alternativa estudada foi a adoção do formato RSS, (*Really Simple Syndication*), um padrão de arquivo XML⁶ criado especialmente para a distribuição de notícias. Apesar de amplamente difundido (por exemplo, sistemas de *weblogs* como o Wordpress já geram automaticamente esta saída para seus usuários), observou-se que o sistema não prevê a existência de alguns campos que a aplicação necessitava (como o crédito da fotografia, uma informação importante no contexto de direitos autorais, e um campo que pudesse ser usado para definir o formato da imagem, permitindo a

⁶ XML é a sigla de Extensible Markup Language, linguagem de marcação de documentos especificada pela World Wide Web Consortium. Largamente utilizada no mundo, aqui no Brasil ela é adotada, por exemplo, na nota fiscal eletrônica.

customização do conteúdo que será exibido na tela). Optou-se então pela adoção do formato JSON⁷, com a criação de um *layout* personalizado com todos os atributos necessários para exibir uma notícia na tela da TV. Os campos pré-definidos, que são serializados em uma classe chamada News, foram especificados no Apêndice C e o modelo do arquivo JSON que a aplicação lê pode ser visualizado no Apêndice D.

Por fim, optou-se em gravar os dados dos favoritos do usuário localmente usando uma base de dados SQLite, evitando assim gravar dados em um servidor de banco de dados externo, o que tornaria a aplicação mais complexa para implantação.

As principais tecnologias usadas para o desenvolvimento do *framework* foram: Java 8, Android Studio (Default JDK 11.0.2), bibliotecas Android (7.1.2, API 25), Leanback (1.0.0), Jackson Core, Firebase Authentication e SQLite Database.

6. Interface gráfica da aplicação

Durante a fase de pesquisa foi possível compreender que o televisor é um dispositivo muito diferente das outras telas às quais estamos acostumados a usar: afinal, o consumo de mídia em um aparelho de TV é geralmente passivo e o seu uso é compartilhado. “A *smart TV* não é pessoal; é um centro de informação e entretenimento para todas as faixas etárias, incluindo idosos, crianças etc. Esses espectadores não técnicos podem enfrentar o problema de conteúdo e sobrecarga cognitiva” (ALAM; KHUSRO; NAEEM, 2017, p. 35, tradução nossa)⁸. Ao programar para uma TV, precisamos levar em conta estas características e dedicar especial atenção à usabilidade da aplicação.

Para produzir o app foram consultados os guias com orientações para desenvolvedores do Google (GOOGLE DEVELOPERS, 2023a) e da Samsung (SAMSUNG DEVELOPERS, 2023) e artigos com dicas de design gráfico. A partir destas leituras, observou-se as seguintes características na construção do *framework*:

- **Distância focal:** é preciso considerar que o usuário geralmente acessa o aparelho de TV a uma distância média de 3 metros, geralmente encostado e relaxado em um sofá ou poltrona. Portanto, as telas da aplicação devem ser simples, com poucos elementos e de fácil visualização.
- **Navegação:** é preciso considerar que a maioria dos controles remotos possuem poucos botões para navegação (os direcionais, um botão “ok” e um botão “voltar”). A cada clique do usuário nestas teclas, o sistema precisa dar uma resposta rápida e distinta. Os princípios de design da Samsung orientam cuidados com o foco (destacando que determinado elemento na tela foi selecionado) e que os movimentos de navegação devem ser previsíveis – o que justifica a preferência pela organização do conteúdo em uma matriz, com linhas e colunas (SAMSUNG DEVELOPERS, 2023).
- **Cores:** aparelhos de TV podem ter diferentes padrões de cores (sRGB ou DCI-P3), padrões de tela (LCS, LED, OLED, QLED) e os usuários ainda podem optar

⁷ JSON é a sigla de JavaScript Object Notation, formato de troca de dados entre sistemas que tem entre uma de suas principais características seu formato compacto e a facilidade de compreensão por parte dos programadores. O padrão é amplamente usado para troca de informações entre *web services*.

⁸ “*Smart TV is not personal; it is an infotainment for all age groups including senior citizen, kids, etc. These non-technical viewers may face the problem of content and cognitive overload*”.

por diferentes modos de exibição (padrão, *vivid*, *movie etc.*) e todos estes elementos alteram as cores das imagens na tela. Para contornar estas diferenças, a orientação é sempre buscar o contraste entre o texto e a cor de fundo. Alguns designers são ainda mais específicos em suas dicas: Brokhina e Gancitano (2021, tradução nossa) orientam que “as cores principais da interface do usuário, como o plano de fundo, por exemplo, geralmente devem ser calmas e escuras”⁹ e Pacheco (2017, tradução nossa) afirma que “cores frias (azul, roxo, cinza) funcionam melhor do que cores mais quentes (vermelho, laranja)”¹⁰.

- **Limites da tela:** como herança dos velhos televisores CRT¹¹ que não seguiam um padrão universal de resolução de tela, ainda hoje sentimos os efeitos do chamado *overscan*. Televisores podem ter resoluções distintas e compensam isto centralizando a imagem, cortando o que não cabe na tela nas laterais. Por este motivo, é recomendado aos desenvolvedores Android posicionarem os elementos a uma distância de 5% das margens. Outros sistemas recomendam margens ainda maiores. Em termos de formato, o padrão atual são telas com proporção 16 por 9. O recomendado é desenhar telas com 960 por 540 pixels e testar se os elementos são redimensionados proporcionalmente em monitores maiores (GOOGLE DEVELOPERS, 2023a).

Ao iniciar um projeto de aplicativo para TV usando o Android Studio, parte destas regras já foram consideradas. Isto porque o Android disponibiliza a biblioteca Leanback, que já possui uma série de layouts de telas prontas. Lançada em 2019 (GOOGLE DEVELOPERS, 2023b), a Leanback oferece uma estrutura de classes e a interface gráfica para montar um catálogo de vídeos – semelhante ao que vemos em sistemas de *streaming*.

Para o desenvolvimento da aplicação buscou-se fazer o máximo de reuso possível destes modelos: o menu de abertura do software, por exemplo, reaproveita a estrutura `GuidedStepSupportFragment`, criada para ser usada no passo-a-passo da customização de uma aplicação de vídeo. Já a tela que exibe as notícias favoritas do usuário é uma simplificação do `BrowseSupportFragment`, construída pelo Leanback para ser um catálogo de vídeos ou fotos, formada por uma série de linhas e uma série de colunas que exibem cards.

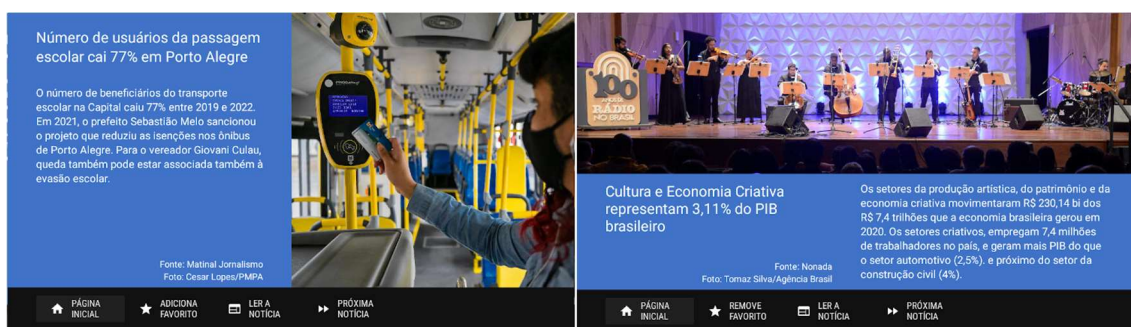


Figura 3. Layouts de telas desenvolvidas para o framework

⁹ “The main colours of the UI, such as the background, for example, should usually be calm and dark”.

¹⁰ “Cool colors (blue, purple, gray) work better than warmer colors (red, orange)”.

¹¹ CRT é a sigla de *Cathodic Ray Tube*, ou Tubo de raios catódicos. São os famosos monitores de tubo, tecnologia que caiu em desuso com o advento das telas de LCD.

Já a principal tela da aplicação, aquela que exibe as notícias, precisou ser desenhada do zero. Para o *framework* foram construídos dois *templates* distintos, um para a exibição de uma imagem vertical e outro para imagem horizontal (ver Figura 3). Futuras versões poderiam apresentar outros formatos, aumentando as possibilidades gráficas.

7. Considerações finais

A versão 1.0 do *framework Reading the News on Android TV* foi finalizada em julho de 2023 e ao longo do processo de desenvolvimento observou-se que houve sinergia entre o referencial teórico que norteou este trabalho e o processo de desenvolvimento de software que se seguiu: é uma experiência muito distinta desenvolver um software para um aparelho de TV em relação a um desktop ou smartphone. A biblioteca Leanback, apesar de complexa, felizmente conduz o usuário neste trabalho, orientando o programador a seguir uma interface que siga uma matriz de linhas e colunas e a colocar o foco nos botões.

O desenho da interface gráfica, no entanto, se mostrou bastante desafiador: poderia se imaginar que seria fácil desenhar interfaces para telas de grandes dimensões, mas é justamente o contrário. É preciso ser minimalista ao dispor os elementos na tela – um dos requisitos funcionais da aplicação (RF003) previa um botão “voltar” no menu de navegação, que foi removido para reduzir o número de elementos da tela. Ao portar a aplicação para uma TV (uma vez que, durante a fase de desenvolvimento, as telas de demonstração do sistema foram testadas no emulador do Android Studio) percebeu-se que vários outros ajustes poderiam ser feitos, melhorando a experiência do usuário – reduziu-se os tamanhos de alguns textos, trocou-se algumas fontes de *bold* para regular e foram feitos ajustes nas margens. Estas mudanças provocaram uma série de melhorias no sistema entre a versão 0.5 e a versão 1.0.

O *Reading the News on Android TV* é um sistema relativamente simples: possui uma tabela de banco de dados, um único objeto a ser serializado, se conecta com dois serviços externos e possui quatro telas. Mas sua construção demandou centenas de horas de trabalho e envolveu um conjunto diverso de atividades, que incluíram reuso de software, programação, design gráfico, versionamento e documentação.

Durante o desenvolvimento do aplicativo optou-se por usar uma API mais antiga do Android, a 25 (Nougat, lançada em 2016), já que o *set-top box* disponível para testes usava a versão 8 do Android. O ponto positivo disto é que o *framework* deve funcionar na grande maioria de dispositivos Android disponíveis no mercado. Por outro lado, isto provocou uma série de dificuldades durante a programação, já que algumas bibliotecas Java não estão disponíveis para APIs mais antigas. É possível que futuras evoluções do sistema demandem a atualização da API, reduzindo o seu alcance, mas ampliando os recursos e melhorando a performance da aplicação.

Como o sistema e suas telas de demonstração não passaram por testes com usuários, não é possível afirmar neste momento se empresas de mídia virão a se interessar pelo projeto e ainda se conseguiriam criar os processos para implementação e viabilização comercial de um *app* de notícias. O próximo passo é divulgar o *framework* e, a partir do *feedback* recebido, avaliar a possibilidade de adicionar novos recursos ou ainda usar o conhecimento aqui adquirido para desenvolver outros aplicativos para *smart TVs*.

Por fim, é importante destacar que um dos objetivos deste projeto também era contribuir com a comunidade de software livre, da qual todos devemos muito. Por este

motivo, o *Reading the News on Android TV* está disponível sob uma licença MIT¹² em um repositório aberto¹³, estando o seu código disponível para quem queira estudá-lo, modificá-lo e mesmo explorá-lo comercialmente.

8. Referências

ALAM, Iftikhar; KHUSRO, Shah; NAEEM, Muhammad. A review of smart TV: Past, present, and future. *In: 2017 International Conference on Open Source Systems and Technologies (ICOSST)*. Lahore: Institute of Electrical and Electronics Engineers Inc., 2017. p. 35–41. Disponível em: <https://ieeexplore.ieee.org/document/8279002>. Acesso em: 1 jul. 2023.

AMAZON.COM INC. Amazon AppStore. **Getting Started Developing Apps and Games for Fire TV**. [S. l.], 2023. Disponível em <https://developer.amazon.com/docs/fire-tv/getting-started-developing-apps-and-games.html>. Acesso em 1 jul. 2023.

AMAZON Fire TV. In: Wikipedia, the free encyclopedia. Flórida: Wikimedia Foundation, 2023. Disponível em: https://en.wikipedia.org/wiki/Amazon_Fire_TV. Acesso em: 1 jul. 2023.

BROKHINA, Yuliia; GANCITANO, Stefania. **10 Tips for UI/UX Design on Smart TV**. Norigin Media, Oslo, 29 jun. 2021. Disponível em: <https://norigintech.medium.com/10-tips-for-ui-ux-design-on-smart-tv-e372eaf98091>. Acesso em: 26 jun. 2023.

BUTLER, David. **History of Roku: Timeline and Facts**. 13 mar. 2020. The Street. Disponível em: <https://www.thestreet.com/technology/history-of-roku>. Acesso em: 5 jul. 2023.

COLLINS, Terry. **Google has sold 55 million Chromecast devices**. CNET, [S. l.], 4 out. 2017. Disponível em <https://www.cnet.com/tech/computing/google-has-sold-55-million-chromecast-and-chromecast-built-in-devices/>. Acesso em 5 jul. 2023.

FIREBASE. **Firestore Authentication, login simples e em várias plataformas**. [S. l.], 2023. Disponível em: <https://firebase.google.com/products/auth?hl=pt-br>. Acesso em 26 jun. 2023.

GATTO, Jean Marie; BERTRAND, Dominique. **Television system in a digital or analog network**. Depositante: Fast France Advanced Systems Technologie Sarl. FR2726670A1. Depósito: 9 nov. 2009. Disponível em: <https://worldwide.espacenet.com/patent/search/family/009468669/publication/FR2726670A1?q=pn%3DFR2726670A1>. Acesso em: 5 jul. 2023.

GOETZENBERGER, Sebastian; MOONEY, Grant; RIEFLER, Bernd. Partnering with Smart TV Platforms: The Content Providers' Dilemma. **JMM International Journal on Media Management**, [s. l.], v. 24, n. 1, p. 45–64, 2022. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/14241277.2022.2055035>. Acesso em: 25 set. 2022.

¹² A licença MIT foi criada pelo Instituto de Tecnologia de Massachusetts (MIT) e é conhecida pelo seu texto curto e seu caráter permissivo – a única condição para adoção do software é dar crédito ao seu autor.

¹³ Disponível em https://github.com/psantunes/Reading_the_News_on_Android_TV

GOOGLE Cast. In: Wikipedia, the free encyclopedia. Flórida: Wikimedia Foundation, 2023. Disponível em: https://en.wikipedia.org/wiki/Google_Cast. Acesso em: 5 jul. 2023.

GOOGLE DEVELOPERS. **Design for TV**. [S. 1.], 2023. Disponível em <https://developer.android.com/design/ui/tv>. Acesso em 25 jun. 2023.

GOOGLE DEVELOPERS. **Leanback**. [S. 1.], 2023c. Disponível em <https://developer.android.com/jetpack/androidx/releases/leanback?hl=pt-br>. Acesso em 23 abr. 2023.

GOOGLE NEWS FROM GOOGLE. **Industry Leaders Announce Open Platform to Bring Web to TV**. [S. 1.], 20 maio 2010. Disponível em: <http://googlepress.blogspot.com/2010/05/industry-leaders-announce-open-platform.html>. Acesso em: 5 jul. 2022.

HP vende webOS para a LG. **Baguete**, Porto Alegre, 26 fev. 2013. Disponível em: <https://www.baguete.com.br/noticias/26/02/2013/hp-vende-webos-para-a-lg>. Acesso em: 5 jul. 2023.

IBGE. **Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal 2021**. Coordenação de Pesquisas por Amostra de Domicílios (Pnad Contínua). Instituto Brasileiro de Geografia e Estatística (IBGE). Rio de Janeiro. Disponível em: <https://biblioteca.ibge.gov.br/biblioteca-catalogo?view=detalhes&id=2101963>. Acesso em: 5 jul. 2023.

JAYGARL, HoJun; LUO, Cheng; KIM, YoonSoo; CHOI, Eunyoung, BRADWICK, Kevin; LANSDELL, Jon. **Professional Tizen Application Development**. Birmingham: Wrox, 2014.

KLEINA, Nilton. **LG libera sistema webOS de Smart TVs para outras fabricantes**. TecMundo, [S. 1.], 24 fev. 2021. Disponível em: <https://www.tecmundo.com.br/software/211568-lg-libera-sistema-webos-smart-tvs-outras-fabricantes.htm>. Acesso em: 5 jul. 2023.

LEAL, Sayonara. Controvérsias na padronização do *middleware* de interatividade da TV digital brasileira: democracia técnica à prova do dispositivo de normatização da inovação Ginga. **Sociedade e Estado**, [s. l.], v. 34, n. 1, p. 49–83, 2019. Disponível em: <http://www.scielo.br/j/se/a/8CgVZ3CGVCsdJcQsmrhV8pL/?lang=pt>. Acesso em: 25 set. 2022.

LEEF, Marc; NAGREE, Hussein. Chime: Anonymous Channel Based Messaging for Smart TVs. [s. l.], 2016. Disponível em: <https://github.com/mleef/Chime>. Acesso em: 21 nov. 2022.

LG ELECTRONICS. WebOS TV Developer. **Comparison with NetCast**. [S. 1.], 2023a. Disponível em: <https://webostv.developer.lge.com/develop/specifications/comparison-netcast>. Acesso em: 5 jul. 2023.

LG ELECTRONICS. WebOS Open Source Edition. **Introduction**. [S. 1.], 2023b. Disponível em: <https://www.webosose.org/about/overview/introduction/>. Acesso em: 5 jul. 2023.

MOZILLA. **First Panasonic Smart TVs powered by Firefox OS Debut Worldwide**. [S.1.], 15 maio 2015. Disponível em: <https://blog.mozilla.org/en/mozilla/first-panasonic-smart-tvs-powered-by-firefox-os-debut-worldwide/>. Acesso em 5 jul. 2023.

PACHECO, Andrea. **TV Guidelines: A quick kick-off on designing for Television Experiences**. UX Collective, Estados Unidos, 24 fev. 2021. Disponível em: <https://uxdesign.cc/guidelines-designing-for-television-experience-524f19ab6357>. Acesso em: 26 jun. 2023.

REID, Rory. HP MediaSmart TV: Clever show-it-all telly. CNET, [S. 1.], 11 jan. 2007. Disponível em: <https://www.cnet.com/tech/home-entertainment/hp-mediasmart-tv-clever-show-it-all-telly/>. Acesso em: 5 jul. 2023.

ROKU INC. Roku. **Roku TV**. [S. 1.], 2023a. Disponível em: <https://www.roku.com/pt-br/products/roku-tv>. Acesso em: 5 jul. 2023.

ROKU INC. Roku Developers. **Development environment overview**. [S. 1.], 2023b. Disponível em: <https://developer.roku.com/pt-br/docs/developer-program/getting-started/architecture/dev-environment.md>. Acesso em: 5 jul. 2023.

SAMSUNG NEWSROOM. **Infographic - History of Samsung Smart TV**. [S. 1.], 21 abr. 2015. Disponível em: <https://news.samsung.com/global/infographic-history-of-samsung-smart-tv>. Acesso em: 5 jul. 2023.

SAMSUNG DEVELOPERS. **Design Principles**. [S. 1.], 2023. Disponível em: <https://developer.samsung.com/smarttv/design/design-principles.html>. Acesso em: 25 jun. 2023.

STATISTA. **Smart TV streaming device market share worldwide as of 2020, by platform**. Disponível em: <https://www.statista.com/statistics/1171132/global-connected-tv-devices-streaming-market-share-by-platform/>. Acesso em: 5 jul. 2023.

SOMMERVILLE, Ian. **Engenharia de software**. 10^a ed. São Paulo: Editora Pearson, 2019.

TIZEN PROJECT. **Tizen Developers**. [S. 1.], 2012. Disponível em <https://developer.tizen.org>. Acesso em 5 jul. 2023.

TREBILCOX-RUIZ, Paul. **Android TV Apps Development: Building for Media and Games**. Apress, 2015.

TVOS. In: Wikipedia, the free encyclopedia. Flórida: Wikimedia Foundation, 2023. Disponível em: <https://en.wikipedia.org/wiki/TvOS>. Acesso em: 5 jul. 2023.

APÊNDICE A

Glossário

Ginga – É o middleware do Sistema Brasileiro da Televisão Digital e Interativa (SBTDi) que permite o desenvolvimento de aplicações interativas para TV digital. A construção de uma tecnologia nacional demandou grande debate da sociedade civil e procurou conciliar interesses mercadológicos e de inclusão digital. O nome do sistema faz referência ao movimento da capoeira.

Media center – Dispositivo usado para acessar conteúdo multimídia, como vídeo, áudio e fotos. Computadores, televisores, consoles de videogames, tocadores de *blu-ray* ou *set-top boxes* podem ser usados como *media center*.

Plataformas de streaming – Serviços que oferecem acesso a conteúdo multimídia via Internet, geralmente mediante assinatura ou através da veiculação de publicidade. A Netflix, a Globoplay e o Youtube, entre outras, são plataformas de *streaming*.

Smart TV - Evolução da tecnologia da televisão, derivada dos avanços provocados pela digitalização e aumento da conectividade, que permite ao aparelho acessar a internet e/ou servidores de mídia e executar aplicativos, games e conteúdo multimídia.

Streaming – Tecnologia de transmissão de dados digital em que as informações são transmitidas e consumidas em fluxo contínuo. Este formato é usado para distribuir conteúdo em áudio e vídeo na internet.

Streaming devices – Dispositivos que conectam televisores à internet, permitindo o acesso a plataformas de *streaming*. Costumam ser uma alternativa barata e portátil para conectar modelos de TV mais antigas à internet. São *streaming devices* o Chromecast (Google), o Fire Stick (Amazon), o Roku Express (Roku) e o Mi TV Stick (Xiaomi).

Set-top box – Dispositivo que se conecta ao televisor permitindo que ele passe a sintonizar conteúdos de fontes externas. Os aparelhos receptores de TV por assinatura são *set-top boxes*, bem como os decodificadores de TV digital. O termo designa também pequenos dispositivos que se conectam à internet, usados para acessar plataformas de *streaming*, canais de IPTV e muitas vezes ligado ao acesso de conteúdo multimídia pirata. *Set-top boxes* se tornaram uma alternativa de baixo custo para transformar televisores normais em *smart TVs*.

Sistema Brasileiro de Televisão Digital (SBTD) – É o padrão técnico para teledifusão digital de televisão brasileiro, adotado no país desde 2007. O padrão nacional é derivado do sistema japonês (ISDB) e tem sido adotado por outros países latino-americanos. À tecnologia japonesa foram adicionadas novas funcionalidades, como uma maior compactação de vídeo e o *middleware* Ginga, que permite o desenvolvimento de aplicações interativas para TV digital.

APÊNDICE B

Requisitos do sistema *Reading the News on Android TV*

Requisitos funcionais

Identificador	RF001		
Nome	Login anônimo		
Versão	0.1	Prioridade	Essencial
Descrição	O usuário pode navegar pela aplicação diretamente, sem fazer cadastro ou login. No entanto, a data de seu primeiro acesso fica registrada na nuvem, para controle e fins estatísticos.		

Identificador	RF002		
Nome	Ler sinopses de notícias		
Versão	0.1	Prioridade	Essencial
Descrição	O usuário pode navegar entre resumos de notícias, que são acessadas através de conexão com a internet. Cada notícia é exibida em uma tela distinta. As notícias possuem os seguintes campos: título, descrição, imagem, fonte, crédito e data.		

Identificador	RF003		
Nome	Navegar entre notícias		
Versão	0.1	Prioridade	Essencial
Descrição	O usuário pode avançar ou voltar de uma tela de notícia para outra, em uma leitura sequencial, através de um menu.		

Identificador	RF004		
Nome	Gravar favoritos/Remover favoritos		
Versão	0.1	Prioridade	Importante
Descrição	O usuário pode salvar as notícias de sua preferência em uma lista de favoritos. Assim, como ele pode gravar uma notícia, ele pode remover ela a qualquer momento.		

Identificador	RF005		
Nome	Ler notícia na íntegra		
Versão	0.1	Prioridade	Pouco importante
Descrição	O usuário pode ler o texto integral da notícia, caso a URL dela tenha sido cadastrada no sistema e se o seu dispositivo tiver um navegador instalado.		

Identificador	RF006		
Nome	Ver favoritos		
Versão	0.1	Prioridade	Importante
Descrição	O usuário pode acessar uma lista com todas as notícias que salvou nos favoritos e reler o texto, independente da data em que foram cadastradas.		

Identificador	RF007		
Nome	Modo quiosque		
Versão	1	Prioridade	Essencial
Descrição	O usuário pode optar por exibir as notícias de forma que elas se sucedem automaticamente, sem o uso do controle remoto, rodando em loop. Desta forma a aplicação pode ser usada para exibição em telões, em locais públicos, de forma passiva.		

Identificador	RF008		
Nome	Sair do sistema		
Versão	0.1	Prioridade	Essencial
Descrição	Permitir que o usuário encerre a aplicação, sem encerrar seu login.		

Requisitos não funcionais

Identificador	RNF001		
Nome	Telas atraentes que obedçam aos princípios do design para TV		
Versão	0.5	Prioridade	Essencial
Descrição	As telas que exibem as sinopses das notícias devem ter um <i>layout</i> agradável e de fácil leitura e o sistema deve obedecer aos princípios de design sugeridos na especificação do Android TV.		

Identificador	RNF002		
Nome	Baixar uma lista de notícias de um arquivo JSON		
Versão	0.2	Prioridade	Importante
Descrição	O sistema deve acessar as notícias de uma fonte externa na internet.		

Identificador	RNF003		
Nome	Gravar as notícias favoritas do usuário localmente		
Versão	0.2	Prioridade	Importante
Descrição	O sistema deve gravar as noticias favoritas no próprio dispositivo do usuário, dispensando o armazenamento de dados num servidor externo.		

Identificador	RNF004		
Nome	Documentação do sistema para que possa ser adotado por desenvolvedores que desejem criar seus aplicativos		
Versão	1	Prioridade	Muito importante
Descrição	Disponibilizar junto ao repositório um passo a passo de como instalar e customizar a aplicação.		

APÊNDICE C

Campos definidos para a classe News, que precisam constar no arquivo JSON que irá alimentar a aplicação, sua descrição e seus tipos

Atributo	Definição	Tipo
id	Identificador único da notícia	Inteiro
title	Título da notícia	String
article	Resumo da notícia	String
bgImageUrl	URL da imagem em tamanho grande que ilustra a notícia O sistema foi testado com imagens horizontais com 1280x360 pixels e imagens horizontais de 640x720 pixels.	String
cardImageUrl	URL da imagem para o card que ilustra a página que exibe as notícias favoritas. A tela de favoritos foi configurada para imagens com proporção 16 por 9. As imagens usadas nos testes tinham 400x225 pixels.	String
source	Fonte da notícia	String
link	Link para o texto integral da notícia	String
photoCredit	Nome do autor da foto	String
publishedAt	Data de publicação do texto	Data
orientation	Orientação do post. É esperado o valor “horizontal” se a imagem for horizontal e “vertical” se a imagem for vertical. Se novos <i>layouts</i> forem criados no futuro, este campo será usado para identificá-los.	String

APÊNDICE D

Modelo de arquivo JSON que o app reconhece

```
[
  {
    "id": "1",
    "title": "Número de usuários da passagem escolar cai 77% em Porto Alegre",
    "article": "O número de beneficiários do transporte escolar na Capital caiu 77% entre 2019 e 2022. Em 2021, o prefeito Sebastião Melo sancionou o projeto que reduziu as isenções nos ônibus de Porto Alegre. Para o vereador Giovani Culau, queda também pode estar associada também à evasão escolar.",
    "source": "Matinal Jornalismo",
    "link": "https://www.matinaljornalismo.com.br/matinal/newsletter/apos-restricoes-numero-de-usuarios-da-passagem-escolar-cai-77-em-porto-alegre/",
    "publishedAt": "2023-05-05",
    "bgImageUrl":
    "https://raw.githubusercontent.com/psantunes/Reading_the_News_on_Android_TV/main/docs/t
    ri_g.jpg",
    "cardImageUrl":
    "https://raw.githubusercontent.com/psantunes/Reading_the_News_on_Android_TV/main/docs/t
    ri_p.jpg",
    "photoCredit": "Cesar Lopes/PMPA",
    "orientation": "vertical"
  },
  {
    "id": "2",
    "title": "Cultura e Economia Criativa representam 3,11% do PIB brasileiro",
    "article": "Os setores da produção artística, do patrimônio e da economia criativa movimentaram R$ 230,14 bi dos R$ 7,4 trilhões que a economia brasileira gerou em 2020. Os setores criativos, empregam 7,4 milhões de trabalhadores no país, e geram mais PIB do que o setor automotivo (2,5%). e próximo do setor da construção civil (4%).",
    "source": "Nonada",
    "link": "https://www.nonada.com.br/2023/04/pesquisa-mostra-que-cultura-e-economia-criativa-representam-311-do-pib-brasileiro-ultrapassando-o-setor-automotivo/",
    "publishedAt": "2023-04-10",
    "bgImageUrl":
    "https://raw.githubusercontent.com/psantunes/Reading_the_News_on_Android_TV/main/docs/
    musica_g.jpg",
    "cardImageUrl":
    "https://raw.githubusercontent.com/psantunes/Reading_the_News_on_Android_TV/main/docs/
    musica_p.jpg",
    "photoCredit": "Tomaz Silva/Agência Brasil",
    "orientation": "horizontal"
  }
]
```