Leandro Cezar Pavin

Pedro Luiz Weselowski da Cunha

Roger Paulino Donegati

**Zeugma: Aplicativo Móvel para Controle de Entretenimento Online**

São Caetano do Sul/SP

2017

Leandro Cezar Pavin

Pedro Luiz Weselowski

Roger Paulino Donegati

**Zeugma: Aplicativo Mobile para Controle de Entretenimento Online**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Caetano do Sul, sob a orientação do Professor Me. Wilson Vendramel, como requisito parcial para a obtenção do diploma de Graduação no Curso de Análise e Desenvolvimento de Sistemas

São Caetano do Sul/SP

2017

**RESUMO**

PAVIN, Leandro Cezar; WESELOWSKI, Pedro Luiz da Cunha; DONEGATI, Roger Paulino. **Zeugma: Aplicativo Mobile para Controle de Entretenimento Online**, 65 fls. Trabalho de Conclusão de Curso – Faculdade de Tecnologia de São Caetano do Sul – Antônio Russo, São Caetano do Sul, 08 de dezembro de 2017.

O consumo de entretenimento online tem crescido significativamente nos últimos anos, e como em todas as áreas, a demanda de novos aplicativos relacionados também. Este trabalho foi desenvolvido com o intuito da realização de um aplicativo móvel, funcional para diversos usuários que consomem conteúdos de entretenimento online como filmes e séries, e permitir a interação entre as pessoas que partilham ou não dos mesmos gostos. Ao longo do desenvolvimento do trabalho foram realizadas diversas pesquisas sobre os recursos e conceitos mais atuais e que melhor atendiam a proposta apresentada, isto possibilitou o aprendizado e utilização prática destes, o que enriqueceu consideravelmente o cunho acadêmico do projeto. O embasamento de aplicação do projeto sempre teve o seu foco voltado para a arquitetura REST utilizando protocolo JSON, um dos modelos mais atuais quando se tratando de Web services. Após o desenvolvimento do trabalho o aplicativo foi introduzido a um período de testes para avaliar sua usabilidade utilizando as 10 heurísticas de Nielsen, com possíveis usuários, considerou-se que o aplicativo conseguiu atingir os objetivos e também obteve uma boa usabilidade, porém com certas possíveis melhorias que poderão ser implementadas futuramente.

**Palavras chave:** Entretenimento, Aplicativo Móvel, Usabilidade, Avaliação Heurística, REST.

**ABSTRACT**

PAVIN, Leandro Cezar; WESELOWSKI, Pedro Luiz da Cunha; DONEGATI, Roger Paulino. **Zeugma: Aplicativo Mobile para Controle de Entretenimento Online**, 65 fls. Trabalho de Conclusão de Curso – Faculdade de Tecnologia de São Caetano do Sul – Antônio Russo, São Caetano do Sul, 08 de dezembro de 2017.

The consumption of online entertainment has grown significantly in the past few years, and as in all areas, the demand for new related applications as well. This project was developed with the purpose of creating a mobile application, functional for several users that consume online entertainment content like movies and tv shows, and allows interaction between people who shares or not the same tastes. Throughout the development of the project, a huge number of researches were carried out on the most current concepts and resources that best met the proposal presented, this made possible the learning and practical use of them, which greatly enriched the academic character of the project. The application foundation of the project had always focused on the REST architecture using JSON protocol, one of the most current models when it comes to Web services. After the development of the project, the application went through a test period to evaluate its usability using the Nielsen’s 10 heuristics, with potential users. It was considered that the application managed to achieve the objectives and obtained a good usability, but with some possible improvements that could be implemented in the future.

**Key-words:** Entertainment, Mobile Application, Usability, Heuristic Evaluation, REST.

**LISTA DE IMAGENS**

[Figura 1 - Representação SOA Mobile 12](#_1t3h5sf)

[Figura 2 – Representação Web service 14](#_2s8eyo1)

[Figura 3 - Representação REST 15](#_3rdcrjn)

[Figura 4 - Exemplo JSON 18](#_3j2qqm3)

[Figura 5 - Representação Firebase 2](#_49x2ik5)2

[Figura 6 - Diagrama de Classes](about:blank) 23

[Figura 7 - Diagrama de Padrão Arquitetural do Aplicativo](about:blank) 24

[Figura 8 – Logotipo do aplicativo 2](#_ihv636)5

[Figura 9 - ScreenShot Tela de Login 2](#_1hmsyys)6

[Figura 10 - ScreenShot Tela de Cadastro 2](#_41mghml)7

[Figura 11 – ScreenShot Tela Principal](#_2grqrue) 28

[Figura 12 - ScreenShot Tela de Pesquisa](#_vx1227) 30

[Figura 13 - ScreenShot Tela de Séries 3](#_3fwokq0)1

[Figura 14 - ScreenShot Tela de Configurações 3](#_1v1yuxt)2

[Figura 15 - ScreenShot Tela de Configurações 3](#_4f1mdlm)4

[Figura 16 - Continuação da Tela de Configurações 3](#_2u6wntf)5

[Figura 17 - Continuação da Tela de Configurações 3](#_19c6y18)6

[Figura 18 - ScreenShot Tela de Filmes Populares 3](#_3tbugp1)7

[Figura 19 - ScreenShot Tela de Séries Populares 38](#_28h4qwu)

[Figura 20 - ScreenShot da Tela de Informações de Séries](#_nmf14n) 39

[Figura 21 - Continuação da Tela de Informações das Séries 4](#_37m2jsg)0

[Figura 22- Continuação da Tela de Informações das Séries 4](#_1mrcu09)1

[Figura 23 - ScreenShot da Tela de Episódios das Séries 4](#_46r0co2)2

[Figura 24 - ScreenShot da Tela de Informações de Filmes 4](#_2lwamvv)3

[Figura 25 - Continuação da Tela de Informações de Filmes 4](#_111kx3o)4

[Figura 26 - Screenshot da Tela de Alteração de Dados Pessoais 4](#_3l18frh)5

[Figura 27 - Screenshot da Tela de Favoritos 4](#_206ipza)6

[Figura 28 - Screenshot da Tela de Perfil do Usuário 4](#_4k668n3)7

[Figura 29 - Screenshot da Tela de Privacidade 48](#_2zbgiuw)

[Figura 30 - Continuação da Tela de Privacidade 49](#_1egqt2p)

[Figura 31 - Screenshot da Tela de Busca de Amigos 5](#_3ygebqi)0

**SUMÁRIO**

[Introdução 8](#_Toc498690938)

[Objetivos Gerais 10](#_Toc498690940)

[Objetivos Específicos 10](#_Toc498690940)

[Justificativa 10](#_Toc498690941)

[Metodologia 11](#_Toc498690942)

[1. Fundamentação Teórica 12](#_Toc498690943)

[1.1 Padrão Arquitetural SOA 12](#_Toc498690944)

[1.2 Web service 13](#_Toc498690945)

[1.3 REST 14](#_Toc498690946)

[1.3.1 Client-Server (Cliente Servidor): 15](#_Toc498690947)

[1.3.2 Stateless (Sem Estado): 16](#_Toc498690948)

[1.3.3 Cache 16](#_Toc498690949)

[1.3.4 Uniform Interface (Interface Uniforme) 16](#_Toc498690950)

[1.3.5 Layered System (Sistema em Camadas) 17](#_Toc498690951)

[1.3.6 Code-On-Demand (Codigo sob Demanda) – Opcional 17](#_Toc498690952)

[1.4 JSON 18](#_Toc498690953)

[2. Desenvolvimento do aplicativo 20](#_Toc498690954)

[2.1 Estudo de viabilidade 20](#_Toc498690955)

[2.2 Requisitos do aplicativo 20](#_Toc498690956)

[2.3 Tecnologias Utilizadas 21](#_Toc498690958)

[2.3.1 Java 2](#_Toc498690959)2

[2.3.2 API’s utilizadas 22](#_Toc498690960)

[2.3.3 Firebase 2](#_Toc498690961)2

[2.4 Diagrama de Classes 2](#_Toc498690962)3

[2.5 Padrão Arquitetural do Aplicativo 2](#_Toc498690963)4

[3. Avaliação do Aplicativo 2](#_Toc498690964)5

[3.1 Logotipo Zeugma 2](#_Toc498690965)5

[3.2 Telas do Aplicativo Zeugma 2](#_Toc498690966)6

[3.3 Avaliação Heurística de Nielsen 5](#_Toc498690967)1

[3.4 Questionário de Avaliação da Usabilidade do Aplicativo Zeugma 5](#_Toc498690978)3

[3.5 Considerações sobre a avaliação heurística 5](#_Toc498690979)8

[Considerações Finais](#_Toc498690980) 59

[Referências 6](#_Toc498690981)0

[APÊNDICE A – Diagrama de Classes Completo 6](#_Toc498690982)2

[APÊNDICE B – Questionário de Avaliação da Usabilidade do Aplicativo Zeugma 6](#_Toc498690983)3

# 

# Introdução

Cinema é a técnica e a arte de fixar e de reproduzir imagens que originam impressão de movimento. Os filmes são produzidos através da gravação de imagens do mundo com câmeras adequadas, ou utilizando técnicas de animação e efeitos visuais específicos. Conforme Barthes (1963) “Nada é mais essencial a uma sociedade que a classificação de suas linguagens. Mudar essa classificação, deslocar a fala, é fazer uma revolução”.

O cinema tem sido classificado como forma de lazer e de entretenimento, mas também pode ser interpretado como sendo uma ferramenta das mais úteis para a propagação de conceitos, formas de pensar, modos de agir. Artistas e pessoas ligadas à área de produção estética apoiam a tese da produção cinematográfica como sendo a “sétima arte”.

Com a evolução de tecnologias para reprodução “virtual” de filmes e séries as pessoas passaram a ter uma nova maneira de acompanhar a “sétima arte”, o entretenimento online vem se tornando mais e mais popular e aplicativos de auxilio para os consumidores dos mesmos se tornando necessários.

Atualmente, as pessoas têm consumido cada vez mais os meios de comunicação, sejam eles internet, televisão. Tomando por base essa ideia, nossa proposta será oferecer a essas pessoas um aplicativo no qual o usuário terá acesso a uma base de dados com diversos conteúdos, onde haverá informações completas sobre filmes e séries, tal como o número de temporadas e episódios, o elenco, as datas de lançamento, sumários dos episódios etc.

Este projeto está subdividido nos seguintes capítulos:

* Capítulo 1: Introdução, encontra-se especificado os objetivos, as justificativas e a metodologia para o desenvolvimento do projeto.
* Capítulo 2: Fundamentação Teórica, neste capítulo consta todo o embasamento de pesquisas, conceitos e recursos utilizados para a realização do projeto.
* Capítulo 3: Desenvolvimento do Projeto, descrito todos os passos realizados para a produção do projeto, estudos, requisitos, tecnologias utilizadas e diagramas são apresentados neste tópico.
* Capítulo 4: Avaliação do Aplicativo, após o desenvolvimento do projeto foi realizada uma pesquisa visando avaliar a usabilidade do aplicativo baseadas nas heurísticas de Nielsen, os resultados são exibidos.

## Objetivos Gerais

Desenvolvimento de um aplicativo que permite integrar diversos conteúdos em um único local, auxiliando o usuário a acompanhar informações gerais e curiosidades sobre filmes e séries, sem a necessidade de acesso à sites externos para obter tais informações.

## Objetivos Específicos

* Especificação dos requisitos.
* Desenvolver um aplicativo funcional para consumidores de entretenimento online.
* Promover disputas amigáveis entre pessoas, facilitando maneiras de interação entre usuários.
* Implementação de um padrão arquitetural.
* Avaliação heurística.

## Justificativa

No mercado existem aplicativos relacionados a conteúdo online, porém, a maioria deles é muito específica, limitada e muitas vezes não atende aos requisitos dos usuários. Um exemplo é o site IMDB, pois apesar de uma vasta quantidade de conteúdo, o seu foco está, na maior parte, em entretenimento norte americano. Por esse motivo resolvemos desenvolver um aplicativo que seja mais abrangente e destinado inicialmente ao público brasileiro, alcançando filmes e séries. Dessa maneira o usuário terá uma maior diversidade de conteúdo em um único aplicativo. Incluiremos também uma série de funcionalidades exclusivas trazendo uma ótima experiência para o usuário final.

A plataforma Android foi escolhida devido a sua fácil inserção de novos aplicativos no mercado, ter um número maior de usuários no Brasil e possuir dispositivos de custos variados tornando o sistema operacional acessível a uma maior faixa da população.

## Metodologia

Para o desenvolvimento deste trabalho, será utilizada uma metodologia de pesquisa exploratória com apoio de bibliografias especializadas referentes ao tema, sites acadêmicos, livros, artigos, teses, além de pesquisas sobre tecnologias necessárias para o desenvolvimento do aplicativo.

# Fundamentação Teórica

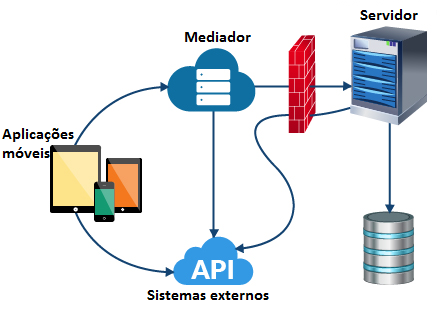
Neste capítulo constam os conceitos utilizados para a realização do trabalho.

## Padrão Arquitetural SOA

Segundo Erl (2008) “SOA é a transformação de recursos de TI em serviços de software descentralizados que podem se comunicar entre si aumentando a flexibilidade dos aplicativos de negócio. ”

A fim de utilizar eficientemente uma SOA, deve ocorrer a interoperabilidade entre diferentes sistemas e linguagens de programação que fornecem a base para a integração entre aplicações em diferentes plataformas, através de um protocolo de comunicação. Um exemplo dessa comunicação depende do conceito de mensagens. Usar mensagens, através de canais de mensagens definidos, diminui a complexidade da aplicação final, permitindo que o desenvolvedor do aplicativo consuma o banco de dados compartilhado. Isto permite que novas funcionalidades sejam desenvolvidas para um formato de negócio de referência comum para cada elemento de dados.

*Figura 1 - Representação SOA Mobile*



Fonte: PROLIFICS, 2017

SOA deve ser visto como um paradigma arquitetural no qual componentes de aplicações são distribuídos, combinados e consumidos. Não podemos então confundir SOA com uma tecnologia, biblioteca de recursos de sistemas ou algo do gênero, e sim uma forma de projetar e organizar a infraestrutura de funcionalidades em um ambiente corporativo.

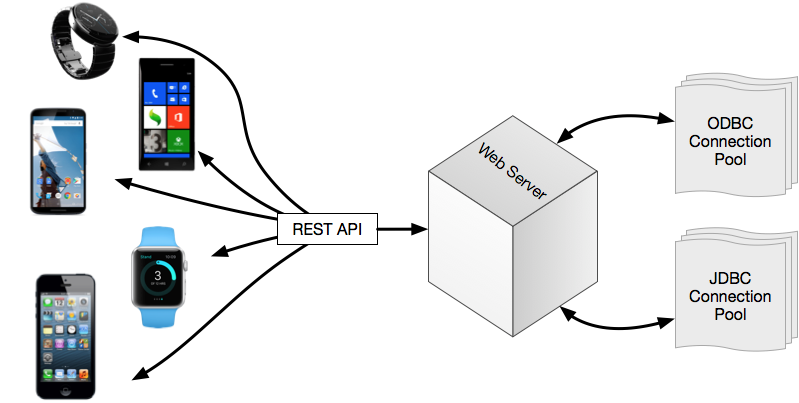
## Web service

Um Web service é utilizado para transferir dados através de protocolos de comunicação para diferentes plataformas, independentemente das linguagens de programação utilizadas nessas plataformas.

Snell (2002) diz que Web services são interfaces acessíveis de rede, para as funcionalidades da aplicação, que utilizam em sua construção tecnologias padrões da Internet.

Essencialmente, o Web service faz com que os recursos da aplicação do software estejam disponíveis sobre a rede de uma forma normalizada. Outras tecnologias fazem a mesma coisa, como por exemplo, os browsers da Internet acessam às páginas Web disponíveis usando por norma as tecnologias da Internet, HTTP e HTML. No entanto, essas tecnologias não são bem-sucedidas na comunicação e integração de aplicações. Existe uma grande motivação sobre a tecnologia Web service pois possibilita que diferentes aplicações comuniquem entre si e utilizem recursos diferentes.

O objetivo dos Web services é a comunicação de aplicações através da Internet. Esta comunicação é realizada com intuito de facilitar a EAI (Enterprise Application Integration) que significa a integração das aplicações de uma empresa, ou seja, interoperabilidade entre a informação que circula numa organização nas diferentes aplicações como, por exemplo, o comércio electrónico com os seus clientes e seus fornecedores. Esta interação constitui o sistema de informação de uma empresa. E para além da interoperabilidade entre as aplicações, a EAI permite definir um workflow entre as aplicações e pode constituir uma alternativa aos ERP (Enterprise Resource Planning).

*Figura 2 – Representação Web service*

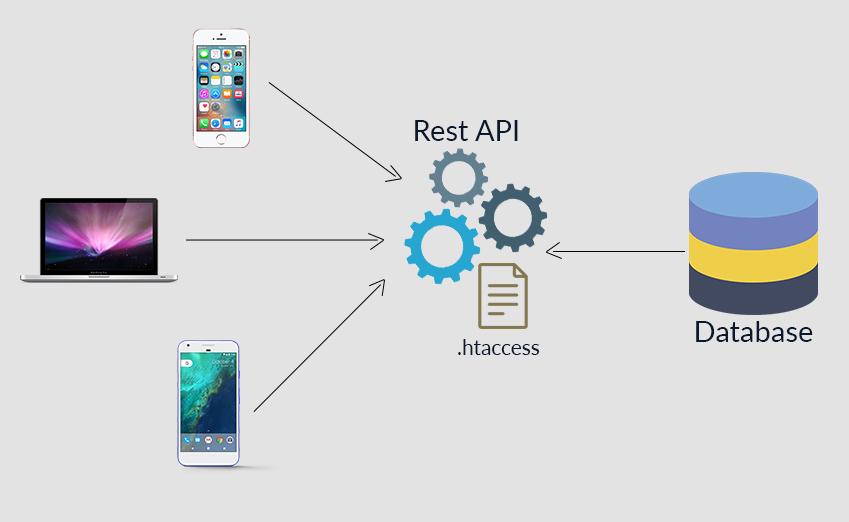
Fonte: SIMBA TECH, 2015

## REST

Segundo Roy Fielding (2000), a arquitetura REST é um conjunto de práticas e ideias que devem ser empregadas em uma aplicação ou software baseado na web. Tendo como objetivo padronizar as aplicações, otimizar os seus serviços, facilitar a identificação de problemas existentes nas mesmas e atrair soluções padronizadas e colaborativas.

A sigla REST significa Representational State Transfer ou traduzida para o português como transferência do estado representacional é um modelo arquitetural atual que foi introduzido e definido no ano de 2000 em uma tese PHD pelo cientista de computação norte-americano Roy T. Fielding que é também um dos principais autores da especificação do famoso protocolo HTTP.

*Figura 3 - Representação REST*



Fonte: CYGNET INFOTECH, 2017

Para alcançar estes objetivos esse modelo utiliza uma série de “regras” denominadas *constraints* (restrições) que sintetizam a arquitetura REST e criam abertura para o termo RESTful, o que costuma causar uma certa confusão para estudantes da área ou profissionais com pouca experiência. Enquanto REST é o modelo arquitetural até aqui descrito, RESTful é uma aplicação ou software que implementa as ideias deste modelo arquitetural e utiliza todas as suas constraints.

Em sequência apresentam-se as *constraints* descritas por Roy Fielding e uma breve explicação sobre cada uma:

### **Client-Server (Cliente Servidor):**

Consiste na ideia de separar o cliente do servidor como é hoje na maioria das aplicações web, uma forma mais fácil de se perceber é pensando nos termos “front-end” e “back-end”, o front-end representa o cliente e é a interface da aplicação com a qual o usuário irá interagir, já o back-end representa o servidor onde será feito todo o processamento dos dados e requisições vindas do front-end. Essa *constraint* acaba levando a vantagem de escalabilidade uma vez que o servidor pode ser melhorado ou alterado sem a necessidade de informar ao usuário ou modificar o front-end.

### **Stateless (Sem Estado):**

Essa *constraint* define que as requisições da aplicação ao servidor devem ser independentes umas das outras, fazendo com que todas as requisições tenham as informações necessárias para serem tratadas pelo servidor, de forma que o servidor não tenha conhecimento das requisições anteriores ou futuras de cada usuário individualmente. Isso acaba trazendo maior segurança para a aplicação e praticidade por se tratar cada requisição separadamente e completamente.

### **Cache**

O cache como em muitas outras áreas da informática se trata de uma pequena porção de memória com dados que são comumente usados por algum sistema, hardware ou software. Nessa porção de memória são colocados esses dados para que não seja necessario processa-los repetidamente durante a execução em questão, dentro da arquitetura REST podemos por exemplo colocar a requisição da página inicial de uma aplicação dentro da cache melhorando assim a performance da aplicação.

### **Uniform Interface (Interface Uniforme)**

Nesta restrição se define que a aplicação tenha uma interface modelada entre o cliente e o servidor e que esta siga padrões importantes, levando em consideração os Recursos, Mensagem Auto Descritivas e Hypermedia.

Este conceito se assemelha bastante ao conceito de Interface de Orientação Objetos, onde temos um objeto que será nossa interface em que descrevemos os padrões nesse objeto e criamos outros objetos que implementam esse objeto, a ideia aqui é que todos os objetos que implementam a interface sigam o mesmo padrão definido na interface.

### **Layered System (Sistema em Camadas)**

Em um sistema em camadas deve ser possível a implementação de elementos intermediário e transparente aos clientes do sistema. Ou seja, deve-se permitir que surjam outras camadas em meios as camadas já existentes do sistema e cada camada deve ter suas interfaces de entrada e saída bem definidas e documentadas. Isso traz novamente um alto grau de escalabilidade ao sistema de forma que o cliente não venha a saber que o sistema está se adaptando e ficando cada vez mais eficiente em atender a demanda necessária e complexa.

### **Code-On-Demand (Codigo sob Demanda) – Opcional**

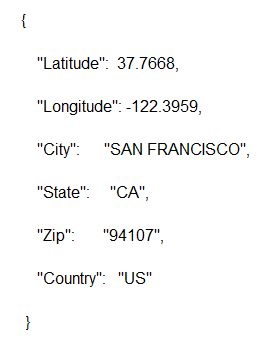
Essa se trata da única constraint considerada opcional dentro de uma arquitetura REST. Ela define que o código ou serviço de sua aplicação seja baixado no dispositivo dinamicamente respeitando a demanda necessária, assim o cliente usa somente o que realmente é necessário e requisitado pelo mesmo.

Esta prática traz maior desempenho a aplicação além de deixa-la reduzida no dispositivo, somente é opcional e não muito utilizada pelo fato de segurança onde o cliente não saberá exatamente o que está sendo baixado o que pode causar certo estranhamento, além de tornar o sistema levemente mais suscetível a explorações ou invasões indesejadas.

## JSON

Segundos Crokford (2006), o JSON (JavasSript Object Notation ou em português Notação de Objeto JavaScript) se trata de um formato de texto leve para troca de dados entre sistemas derivado do JavaScript, este formato além de leve é também simples de ser lido por nós humanos e não apenas interpretado por programas como em outros protocolos. O JSON se utiliza de solicitações assíncronas e tem a sua própria extensão para arquivo que é JSON.

*Figura 4 - Exemplo JSON*



Fonte: CROCKFORD, 2006

Especificado originalmente por Douglas Crockford no início dos anos 2000, foi o próprio Douglas Crockford o responsável por popularizar o JSON como uma alternativa ao XML que era comumente utilizado na época. E desde então o JSON vem sendo cada vez mais utilizado e adotado pelos mais variados sistemas que buscam suas características práticas, que se enquadram muito bem no ambiente de sistemas distribuídos na Web que temos atualmente.

Como especificado por Douglas Crockford o JSON pode representar 4 tipos primitivos de informações: strings, números, booleans e null. Toda informação contida em um JSON é agrupada por pares, estes pares são sempre respectivamente: O nome do item e a informação contida nele. Os objetos são separados uns dos outros por chaves “{}” como se fossem blocos e as informações dentro desses blocos são separadas por virgulas.

Como apontado por Gama (2017), o JSON traz consigo diversas vantagens em relação ao seu antecessor XML (que ainda é usado). Essas vantagens são: compreensão por humanos já citada, a fácil conversão de seus tipos de dados, uma melhora significativa na performance de transporte de dados, o tamanho do arquivo reduzido e como mencionado a pouco o JSON retorna objetos do tipo JSON que podem ser facilmente convertidos para objetos de linguagens orientadas a objetos, simplificando e facilitando muito a codificação, tratamento e processamento dos dados.

Além de todos os benefícios e praticidades já citadas o JSON também é uma tendência crescente entre as maiores empresas de desenvolvimento do mundo em aplicações como: Google, Facebook, Twitter, Yahoo!, etc. Isso pode ser levado como um forte de indicio de que essa tecnologia receberá um bom suporte e permanecerá funcional por pelo menos alguns anos no mercado.

# Desenvolvimento do aplicativo

O desenvolvimento do aplicativo Zeugma teve como base o modelo de prototipação. Com esse modelo, foram gerados protótipos até o projeto estar na versão definitiva do aplicativo.

Nas seções a seguir foram levantadas informações para o desenvolvimento do aplicativo.

## Estudo de viabilidade

Para o desenvolvimento do projeto foi necessário um estudo para visualização das viabilidades impostas. Requisitos foram levantados, recursos e ferramentas avaliados.

## Requisitos do aplicativo

*Quadro 1 – Requisitos Funcionais*

|  |  |
| --- | --- |
| **ID do Requisito** | **Descrição** |
| RF01 | O aplicativo deverá ser capaz de armazenar o cadastro dos usuários. |
| RF02 | O aplicativo deverá disponibilizar listas de conteúdos para os usuários, sendo possível remover e editar as listas conforme desejado. |
| RF03 | O aplicativo irá filtrar os conteúdos melhor avaliados e irá mostrar na tela determinada. |
| RF04 | O aplicativo deverá permitir interação entre os usuários. |

Quadro nosso

*Quadro 2 – Requisitos Não Funcionais*

|  |  |
| --- | --- |
| **ID do Requisito** | **Descrição** |
| RNF01  (Usabilidade) | O aplicativo deverá ter uma interface baseada nas heurísticas de Nielsen, de forma que dispense muitas explicações. |
| RF02  (Interoperabilidade) | O aplicativo deverá disponibilizar login no mesmo com os dados necessários ou disponibilizar login via API externa (Facebook). |
| RNF03  (Segurança) | O aplicativo deve utilizar um servidor externo que proteja suas conexões para armazenar os dados dos usuários. |
| RNF04  (Portabilidade) | O aplicativo deve ser funcionar em todos dispositivos Android acima da versão 4.0 |
| RNF05  (Portabilidade) | O aplicativo deve ser feito permitindo uma fácil portabilidade para com os dados, caso seja necessário a troca de servidor e até mesmo de linguagem para com o banco de dados. |
| RNF06  (Interoperabilidade) | O aplicativo deve possibilitar a implantação futura de mais API’s externas para absorver dados necessários. |
| RNF07  (Segurança) | O aplicativo deve realizar conexões seguras com sistemas externos sem expor dados importantes do usuário. |

Quadro nosso

## Tecnologias Utilizadas

Esta seção irá apresentar conceitos fundamentais relacionados ao trabalho e como eles foram aplicados (plataformas usadas para desenvolvimento, ambiente de desenvolvimento).

### **Java**

O Java foi utilizado nesse projeto por ser uma linguagem de conhecimento dos integrantes do grupo e por, justamente, ser a linguagem de maior utilização no desenvolvimento de aplicativos para a plataforma Android.

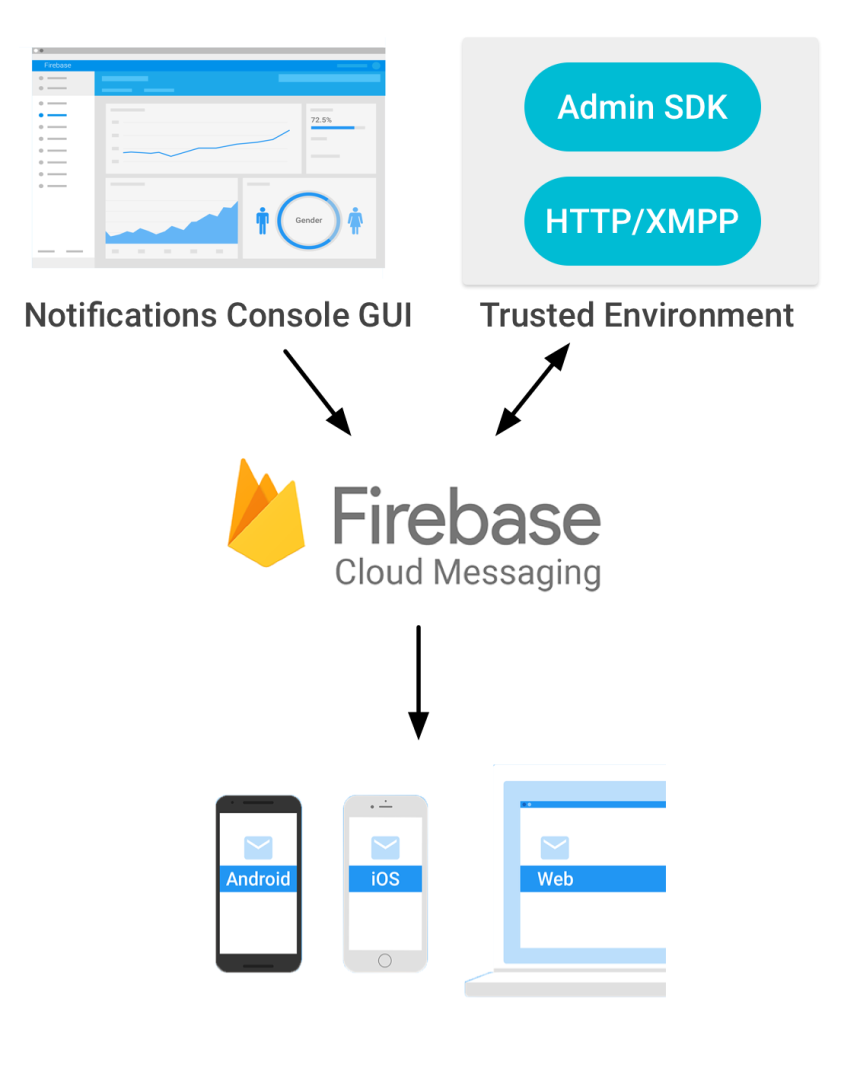
### **API’s utilizadas**

API é uma Interface de Programação de Aplicativos. Para a realização do projeto a API base de desenvolvimento foi o TheMovieDB que fornece toda biblioteca de informações e ferramentas que possibilitam trabalhar dentro deles.

### **Firebase**

O Firebase Cloud Messaging (FCM) é uma ferramenta de mensagens entre plataformas disponibilizada pela Google, que permite o envio confiável de notificações sem custo. Para que as notificações via email para usuários do aplicativo Zeugma ocorram, essa solução é utilizada.

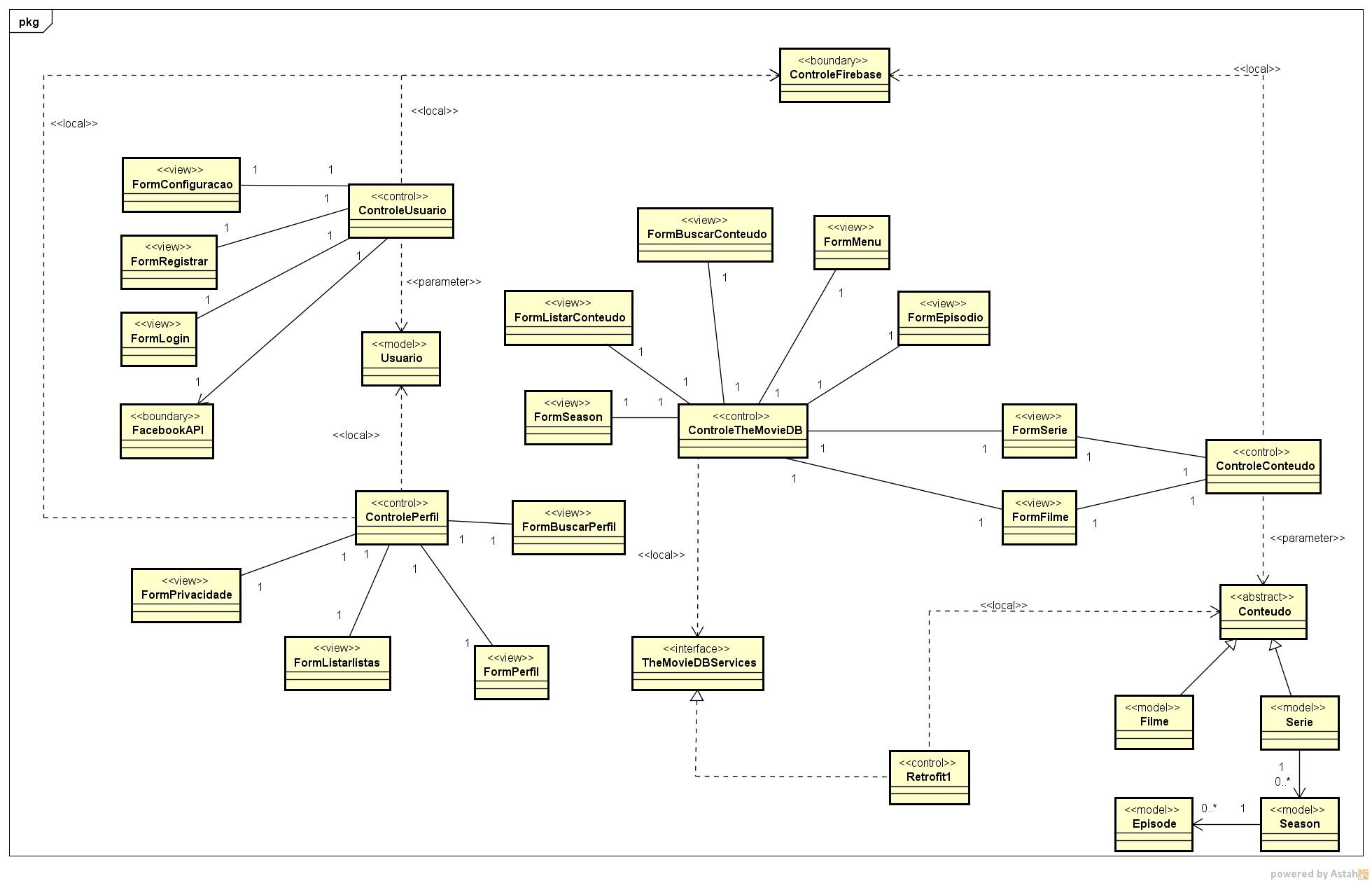
*Figura 5 - Representação Firebase*



Fonte: FIREBASE GOOGLE, 2016

## Diagrama de Classes

Neste tópico é mostrado o diagrama de classes que representa o escopo do aplicativo. O diagrama de classe de projetos encontra-se para melhor visualização e com informações completas no apêndice A.



*Figura 6 - Diagrama de Classes com notação simplificada*

Figura nossa

## Padrão Arquitetural do Aplicativo

O aplicativo utiliza alguns serviços externos e a comunicação ocorre da seguinte maneira: Caso o usuário escolha efetuar o login via a API do Facebook uma solicitação é encaminhada ao Facebook e é retornado se foi possível a realização do login. São utilizadas as API’s do TheMovieDB para recolher as informações necessárias via protocolo JSON. O Retrofit é utilizado como uma espécie de “ponte” entre o aplicativo e a API do TheMovieDB.

*Figura 7 - Diagrama de Padrão Arquitetural do Aplicativo*

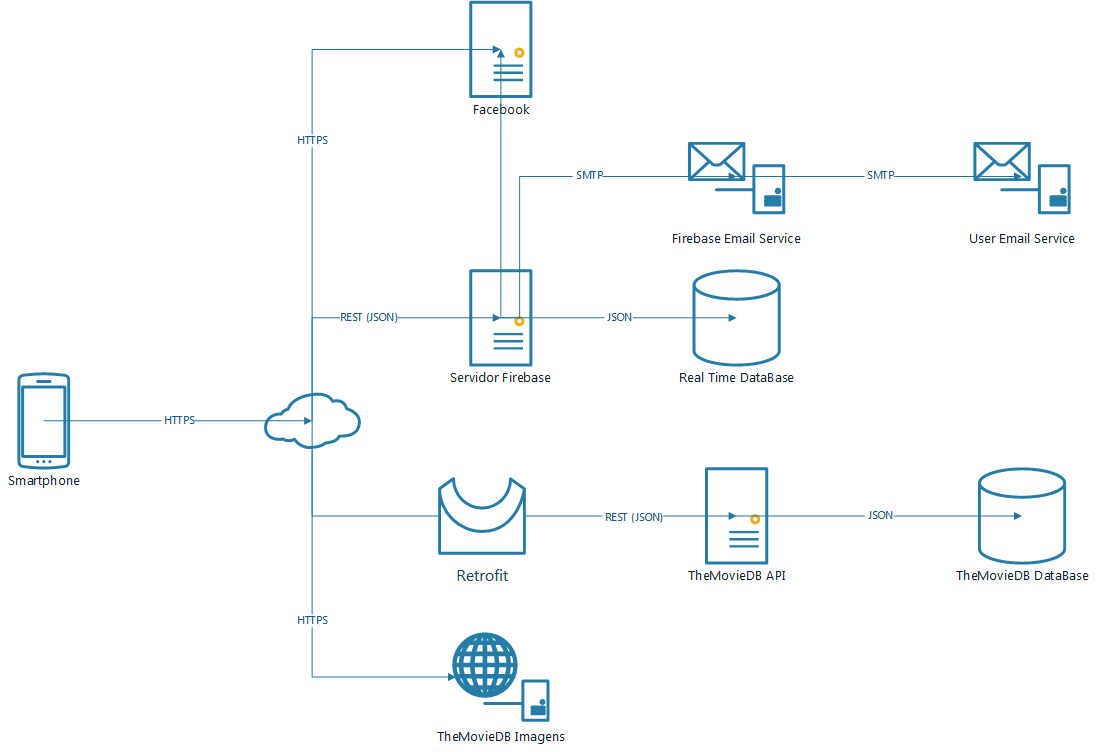


Figura nossa

# Avaliação do Aplicativo

Os usuários podem obter o aplicativo gratuitamente na própria loja do Android, apenas buscando pelo nome do aplicativo. No próximo tópico será ilustrado de forma simples a utilização das ferramentas desenvolvidas (Aplicativo) explicando algumas de suas utilidades.

## 3.1 Logotipo Zeugma

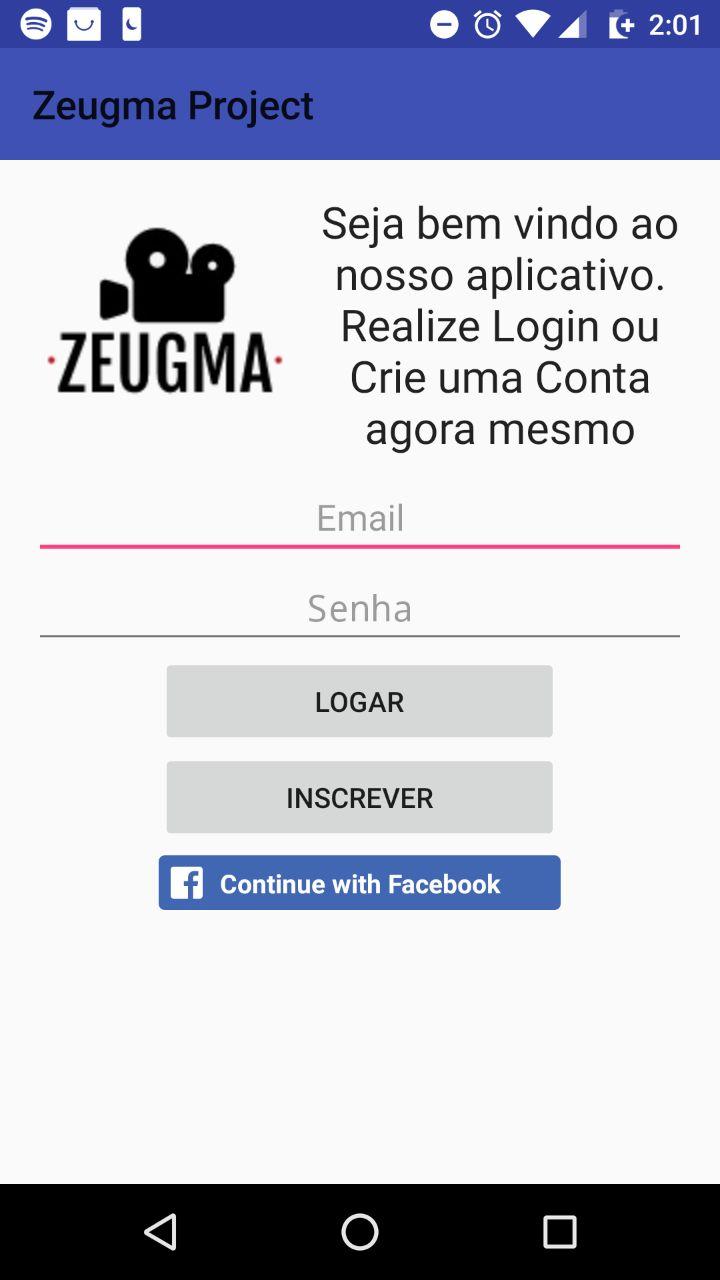
*Figura 8 – Logotipo do aplicativo*

****

Figura nossa

## Telas do Aplicativo Zeugma

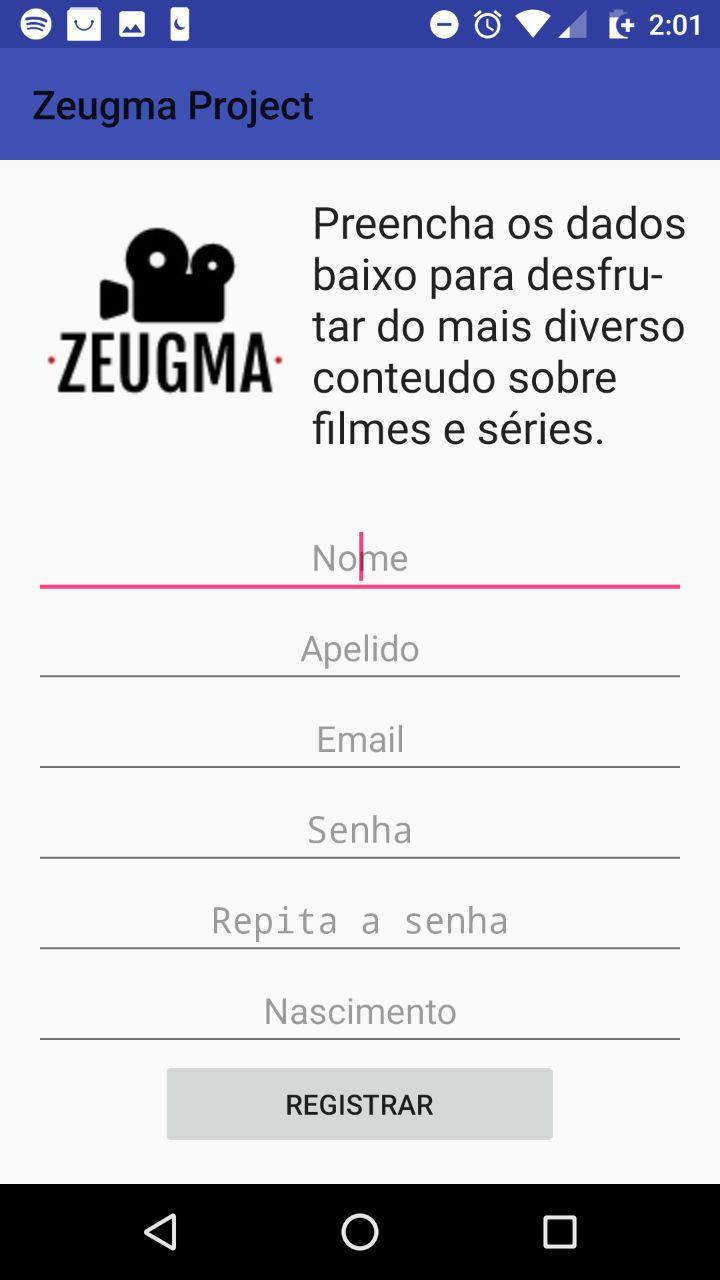
*Figura 9 - ScreenShot Tela de Login*



Print do aplicativo

Logo ao abrir o aplicativo, na Figura 9 o usuário deverá efetuar o login para ter acesso ao conteúdo do aplicativo. Se o usuário não for cadastrado ele poderá se cadastrar clicando no botão inscrever. E ainda há a possibilidade de conectar-se com o Facebook.

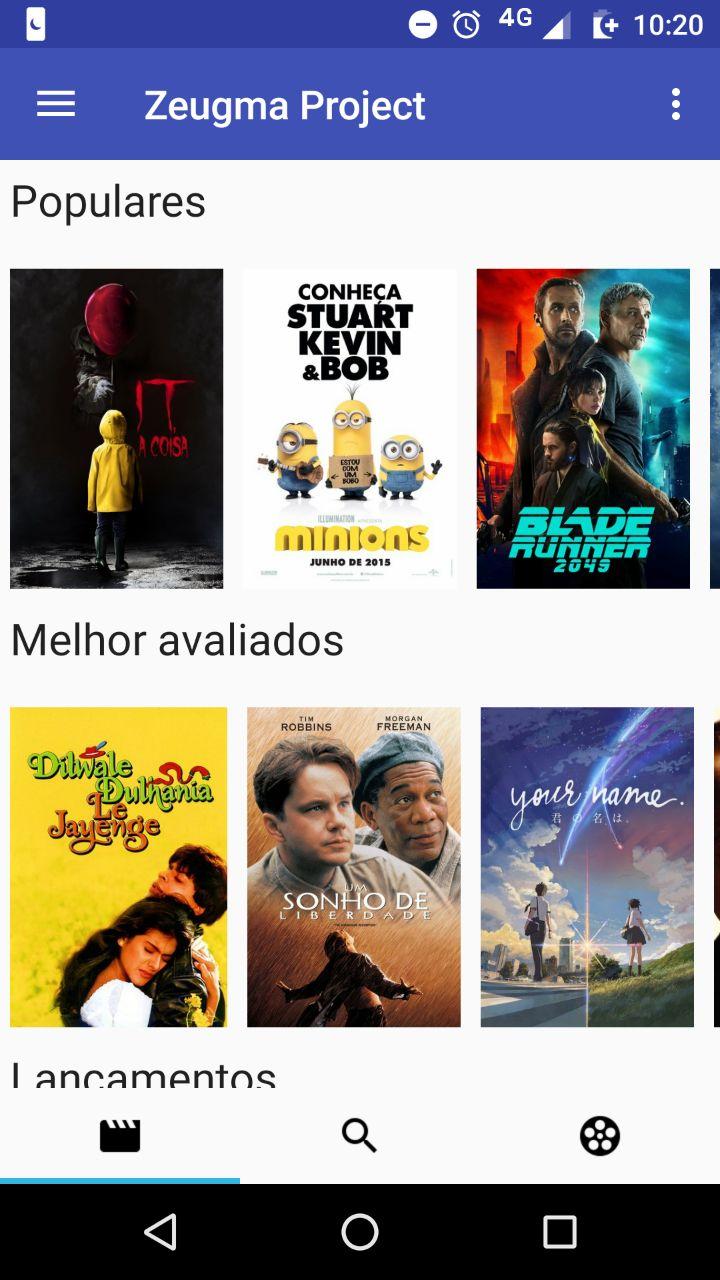
*Figura 10 - ScreenShot Tela de Cadastro*

****

Print do aplicativo

Na figura 10, apresenta-se a tela de cadastro, onde o usuário deverá inserir os dados pessoais para posteriormente efetuar o login. Nesta etapa também enviado um e-mail de confirmação de conta para esta ser ativada.

*Figura 11 – ScreenShot Tela Principal*

****

Print do aplicativo

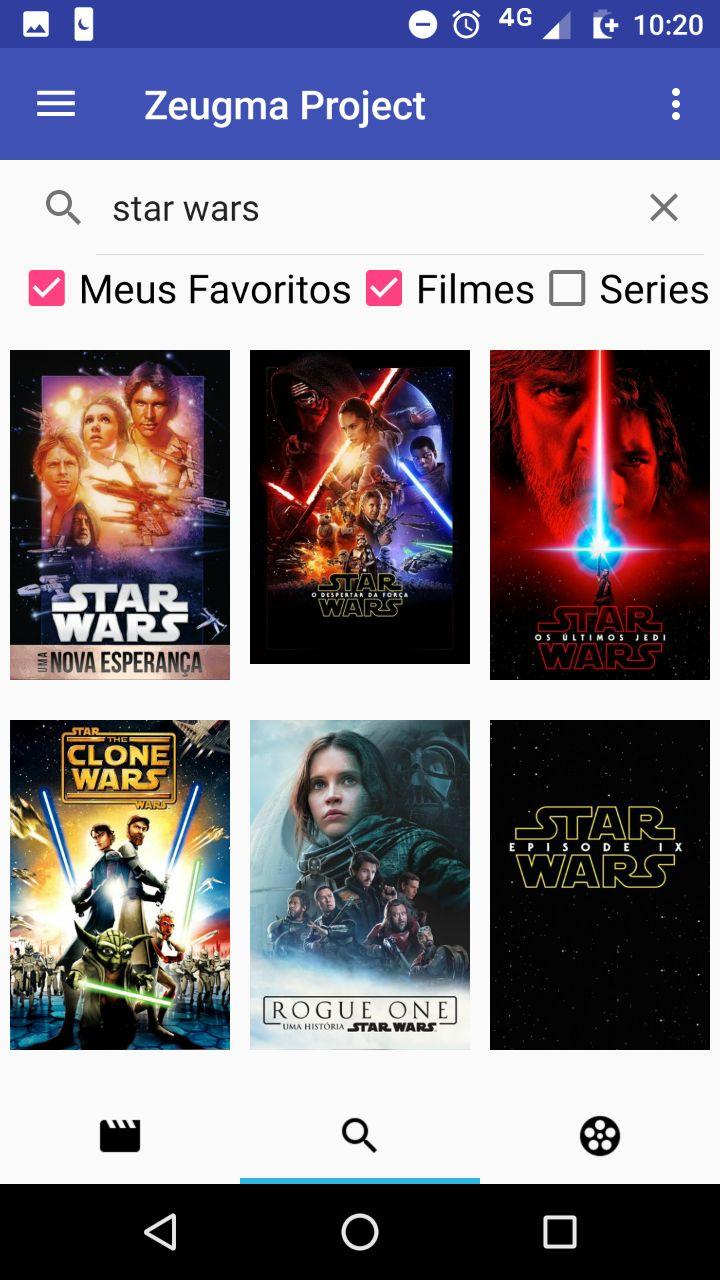
Após efetuar o login, na figura 11 o usuário terá acesso a todo conteúdo do aplicativo podendo ver informações sobre filmes e séries.

 Neste ícone são mostrados todos os filmes disponíveis no aplicativo, desde os recentes lançamentos aos melhores avaliados.

Ao clicar neste ícone o usuário conseguirá pesquisar por filmes e séries no qual deseja saber as informações sobre.

 Neste ícone são mostradas todas as séries que se encontram no aplicativo, também desde os recentes lançamentos aos melhores avaliados.

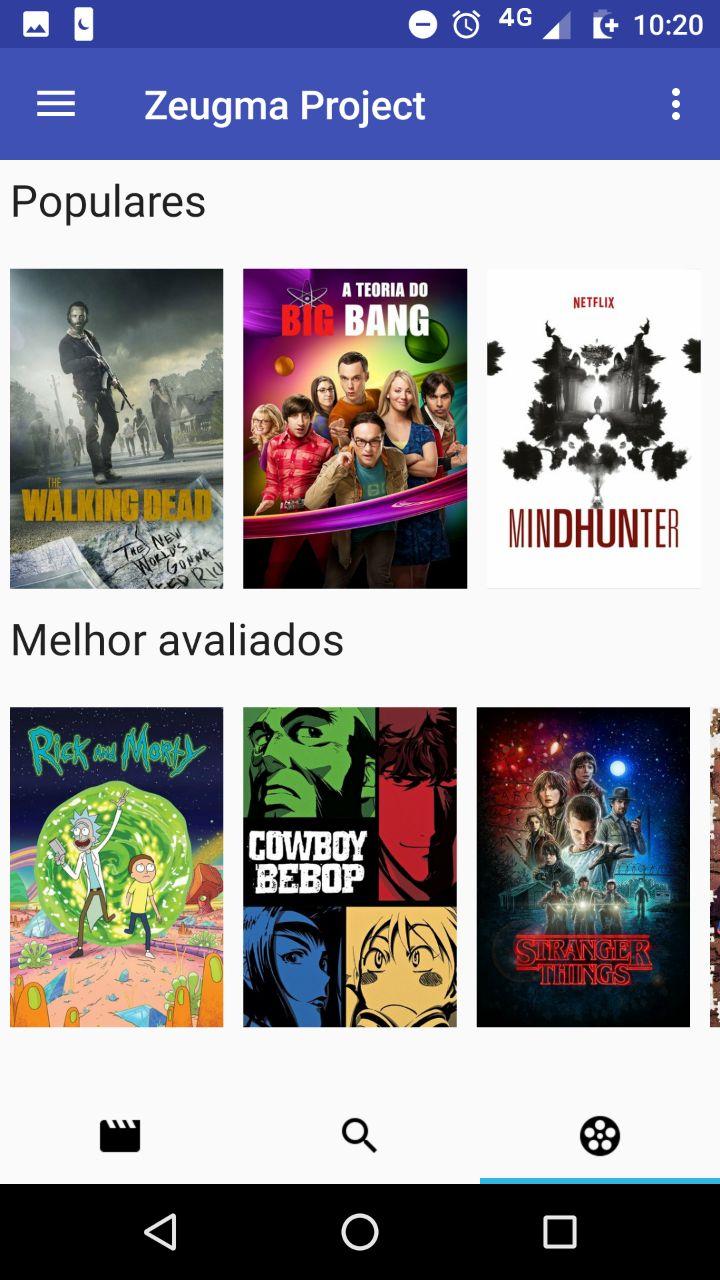
*Figura 12 - ScreenShot Tela de Pesquisa*

****

Print do aplicativo

Na figura 12, o usuário conseguirá pesquisar por filmes e séries no qual deseja saber as informações sobre.

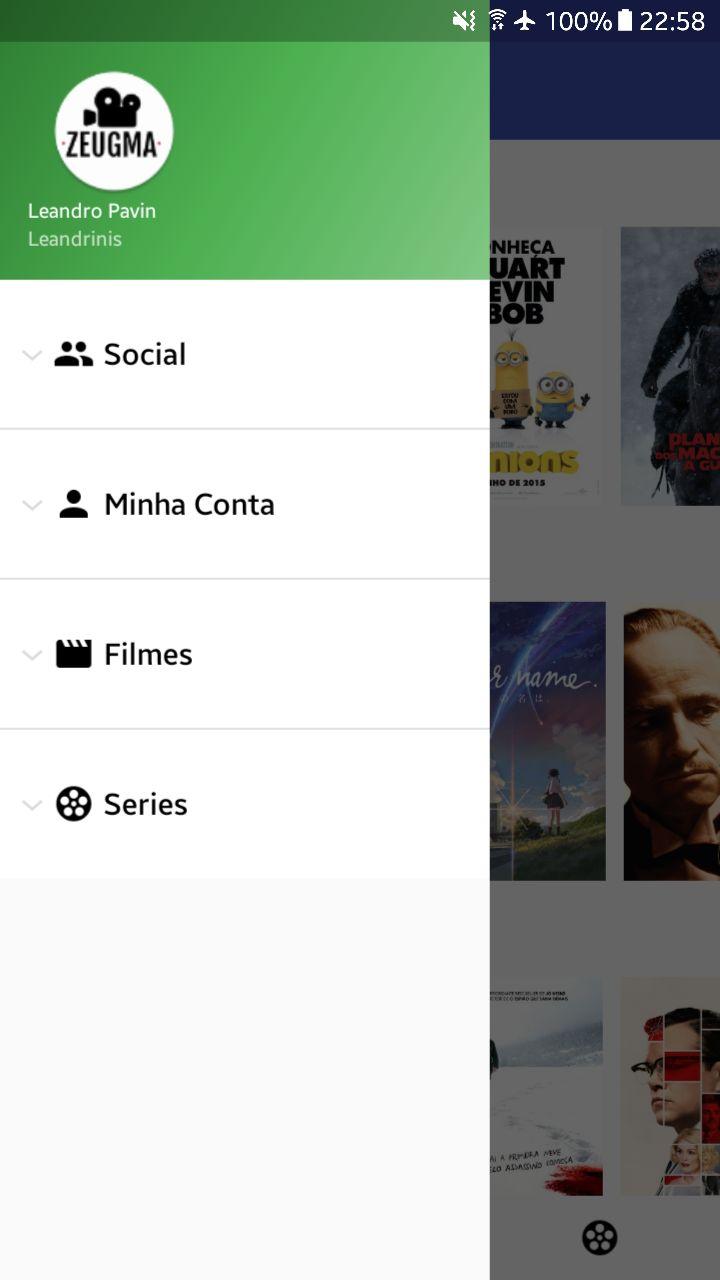
*Figura 13 - ScreenShot Tela de Séries*

****

Print do aplicativo

Na figura 13, são mostradas todos as séries disponíveis no aplicativo, desde os recentes lançamentos aos melhores avaliados.

*Figura 14 - ScreenShot Tela de Configurações*



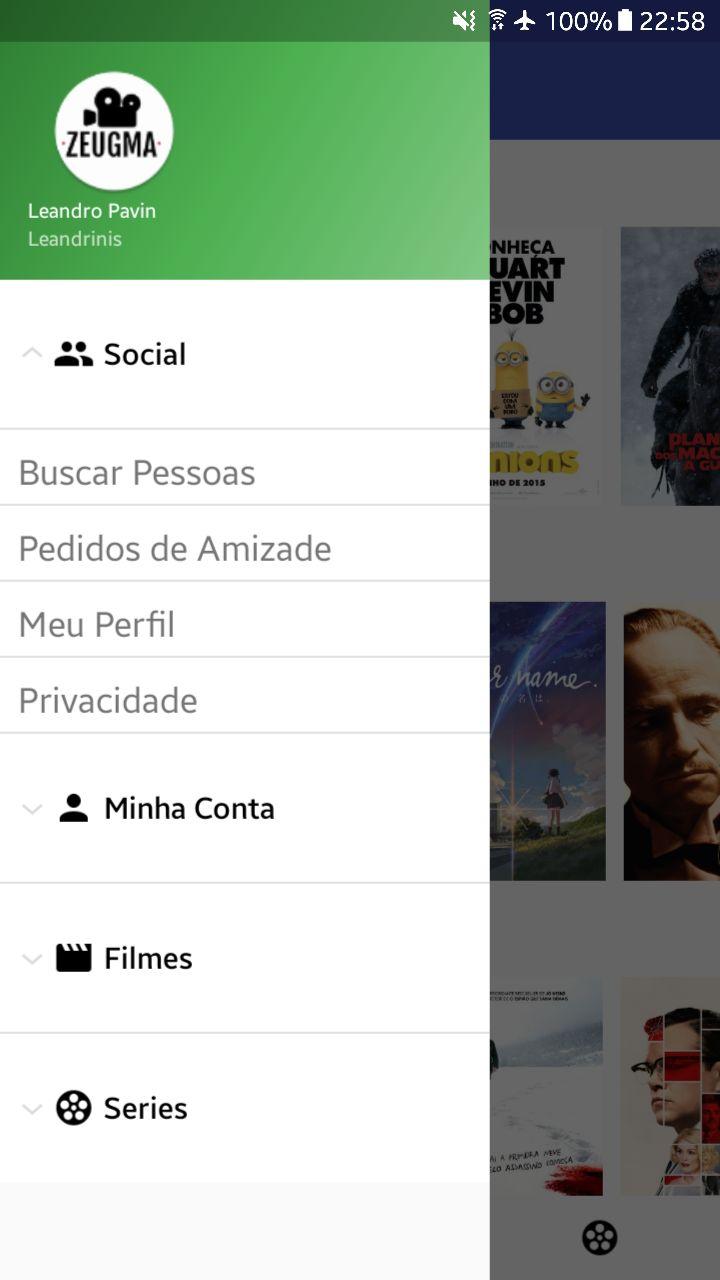
Print do aplicativo

Na figura 14, o usuário tem acesso a todas as configurações do aplicativo, sendo um menu geral onde o utilizador consegue acessar todas as outras telas do aplicativo.

 Ao clicar neste ícone o usuário consegue buscar e adicionar amigos procurando pelo nome da pessoa, além de conseguir acessar o próprio perfil e alterar as configurações de privacidade.

 Ao clicar neste ícone o utilizador terá acesso aos filmes e séries marcados como: favoritos, assistidos, curtidos, discutidos. Além disso também poderá acessar seus dados de registro, podendo a qualquer momento efetuar uma alteração.

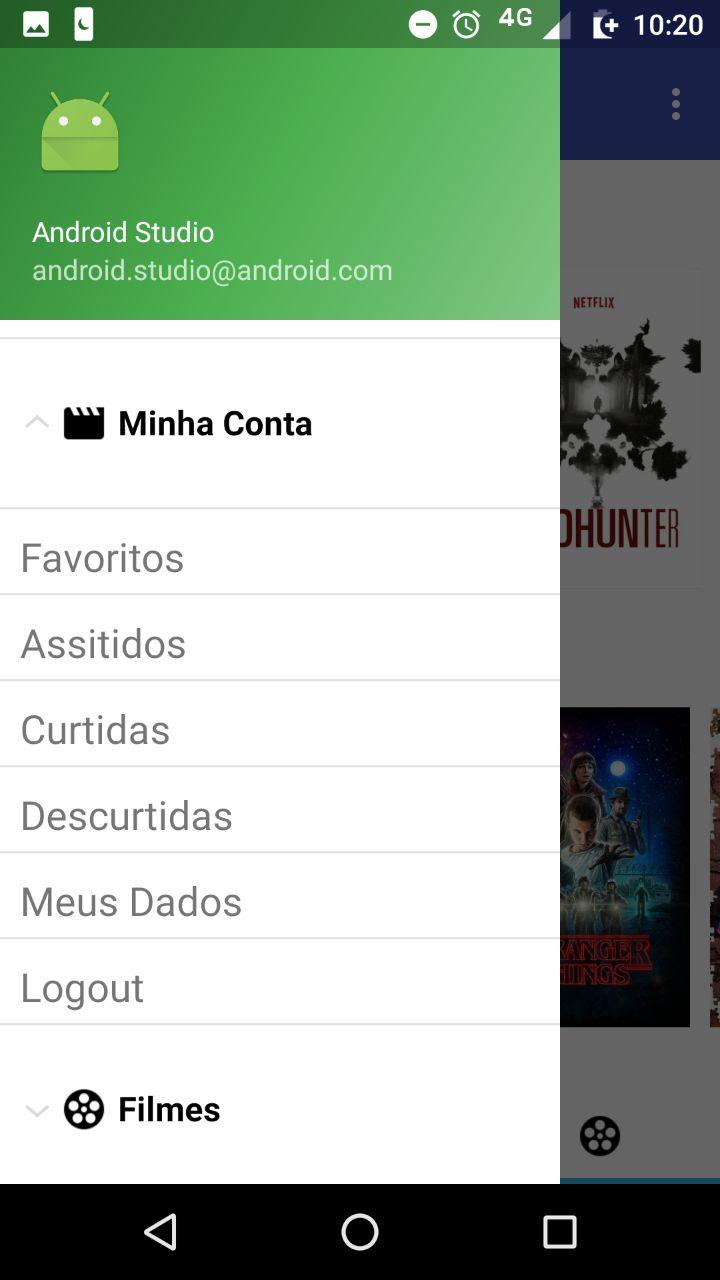
*Figura 15 - ScreenShot Tela de Configurações*



Print do aplicativo

Na figura 15, ao clicar em Buscar Pessoas, tem-se a possibilidade de adicionar amigos procurando pelo nome ou apelido, contem também acesso ao perfil e alterações de privacidade.

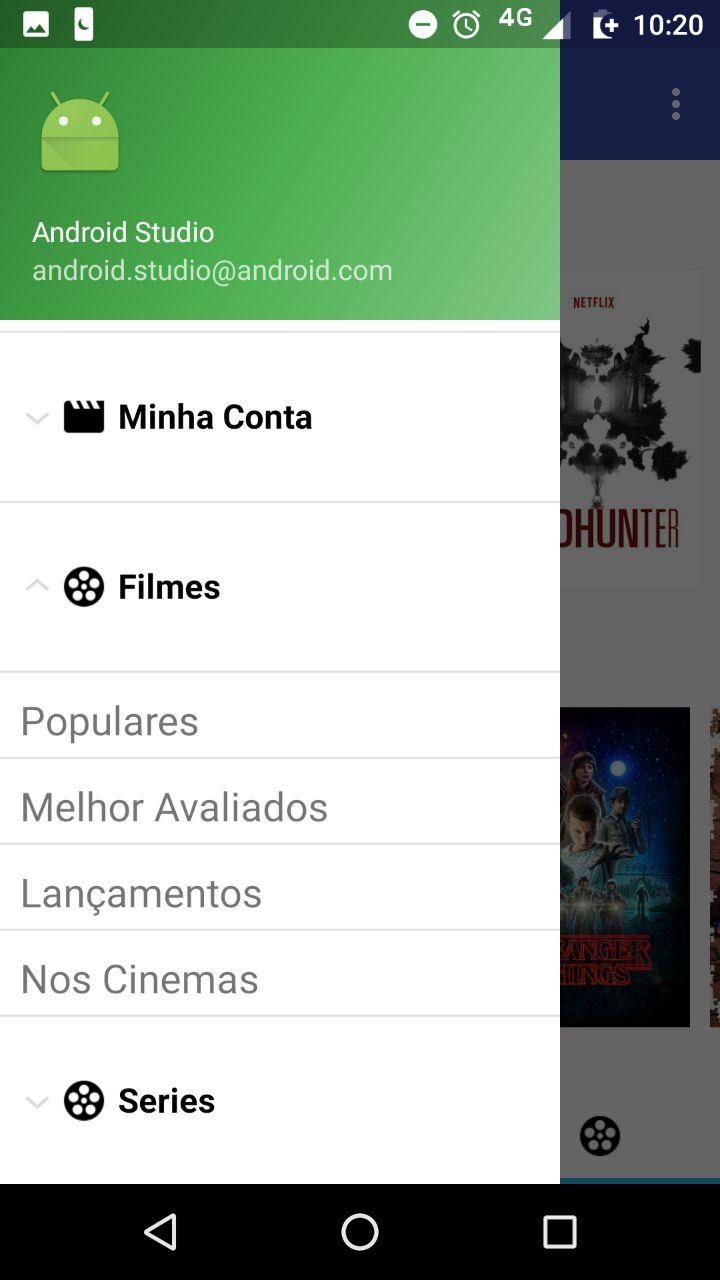
*Figura 16 - Continuação da Tela de Configurações*

****

Print do aplicativo

Na figura 16, é apresentado a conta pessoal de cada usuário, onde este poderá acessar os filmes ou séries salvas como favoritos, os marcados como assistidos, curtidas e discutidas, acesso aos dados da conta e efetuar o logout.

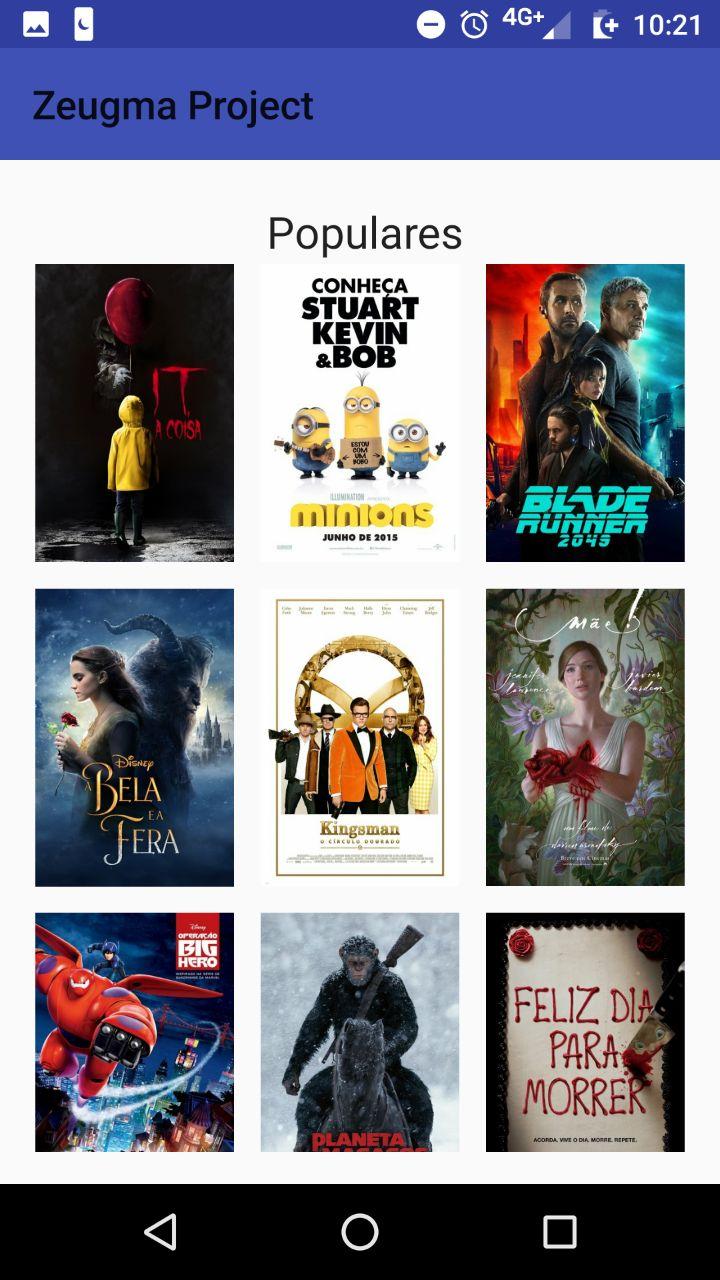
*Figura 17 - Continuação da Tela de Configurações*

****

Print do aplicativo

Na figura 17, o usuário conseguirá acessar os filmes por categoria, sendo elas: Populares, Melhor Avaliados, Lançamentos e os que ainda estão em cartazes Nos Cinemas. O menu de Séries apresenta as mesmas categorias do menu dos filmes porem baseado em séries.

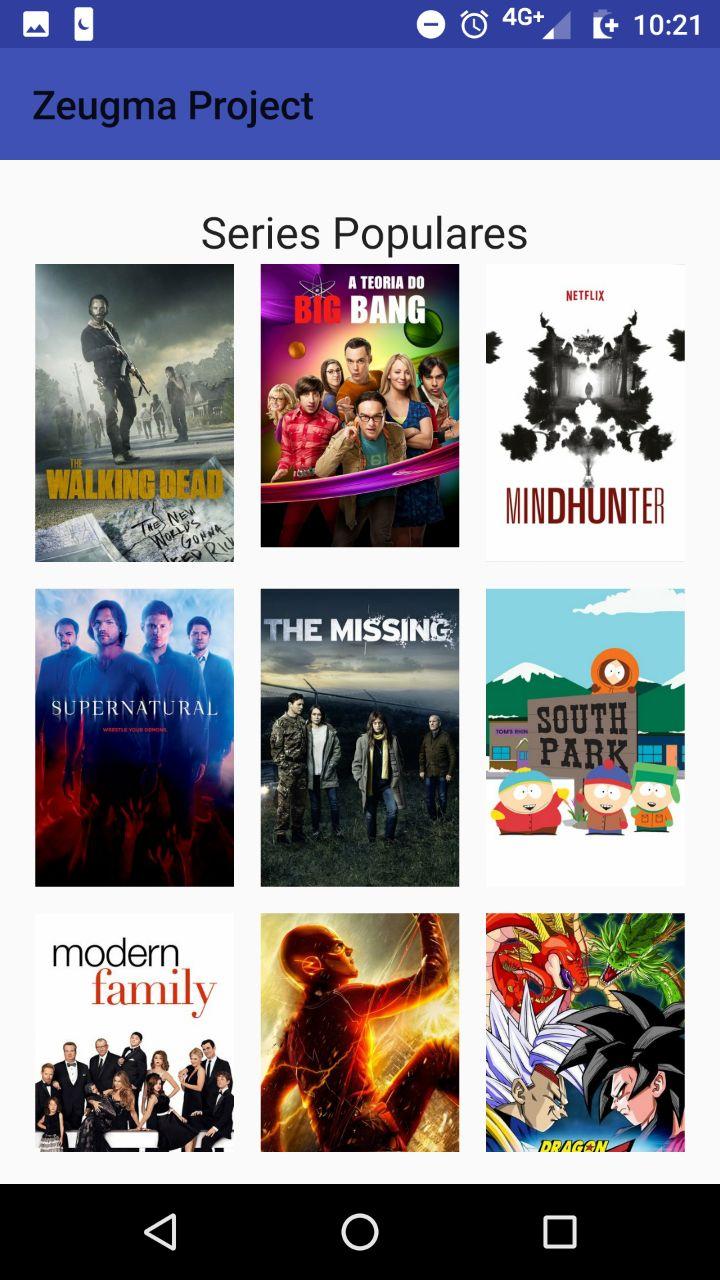
*Figura 18 - ScreenShot Tela de Filmes Populares*

****

Print do aplicativo

Na figura 18, são apresentados todos os filmes da categoria populares, ou seja, todos os filmes que estão em alta no momento.

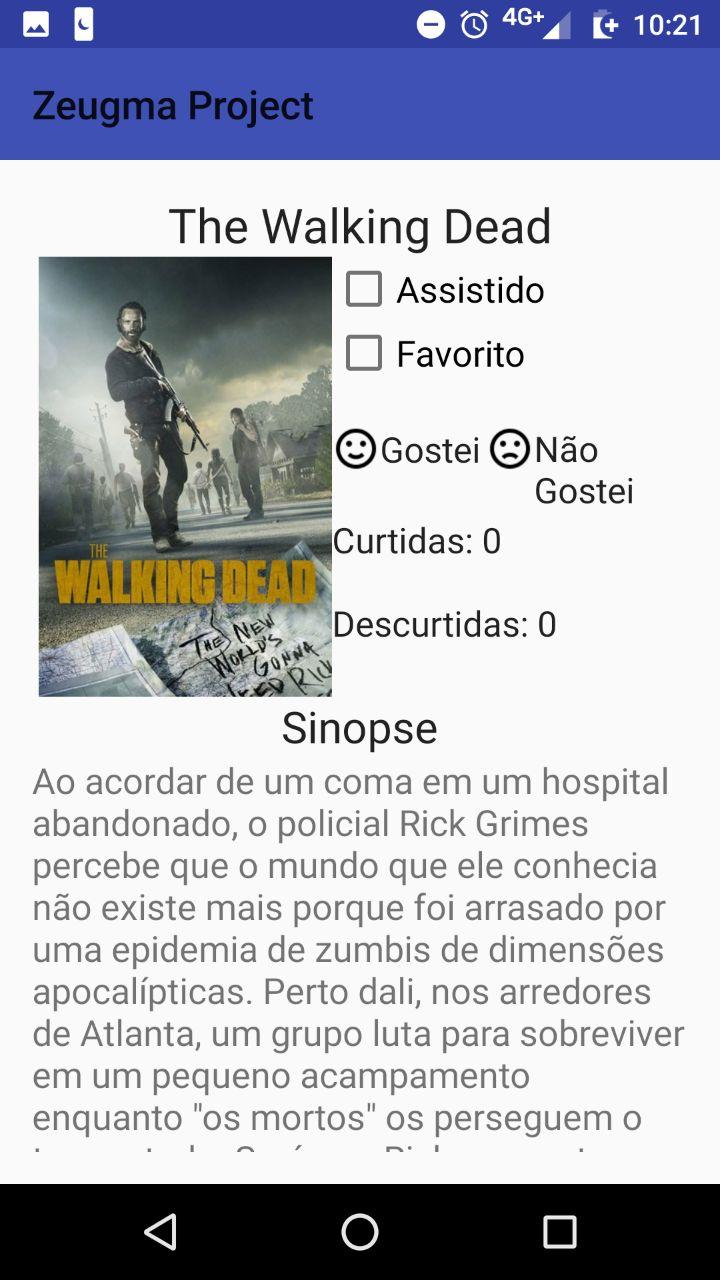
*Figura 19 - ScreenShot Tela de Séries Populares*

****

Print do aplicativo

Na figura 19, são apresentadas todos as séries da categoria populares, ou seja, todos as séries que estão em alta no momento.

*Figura 20 - ScreenShot da Tela de Informações de Séries*

****

Print do aplicativo

Na figura 20, são mostradas todas as informações decorrentes da série selecionada. Tendo a opção de marcá-la como assistida e favorita, além disso é possível classificar se gostou ou não da série e obter todas informações importantes da mesma como sinopse, gênero, temporadas, episódios, duração etc.

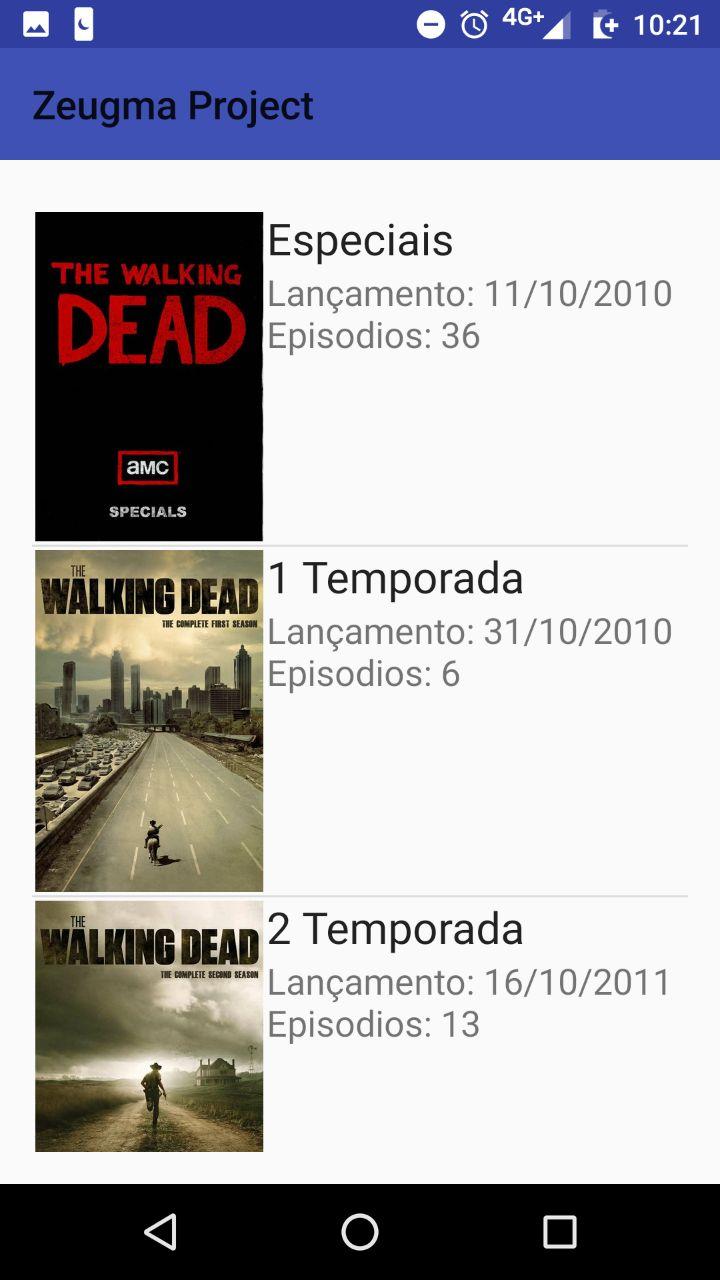
*Figura 21 - Continuação da Tela de Informações das Séries*

****

Print do aplicativo

Na figura 21, são mostradas todas as informações decorrentes da série selecionada. Tendo a opção de marcá-la como assistida e favorita, além disso é possível classificar se gostou ou não da série e obter todas informações importantes da mesma como sinopse, gênero, temporadas, episódios, duração etc.

*Figura 22- Continuação da Tela de Informações das Séries*

****

Print do aplicativo

Na figura 22, são mostradas todas as informações decorrentes da série selecionada, nesta etapa o utilizador conseguirá selecionar a temporada desejada, após a seleção é exibida uma lista de episódios.

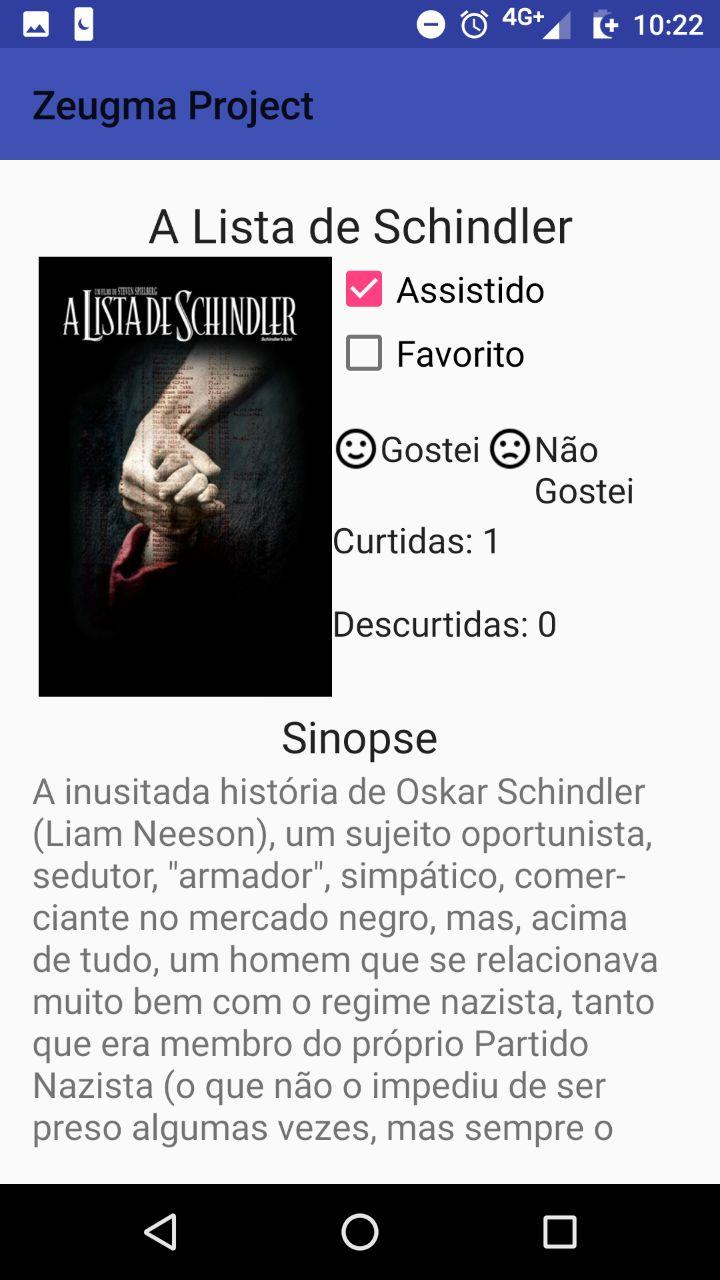
*Figura 23 - ScreenShot da Tela de Episódios das Séries*

****

Print do aplicativo

Na figura 23, são mostradas todas as informações decorrentes da série selecionada. Como dito anteriormente após selecionar a temporada desejada é exibida a lista de episódios, ao clicar em algum episódio especifico, o aplicativo mostrará todas as informações sobre o episódio selecionado.

*Figura 24 - ScreenShot da Tela de Informações de Filmes*

****

Print do aplicativo

Na figura 24, são mostradas todas as informações decorrentes do filme selecionado. Tendo a opção de marca-lo como assistido e favorito, além disso, é possível classificar se gostou ou não do filme e obter todas informações importantes do filme como sinopse, gênero, duração, lançamento, produtora etc.

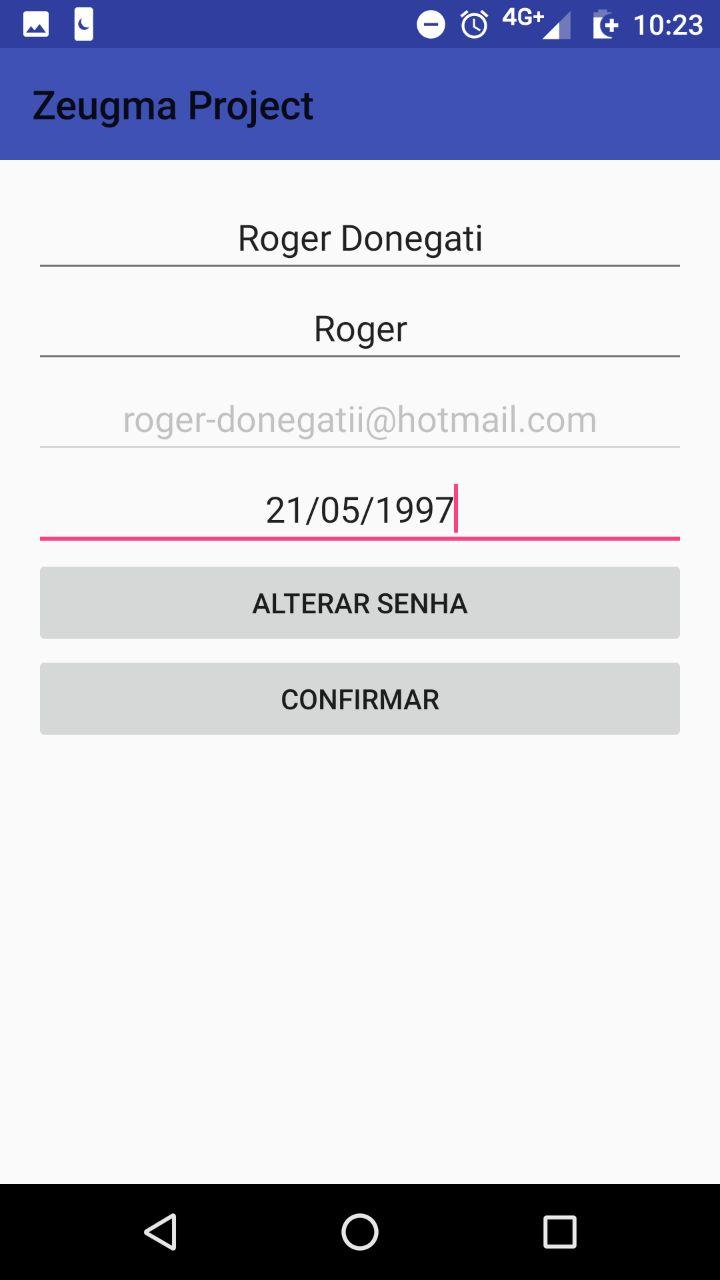
*Figura 25 - Continuação da Tela de Informações de Filmes*

****

Print do aplicativo

Na figura 25, são mostradas todas as informações decorrentes do filme selecionado. Tendo a opção de marca-lo como assistido e favorito, além disso, é possível classificar se gostou ou não do filme e obter todas informações importantes do filme como sinopse, gênero, duração, lançamento, produtora etc.

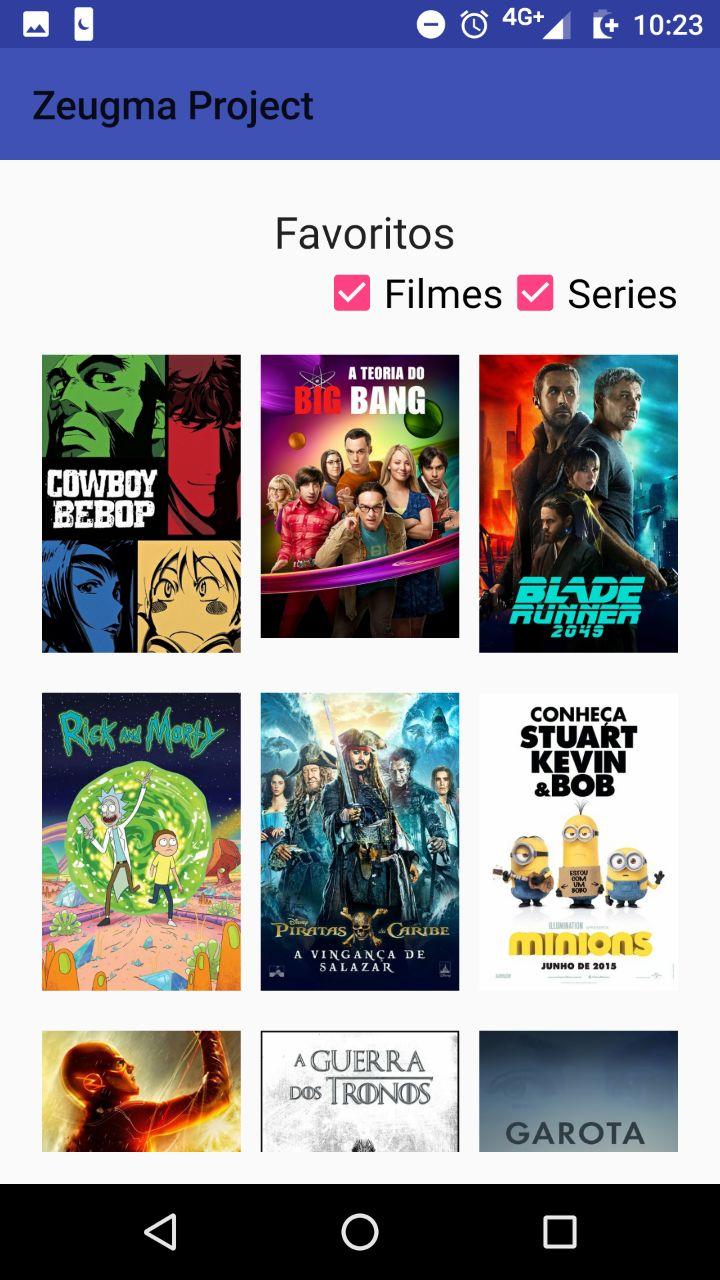
*Figura 26 - Screenshot da Tela de Alteração de Dados Pessoais*

****

Print do aplicativo

Na figura 26, tem-se a função de alterar os dados do usuário caso ele deseje, em caso de alteração de senha será encaminhado um e-mail de confirmação de alteração e só após a confirmação a alteração será efetivada.

*Figura 27 - Screenshot da Tela de Favoritos*

****

Print do aplicativo

Na figura 27, tem-se função de mostrar todos os filmes e séries salvas como favorito pelo usuário, podendo ser acessada a qualquer momento sem ter a necessidade de procurar pelo filme novamente.

*Figura 28 - Screenshot da Tela de Perfil do Usuário*

****

Print do aplicativo

Na figura 28, tem-se a função de mostrar o perfil de cada usuário. São mostradas informações pertinentes ao aplicativo, como tempo gasto com filmes e séries, os filmes marcados como: favoritos, assistidos, curtidos e discutidos. Apresenta-se também como recurso as disputas amigáveis entre os usuários, onde pode ser realizada, através do tempo gasto, com séries e filmes e a quantidade de filmes assistidos.

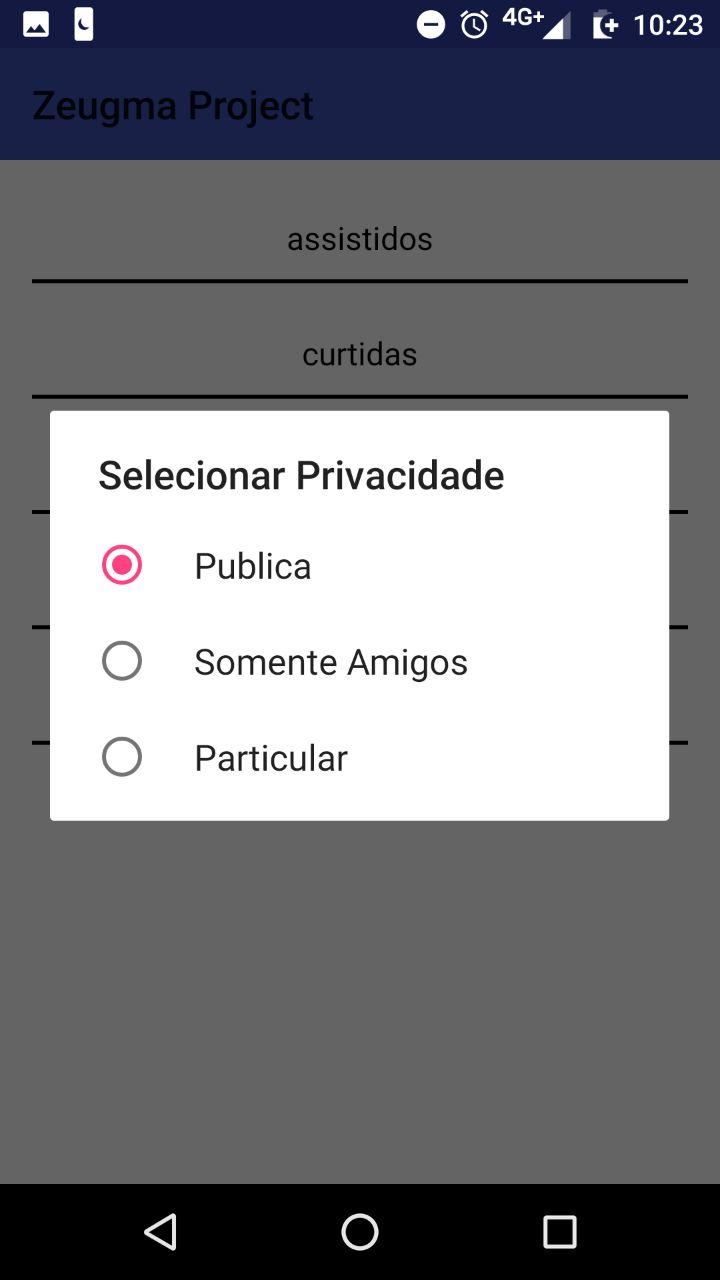
*Figura 29 - Screenshot da Tela de Privacidade*

****

Print do aplicativo

Na figura 29, encontra-se a função de configurar a privacidade de acordo com cada usuário, com três opções, publica, amigos e por fim privado.

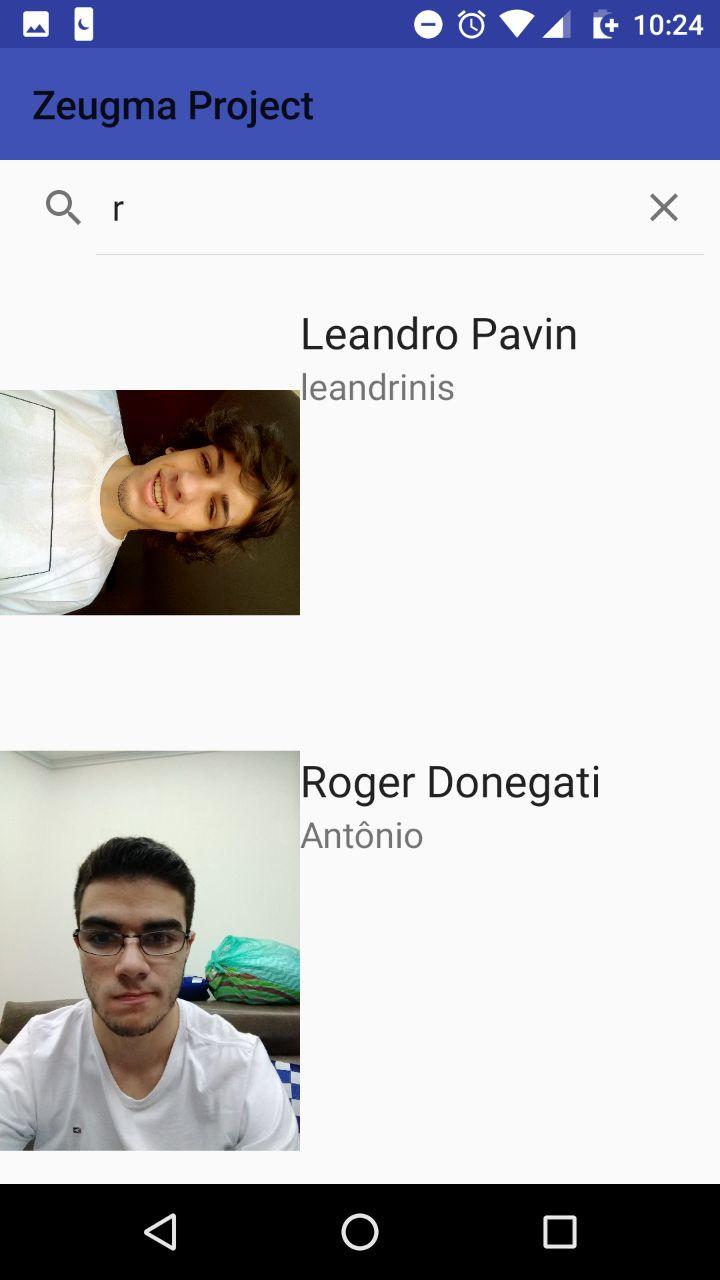
*Figura 30 - Continuação da Tela de Privacidade*

****

Print do aplicativo

Na figura 30, tem-se a função configurar a privacidade de acordo com cada usuário, o utilizador terá três opções, publica, amigos e por fim privado.

*Figura 31 - Screenshot da Tela de Busca de Amigos*

****

Print do aplicativo

Na figura 31, tem-se a função de buscar amigos, o usuário poderá digitar o nome de outro usuário e adiciona-lo como amigo, para comparar estatísticas e ver o seu perfi, a menos que esteja como privado.

## 3.3 Avaliação Heurística de Nielsen

Com a popularização de sistemas e aplicações destinados a pessoas comuns, as empresas passaram a ter que planejar como tornar interfaces cada vez mais simples e intuitivas. Jakob Nielsen após realizar testes e pesquisas obteve algumas “regras” que poderiam ser aplicadas para melhorar a interação humano computador desses sistemas, estas foram divididas em 10 heurísticas de usabilidade. Caso o sistema atenda todas as heurísticas propostas, ele pode ser considerado um sistema de um bom nível de usabilidade. Sendo assim, apresentam-se as heurísticas simplificadas por Marco (2016):

### **Visibilidade do status do sistema**

O sistema deve sempre manter os usuários informados sobre o que está acontecendo, através de feedback adequado dentro de um prazo razoável.

### **Correspondência entre sistema e mundo real**

O sistema deve falar o idioma dos usuários, com palavras, frases e conceitos familiares para o usuário, em vez de termos orientados para o sistema. Siga as convenções do mundo real, fazendo com que a informação apareça de forma natural e lógica.

### **Controle e liberdade do usuário**

Os usuários geralmente escolhem as funções do sistema por engano e precisarão de uma "saída de emergência" claramente marcada para deixar o estado indesejado sem ter que passar por um diálogo prolongado. Apoie desfazer e refazer.

### **Consistência e padrões**

Os usuários não devem ter que se perguntar se diferentes palavras, situações ou ações significam o mesmo. Siga as convenções da plataforma.

### **Prevenção de erros**

Mesmo melhor do que boas mensagens de erro é um design cuidadoso que impede que ocorra um problema em primeiro lugar. Elimine as condições propensas a erros ou verifique por elas e apresente aos usuários uma opção de confirmação antes de se comprometerem com a ação.

### **Reconhecimento em vez de recordar**

Minimize a carga de memória do usuário, tornando visíveis objetos, ações e opções. O usuário não deve ter que lembrar as informações de uma parte do diálogo para outra. As instruções para o uso do sistema devem ser visíveis ou facilmente recuperáveis ​​sempre que apropriado.

### **Flexibilidade e eficiência de uso**

Aceleradores - não vistos pelo usuário novato - muitas vezes podem acelerar a interação para o usuário especialista, de modo que o sistema possa atender a usuários inexperientes e experientes. Permitir que os usuários adaptem as ações frequentes.

### **Design estético e minimalista**

Os diálogos não devem conter informações que sejam irrelevantes ou raramente necessárias. Toda unidade extra de informação num diálogo compete com as unidades de informação relevantes e diminui a sua visibilidade relativa.

### **Ajude os usuários a reconhecer, diagnosticar e recuperar erros**

As mensagens de erro devem ser expressas em linguagem simples (sem códigos), indicam com precisão o problema e sugerem de forma construtiva uma solução.

### **Ajuda e documentação**

Mesmo que seja melhor se o sistema pode ser usado sem documentação, pode ser necessário fornecer ajuda e documentação. Qualquer informação desse tipo deve ser fácil de pesquisar, focada na tarefa do usuário, listar etapas concretas a serem realizadas e não ser muito grande.

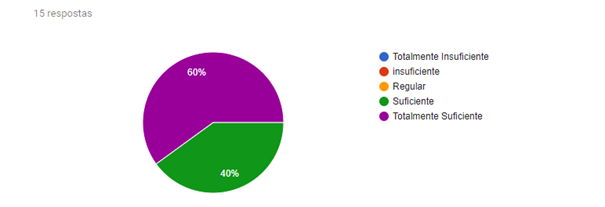
## Questionário de Avaliação da Usabilidade do Aplicativo Zeugma

Este questionário foi elaborado com o intuito de levantar os pontos fracos e pontos fortes do Aplicativo de gerenciamento de conteúdo online, o Zeugma.

Foi disponibilizado o uso do aplicativo à 15 estudantes do ensino superior da Faculdade de Tecnologia de São Caetano do Sul – Antônio Russo e em seguida aplicado o questionário.

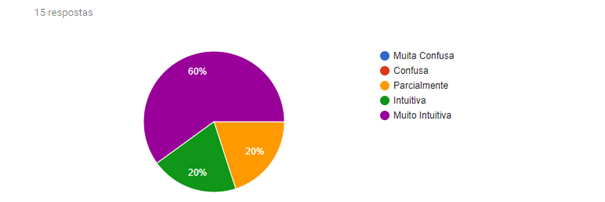
As informações coletadas através do mesmo serão apresentadas na documentação do aplicativo, validando as heurísticas de Nielsen nos testes de Usabilidade.

1. Ao utilizar o aplicativo, fica clara a visibilidade do status? Ou seja, é bem definida a tarefa que você está realizando no aplicativo?



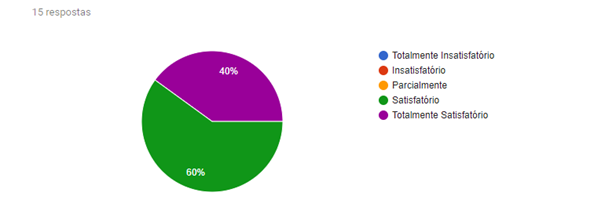
Com base na pesquisa é possível afirmar que a heurística de visibilidade de status do sistema é atendida.

1. Quanto ao aplicativo, há uma boa correspondência entre o sistema e o mundo real? Considerando suas representações gráficas como ícones e imagens. São de fácil entendimento e autoexplicativas?



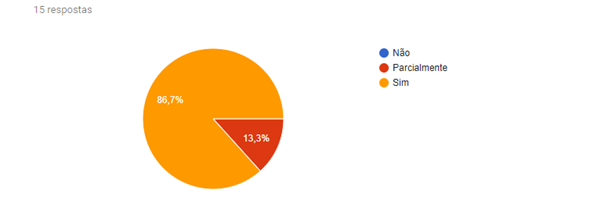
Com base na pesquisa é possível afirmar que a heurística de correspondência entre sistema e mundo real é atendida.

1. Ao utilizar o aplicativo, você sente que tem controle e liberdade para realizar as tarefas desejadas?

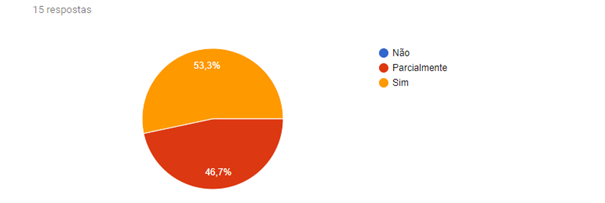


Com base na pesquisa é possível afirmar que a heurística de controle e liberdade do usuário é atendida.

1. Quanto ao design do aplicativo, o mesmo segue os padrões utilizados em dispositivos móveis Android?

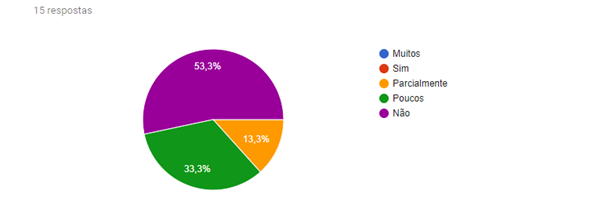


Com base na pesquisa realizada é possível afirmar que a heurística de consistência e padrões foi satisfeita.

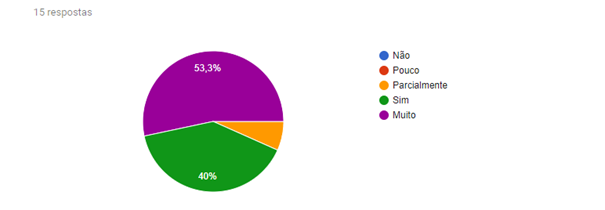
1. O aplicativo fornece uma interface que te ajuda a prevenir erros durante a utilização? 

Com base na pesquisa realizada é possível afirmar que a heurística de prevenções de erros foi satisfeita.

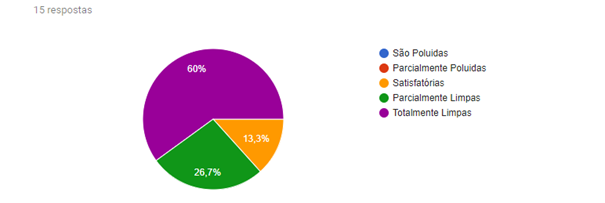
1. Para navegar entre as telas do aplicativo, é necessário a memorização dos “caminhos” para chegar no destino desejado?



Com base na pesquisa realizada é possível afirmar que a heurística reconhecimento em vez de recordar é atendida.

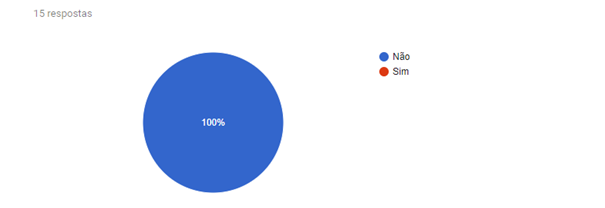
1. O aplicativo disponibiliza uma navegação flexível e eficiente? 

Com base na pesquisa realizada é possível afirmar que a heurística flexibilidade e eficiência de uso foi satisfeita.

1. As telas do aplicativo são esteticamente limpas, e sem muita informação desnecessária?

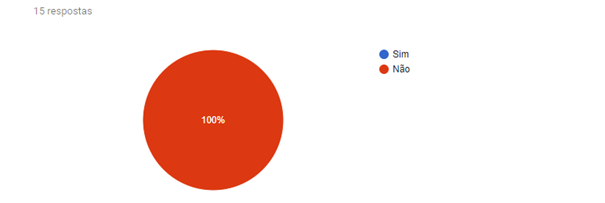
Com base na pesquisa realizada é possível afirmar que a heurística de design estético e minimalista é atendida

1. Existe a opção de feedback no aplicativo, sobre erros e sugestões.



O aplicativo não prove de uma interface de feedback para reportar erros e sugestões de usuários. A heurística de ajuda aos usuários de reconhecer, diagnosticar e recuperar erros não foi atendida.

1. É necessário ajuda e documentação para a navegação no aplicativo?



Com base na pesquisa realizada é possível afirmar que a heurística de ajuda e documentação não foi satisfeita, pois os usuários optaram por dispensar materiais de ajuda e documentação.

## Considerações sobre a avaliação heurística

A avalição heurística retornou resultados importantes sobre a usabilidade do aplicativo, ficando claro quais foram os pontos fortes e os pontos onde o aplicativo pode ser melhorado.

Analisando as informações levantadas nota-se que houveram duas heurísticas que não foram atendidas, a primeira delas sendo a nona heurística de Nielsen “ajude os usuários a reconhecer, diagnosticar e recuperar erros”. Quanto a essa heurística é possível que o questionário não tenha passado a ideia esperada tornando assim o seu resultado questionável. Embora o aplicativo aparente não comtemplar totalmente a heurística há alguns pontos que poderiam ter sido melhor abordados na questão referente a essa heurística. É valido ser considerado também o fato de que com as ferramentas atuais, o envio de diagnósticos e relatórios de erros pelo usuário tem se tornado menos necessário, pois tais ferramentas possuem sistemas automatizados para detecção de erros.

A segunda heurística não atendida; “ajuda e documentação”, embora prevista por Nielsen como algo necessário e positivo, deve ser considerado o período e o ambiente em que as heurísticas foram descritas. Essas foram originalmente criadas para sistemas de computadores e embora também se encaixem bem para aplicações móveis existem certos adendos a serem revistos. Com o avanço da tecnologia, os usuários têm-se acostumado ao manuseio de aplicações móveis dispensando e até mesmo evitando o uso de materiais de ajuda e documentação como mostrado nos resultados do questionário.

# Considerações Finais

Com a realização deste trabalho, considera-se que, atualmente os sistemas estão cada vez mais distribuídos, procurando sempre a facilidade na utilização de recursos e ferramentas, uma ampla conectividade entre serviços e compartilhamento de informações. Muitas vezes empresas necessitam de certas informações e recursos e por meio dessa reutilização de serviços, tempo e qualidade vem sendo ganho.

Tecnologias foram criadas para que esse tipo de comunicação fosse possível e viável, em um mundo onde tudo é informatizado, conectado a Web e a informação precisa estar o mais rápido possível onde necessitada, os Web services e API’s compartilhadas por exemplo, são essenciais e cada vez mais reais.

Para que fosse possível utilizar tais ferramentas neste trabalho, foi necessário o estudo e aprendizado de certos recursos, os principais são: JAVA, JavaScript, protocolo JSON, AndroidStudio, REST, padrão arquitetural SOA (Service-Oriented Architecture) e Web services.

A realização do projeto atingiu os objetivos almejados na fase de planejamento. O aplicativo é funcional e atende aos padrões de usabilidade criados por Nielsen, todas as tecnologias e recursos necessários para o desenvolvimento do mesmo foram aprendidas e utilizadas em diversos conceitos abordados. A produção desse projeto possibilitou a expansão dos horizontes de aprendizado e instigou a continuação do projeto pós curso.

O aplicativo será publicado para download gratuito dentro da maior loja de aplicativos da plataforma android e continuará recebendo atualizações como a implementação de novos filtros de busca, uma interface para feedback dos usuários, integração com novas API’s de login e melhorias no desempenho, buscando sempre a excelência em usabilidade, confiabilidade, disponibilidade e integridade.

# Referências

ADOROCINEMA. **Filmes, trailers, horários e salas de cinema, Notícias, criticas.** Disponível em: <http://www.adorocinema.com/>. Acesso em 04 de nov de 2017.

AUMONT, Jacques; MARIE, Michel. **Dicionário teórico e crítico de cinema.** Trad. Eloisa Araújo Ribeiro. Campinas: Papirus, 2003.

BARTHES, Roland. Entretien avec Roland Barthes – Entrevista com Michel Delahaye e Jacques Rivette. **Cahiers du cinéma**, n. 147, tomo XXV, setembro de 1963, p. 20-30.

CROCKFORD, Douglas. **The application/JSON media type for javascript object notation (JSON)**. (2006).

CYGNET INFOTECH. **REST- An Ideal Web Service to Integrate Applications/Products.** 2017. Disponivel em <<http://www.cygnet-infotech.com/blog/rest-an-ideal-web-service-to-integrate-applications>> Acesso em 24 de out 2017.

DEITEL, Harvey; DEITEL, Paul. **Java for programmers**. 2ª ed. EUA Springer, 2015.

DEVMEDIA. **Introdução ao formato JSON.** 2012. Disponível em <http://www.devmedia.com.br/introducao-ao-formato-JSON/25275> Acesso em 24 de out 2017.

ERL, Thomas. **Soa: principles of service design**. Upper Saddle River: Prentice Hall, 2008.

ERL, Thomas. **Service-oriented architecture (SOA): concepts, technology, and design**. 2005.

FIELDING, Roy Thomas; **Doctor of philosophy.** University of California, Irvine 2000.

GAMA, Alexandre. **O que é JSON**. Disponível em: < [www.devmedia.com.br/o-que-e-JSON](http://www.devmedia.com.br/o-que-e-json)> Acesso em 23 de jun 2017.

IMDb. **Movies, TV and Celebrities.** Disponível em: <http://www.imdb.com/>. Acesso em 04 de nov de 2017.

JAVA FREE ORG: **Forum Main Page - Uol** Disponivel em: <javafree.uol.com.br> Acesso em 02 de nov de 2017

JSON. **Introdução ao JSON**. Disponível em <http://www.JSON.org/JSON-pt.html>Acesso em 24 de out 2017.

LECHETA, R. Ricardo. **Web services Restful.** 1ª ed. Rio de Janeiro: Editora Novatec, 2015.

MAUERHOFER, Hugo. A psicologia da experiência cinematográfica. In: XAVIER, Ismail (org.). **A experiência do cinema:** antologia. Trad. Teresa Machado. Rio de Janeiro: Edições Graal; Embrafilme, 1983, p. 375-380.

MOLICH, R., and NIELSEN, J. **Improving a human-computer dialogue**, *Communications of the ACM* 33, 3 (March), 338-348, 1990.

NIELSEN, J., and MOLICH, R. **Heuristic evaluation of user interfaces**, *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1-5 April), 249-256, 1990.

NIELSEN, J. **Enhancing the explanatory power of usability heuristics**. *Proc. ACM CHI'94 Conf.* (Boston, MA, April 24-28), 152-158, 1994a

NIELSEN, J. **Heuristic evaluation**. In NIELSEN, J., and MACK, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY, 1994b.

Marco, Bruno. **10 Heurísticas de Nielsen. Uma fórmula pra evitar erros básicos de usabilidade.**

PROLIFICS. **IT Solutions: Customized to Your Advantage.** Disponível em: <https://prolifics.com/blog/mobile-middleware-service-oriented-architecture-vs-mobility-service> Acesso em 08 de nov de 2017

SIMBA TECH. **Data Connectivity and Data Integration Solutions.** Disponível em: <https://www.simba.com/blog/remote-data-access-on-mobile-devices/> Acesso em 04 de nov de 2017

SNELL, James; TIDWELL, Doug; KULCHENKO, Pavel. **Webservice Programing erung mit SOAP**. O'Reilly Germany, 2002.

WAZLAWICK, Raul. **Metodologia de pesquisa para ciência da computação**. Elsevier Brasil, 2008.

# APÊNDICE A – Diagrama de Classes de ProjetoC:\Users\Duperron Tecnologia\Desktop\Class Diagram0.png

# APÊNDICE B – Questionário de Avaliação da Usabilidade do Aplicativo Zeugma

1- Ao utilizar o aplicativo, fica clara a visibilidade do status? Ou seja, é bem definida a tarefa que você está realizando no aplicativo?

* Totalmente insuficiente
* Insuficiente
* Regular
* Suficiente
* Totalmente suficiente

2- Quanto ao aplicativo, há uma boa correspondência entre o sistema e o mundo real? Considerando suas representações gráficas como ícones e imagens. São de fácil entendimento e autoexplicativas?

* Muito confusa
* Confusa
* Parcialmente
* Intuitiva
* Muito intuitiva

3- Ao utilizar o aplicativo, você sente que tem controle e liberdade para realizar as tarefas desejadas?

* Totalmente insatisfatório
* Insatisfatório
* Parcialmente
* Satisfatório
* Totalmente satisfatório

4- Quanto ao design do aplicativo, o mesmo segue os padrões utilizados em dispositivos móveis Android?

* Não
* Parcialmente
* Sim

5- O aplicativo fornece uma interface que te ajuda a prevenir erros durante a utilização?

* Não
* Parcialmente
* Sim

6- Para navegar entre as telas do aplicativo, é necessário a memorização dos “caminhos” para chegar no destino desejado?

* Muitos
* Sim
* Parcialmente
* Poucos
* Não

7- O aplicativo disponibiliza uma navegação flexível e eficiente?

* Não
* Pouco
* Parcialmente
* Sim
* Muito

8- As telas do aplicativo são esteticamente limpas, e sem muita informação desnecessária?

* São poluídas
* Parcialmente poluídas
* Satisfatórias
* Parcialmente limpas
* Totalmente limpas

9- Existe a opção de feedback no aplicativo, sobre erros e sugestões?

* Não
* Sim

10- É necessário ajuda e documentação para a navegação no aplicativo?

* Sim
* Não