

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Raphael Luca de Castella e Bertholucci

**Aplicativo móvel para coleta de dados sobre o
impacto da hospitalização em crianças e
adolescentes**

Uberlândia, Brasil

2018

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Raphael Luca de Castella e Bertholucci

**Aplicativo móvel para coleta de dados sobre o impacto
da hospitalização em crianças e adolescentes**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau
de Bacharel em Ciência da Computação.

Orientador: Maria Adriana Vidigal Lima

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2018

Raphael Luca de Castella e Bertholucci

Aplicativo móvel para coleta de dados sobre o impacto da hospitalização em crianças e adolescentes

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 13 de dezembro de 2018:

Maria Adriana Vidigal Lima
Orientador

Rodrigo Sanches Miani

Renan Gonçalves Cattelan

Uberlândia, Brasil
2018

Dedico este trabalho a todos que contribuíram direta ou indiretamente em minha formação acadêmica, em especial ao meus pais que não mediram esforços para que eu chegasse até esta etapa da minha vida.

Agradecimentos

Agradeço a todos que contribuíram no decorrer desta jornada, em especialmente:

A Deus, por ter me dado foco, forças e fé para que eu conseguisse realizar este sonho. Além de ter me dado saúde e disposição para concluir este projeto.

Aos meus pais, por todo apoio, incentivo e amor no decorrer desta trajetória. Agradeço por sua confiança, por todos ensinamentos e pela incansável dedicação.

A minha namorada Júlia, que sempre me ajudou nos momentos difíceis, esteve ao meu lado durante este trabalho e pela compreensão em todos os períodos.

Aos meus amigos pelo companheirismo e disponibilidade para me auxiliar em vários momentos.

A minha orientadora Maria Adriana, pela confiança e pelo grande auxílio no desenvolvimento de cada etapa deste projeto.

A todos os professores que contribuíram com minha graduação.

Resumo

Este projeto tem como principal objetivo a implementação de um aplicativo móvel para a coleta de dados sobre o impacto da hospitalização infantil. O aplicativo deve permitir que crianças e adolescentes em tratamento de saúde e seus responsáveis possam responder, de forma interativa, a uma série de questões que viabilizam a avaliação do enfrentamento da hospitalização, baseada na escala COPHAT (*Coping* da Hospitalização, Adoecimento e Tratamento) elaborada para crianças e adolescentes internadas e seus respectivos pais/ou responsáveis num Hospital Geral. Atualmente a coleta de dados é realizada através de questionários em folhas de papel. Este trabalho propõe a implementação de um aplicativo em plataforma Android para coleta e armazenamento dos dados em banco de dados e a posterior organização e exportação dos dados em planilha eletrônica.

Palavras-chave: Aplicativo, Plataforma Android, Enfrentamento de hospitalização infantil.

Lista de ilustrações

Figura 1 – Exemplo de quadro do Kanban	16
Figura 2 – Primeira Etapa	25
Figura 3 – Segunda Etapa	26
Figura 4 – Terceira Etapa	27
Figura 5 – Quarta Etapa	27
Figura 6 – Esquema do Banco de Dados	29
Figura 7 – Tela inicial	30
Figura 8 – Tela de registro do paciente	31
Figura 9 – Tela de registro dos pais	32
Figura 10 – Tela de registro do diagnóstico	33
Figura 11 – Tela de registro da escolaridade	34
Figura 12 – Tela de registro da cidade	35
Figura 13 – Tela de erro no registro	36
Figura 14 – Tela de escolher a opção de questionário	37
Figura 15 – Tela de pergunta	38
Figura 16 – Tela de sub-pergunta	39
Figura 17 – Tela de erro ao não selecionar resposta	40
Figura 18 – Tela Final	41
Figura 19 – Planilha de Respondentes	43
Figura 20 – Planilha de Perguntas	44
Figura 21 – Síntese dos resultados do teste de usabilidade.	46

Lista de abreviaturas e siglas

CMS	Content Manager System
COPHAT	<i>Coping</i> da Hospitalização, Adoecimento e Tratamento
COPHAT-CA	<i>Coping</i> da Hospitalização, Adoecimento e Tratamento para Crianças ou Adolescentes
COPHAT-P	<i>Coping</i> da Hospitalização, Adoecimento e Tratamento para Pais ou Responsáveis
IDE	Integrated Development Environment
TOC	Theory of Constraints
UX	User Experience
UI	User Interface
XML	Extensible Markup Language

Sumário

1	INTRODUÇÃO	10
1.1	Objetivos	10
1.2	Motivação	11
1.3	Método	12
1.4	Organização do Trabalho	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Metodologia Ágil de Kanban	14
2.2	Kotlin	16
2.2.1	Kotlin vs Java	17
2.2.2	Desvantagens	19
2.3	Room Persistence Library	19
2.4	User Experience	20
2.4.1	O que é o <i>User Experience</i> ?	20
2.4.2	Por que é tão importante trazer os usuários para o centro do projeto?	21
2.4.3	Diferenças e semelhanças entre UX, UI e usabilidade	21
2.5	Testes de Usabilidade	21
2.5.1	O que é medido?	22
2.5.2	Envolvidos	22
2.5.3	Preparação Prévia	23
2.5.4	Think-Aloud	23
3	DESENVOLVIMENTO	24
3.1	Metodologia Kanban	24
3.1.1	Início do Projeto	24
3.1.2	Primeira Etapa	25
3.1.3	Segunda Etapa	26
3.1.4	Terceira Etapa	26
3.1.5	Quarta Etapa	26
3.2	Modelo do Banco de Dados	28
3.3	Fluxo de Telas	30
3.3.1	Tela inicial	30
3.3.2	Fluxo de cadastro	31
3.3.3	Questionário	35
3.3.4	Finalização	41
3.4	Testes de usabilidade	45

3.4.1	Preparação do Teste	45
3.4.2	Resultados	46
4	CONCLUSÃO	47
4.1	Considerações Finais	47
4.2	Contribuições	47
4.3	Trabalhos Futuros	47
	REFERÊNCIAS	48
	APÊNDICES	50
	APÊNDICE A – CÓDIGO FONTE DA CRIAÇÃO DO BANCO DE DADOS UTILIZANDO ROOM	51
	APÊNDICE B – EXEMPLO DE UMA ENTIDADE DO BANCO DE DADOS	52
	APÊNDICE C – CÓDIGO FONTE DA EXPORTAÇÃO DA PLANILHA	53
	ANEXOS	59
	ANEXO A – FICHA CADASTRAL DO PACIENTE	60
	ANEXO B – QUESTIONÁRIO PARA OS PACIENTES	62
	ANEXO C – QUESTIONÁRIO PARA OS PAIS OU RESPONSÁVEIS PELO PACIENTE	65

1 Introdução

A Psicologia Pediátrica engloba o domínio interdisciplinar relativo ao funcionamento e desenvolvimento físico, cognitivo, social, emocional e da relação com a saúde e a doença em crianças, adolescentes e seus familiares. No Brasil, [Dias, Baptista e Baptista \(2003\)](#) observaram que crianças acometidas com doenças crônicas, infecciosas, neurológicas e acidentes tendem a apresentar transtornos comportamentais e/ou psicológicos após o período de hospitalização de cinco dias. Nesse contexto, o estudo das estratégias de enfrentamento (*coping*) utilizadas pelas crianças se faz importante, para ajudá-las a lidar melhor com essa situação, que impõe ansiedade e estresse ([MORAES; ENUMO, 2008](#)).

No intuito de substituir os questionários em papel frequentemente utilizados para o estudo do enfrentamento hospitalar em crianças e adolescentes, propõe-se neste trabalho a construção de um aplicativo que possibilite a apresentação das questões e a coleta das respectivas respostas utilizando-se dispositivos móveis. Especificamente, pretende-se definir um ambiente gráfico e intuitivo com personagens simples, além da utilização de recursos gráficos na apresentação do questionário a ser aplicado às crianças ou adolescentes e seus pais ou responsáveis. Desta forma espera-se obter um maior envolvimento do paciente e do seu acompanhante com as questões apresentadas.

Com o crescimento da tecnologia e da inclusão digital, um aplicativo móvel possibilita melhor experiência para uso de recursos e interface dos dispositivos, otimizando a navegação e a agilidade das ações. Além disso possui a disponibilidade do uso *offline*, ou seja, muitos aplicativos armazenam informações que possibilitam navegação mesmo sem acesso à Internet. Portanto a elaboração da coleta de dados por meio de um *app* (aplicativo) é viável no ambiente proposto ([TECHAMERICA, 2011](#)). Ademais, do ponto de vista dos profissionais da psicologia pediátrica, a tabulação e a organização dos dados coletados, tanto do paciente quanto do seu responsável, permite uma melhor avaliação dos efeitos sofridos na hospitalização e, conseqüentemente, mais agilidade na análise e na identificação de formas de prevenir danos emocionais e/ou comportamentais gerados pela situação da hospitalização.

1.1 Objetivos

O objetivo principal desta proposta é o desenvolvimento de um aplicativo, para dispositivos móveis com sistema Android, com a finalidade de automatizar os questionários de enfrentamento de hospitalização em crianças e adolescentes (COPHAT - *Coping* da Hospitalização, Adoecimento e Tratamento) desenvolvidos por [Amaral e Enumo \(2017\)](#). O aplicativo deve proporcionar uma maior interação entre os respondentes e as questões

colocadas, além de proporcionar uma coleta mais eficiente das respostas. Ainda, o armazenamento das questões e das respostas coletadas permite que os dados sejam tabulados e organizados em planilhas eletrônicas. A geração das planilhas é uma das finalidades do aplicativo, pois estas são o mecanismo mais frequentemente utilizado por profissionais da psicologia para análise dos resultados, seja para a definição de diagnósticos e também para o estudo de formas de prevenção de danos emocionais e ou comportamentais que podem ser ocasionados pela hospitalização.

Atualmente com o avanço tecnológico, o que espera-se de um aplicativo, é tornar a vida das pessoas mais prática, ou seja, uma tarefa que leva muito tempo para ser executada, um aplicativo pode ser a solução para aumentar a produtividade e eficiência. Então, o objetivo principal deste trabalho é substituir os questionários de papel por um aplicativo com uma interface gráfica simples e intuitiva. Além disso ao final, é possível gerar uma planilha com os dados armazenados no banco de dados do smartphone com apenas um clique.

1.2 Motivação

O *coping* pode ser definido como um conjunto de esforços - comportamentais e cognitivos - utilizado com o objetivo de lidar com situações específicas, que surgem em situações de estresse e sobrecarregam os recursos pessoais (FOLKMAN; LAZARUS, 1980 apud MORAES; ENUMO, 2008). Os métodos de *coping* das crianças e adolescentes são mais específicos e estão de acordo com o desenvolvimento das habilidades cognitivas, sociais e regulação da emoção. O modelo de *coping* desenvolvido por Amaral e Enumo (2017) propõe as escalas COPHAT considerando quatro conceitos principais:

- (a) é um processo entre indivíduo e ambiente;
- (b) sua funcionalidade é administrar ao invés de controlar a situação aversiva;
- (c) seus processos implicam como a situação é percebida e interpretada pelo indivíduo;
- (d) compõe-se de uma mobilização de esforços cognitivos e comportamentais para administrar (reduzir, minimizar ou tolerar) os sentimentos que surgem a partir da interação com o ambiente hospitalar.

A escala COPHAT-CA foi desenvolvida para as crianças e adolescentes e a COPHAT-P para os pais e responsáveis. Em ambas, são considerados 34 itens correspondentes a 5 categorias:

1. Aspectos ligados a compreensão da doença e tratamento: itens que descrevem o grau de entendimento da doença e do tratamento;

2. Aspectos ligados a hospitalização: itens que retratam a hospitalização propriamente dita e suas formas de enfrentá-la;
3. Aspectos correspondentes à realização e êxito do tratamento: itens que representam como o tratamento é enfrentado, via pensamentos, sentimentos ou comportamentos;
4. Aspectos associados aos efeitos colaterais do tratamento: itens que descrevem consequências do tratamento;
5. Aspectos relacionados à expectativa do retorno às atividades escolares: itens que explicitem pensamentos ou sentimentos frente ao retorno a escola após o tratamento.

A análise das duas escalas, COPHAT-CA e COPHAT-P permitem também conhecer se filhos e pais estão percebendo a hospitalização e o tratamento da mesma forma. Ao responderem a escala, tanto crianças ou adolescentes quanto seus pais ou responsáveis estarão reconhecendo seus comportamentos de *coping* diante da internação hospitalar e podem ser estimulados emitir respostas positivas.

Neste contexto, o presente trabalho colabora com as escalas em questão, propiciando um ambiente interativo para que os profissionais de psicologia pediátrica possam coletar os dados de forma mais rápida e eficiente, e possam em seguida planejar intervenções mais pontuais e efetivas direcionadas às crianças e adolescentes internados e seus responsáveis.

1.3 Método

Para a construção do aplicativo, foi utilizado um método de desenvolvimento ágil, o Kanban ([SUGIMORI et al., 1977](#)), que consiste de um quadro de progresso com três colunas: “To Do”, “Doing” e “Done”. Nele, todas as tarefas começam em “To Do” e quando uma tarefa está em progresso, deve ser movida para “Doing” e logo após ser concluída, deve ser movida para “Done”. Esta metodologia Kanban, facilita a visualização do desenvolvimento do projeto.

Para a organização das tarefas, foi utilizada parte da metodologia Scrum ([TAKEUCHI; NONAKA, 1986](#)), que define a distribuição de atividades em semanas de desenvolvimento, isto é, em “sprints” (termo utilizado para divisão de tarefas ao longo do projeto). As perguntas e respostas serão armazenadas em um banco de dados nativo do dispositivo, mais precisamente o Room (derivado do SQLite) do Android. Para a linguagem de programação, a principal utilizada no desenvolvimento será a linguagem Kotlin e a interface gráfica será feita em XML através do ambiente de desenvolvimento integrado (IDE) Android Studio.

Espera-se como resultado do projeto um aplicativo de simples entendimento, ligado a uma interface aprimorada que leve o usuário a ter uma experiência diferente em relação

à responder os questionários em papel.

1.4 Organização do Trabalho

No Capítulo 2 está descrita a Fundamentação teórica utilizada neste trabalho. Tais como: Metodologia ágil de Kanban, Kotlin, Room Persistence Library, User Experience e Testes de Usabilidade. No Capítulo 3 é apresentado o desenvolvimento do projeto atrelado à tudo que foi exposto no Capítulo 2. Por fim, o Capítulo 4 expõe as conclusões obtidas do trabalho desenvolvido e apresenta perspectivas para trabalhos futuros.

2 Fundamentação Teórica

Este capítulo apresenta um levantamento bibliográfico e teórico sobre metodologia ágil de Kanban, Kotlin, Room Persistence Library, User Experience e testes de usabilidade. Estes temas são relevantes e formam a base para o desenvolvimento do aplicativo proposto no presente trabalho.

2.1 Metodologia Ágil de Kanban

O Kanban é uma metodologia usada para integrar o desenvolvimento ágil à uma produção de software, onde alguns requisitos são necessários, tais como: (a) comunicação de capacidade em tempo real e (b) transparência total de trabalho. As tarefas criadas são representadas por meio de um quadro, onde fica mais fácil a visualização do progresso e permite que os integrantes da equipe vejam qual a situação de cada parte do trabalho a qualquer instante.

Este método, geralmente é representado por um quadro de progresso, onde é possível retratá-lo por algum software ou até mesmo por uma folha de papel, onde as tarefas são escritas em cartões e correspondem à um fluxo pré-estabelecido de etapas. Na medida que o projeto é desenvolvido, os cartões mudam de etapa e toda vez que uma tarefa é identificada, um novo cartão é escrito.

Segundo [Godoy \(2014\)](#), o Kanban implementa conceitos da Teoria das Restrições (TOC – Theory of Constraints) através de um modelo de sistema “puxado”, procurando identificar e administrar restrições que limitam o desempenho do sistema. Em poucas palavras, o modelo de sistema “puxado” do Kanban relaciona a capacidade ao número de cartões ou tickets que podem ser colocados em circulação, através de um mecanismo de sinalização, onde cada novo trabalho só pode ser iniciado quando um cartão estiver disponível.

O Kanban é uma forma simples de visualização do desenvolvimento do projeto, onde os cartões colocados em um quadro ou mural, dá a possibilidade da equipe verificar rapidamente a situação do trabalho. Além disso, uma das principais vantagens da utilização desse método é a maior assimilação das informações por parte da complexidade do trabalho, pois geralmente estão sob forma de gráficos, símbolos e diagramas. Outra vantagem é a transparência, simplesmente pelo fato de todas as informações necessárias estarem expostas a todos da equipe, permitindo maior interatividade, melhor comunicação e mais agilidade na entrega do projeto.

A função do quadro do Kanban é assegurar que o trabalho da equipe seja visua-

lizado, induzindo uma maior padronização e que todos os impedimentos e dependências sejam identificados e resolvidos. Representado pela Figura 1, um quadro básico do Kanban exibe os principais estágios do trabalho, sendo divididos em três: “To Do”, “Doing” e “Done”. Por outro lado, nem sempre estes estágios atendem a todos os tipos de projeto, onde dependendo do tamanho, da estrutura e dos objetivos da equipe, a sequência de trabalho pode ser reestruturada para atender à exclusividade de cada equipe.

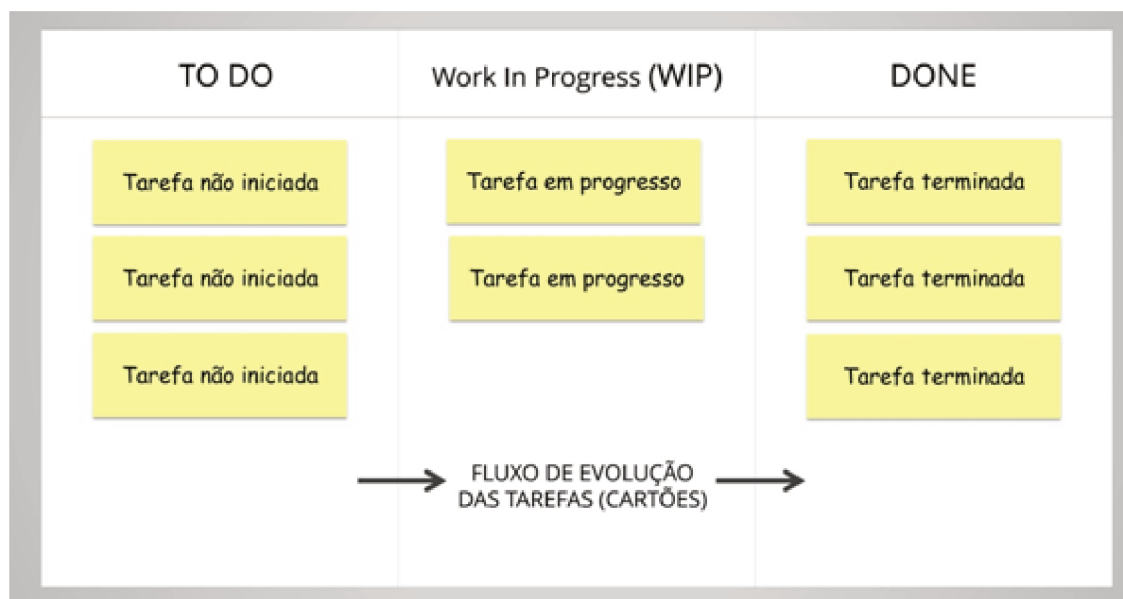


Figura 1 – Exemplo de quadro do Kanban

As metodologias ágeis estão cada vez mais difundidas no mercado de trabalho, onde, segundo [Camargo \(2018\)](#), empresas implementam, na maioria das vezes, a metodologia Scrum incorporada ao quadro do Kanban, pelo qual juntam-se as duas técnicas de desenvolvimento para agilizar as entregas propostas pelo cliente. Então, pelo fato desde projeto não possuir uma equipe, foi escolhido, para o desenvolvimento, a utilização da metodologia ágil Kanban ao invés da Scrum.

2.2 Kotlin

O Kotlin é uma linguagem estaticamente tipada, desenvolvida pela JetBrains, cuja sintaxe é mais expressiva e concisa do que a do Java ([CORDEIRO, 2018](#)), e oferece recursos como expressões lambda, sobrecarga de operadores, templates de strings e muito mais. Como o Java e o Kotlin são altamente interoperáveis, elas podem ser usadas juntas no mesmo projeto. Assim como Java, Kotlin é uma linguagem de programação baseada em uma máquina virtual JVM.

De acordo com [Developers \(2018\)](#), o Kotlin é uma linguagem moderna e expressiva, além de ser intuitiva para desenvolvedores Java. Grande parte da linguagem é semelhante e as diferenças nos conceitos básicos podem ser aprendidas rapidamente. Para um programador Java experiente, aprender o Kotlin se torna uma tarefa muito fácil, caso contrário, será como aprender qualquer outra linguagem. Algumas características interessantes da linguagem são:

- **Expressiva:** É possível escrever mais com menos linhas de código.

- **Segura:** O Kotlin tem seguro contra valores nulos, onde são situações de valores nulos em tempo de compilação, para evitar exceções em tempo de execução.
- **Multi paradigma:** O Kotlin é uma linguagem orientada a objetos, não uma linguagem funcional pura.

No entanto, como muitas outras linguagens modernas, ela usa muitos conceitos funcionais, como expressões lambda, para resolver alguns problemas de uma maneira muito mais fácil.

- **Funções de Extensão:** Isso significa que podemos estender qualquer classe e adicionar novos recursos, mesmo se não tivermos acesso ao código-fonte.
- **Interoperável:** É possível continuar usando a maioria das bibliotecas e códigos escritos em Java, porque a interoperabilidade entre elas é excelente. É até permitido criar projetos mistos, com os arquivos Kotlin e Java coexistindo.

2.2.1 Kotlin vs Java

Na plataforma Android, a linguagem de programação Java ainda é a mais utilizado para desenvolver aplicativos, mas este cenário vem mudando com a chegada do Kotlin, onde programadores aceitam o desafio de aprender uma nova linguagem para tornar o desenvolvimento do aplicativo mais eficiente. O Kotlin adiciona novas capacidades importantes aos projetos Android que ainda não são possíveis na tecnologia Java.

1. **Curva de Aprendizado:** Para um programador Java experiente, aprender o Kotlin não é uma tarefa difícil, onde é possível aprendê-lo em apenas algumas horas. Além disso, a [JetBrains \(2016\)](#) possui uma documentação oficial completa e muito bem feita.
2. **Suporte:** Segundo [Cordeiro \(2018\)](#), para desenvolvimento Android utilizando o Android Studio, existe um ótimo suporte para o uso do Kotlin. Além disso, ele interage perfeitamente com a linguagem Java podendo-se utilizar arquivos das duas linguagens no mesmo projeto. Assim, o uso de bibliotecas Java em projetos Kotlin ou vice-versa simplesmente funciona. Pode-se facilmente incluir qualquer biblioteca escrita em Kotlin dentro do código Java.
3. **Null Safety (Proteção contra nulo):** Como no Java existem muitos problemas com referências nulas, o Kotlin possui uma proteção para que isso não aconteça. O compilador não permite que uma variável seja utilizada sem que ela seja inicializada ou que exista a possibilidade dela ser nula. Mas, se for necessário a declaração de uma variável de valor nulo, deve-se declarar o tipo como “anulável”, adicionando-se um ponto de interrogação após o tipo. Por exemplo:

```
var nonNullable: String = "My string"
var nullable: String?
```

4. **Funções Estendidas:** Essa função oferece comportamentos a uma classe sem estendê-la diretamente. Para isso, é necessário prefixar seu nome com o tipo de classe que será estendido.

```
StringExtensions.kt
fun String.novaFuncao(): String {
    // implementacao
}
```

5. **Lambdas:** Uma função lambda recebe funções como parâmetros ou retorna uma função. Elas podem ser armazenadas em variáveis para uso subsequente, que são criadas incorporadas a outras funções ou passadas como parâmetros.

```
// Criando a funcao
fun main(args: Array<String>) {
    val sum = { x: Int, y: Int -> x + y }
}
```

6. **Classes de Dados (Data Classes):** Em Java são necessárias as implementações dos métodos get/set de cada atributo para que seja possível armazenar dados. No Kotlin este recurso traz uma grande economia de tempo, uma vez que o compilador irá gerar todos os getters e setters.

```
data class User(var name: String, var age: Int)
```

7. **Imutabilidade:** Segundo [Cordeiro \(2018\)](#), uma grande preocupação que os desenvolvedores precisam ter ao desenvolver aplicativos multithread é a administração do estado. Se uma variável é mutável, ela pode ser alterada por qualquer thread que possa acessá-la. Ou seja, se for possível alterar os dados por várias fontes, será necessário implementar manualmente a sincronização, o que evita a perda de dados, mas aumenta a complexidade do código e o tempo de execução. Se os dados nunca podem ser alterados, eles podem ser acessados por várias threads sem problema, uma vez que são imutáveis.

No Kotlin, é possível declarar variáveis com as palavras-chave **var** e **val**. Onde declara-se uma variável que pode ser alterada e uma variável que uma vez atribuído um valor nunca pode mudar respectivamente. Essa característica dá ao desenvolve-

dor e ao compilador confiança de que a variável não pode ser alterada. Já em Java temos a funcionalidade semelhante utilizando a palavra-chave **final**.

O Kotlin roda em cima da Máquina Virtual do Java e facilita muitas coisas para o desenvolvedor Android, então basicamente, ele ajuda a escrever códigos mais limpos e de forma mais fácil (CORDEIRO, 2018). Depois disso, uma vez que o código esteja sendo executado, ele cria um arquivo de classe semelhante ao que é obtido quando o arquivo `.java` é executado.

Assim, ao final das contas, o Kotlin é um recurso complementar e um bônus para que o desenvolvedor Android possa codificar de forma eficiente, com melhor desempenho e de forma amigável. O suporte oficial à linguagem Kotlin está disponível no Android Studio a partir da versão 3.0. O Kotlin vem evoluindo cada vez mais para a plataforma Android (ZANELLA, 2017), e já é utilizada por grandes empresas como padrão para desenvolvimento dos seus aplicativos.

2.2.2 Desvantagens

Embora desenvolver aplicativos Android utilizando o Kotlin seja, em muitos casos uma alternativa melhor que Java, existem alguns desafios (CORDEIRO, 2018). Por exemplo, um aplicativo construído em Kotlin provavelmente resultará em um tamanho de pacote de arquivos maior do que um construído puramente em Java. Isso ocorre porque o Kotlin tem sua própria biblioteca que é adicionada ao aplicativo. Além disso, o tempo de compilação de códigos Kotlin é um pouco mais lento usando-se o Gradle ¹ (sistema de automatização de *builds*).

2.3 Room Persistence Library

Room (LLC, 2018) é uma das bibliotecas existentes dentro do conjunto *Android JetPack* apresentado durante o Google I/O de 2018. Ela auxilia os desenvolvedores criando uma abstração das camadas de banco de dados (SQLite). Suas camadas são representadas por três componentes:

- **Entity:** são entidades responsáveis por mapear as tabelas.
- **DAO (Data Access Object):** são as interfaces utilizadas para acessar os dados armazenados no banco, onde é possível criar consultas personalizadas para as operações de select, insert, delete e update.

¹ <https://gradle.org/>

- **Database:** é a representação da classe abstrata do Banco de Dados. Ela recebe uma anotação que identifica as Entities, os DAO's e seus Converters. É também responsável por fazer o gerenciamento do banco de dados.

A biblioteca Room possui alguns pontos interessantes que podem ser destacados: o primeiro deles é o fato da criação de uma base de dados ser simples e de fácil entendimento. Ainda, a implementação utiliza o conceito de *annotations*, que definem entidades, consultas e definições específicas do banco de dados. Algumas classes podem ser geradas pelo próprio componente, sendo este responsável por conter a maior parte do código e algumas lógicas que garantem o bom funcionamento da base de dados. Podem ser gerados modelos (*schemas*) do banco pra facilitar a verificação do estado da base de dados: se alguma alteração foi feita, e como se encontra (num dado instante) a estrutura das tabelas.

O uso da Room possibilita a criação de diferentes bases de dados que podem funcionar de forma independente. Este fato é de grande importância para projetos que se baseiam em vários módulos que são independentes uns dos outros. No presente trabalho, utilizou-se apenas uma base de dados.

As entidades suportam todos os tipos primitivos para tipagem dos campos, ademais, existe a possibilidade de criação de *converters*, para converter classes complexas em tipos primitivos suportados pelo os quais o banco suporta. Também é possível relacionar campos a outras entidades, criando assim relacionamento entre as tabelas.

O fato de ser uma biblioteca do Google traz um dos pontos mais relevantes: a possibilidade de que a biblioteca será mantida e atualizada frequentemente, além de fazer parte dos *Android Architecture Components* que devem se tornar as principais estruturas adotadas em projetos Android. A biblioteca Room oferece uma boa documentação, disponível em várias línguas, além de um rápido retorno na resolução de bugs.

2.4 User Experience

2.4.1 O que é o *User Experience*?

User Experience (UX), ou experiência de usuário, é a arte de trazer o usuário para o centro do projeto, sabendo conciliar suas necessidades, anseios e satisfação com os objetivos do produto. Em outras palavras, User Experience é um método de engenharia e design que cria sistemas que funcionam melhor para o usuário pretendido. Os usuários devem ser incluídos no processo de design através de pesquisas do usuário e testes de usabilidade (GARRETT, 2010).

A experiência do usuário diante um produto vai muito além de dar aos clientes

o que eles falam que querem, ou apenas cumprir uma lista de tarefas determinadas. A verdadeira experiência do usuário é atender as necessidades do cliente incluindo simplicidade e elegância à um produto que cria desejo e que seja fáceis de usar. Para que isso seja adquirido, é necessário a fusão de diversas áreas para chegar a uma experiência de alta qualidade.

2.4.2 Por que é tão importante trazer os usuários para o centro do projeto?

A análise de User Experience permite identificar os diversos modelos de pensamento dos usuários de um projeto. Dessa forma, é possível estruturar e projetar produtos que se adequam à estes usuários e atendem suas necessidades. Para entender o que faz sentido para um usuário é necessário conversar, ouvir e observar. Mas, entender o usuário não dá a certeza de que o projeto será um sucesso, mas aumenta significativamente as chances

2.4.3 Diferenças e semelhanças entre UX, UI e usabilidade

É importante distinguir User Experience de User Interface, embora a interface do usuário seja, obviamente, uma parte extremamente importante do design. Contudo, muitas pessoas confundem UX e UI pensando que são a mesma coisa ou invertendo os papéis, mas na verdade são diferentes, onde UX representa tudo que o usuário está sentindo ao utilizar o produto, e UI representa, como o próprio nome diz, a interface do usuário, ou, em outras palavras, o que o usuário vê na tela do dispositivo. Então para que um produto chegue em uma alta qualidade, não basta apenas a interface do usuário estar perfeita se a experiência estiver ruim.

2.5 Testes de Usabilidade

O Teste de usabilidade é uma técnica de pesquisa utilizada para avaliar um produto ou serviço (CAELUM, 2018). Os testes são realizados com usuários representativos do público-alvo. Cada participante tenta realizar tarefas típicas enquanto o analista observa, ouve e anota.

O objetivo é observar usuários reais usando o produto para descobrir problemas e pontos de melhorias. O produto, que pode ser um site, uma aplicação web, um produto físico, não precisa estar completamente desenvolvido. Protótipos são vastamente utilizados em testes de usabilidade para validação do que está sendo feito.

2.5.1 O que é medido?

Segundo [Caelum \(2018\)](#) testes de usabilidade geralmente envolvem medir quão bem os participantes respondem a quatro áreas:

- **Desempenho:** Quanto tempo e quantos passos são necessários para que a pessoa complete tarefas básicas?
- **Precisão:** Quantos erros a pessoa cometeu? E eles foram fatais ou a pessoa conseguiu se recuperar com as informações recebidas pelo sistema?
- **Lembrança:** O quanto a pessoa se lembra mais tarde ou depois de períodos sem usar?
- **Resposta emocional:** Como a pessoa se sentiu depois de completar a tarefa? A pessoa estava confiante ou estressada? Ela recomendaria o produto a um amigo?

2.5.2 Envolvidos

Um teste de usabilidade típico é composto, além do participante, por um moderador, que dá as instruções ao participante de como o teste será realizado e passa a ele as tarefas que deverão ser executadas; e o observador, normalmente o designer, ou no caso deste projeto o próprio desenvolvedor, envolvido no desenvolvimento do produto.

O ideal é manter o mínimo de pessoas como observador, para que o participante não se sinta desconfortável no momento do teste. Porém, em algumas situações é interessante que o gerente do produto ou alguém do time de desenvolvimento também participe como observador, pois dessa forma eles conseguem ter uma visão mais real do produto que estão desenvolvendo ([CAELUM, 2018](#)). Alguns pontos devem ser considerados durante um teste de usabilidade, tais como:

- **Local:** O teste de usabilidade pode ser feito tanto no próprio ambiente onde a pessoa utilizará (ou utiliza) o produto, como em um laboratório especial para isso. Ambos tem suas vantagens.
- **Moderador:** A introdução ao teste deve ser feita pelo moderador, explicando o objetivo do teste e deixando bem claro que o que está sendo testado é o produto e não o participante; que o papel do usuário naquele teste é de colaborador, ao ajudar a melhorar o produto. Depois da introdução, o moderador passa ao participante as tarefas que ele precisa realizar.

2.5.3 Preparação Prévia

Para que um teste seja bem eficaz no desenvolvimento do produto, alguns pontos devem ser estudados, tais como ([MCCRACKEN, 2016](#)):

- **Anotar observações:** Durante a realização do teste, assim como em qualquer pesquisa, o ideal é que o designer tenha um pouco de tempo entre um teste e outro para consolidar as informações já obtidas, pois corre-se o risco de esquecer detalhes importantes se deixar para fazer isso um tempo depois.
- **Ajuste do protótipo entre testes:** Também, dependendo do tempo entre um teste e outro, o protótipo já pode ser ajustado caso sejam percebidas falhas graves no desenho que esteja prejudicando os resultados dos testes.
- **Elaborar relatório:** Depois de concluídos os testes, o designer deve consolidar todas as informações obtidas, conversar com os envolvidos no desenvolvimento do produto e apresentar o relatório dos achados do teste, para juntos fazerem um plano de ação de tudo aquilo que precisa ser revisto e ajustado no design do produto.

2.5.4 Think-Aloud

“Thinking Aloud” é um protocolo verbal que pode ser utilizado em testes de usabilidade ([BOREN; RAMEY, 2000](#)). Este protocolo envolve os participantes, literalmente, pensando em voz alta enquanto estão executando um conjunto de tarefas, e são convidados a dizer o que vier à sua mente enquanto completam a tarefa. É possível incluir o que eles estão olhando, pensando, fazendo e sentindo. Isso dá aos observadores uma visão dos processos cognitivos do participante (em vez de apenas seu produto final), para tornar os processos mentais tão explícitos quanto possível durante o desempenho da tarefa ([WIKIPEDIA, 2017](#)).

O método de Think-aloud foi proposto por [Ericsson e Simon \(1993\)](#) e pode ser aplicado em cenários distintos: (i) o observador define tarefas específicas ao usuário, ou (ii) o usuário é livre para escolher sua tarefa. No primeiro caso, o observador pode se concentrar em problemas específicos e no segundo, o observador pode captar problemas que ocorrem naturalmente. Em ambos os casos, todas as verbalizações são transcritas para serem posteriormente analisadas. A tarefa principal do observador é anotar o que acontece, sendo que a aplicação pode também ser gravada em vídeo.

3 Desenvolvimento

Este capítulo apresenta como foi estruturado o desenvolvimento do projeto. Será mostrado cada etapa integrada à metodologia Kanban, além da distribuição das tarefas no formato de *sprints*, a estruturação do banco de dados, telas implementadas no aplicativo, modelo de planilha que contém os dados armazenados e por fim, informações sobre os testes de usabilidade realizados.

3.1 Metodologia Kanban

Como este projeto foi feito com base na metodologia ágil Kanban, foram separadas etapas de desenvolvimento até a finalização do aplicativo. Para facilitar a visualização, foi utilizado o Trello ([TRELLO, 2011](#)) (ferramenta para construção dos quadros do Kanban). Segue abaixo, a descrição das etapas, onde cada tarefa, engloba todas as funções da tela implementada ou do banco de dados.

3.1.1 Início do Projeto

No início do projeto, após feita uma análise de requisitos, foram definidas as seguintes tarefas:

1. Criação do modelo do Banco de Dados
2. Implementação do banco de dados através do Room
3. Teste da inicialização do Banco de Dados
4. Implementação da tela de Splash
5. Teste da tela de Splash adicionando elementos no Banco de Dados
6. Implementação da tela de cadastro do paciente
7. Teste da tela de cadastro do paciente
8. Implementação da tela de cadastro dos pais do paciente
9. Teste da tela de cadastro dos pais
10. Implementação da tela de cadastro de diagnóstico
11. Teste da tela de cadastro de diagnóstico
12. Implementação da tela de cadastro de escolaridade
13. Teste da tela de cadastro de escolaridade

14. Implementação da tela de cadastro de cidade
15. Teste da tela de cadastro de cidade
16. Implementação da tela de selecionar opção de questionário
17. Teste da tela de seleção de questionário
18. Implementação da tela genérica de questionário
19. Teste do Questionário
20. Implementação da tela final
21. Teste das ações da tela final
22. Implementação da exportação de planilha
23. Teste da planilha com validação dos dados
24. Integração com o banco de dados no final do questionário

3.1.2 Primeira Etapa

Na primeira etapa do desenvolvimento foram executadas as tarefas de criação do modelo do banco de dados, implementação do banco de dados utilizando a ferramenta Room (recomendada pelo Google), foram feitos testes de inicialização do banco através da biblioteca Android Debug Database (SHEKHAR, 2016), além disso foi implementada e testada a tela de Splash, onde os dados fixos do aplicativo são inseridos no banco de dados. A Figura 2 apresenta a definição das atividades da Primeira Etapa.

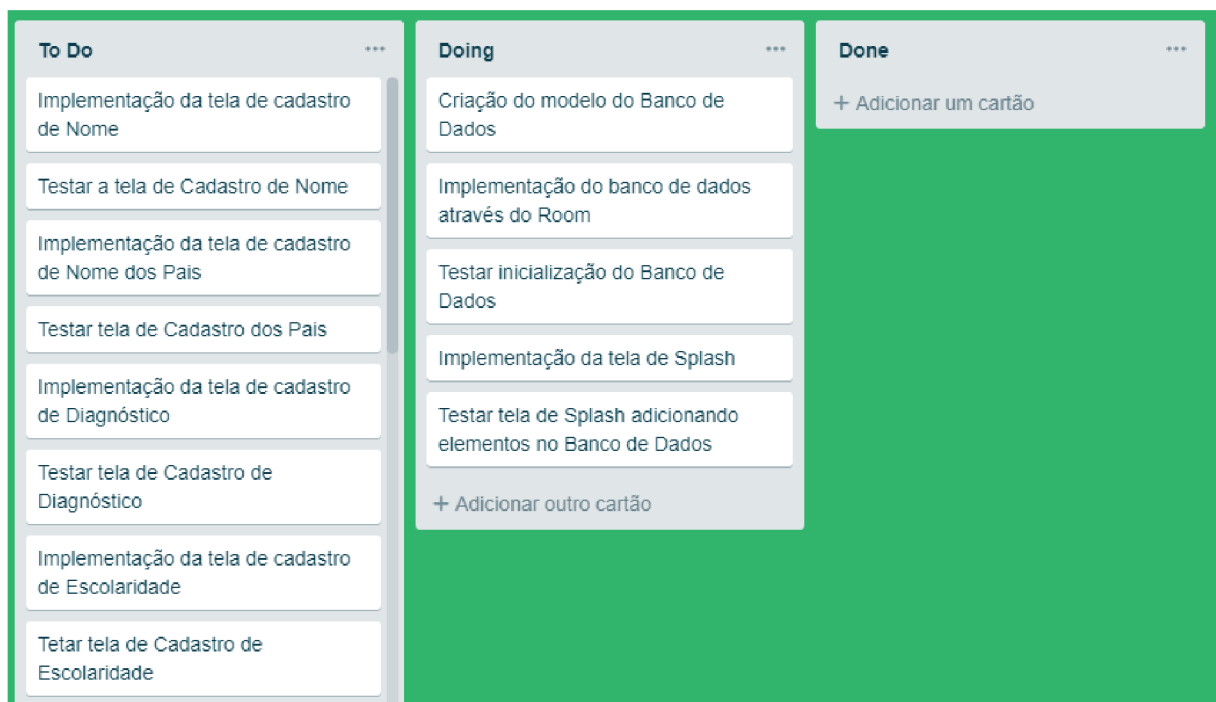


Figura 2 – Primeira Etapa

3.1.3 Segunda Etapa

Na segunda etapa, foi feito o fluxo de registro do paciente, onde foram implementados as telas de registro de nome, registro dos pais, do diagnóstico, da escolaridade e da cidade, juntamente com os testes devidamente executados validando os campos preenchidos. Ademais, foi implementada a tela de selecionar opção de questionário, onde o usuário pode escolher entre os questionário para as crianças ou para os pais. A figura 3 apresenta a definição das atividades da Segunda Etapa.



Figura 3 – Segunda Etapa

3.1.4 Terceira Etapa

Já na terceira etapa, as tarefas executadas foram: implementação do questionário, onde foi feito um layout genérico para perguntas e respostas, juntamente com o pop-up de sub-pergunta, e a tela final do aplicativo, pelo qual o usuário pode escolher entre recomençar, finalizar ou exportar a planilha com as informações contidas no banco de dados. Além de tudo, os testes foram devidamente realizados, validando cada comportamento do aplicativo. A figura 4 apresenta a definição das atividades da Terceira Etapa.

3.1.5 Quarta Etapa

Por fim, como última etapa do desenvolvimento, foi feito a implementação do código para gerar a planilha (citada na sub-seção anterior), onde é gerado um arquivo do

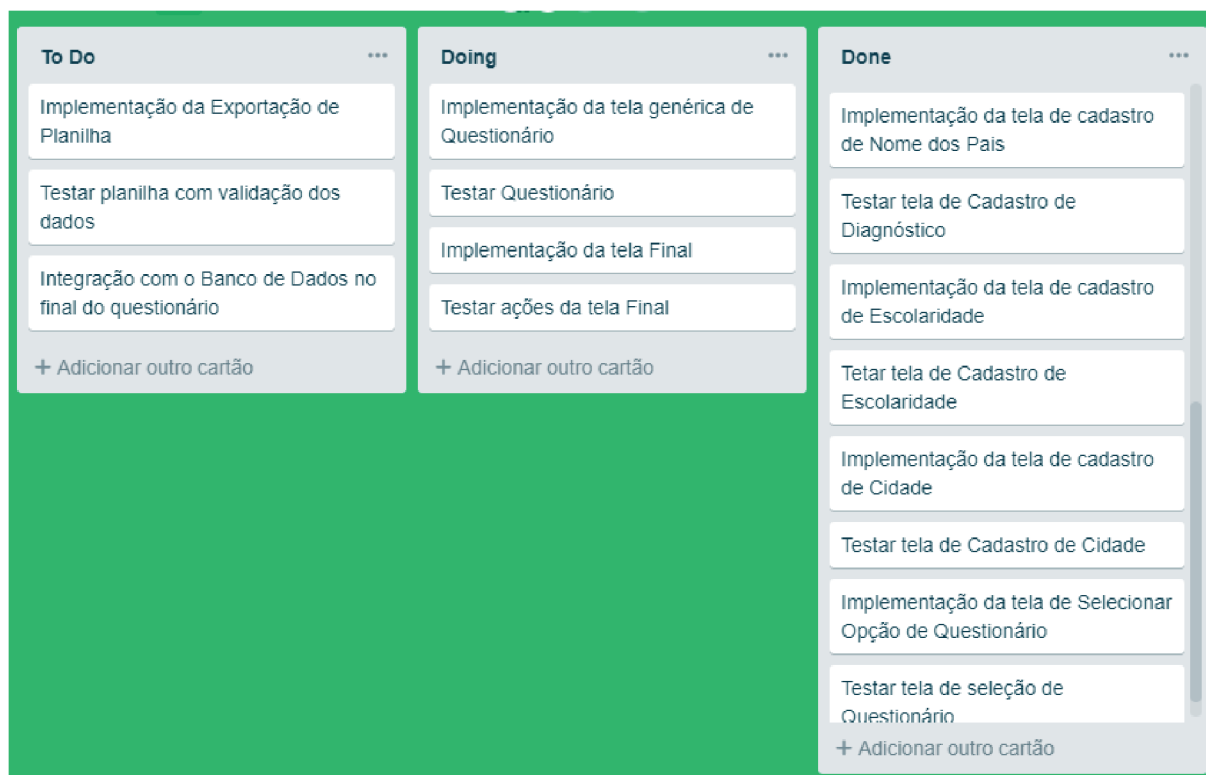


Figura 4 – Terceira Etapa

tipo .xls e guardado na memória interna do telefone. A figura 5 apresenta a definição das atividades da Quarta Etapa.

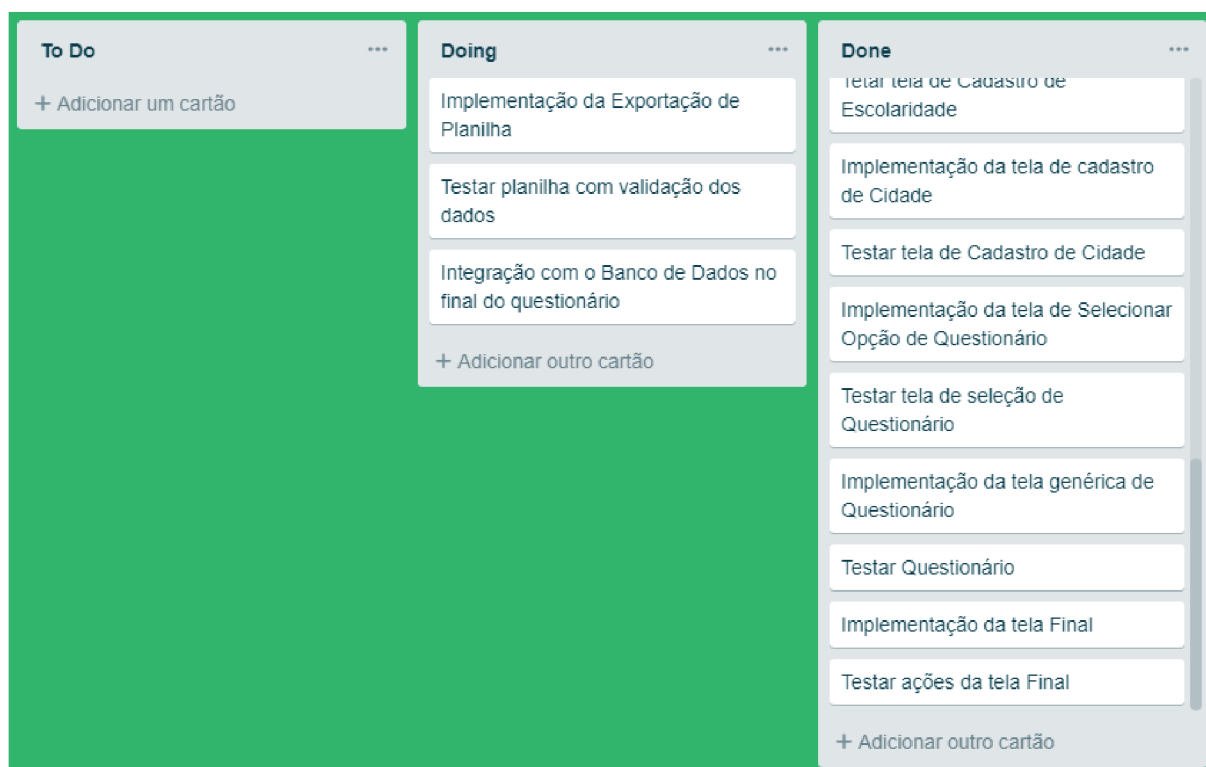


Figura 5 – Quarta Etapa

3.2 Modelo do Banco de Dados

O modelo de banco de dados foi estruturado em 10 tabelas, onde cada uma tem sua característica específica, segue as tabelas:

- Questionário - contém o id (chave primária) dos questionários respondidos.
- Seção - contém o id (chave primária), o nome da seção e o tipo.
- Pergunta - contém todas as perguntas do questionário, juntamente com seu id único e as chaves estrangeiras das tabelas Seção, Questionário e Escala.
- Sub-Pergunta - contém as sub-perguntas existentes, possui o nome e o id (chave primária).
- Sub-Resposta - contém as sub-repostas fixas, possui o id (chave-primária) e o nome.
- Escala - contém o id e o tipo de escala das respostas existentes.
- Valor Escala - contém as respostas existentes junto com a chave estrangeira do id da tabela Escala.
- Resposta - contém as informações variáveis, o que o usuário respondeu no questionário. Possui o id (chave primária), o título da resposta, a pergunta associada aquela resposta, a sub-pergunta e a sub-resposta se existir, a pontuação daquela resposta e o id da Avaliação como chave estrangeira.
- Avaliação - contém seu id (chave primária), a data que o questionário foi finalizado, o total de pontos relacionados às respostas, o id do Respondente e o id do questionário respondido.
- Respondente - contém todas as informações do paciente relacionado ao questionário.

Na página seguinte, segue o esquema do banco de dados desenhado utilizando a ferramenta SQLDBM, representado pela figura 6.

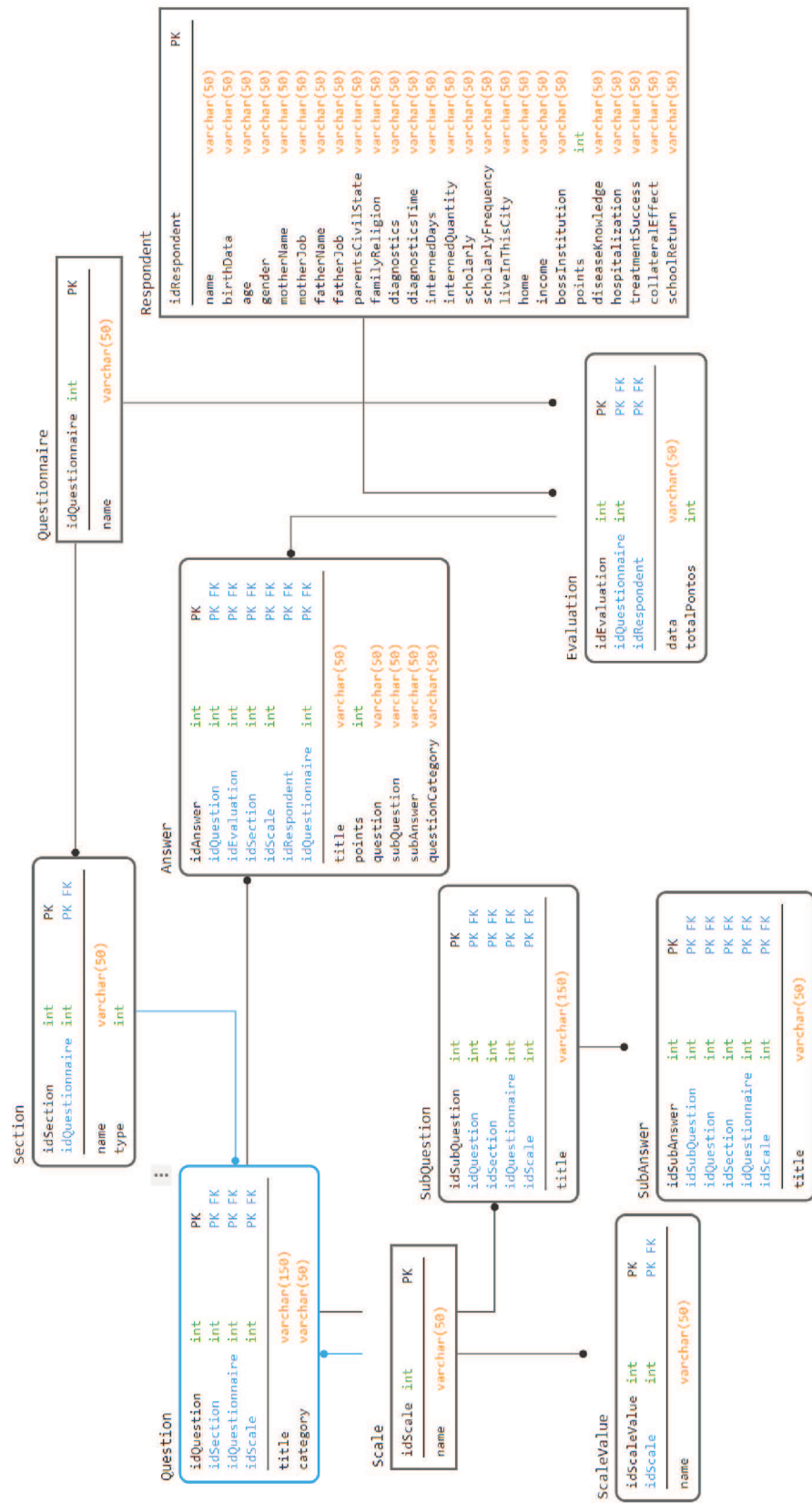


Figura 6 – Esquema do Banco de Dados

3.3 Fluxo de Telas

3.3.1 Tela inicial

Na tela inicial do aplicativo, representada pela figura 7, estão presentes a logo do aplicativo, o texto inicial e dois botões, onde o usuário pode executar as seguintes ações:

- Começar o cadastro - no botão COMEÇAR
- Exportar a planilha caso houver conteúdo no banco de dados - no botão no canto superior direito da tela (este botão só é mostrado quando há conteúdo no banco de dados)

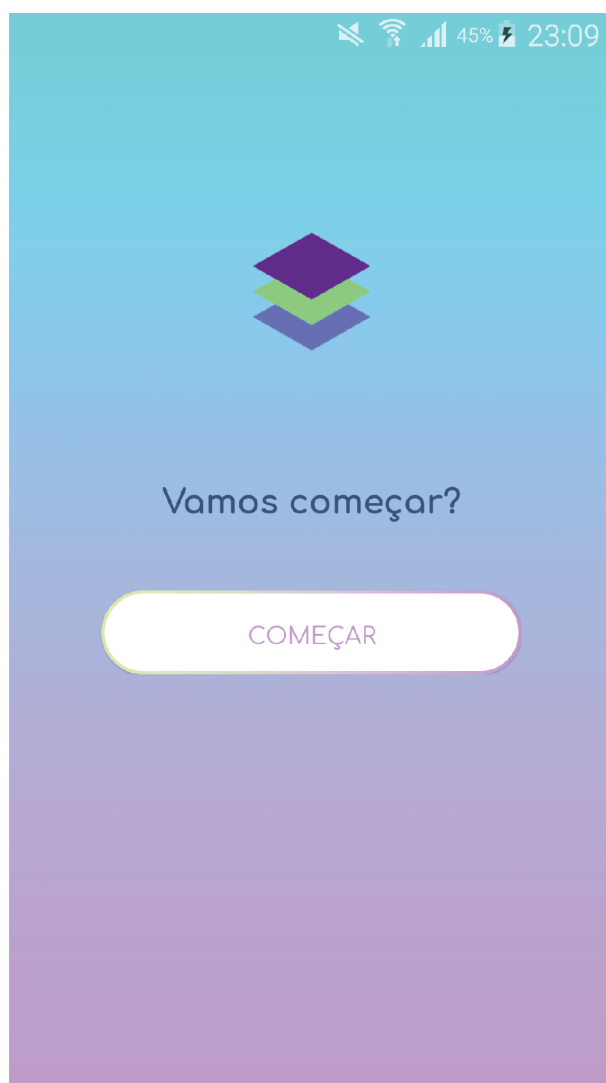


Figura 7 – Tela inicial

3.3.2 Fluxo de cadastro

Nas telas de cadastro, é necessário que o usuário preencha as informações corretamente do paciente que está sendo analisado. Existem validações para os campos digitados para manter os dados do paciente mais concisos e coniventes, evitando que o usuário digite qualquer coisa fora do normal.

Na primeira tela de registro, representada pela figura 8, o usuário deve preencher os seguintes dados relacionados ao paciente:

- Nome
- Data de aniversário
- Idade no dia da entrevista
- Gênero



Figura 8 – Tela de registro do paciente

Na segunda tela, representada pela figura 9, devem ser preenchidas as seguintes informações:

- Nome da mãe
- Profissão da mãe
- Nome do pai
- Profissão do pai
- Estado civil dos pais
- Religião da família

Qual o nome e profissão da mãe?

Nome da mãe

Profissão da mãe

Qual o nome e profissão do pai?

Nome do pai

Profissão do pai

Qual o estado civil dos pais?

☐ Casado

☐ Amasiado

☐ Divorciado/Separado

Figura 9 – Tela de registro dos pais

Já na terceira tela, representada pela figura 10, são esses dados:

- Diagnóstico
- Quanto tempo de diagnóstico
- Quantos dias internado
- Quantas internações

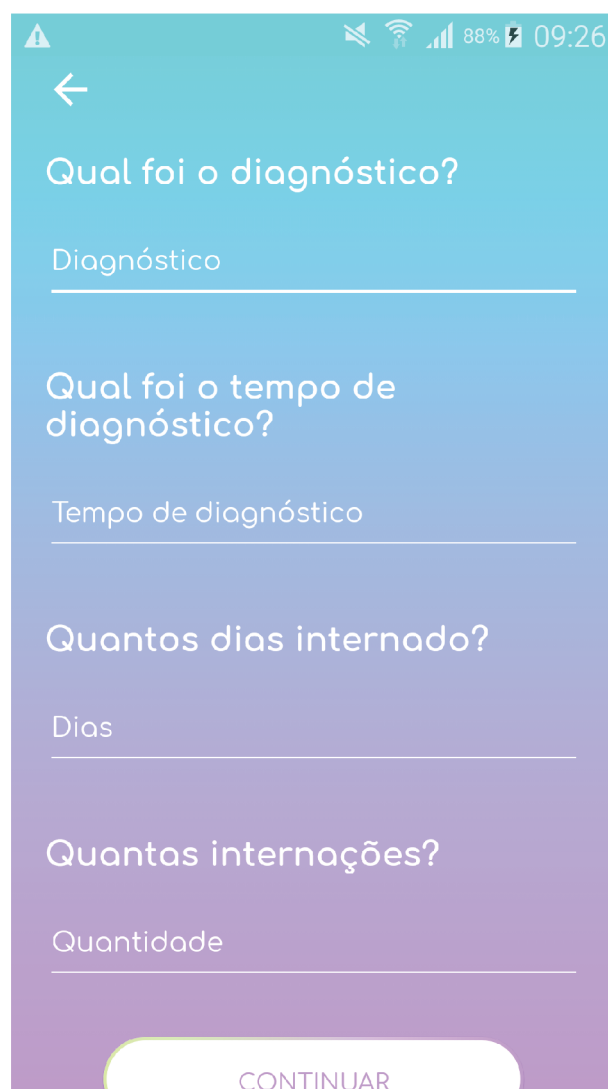


Figura 10 – Tela de registro do diagnóstico

Na quarta tela, representada pela figura 11, são essas informações:

- Escolaridade
- Se está frequentando a escola fora da internação

E por fim, na quinta e última tela de registro, representada pela figura 12, são essas:

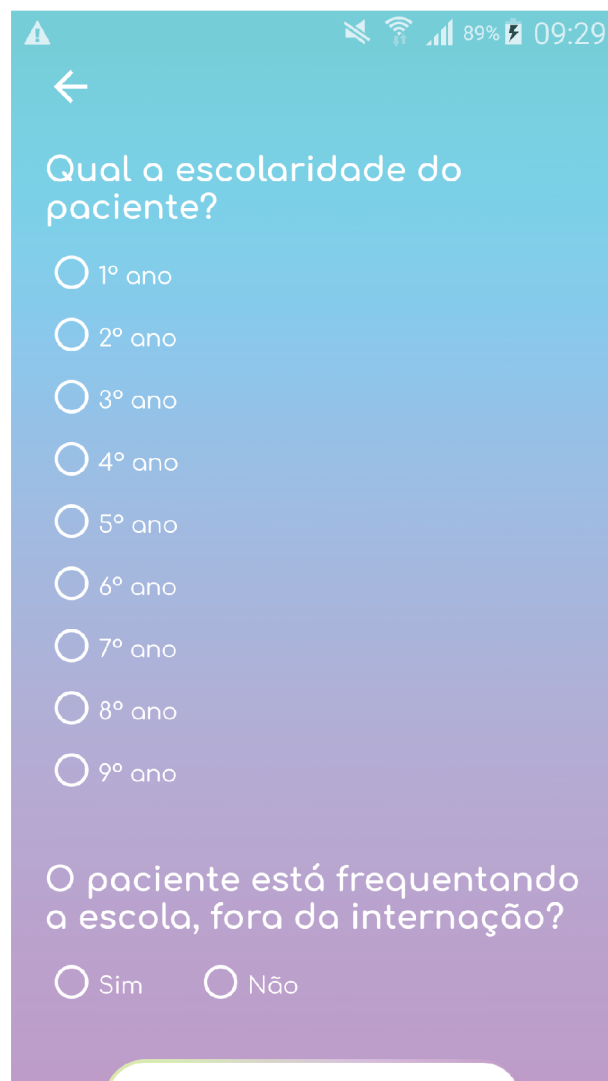
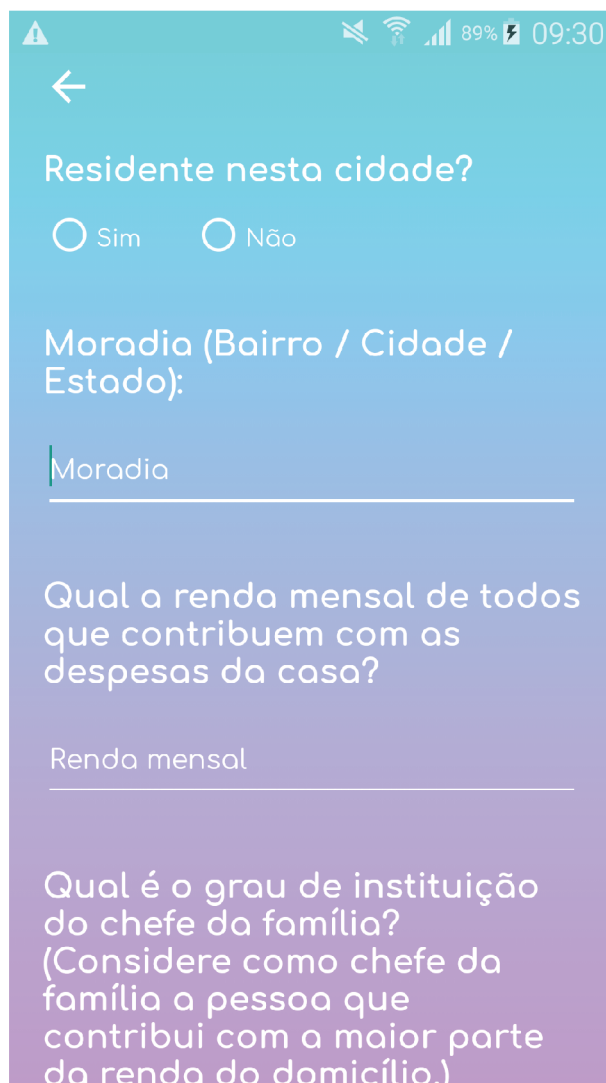
The image shows a mobile application interface on a smartphone. At the top, there is a status bar with icons for signal, Wi-Fi, battery (89%), and time (09:29). Below the status bar is a teal header with a white back arrow icon. The main content area has a light blue background with the text "Qual a escolaridade do paciente?" in white. Below this text are nine radio button options, each followed by a grade: "1º ano", "2º ano", "3º ano", "4º ano", "5º ano", "6º ano", "7º ano", "8º ano", and "9º ano". At the bottom of the form, there is a question "O paciente está frequentando a escola, fora da internação?" followed by two radio button options: "Sim" and "Não". The bottom of the screen shows a white home indicator bar.

Figura 11 – Tela de registro da escolaridade

- Se reside nesta cidade
- Moradia
- Renda mensal de todos que contribuem com as despesas da família
- Grau de instrução do chefe da família

Além disso, caso alguma informação de qualquer tela de registro não for preenchida, um *pop-up* de erro, representado pela Figura 13, é exibido informando ao usuário que é necessário preencher todos os campos para continuar.

Após a realização do cadastro completo do paciente, deve-se escolher entre as duas opções existentes de questionário, representadas pela Figura 14, dentre elas, existem perguntas para o paciente (criança) e para os pais.



The screenshot shows a mobile application interface with a teal header and a purple gradient body. At the top, there is a back arrow and a status bar with icons for signal, Wi-Fi, and battery (89%) at 09:30. The main content area contains three questions, each with a corresponding input field:

- Residente nesta cidade?** with radio buttons for ☐ Sim and ☐ Não.
- Moradia (Bairro / Cidade / Estado):** with a text input field containing the word "Moradia".
- Qual a renda mensal de todos que contribuem com as despesas da casa?** with a text input field containing "Renda mensal".
- Qual é o grau de instituição do chefe da família? (Considere como chefe da família a pessoa que contribui com a maior parte da renda do domicílio.)** with a text input field.

Figura 12 – Tela de registro da cidade

3.3.3 Questionário

Logo em seguida, o questionário é iniciado conforme a escolha da tela anterior (Figura 14). O modelo padrão de pergunta é composto por título e cinco opções de resposta (podendo variar conforme a pergunta). Além disso, na Figura 15, mostra-se circulado no canto superior esquerdo quantas perguntas já foram respondidas e o total de perguntas do questionário, e também, circulado no canto superior direito, o tipo do questionário que está sendo respondido, variando entre “Crianças” e “Pais”.

Além disso, depois que uma opção de resposta é selecionada, é verificado se existe uma sub-pergunta, se sim, um diálogo, representado pela figura 16, é mostrado na tela contendo esta sub-pergunta, junto às suas possíveis respostas, contendo opções selecionáveis ou caixa de texto.

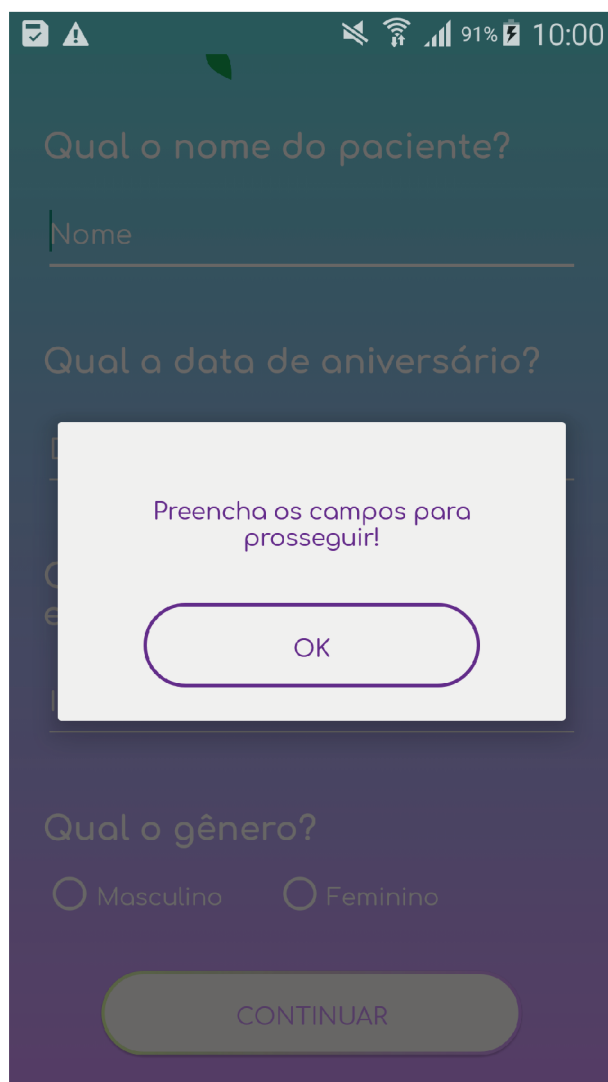
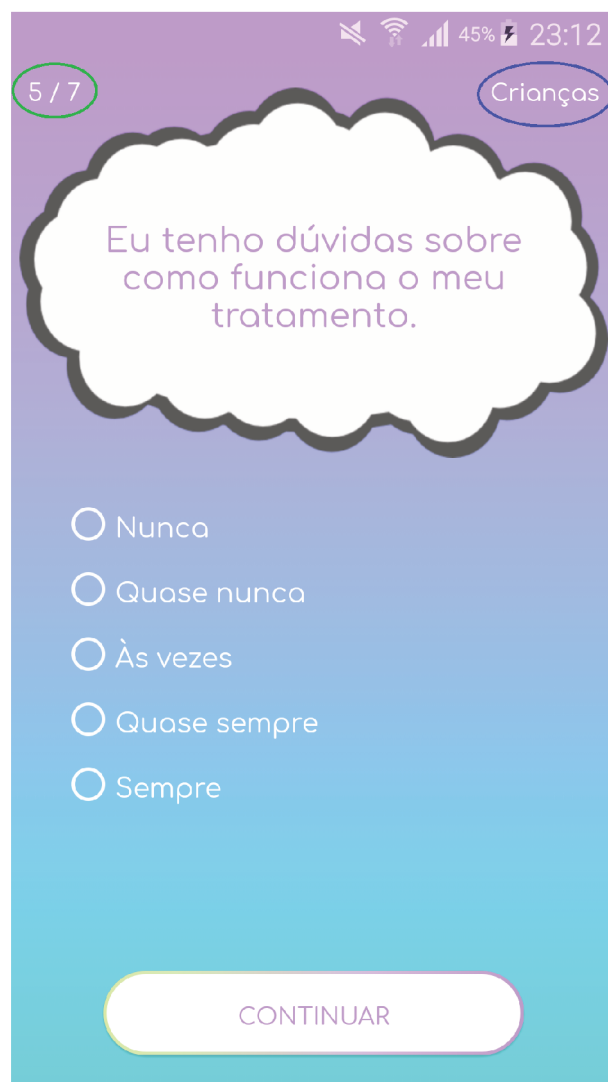


Figura 13 – Tela de erro no registro



Figura 14 – Tela de escolher a opção de questionário



The screenshot shows a mobile application interface with a purple-to-blue gradient background. At the top, the status bar displays icons for signal, Wi-Fi, and battery at 45%, along with the time 23:12. In the top left corner, a green circle contains the text "5 / 7". In the top right corner, a blue circle contains the text "Crianças". A large white cloud-shaped speech bubble with a black outline is centered on the screen, containing the text "Eu tenho dúvidas sobre como funciona o meu tratamento." Below the speech bubble, there are five radio button options: "Nunca", "Quase nunca", "Às vezes", "Quase sempre", and "Sempre". At the bottom of the screen, there is a white rounded rectangular button with a purple border and the text "CONTINUAR".

5 / 7

Crianças

Eu tenho dúvidas sobre
como funciona o meu
tratamento.

☐ Nunca

☐ Quase nunca

☐ Às vezes

☐ Quase sempre

☐ Sempre

CONTINUAR

Figura 15 – Tela de pergunta



Figura 16 – Tela de sub-pergunta

Ademais, se por ventura uma opção das respostas não for selecionada, um pop-up de erro, representado pela Figura 17, é exibido para que o usuário escolher uma alternativa. Da mesma forma acontece com as sub-respostas.



Figura 17 – Tela de erro ao não selecionar resposta

3.3.4 Finalização

Por fim, após o questionário ter sido respondido por completo, uma tela final, representada pela figura 18, é mostrada com as opções de fechar o questionário, iniciar novo questionário, exportar uma planilha para a memória interna do dispositivo ou continuar para o próximo questionário, como por exemplo: se o primeiro questionário respondido foi do paciente, ao clicar continuar respondendo, o usuário será redirecionado para o questionário dos pais e vice-versa .

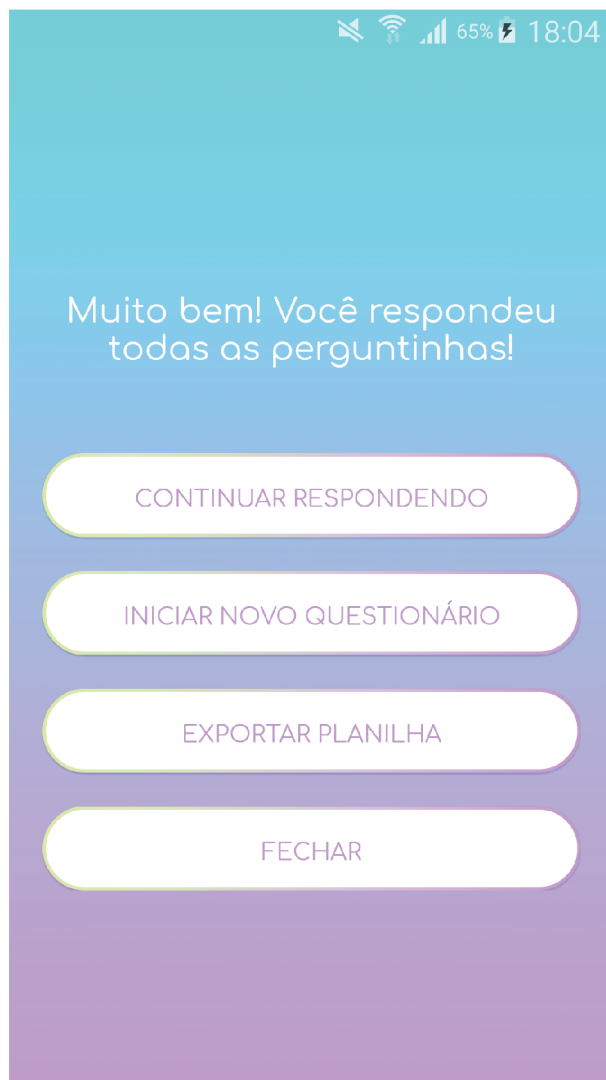


Figura 18 – Tela Final

Ao selecionar “EXPORTAR PLANILHA” é gerado uma planilha que contém duas páginas, representada pelas imagens 19 e 20, onde a primeira mostra o nome do paciente, o número do questionário respondido, a quantidade total de pontos que será utilizada para análise do estado da criança e os pontos separados por categoria da pergunta. Já na segunda página, existe a pergunta que foi respondida, junto com a sua resposta, a categoria desta, sua pontuação, o número do questionário, e caso houver, sua sub-pergunta junto da sua sub-resposta.

Para exportar a planilha foi utilizado uma biblioteca do Apache chamada POI ([APACHE, 2002](#)) que permite o uso do objeto `HSSFWorkbook`, onde a planilha é criada. Nesse objeto é possível customizar as colunas (com estilos diferentes) e adicionar o conteúdo necessário. Em seguida, após a criação da planilha com os dados inseridos na tabela, esta é exportada para a memória interna do telefone, ou mais especificamente no caminho: “Meus Arquivos/Todos os Arquivos/COPHAT/”.

Esta planilha ficará armazenada em um dispositivo designado para a coleta de dados, onde o hospital terá um smartphone ou um tablet para aplicar o questionário. Então os dados ficarão armazenados na memória interna do aparelho, onde o COPHAT será aplicado uma vez por mês.

1	Nome	Avaliação	Pontos	Compreensão doença	Internação	Realização e êxito do tratamento	Efeitos colaterais	Expectativa do retorno à escola
2	João Da Silva	0	22	8	6	3	1	4
3	Silvia Santos	1	10	2	3	3	2	0
4	Total		32	10	9	6	3	4

Figura 19 – Planilha de Respondentes

Id	Pergunta	Resposta	Sub-Pergunta	Sub-Resposta	Pontos	Avaliação	Categoria
1	Quando eu pergunto sobre a minha doença, as	Sempre			4	0	Compreensão doença
2	Eu fico preocupado(a) quando fico sabendo que	Às vezes			2	0	Internação
3	É difícil para mim receber resultados ruins de e	Quase sempre	Parece que sinto:	Medo Tristeza	3	0	Realização e êxito do tratamento
4	Eu sinto mal estar com o tratamento.	Quase nunca	Quais?	Enjojo	1	0	Efeitos colaterais
5	Eu acho que será difícil acompanhar as ativida	Sempre			4	0	Expectativa do retorno à escola
6	Eu tenho dúvidas sobre como funciona o meu t	Sempre			4	0	Compreensão doença
7	Eu tenho raiva de ficar internado(a).	Sempre			4	0	Internação
8	Quando meu filho (a) pergunta algo sobre a doi	Às vezes			2	1	Compreensão doença
9	Meu filho(a) fica ansioso (a) quando vai ser inte	Quase sempre			3	1	Internação
10	É difícil para meu filho (a) receber resultados ru	Quase sempre	Ele (a) parece sentir:	Raiva Medo Tristeza	3	1	Realização e êxito do tratamento
11	Meu filho (a) se sente mal com o tratamento.	Às vezes	Marque os sintomas:	Fraqueza Sono Initacã	2	1	Efeitos colaterais
12	Meu filho (a) acha que será difícil acompanhar	Nunca		Fraqueza Sono Initacã	0	1	Expectativa do retorno à escola

Figura 20 – Planilha de Perguntas

3.4 Testes de usabilidade

Foi utilizado o método de Think-Aloud para um teste rápido da primeira versão do aplicativo. Foram escolhidas 4 pessoas (meus pais, minha namorada e um amigo) para naturalmente avançarem nas telas do aplicativo, preenchendo a ficha cadastral, e em seguida, respondendo ao questionário como paciente ou como responsável. Os itens a serem avaliados no teste são: funcionalidade, intuitividade e agradabilidade do design. As métricas a serem avaliadas são o tempo gasto para responder a ficha e o tempo gasto para preencher o questionário.

3.4.1 Preparação do Teste

A preparação do testes envolveu a escolha dos usuários, com diferentes perfis, e a definição das atividades, regras e da verbalização, apresentadas aos usuários em um roteiro:

1. TAREFA: Preencha aos campos existentes no fluxo de cadastro da forma mais coerente possível com os seus dados verdadeiros e em seguida, responda todas as perguntas do questionário do aplicativo de acordo com sua experiência pessoal.
2. REGRAS:
 - a) Simplesmente verbalize o que estiver fazendo, como está fazendo, como se estivesse pensando em voz alta;
 - b) O observador não poderá ajudá-lo;
 - c) Você poderá sair a qualquer momento;
 - d) Você deve executar as tarefas conversando.
3. VERBALIZAÇÃO : Tente em cada atividade informar se:
 - a) Sente alguma dificuldade? Qual?
 - b) Sente-se incomodado? Com o quê?
 - c) Sente-se satisfeito? Com o quê?
 - d) Pensou em alguma resposta que não consta nas alternativas?

Dessa forma, à medida que foram relatados os erros, estes foram anotados para serem analisados se procedem e no caso das melhorias, estas seriam avaliadas para possíveis mudanças futuramente.

3.4.2 Resultados

As tabelas apresentadas na Figura 21 reúnem os resultados obtidos com o teste de usabilidade aplicado, considerando problemas funcionais e intuitivos, problemas com design, pontos positivos que foram apontados e tempo médio. O Teste de Usabilidade mostrou sua utilidade na identificação de diversos problemas de usabilidade, permitindo também identificar o grau de importância dos problemas identificados e o custo/benefício da sua correção e manutenção.

Seção	Problema funcional relatado
Ficha	impossibilidade de esconder o teclado ao perder o foco
Escolha	dificuldade com qual botão selecionar
Quest	impossibilidade de voltar a pergunta

Seção	Pergunta	Problema intuitivo relatado
Ficha	-	dificuldade de entender as perguntas
Quest	1	dificuldade de entender a pergunta
Quest	3	dificuldade de entender a sub-pergunta
Quest	4	dificuldade de entender a sub-pergunta

Seção	Problema com design relatado
Ficha	dificuldade com a organização das perguntas
Quest	impossibilidade de selecionar mais opções
Quest	vontade de escrever comentários na pergunta, além da resposta

Seção	Pontos positivos
Ficha	design agradável
Quest	construção das perguntas interessante (múltipla escolha)
Ficha	quantidade de perguntas adequadas por tela

Seção	Tempo médio
Ficha	5 à 10 minutos
Questionário	10 à 20 minutos

Figura 21 – Síntese dos resultados do teste de usabilidade.

4 Conclusão

4.1 Considerações Finais

Produzir este projeto foi importante para ampliar os conhecimentos sobre um tema tão presente na realidade profissional. Técnicas utilizadas como Kanban e parte do Scrum, foram essenciais para o desenvolvimento do produto final. Além disso, a utilização do Room como banco de dados, facilitou a visualização da construção do banco no Android. E por último, o Kotlin, linguagem estaticamente tipada que foi de extrema importância para o projeto, pois esta linguagem torna o entendimento do código mais claro além de estar presente nas maiores empresas da atualidade.

4.2 Contribuições

Este projeto irá ajudar de forma significativa na coleta de dados e posterior análise dos mesmos para profissionais da área da psicologia. Com o uso de um aplicativo, será possível a interatividade dos pacientes na imersão com o questionário, viabilizando o uso de um dispositivo eletrônico para responder perguntas de forma dinâmica.

4.3 Trabalhos Futuros

Para dar continuação a este projeto, seria interessante uma integração com um back-end incorporado a um CMS (Content Manager System), onde é possível adicionar, remover ou editar as perguntas do questionário através de uma interface Web, pelo qual o aplicativo irá consumir uma API, para carregar os dados fornecidos.

Além disso, a implementação de uma versão para iOS iria abranger uma quantidade maior de dispositivos móveis, podendo enfim ser utilizado como um “esqueleto” para outros tipos de questionários. Em outras palavras, este projeto pode servir de base para outros aplicativos que coletam dados de usuários.

Por fim, como ideia para um projeto futuro, juntamente com a integração Web, pode-se adequar o questionário à outros tipos de pesquisa, por exemplo: algum outro pesquisador que deseja utilizar o formato deste questionário, porém com perguntas diferentes, seria possível adicioná-las na plataforma WEB deixando o questionário com perguntas variáveis.

Referências

- AMARAL, J. H. Ferreira do; ENUMO, S. F. Coping of hospitalization, ailment, and treatment (cophat): Design and assessment of an instrument for parents and hospitalized children and adolescents. In: WILEY 111 RIVER ST, HOBOKEN 07030-5774, NJ USA. *PSYCHO-ONCOLOGY*. [S.l.], 2017. v. 26, p. 85–86. Citado 5 vezes nas páginas 10, 11, 60, 62 e 65.
- APACHE. *Apache POI*. 2002. Disponível em: <<http://poi.apache.org/>>. Acesso em: 23 out 2018. Citado na página 42.
- BOREN, T.; RAMEY, J. Thinking aloud: Reconciling theory and practice. *IEEE transactions on professional communication*, IEEE, v. 43, n. 3, p. 261–278, 2000. Citado na página 23.
- CAELUM. *Testes de Usabilidade*. 2018. Disponível em: <<https://www.caelum.com.br/apostila-ux-usabilidade-mobile-web/usabilidade/>>. Acesso em: 18 nov 2018. Citado 2 vezes nas páginas 21 e 22.
- CAMARGO, R. *Metodologia Agile: quais as mais utilizadas?* 2018. Disponível em: <<https://robsoncamargo.com.br/blog/Metodologia-Agile-quais-as-mais-utilizadas>>. Citado na página 16.
- CORDEIRO, F. *Kotlin: Introdução e Primeiros Passos*. 2018. Disponível em: <<https://www.androidpro.com.br/blog/kotlin/kotlin/>>. Acesso em: 20 set 2018. Citado 4 vezes nas páginas 16, 17, 18 e 19.
- DEVELOPERS, G. *Desenvolver apps para Android com o Kotlin*. 2018. Disponível em: <<https://developer.android.com/kotlin/?hl=pt-br>>. Citado na página 16.
- DIAS, R. R.; BAPTISTA, M. N.; BAPTISTA, A. S. D. Enfermaria de pediatria: avaliação e intervenção psicológica. *Psicologia hospitalar: teoria, aplicações e casos clínicos*, Guanabara Koogan Rio de Janeiro, p. 53–73, 2003. Citado na página 10.
- ERICSSON, K.; SIMON, H. *Protocol analysis; Verbal reports as data (revised edition)*. *Bradfordbooks*. [S.l.]: MIT Press, Cambridge, 1993. Citado na página 23.
- FOLKMAN, S.; LAZARUS, R. S. An analysis of coping in a middle-aged community sample. *Journal of health and social behavior*, JSTOR, p. 219–239, 1980. Citado na página 11.
- GARRETT, J. *The Elements of User Experience: User-Centered Design for the Web and Beyond*. Pearson Education, 2010. (Voices That Matter). ISBN 9780321624642. Disponível em: <<https://books.google.com.br/books?id=9QC6r5OzCpUC>>. Citado na página 20.
- GODOY, R. M. B. de. *Kanban: 4 passos para implementar em uma equipe*. 2014. Citado na página 14.
- JETBRAINS. *Kotlin*. 2016. Disponível em: <<https://kotlinlang.org/>>. Acesso em: 10 jun 2017. Citado na página 17.

- LLC, G. *Room Persistence Library*. 2018. Disponível em: <<https://developer.android.com/training/data-storage/room>>. Acesso em: 15 out 2018. Citado na página 19.
- MCCRACKEN, C. *How to Conduct Usability Testing from Start to Finish*. 2016. Disponível em: <<https://uxmastery.com/beginners-guide-to-usability-testing/>>. Citado na página 23.
- MORAES, E. O.; ENUMO, S. R. F. Estratégias de enfrentamento da hospitalização em crianças avaliadas por instrumento informatizado. *Psico-USF*, Universidade São Francisco, v. 13, n. 2, p. 221–231, 2008. Citado 2 vezes nas páginas 10 e 11.
- SHEKHAR, A. *Android Debug Database*. 2016. Disponível em: <<https://github.com/amitshekharitbhu/Android-Debug-Database>>. Acesso em: 17 set 2018. Citado na página 25.
- SUGIMORI, Y. et al. Toyota production system and kanban system materialization of just-in-time and respect-for-human system. *The International Journal of Production Research*, Taylor & Francis, v. 15, n. 6, p. 553–564, 1977. Citado na página 12.
- TAKEUCHI, H.; NONAKA, I. The new new product development game. *Harvard business review*, v. 64, n. 1, p. 137–146, 1986. Citado na página 12.
- TECHAMERICA. *ANSI/EIA 649-B: Configuration Management Standard*. EUA, 2011. Citado na página 10.
- TRELLO. *Trello*. 2011. Disponível em: <<https://trello.com/>>. Acesso em: 20 ago 2018. Citado na página 24.
- WIKIPEDIA. *Thinking Aloud Protocol*. 2017. Disponível em: <https://en.wikipedia.org/wiki/Think_aloud_protocol>. Acesso em: 04 dez 2018. Citado na página 23.
- ZANELLA, J. R. *Kotlin: explorando uma nova linguagem*. 2017. Disponível em: <<https://medium.com/android-dev-br/kotlin-explorando-uma-nova-linguagem-EFB88F-799ffc99a1a2>>. Citado na página 19.

Apêndices

APÊNDICE A – Código fonte da criação do banco de dados utilizando Room

```
//ENTIDADES
@Database(entities = [QuestionData::class, AnswerData::class,
    RespondentData::class,
        ScaleData::class, ScaleValueData::class,
        QuestionnaireData::class,
        EvaluationData::class, SectionData::class,
        SubQuestionData::class, SubAnswerData::class], version =
        1)
abstract class AppDatabase : RoomDatabase() {

    abstract fun questionsDAO(): DatabaseDao

    companion object {
        private var INSTANCE: AppDatabase? = null

        //VERIFICAR INSTANCIA DO BANCO DE DADOS
        fun getInstance(context: Context): AppDatabase? {
            if (INSTANCE == null) {
                synchronized(AppDatabase::class) {
                    INSTANCE = Room.databaseBuilder(context.applicationContext,
                        AppDatabase::class.java, "cophat-database")
                        .allowMainThreadQueries()
                        .build()
                }
            }
            return INSTANCE
        }

        //DESTRUIR INSTANCIA DO BANCO DE DADOS
        fun destroyInstance() {
            INSTANCE = null
        }
    }
}
```

APÊNDICE B – Exemplo de uma entidade do banco de dados

@Entity

```
class RespondentData(@PrimaryKey(autoGenerate = true) val id: Int,
    @ColumnInfo(name = "name") val name: String,
    @ColumnInfo(name = "birthDate") val birthDate: String,
    @ColumnInfo(name = "age") val age: String,
    @ColumnInfo(name = "gender") val gender: String,
    @ColumnInfo(name = "motherName") val motherName: String,
    @ColumnInfo(name = "motherJob") val motherJob: String,
    @ColumnInfo(name = "fatherName") val fatherName: String,
    @ColumnInfo(name = "fatherJob") val fatherJob: String,
    @ColumnInfo(name = "parentsCivilState") val
        parentsCivilState: String,
    @ColumnInfo(name = "familyReligion") val familyReligion:
        String,
    @ColumnInfo(name = "diagnostics") val diagnostics: String,
    @ColumnInfo(name = "diagnosticsTime") val diagnosticsTime:
        String,
    @ColumnInfo(name = "internedDays") val internedDays: String,
    @ColumnInfo(name = "internedQuantity") val
        internedQuantity: String,
    @ColumnInfo(name = "scholarly") val scholarship: String,
    @ColumnInfo(name = "scholarFrequency") val
        scholarFrequency: String,
    @ColumnInfo(name = "liveInThisCity") val liveInThisCity:
        String,
    @ColumnInfo(name = "home") val home: String,
    @ColumnInfo(name = "income") val income: String,
    @ColumnInfo(name = "bossInstitution") val bossInstitution:
        String,
    @ColumnInfo(name = "idEvaluation") val idEvaluation: Int,
    @ColumnInfo(name = "points") val points: Int)
```

APÊNDICE C – Código fonte da exportação da planilha

```

class ExportWorkbook {

    companion object {

        private val columns1 = arrayOf("Nome", "Avaliacao", "Pontos",
            Category.CATEGORYONE.title, Category.CATEGORYTWO.title,
            Category.CATEGORYTHREE.title, Category.CATEGORYFOUR.title,
            Category.CATEGORYFIVE.title)

        private val columns2 = arrayOf("Id", "Pergunta", "Resposta",
            "Sub-Pergunta", "Sub-Resposta", "Pontos", "Avaliacao", "Categoria")

        fun saveExcelFile(mActivity: BaseActivity, respondentList:
            List<RespondentData>, answerList: List<AnswerData>) {
            var points = 0
            var categoryOne = 0
            var categoryTwo = 0
            var categoryThree = 0
            var categoryFour = 0
            var categoryFive = 0

            for (respondent in respondentList) {
                points += respondent.points
                categoryOne += respondent.categoryOne
                categoryTwo += respondent.categoryTwo
                categoryThree += respondent.categoryThree
                categoryFour += respondent.categoryFour
                categoryFive += respondent.categoryFive
            }

            //BIBLIOTECA PARA CRIACAO DE PLANILHA
            val workbook = HSSFWorkbook()

            val sheetRespondent = workbook.createSheet("RESPONDENTES")
                //CRIACAO DA PLANILHA DOS RESPONDENTES
            val sheetAnswers = workbook.createSheet("RESPOSTAS") //CRIACAO DA
                PLANILHA DAS RESPOSTAS
        }
    }
}

```

```
//CUSTOMIZACAO DA PRIMEIRA LINHA DA PLANILHA
val headerFont = workbook.createFont()
headerFont.boldweight = HSSFFont.BOLDWEIGHT_BOLD
headerFont.color = IndexedColors.BLUE.index

val totalFont = workbook.createFont()
totalFont.boldweight = HSSFFont.BOLDWEIGHT_BOLD
totalFont.color = IndexedColors.GREEN.index

//CUSTOMIZACAO DAS CELULAS DA PLANILHA
val headerCellStyle = workbook.createCellStyle()
headerCellStyle.setFont(headerFont)
headerCellStyle.alignment = CellStyle.ALIGN_CENTER
headerCellStyle.verticalAlignment = CellStyle.VERTICAL_CENTER

val totalCellStyle = workbook.createCellStyle()
totalCellStyle.setFont(totalFont)
totalCellStyle.alignment = CellStyle.ALIGN_CENTER
totalCellStyle.verticalAlignment = CellStyle.VERTICAL_CENTER

val cs = workbook.createCellStyle()
cs.alignment = CellStyle.ALIGN_CENTER
cs.verticalAlignment = CellStyle.VERTICAL_CENTER

// CRIACAO DA PRIMEIRA LINHA OU "CABECALHO"
val headerRowRespondents = sheetRespondent.createRow(0)
val headerRowAnswers = sheetAnswers.createRow(0)

// CRIACAO DAS COLUNAS DO CABECALHO
for (col in columns1.indices) {
    val cell = headerRowRespondents.createCell(col)
    cell.setCellValue(columns1[col])
    cell.setCellStyle(headerCellStyle)
}
for (col in columns2.indices) {
    val cell = headerRowAnswers.createCell(col)
    cell.setCellValue(columns2[col])
    cell.setCellStyle(headerCellStyle)
}

//PREENCHENDO AS LINHAS DA PLANILHA
```

```
var rowIdxRespondents = 1
for (respondent in respondentList) {
    val row = sheetRespondent.createRow(rowIdxRespondents++)

    row.createCell(0).setCellValue(respondent.name)
    row.getCell(0).setCellStyle(cs)
    sheetRespondent.setColumnWidth(0, (20 * 500))

    row.createCell(1).setCellValue(respondent.idEvaluation.toString())
    row.getCell(1).setCellStyle(cs)
    sheetRespondent.setColumnWidth(1, (5 * 500))

    row.createCell(2).setCellValue(respondent.points.toString())
    row.getCell(2).setCellStyle(cs)
    sheetRespondent.setColumnWidth(2, (5 * 500))

    row.createCell(3).setCellValue(respondent.categoryOne.toString())
    row.getCell(3).setCellStyle(cs)
    sheetRespondent.setColumnWidth(3, (12 * 500))

    row.createCell(4).setCellValue(respondent.categoryTwo.toString())
    row.getCell(4).setCellStyle(cs)
    sheetRespondent.setColumnWidth(4, (8 * 500))

    row.createCell(5).setCellValue(respondent.categoryFour.toString())
    row.getCell(5).setCellStyle(cs)
    sheetRespondent.setColumnWidth(5, (15 * 500))

    row.createCell(6).setCellValue(respondent.categoryFour.toString())
    row.getCell(6).setCellStyle(cs)
    sheetRespondent.setColumnWidth(6, (10 * 500))

    row.createCell(7).setCellValue(respondent.categoryFive.toString())
    row.getCell(7).setCellStyle(cs)
    sheetRespondent.setColumnWidth(7, (15 * 500))
}

addTotal(respondentList.size + 1, sheetRespondent, totalCellStyle,
    points, categoryOne, categoryTwo, categoryThree, categoryFour,
    categoryFive)

var rowIdxAnswers = 1
```

```
for (answer in answerList) {
    val row = sheetAnswers.createRow(rowIdxAnswers++)
    row.createCell(0).setCellValue(answer.id.toDouble())
    row.getCell(0).setCellStyle(cs)
    sheetAnswers.setColumnWidth(0, (5 * 500))

    row.createCell(1).setCellValue(answer.question)
    sheetAnswers.setColumnWidth(1, (20 * 500))

    row.createCell(2).setCellValue(answer.title)
    row.getCell(2).setCellStyle(cs)
    sheetAnswers.setColumnWidth(2, (10 * 500))

    row.createCell(3).setCellValue(answer.subQuestion)
    sheetAnswers.setColumnWidth(3, (20 * 500))

    row.createCell(4).setCellValue(answer.subAnswer)
    row.getCell(4).setCellStyle(cs)
    sheetAnswers.setColumnWidth(4, (10 * 500))

    row.createCell(5).setCellValue(answer.points.toDouble())
    row.getCell(5).setCellStyle(cs)
    sheetAnswers.setColumnWidth(5, (5 * 500))

    row.createCell(6).setCellValue(answer.idEvaluation.toDouble())
    row.getCell(6).setCellStyle(cs)
    sheetAnswers.setColumnWidth(6, (5 * 500))

    row.createCell(7).setCellValue(answer.questionCategory)
    row.getCell(7).setCellStyle(cs)
    sheetAnswers.setColumnWidth(7, (15 * 500))
}

//EXPORTANDO PARA O ARQUIVO XLS
val mFolder = File(Environment.getExternalStorageDirectory(),
    "COPHAT")
if (!mFolder.exists()) {
    mFolder.mkdirs()
}
val file = File(mFolder, "COPHAT.xls")
val os = FileOutputStream(file)
workbook.write(os)
```

```
os.close()

//VERIFICANDO SE HOUVE ALGUM PROBLEMA NA CRIACAO NA PLANILHA
if (workbook.getSheet("RESPONDENTES").physicalNumberOfRows ==
    respondentList.size + 2 &&
    workbook.getSheet("RESPOSTAS").physicalNumberOfRows - 1 ==
        answerList.size) {
    val dialog =
        AlertDialogFragment.newInstance(mActivity.getString(R.string.workbook_s
        dialog.show(mActivity.supportFragmentManager, "EXPORT_DONE")
} else {
    val dialog = AlertDialogFragment.newInstance("Ocorreu um erro
        ao exportar planilha")
    dialog.show(mActivity.supportFragmentManager, "EXPORT_DONE")
}
}

private fun addTotal(index: Int, sheet: HSSFSheet, cs: HSSFCellStyle,
    points: Int, categoryOne: Int, categoryTwo: Int, categoryThree:
    Int, categoryFour: Int, categoryFive: Int) {
    val row = sheet.createRow(index)

    row.createCell(0).setCellValue("Total")
    row.getCell(0).setCellStyle(cs)
    sheet.setColumnWidth(0, (20 * 500))

    row.createCell(1).setCellValue("")
    row.getCell(1).setCellStyle(cs)
    sheet.setColumnWidth(1, (5 * 500))

    row.createCell(2).setCellValue(points.toString())
    row.getCell(2).setCellStyle(cs)
    sheet.setColumnWidth(2, (5 * 500))

    row.createCell(3).setCellValue(categoryOne.toString())
    row.getCell(3).setCellStyle(cs)
    sheet.setColumnWidth(3, (12 * 500))

    row.createCell(4).setCellValue(categoryTwo.toString())
    row.getCell(4).setCellStyle(cs)
    sheet.setColumnWidth(4, (8 * 500))
```

```
        row.createCell(5).setCellValue(categoryThree.toString())
        row.getCell(5).setCellStyle(cs)
        sheet.setColumnWidth(5, (15 * 500))

        row.createCell(6).setCellValue(categoryFour.toString())
        row.getCell(6).setCellStyle(cs)
        sheet.setColumnWidth(6, (10 * 500))

        row.createCell(7).setCellValue(categoryFive.toString())
        row.getCell(7).setCellStyle(cs)
        sheet.setColumnWidth(7, (15 * 500))
    }
}
```

Anexos

ANEXO A – Ficha cadastral do paciente

O presente anexo apresenta as questões que compõem a Ficha Cadastral do Paciente, necessária para a coleta das informações pessoais, familiares, sociais e econômicas do paciente e desenvolvida no trabalho de [Amaral e Enumo \(2017\)](#) como instrumento integrante do estudo de *coping* de hospitalização para crianças e adolescentes.

1. Nome pais e profissão:
 - a) Mãe:
 - b) Pai:
2. Estado civil dos pais:
 - a) Casado
 - b) Amasiado
 - c) Divorciado/Separado
 - d) Solteiro
3. Religião da família:
 - a) Católica
 - b) Evangélica
 - c) Espírita
 - d) Outra:
4. Nome do paciente:
5. Nascimento: / /
6. Idade no dia da entrevista:
7. Sexo:
 - a) Masculino
 - b) Feminino
8. Diagnóstico:
9. Tempo de diagnóstico:
10. Número de dias internado:

11. Quantas internações:

12. Escolaridade:

- a) Primeiro ano
- b) Segundo ano
- c) Terceiro ano
- d) Quarto ano
- e) Quinto ano
- f) Sexto ano
- g) Sétimo ano
- h) Oitavo ano
- i) Nono ano

13. Está frequentando a escola durante a internação:

- a) Sim
- b) Não

14. Residente nesta cidade?

- a) Sim
- b) Não

15. Moradia (Bairro/Cidade/Estado):

16. Qual é a renda mensal de todos que contribuem com as despesas da casa?

17. Qual é o grau de instrução do chefe da família? (Considere como chefe da família a pessoa que contribui com a maior parte da renda do domicílio.)

- a) Analfabeto / Fundamental I incompleto
- b) Fundamental I completo / Fundamental II incompleto
- c) Fundamental completo/Médio incompleto
- d) Médio completo/Superior incompleto
- e) Superior completo

ANEXO B – Questionário para os pacientes

Este anexo contém o questionário para os pacientes, que são crianças ou adolescentes, desenvolvido utilizando-se a escala COPHAT-CA ([AMARAL; ENUMO, 2017](#)). Todas as perguntas do questionário tem como alternativas as seguintes possibilidades: **Nunca**, **Quase nunca**, **Às vezes**, **Quase sempre** e **Sempre**. Além disso, algumas possuem sub-perguntas.

1. Quando eu pergunto sobre a minha doença, as respostas ainda me deixam com dúvidas.
2. Eu fico preocupado(a) quando fico sabendo que vou ser internado(a).
3. É difícil para mim receber resultados ruins de exames.

(Sub-Pergunta) - Parece que sinto:

- a) Raiva
- b) Medo
- c) Tristeza

4. Eu sinto mal estar com o tratamento.

(Sub-Pergunta) - Quais?

- a) Cansaço
- b) Falta de apetite
- c) Enjoo
- d) Dor
- e) Tontura
- f) Fraqueza
- g) Sono
- h) Irritação
- i) Dificuldade de atenção
- j) Suor
- k) Dificuldade motora
- l) Outros:

5. Eu acho que será difícil acompanhar as atividades na escola.

6. Eu tenho dúvidas sobre como funciona meu tratamento.
7. Eu tenho raiva de ficar internado(a).
8. Eu atrapalho o tratamento.
9. É difícil para mim aceitar as mudanças no meu corpo por causa do tratamento.
(Sub-Pergunta) - Quais?
 - a) Inchaço
 - b) Perda de peso
 - c) Ganho de peso
 - d) Queda de cabelo
 - e) Marcas de injeção/agulha
 - f) Manchas
 - g) Perdas de partes do corpo
 - h) Palidez
 - i) Olho amarelo
 - j) "Caroços"
 - k) Ter cateter ou fístula
 - l) Outros:
10. Eu tenho medo de que os colegas de escola riam de mim.
11. Eu não sei por que tenho que tomar remédios.
12. É insuportável ficar internado (a), mesmo podendo brincar no hospital.
13. É difícil para mim quando “furam a minha veia”.
14. Eu tenho dificuldades para dormir por causa do tratamento.
15. Eu tenho medo de me machucar na escola.
16. A equipe do hospital esconde de mim coisas sobre meu tratamento.
17. É difícil para mim ficar internado (a).
18. É difícil para mim, quando tiram meu sangue.
19. É difícil para mim não poder fazer as coisas que fazia antes do tratamento.
20. Eu tenho medo de voltar à escola.

-
21. Eu tenho dúvidas sobre o nome da doença que tenho.
22. Ficar internado (a) é ruim para mim, pois sinto falta da minha casa e das minhas coisas.
23. Eu me mexo quando vou fazer exames com aparelhos.
24. Eu tenho dificuldades em depender dos outros.
25. Eu acho que vai ser difícil conseguir fazer tudo o que o (a) professor (a) pedir.
26. Eu tenho dúvidas por que preciso fazer este tratamento.
27. Eu fico triste de ficar internado(a).
28. Eu acho que vou ficar doente para sempre.
29. Eu me sinto diferente dos outros.
- (Sub-Pergunta) - Parece que sinto:
- a) Raiva
 - b) Medo
 - c) Tristeza por causa disso
30. Eu tenho medo que meu professor (professora) não me aceite.
31. É difícil para mim pensar no futuro.
- (Sub-Pergunta) - Parece que sinto:
- a) Raiva
 - b) Medo
 - c) Tristeza
32. Eu acho ruim quando as outras pessoas perguntam sobre a minha doença/tratamento.
- (Sub-Pergunta) - Parece que sinto:
- a) Raiva
 - b) Medo
 - c) Tristeza
33. Eu fico com medo de estar (ficar) internado(a).
34. É difícil para mim ver a família preocupada comigo e a minha doença.

ANEXO C – Questionário para os pais ou responsáveis pelo paciente

O presente anexo contém o questionário que deve ser aplicado aos pais ou responsáveis pelos pacientes infantis ou adolescentes hospitalizados ([AMARAL; ENUMO, 2017](#)). Este questionário foi desenvolvido utilizando a escala COPHAT-P. Todas as perguntas do questionário tem como alternativas as seguintes possibilidades: **Nunca, Quase nunca, Às vezes, Quase sempre e Sempre**. Além disso, algumas possuem sub-perguntas.

1. Quando meu filho (a) pergunta algo sobre a doença, ele (ela) ainda permanece com dúvidas.
2. Meu filho(a) fica ansioso (a) quando vai ser internado (a).
3. É difícil para meu filho (a) receber resultados ruins de exames.

(Sub-Pergunta) - Ele (a) parece sentir:

- a) Raiva
- b) Medo
- c) Tristeza

4. Meu filho (a) se sente mal com o tratamento.

(Sub-Pergunta) - Marque os sintomas:

- a) Cansaço
- b) Falta de apetite
- c) Enjoo
- d) Dor
- e) Tontura
- f) Fraqueza
- g) Sono
- h) Irritação
- i) Dificuldade de atenção
- j) Suor
- k) Dificuldade motora

1) Outros:

5. Meu filho (a) acha que será difícil acompanhar as atividades na escola.
6. Meu filho (a) tem dúvidas sobre como funciona seu tratamento.
7. Meu filho (a) tem raiva de ficar internado(a).
8. Meu filho (a) atrapalha o tratamento.
9. É difícil para meu filho (a) aceitar as mudanças que ocorram no seu corpo por causa do tratamento.

(Sub-Pergunta) - Quais?

- a) Inchaço
- b) Perda de peso
- c) Ganho de peso
- d) Queda de cabelo
- e) Marcas de injeção/agulha
- f) Manchas
- g) Perdas de partes do corpo
- h) Palidez
- i) Olho amarelo
- j) "Caroços"
- k) Ter cateter ou fístula
- l) Outros:

10. Meu filho (a) tem medo de que os colegas de escola riam dele (dela).
11. Meu filho (a) não sabe por que tem que tomar os remédios.
12. É insuportável para meu filho(a) suportar a internação, mesmo que ele(ela) possa brincar no hospital.
13. É difícil para meu filho (a) quando o profissional “pega a sua veia”.
14. Meu filho (a) tem dificuldades para dormir por causa do tratamento.
15. Meu filho (a) tem medo de se machucar na escola.
16. A equipe esconde do meu filho (a) o motivo do tratamento dele(a).
17. É difícil para o meu filho (a) ficar internado (a).

18. É difícil para o meu filho (a) que tirem sangue dele (a).
19. É difícil para meu filho (a) não poder fazer as coisas que fazia antes do tratamento.
20. Meu filho (a) tem medo de voltar à escola.
21. Meu filho (a) desconhece o nome de sua doença.
22. Ficar internado (a) é ruim para o meu filho (a), pois ele (ela) sente falta de casa e de suas coisas.
23. Meu filho (a) se mexe na hora de fazer os exames feitos com aparelhos.
24. Meu filho (a) tem dificuldades em depender dos outros.
25. Meu filho(a) pensa que será difícil conseguir fazer tudo que o professor (a) pedir.
26. Meu filho (a) tem dúvidas sobre o motivo de seu tratamento.
27. Meu filho (filha) fica triste por estar internado(a).
28. Meu filho (a) tem medo de continuar doente.
29. Meu filho (a) sente-se diferente dos outros.
(Sub-Pergunta) - Ele (a) parece sentir:
 - a) Raiva
 - b) Medo
 - c) Tristeza por causa disso
30. Meu filho (a) acha que seu professor (sua professora) terá dificuldades em aceitá-lo (la).
31. É difícil para meu filho (a) pensar no futuro.
(Sub-Pergunta) - Ele (a) parece sentir:
 - a) Raiva
 - b) Medo
 - c) Tristeza
32. Meu filho (a) acha ruim quando as outras pessoas perguntam sobre sua doença/tratamento.
(Sub-Pergunta) - Ele (a) parece sentir:
 - a) Raiva

b) Medo

c) Tristeza

33. Meu filho (filha) fica com medo por estar internado(a).

34. É difícil para meu filho (a) ver a família preocupada com ele (ela).