

Luis Oswaldo dos Santos Ganoza

**Plataforma tecnológica para pesquisas sobre
saúde mental e tempo de uso de smartphones**

Florianópolis

2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Ganoza, Luis Oswaldo

Plataforma tecnológica para pesquisas sobre saúde mental e tempo de uso de smartphones / Luis Oswaldo Ganoza ; orientador, Jônata Tyska Carvalho, 2022.

61 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Ciências da Computação, Florianópolis, 2022.

Inclui referências.

1. Ciências da Computação. 2. desenvolvimento mobile. 3. sobrecarga de informação. 4. sobrecarga de comunicação. 5. saúde mental. I. Tyska Carvalho, Jônata. II. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. III. Título.

Luis Oswaldo dos Santos Ganoza

Plataforma tecnológica para pesquisas sobre saúde mental e tempo de uso de smartphones

Monografia submetida ao Programa de Graduação em Ciência da Computação para a obtenção do Grau de Bacharel.

Universidade Federal de Santa Catarina
Departamento de Informática e Estatística
Ciência da Computação

Orientador: Jônata Tyska Carvalho

Florianópolis

2022

Plataforma tecnológica para pesquisas sobre saúde mental e tempo de uso de smartphones

Monografia submetida ao Programa de Graduação em Ciência da Computação para a obtenção do Grau de Bacharel.

Trabalho aprovado. Florianópolis, 26 de dezembro de 2022:

Jônata Tyska Carvalho
Orientador

Renato Fileto
Convidado 1

Mateus Grellert da Silva
Convidado 2

Florianópolis

2022

Agradecimentos

Os primeiros agradecimentos vão para minha família por todo amor, carinho e suporte que recebi e recebo até hoje, principalmente o emocional. Meus pais Rosângela e Javier. Minha tia Rosane. Minha irmã Gisely. A pessoa que sou hoje é graças à vocês.

Também gostaria de agradecer ao meu Orientador Jônata Tyska Carvalho pela atenção, apoio, empolgação, conhecimento e principalmente paciência. Nenhum dos dois imaginava que o trabalho levaria tanto tempo para ficar pronto, mas cá estamos.

Depois, agradeço a meus amigos. No começo pensei em tomar o tempo de mencionar cada pessoa, cada uma que foi importante para mim porém sinto que vou esquecer alguém e me culpar até fazer um novo trabalho como esse para incluir a pessoa. Por isso agradecerei de forma genérica. "Amigos". Vocês não conseguem imaginar a importância de vocês na minha jornada pela universidade e pela vida e o quão grato eu sou por cada minuto com vocês. Fê tu é os guri.

Meus últimos agradecimentos são para a comunidade da UFSC como um todo e para os membros da Gestão Turing-Completa do Centro Acadêmico Livre da Computação.

Resumo

A imensa quantidade de dados aos quais somos expostos cotidianamente em virtude do desenvolvimento da Internet e da popularização dos *smartphones*, assim como a intensificação dos processos de comunicação que tais ferramentas proporcionam, levaram ao aprofundamento de condições conhecidas como sobrecarga de informação (*information overload*) e sobrecarga de comunicação (*communication overload*). Diversos estudos que averiguam as consequências de tais condições vêm sendo feitos. A principal consequência é o impacto negativo à saúde mental do indivíduo afetado, porém elas podem ir desde incapacidade de realizar tarefas que exponham o indivíduo à tamanha imensidade de informações, até dificultando os processos de comunicação cotidianos. Apesar do número de estudos já realizados na área, a maioria faz uso de dados de tempo de utilização do celular fornecidos pelos participantes. Além disso podemos encontrar outros estudos que deixam claro a diferença existente entre a percepção de um usuário quanto ao seu tempo de utilização do celular e o tempo real de utilização. Este trabalho tem como objetivo desenvolver uma plataforma tecnológica com capacidade de realizar pesquisas que envolvam coleta automática de dados de tempo de uso de *smartphones*, combinando esses dados com respostas dos usuários em escalas de avaliação de saúde mental cadastrada por pesquisadores da área.

Palavras-chaves: sobrecarga de informação. sobrecarga de comunicação. desenvolvimento mobile. saúde mental;

Sumário

1	INTRODUÇÃO	11
1.1	Motivação	11
1.2	Objetivos	12
1.3	Método de Pesquisa	13
1.4	Organização do Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Sobrecarga de informação e comunicação	15
2.2	Revisão do estado da arte em pesquisas de saúde mental e <i>smarpho- nes</i>	16
2.2.1	Protocolo de pesquisa	16
2.2.2	Resultados	16
2.3	Conclusão	18
3	DESENVOLVIMENTO	19
3.1	Visão geral	19
3.2	Servidor	20
3.2.1	Framework	20
3.2.2	Segurança	21
3.2.3	Banco de dados	21
3.2.4	Visão geral	22
3.2.5	Autenticação	23
3.2.6	Pesquisa e seus componentes	24
3.3	Site	25
3.3.1	Framework	26
3.3.2	Visão geral	27
3.3.3	Autenticação	27
3.3.4	Requisições HTTP	29
3.3.5	Pesquisa e seus componentes	29
3.4	Aplicativo	30
3.4.1	Sistema operacional	30
3.4.2	Linguagem	31
3.4.3	Visão geral	31
3.4.4	Componentes Auxiliares	31
3.4.5	Fluxo geral	32
3.4.6	Coleta de tempo de uso	35

4	TESTES E VALIDAÇÃO	51
4.1	Coleta do tempo de uso	51
4.2	Servidor	51
4.3	Resultados	52
4.3.1	Testes de precisão	52
4.3.2	Testes de carga e estresse	53
4.3.3	Requisitos	54
4.4	Conclusão	54
	REFERÊNCIAS	61
	APÊNDICE A – CÓDIGOS DA PLATAFORMA	65
A.0.1	Site	65
A.0.2	Aplicativo	65
A.0.3	Servidor	65
	APÊNDICE B – ARTIGO	67

1 Introdução

Este capítulo apresenta o tema de pesquisa do Trabalho de Conclusão de Curso, o escopo no qual o problema em questão será tratado e a justificativa do projeto. Na Seção 1.1 é apresentado o contexto e a motivação para a realização do trabalho. Na Seção 1.2 são introduzidos os objetivos gerais e específicos. Na Seção 1.3, são abordados os métodos de pesquisa envolvidos para alcançar a solução proposta. Uma visão geral do restante do trabalho é indicada na Seção 1.4.

1.1 Motivação

Sobrecarga de informação (*information overload*) é um conceito que vem sendo estudado há décadas e pode ser resumido como um recebimento excessivo de informação que ultrapassa o limite que um indivíduo é capaz de considerar em uma tomada de decisão (EPPLER; MENGIS, 2004; BENITO-RUIZ, 2009; BAWDEN; ROBINSON, 2009), podendo levar tanto a uma maior dificuldade de tomar uma decisão quanto a problemas de saúde (EPPLER; MENGIS, 2004).

O aumento de informações com as quais os indivíduos entram em contato diariamente é uma consequência direta da massiva expansão da Internet e da popularização dos *smartphones*. Informações que antes estavam disponíveis através de livros, enciclopédias, dicionários, jornais e afins passaram a ser encontradas em um único local e em maiores quantidades após a popularização de computadores com acesso à Internet, porém com limitações maiores quanto a mobilidade. Com a popularização dos *smartphones*, essas últimas restrições de mobilidade deixaram de existir e a facilidade de acesso foi imensamente potencializada, passando a estar disponível em apenas alguns toques em qualquer lugar.

Como se já não bastasse, os *smarthphones* aliados à internet e às redes sociais nos dão condições de interagir com uma imensidão de indivíduos ao longo do dia de forma instantânea, seja através de aplicativos de comunicação como WhatsApp, Telegram e Gmail, ou de redes sociais, como Facebook e Instagram. Os indivíduos que passam pelos processos de comunicação possuem limites, por mais que relativamente subjetivos, similares aos encontrados nos processos de informação, logo não é uma surpresa que o termo sobrecarga de comunicação (*communication overload*) exista nesse contexto (BENITO-RUIZ, 2009). Sobrecarga de comunicação pode ser definida como um estado onde as demandas por comunicação, através dos mais diversos meios (inclusive os citados anteriormente), excedem as capacidades comunicativas de um indivíduo (BENITO-RUIZ, 2009; LEE; SON; KIM, 2016; CHO et al., 2011; STEPHENS et al., 2017).

É importante salientar que o período de pandemia pelo qual passamos por causa do Coronavírus resultou no aumento da utilização dos celulares. É trivial imaginar que as medidas de isolamento social, não importa o quão brandas sejam, geram aumento da utilização dos *smarthphones* por diversas razões, seja para manter contato com amigos e familiares ou até mesmo para dar continuidade aos estudos.

Diversas pesquisas têm sido realizadas para avaliar os índices de utilização das redes sociais e sua relação com depressão, ansiedade, estresse (MOROMIZATO et al., 2017; BARRY et al., 2017; KELLY et al., 2019; SEABROOK; KERN; RICKARD, 2016; JELENCHICK; EICKHOFF; MORENO, 2013; BERRYMAN; NEGY, 2018) e bem-estar (VERDUYN et al., 2017). Algumas também foram feitas relacionando questões de saúde com utilização de *smarthphones* de forma geral (LEVINE; DVORAK; HALL, 2016; HAWI; SAMAHA, 2017; PANOVA et al., 2020; MONTEZUMA, 2020; NASCIMENTO et al., 2020; ALHASSAN et al., 2018; SOHN et al., 2019; SAMAHA; HAWI, 2016; GUO et al., 2020) e tempo de tela (*screen time*), que inclui *smarthphone*, televisores e *video games* (LIU; WU; YAO, 2015).

Apesar do elevado número de pesquisas nessa área, elas apresentam limitações, inclusive pontuados pelos autores e autoras. Dentre elas, baixo número de participantes e grupos muito homogêneos (HAWI; SAMAHA, 2017), dados reportados pelos entrevistados (baixa confiabilidade) (PANOVA et al., 2020; MONTEZUMA, 2020; ALHASSAN et al., 2018; SAMAHA; HAWI, 2016; HAUG et al., 2015; GUO et al., 2020; KIL et al., 2021) e não consideração de fatores econômicos-sociais (MONTEZUMA, 2020). Além disso, foram encontradas poucas pesquisas realizadas no Brasil (MOROMIZATO et al., 2017; MONTEZUMA, 2020; NASCIMENTO et al., 2020).

Este trabalho tem como objetivo desenvolver uma plataforma tecnológica (sistema web + aplicativo) com capacidade de medir com maior precisão a utilização dos *smarthphones* tanto de forma geral (tempo total de uso do *smarthphone*), quanto a utilização de aplicativos individualmente, para aumentar a confiabilidade de tais dados. A plataforma também terá capacidade de coletar dados manualmente, através de escalas cadastradas pelos próprios pesquisadores, para aumentar a abrangência de dados coletados e a capacidade de atender a múltiplas pesquisas de forma simultânea, para estimular um maior número de pesquisas na área.

1.2 Objetivos

Este trabalho possui como objetivo geral a disponibilização de uma plataforma tecnológica com suporte para a realização de pesquisas envolvendo tempo de uso de *smartphones* e escalas de saúde mental, de forma a facilitar o desenvolvimento de pesquisas na área, assim como aumentar a confiabilidade dos dados que puderem ser coletados automaticamente. Seguem os objetivos específicos:

1. Levantamento de requisitos com pesquisadores e profissionais de saúde mental;
2. Desenvolver o aplicativo para a plataforma Android que colete tempo de uso de demais aplicativos e o tempo de uso total do dispositivo, além de permitir a inserção de questionários para pesquisas;
3. Desenvolvimento de interface web para o cadastro de pesquisas, escalas e acesso aos dados coletados;
4. Estabelecer um banco de dados capaz de armazenar tais dados e de forma consistente e segura;
5. Avaliação da plataforma através de métricas de desempenho de software e precisão de coleta dos dados relevantes.

1.3 Método de Pesquisa

O trabalho será dividido em 4 etapas: estudo, análise de requisitos e projeto, desenvolvimento, testes e validação. A etapa de estudos contará com revisão da bibliografia, para ajudar na compreensão dos conceitos de sobrecarga de informação e de comunicação, vício em *smartphones* e como esses conceitos se relacionam entre si. Essa etapa também contará com um estudo do estado da arte em pesquisas que envolvam uso de *smartphones* e saúde mental. O estudo do estado da arte nos ajudará a compreender quais os últimos avanços nessa área de pesquisa.

A etapa seguinte, análise de requisitos e projeto, contará com a colaboração de pesquisadoras e pesquisadores da área da saúde mental para a realização da análise de requisitos, de forma a garantir que a plataforma a ser desenvolvida apresente as funcionalidades necessárias para sua utilização por profissionais. Ainda nessa etapa realizaremos o projeto do software contando com diagramas de classe e arquitetura de comunicação.

A terceira etapa, desenvolvimento, é onde produziremos a plataforma. O código será desenvolvido para o sistema Android, com uma plataforma web para apoio aos pesquisadores. Utilizaremos Android Studio e a linguagem Java. Decisões relacionadas ao banco de dados, integração da plataforma e servidor serão feitas posteriormente.

A etapa seguinte, testes e validação, é onde garantiremos a capacidade de performance da plataforma, bem como de precisão dos dados coletados. Realizaremos testes de carga e estresse para avaliarmos o comportamento da plataforma em condições normais e extremas.

1.4 Organização do Trabalho

O restante do trabalho será dividido em capítulos. O Capítulo 2 apresenta conceitos pertinentes ao trabalho, além de incluir uma revisão do estado da arte em pesquisas de saúde mental e smartphones. No Capítulo 3 apresentamos a plataforma e seu funcionamento geral, assim como alguns detalhes de implementação. O Capítulo 4 descreve os testes feitos para a validação da plataforma. O Capítulo 4.3, apresenta os resultados. O Capítulo 4.4 contém a conclusão e considerações finais do trabalho. No final do trabalho estão os apêndices com os códigos realizados para esse de trabalho e o artigo. O Apêndice A.0.1 contém os códigos do site. No Apêndice A.0.2 está os códigos do aplicativo e no Apêndice A.0.3 os códigos do servidor. No Apêndice B podemos encontrar o Artigo referente à este trabalho.

2 Fundamentação Teórica

Neste capítulo apresentaremos conceitos teóricos relacionados à importância do trabalho. Também faremos uma breve resvisão do estado da arte, Na Seção 2.1 apresentaremos os conceitos importantes para o trabalho. A Seção 2.2 apresenta o estado da arte de pesquisas relacionadas à saúde mental e o uso de *smartphones*. Por fim, a Seção 2.3 contém uma breve conclusão baseada na Seção 2.2.

2.1 Sobrecarga de informação e comunicação

O conceito de sobrecarga de informação (*information overload*) é bastante similar ao de sobrecarga de comunicação (*communication overload*). Ambos são processos que ocorrem quando um indivíduo estapola seus limites, seja de informação, seja de comunicação, e que apresenta consequências tanto relacionadas aos processos, quanto para a saúde do indivíduo. Apesar de apresentarem características similares, daremos início analisando suas particularidades.

Sobrecarga de informação (SI) pode ser descrita como um processo que ocorre quando um indivíduo excede seus "limites de informação", ou seja, quando a quantidade de informação sobre determinado assunto que o indivíduo precisa levar em consideração para tomar uma decisão dificulta a tomada de decisão em função da quantidade de informação que precisa ser filtrada e processada pelo indivíduo (BAWDEN; ROBINSON, 2009; SCHICK; GORDON; HAKA, 1990; EPPLER; MENGIS, 2004). Por mais que as definições mais clássicas de SI não estejam relacionadas com uso de aparelhos celulares, redes sociais e afins, alguns trabalhos atuais relacionam os assuntos (KOROLEVA; KRASNOVA; GÜNTHER, 2010; LEE; SON; KIM, 2016; MATTHES et al., 2020). Tanto as definições mais clássicas quanto os trabalhos mais modernos afirmam que as consequências da SI não se limitam à dificuldade de tomar uma decisão, podendo resultar em problemas relacionados à saúde mental e bem-estar do indivíduo.

Já a sobrecarga de comunicação (SC), por mais que apresente uma definição relativamente similar, não possui consequências diretas cognitivas (aumento da dificuldade na tomada de decisão, por exemplo). SC é um processo relacionado tanto ao número de canais de comunicação aos quais um indivíduo tem que atender quanto à qualidade e complexidade das informações recebidas em processos de comunicação. A principal diferença entre SI e SC parte da compreensão de que "informação" trata de conhecimento, de mensagens que auxiliam ou interferem num processo de tomada de decisão, enquanto comunicação se refere à mensagens simples, cotidianas, que não necessariamente impactam num processo de tomada de decisão (CHO et al., 2011).

2.2 Revisão do estado da arte em pesquisas de saúde mental e *smarphones*

Para compreender melhor os processos de sobrecarga e suas consequências para a saúde mental dos indivíduos, além da relação de tais sobrecargas com o uso de aparelhos celulares uma revisão da literatura foi feita, focada especificamente em pesquisas envolvendo tais assuntos. Tal pesquisa se tornou uma revisão do estado da arte a partir do momento que passou a ser utilizada para melhor compreender as tecnologias utilizadas nas pesquisas e avaliações de tais relações.

2.2.1 Protocolo de pesquisa

Para garantir a adequação dos trabalhos encontrados foram aplicados filtros sobre eles. Os critérios de inclusão foram: 1) ano de publicação, a partir de 2011 e 2) A presença de combinações das palavras chave: *information overload*, *smartphone usage*, *mental health*, além da presença de palavras como *self report usage*. O critério de exclusão para trabalhos cotendo o primeiro conjunto de palavras chafe foi 1) a não adequação do trabalho escolhido ao escopo do nosso, ou seja trabalhos que não envolvessem pesquisa com questionários pré-definidos e com levantamento dos dados feitos diretamente com os participantes.

2.2.2 Resultados

As primeiras pesquisas feitas relacionaram diretamente SI e saúde mental. Poucos trabalhos foram encontrados com foco nas pesquisas, por mais que os trabalhos confirmem que que parte das consequências da SI estão relacionadas à saúde mental dos indivíduos. (BAWDEN; ROBINSON, 2009; EPPLER; MENGIS, 2004)

As pesquisas seguintes foram sobre a relação de *smartphones* com saúde mental. Para tal, utilizamos a ferramenta do *Google Scholar* buscando pelas palavras chave "*smartphones usage*" e "uso de celulares" e "*mental health*" e "saúde mental". Os critérios de inclusão e de exclusão foram descritos na seção seguinte 2.2.1. O número de pesquisas encontradas foi bastante limitado, especialmente depois de aplicados o filtros.

Para definirmos o tamanho dos grupos que utilizaremos nas divisão a seguir faremos alguns cálculos de tamanho de amostra. Para que possamos definir uma população total para todos esses estudos fizemos alguns testes levando em consideração a população dos países onde as pesquisas foram realizadas, como Brasil, Estados Unidos e Reino Unido e concluimos que para populações tão grandes os valores do intervalo de confiança se mantiveram os mesmos entre 0.5% da menor população, Reino Unido, até 100% da maior população, Estados Unidos. Para valores fixos de níveis de confiança em 95% e população total variando conforme descrito acima, encontramos os seguintes tamanhos de amostra baseado em intervalos de confiança: 196

	Automática	Manual
grupos pequenos	(LIN et al., 2015)	(MONTEZUMA, 2020)
grupos médios	-	(HAWI; SAMAHA, 2017; SAMAHA; HAWI, 2016; KIL et al., 2021)
grupos grandes	-	(PANOVA et al., 2020; ALHASSAN et al., 2018; GUO et al., 2020)

Tabela 1 – Tabela classificando o primeiro conjunto de pesquisas baseado no número de participantes e o tipo de coleta de dados

para um intervalo de 7, 202 até 600 para intervalos entre 7 e 4 e acima de 600 para intervalos de confiança menores que 4.

Dos resultados encontrados, podemos separar as pesquisas em 2 categorias baseadas no tipo de coleta: i) fizeram uso de coleta automática de dados; ii) não fizeram uso de coleta automática de dados. Podemos também separar em 3 categorias baseadas no número de participantes descritos acima: a) grupos pequenos, menos de 200 participantes; b) grupos médios, entre 201 e 600 participantes e c) grupos grandes, acima de 601 participantes. Podemos ver os trabalhos separados em tais grupos na Tabela 1.

Em seguida nossa pesquisa focou na relação entre uso de redes sociais e saúde mental. A metodologia de pesquisa foi similar a anterior, apenas alterando as palavras de "smartphone usage" e "uso de celular" para "social media" e "redes sociais" respectivamente. Também aplicamos os mesmos filtros descritos na seção seguinte 2.2.1. Os trabalhos encontrados podem ser divididos nas mesmas categorias que os trabalhos anteriores e podemos ver os trabalhos separados em tais grupos na Tabela 2.

O recorrente uso de coleta manual com foco no tempo de uso percebido, seja tempo de uso do smartphone, seja de alguma rede social específica, levanta a questão da diferença do tempo de uso real e do tempo de uso percebido pelo usuário. Poucas pesquisas que abordam esse tema foram encontradas. Enquanto (LEE et al., 2017; OHME et al., 2021) concluem que o tempo de uso percebido é menor que o tempo de uso real (*under-report*), (LEE et al., 2021) conclui que o tempo de uso percebido é maior que o tempo de uso real (*over-report*). Tais pesquisas corroboram com a idéia que a percepção do usuário é bastante distorcida.

	Automática	Manual
grupos pequenos	-	(O'REILLY et al., 2018; BARRY et al., 2017; JELENCHICK; EICKHOFF; MORENO, 2013)
grupos médios	-	(BERRYMAN; NEGY, 2018)
grupos grandes	-	(KELLY et al., 2019)

Tabela 2 – Tabela classificando o segundo conjunto de pesquisas baseado no número de participantes e o tipo de coleta de dados

2.3 Conclusão

O número de pesquisas que faz uso do tempo de uso, seja do smartphone, seja de alguma rede social, é bastante elevado, como pudemos ver. Isso nos leva a entender que o tempo de uso é uma variável importante para pesquisas que relacionam o uso de smartphones e redes sociais com questões de saúde mental. Se aliarmos nossa compreensão de que tais valores são importantes, com as pesquisas que demonstram que o tempo de uso percebido pelo usuário é diferente do tempo de uso real, podemos questionar a validade de algumas conclusões de pesquisas que fazem uso da coleta manual de tais dados.

Os celulares atualmente já fazem uso de sensores para coletar diversos dados que podem ser relevantes para pesquisas, isso inclui o tempo de uso geral (do smartphone como um todo) e o tempo de uso específico (individual, de cada aplicativo). Se esses dados já são coletados automaticamente, por que não fazer uso deles para aumentar a confiabilidade dos dados utilizados em pesquisas?

Com isso, podemos entender a pertinência de uma plataforma especializada para a realização das pesquisas, podendo tanto aumentar a confiabilidade de alguns dados, através da coleta automatizada, como aumentar a abrangência das pesquisas, dada a facilidade de divulgação e disseminação das pesquisas através de um aplicativo.

3 Desenvolvimento

A plataforma proposta foi feita com base em reuniões de análise de requisitos com a participação de profissionais da área clínica de psicologia e psiquiatria. Dessa análise foram retirados requisitos fundamentais como: coleta do tempo de uso dos demais aplicativos de forma automatizada, sem nenhuma interação com o usuário, plataforma para interação dos pesquisadores com o sistema de forma visual e filtro de pesquisas baseada em parâmetros definidos pelos responsáveis pelas pesquisas. O documento de requisitos completo pode ser visto na Figura 1.

Introdução		Este documento especifica os requisitos da "Plataforma Tecnológica para pesquisas sobre saúde mental e tempo de uso de smartphones"		
Requisitos funcionais	Descrição	Prioridade		
		Essencial	Importante	Desejável
RF1. Coleta automática do tempo de uso	O sistema deve garantir que a coleta do tempo de uso seja feita e que não necessite de interação com o usuário	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RF2. Coleta realizada na mesma frequência para todos	O sistema deve garantir que o tempo de uso coletado acontece sempre no mesmo horário para todos os usuários	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RF3. Filtro de pesquisas	O sistema deve filtrar automaticamente pesquisas baseada em parâmetros definidos pelos pesquisadores	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
RF4. Plataforma visual para interação dos pesquisadores	O sistema deve possuir alguma plataforma que permita a interação com os pesquisadores que não seja através de um aplicativo de celular	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Requisitor não funcionais		Prioridade		
		Essencial	Importante	Desejável
RNF1. Autenticação	O sistema deve garantir a autenticação dos usuários, seja pesquisadores ou participantes	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RNF2. Portabilidade para pesquisadores	O acesso dos pesquisadores deve ser possível nas mais diversas plataformas	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 1 – Visão geral da plataforma

Neste capítulo iremos descrever em diferentes níveis o funcionamento da plataforma. Na Seção 3.1 é apresentado em linhas gerais tanto a utilização do site e do aplicativo quanto a comunicação superficial deles com o servidor. Os 3 componentes da plataforma, servidor, site e aplicativo, serão abordados nas suas respectivas Seções 3.2 3.3, 3.4, onde veremos mais a fundo o funcionamento de cada um deles.

3.1 Visão geral

A plataforma proposta é dividida em três componentes: Site, Aplicativo e Servidor. O Site é por onde os pesquisadores irão interagir com a plataforma, criando, atualizando e acompanhando suas pesquisas. O Aplicativo é por onde os participantes irão interagir com a plataforma, buscando pesquisas para participar e respondendo questionários das pesquisas. O aplicativo também é responsável por coletar o tempo de uso dos demais aplicativos, informação fundamental para o nosso trabalho. Por fim o servidor é responsável por prover as informações tanto para o site quanto para o aplicativo, armazenando, atualizando e deletando essas informações quando necessário. Podemos ver uma versão simplificada dessas relações entre os componentes na Figura 2.

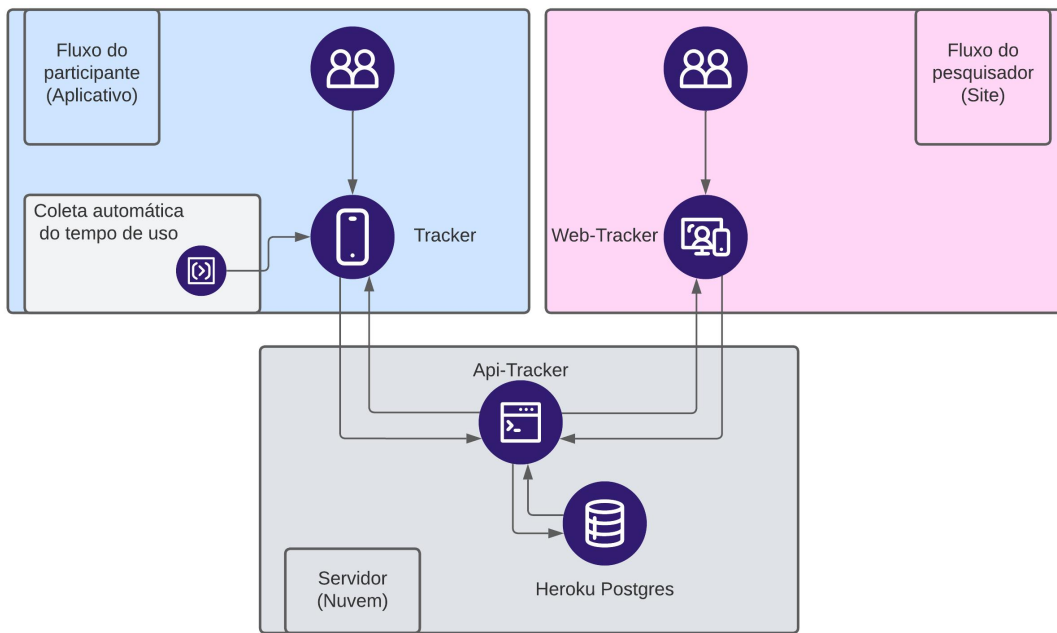


Figura 2 – Visão geral da plataforma

3.2 Servidor

Primeiro temos o servidor, que é responsável por armazenar as informações dos usuários tanto do site quanto do aplicativo, as pesquisas e seus componentes, as respostas de tais pesquisas e o tempo de uso coletado e responder as requisições dos demais componentes com as informações requisitadas.

3.2.1 Framework

A escolha de framework para o desenvolvimento do servidor foi feita entre Flask e FastAPI. Apesar de ambos serem frameworks minimalistas para Python, Flask, por ser mais antigo já foi amplamente testado e validado, possui uma grande comunidade com diversos materiais online e é facilmente escalável.

As vantagens do FastAPI são, além de ser ainda mais simples que Flask, possui documentação automatizada e validação de dados embutidas. Apesar dos materiais não serem tão extensos e numerosos quanto Flask, a documentação de ambos é sólida e clara, facilitando a utilização inicial de ambas e em função disso escolheremos FastAPI ¹.

¹ <https://fastapi.tiangolo.com/tutorial/>

3.2.2 Segurança

A segurança será controlada pelo servidor através do protocolo OAuth ², apesar de sabermos que OAuth é um protocolo de autorização e não de autenticação. Isso quer dizer que o acesso aos recursos do servidor será dada baseada num token gerado pelo servidor com as credenciais do usuário. O token será enviado com as requisições, tanto do site quando do aplicativo, no cabeçalho *Authorization* e será verificado no servidor para toda a requisição feita. Se o token for válido, então o acesso é permitido.

Teoricamente deveríamos, para seguir o padrão do OAuth, gerar um token de acesso, *access_token* e um de atualização *refresh_token*. O de acesso é o que deve ser enviado nas requisições, porém possui um tempo de expiração. Quando o token de acesso expira, é utilizado o token de atualização para gerar novos token de acesso e de atualização. Nossa plataforma não segue todos os padrões do OAuth. Nosso token de acesso é gerado com um tempo de expiração de anos e sem nenhum token de atualização.

É importante salientarmos que a única informação do usuário que é criptografada antes de ser salva é a senha, ou seja nunca utilizamos a senha real do usuário para nada. Sempre que uma senha é recebida passamos ela pela função de hash *bcrypt* seja para salvar ou para comparar com a salva. A função *bcrypt* é especialmente feita para o hash de senhas, apesar de ser mais custosa tanto em questão de tempo quanto de espaço que função como *SHA-256*.

3.2.3 Banco de dados

Os dados que serão utilizados na nossa plataforma podem ser encontrados nas Figuras 3 e 4.

As entidades que encontramos no nosso banco são: Pesquisador, Participante, Pesquisa, Questionário, Questão, Alternativa, Resposta e Tempo de Uso. Cada uma dessas entidades é representadas por uma tabela no banco. Além disso existe uma única tabela auxiliar para relacionar Participantes com Pesquisas. Podemos visualizar tais relações e os dados pertencentes à cada tabela na figura 5.

Ao longo das diferentes etapas do desenvolvimento utilizamos diferentes SGBDs. Num primeiro momento, enquanto desenvolvíamos apenas localmente, fizemos uso do SQLite ³ por ser um dos SGBDs mais simples e rápidos, ótimo para um sistema relativamente pequeno como o nosso.

Quando passamos pra o desenvolvimento e o *deploy* através da plataforma Heroku ⁴ precisamos mudar nossa abordagem. O Heroku é uma plataforma como serviço (PaaS - *Platform as a Service*), o que permite que nosso desenvolvimento não precisasse se preocupar com toda

² <https://oauth.net/>

³ <https://www.sqlite.org/index.html>

⁴ <https://www.heroku.com/platform>

Variável	Nome	Valores possíveis	Descrição
Número ID Pesquisador	ID	1-999	Valor gerado para o identificador do pesquisador de forma sequencial
Nome Pesquisador	Nome	Sequências de caracteres	Nome definido pelo pesquisador na criação de conta
Email Pesquisador	Email	Sequências de caracteres	Email definido pelo pesquisador na criação de conta
Senha Pesquisador	Senha	Sequências de caracteres	Hash da senha definido pelo sistema a partir da senha definida pelo pesquisador
Número ID Pesquisa	ID	1-999	Valor gerado para o identificador da pesquisa de forma sequencial
Data Início Pesquisa	Data_inicio	AAAA-mm-dd hh:mm:ss.mmmm	Valor gerado pelo sistema do início de uma pesquisa
Data Fim Pesquisa	Data_fim	AAAA-mm-dd hh:mm:ss.mmmm	Valor gerado pelo sistema do fim de uma pesquisa
Descrição Pesquisa	Descrição	Sequências de caracteres	Descrição definida pelo pesquisador para a pesquisa
Visibilidade Pesquisa	Visibilidade	Sequências de caracteres	Definição se uma pesquisa é pública ou privada
Idade Início Pesquisa Filtro	Idade_inicio	1-999	Filtro para definir a idade mínima para um participante da pesquisa
Idade Fim Pesquisa Filtro	Idade_fim	1-999	Filtro para definir a idade máxima para um participante da pesquisa
Salário Início Pesquisa Filtro	Salario_inicio	1.0-999.0	Filtro para definir o salário mínimo para um participante da pesquisa
Salário Fim Pesquisa Filtro	Salario_fim	1.0-999.0	Filtro para definir o salário máximo para um participante da pesquisa
Raça Pesquisa Filtro	Raça	Sequências de caracteres	Filtro para definir a raça para um participante da pesquisa
Gênero Pesquisa Filtro	Genero	Sequências de caracteres	Filtro para definir o gênero para um participante da pesquisa
Orientação Sexual Pesquisa Filtro	Orientação_sexual	Sequências de caracteres	Filtro para definir a orientação sexual para um participante da pesquisa
Código Pesquisa	Codigo	Sequências de caracteres	Código gerado pelo sistema caso a pesquisa possua visibilidade privada
Módulo Pesquisa	Modulos	Sequências de caracteres	Módulos definidos pelo pesquisador que devem ser ativos para a pesquisa (apenas tempo de uso no momento)
Número ID Participante	ID	1-999	Valor gerado para o identificador do participante de forma sequencial
Nome Participante	Nome	Sequências de caracteres	Nome escolhido pelo participante na criação de conta
Data Nascimento Participante	Data_nascimento	AAAA-mm-dd	Data de nascimento definida pelo participante

Figura 3 – Parte 1 do dicionário de dados da plataforma

a parte de arquitetura para garantir a disponibilização online do nosso servidor. A plataforma do Heroku, por questões de armazenamento e economia, constantemente apaga os arquivos localmente e puxa eles do repositório do GitHub relacionado ao projeto. Se utilizarmos SQLite, que é baseado na geração e manutenção de arquivos localmente, tais arquivos serão apagados e puxados do GitHub diariamente, apagando todas as informações salvas.

Para solucionar isso a plataforma disponibiliza um serviço baseado no SGBD Postgres chamado de Heroku Postgres ⁵. Por ser a única alternativa de banco relacional que possuímos na plataforma, não tiver outra escolha que não fazer uso dela.

3.2.4 Visão geral

Parte crucial do funcionamento do servidor é a injeção de dependências e a definição de esquemas para a entrada de dados no servidor. A injeção de dependências com FastAPI é bastante simples se compararmos com frameworks mais complexos, bastando adicionar `Depends` à um parametro na rota de operação.

O FastAPI trata de executar a função passada para o `Depends` e inicializar a variável com o retorno da função passada. Uma dependência que todos os endpoints possuem é a de abrir

⁵ <https://devcenter.heroku.com/categories/heroku-postgres>

Gênero Participante	Genero	Sequências de caracteres	Gênero escolhido pelo participante
Raça Participante	Raça	Sequências de caracteres	Raça escolhida pelo participante
Salário Participante	Salario	1.0-999.0	Salário definido pelo participante
Orientação Sexual Participante	Orientação_sexual	Sequências de caracteres	Orientação sexual definida pelo participante
Usuário Participante	Nome	Sequências de caracteres	Usuário usado para criação de conta do participante
Senha Participante	Senha	Sequências de caracteres	Hash da senha definido pelo sistema a partir da senha definida pelo participante
Número ID Questionário	ID	1-999	Valor gerado para o identificador do questionário de forma sequencial
Público Questionário	Publico	0-1	Valor que define se um questionário é público ou privado
Título Questionário	Titulo	Sequências de caracteres	Título do questionário criado pelo pesquisador
Número ID Questão	ID	1-999	Valor gerado para o identificador da questão de forma sequencial
Pergunta Questão	Pergunta	Sequências de caracteres	Texto da pergunta definido pelo pesquisador
Ordem Questão	Ordem	1-999	Ordem na qual a pergunta deve aparecer para o participante
Tipo Questão	Tipo	Sequências de caracteres = "multipla_escolha" ou "descritiva"	Tipo de pergunta definida pelo pesquisador. Pode ser tanto de múltipla escolha, que permite a criação de alternativas ou descritiva que não permite
Número ID Alternativa	ID	1-999	Valor gerado para o identificador da alternativa de forma sequencial
Texto Alternativa	Texto	Sequências de caracteres	Texto da alternativa definido pelo pesquisador
Valor Alternativa	Valor	1-999	Peso dado para cada alternativa pelo pesquisador
Número ID Resposta	ID	1-999	Valor gerado para o identificador da resposta de forma sequencial
Texto Resposta	Texto	Sequências de caracteres	Texto da alternativa escolhida pelo participante no caso de uma resposta à uma pergunta de múltipla escolha ou texto inserido pelo participante no caso de uma pergunta descritiva
Valor Resposta	Valor	1-999	Peso da alternativa escolhida pelo usuário no caso de uma pergunta de múltipla escolha ou 0 no caso de uma pergunta descritiva
Número ID Tempo de Uso	ID	1-999	Valor gerado para o identificador do tempo de uso de forma sequencial
Tempo de Uso Coletado	Tempo_coletado	Sequências de caracteres	Texto contendo o tempo de uso enviado pelo aplicativo no formato "nome_do_aplicativo": "uso_em_milisegundos"

Figura 4 – Parte 2 do dicionário de dados da plataforma

uma sessão do banco e encerrá-la antes de retornar a resposta apropriada. Isso é feito através da função `get_db`.

O recebimento de dados também é simples, bastando definir um modelo *pydantic* e definir o tipo da variável que será recebida no corpo da requisição, como o modelo definido anteriormente. O FastAPI trata de ler o corpo da requisição como JSON, converter os valores para os tipos correspondentes, validar os dados e inicializar a variável com o valor convertido e validado.

Assim como os demais componentes, o servidor apresenta bastante similaridade entre algumas de suas funcionalidades e por isso iremos apresentar o fluxo geral e detalhar as especificidades conforme necessário. As requisições podem ser divididas em grupos: Autenticação, Pesquisa, Questionário, Questão, Alternativa e Resposta.

3.2.5 Autenticação

Os endpoints de Autenticação são usados tanto para criar uma conta quanto para acessar uma conta criada. No caso da criação de conta é checado se já existe um usuário com essas informações e em seguida se as informações são válidas. Caso já exista usuário com as informações ou as informações sejam inválidas, um erro é retornado, caso contrário as informações são salvas com a senha encriptada e um token JWT é gerado para o usuário e retornado. O login

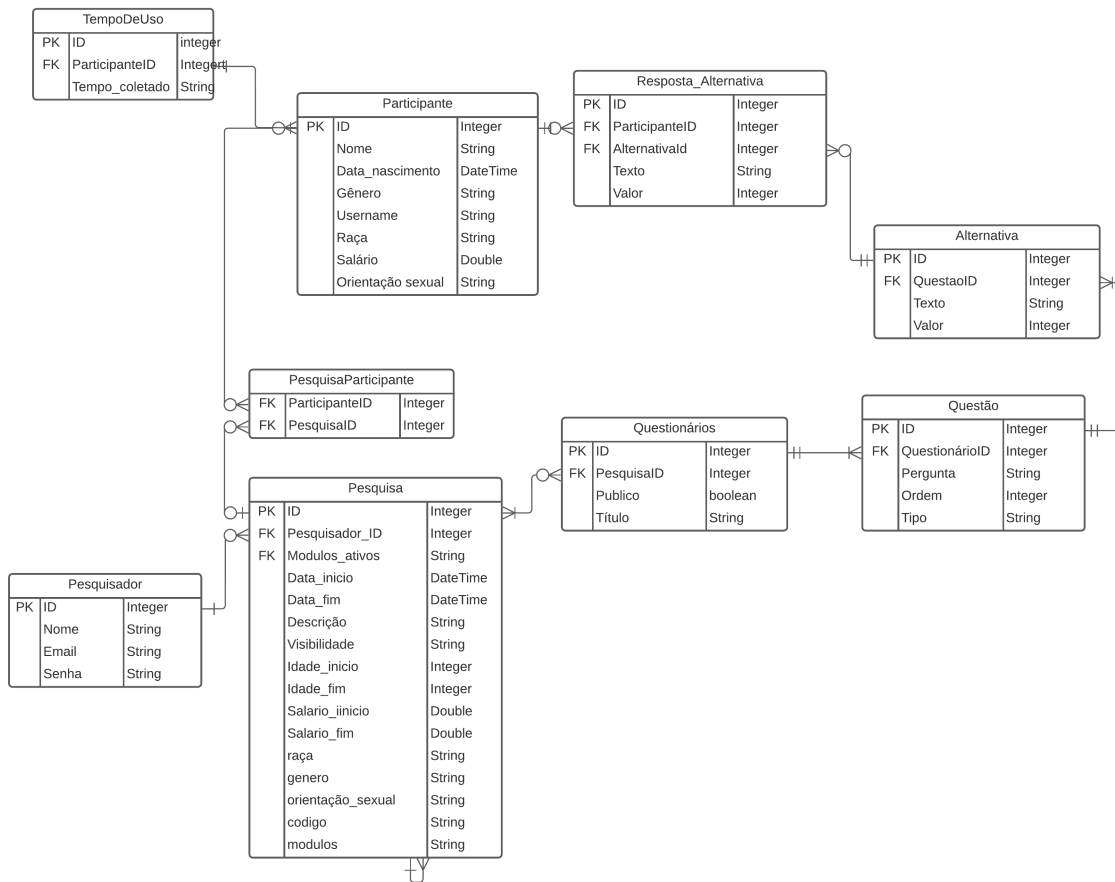


Figura 5 – Diagrama do banco de dados

checa a existência do usuário e se a senha recebida condiz com a encriptada salva. Um token JWT é retornado em caso de sucesso.

Todos os endpoints além dos de autenticação checam o cabeçalho *Authorization* para pegar o token JWT que deve ser passado e através do token pegar o usuário logado. Tal processo é feito através da injeção de dependência que recebe como função `get_current_user`

3.2.6 Pesquisa e seus componentes

A Pesquisa e seus componentes possuem alguns endpoints bastante similares: `pegarTodas`, `adicionar`, `editar` e `deletar`. Pegar todas é uma requisição GET que recebe apenas os ids das suas entidades agregadoras, se existirem, pelo caminho seguindo o seguintes modelos de exemplo: `</pesquisas/{id}/questionarios/{id}/questoes>` para pegar questões e `</pesquisas>` para pegar pesquisas. Como podemos ver, pesquisa não recebe nenhum id por não ser agregada por nenhum outra componente, enquanto questões recebe o id de questionário, que é o agregador de questão, e o id de pesquisa, que é o agregador de questionário. Todos esses ids são usados em seguida para pegar no banco as informações e retornar ao usuário.

Adicionar, que é uma requisição POST, segue o mesmo padrão de caminho porém re-

cebe no corpo os dados da entidade a ser adicionada no banco. Os dados recebidos são convertidos para o modelo que representa a entidade no banco e então são salvos. Por fim é retornado ao cliente os dados recém adicionados, apesar do mais comum para POST ser o retorno do código 201 com uma referência para o recurso criado.

Atualizar e deletar, que são respectivamente PUT e DELETE recebem no final do caminho o id da entidade de interesse: `<.../questionarios/{questionnaire_id}/questoes/id>`. Ambos os endpoints verificam a existência da entidade no banco antes de procederem com suas operações e retornam com o código 204 para o cliente no caso de sucesso. Enquanto o atualizar substitui no banco as informações antigas pelas recebidas o de deleção apenas remove do banco a entidade.

No caso dos endpoints de Pesquisa, existe uma validação adicional que verifica o estado da pesquisa para permitir a atualização. Existem também dois endpoints PATCH adicionais, um que adiciona um participante à pesquisa e outro que checa o estado atual da pesquisa e avança para o estado seguinte. Se for inativa é ativada, se for ativa é encerrada.

No endpoint de atualizar do Questionário, é verificado se a entidade em questão é tipo *template* e se o usuário está tentando mudar o tipo dela. Se esse for o caso a atualização é bloqueada. Ainda se tratando de questionários, existe a possibilidade do usuário adicionar múltiplos questionários do tipo *template* de uma única vez. Os endpoints responsáveis por isso são dois: o primeiro pega todos os questionários públicos. O segundo recebe uma lista de ids que é usada para pegar todas as informações necessárias para criar cópias desses questionários, já populados com as mesmas questões e alternativas, para a pesquisa solicitada.

Existem endpoints específicos para participantes. Tais endpoints possuem o comportamento similar aos já explicados para os usuário, porém são suficientemente únicos para explicarmos separadamente. O pegarTodas para pesquisas aplica os filtros da pesquisa para retornar apenas as pesquisas às quais o participante pode participar, enquanto o pegarTodas para questionários filtra por participantes e por pesquisa. Também existe um endpoint único para pegar as questões de um questionário incluindo todas as alternativas de cada questões. Existe dois POSTs, um com a função de adicionar respostas relacionadas as perguntas e o segundo para adicionar o tempo de uso coletado. O endpoint de PATCH para o participantes cumpre a função de atualizar apenas os filtros do participante.

3.3 Site

O site, apesar de secundário perante os demais componentes, tem como propósito facilitar que pesquisadores criem, modifiquem, excluam e acompanhem pesquisas. Apesar do diagrama de uso do site, como podemos ver na Figura 6, conter diversos casos, o uso do site é bastante linear.

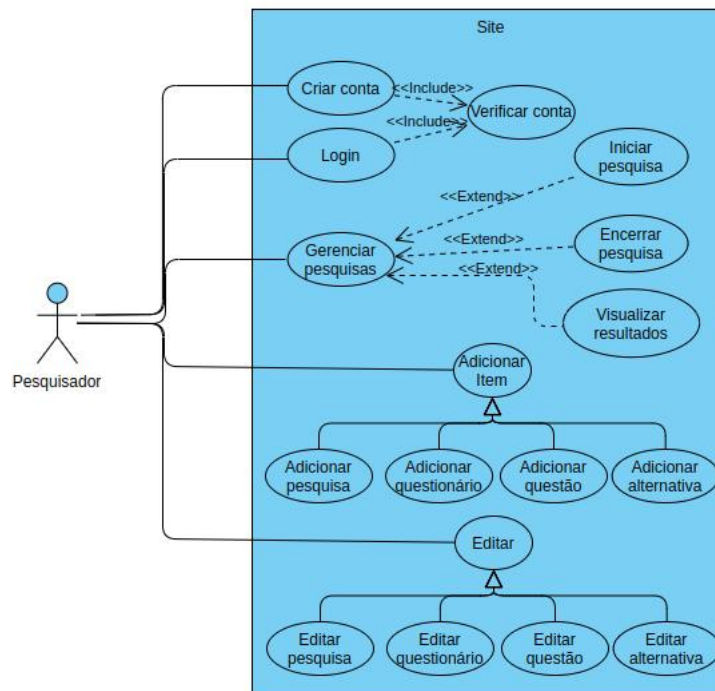


Figura 6 – Diagrama de uso do site

3.3.1 Framework

A escolha de framework para o site foi feita entre alguns dos frameworks mais populares: React, Vue e Angular, Django e Spring. Apesar de React não ser propriamente um framework, no nosso contexto ele equivale a um. Angular, Django e Spring foram prontamente descartados dada a complexidade e curva de aprendizado íngreme, enquanto Vue, apesar de ser uma das possíveis melhores escolhas dado o crescimento recente e a popularização rápida, ainda apresentava poucos materiais introdutórios que pudessem garantir um aprendizado rápido para solucionar os possíveis problemas que pudessemos encontrar ao longo do desenvolvimento.

React ⁶ se mostrou a melhor escolha para nosso problema. Apesar de não possuir uma curva de aprendizado tão macia quanto Vue, a curva não é tão íngreme quanto dos outros frameworks. Além disso, os materiais e componentes disponíveis é relativamente maior.

Como dito anteriormente, React não é propriamente um framework, sendo considerada uma biblioteca de componentes para JS o que garante uma maior flexibilidade por não definir nenhum de padrão de projeto. Componentes são uma forma de reutilizar pedaços de código da interface gráfica. React também faz uso de uma extensão de sintaxe do JavaScript chamada JSX que permite escrever funções que retornam HTML puro em JavaScript.

⁶ <https://reactjs.org/docs/getting-started.html>

3.3.2 Visão geral

O site começa verificando se o usuário já logou antes, se sim, utiliza as informações do último login, se não, carrega a página de login. Nesse primeiro momento o usuário pode escolher entre criar uma conta, que carrega uma nova página, ou acessar uma conta criada. Em ambos os casos uma mensagem será enviada para o servidor contendo as informações a serem validadas e caso a validação seja bem-sucedida, salva as informações para um login futuro.

Uma vez logado o site requisita as informações das pesquisas para carregar a primeira página. Nessa página o usuário pode tanto adicionar novas pesquisas, editar e deletar pesquisas existentes, iniciar e finalizar pesquisas, baixar resultados, navegar para os questionários das pesquisas e deslogar. Cada uma dessas ações é acompanhada de uma requisição ao servidor com suas respectivas propriedades.

A partir desse ponto, todas as seguintes páginas são responsáveis por representarem suas respectivas entidades: Questionário, Questão e Alternativa e todas seguem o mesmo padrão de exibição e interação que a página de Pesquisa, com exceção do Questionário que, ao adicionar um novo questionário verifica se o usuário quer buscar por questionários públicos existentes ou criar um do zero. Caso seja escolhido criar um do zero o fluxo é igual ao das outras páginas, porém caso decida buscar por existentes uma janela é aberta para que o usuário possa navegar pelos questionários e suas respectivas questões e alternativas e escolher quais quer adicionar à sua pesquisa.

3.3.3 Autenticação

As primeiras páginas que o usuário tem acesso são as de login e a de criação de conta, que podem ser vistas na Figura 7 e na Figura 8 respectivamente. Ambas foram algumas das primeiras páginas desenvolvidas e seguem um padrão para a atualização dos dados diferente das demais. As páginas não passam de formulários simples que atualizam seus campos em tempo real fazendo uso do *hook* `useRef`. Além disso, ambas as páginas utilizam funções do contexto `AuthContext` que possui um `AuthProvider` contendo funções para login, criação de conta, logout e também é responsável por armazenar a resposta das requisições de autenticação do servidor no armazenamento local do navegador em caso de sucesso.

O `AuthProvider` recebe como filho todas as rotas do nosso site, garantindo que a função de logout seja acessível por todas as páginas e a possibilidade de definir páginas privadas, que só são possíveis serem acessadas caso o usuário tenha concluído a autenticação. O que define se um usuário está autenticado é a presença da constante `token` do `AuthProvider`.

A função de criação de conta é apenas uma requisição POST que é feita para o servidor, onde é passado no corpo as informações preenchidas pelo usuário no formato JSON. Já a função de login segue o padrão OAuth 2.0 Grant Type que, apesar de não ser mais recomendado, é uma forma simples de fazer a verificação do usuário. A função também é um POST, porém o formato



Login do sistema

Email

Senha

Entrar

Criar conta

Figura 7 – Tela de login do site



Criação de conta

Nome

Email

Senha

Criar

Login

Figura 8 – Tela de criação de conta do site

das informações é *grant_type* e não JSON.

A última responsabilidade do `AuthProvider` é verificar a existência de um token JWT no armazenamento local do navegador. Caso exista, esse token é passado para a constante `token`, garantindo acesso ao resto do site.

3.3.4 Requisições HTTP

Além das requisições citadas na Subseção 3.3.3 o site possui mais 3 funções responsáveis pelo envio de requisições ao servidor: `httpFetch`, `httpFetchWithBody` e `httpFetchDownloader`. A primeira é responsável pelas requisições GET, recebendo o *endpoint* para qual enviará a requisição, bem como o token que será colocado no cabeçalho da requisição para a autenticação no servidor. Já a `httpFetchWithBody` é responsável pelas requisições de POST, PUT, PATCH e DELETE. Recebendo tanto o método HTTP quanto o endpoint, o corpo e o cabeçalho. No caso de requisições GET, POST e PATCH os métodos responsáveis aguardam o retorno para que o mesmo possa ser lido como um JSON e retornado para a função anterior. Por último temos o `httpFetchDownloader` que faz uma requisição GET recebendo o id da pesquisa, o token passado no cabeçalho para a autenticação e o nome do arquivo a ser baixado. Após a requisição, é criado um blob e a partir dele se cria um objetoURL com as informações do arquivo e um botão fictício que então é "clicado" para que o arquivo possa ser baixado.

3.3.5 Pesquisa e seus componentes

Tanto a pesquisa quanto os questionário, questões e alternativas seguem um mesmo padrão, que iremos explicar de forma genérica para todos e depois com mais detalhes nas diferenças de cada um.

O modelo base que seguimos é dividido em 3 partes: Página, Lista e Ítem. A página é responsável por renderizar as informações pertinentes, é quem se comunica com o resto das funções do site, como logout e as requisições, é quem armazena as informações respondidas pelo servidor para serem passadas para a Lista e é responsável pela exibição dos *breadcrumbs*, que facilitam a navegação no site. Através do *hook* `useEffect`, que é chamado apenas na primeira vez que a página é renderizada, é feita uma requisição com `httpFetch` onde são pegos os dados que são passados para a Lista e em seguida para o Ítem. A página contém funções `handleSubmit`, `handleEdit` e `handleDelete` que fazem uso do `httpFetchWithBody` para alterar os dados na página.

As principais funcionalidades da Lista são tanto definir os formulários para a entrada de dados por parte do usuário quando de definir uma tabela que será preenchida com as informações do Ítem. A entrada de dados por parte do usuário é feita através de um componente *modal*, que não passa de uma janela iterativa aberta na página, que é preenchido com os respectivos formulários necessário em cada página. Podemos ver o modal da pesquisa na Figura 9

A Lista faz uso do *hook* `useState` para dois fins: decidir quando um *modal* deve ser aberto e fechado e para acesso fácil às informações necessárias na página com capacidade de atualização em tempo real. A função `handleChange` é responsável por atualizar os valores do objeto da Lista e é chamada sempre que algum campo é atualizado com uma entrada do usuário.

Por fim o Ítem é quem preenche a Lista, contendo as informações individuais de cada

célula da tabela que serão apresentados ao usuário. É também quem contém os identificadores dos dados para a interação com o servidor, seja para a deleção dos dados ou para a atualização deles. Além disso, o Ítem contém os botões que permitem tais atualizações e deleções.

Os três componentes citados até agora, Página, Lista e Ítem contém, além disso, algumas funções para a comunicação entre eles e para a exibição condicional de botões. Podemos ver todas as páginas com seus componentes propriamente preenchidos nas figuras 10, 11, 12 e 13.

A Lista de Pesquisa, por ter que lidar com a entidade com maior número de campos, faz uso de um modelo de pesquisa contendo valores padrões para todos os campos de uma pesquisa. Esse modelo é usado para definir o estado `newResearch`.

A Lista de Pesquisa também contém a definição de listas para preencher os menus *drop-down*. Na criação do formulário se itera sobre a lista em questão e se cria um `MenuItem` para cada valor presente na lista, como podemos ver na Figura 14.

A Lista de Questionários também apresenta algumas diferenças para que possa permitir a inclusão de templates públicos já criados. A visualização dos questionários disponíveis envolve, além da exibição dos dados do questionário, dados das suas perguntas e das alternativas de suas perguntas, como podemos ver na Figura 15. Para isso fazemos uso de duas variáveis novas: `viewState` e `selectedIds`. A primeira é responsável por definir qual componente será renderizado no *modal* de inclusão de novo questionário, enquanto a segunda é responsável por armazenar o Id dos questionários selecionados pelo usuário.

3.4 Aplicativo

Por último tempos o aplicativo que é o principal componente da nossa plataforma e é a razão da existência das demais componentes. Sua principal funcionalidade é a coleta do tempo de uso dos demais aplicativos e embora no nosso protótipo a coleta esteja sendo restringida ao tempo de uso do próprio aplicativo, o funcionamento original coleta o tempo de uso de todos os aplicativos instalados no *smartphone*. Em função disso, todas as conclusões que chegaremos no que diz respeito a precisão da coleta e seu funcionamento podem ser extendidas aos demais aplicativos instalados. Podemos ver o diagrama de caso de uso do aplicativo, na Figura 18.

3.4.1 Sistema operacional

A escolha entre IOS e Android foi bastante simples. Além do Android ser muito mais popular que o IOS ⁷, o autor do trabalho não possui acesso à um celular com o sistema operacional Android.

⁷ <https://gs.statcounter.com/os-market-share/mobile/brazil>

3.4.2 Linguagem

Entre as duas possíveis linguagens para desenvolvimento com Android, Java e Kotlin, apesar de Kotlin se mostrar cada vez mais popular a familiaridade do autor com Java pesou de forma considerável nessa escolha. É importante salientar que o Android não faz uso de Java de uma forma convencional, sendo utilizada uma API Java, que dá acesso a apenas algumas funcionalidades da linguagem. Por exemplo, apesar de Java 8 possuir estruturas conhecidas como Stream, que facilita o trabalho com estruturas de agregação de dados, como Listas, Conjuntos e Mapas, a API Java usada para o Android não possui.

3.4.3 Visão geral

O fluxo que acontece no aplicativo começa de forma similar ao fluxo do site, com uma verificação se o usuário já se autenticou no aplicativo anteriormente e caso não tenha, carrega a tela de login. Nesse momento o usuário pode escolher entre logar e criar uma conta. Ambos os casos enviam para o servidor suas respectivas informações que são validadas pelo e caso a validação seja bem-sucedida, o aplicativo salva as informações para um login futuro.

A tela seguinte à autenticação permite o usuário buscar por pesquisas para participar, abrir uma das pesquisas que já participa, adicionar filtros ou deslogar. Buscar pesquisas para participar lista as pesquisas disponíveis para este usuário das quais ele pode escolher uma por vez para fazer parte, adicionar filtros carrega uma nova página com os possíveis filtros para o usuário preencher. Por fim, abrir uma pesquisa da qual participa carrega uma página contendo todos os questionários dessa pesquisa, ao selecionar um deles o usuário é direcionado para a primeira pergunta do questionário e ao responde-lá é direcionado para a seguinte. Após a última ser respondida todas elas são enviadas para o servidor com as respectivas respostas do usuário. O aplicativo também é responsável por coletar o tempo de uso dos demais aplicativos instalados no celular diariamente. Podemos acompanhar o fluxo geral através da Figura 16 e da Figura 17.

3.4.4 Componentes Auxiliares

Nosso aplicativo faz uso de alguns componentes auxiliares, como a classe `HttpRequest`, que é responsável por criar as requisições que serão enviadas ao servidor e uma classe que auxilia na exibição de logs ao longo do fluxo do aplicativo. As requisições que são criadas pela classe auxiliar, são utilizadas para facilitar o uso da biblioteca `OkHttp`.

A biblioteca `OkHttp` é responsável pelo envio e recebimento de informações entre o aplicativo e o servidor. Como a chamada é assíncrona e não queremos que a thread principal do aplicativo seja bloqueada no aguardo da resposta, todas as chamadas definirão dois callbacks que serão chamados a depender do sucesso da chamada ou não, `onFailure` e `onResponse`.

Todos os `onFailure` no nosso código apenas exibem uma mensagem para o usuário avisando do erro, já o `onResponse` é diferente para cada requisição.

3.4.5 Fluxo geral

A explicação inicial do aplicativo será feita seguindo o fluxo de utilização dele e conforme passarmos por atividades, que são as classes que representam as janelas onde aplicaremos nossa UI, iremos nos aprofundar nos modelos de UI que serão usados na atividade.

A primeira etapa no ciclo de vida de uma atividade, e a única que é necessariamente implementada em todas as nossas atividades, é a `onCreate`, onde é definida a lógica que deve ser executada apenas uma vez ao longo do ciclo de vida da atividade, isso inclui inicialização de variáveis, verificações do estado atual do aplicativo, definição da UI a ser renderizada na página e afins.

Nossa primeira atividade, `MainActivity`, é a responsável pelo login, porém começaremos nossa explicação pela atividade `CreateAccountActivity`, que é responsável pela criação de conta do participante. A atividade carrega como conteúdo da página a interface definida pelo `layout account_creation`, que podemos ver na Figura 19. A principal função dessa atividade é a `createAccount` onde são pegadas as informações inseridas pelo usuário nos campos e é feita uma requisição HTTP ao servidor. No caso de sucesso verificamos se o código da resposta é 200 e se for salvamos o token JWT extraído da resposta e chamamos a função `renderResearch`.

A função `renderResearch` faz uma chamada ao servidor solicitando as pesquisas do usuário e usa a função `getRequestBuilder` para formar a requisição, passando o endpoint e o token que será adicionado ao cabeçalho `Authorization`. É verificado se o código da resposta é 200 e se for, iniciamos a atividade `ResearchActivity` passando para ela, através do pacote de dados que será lido na criação da atividade, as pesquisas extraídas da resposta do servidor. Esse pacote de dados que é utilizado sempre que queremos inicializar uma nova atividade no aplicativo é um objeto do tipo `Intent` e a partir daqui nos referenciaremos a ele por esse nome.

Voltando para a atividade `MainActivity`, sua função `onCreate` além de conter a inicialização de alguns objetos que serão úteis, também contém lógicas importantes. A primeira delas é a verificação de se o usuário já se autenticou anteriormente e isso é feito verificando a existência de um token JWT salvo nas preferências do aplicativo. Se esse for o caso passamos direto para a função `renderResearch` passando o token existente. Se não, definimos o conteúdo da página com o `layout login` e relacionaremos variáveis aos objetos definidos no `layout`. Após isso verificamos as permissões que o aplicativo possui, sendo a mais importante delas a `PACKAGE_USAGE_STATS`, que é a permissão que garante o acesso ao tempo de uso dos demais aplicativos. Caso o usuário ainda não tenha concedido tal permissão, iniciamos as atividades para requisitá-la. O mesmo é feito para a permissão de alarme. Por fim, a UI é

carregada, dado a possibilidade para que o usuário interaja com o aplicativo.

Caso o usuário clique no botão de criação de conta a atividade atual é encerrada e inicia a atividade `CreateAccountActivity`. Caso o usuário preencha os campos e tente logar, uma requisição HTTP é feita para o servidor verificar as informações. Caso a requisição seja bem sucedida se verifica o código retornado e se for 200 extrai as informações do JSON recebido, salva o token JWT retornado e chama a função `renderResearch`, que se comporta de forma idêntica à função de mesmo nome da atividade `CreateAccountActivity`.

A próxima classe do nosso fluxo, `ResearchActivity`, é responsável por exibir as pesquisas que o participante já adicionou a sua conta e permitir que o participante busque novas pesquisas. Através dela o usuário pode tanto abrir uma de suas pesquisas, que executará a função `retrieveData`, quanto iniciar, através da função `newResearch`, a atividade responsável pela a adição de novas pesquisas. Além disso, essa classe também é responsável por receber os dados através do *intent*, converter eles para um conjunto de `Research`, que a classe que representa as pesquisas no nosso modelo, e pela inicialização nos alarmes responsáveis pela coleta do tempo de uso, que entraremos mais em detalhes ao fim da seção.

O layout que é renderizado ao iniciarmos a atividade possui alguns detalhes. Se compararmos a Figura 21, que é o que o usuário do aplicativo vê, com a Figura 22 que é o layout antes de ser renderizado, vemos que após a renderização os ítems contidos no layout, representados pelos valores `Item 0`, `Item 1` e demais são substituídos por um componente com o título da pesquisa, no caso da imagem "Pesquisa com filtro" e um botão "Abrir". Isso acontece porque a exibição de listas em celulares pode ser bastante custosa a depender do tamanho da lista. Para contornar isso o Android possui uma componente com o objetivo de simular uma lista possivelmente infinita que preenche os ítems para o usuário apenas quando eles precisam ser exibidos na tela no momento, permitindo o usuário arrastar a interface para cima ou para baixo para navegar pela lista. Não entraremos em detalhes quanto a implementação desses mecanismos, porém sempre que exibirmos listas no nosso aplicativo, iremos fazer uso desse componente e preencheremos a lista com as informações pertinentes a depender de cada página.

A função `newResearch` faz uma requisição HTTP GET para o servidor pedindo todas as pesquisas às quais o participante tem acesso. Caso a resposta seja 200 essas pesquisas são adicionadas ao *intent* que é usado para iniciar a atividade `NewResearchActivity`. Similarmente a função `retrieveData` também faz uma requisição HTTP GET para o servidor, passando no caminho o id da pesquisa em questão com o objetivo de receber todos os questionários que fazem parte da pesquisa selecionada pelo usuário e ao receber com sucesso uma resposta com código 200, inicia a atividade `QuestionnaireActivity` passando o corpo da resposta pelo *intent*

A atividade `NewResearchActivity` é bastante simples e renderiza o mesmo *layout* que a atividade anterior, porém esconde o botão de novas pesquisas e preenche a lista com um modelo contendo além do título e o botão a descrição da pesquisa. Ela converte os dados re-

cebidos em pesquisas e apresenta para que o participante escolha entre elas. Após escolhida a pesquisa, uma requisição HTTP PATCH é feita para o servidor usando `patchRequestBuilder` para gerar a requisição apropriada passando o id da pesquisa a ser adicionada. No caso de uma resposta com código 204 é chamada a função `renderResearch` que já vimos anteriormente.

A atividade `QuestionnaireActivity` tem na sua função `onCreate` a inicialização das variáveis, isso inclui ler os dados recebidos pelo `intent` e converte-los para uma lista de objetos do tipo `Questionnaire`, modelo usado para representar questionários no sistema. Depois disso são exibidas as opções de questionário para o usuário que, ao escolher uma das opções chama a função `retrieveData` passando como parametro o id do questionário selecionado. A função trata de fazer uma requisição para o servidor para pegar todas as questões pertencentes ao questionário, assim como todas as alternativas pertencentes às questões e passar os dados recebidos para a última atividade, `QuestionActivity`.

Nossa próxima atividade `QuestionActivity` é a única que faz uso de fragmentos para a exibição dos conteúdos. Fragmentos são pedaços reutilizáveis da UI que precisam obrigatoriamente fazer parte de, ou outro fragmento, ou de uma atividade para poderem ser renderizados. No nosso caso possuímos dois fragmentos `MultipleAlternativeFragment` e `DescriptiveAlternativeFragment`.

Para o funcionamento condicional do fragmento precisamos olhar para a função `onCreate` da atividade, que além de converter os dados para `Question`, é responsável por definir como conteúdo da página o `layout base_fragment`, que é um fragmento com um único elemento, `FrameLayout` que serve como `placeholder`. Se a questão a ser renderizada no momento for de múltipla escolha, então o fragmento `MultipleAlternativeFragment` substituirá o placeholder, caso contrário o fragmento será substituído pelo `DescriptiveAlternativeFragment`. Ambos os fragmentos são bastante similares, possuindo como principal diferença a função `onViewCreated` que é responsável por definir quais elementos serão exibidos no fragmento. No caso do `DescriptiveAlternativeFragment` essa função possui um `textView` que serve de entrada de dados para o usuário digitar a resposta, enquanto no `DescriptiveAlternativeFragment` a função itera pela lista de alternativas, representadas pela classe `Alternative` e preenche o fragmento com botões radiais contendo todas as alternativas.

Em ambos os casos, caso o botão "próxima" seja pressionado pelo usuário a resposta do usuário é incluída na lista de alternativas selecionadas, `selectedAlternatives`, e a função `processNextQuestion` é chamada. A função verifica se ainda existe questões no questionário para serem respondidas, se sim então a próxima questão da lista é pega e o processo todo se repete até voltarmos a esse ponto. Se a questão é de fato a última, então uma requisição do tipo POST é formada na função `postRequestBuilder` contendo como corpo a lista de respostas do usuário e em caso de um resposta com código 204, volta para a atividade `Ques-`

tionnaireActivity.

3.4.6 Coleta de tempo de uso

Parte fundamental do aplicativo é a coleta do tempo de uso. Como já mencionamos na Seção 3.4.5, a coleta de uso é feita baseada na inicialização de um alarme reliazada na atividade `ResearchAtividade`, onde criamos um alarme que ao ser ativado dispara um sinal que inicia a execução da função `onReceive` da classe `MyBroadcastReceiver`. Essa função é responsável tanto por criar um novo alarme com o mesmo objetivo do anterior para ser disparado em 24 horas quando de coletar o tempo de uso e enviar para o servidor.

Nosso aplicativo usa um alarme especial que possui permissões para ser disparado de forma precisa mesmo que o celular esteja em modo de descanso. Essas informações são importantes porque impactam diretamente na precisão da coleta do tempo de uso. O Android toma a liberdade de alterar o momento do disparo de alarmes com o intuito de poupar bateria, seja atrasado ou adiantando o disparo para que ele seja feito junto com demais processos que acontecem próximos ao do disparo definido na criação do alarme. Ao definirmos o alarme como "exato", informamos ao sistema operacional que ele não deve alterar o momento do disparo, se mantendo fiel ao definido pelo alarme.

A coleta do tempo de uso se restringe ao tempo no qual cada aplicativo ficou em *foreground* nas últimas 24 horas, ou seja, o tempo total em que o aplicativo passou aberto em primeiro plano pelo usuário. Isso quer dizer que se um usuário passou 1 hora com um aplicativo em segundo plano o tempo não será contabilizado. O tempo é entregue pelo sistema operacional em milisegundos, que no nosso caso é convertido para o formato hh:mm:ss para facilitar a leitura.

Depois da coleta os dados são transformados em texto seguindo o modelo "Pacote do aplicativo: tempo coletado". O texto é então adicionado ao corpo de uma requisição que é enviada ao servidor para ser salva.

The image shows a modal form for adding a new survey. The modal is overlaid on a table of existing surveys. The modal contains the following fields:

- Título
- Descrição
- Visibilidade
- Idade inicial
- Idade final
- Salário inicial
- Salário final
- Gênero
- Orientação sexual
- Raça/Etnia

At the bottom of the modal are two buttons: CANCELAR and ENVIAR.

Título	Descrição	Visibilidade	Estado
Pesquisa exemplo	Descrição breve da pesquisa exemplo	public	ina
Outra Pesquisa	Mais uma descrição	public	ina
Pesquisa com filtros	Pesquisa particular	privada	ina

Logout

Questionários

Ações

EXIBIR QUESTIONÁRIOS

EXIBIR QUESTIONÁRIOS

EXIBIR QUESTIONÁRIOS

INICIAR PESQUISA

INICIAR PESQUISA

INICIAR PESQUISA

EDITAR

EDITAR

EDITAR

EXCLUIR

EXCLUIR

EXCLUIR

Figura 9 – Modal para a adição de uma pesquisa

pesquisas

Suas pesquisas

ADICIONAR PESQUISA

Título	Descrição	Visibilidade	Estado	Início	Encerramento	Módulos ativos	Filtros	Questionários	Ações
Pesquisa exemplo	Descrição breve da pesquisa exemplo	public	inativa			EXIBIR MÓDULOS	● ● ● ●	EXIBIR QUESTIONÁRIOS	INICIAR PESQUISA EDITAR EXCLUIR
Outra Pesquisa	Mais uma descrição	public	inativa			EXIBIR MÓDULOS	● ● ● ●	EXIBIR QUESTIONÁRIOS	INICIAR PESQUISA EDITAR EXCLUIR
Pesquisa com filtros	Pesquisa particular	privada	inativa			EXIBIR MÓDULOS	20-30 anos Acima de 5000 BRL Bisexual	EXIBIR QUESTIONÁRIOS	INICIAR PESQUISA EDITAR EXCLUIR

Logout

Figura 10 – Tela de pesquisas

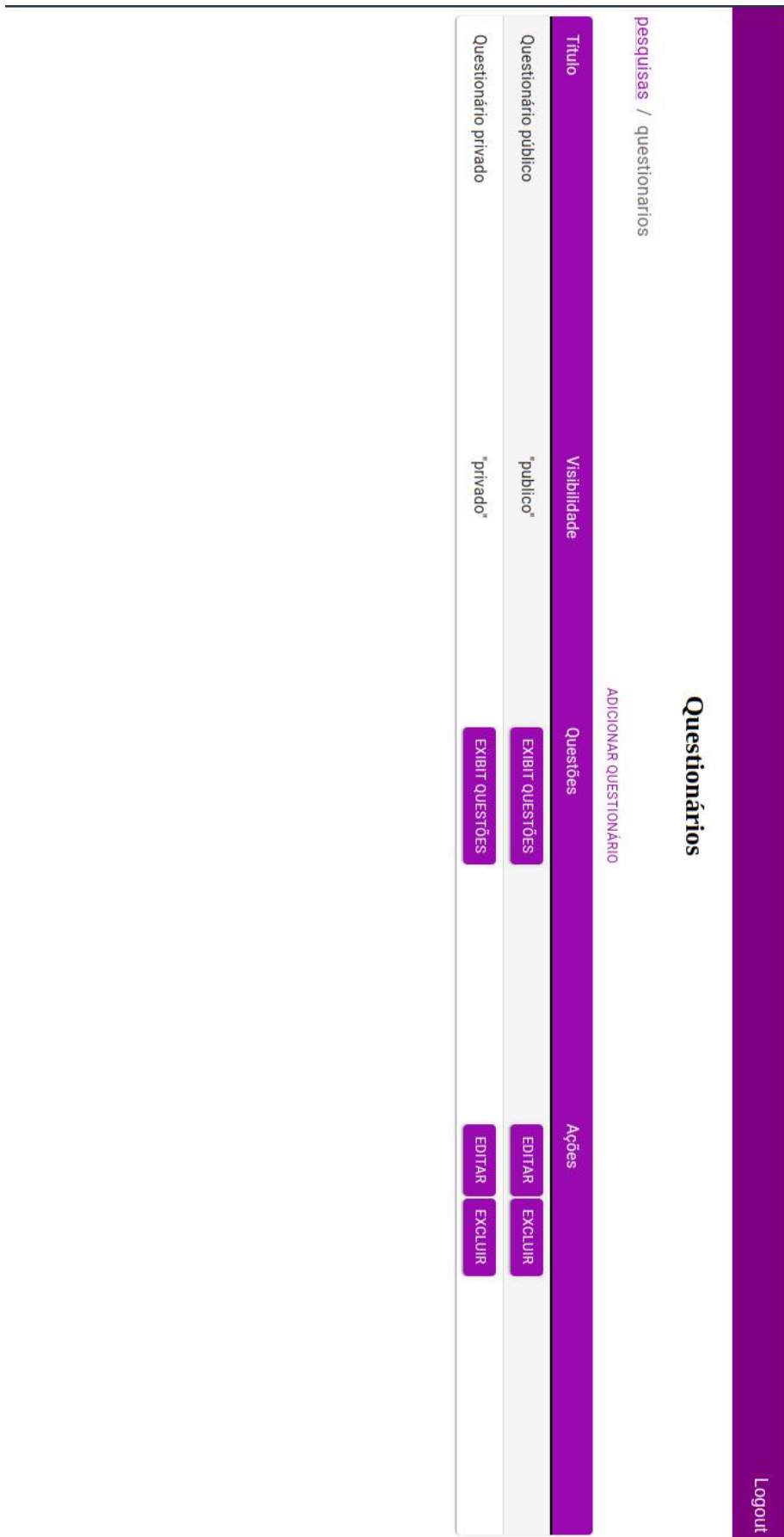


Figura 11 – Tela dos questionários de uma pesquisa

Logout

pesquisas / questionarios / questoes

Questões

ADICIONAR QUESTÃO

ID	Pergunta	Ordem	Tipo	Alternativas	Ações
1	Primeira questão	0	descritiva	EXIBIR ALTERNATIVAS	EDITAR EXCLUIR
2	Segunda questão	0	descritiva	EXIBIR ALTERNATIVAS	EDITAR EXCLUIR
3	Terceira questão	0	múltipla_escolha	EXIBIR ALTERNATIVAS	EDITAR EXCLUIR

Figura 12 – Tela dos questionários de uma pesquisa

Alternativas

pesquisas / questionarios / questoes / alternativas

ADICIONAR ALTERNATIVA

Texto	Valor	Ações
Primeira Alternativa	0	EDITAR EXCLUIR
Segunda Alternativa	1	EDITAR EXCLUIR
Terceira Alternativa	2	EDITAR EXCLUIR

Figura 13 – Tela dos questionários de uma pesquisa

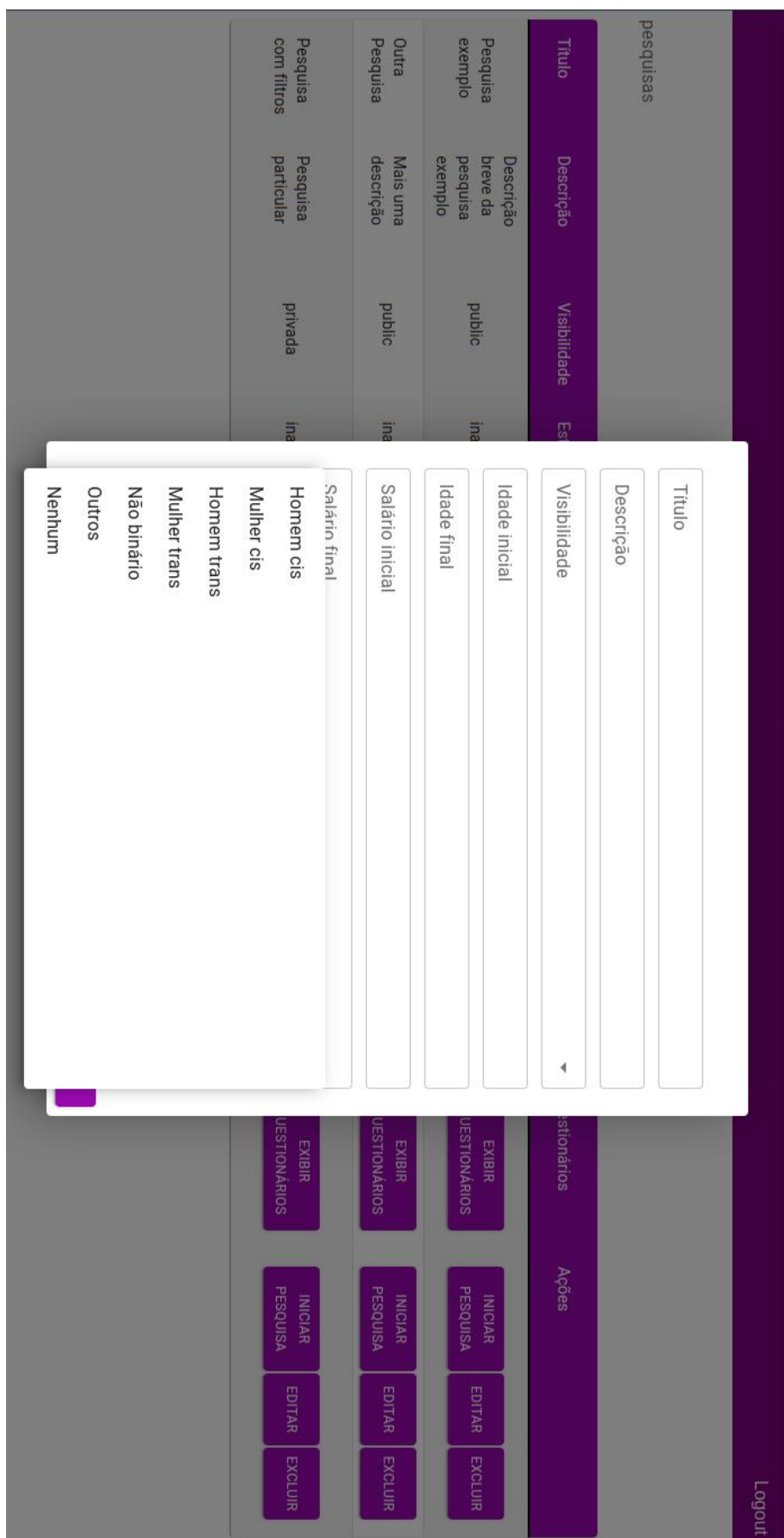


Figura 14 – Tela de pesquisas

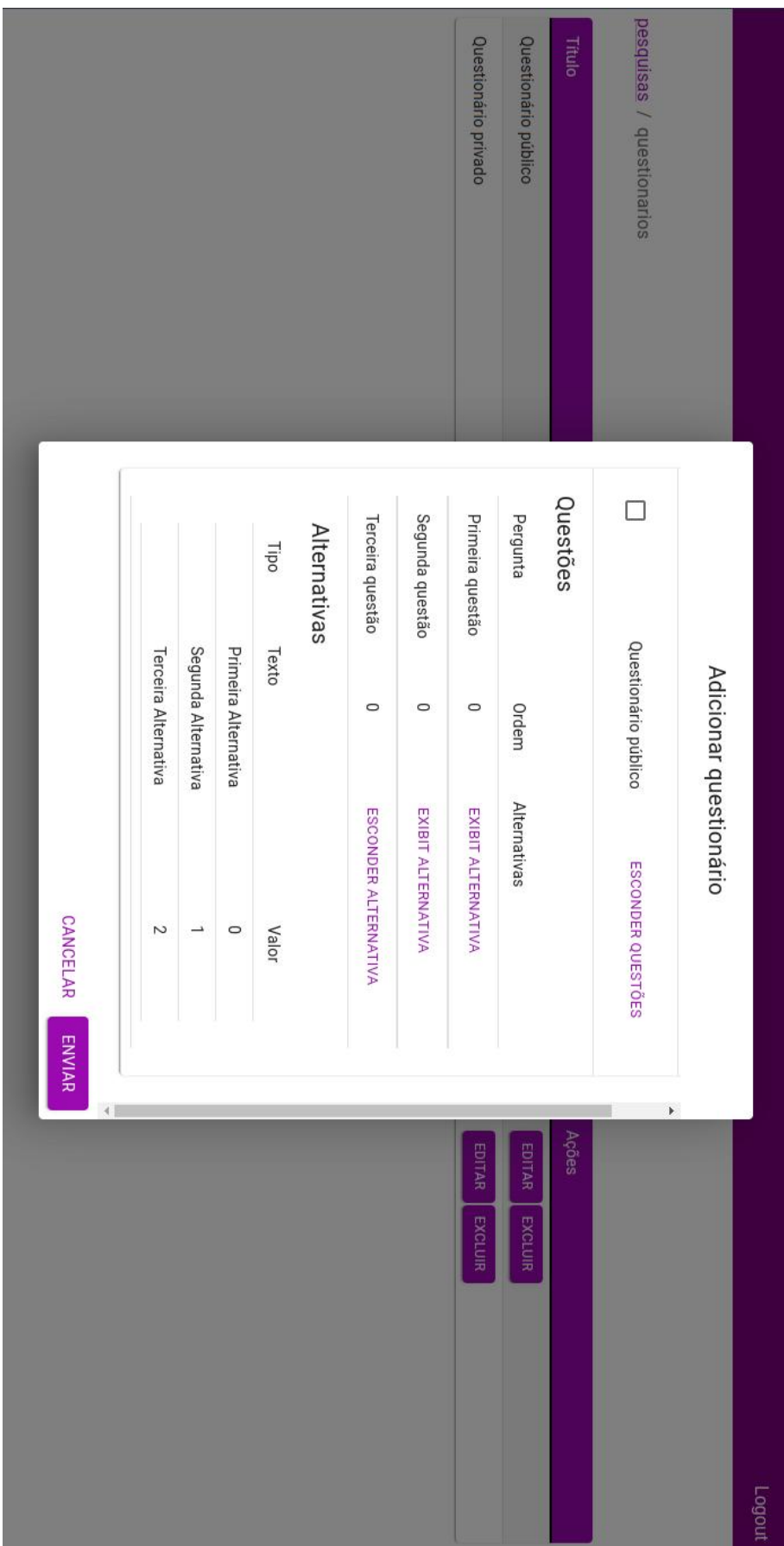


Figura 15 – Tela de pesquisas

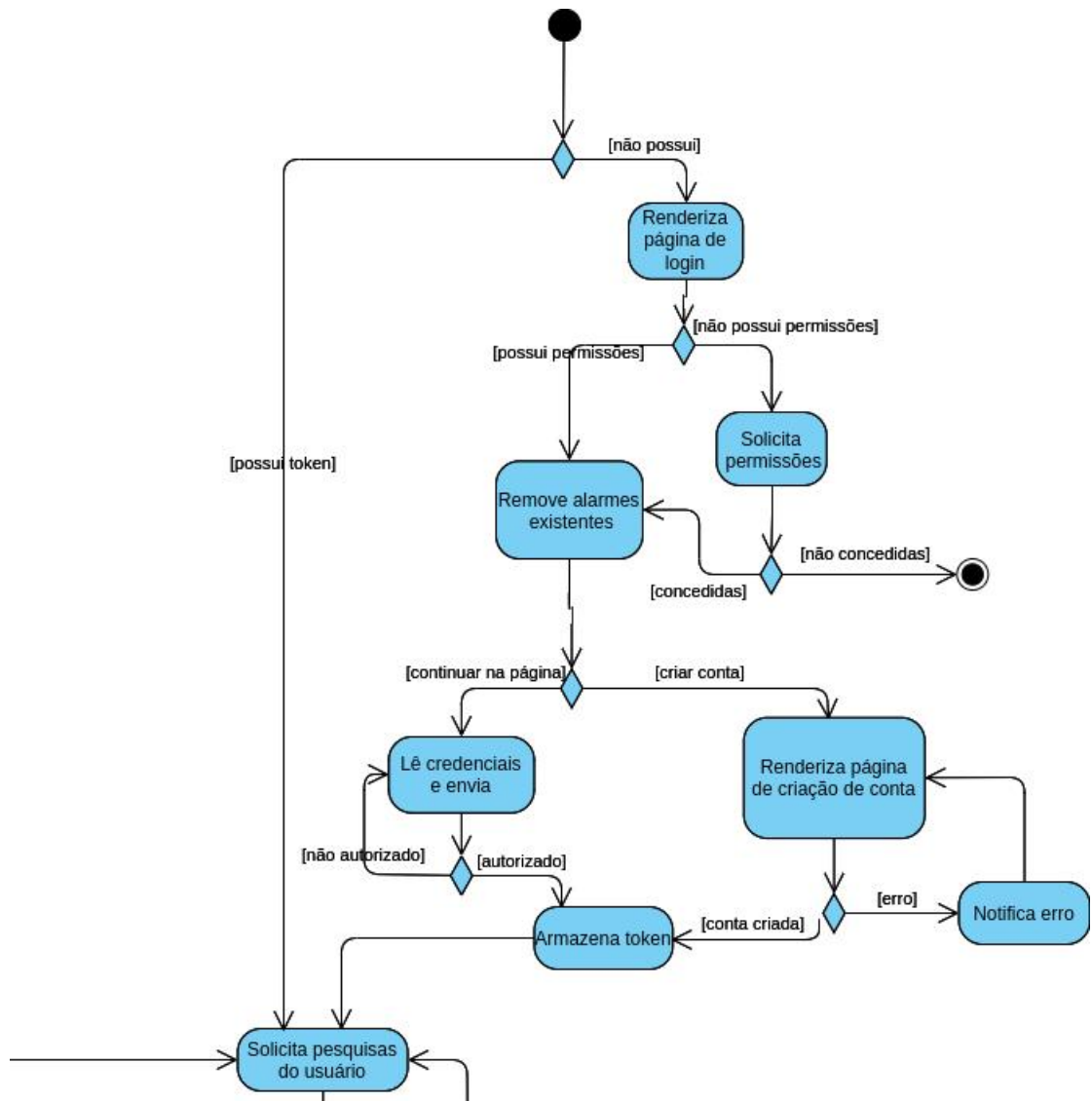
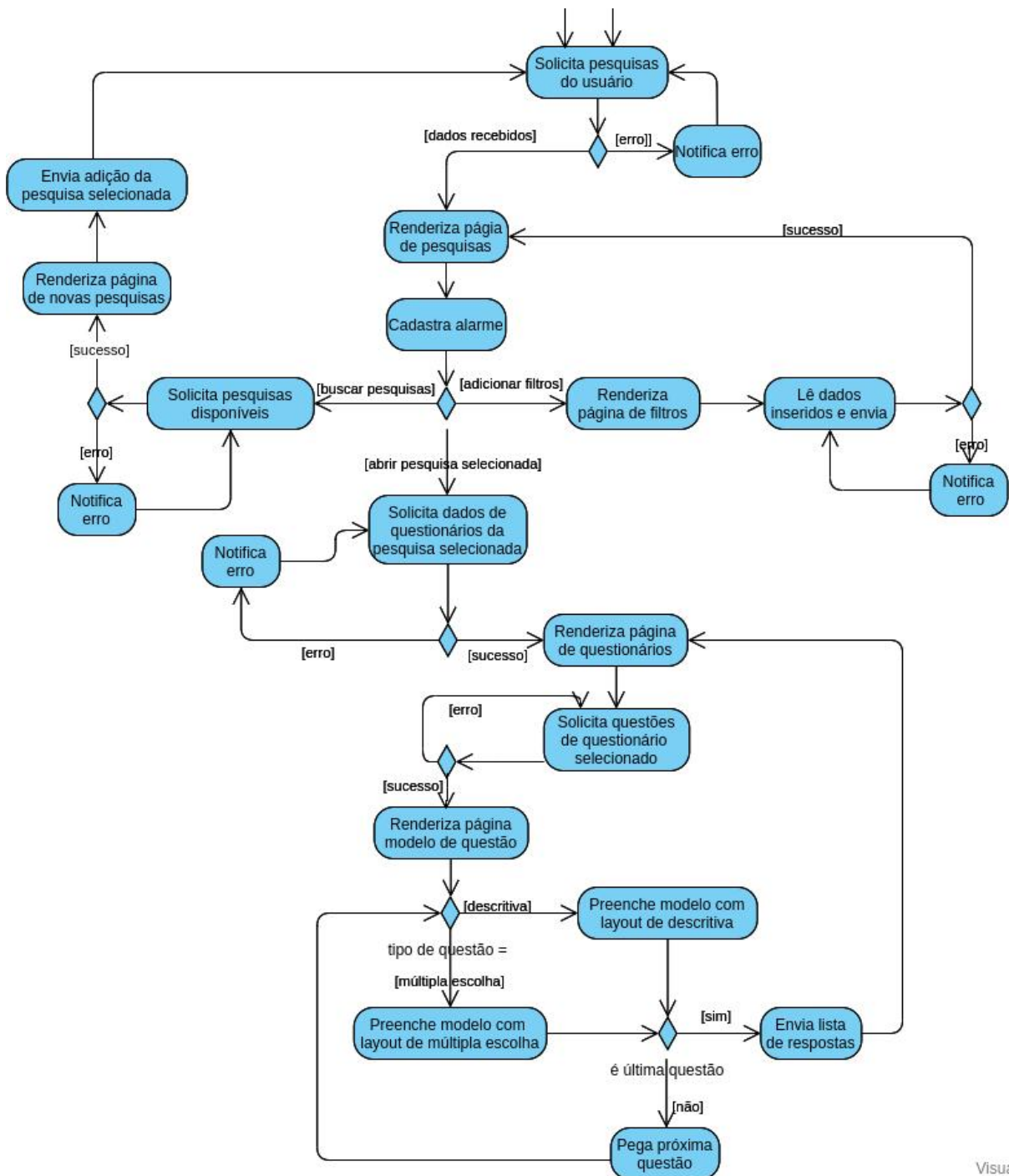


Figura 16 – Primeira parte do diagrama de atividade do aplicativo



Visual

Figura 17 – Segunda parte do diagrama de atividade do aplicativo

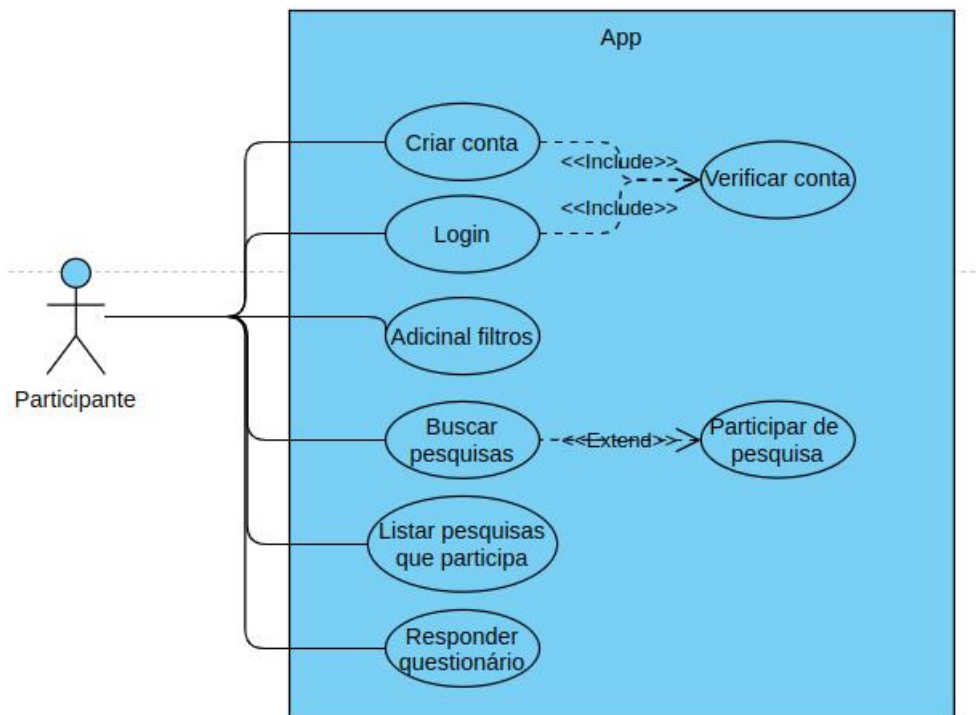


Figura 18 – Diagrama de uso do aplicativo

Tela de criação de conta com o seguinte conteúdo:

NÃO USE DADOS REAIS EM NENHUM DOS CAMPOS ABAIXO

CPF
xxx.xxx.xxx-xx

E-mail

Senha
Senha

Qual nome você prefere usar?

CANCELAR CRIAR CONTA

Figura 19 – Tela de criação de conta

CPF

xxx.xxx.xxx-xx

Senha

Senha

ESQUECI MINHA SENHA

ACESSAR

CRIAR CONTA

Figura 20 – Tela de login



Figura 21 – Tela de pesquisas

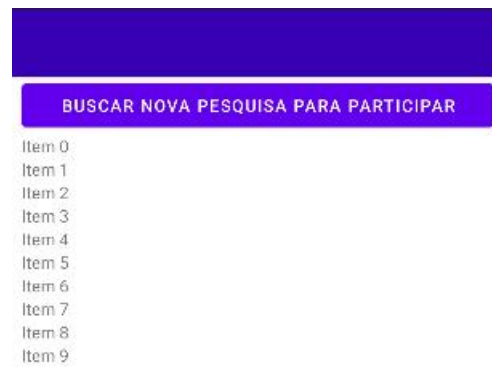


Figura 22 – Layout para exibição de pesquisas

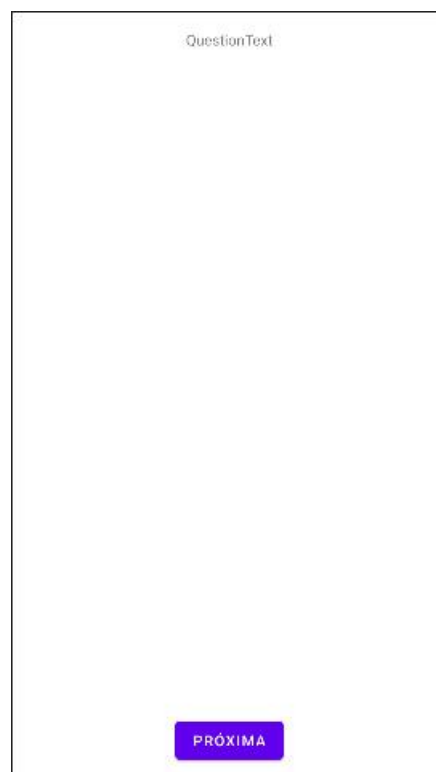


Figura 23 – Layout do fragmento para questões de multipla alternativa

The diagram illustrates the layout for a descriptive question interface. It features a vertical rectangular frame. At the top, the text "QuestionText" is positioned. Below it, the text "Sua resposta" is centered. At the bottom, there is a horizontal line, and below that line is a purple button with the text "PRÓXIMA" in white capital letters.

Figura 24 – Layout do fragmento para questões descritivas

4 Testes e validação

Neste capítulo iremos descrever os testes e validações da coleta de tempo de uso do aplicativo na Seção 4.1 e os testes de carga do servidor na Seção 4.2. Não realizaremos testes no site porque nenhum componente do site é um componente crítico e sua utilização é secundária perante os outros dois componentes.

4.1 Coleta do tempo de uso

De todas as métricas que podemos usar para avaliar a coleta do tempo de uso a precisão é a mais importante para o nosso trabalho. Iremos realizar dois tipos de teste para avaliar a precisão da coleta do tempo de uso: testes de coleta contínua e testes de coleta intermitente.

Nos testes de coleta contínua o aplicativo fica em primeiro plano no celular utilizado entre as coletas. Já nos testes de coleta intermitente o aplicativo será colocado em primeiro plano por períodos específicos e em seguida será suspenso ou encerrado até o horário da coleta. Além de testarmos a precisão, também estaremos avaliando a capacidade do aplicativo de enviar os dados nos três possíveis estados do ciclo de vida, em primeiro plano, suspenso e encerrado.

Os testes de coleta intermitente terão o aplicativo aberto em intervalos que precisarão de controle manual, ou seja, será responsabilidade do autor abrir, fechar e contabilizar por quanto tempo o aplicativo passou em primeiro plano. Isso adiciona um grau de imprecisão na avaliação da coleta do tempo de uso, porém utilizaremos temporizadores para garantirmos um grau de imprecisão suficientemente baixo.

O aparelho que será utilizado para a realização dos testes será um Samsung, modelo SM-A135M/DS, com a versão do sistema operacional 12, que faz uso da API de versão 31.

4.2 Servidor

Para o servidor faremos testes de carga e de estresse. Para isso, utilizaremos o LoadView¹ que é uma ferramenta de testes automatizados. A ferramenta recebe alguns parâmetros e se encarrega de realizar as requisições e gerar um relatório ao final. Usaremos tal relatório na Seção 4.3

¹ <https://www.loadview-testing.com/>

Intervalo de coleta	Tempo manual	Tempo automático	Diferença
00:05:00	00:05:25	00:04:11	00:01:14
00:05:00	00:10:25	00:04:11	00:06:14
00:15:00	00:15:34	00:09:52	00:05:42
00:15:00	00:30:24	00:29:25	00:00:59
00:15:00	00:45:24	00:29:25	00:15:59
00:15:00	01:00:27	00:49:25	00:11:02
00:30:00	00:30:39	00:19:58	00:10:41
01:00:00	01:00:27	00:46:51	00:13:36
01:00:00	02:00:30	01:46:51	00:13:39
01:00:00	03:00:37	02:46:51	00:13:46

Tabela 3 – Tabela com os resultados da coleta contínua

4.3 Resultados

Nessa seção sobriremos os resultados dos testes de precisão da coleta do tempo de uso do aplicativo, na Subseção 4.3.1, e dos testes de carga e estresse realizados no servidor, na Subseção 4.3.2. Por fim, discutiremos a completude dos requisitos na Subseção 4.3.3.

Precisa ficar claro que por não possuímos autorização para a realização de testes envolvendo mais participantes todos os testes foram realizados pelo autor utilizando um único dispositivo celular.

4.3.1 Testes de precisão

Os testes de precisão foram feitos em um único dispositivo e o tempo coletado foi apenas do aplicativo desenvolvido.

Os resultados dos testes de precisão serão divididos em duas tabelas. A primeira, Tabela 3 contém os resultados da coleta contínua, enquanto a segunda, Tabela 4 contém os resultados da coleta intermitente.

A tabela da coleta contínua contém três parâmetros, intervalo de coleta, tempo manual e tempo automático. O intervalo é o valor que usamos para definir quando cada coleta é realizada, ou seja um intervalo de coleta de 5 minutos coleta os dados de uso e envia para o servidor a cada 5 minutos. É importante ressaltar que sempre que o intervalo de coleta é alterado, o tempo de uso do aplicativo é reiniciado. O tempo manual é o tempo coletado pelo autor no qual o aplicativo ficou em primeiro plano no celular. O tempo automático é o tempo coletado pelo próprio aplicativo. Tanto o tempo manual quanto o automático são acumulativos por intervalo, por isso vemos o tempo manual crescendo baseado no intervalo de coleta na Tabela 3.

Intervalo de coleta	Tempo manual	Tempo automático	Diferença
12:00:00	00:18:40	00:18:18	00:00:22
12:00:00	00:58:46	00:58:34	00:00:22
12:00:00	00:55:30	00:55:02	00:00:28
12:00:00	01:20:40	01:21:08	00:00:28
24:00:00	00:50:00	00:50:08	00:00:08
24:00:00	01:00:00	00:59:48	00:00:12
24:00:00	01:42:00	01:42:15	00:00:15

Tabela 4 – Tabela com os resultados da coleta contínua

A tabela de coleta intermitente possui os mesmos três parâmetros, porém é importante ressaltar que, diferente da coleta contínua, a coleta intermitente divide o tempo total de forma desigual sempre entre 4 intervalos de coleta. Por exemplo um tempo total de 50 minutos pode ter sido coletado com parcelas de 25 minutos, 5 minutos, 3 minutos e 17 minutos. Da mesma forma que a coleta contínua, os valores do tempo manual crescem conforme o intervalo. O último detalhe da coleta intermitente é que o nosso aplicativo sempre coleta às últimas 24 horas, logo um intervalo de 24h nunca acumula com qualquer intervalo anterior.

4.3.2 Testes de carga e estresse

A ferramenta que utilizamos para a realização dos testes recebe alguns parâmetros que serão utilizados no teste. Número de usuários inicial, aumento de usuários ao longo de um período determinado e por último manutenção do número de usuários por um período determinado. No nosso caso iniciamos o teste com 5 usuário, aumentamos os usuários em 19 por minuto ao longo de 5 minutos e por fim mantivemos 100 usuários ao longo de 7 minutos. O endpoint utilizado do servidor para teste foi o `</pesquisas>`.

Como podemos ver na Figura 25 o número de sessões iniciadas é reduzido rapidamente depois do minuto 6, mesmo que, segundo nossos parâmetros, o número de sessões deveria continuar subindo até o último minuto. Isso se deu em função de interrompermos o teste de forma prematura por percebermos o número de erros retornado pelo servidor. Como podemos ver se compararmos as figuras Figura 26, que contém o número total de sessões, e Figura 27, que contém tanto o número total de sessões em verde quanto o número de sessões com erro em vermelho, podemos ver que todas as requisições feitas a partir da 40ª resultou em erro por parte do servidor.

Apesar do erro, o tempo médio de resposta do servidor, antes dos erros começarem a acontecer na marca dos 25 segundos, é relativamente baixo, 154ms. Como podemos ver na Figura 28

Isso se deu pelo limite de sessões existente no servidor, que é de 40 sessões. Tal limita-

ção nos leva a considerar desnecessário qualquer outro teste além do inicial.

4.3.3 Requisitos

Se voltarmos a avaliar os requisitos contidos na Figura 1 após o desenvolvimento podemos contar que todos eles foram cumpridos.

A coleta do tempo de uso de forma automática através de alarmes descrita na Seção 3.4.6 cumpre com o RF1. Coleta automática do tempo de uso e o RF2. Coleta realizada na mesma frequência para todos.

Se analisarmos a estrutura das tabelas Pesquisa e Participante no banco pela Figura 5, veremos que a tabela Pesquisa contém diversos filtros enquanto a tabela Participante apresenta informações pessoais baseadas nas categorias dos filtros. Ao longo da Seção 3.4.5 vimos que as informações do Participante são utilizadas na aplicação dos filtros da Pesquisa para exibir apenas as Pesquisas apropriadas, cumprindo com o RF3. Filtro de pesquisas.

Toda a Seção 3.3 descreve o desenvolvimento do site que é utilizado para interação da plataforma com os pesquisadores. Isso cumpre tanto com o RF4. Plataforma visual para interação dos pesquisadores quanto com o RNF2. Portabilidade para pesquisadores.

Por último o RNF1. Autenticação pode ser visto sendo cumprido tanto nas Seções 3.3.3 e 3.4 quanto mais propriamente na Figura 5 onde vemos as informações dos usuários nas tabelas Participante e Pesquisador.

4.4 Conclusão

Pudemos perceber que, segundo os testes de coleta contínua os valores da coluna de diferença são preocupantemente altos, que resultaria em coletas pouco precisas do tempo de uso do aplicativo, que vai no sentido oposto ao proposto por este trabalho. Porém, se analisarmos com mais cuidado a tabela, percebemos que a coluna de tempo automático é curiosamente repetitiva dentro de cada intervalo com mudanças a cada 15 minutos. O último intervalo da coleta contínua, de 1 hora, é especialmente curioso, pois após a primeira coleta, que possui um erro de aproximadamente 15 minutos, apesar de propagar o erro, coleta os intervalos seguintes de forma correta.

Se considerarmos os valores da coluna de diferença da coleta intermiente, perceberemos que os valores não passam de 30 segundos. Apesar de um número menor de testes, não percebemos o mesmo problema de valores repetidos na coleta automática. Isso nos leva a acreditar que exista alguma restrição não descrita na documentação que impede a coleta do tempo de uso em intervalos curtos de tempo. Restrição que não se aplica a coletas com intervalos tão grandes quanto 12 horas e mais importante 24 horas, que é o caso do nosso aplicativo.

Se compararmos o resultado dos testes intermitentes de precisão da coleta do tempo de uso com os resultados obtidos em (LEE et al., 2017; OHME et al., 2021; LEE et al., 2021) que chegam a estimar o erro da coleta manual, feita pelo usuário, em 40% para estimativas inferiores ao tempo real, podemos perceber que a solução proposta neste trabalho, de um aplicativo para coleta e envio automatizado de dados de uso, apresenta uma margem de erro de coleta menor.

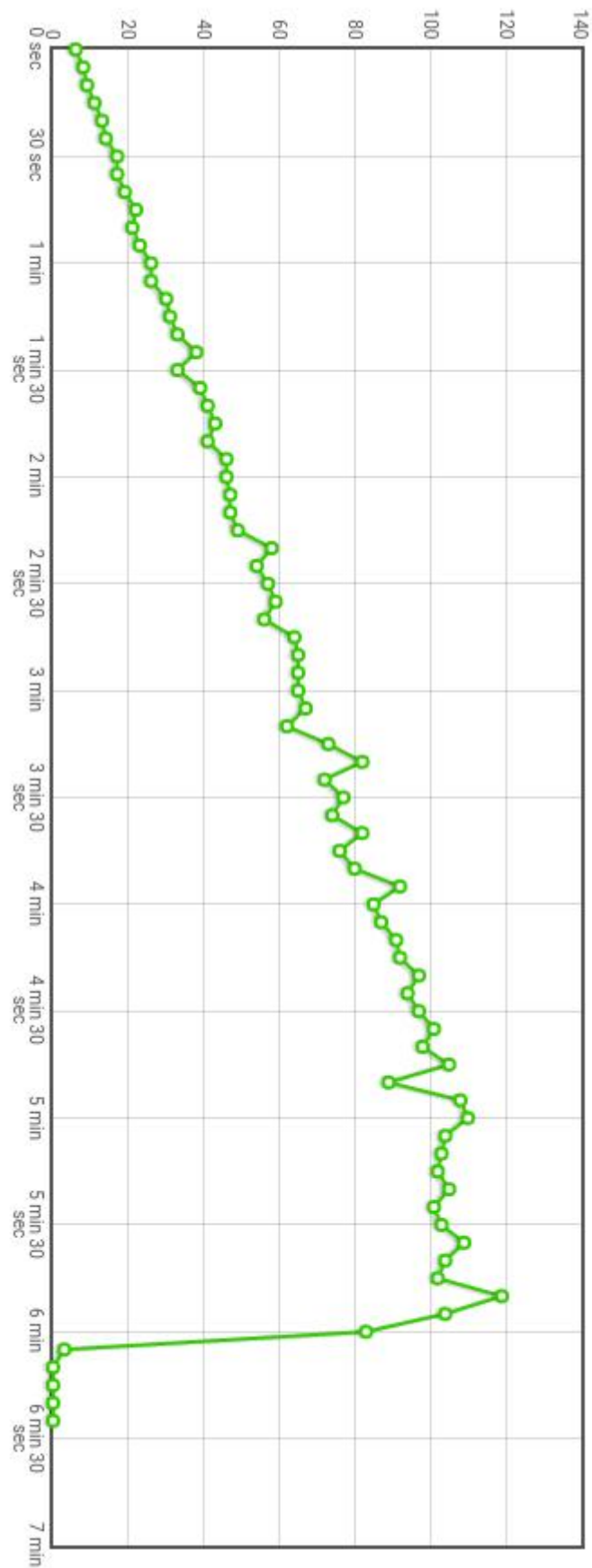


Figura 25 – Gráfico referente ao número de seções iniciadas

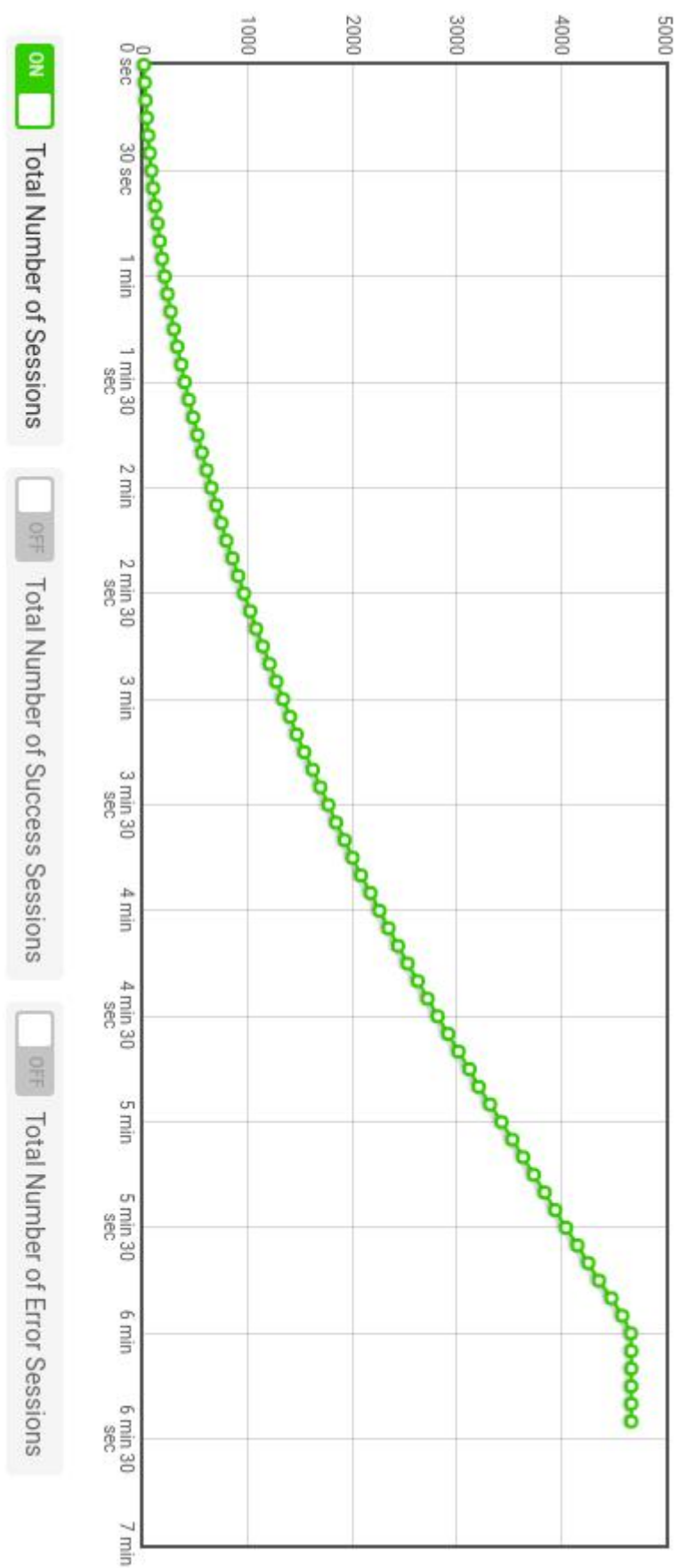


Figura 26 – Gráfico referente ao número total de sessões

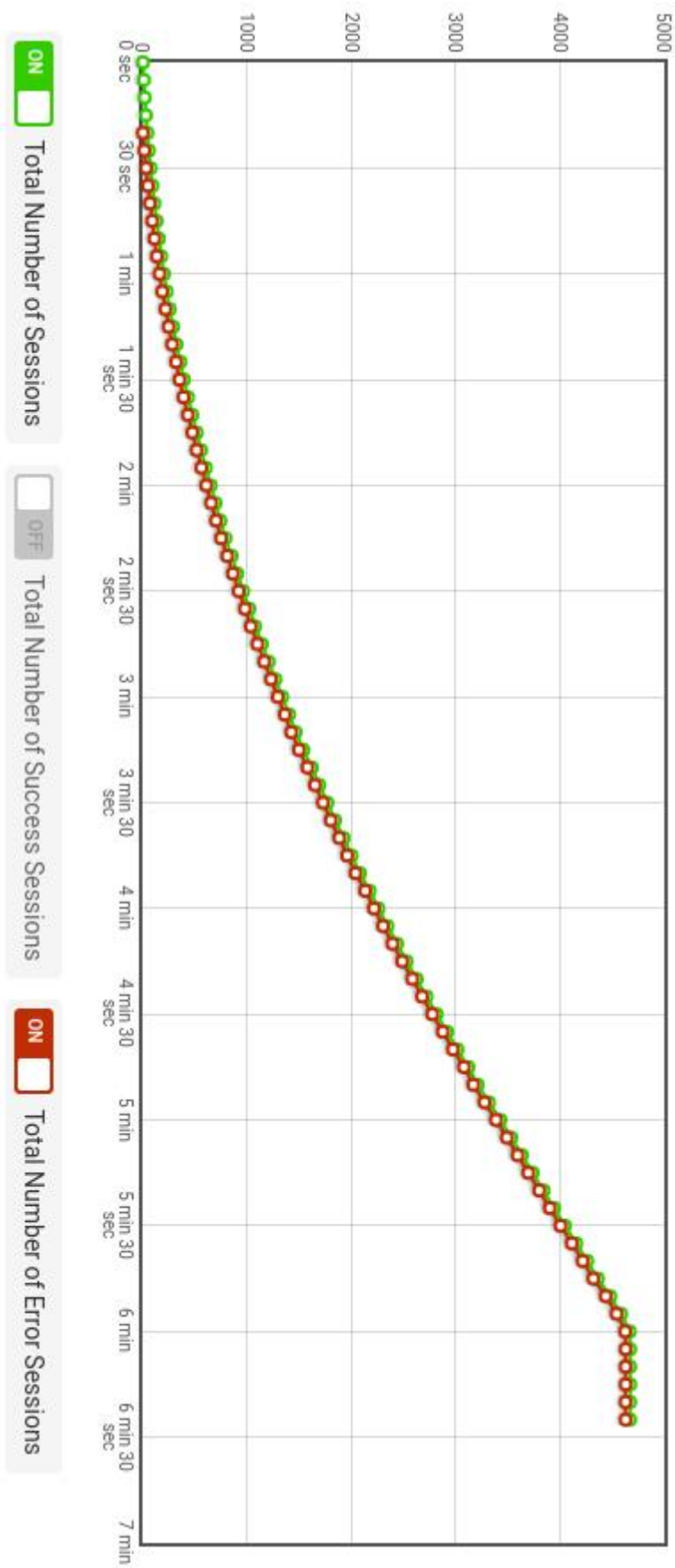


Figura 27 – Gráfico referente ao número total de sessões com erro

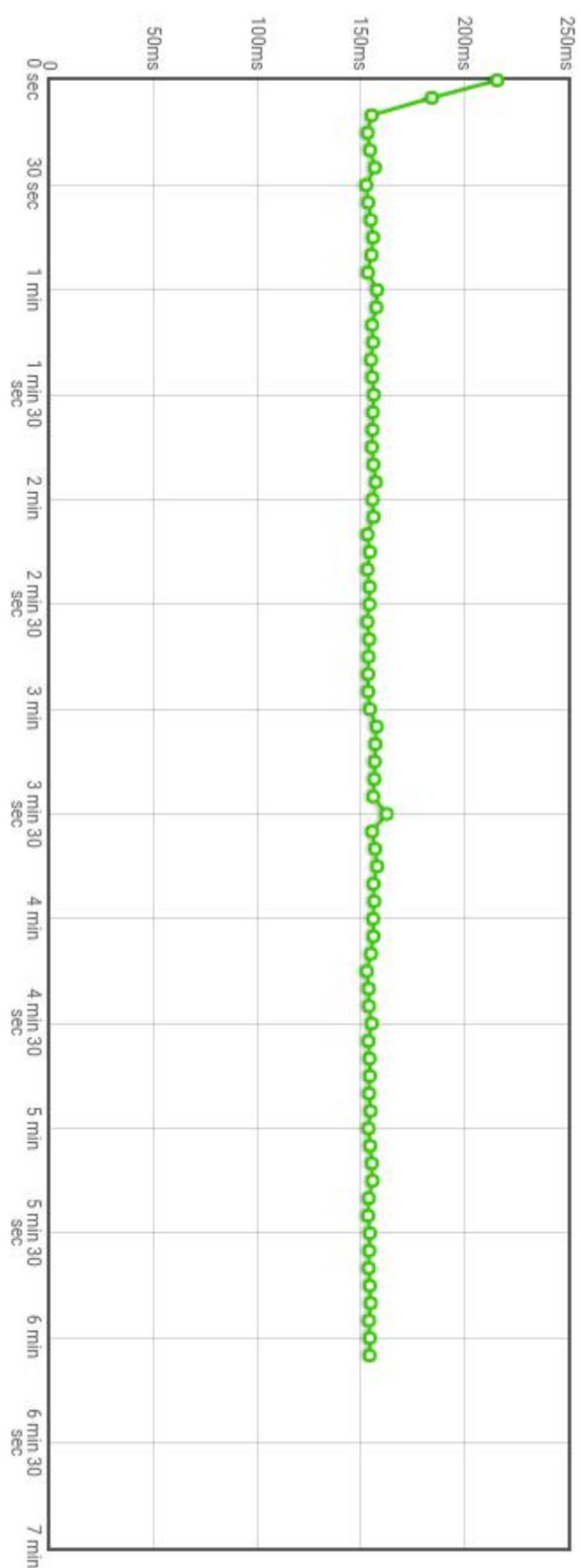


Figura 28 – Gráfico referente ao tempo de resposta do servidor

Referências

- ALHASSAN, A. A. et al. The relationship between addiction to smartphone usage and depression among adults: a cross sectional study. *BMC Psychiatry*, v. 18, 2018.
- BARRY, C. T. et al. Adolescent social media use and mental health from adolescent and parent perspectives. *Journal of Adolescence*, v. 61, p. 1 – 11, 2017. ISSN 0140-1971. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0140197117301318>>.
- BAWDEN, D.; ROBINSON, L. The dark side of information: overload, anxiety and other paradoxes and pathologies. *Journal of Information Science*, v. 35, n. 2, p. 180–191, 2009. Disponível em: <<https://doi.org/10.1177/0165551508095781>>.
- BENITO-RUIZ, E. Infoxicação 2.0. In: _____. *Handbook of Research on Web 2.0 and Second Language Learning*. Hershey: IGI Global, 2009. cap. 4, p. 60–79.
- BERRYMAN, C.; NEGY, C. Social media use and mental health among young adults. *Psychiatric Quarterly*, v. 89, p. 1–8, 06 2018.
- CHO, J. et al. The rate and delay in overload: An investigation of communication overload and channel synchronicity on identification and job satisfaction. *Journal of Applied Communication Research*, Routledge, v. 39, n. 1, p. 38–54, 2011. Disponível em: <<https://doi.org/10.1080/00909882.2010.536847>>.
- EPPLER, M.; MENGIS, J. The concept of information overload: A review of literature from organization science, accounting, marketing, mis, and related disciplines. *Inf. Soc.*, v. 20, p. 325–344, 11 2004.
- GUO, N. et al. Problematic smartphone use and mental health in chinese adults: A population-based study. *International Journal of Environmental Research and Public Health*, v. 17, n. 3, 2020. ISSN 1660-4601. Disponível em: <<https://www.mdpi.com/1660-4601/17/3/844>>.
- HAUG, S. et al. Smartphone use and smartphone addiction among young people in switzerland. *Journal of Behavioral Addictions J Behav Addict*, Akadémiai Kiadó, Budapest, Hungary, v. 4, n. 4, p. 299 – 307, 2015. Disponível em: <<https://akjournals.com/view/journals/2006/4/4/article-p299.xml>>.
- HAWI, N. S.; SAMAHA, M. Relationships among smartphone addiction, anxiety, and family relations. *Behaviour & Information Technology*, Taylor Francis, v. 36, n. 10, p. 1046–1052, 2017. Disponível em: <<https://doi.org/10.1080/0144929X.2017.1336254>>.
- JELENCHICK, L. A.; EICKHOFF, J. C.; MORENO, M. A. “facebook depression?” social networking site use and depression in older adolescents. *Journal of Adolescent Health*, v. 52, n. 1, p. 128–130, 2013. ISSN 1054-139X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1054139X12002091>>.
- KELLY, Y. et al. Social media use and adolescent mental health: Findings from the uk millennium cohort study. *EClinicalMedicine*, v. 6, 01 2019.

- KIL, N. et al. Examining associations between smartphone use, smartphone addiction, and mental health outcomes: A cross-sectional study of college students. *Health Promot Perspect*, Tabriz University of Medical Sciences, v. 11, n. 1, p. 36–44, 2021. ISSN 22286497. Disponível em: <<https://hpp.tbzmed.ac.ir/Article/hpp-32876>>.
- KOROLEVA, K.; KRASNOVA, H.; GÜNTHER, O. 'stop spamming me!' - exploring information overload on facebook. In: . [S.l.: s.n.], 2010. v. 4, p. 447.
- LEE, A. R.; SON, S.-M.; KIM, K. K. Information and communication technology overload and social networking service fatigue: A stress perspective. *Computers in Human Behavior*, v. 55, p. 51 – 61, 2016. ISSN 0747-5632. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0747563215300893>>.
- LEE, H. et al. Comparing the Self-Report and Measured Smartphone Usage of College Students: A Pilot Study. *Psychiatry Investig*, v. 14, n. 2, p. 198–204, Mar 2017.
- LEE, P. H. et al. Validation of Self-Reported Smartphone Usage Against Objectively-Measured Smartphone Usage in Hong Kong Chinese Adolescents and Young Adults. *Psychiatry Investig*, v. 18, n. 2, p. 95–100, Feb 2021.
- LEVINE, J.; DVORAK, R.; HALL, B. Problematic smartphone use: A conceptual overview and systematic review of relations with anxiety and depression psychopathology. *Journal of Affective Disorders*, in press, 08 2016.
- LIN, Y.-H. et al. Time distortion associated with smartphone addiction: Identifying smartphone addiction via a mobile application (app). *Journal of Psychiatric Research*, v. 65, p. 139–145, 2015. ISSN 0022-3956. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0022395615001004>>.
- LIU, M.; WU, L.; YAO, S. Dose-response association of screen timebased sedentary behaviour in children and adolescents and depression: A meta-analysis of observational studies. *British journal of sports medicine*, v. 9, 11 2015.
- MATTHES, J. et al. “too much to handle”: Impact of mobile social networking sites on information overload, depressive symptoms, and well-being. *Computers in Human Behavior*, v. 105, p. 106217, 2020. ISSN 0747-5632. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0747563219304364>>.
- MONTEZUMA, L. Q. V. Ansiedade e dependência de smartphone em acadêmicos do curso de odontologia da universidade federal do ceará. Faculdade de Farmácia, Odontologia e Enfermagem, Universidade Federal do Ceará, 2020.
- MOROMIZATO, M. A.-r. S. et al. O Uso de Internet e Redes Sociais e a Relaçã com Índicios de Ansiedade e Depressão em Estudantes de Medicina. *Revista Brasileira de Educação* , scielo, v. 41, p. 497 – 504, 12 2017. ISSN 0100-5502. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-55022017000400497&nrm=iso>.
- NASCIMENTO, D. S. et al. Consequências do uso excessivo do smartphone na saúde mental de estudantes universitários: Uma revisão integrativa. *Cadernos Brasileiros de Saúde Mental/Brazilian Journal of Mental Health*, v. 12, n. 32, p. 1–15, ago. 2020. Disponível em: <<https://periodicos.ufsc.br/index.php/cbsm/article/view/69812>>.

OHME, J. et al. Mobile data donations: Assessing self-report accuracy and sample biases with the ios screen time function. *Mobile Media & Communication*, v. 9, n. 2, p. 293–313, 2021. Disponível em: <<https://doi.org/10.1177/2050157920959106>>.

O'REILLY, M. et al. Is social media bad for mental health and wellbeing? exploring the perspectives of adolescents. *Clinical Child Psychology and Psychiatry*, v. 23, n. 4, p. 601–613, 2018. PMID: 29781314. Disponível em: <<https://doi.org/10.1177/1359104518775154>>.

PANOVA, T. et al. Specific smartphone uses and how they relate to anxiety and depression in university students: a cross-cultural perspective. *Behaviour & Information Technology*, Taylor Francis, v. 39, n. 9, p. 944–956, 2020. Disponível em: <<https://doi.org/10.1080/0144929X.2019.1633405>>.

SAMAH, M.; HAWI, N. S. Relationships among smartphone addiction, stress, academic performance, and satisfaction with life. *Computers in Human Behavior*, v. 57, p. 321–325, 2016. ISSN 0747-5632. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0747563215303162>>.

SCHICK, A. G.; GORDON, L. A.; HAKA, S. Information overload: A temporal approach. *Accounting, Organizations and Society*, v. 15, n. 3, p. 199–220, 1990. ISSN 0361-3682. Disponível em: <<https://www.sciencedirect.com/science/article/pii/036136829090005F>>.

SEABROOK, E. M.; KERN, M. L.; RICKARD, N. S. Social networking sites, depression, and anxiety: A systematic review. *JMIR Ment Health*, v. 3, n. 4, p. e50, Nov 2016. ISSN 2368-7959. Disponível em: <<http://mental.jmir.org/2016/4/e50/>>.

SOHN, S. Y. et al. Prevalence of problematic smartphone usage and associated mental health outcomes amongst children and young people: a systematic review, meta-analysis and grade of the evidence. *BMC Psychiatry*, v. 19, 2019. ISSN 1471-244X.

STEPHENS, K. K. et al. Reconceptualizing communication overload and building a theoretical foundation. *Communication Theory*, v. 27, n. 3, p. 269–289, 03 2017. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/comt.12116>>.

VERDUYN, P. et al. Do social network sites enhance or undermine subjective well-being? a critical review: Do social network sites enhance or undermine subjective well-being? *Social Issues and Policy Review*, v. 11, p. 274–302, 01 2017.

APÊNDICE A – Códigos da plataforma

A.0.1 Site

O código pode ser acessado através do seguinte repositório:
<https://github.com/LuisOw/web-tracker>.

A.0.2 Aplicativo

O código pode ser acessado através do seguinte repositório:
<https://github.com/LuisOw/tracker/tree/main/app/src/main>.

A.0.3 Servidor

O código pode ser acessado através do seguinte repositório:
<https://github.com/LuisOw/api-tracker>.

APÊNDICE B – Artigo

Plataforma tecnológica para pesquisas sobre saúde mental e tempo de uso de smartphones

Luis O. S. Ganoza¹ Jônata Tyska Carvalho¹

¹Departamento de Informática e Estatística – Universidade Federal de Santa Catarina
Caixa Postal 476 – 88040-900 – Florianópolis – SC – Brasil

`luis.oswaldo@grad.ufsc.br, jonata.tyska@ufsc.br`

Abstract. *Over the last few years, several studies have been carried out with the aim of understanding more deeply the relationship and impact of smartphone use with mental health. Such surveys make use of usage time data collected by the participant, which have low reliability. This article proposes a platform that helps in the development of research and increases the accuracy of smartphone usage time collected.*

Resumo. *Ao longo dos últimos anos diversas pesquisas vem sendo feitas com o objetivo de entender mais afundo a relação e o impacto do uso de smartphones com saúde mental. Tais pesquisas fazem uso de dados de tempo de uso levantados pelo participante, que possuem baixa confiabilidade. Este artigo propõe uma plataforma que ajuda no desenvolvimento de pesquisas e aumenta o grau de precisão do tempo de uso de smartphones.*

1. Introdução

O aumento de informações com as quais os indivíduos entram em contato diariamente é uma consequência direta da massiva expansão da Internet e da popularização dos *smartphones*. Informações que antes estavam disponíveis através de livros, enciclopédias, dicionários, jornais e afins passaram a ser encontradas em um único local e em maiores quantidades após a popularização de computadores com acesso à Internet, porém com limitações maiores quanto a mobilidade. Com a popularização dos *smartphones*, essas últimas restrições de mobilidade deixaram de existir e a facilidade de acesso foi imensamente potencializada, passando a estar disponível em apenas alguns toques em qualquer lugar.

Diversas pesquisas têm sido realizadas para avaliar os índices de utilização de smartphones de forma geral e sua relação com depressão, ansiedade e estresse, sejam elas realizadas no Brasil [Montezuma 2020] e [Nascimento et al. 2020] ou fora dele [Levine et al. 2016], [Liu et al. 2015], [Hawi and Samaha 2017], [Panova et al. 2020], [Alhassan et al. 2018], [Sohn et al. 2019], [Samaha and Hawi 2016], [Guo et al. 2020]. Apesar do elevado número de pesquisas nessa área, elas apresentam limitações, inclusive pontuados pelos autores e autoras. Dentre elas, baixo número de participantes e grupos muito homogêneos, não consideração de fatores econômicos-sociais e o principal deles, dados reportados pelos entrevistados (baixa confiabilidade).

Além disso podemos encontrar pesquisas [Lee et al. 2021], [Lee et al. 2017], [Ohme et al. 2021] que apontam para uma baixa confiabilidade no que diz respeito à coleta do tempo de uso de smartphones pelos usuários. Se combinarmos tais resultados com

a importância do tempo de uso para as pesquisas de saúde mental citadas anteriormente encontramos um problema.

2. Solução proposta

Este trabalho tem como objetivo desenvolver uma plataforma tecnológica (sistema web + aplicativo) com capacidade de medir com maior precisão a utilização dos *smarthphones* tanto de forma geral (tempo total de uso do *smarthphone*), quanto a utilização de aplicativos individualmente, para aumentar a confiabilidade de tais dados. A plataforma também terá capacidade de coletar dados manualmente, através de escalas cadastradas pelos próprios pesquisadores, para aumentar a abrangência de dados coletados e a capacidade de atender a múltiplas pesquisas de forma simultânea, para estimular um maior número de pesquisas na área.

2.1. Servidor e Site

Tanto o site quanto o servidor, apesar de secundários se comparados ao aplicativo, são componentes fundamentais da solução proposta. É através do site que pesquisadores podem facilmente gerenciar diversas pesquisas seja adicionando novas pesquisas, alterando e excluindo pesquisas existentes e também acompanhar os resultados das pesquisas. O servidor por sua vez é quem se comunica tanto com o site quanto com o aplicativo e faz a ponte de ambos com o banco de dados.

O banco de dados possui 9 (nove) tabelas, Pesquisador, Pesquisa, Questionário, Questão, Alternativa, Alternativa.Resposta, Participante e TempoDeUso e 1 (uma) tabela que relaciona Pesquisas com Participantes. Podemos acompanhar visualmente tais tabelas e relações na Figura 1.

O servidor também é responsável pela autenticação e garantia de autorização para acesso aos recursos do sistema. Depois de se autenticar o usuário recebe um token de acesso que precisará ser passado no cabeçalho das requisições para o servidor. Através do token é verificado se o usuário tem acesso aos recursos que solicita. Importante salientar que aplicamos a função de hash *bcrypt* para armazenar apenas o hash das senhas dos usuários.

2.2. Aplicativo

O aplicativo é peça chave da nossa solução. É através dele que os participantes irão interagir com nosso sistema, garantindo fácil capacidade de divulgação para que os pesquisadores possam atingir grupos maiores em suas pesquisas. É também responsabilidade do aplicativo coletar o tempo de uso dos demais aplicativos instalados no dispositivo.

O aplicativo foi desenvolvido para Android e através de funções do sistema temos acesso à diversas informações de utilização do dispositivo, desde tempo de uso de aplicativos até uso de bateria. Para o sistema, tempo de uso é considerado o tempo no qual um aplicativo passa em primeiro plano e essa é a informação que coletamos para enviar ao servidor.

A coleta é controlada através de alarmes, outra função disponível pelo sistema operacional. Através dela podemos definir um alarme para ser disparado a cada 24 (vinte e quatro) horas com a função de coletar o tempo de uso do último dia de todos os aplicativos instalados no dispositivo e enviar para o servidor.

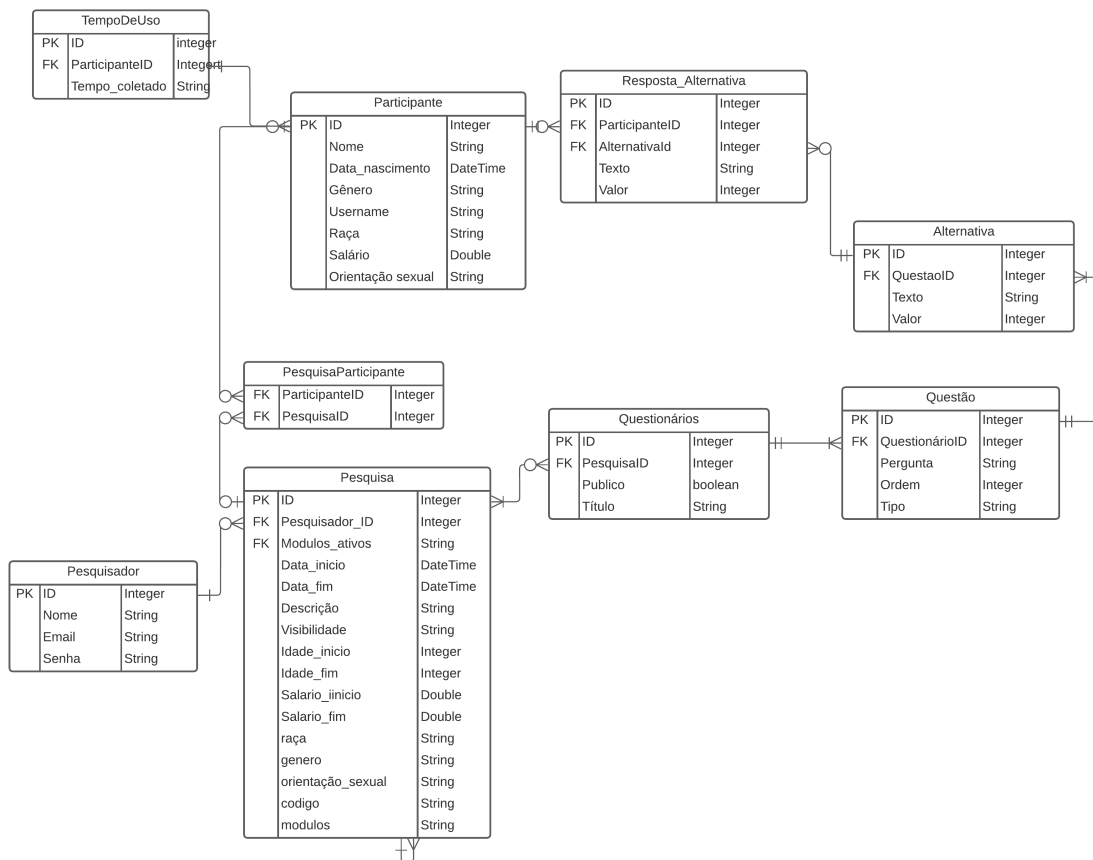


Figure 1. Diagrama do banco de dados

Ambas as funções mencionadas anteriormente do sistema operacional necessitam de permissões específicas, que são solicitadas ao usuário no início da utilização do aplicativo pela primeira vez.

3. Avaliando a precisão

Para avaliar a precisão do método de coleta proposto foram feitas 2 (duas) baterias de teste, a primeira de testes contínuos e a segunda de testes intermitentes. O resultado dos testes contínuos podem ser vistos na Tabela 1 e os resultados dos testes intermitentes podem ser vistos na Tabela 2.

A tabela da coleta contínua contém três parâmetros, intervalo de coleta, tempo manual e tempo automático. O intervalo é o valor que usamos para definir quando cada coleta é realizada, ou seja um intervalo de coleta de 5 minutos coleta os dados de uso e envia para o servidor a cada 5 minutos. É importante ressaltar que sempre que o intervalo de coleta é alterado, o tempo de uso do aplicativo é reiniciado. O tempo manual é o tempo coletado pelo autor no qual o aplicativo ficou em primeiro plano no celular. O tempo automático é o tempo coletado pelo próprio aplicativo.

A tabela de coleta intermitente possui os mesmos três parâmetros, porém é importante ressaltar que, diferente da coleta contínua, a coleta intermitente divide o tempo total de forma desigual sempre entre 4 intervalos de coleta. Por exemplo um tempo total de 50

Intervalo de coleta	Tempo manual	Tempo automático	Diferença
00:05:00	00:05:25	00:04:11	00:01:14
00:05:00	00:10:25	00:04:11	00:06:14
00:15:00	00:15:34	00:09:52	00:05:42
00:15:00	00:30:24	00:29:25	00:00:59
00:15:00	00:45:24	00:29:25	00:15:59
00:15:00	01:00:27	00:49:25	00:11:02
00:30:00	00:30:39	00:19:58	00:10:41
01:00:00	01:00:27	00:46:51	00:13:36
01:00:00	02:00:30	01:46:51	00:13:39
01:00:00	03:00:37	02:46:51	00:13:46

Table 1. Tabela com os resultados da coleta contínua

Intervalo de coleta	Tempo manual	Tempo automático	Diferença
12:00:00	00:18:40	00:18:18	00:00:22
12:00:00	00:58:46	00:58:34	00:00:22
12:00:00	00:55:30	00:55:02	00:00:28
12:00:00	01:20:40	01:21:08	00:00:28
24:00:00	00:50:00	00:50:08	00:00:08
24:00:00	01:00:00	00:59:48	00:00:12
24:00:00	01:42:00	01:42:15	00:00:15

Table 2. Tabela com os resultados da coleta contínua

minutos pode ter sido coletado com parcelas de 25 minutos, 5 minutos, 3 minutos e 17 minutos.

4. Considerações finais

Pudemos perceber que, segundo os testes de coleta contínua os valores da coluna de diferença são preocupantemente altos, que levaria à coletas pouco precisas do tempo de uso dos aplicativos instalados, que vai no sentido oposto ao proposto por este trabalho. Porém, se analisarmos com mais cuidado a tabela, percebemos que a coluna de tempo automático é curiosamente repetitiva dentro de cada intervalo com mudanças a cada 15 minutos. O último intervalo da coleta contínua, de 1 hora, é especialmente curioso, pois após a primeira coleta, que possui um erro de aproximadamente 15 minutos, apesar de propagar o erro, coleta os intervalos seguintes de forma correta.

Se considerarmos os valores da coluna de diferença da coleta intermiente, perceberemos que os valores não passam de 30 segundos. Apesar de um número menor de testes, não percebemos o mesmo problema de valores repetidos na coleta automática. Isso nos leva a acreditar que exista alguma restrição à coleta do tempo de uso em intervalos curtos de tempo. Restrição que não se aplica a coletas com intervalos tão grandes

quanto 12 horas e mais importante 24 horas, que é o caso do nosso aplicativo.

Se compararmos o resultado dos testes intermitentes de precisão da coleta do tempo de uso com os resultados obtidos em [Lee et al. 2021], [Lee et al. 2017], [Ohme et al. 2021] que chegam a estimar o erro da coleta manual, feita pelo usuário, em 40% para estimativas inferiores ao tempo real, podemos perceber que a solução proposta neste trabalho, de um aplicativo para coleta e envio automatizado de dados de uso, apresenta uma margem de erro de coleta menor.

References

- Alhassan, A. A., Alqadhib, E. M., Taha, N. W., Alahmari, R. A., Salam, M., and Almutairi, A. F. (2018). The relationship between addiction to smartphone usage and depression among adults: a cross sectional study. *BMC Psychiatry*, 18.
- Guo, N., Luk, T. T., Ho, S. Y., Lee, J. J., Shen, C., Oliffe, J., Chan, S. S.-C., Lam, T. H., and Wang, M. P. (2020). Problematic smartphone use and mental health in chinese adults: A population-based study. *International Journal of Environmental Research and Public Health*, 17(3).
- Hawi, N. S. and Samaha, M. (2017). Relationships among smartphone addiction, anxiety, and family relations. *Behaviour & Information Technology*, 36(10):1046–1052.
- Lee, H., Ahn, H., Nguyen, T. G., Choi, S. W., and Kim, D. J. (2017). Comparing the Self-Report and Measured Smartphone Usage of College Students: A Pilot Study. *Psychiatry Investig*, 14(2):198–204.
- Lee, P. H., Tse, A. C. Y., Wu, C. S. T., Mak, Y. W., and Lee, U. (2021). Validation of Self-Reported Smartphone Usage Against Objectively-Measured Smartphone Usage in Hong Kong Chinese Adolescents and Young Adults. *Psychiatry Investig*, 18(2):95–100.
- Levine, J., Dvorak, R., and Hall, B. (2016). Problematic smartphone use: A conceptual overview and systematic review of relations with anxiety and depression psychopathology. *Journal of Affective Disorders*, in press.
- Liu, M., Wu, L., and Yao, S. (2015). Dose-response association of screen timebased sedentary behaviour in children and adolescents and depression: A meta-analysis of observational studies. *British journal of sports medicine*, 9.
- Montezuma, L. Q. V. (2020). Ansiedade e dependência de smartphone em acadêmicos do curso de odontologia da universidade federal do ceará.
- Nascimento, D. S., Veloso, F. C. S., Sampaio, D. L., Oliveira, M. J. C., and Kassab, S. B. (2020). Consequências do uso excessivo do smartphone na saúde mental de estudantes universitários: Uma revisão integrativa. *Cadernos Brasileiros de Saúde Mental/Brazilian Journal of Mental Health*, 12(32):1–15.
- Ohme, J., Araujo, T., de Vreese, C. H., and Piotrowski, J. T. (2021). Mobile data donations: Assessing self-report accuracy and sample biases with the ios screen time function. *Mobile Media & Communication*, 9(2):293–313.
- Panova, T., Carbonell, X., Chamarro, A., and Puerta-Cortés, D. X. (2020). Specific smartphone uses and how they relate to anxiety and depression in university students: a cross-cultural perspective. *Behaviour & Information Technology*, 39(9):944–956.

Samaha, M. and Hawi, N. S. (2016). Relationships among smartphone addiction, stress, academic performance, and satisfaction with life. *Computers in Human Behavior*, 57:321–325.

Sohn, S. Y., Rees, P., Wildridge, B., Kalk, N. J., and Carter, B. (2019). Prevalence of problematic smartphone usage and associated mental health outcomes amongst children and young people: a systematic review, meta-analysis and grade of the evidence. *BMC Psychiatry*, 19.