UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS

GIAN CARLOS CASAGRANDE

SOFTWARE DE PRONTUÁRIO ODONTOLÓGICO VETERINÁRIO DIGITAL PARA DISPOSITIVOS MÓVEIS

BENTO GONÇALVES

2021

GIAN CARLOS CASAGRANDE

SOFTWARE DE PRONTUÁRIO ODONTOLÓGICO VETERINÁRIO DIGITAL PARA DISPOSITIVOS MÓVEIS

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Orientador: Prof. Me. Alexandre Erasmo Krohn Nascimento

BENTO GONÇALVES

2021

GIAN CARLOS CASAGRANDE

SOFTWARE DE PRONTUÁRIO ODONTOLÓGICO VETERINÁRIO DIGITAL PARA DISPOSITIVOS MÓVEIS

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Aprovado em 30/11/2021

BANCA EXAMINADORA

Prof. Me. Alexandre Erasmo Krohn Nascimento
Universidade de Caxias do Sul - UCS

Prof. Dra. Scheila de Avila e Silva
Universidade de Caxias do Sul - UCS

Prof. Me. Gabriele Dani Bonetti
Universidade de Caxias do Sul - UCS

Este trabalho é dedicado aos meus pais que nunca mediram esforços para oferecer apoio em minha vida.

AGRADECIMENTOS

A Deus, pela minha vida, e por me permitir vencer todos os desafios e obstáculos encontrados ao longo da minha trajetória acadêmica.

Aos meus pais, Vilson Casagrande e Janete Cassinelli Casagrande, e meu irmão, Giovani Henrique Casagrande, que me incentivaram e me apoiaram nos momentos difíceis.

Aos meus amigos e colegas de curso, especialmente ao Gabriel Szimanski Moro e ao Leonardo Bertele Tosin, companheiros de estudos no decorrer do curso.

Aos professores da Universidade de Caxias do Sul, em especial ao meu orientador, Alexandre Erasmo Krohn Nascimento, pelo suporte prestado neste trabalho de conclusão.

À professora Scheila de Avila e Silva, por ter proposto o tema de pesquisa presente neste trabalho.

À veterinária Manoela Maria Bianchi, pelo apoio com materiais de estudo, essenciais para realização da pesquisa bibliográfica da área de Odontologia Veterinária.

A todos vocês, meu muito obrigado!

“Tente uma, duas, três vezes e se possível tente a quarta, a quinta e quantas vezes for necessário. Só não desista nas primeiras tentativas, a persistência é amiga da conquista. Se você quer chegar aonde a maioria não chega, faça o que a maioria não faz.”

Bill Gates

RESUMO

A Odontologia Veterinária é uma especialidade da área da Medicina Veterinária que atua para trazer o bem-estar da saúde bucal dos animais. Os profissionais que trabalham nesta área enfrentam problemas com a forma em que são feitos os registros de prontuários dos animais atendidos, onde fichas odontológicas são utilizadas de forma impressa, e preenchidas por meio de escrita à mão, o que pode gerar um acúmulo físico de documentos, rasuras, e até a falta de entendimento entre a escrita de um profissional e de outro. Com isso, podem ser encontradas dificuldades na obtenção de informações importantes de um paciente, e assim, levar à perda de um histórico clínico. Constatando estes problemas, fazem-se necessárias soluções inovadoras que visam melhorar estes métodos de registros, contando com o auxílio da tecnologia. Logo, neste trabalho, criou-se uma aplicação para dispositivos móveis, que usufrui da praticidade e mobilidade oferecida pelos mesmos, possibilitando assim, digitalizar a forma de realizar os atendimentos. Para isso, implementou-se a capacidade de gerenciamento de uma clínica com clientes e animais, criação e visualização de prontuários por meio de fichas, odontograma, vínculo de imagens à um atendimento, e gerenciamento de usuários, que são características de uso essenciais para auxiliar profissionais em consultórios veterinários.

Palavras-chave: Aplicativo. Odontologia Veterinária. Dispositivos móveis.

LISTA DE FIGURAS

Figura 1 – Ficha odontológica de exame clínico canina 16
Figura 2 – Odontograma do cão 17
Figura 3 – Cavidade oral do cachorro 18
Figura 4 – Anatomia do crânio canina 19
Figura 5 – Anatomia da maxila e mandíbula canina 19
Figura 6 – Representações gráficas da maxila e mandíbula de diferentes espécies. . . . 20
Figura 7 – Representações gráficas do crânio e da cavidade oral canina 21
Figura 8 – *Software* Easy Vet Dental Charts 22
Figura 9 – *Software* Animal Dental Chart 22
Figura 10 – Odontograma no *software* BensVET 23
Figura 11 – Comparativo de vendas de dispositivos móveis 24
Figura 12 – Plataformas operacionais mais vendidas no mercado 25
Figura 13 – Processo Iconix 33
Figura 14 – Topologia do aplicativo 36
Figura 15 – Diagrama de Casos de Uso 38
Figura 16 – Diagrama de Domínio do sistema 51
Figura 17 – Diagrama de Domínio do sistema - Parte 1 52
Figura 18 – Diagrama de Domínio do sistema - Parte 2 53
Figura 19 – Diagrama de Domínio do sistema - Parte 3 54
Figura 20 – Diagrama de Domínio do sistema - Parte 4 55
Figura 21 – Diagrama de Domínio do sistema - Parte 5 56
Figura 22 – Interface gráfica - Criação de usuário administrador 60
Figura 23 – Interface gráfica - Menu de opções do administrador 61
Figura 24 – Interface gráfica - Dados da clínica 62
Figura 25 – Interface gráfica - Gerenciamento de usuários no sistema 63
Figura 26 – Interface gráfica - Autenticação do usuário 64
Figura 27 – Interface gráfica - Menu com opções disponíveis para os usuários 65
Figura 28 – Interface gráfica - Abas do sistema 66
Figura 29 – Interface gráfica - Cadastro de clientes 67
Figura 30 – Interface gráfica - Cadastro de animais 68
Figura 31 – Interface gráfica - Atendimentos 69
Figura 32 – Interface gráfica - Tela de atendimento com uma ficha clínica 71
Figura 33 – Interface gráfica - Atividade dedicada ao odontograma 73
Figura 34 – Interface gráfica - Tela com representações gráficas da maxila e mandíbula, cavidades orais e crânios 74
Figura 35 – Interface gráfica - Imagens do atendimento 75

Figura 36 – Interface gráfica - Seleção de um atendimento com opção de exportação . . 76
Figura 37 – Robustez - Criar usuário 77
Figura 38 – Robustez - Gerenciar permissões de usuário 78
Figura 39 – Robustez - Editar conta de usuário 78
Figura 40 – Robustez - Cadastrar clínica 79
Figura 41 – Robustez - Cadastrar clientes 79
Figura 42 – Robustez - Cadastrar animais 80
Figura 43 – Robustez - Realizar atendimento 81
Figura 44 – Robustez - Exportar fichas de atendimento 82
Figura 45 – Etapas de desenvolvimento dos casos de uso 85
Figura 46 – Organização estrutural do banco de dados 86
Figura 47 – Odontograma com possibilidade de desenho e escolha de cor 88

LISTA DE QUADROS

Quadro 1 – Planos e preços do sistema BensVET 23
Quadro 2 – Requisitos Funcionais do sistema 37
Quadro 3 – Requisitos Não-Funcionais do sistema 38
Quadro 4 – Caso de Uso 01 - Criar e vincular usuários 39
Quadro 5 – Caso de Uso 02 - Gerenciar permissões de usuário 40
Quadro 6 – Caso de Uso 03 - Editar conta de usuário 41
Quadro 7 – Caso de Uso 04 - Cadastrar clínica 42
Quadro 8 – Caso de Uso 05 - Cadastrar clientes 43
Quadro 9 – Caso de Uso 06 - Cadastrar animais 44
Quadro 10 – Caso de Uso 07 - Realizar atendimento 45
Quadro 11 – Caso de Uso 08 - Preencher fichas de atendimento clínico e de procedimento 46
Quadro 12 – Caso de Uso 09 - Preencher odontogramas, cavidades orais e crânios . . . 47
Quadro 13 – Caso de Uso 10 - Inserir observações de atendimento no prontuário 48
Quadro 14 – Caso de Uso 11 - Inserir imagens no atendimento 49
Quadro 15 – Caso de Uso 12 - Exportar fichas de atendimento 50
Quadro 16 – Descrição das classes do modelo de domínio 57
Quadro 17 – Grupos de dados da ficha de exame clínico 112
Quadro 18 – Grupos de dados da ficha de procedimento 114

LISTA DE ABREVIATURAS E SIGLAS

API *Application Programming Interface*

CSS *Cascading Style Sheets*

CFMV Conselho Federal de Medicina Veterinária

CRMV Conselho Regional de Medicina Veterinária

DP Densidade de pixel

GB *Gigabyte*

HTML *HyperText Markup Language*

IDE *Integrated Development Environment*

MVVM *Model-view-viewmodel*

PDF *Portable Document Format*

RS Rio Grande do Sul

RAM *Random Access Memory*

SRD Sistema de Referência do Dispositivo

SRU Sistema de Referência do Universo

UML *Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO 13
1.1 OBJETIVOS 13
1.1.1 Objetivos Gerais 13
1.1.2 Objetivos Específicos 14
1.2 ESTRUTURA DO TRABALHO 14

2 REFERENCIAL TEÓRICO 15
2.1 ODONTOLOGIA VETERINÁRIA 15
2.1.1 Exames e fichas odontológicas 15
2.1.2 Anatomia Dentística e Odontograma 17
2.1.2.1 Cavidades Orais 18
2.1.2.2 Representações Gráficas em fichas 20
2.2 *SOFTWARES* DISPONÍVEIS NO MERCADO 21
2.3 DISPOSITIVOS MÓVEIS 24
2.4 PLATAFORMAS MÓVEIS 24
2.4.1 Sistema Operacional Android 25
2.4.2 Sistema Operacional iOS 25
2.5 DESENVOLVIMENTO NATIVO 26
2.5.1 Java 27
2.5.2 Kotlin 27
2.5.3 Objective-C 27
2.5.4 Swift 27
2.6 DESENVOLVIMENTO HÍBRIDO 27
2.6.1 Flutter 28
2.6.2 Ionic 29
2.6.3 React Native 29
2.6.4 Xamarin 29
2.7 MÉTODOS DE ARMAZENAMENTO 30
2.7.1 Armazenamento local 30
2.7.2 Armazenamento em rede com servidor próprio 30
2.7.3 Armazenamento em nuvem 31
2.7.4 Firebase 32
2.8 PROCESSO DE DESENVOLVIMENTO 32

3 PROPOSTA DE SOLUÇÃO 34

4 PROJETO DO SISTEMA 35
4.1 ARQUITETURA DO SISTEMA 36
4.2 REQUISITOS DO SISTEMA 37
4.2.1 Requisitos Funcionais 37
4.2.2 Requisitos Não-Funcionais 37
4.3 CASOS DE USO 38
4.4 MODELO DE DOMÍNIO 51
4.5 INTERFACES DE TELA DO APLICATIVO 60
4.5.1 Criação de usuário administrador 60
4.5.2 Menu de opções do administrador 61
4.5.3 Dados da clínica 62
4.5.4 Gerenciamento de usuários 63
4.5.5 Autenticação de usuário no sistema 64
4.5.6 Dados de usuário 65
4.5.7 Abas do sistema 66
4.5.8 Cadastro de clientes 67
4.5.9 Cadastro de animais 68
4.5.10 Realização de atendimento 69
4.5.11 Preenchimento de ficha de atendimento 70
4.5.12 Representações gráficas 72
4.5.13 Inserção e visualização de imagens no atendimento 75
4.5.14 Exportação de ficha de atendimento 76
4.6 DIAGRAMAS DE ROBUSTEZ 77

5 IMPLEMENTAÇÃO 83
5.1 INSTALAÇÃO DAS FERRAMENTAS DE DESENVOLVIMENTO 83
5.2 COMPONENTES E ARQUITETURA DO PROJETO 84
5.3 ETAPAS DE DESENVOLVIMENTO 84

6 CONCLUSÃO 91

REFERÊNCIAS 93

ANEXO A – MANUAL DE UTILIZAÇÃO DO APLICATIVO 96

ANEXO B – DADOS DE PREENCHIMENTO DA FICHA DE EXAME CLÍNICO 112

ANEXO C – DADOS DE PREENCHIMENTO DA FICHA DE PROCEDIMENTO 114

1 INTRODUÇÃO

Algumas clínicas veterinárias especializadas em atendimentos odontológicos ainda utilizam métodos tradicionais de registros de atendimentos aos pacientes, e por esse motivo, não dispõem da informatização de conteúdo. Desse modo, exames e coletas de dados são registrados em fichas clínicas e formulários impressos.

Para melhor entender o processo, analisou-se um relato feito por uma médica veterinária chamada Manoela Maria Bianchi, registrada no Conselho Regional de Medicina Veterinária (CRMV) do estado do Rio Grande do Sul (RS), pelo número 13.163. Através da mesma, que trabalha em uma clínica veterinária de Caxias do Sul - RS, disponibilizando atendimentos voltados à odontologia para animais, constatou-se que nos atendimentos, utiliza-se um prontuário clínico. Nesses, são registradas todas as informações de alterações encontradas no momento de uma consulta, ou de um tratamento, a fim de manter um histórico bucal do paciente.

Nessa clínica são atendidos animais de diferentes tipos, sendo os mais comuns: cães, gatos, mustelídeos, lagomorfos e roedores. Os prontuários destes animais incluem em sua estrutura, o registro de exame clínico geral e específico, exames solicitados, orçamentos, diagnósticos, planos de tratamento, e condutas realizadas em um atendimento. Todo o registro destes atendimentos é feito de forma manuscrita, sem ter auxílio de um sistema computacional que ofereça ferramentas digitais específicas para seu caso de uso.

Mediante as informações coletadas, são apresentados tópicos específicos da especialidade de Odontologia Veterinária, que são relevantes para serem compreendidos em um estudo de projeto para a informatização do registro dos atendimentos.

Nesse projeto, para atingir os objetivos, foi realizada uma pesquisa bibliográfica referente aos assuntos pertinentes, e por meio deles, desenvolveu-se um aplicativo nativo para *smartphones* da plataforma Android. O mesmo conta com persistência de dados hospedados em um serviço na nuvem, e suas etapas de desenvolvimento foram conduzidas pela metodologia Iconix.

1.1 OBJETIVOS

As Seções 1.1.1 e 1.1.2 apontam os objetivos do trabalho.

1.1.1 Objetivos Gerais

Propor um aplicativo focado em dispositivos móveis, sendo *smartphones* da plataforma Android, com intuito de informatizar o registro dos atendimentos feitos em clínicas especializadas em Odontologia Veterinária.

1.1.2 Objetivos Específicos

1. Compreender as necessidades de ferramentas digitais para profissionais de Odontologia Veterinária.
2. Compreender conceitos relacionados à área de Odontologia e Medicina Veterinária.
3. Desenvolver um aplicativo que resolva o problema apresentado.
4. Obter avaliação à solução entregue aos profissionais.

1.2 ESTRUTURA DO TRABALHO

Os itens a seguir apontam como este trabalho encontra-se organizado:

- No Capítulo 2 descreve-se o estudo da bibliografia associada à Odontologia Veterinária, juntamente aos *softwares* existentes para esta área no mercado. Também são analisadas as plataformas operacionais, e ferramentas de desenvolvimento de sistemas.
- No Capítulo 3 demonstra-se o que se busca desenvolver para atender as necessidades apresentadas.
- No Capítulo 4 determina-se quais ferramentas de desenvolvimento foram empregadas para implementar a solução.
- No Capítulo 5 explica-se o processo de criação do aplicativo, bem como a configuração das ferramentas utilizadas. Descreve-se também, a maneira que foram desenvolvidos os casos de uso, junto aos desafios encontrados.

2 REFERENCIAL TEÓRICO

Neste capítulo apresenta-se a fundamentação teórica necessária do conteúdo da área a ser estudada para a construção do aplicativo. Descreve-se ainda, as tecnologias e metodologias utilizadas para a etapa do desenvolvimento do aplicativo móvel proposto neste projeto.

2.1 ODONTOLOGIA VETERINÁRIA

A história da Odontologia Veterinária vem sendo identificada desde a antiguidade, pois Kertesz (1993) menciona que nas obras de Aristóteles já haviam registros de doenças bucais que eram encontradas em cavalos.

Com o passar dos anos, o estudo de Odontologia na área da Medicina Veterinária veio sendo aprimorado, e fatos marcantes tiveram impactos positivos para contribuir sobre a mesma. No Brasil por exemplo, o Conselho Federal de Medicina Veterinária (CFMV), atribuiu por meio da resolução número 625, de 16 de março de 1995, o reconhecimento da Odontologia Veterinária como título de uma das especialidades da Medicina Veterinária. (BRASIL, 1995).

Para melhor entender o conceito de Odontologia, Gorrel (2010) diz que essa área estuda as doenças de todas as estruturas da cavidade oral, como dentes, tecidos, mucosa oral, glândulas salivares, e etc. Dentro da Medicina Veterinária observa-se que essa abordagem é uma especialidade de grande importância, pois assim, estudos e técnicas são dedicados para trazer cuidados e tratar a saúde bucal dos animais domésticos e em geral.

O dentista veterinário, ou odontologista, nome dado ao profissional responsável, como afirma Silva (2013), empenha a função de tratar doenças na boca dos animais atendidos, pois eles também, em sua região oral podem estar sujeitos a serem afetados por problemas e doenças graves.

Assim sendo, para fazer um diagnóstico de doenças, fatores devem ser analisados antes de aplicar um tratamento, e para isso, consultas e exames são utilizados.

2.1.1 Exames e fichas odontológicas

Quando se busca atendimento hospitalar, comumente uma série de condutas são tomadas durante o atendimento para melhor identificar os problemas apresentados.

Para os animais, quando são atendidos, Gioso (2003) diz que como em todo exame clínico, o exame odontológico no animal deve ser composto por anamnese completa e exame físico geral, e que além disso, a cavidade oral também deve ser examinada.

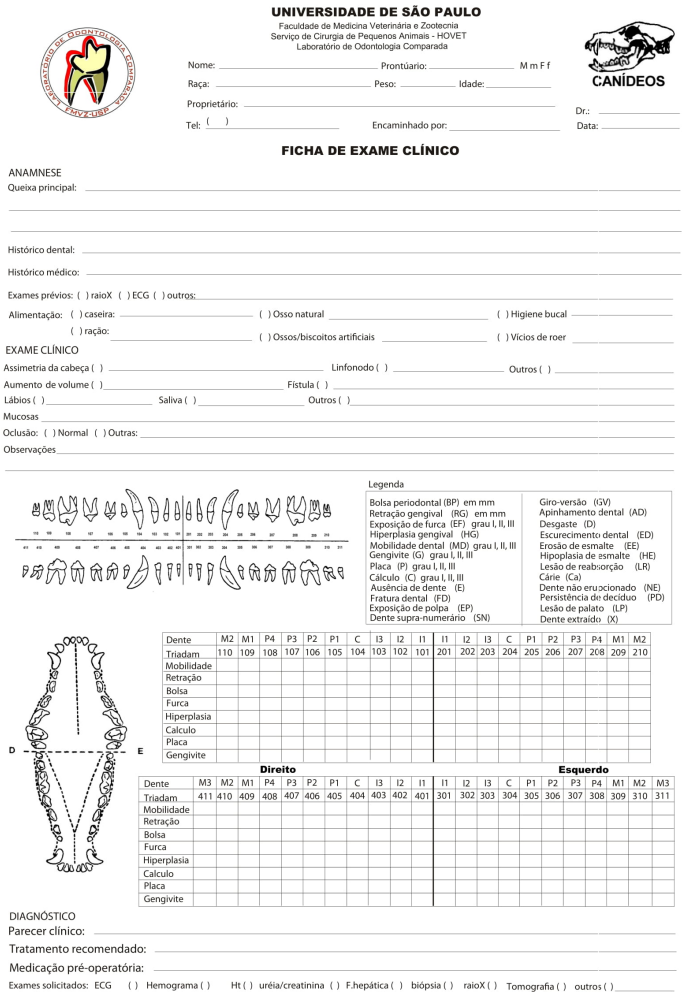
A anamnese, conforme explica Barros (2016), é uma entrevista realizada pelo profissi-

onal de saúde aplicada ao seu paciente, com intuito de relembrar e coletar todos os fatos que se relacionam com o problema apresentado, e assim “identificar e obter informações pertinentes sobre o paciente.” Barros (2016, pp. 20).

A coleta de anamnese na Odontologia Veterinária, descrita por Gioso (2003), pode ser classificada em geral e específica, onde na geral, são questionados todos os sintomas encontrados no animal atendido, atentando-se a quadros mórbidos já diagnosticados ou semelhantes aos atuais. Já na específica, Gioso (2003) cita alguns itens que são avaliados, como: tipo de alimentação, vícios de roer ou morder objetos duros, halitose, higienização bucal, tratamentos dentais prévios, dentre outros.

Ao examinar a parte oral, Gioso (2003) explica que todas alterações ou lesões devem ser registradas em fichas específicas feitas para odontologia, e de preferência, usando esquemas de dentição, conhecidos como odontogramas. Exemplifica-se uma ficha de atendimento clínico através da Figura 1.

Figura 1 – Ficha odontológica de exame clínico canina



UNIVERSIDADE DE SÃO PAULO
Faculdade de Medicina Veterinária e Zootecnia
Serviço de Cirurgia de Pequenos Animais - HOVET
Laboratório de Odontologia Comparada

CANÍDEOS

Nome: _____ Prontúario: _____ M m F f
Raça: _____ Peso: _____ Idade: _____
Proprietário: _____ Dr.: _____
Tel: ( ) _____ Encaminhado por: _____ Data: _____

FICHA DE EXAME CLÍNICO

ANAMNESE
Queixa principal: _____
Histórico dental: _____
Histórico médico: _____
Exames prévios: ( ) raioX ( ) ECG ( ) outros: _____
Alimentação: ( ) caseira: _____ ( ) Osso natural _____ ( ) Higiene bucal _____
( ) ração: _____ ( ) Ossos/biscoitos artificiais _____ ( ) Vícios de roer _____

EXAME CLÍNICO
Assimetria da cabeça ( ) _____ Linfonodo ( ) _____ Outros ( ) _____
Aumento de volume ( ) _____ Fístula ( ) _____
Lábios ( ) _____ Saliva ( ) _____ Outros ( ) _____
Mucosas _____
Oclusão: ( ) Normal ( ) Outras: _____
Observações _____

Legenda

Bolsa periodontal (BP) em mm	Giro-versão (GV)
Retração gengival (RG) em mm	Apinhamento dental (AD)
Exposição de furca (EF) grau I, II, III	Desgaste (D)
Hiperplasia gengival (HG)	Escurecimento dental (ED)
Mobilidade dental (MD) grau I, II, III	Erosão de esmalte (EE)
Gengivite (G) grau I, II, III	Hipoplasia de esmalte (HE)
Placa (P) grau I, II, III	Lesão de reabsorção (LR)
Cálculo (C) grau I, II, III	Cárie (Ca)
Ausência de dente (E)	Dente não erupcionado (NE)
Fratura dental (FD)	Persistência de decíduo (PD)
Exposição de polpa (EP)	Lesão de palato (LP)
Dente supra-numerário (SN)	Dente extraído (X)

Dente	M2	M1	P4	P3	P2	P1	C	I3	I2	I1	I1	I2	I3	C	P1	P2	P3	P4	M1	M2
Triadam	110	109	108	107	106	105	104	103	102	101	201	202	203	204	205	206	207	208	209	210
Mobilidade																				
Retração																				
Bolsa																				
Furca																				
Hiperplasia																				
Calculo																				
Placa																				
Gengivite																				

D E

Direito **Esquerdo**

Dente	M3	M2	M1	P4	P3	P2	P1	C	I3	I2	I1	I1	I2	I3	C	P1	P2	P3	P4	M1	M2	M3
Triadam	411	410	409	408	407	406	405	404	403	402	401	301	302	303	304	305	306	307	308	309	310	311
Mobilidade																						
Retração																						
Bolsa																						
Furca																						
Hiperplasia																						
Calculo																						
Placa																						
Gengivite																						

DIAGNÓSTICO
Parecer clínico: _____
Tratamento recomendado: _____
Medicação pré-operatória: _____
Exames solicitados: ECG ( ) Hemograma ( ) Ht ( ) uréia/creatinina ( ) F.hepática ( ) biópsia ( ) raioX ( ) Tomografia ( ) outros ( ) _____

Fonte: USP (2014)

Observa-se que em uma ficha existe um quadro de legenda com abreviações, onde Gorrel (2013) diz serem frequentemente utilizadas na área, que são de fundamental importância estarem disponíveis para nomear anormalidades ou tratamentos.

Gorrel (2010) afirma que além do uso de odontograma, para se ter um prontuário odontológico mais completo, informações complementares são empregadas com uso de anotações, radiografias e etc.

Como visto, um dos elementos mais importantes que compõem uma ficha clínica, é o odontograma, que faz parte da anatomia dentística, e será abordado na Seção 2.1.2

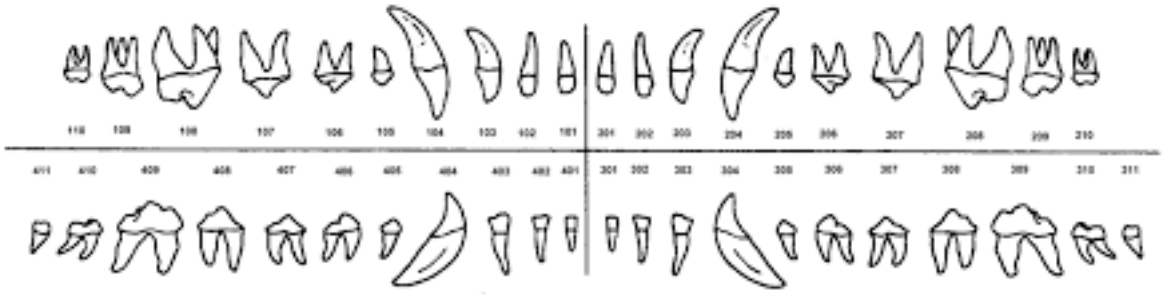
2.1.2 Anatomia Dentística e Odontograma

Através de Silva (2013), pode-se entender que a anatomia dentística é de fundamental importância, pois ajuda a interpretar todos os itens que compõem a cavidade oral, e também a diferenciar os tipos de dentes que formam a arcada dentária. Ela também auxilia para saber como eles estão ordenados dentro da cavidade oral, e como utilizar os sistemas de notação dental.

Assim, para auxiliar na visualização e registro de consultas e procedimentos, o odontograma é essencial, e nele Gorrel (2010) explica que o mesmo é uma imagem que representa a arcada dentária do paciente, onde estão ilustrados todos os seus dentes, servindo para desenhar ou marcar achados e tratamentos. Nisso, cada dente é nomeado com um número, letras maiúsculas, ou pares numéricos. Independente da forma empregada, ela segue um padrão definido, e poderá ser entendido por qualquer dentista profissional.

A Figura 2 ilustra como se organiza uma estrutura dentária, através do exemplo de odontograma de um cachorro.

Figura 2 – Odontograma do cão



Fonte: Gioso (2007)

Lobprise & Dodd (2019) consideram que odontogramas podem variar desde papéis impressos, até sistemas eletrônicos, e que esses devem ser específicos para cada espécie de animal, pois as dentições mudam.

2.1.2.1 Cavidades Orais

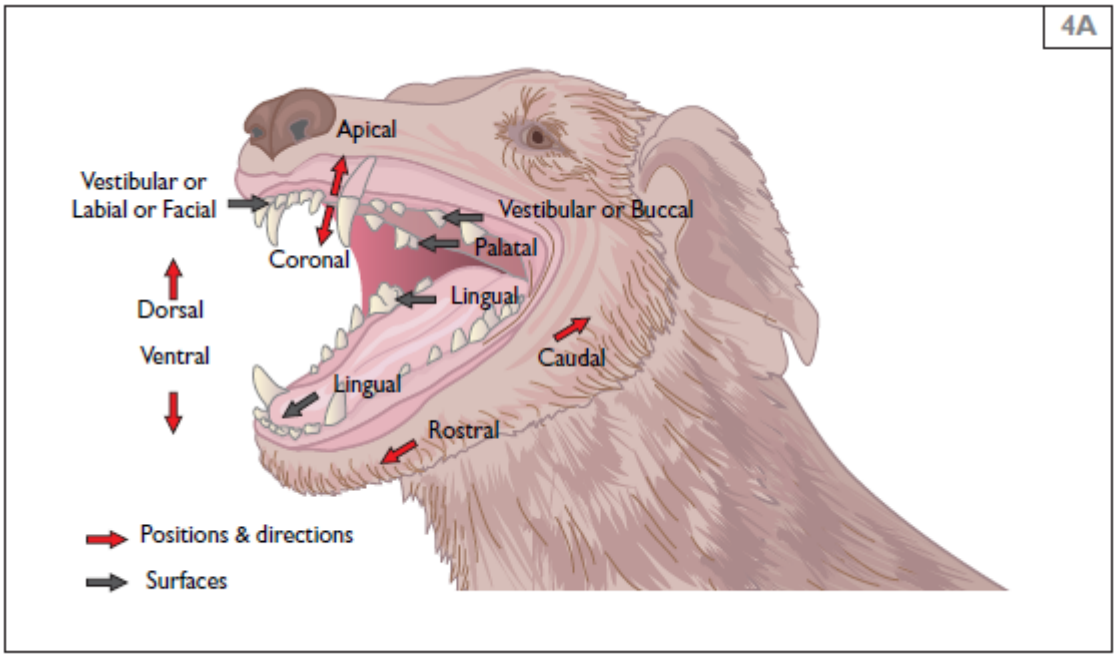
Na anatomia bucal estão inclusas as cavidades orais, existindo diferentes tipos de representações gráficas da mesma.

O conceito de cavidade oral, segundo Lobprise & Dodd (2019), consiste na área que se estende desde os lábios até a faringe oral no nível da tonsila palatina (amígdalas), ilustrados na Figura 3. É limitada pelos lábios, e lateralmente pelas bochechas. Dorsalmente limita-se pelos palatos moles e duros, e ventralmente pelo assoalho da boca.

A cavidade oral, como explicada por Lobprise & Dodd (2019), geralmente é subdividida em duas partes, o vestíbulo, e a cavidade oral propriamente dita.

A Figura 3 mostra por meio de uma representação gráfica, um exemplo das áreas que compõem a cavidade oral de um cachorro.

Figura 3 – Cavidade oral do cachorro

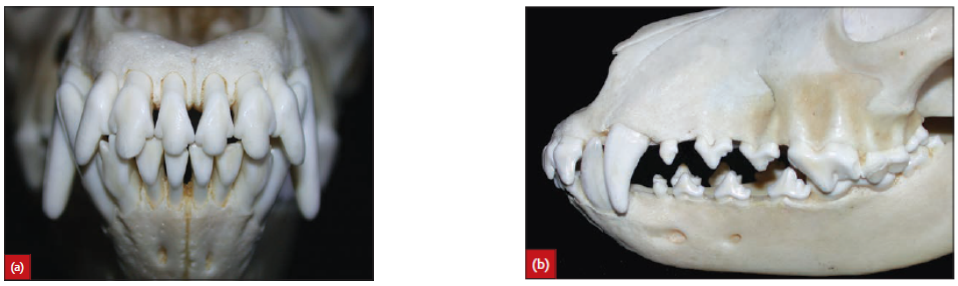


Fonte: (NIEMIEC, 2011, pp. 15)

O vestíbulo é definido pelo espaço entre os lábios, bochechas, dentes, gengivas, e cristas alveolares. Já a cavidade oral propriamente dita estende-se desde a crista alveolar e os dentes à faringe oral. Adicionalmente, é definido pelo céu e assoalho da boca, e preenchido pela língua.

Por meio da cavidade oral, utilizam-se ilustrações que são úteis em registros de atendimentos. Em Reiter & Gracis (2018) pode-se observar que as mesmas podem ser: desenhos laterais do crânio nos lados esquerdo e direito, parte frontal, palatinal e lingual, pois nelas estão presentes todas as formas de visualizar cada estrutura bucal, dentes, e as suas faces. Na Figura 4, por exemplo, exibem-se as diferentes formas de representar um crânio.

Figura 4 – Anatomia do crânio canina

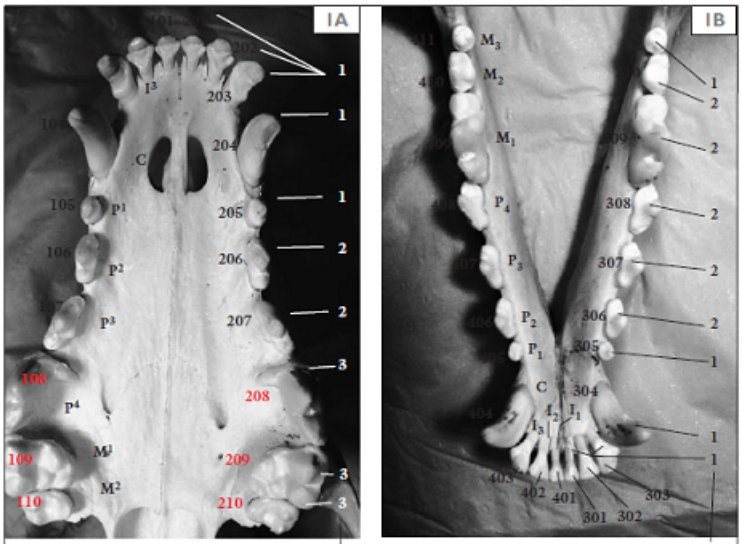


(a) Frontal (b) Lateral

Fonte: Reiter & Gracis (2018)

Há também uma outra representação utilizada nas fichas dos animais, onde ela exibe a anatomia dental através da visão da maxila e mandíbula. A Figura 5 demonstra essa visão com um exemplo real, ilustrando na parte esquerda a maxila, e à direita a mandíbula, ambas de um cachorro.

Figura 5 – Anatomia da maxila e mandíbula canina



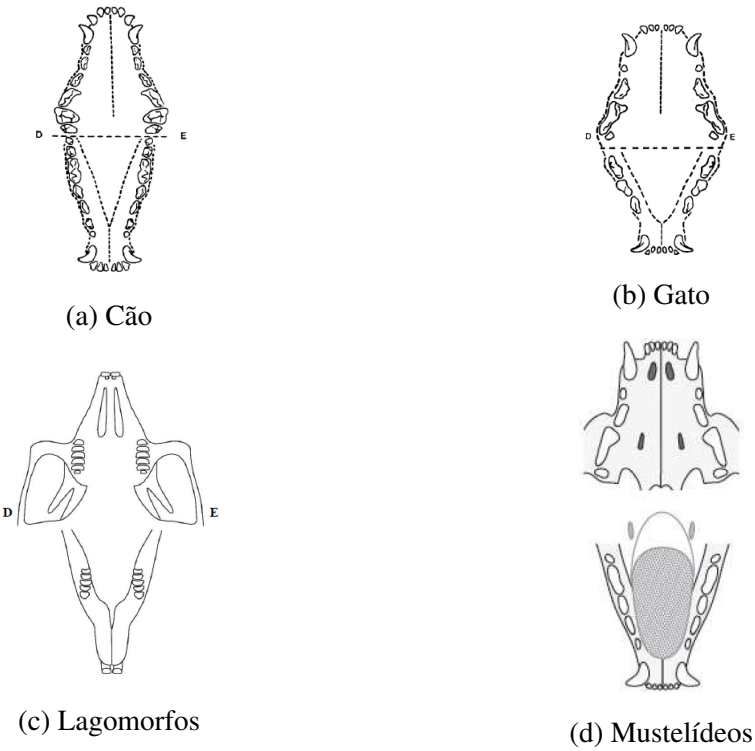
I Modified Triadan numbering system in the dog. **A:** Maxilla; **B:** Mandible. (Three-rooted teeth in red.)

Fonte: Niemiec (2011, pp. 11)

2.1.2.2 Representações Gráficas em fichas

O formato de representação da maxila e mandíbula é amplamente utilizado nas fichas de diferentes espécies de animais, e a Figura 6 exemplifica.

Figura 6 – Representações gráficas da maxila e mandíbula de diferentes espécies.



(a) Cão

(b) Gato

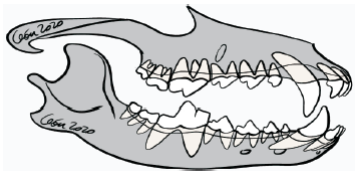
(c) Lagomorfos

(d) Mustelídeos

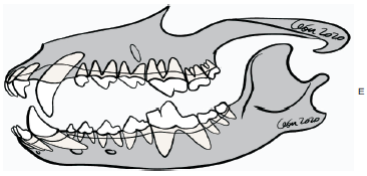
Fonte: Extraído de USP (2014)

Do mesmo modo, são utilizadas as representações do crânio do animal por meio de desenhos, como evidenciados alguns dos seus tipos através da Figura 7. Leon (2020) diz que o desenho do crânio permite ao profissional fazer anotações sobre alterações buco-maxilares e dentárias. Já sobre a ilustração da boca aberta, representada na Figura 7d, Leon (2020) explica que ela permite ao profissional apontar alterações em tecidos moles, tais como, o palato, gengiva e também as mucosas orais.

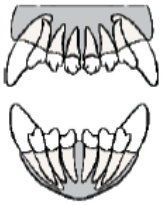
Figura 7 – Representações gráficas do crânio e da cavidade oral canina



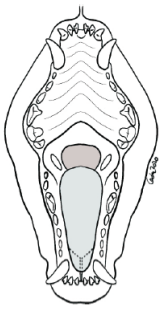
(a) Lado direito



(b) Lado esquerdo



(c) Frontal



(d) Aberta

Fonte: Extraído de Leon (2020)

Como visto, fichas clínicas são usadas comumente em meio impresso, e Lobprise & Dodd (2019) ressaltam que também se empregam em meios eletrônicos. Para conhecer esta última opção, busca-se conhecer um pouco das opções que o mercado oferece. Esta abordagem será vista na Seção 2.2

2.2 *SOFTWARES* DISPONÍVEIS NO MERCADO

Não somente os instrumentos e métodos de atendimentos odontológicos evoluíram com o passar dos anos, tecnologias e aplicações computacionais também vieram acompanhando o processo de evolução. Dessa maneira, soluções profissionais foram sendo disponibilizadas no mercado de aplicativos móveis, e dentre elas, estão aplicações voltadas a consultórios veterinários, que podem ser desde gerenciadores de clínicas, até propostas mais completas com inclusão de registros de consultas médicas.

Porém, o maior problema, é que esses sistemas não possuem muita popularidade no Brasil, tendo maiores opções disponíveis em outros países, o que acaba muitas vezes por dificultar o acesso e uso dos mesmos por não possuírem tradução para a língua portuguesa, ou pelo método de pagamento não ser da mesma moeda Brasileira.

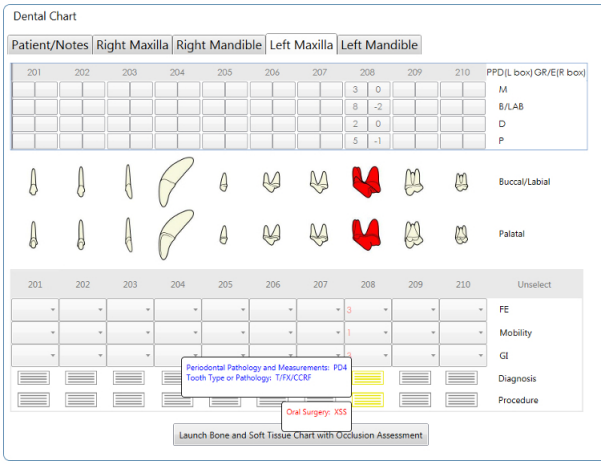
Consultando-se na internet, usando o termo em inglês "*software for veterinary dental chart*", imediatamente observa-se algumas opções de sistemas veterinários com função de prontuário odontológico. Dentre eles encontrou-se exemplos como o *software* "Animal Dental Chart" e "Vet Dental Charts", para as plataformas Windows, e Web, respectivamente. As figuras 8 e 9 mostram as imagens extraídas destes sistemas nos seus sites.

Figura 8 – *Software* Easy Vet Dental Charts



Fonte: Extraído de Vet Dental Charts (2021)

Figura 9 – *Software* Animal Dental Chart

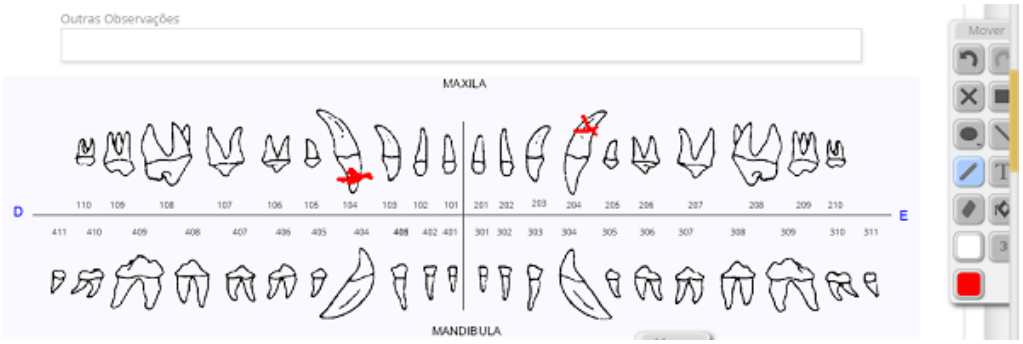


Fonte: Extraído de Animal Dental Chart Inc. (2021)

Contudo, os exemplares encontrados estão na língua inglesa, além de não possuírem suporte como aplicativo para a plataforma Android.

Já no Brasil, por exemplo, fazendo uma pesquisa por sistemas de Odontologia Veterinária, encontrou-se o BensVET, que dispõe de um prontuário odontológico com odontograma, que pode ser visto na Figura 10, onde o mesmo suporta ferramentas de desenho.

Figura 10 – Odontograma no *software* BensVET



Fonte: Extraído de BensVET (2021)

Mesmo que existam algumas opções presentes no mercado, observa-se que elas não são totalmente de uso gratuito, tendo apenas um período de testes, ou nos casos de serem gratuitas, oferecem recursos limitados, o que acaba inviabilizando a adoção de um sistema por não suprirem as necessidades. Para exemplificar, o Quadro 1 mostra quais os valores e funcionalidades existentes nos planos do sistema BensVET.

Quadro 1 – Planos e preços do sistema BensVET

Nome do plano	**Preço**	**Características**
Plano Grátis	Gratuito	Funcionalidades limitadas e aplicativo não incluso.
Plano Básico	R$197,00/mês	Maior número de funcionalidades liberadas, e aplicativo não incluso.
Plano Premium	R$327,00/mês	Plano em oferta, com todas as funcionalidades liberadas e aplicativo incluso.
Premium Plus	R$417,00/mês	Todas as funcionalidades liberadas com aplicativo incluso.

Fonte: Extraído de BensVET (2021)

Adicionalmente, o que acaba sendo outro fator inconveniente nos planos gratuitos, é que não oferecem aplicativos para celulares, fazendo com que seu uso seja exclusivo por meio de navegadores de internet, ou seja, são de plataforma Web, e sua usabilidade se torna melhor em computadores.

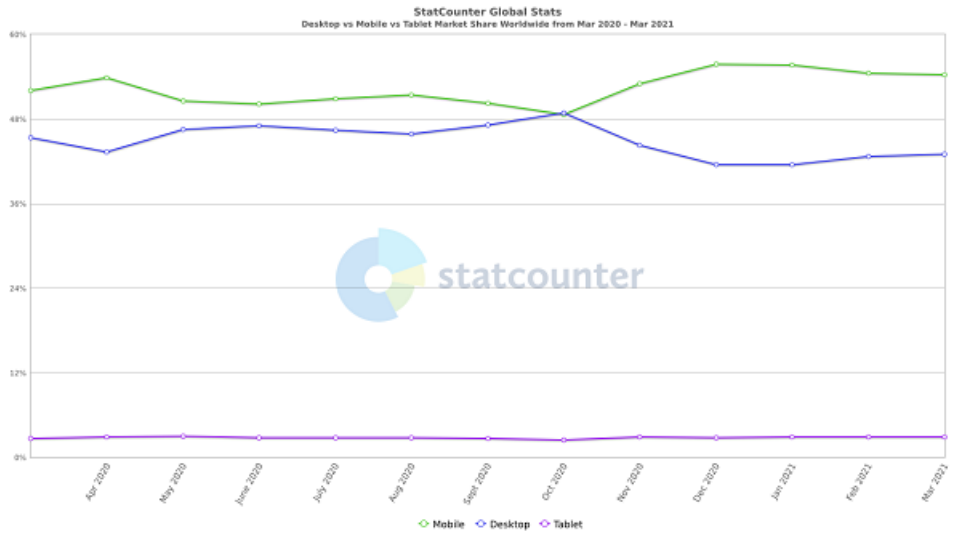
Por esses motivos, a criação de soluções inovadoras é relevante para aumentar o número de opções de sistemas disponíveis no mercado Brasileiro, e ampliar a compatibilidade entre as plataformas operacionais para os mesmos. Para tanto, ao pretender criar novas propostas, é preciso entender o conceito de desenvolvimento para um tipo de equipamento que será fornecida a ferramenta, que neste caso, serão dispositivos móveis.

2.3 DISPOSITIVOS MÓVEIS

Com o passar dos anos, a tecnologia acabou tornando-se algo totalmente presente em nosso dia a dia, principalmente quando se trata de dispositivos móveis, sejam eles relógios inteligentes, celulares ou *tablets*. Com isso, percebe-se que a quantidade de aplicativos nas lojas está aumentando gradativamente para oferecer cada vez mais funcionalidades e utilidades para auxiliar em tarefas cotidianas.

Diante disso, a maior evidência que se pode observar, é que o *smartphone* fez-se indispensável, liderando o uso dos aparelhos móveis, como observa-se no gráfico da Figura 11, extraído de StatCounter (2021a).

Figura 11 – Comparativo de vendas de dispositivos móveis



Fonte: StatCounter (2021a)

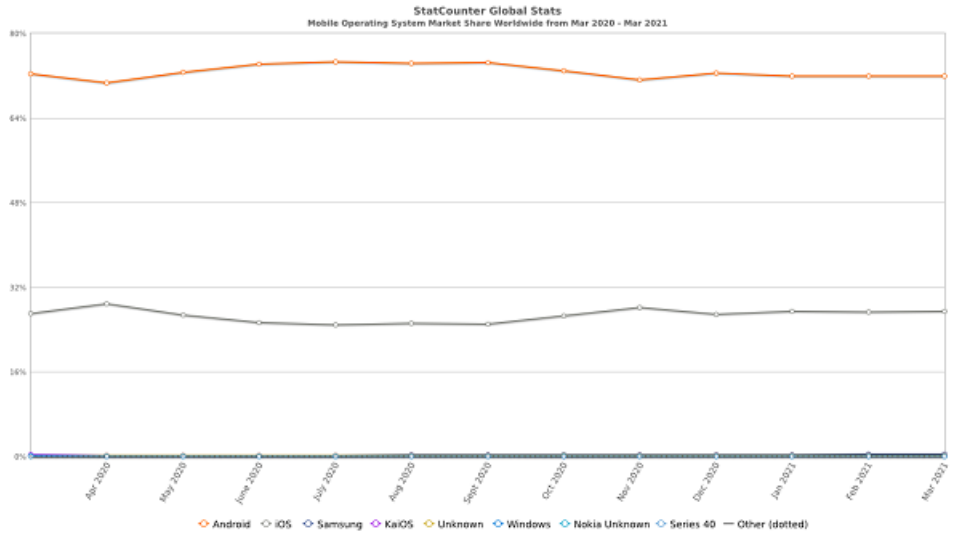
Este fato é marcado pelo *smartphone* por ele conseguir englobar e até dispensar o uso de outros tipos de dispositivos, pois é capaz de incorporar um maior número de utilidades, e trazer versatilidade e mobilidade, visto que grande parte das funções podem ser realizadas através dele.

2.4 PLATAFORMAS MÓVEIS

Na atualidade, como observado na Figura 12, obtida em StatCounter (2021b), as duas plataformas que dominam o mercado do segmento *mobile*, são os sistemas Android (Google), seguido pelo iOS (Apple).

Por meio deste fator, serão consideradas as duas plataformas líderes para estudar o desenvolvimento do trabalho.

Figura 12 – Plataformas operacionais mais vendidas no mercado



Fonte: StatCounter (2021b)

Diante desta análise, presume-se que os desenvolvedores Android possam ter maiores vantagens e incentivos para produzir seus projetos, visto que a maior parte dos dispositivos móveis vendidos no mercado são liderados pela sua plataforma, e com isso, terão mais dispositivos disponíveis para utilizar suas aplicações.

2.4.1 Sistema Operacional Android

Conforme Monteiro (2013), o Android é um sistema operacional de código aberto baseado em Linux, e foi inicialmente desenvolvido pela empresa Android Inc. Dois anos após a sua fundação, a empresa Google adquiriu o projeto, criando a Open Handset Alliance, um grupo de empresas de tecnologia, que buscam desenvolver uma plataforma aberta e gratuita, com foco para dispositivos móveis.

Atualmente, segundo Google (2021), o sistema operacional conta com mais de 2,5 bilhões de dispositivos ativos. Como o foco do sistema são os dispositivos móveis, pode-se encontrar a plataforma instalada em *smartphones*, *tablets*, relógios, aparelhos de TV e em multimídias de carros.

A plataforma conta com uma loja oficial de conteúdo chamada Google Play, e oferece cerca de mais de dois milhões de aplicativos para os dispositivos Android, criados por desenvolvedores do mundo inteiro. Através dela, é possível encontrar desde jogos, até conteúdos digitais, como livros, filmes, músicas, e também aplicativos mais sofisticados de uso profissional.

2.4.2 Sistema Operacional iOS

O iOS, como descrito por Steil (2012), foi introduzido em 2007 com o lançamento do iPhone, sendo um sistema operacional rodando dentro do dispositivo, que inicialmente se

chamava de iPhone OS. Passando-se os anos, com a criação de novos dispositivos pela Apple, o sistema operacional mudou a sua nomenclatura para iOS. Hoje, como detalha Apple (2021), o iOS encontra-se disponível na versão 14, sendo esta a mais recente.

O sistema possui uma loja de aplicativos chamada App Store, que alcança mais de 1,5 bilhão de aparelhos, em 175 regiões. A loja fornece conteúdos e aplicativos digitais, e atualmente, estão disponíveis mundialmente mais de 1,8 milhão de aplicativos.

Ao pretender realizar o desenvolvimento de aplicações para um tipo de plataforma, deverão ser previamente avaliados dois modelos de desenvolvimento existentes, que são híbrido e nativo, abordados nas Seções 2.5 e 2.6.

2.5 DESENVOLVIMENTO NATIVO

Na abordagem de desenvolvimento nativa, Panhale (2016) descreve que as aplicações são desenvolvidas utilizando exclusivamente o código puro das linguagens suportadas na plataforma que se pretende entregar a aplicação, usando instrumentos por ela oferecidas, como uma *Integrated Development Environment (IDE)*, que combina ferramentas de programação em um único *software*. Com isso, múltiplas funções nativas são exploradas através de uma *Application Programming Interface (API)*, responsável por fazer a comunicação entre os seus recursos.

Em Meier & Lake (2018), é explicado que no Android, as linguagens de programação Java e Kotlin são definidas como padrão para o desenvolvimento, juntamente à ferramenta Android Studio, sua IDE oficial. Já no caso do iOS, Silveira & Jardim (2014) caracterizam que as linguagens de programação podem ser Objective-C ou Swift, com uso da IDE XCode.

Através de uma aplicação feita de forma nativa, Panhale (2016) ressalta que pode-se encontrar vantagens e desvantagens em seu uso e desenvolvimento. Como vantagens, encontram-se:

- Aproveitamento de recursos nativos das plataformas dos dispositivos.
- Maior desempenho e performance.
- Maior quantidade de APIs disponíveis.

Já como desvantagens, Panhale (2016) diz que, como é necessário ter um código específico para cada plataforma, por consequência haverá um aumento de tempo e custo de desenvolvimento para produzir ou fazer alterações em cada aplicativo.

Quando um projeto nativo for elaborado, é de importância definir qual linguagem de programação será escolhida para implementar a aplicação. Pode-se conhecer um pouco das que são disponibilizadas para este fim, nas Subseções 2.5.1, 2.5.2, 2.5.3 e 2.5.4.

2.5.1 Java

Como explica Claro & Sobral (2008), Java é uma linguagem de programação baseada na Linguagem C, criada em 1995, pela Sun Microsystems, sendo adquirida pela Oracle posteriormente. Como pode-se ver em Meier & Lake (2018), ela é uma das ferramentas oficiais para criar aplicativos Android por ter grande poder, e também por ser uma das mais conhecidas pelos desenvolvedores. A mesma traz diversas opções de bibliotecas que podem ser utilizadas para funções específicas, a fim de criar aplicações mais robustas.

2.5.2 Kotlin

Kotlin, documentado em Kotlin (2021), é uma linguagem de programação de código aberto, oficialmente lançada em 2016. Em 2017, Kotlin integrou o grupo de linguagens oficiais do sistema Android.

Segundo Meier & Lake (2018), ela tem como característica poder ser usada com os paradigmas orientado a objetos e funcionais, e também poder operar junto com a linguagem Java. O intuito da criação desta linguagem foi de apresentar melhorias, como redução da verbosidade, aumentar a segurança de ponteiros nulos, dentre outras.

2.5.3 Objective-C

Segundo Kaczmarek *et al.* (2018), Objective-C foi criada na década de 1980, sendo uma derivação da linguagem C e o Smalltalk, herdando a sintaxe, instruções, e tipos primitivos. Adicionalmente, ela possui suporte a orientação à objetos, e é utilizada para o desenvolvimento de aplicações para a plataforma iOS, tornando-se popular pelo mesmo motivo.

2.5.4 Swift

Como apresentado em Swift (2021), é uma linguagem de programação desenvolvida pela Apple, lançada em 2014 para a criação de aplicações para iOS. Swift é de código aberto, e pode operar junto à linguagem Objective-C. Foi criada com intuito de prover uma abordagem moderna de segurança, mais desempenho, e padrões de design de *software*.

2.6 DESENVOLVIMENTO HÍBRIDO

Através da filosofia de desenvolvimento híbrida, busca-se criar aplicações feitas para "muitos dispositivos móveis com diferentes tamanhos de tela e fabricantes." (PANHALE, 2016, pp. 18, tradução nossa).

Panhale (2016) explica que as aplicações híbridas são conhecidas por serem multiplataforma, ou seja, compatíveis em mais que uma plataforma ao mesmo tempo. Essa forma de projetar aplicações é feita através de um mesmo código fonte, que geralmente usa tecnologias

simples, como *HyperText Markup Language (HTML)*, *Cascading Style Sheets (CSS)*, e JavaScript, que são fornecidas através de um *framework*. Por isso, aplicativos híbridos podem ser caracterizados como uma fusão de uma aplicação Web e uma aplicação nativa.

Um *framework* utilizado em aplicações híbridas, nas palavras de Sommerville (2019), é caracterizado como um conjunto de ferramentas genéricas que permitem a reusabilidade e facilitam ao desenvolvedor resolver problemas frequentes, não necessitando reescrever códigos, economizando assim, tempo e custo.

Em primeiro momento, essa seria a escolha ideal para viabilizar a compatibilidade entre as plataformas móveis, porém existem pontos de vantagens e desvantagens a serem considerados.

Como características vantajosas, Panhale (2016) aponta o uso de código independente para ambas as plataformas, o que torna o desenvolvimento mais produtivo e prático, e com isso, traz menor custo de desenvolvimento, e maior custo-benefício.

Já como desvantagens, ainda do ponto de vista de Panhale (2016), existem limitações de recursos e menor performance, além de oferecer menor quantidade de APIs, comparado aos recursos nativos do aparelho.

Dessa forma, para auxiliar no desenvolvimento multiplataforma, no mercado existem vários *frameworks* com suas próprias características que podem ser úteis para cada finalidade. Portanto, de acordo com a necessidade do negócio, deverá ser escolhido um *framework* adequado, e dentre os disponíveis, Smith (2021) aponta alguns exemplos como:

- Flutter
- Ionic
- React Native
- Xamarin

Para compreender suas finalidades, analisa-se como cada um deles funciona, nas Seções 2.6.1, 2.6.2, 2.6.3, e 2.6.4.

2.6.1 Flutter

Napoli (2020) descreve que Flutter é um *framework* de código aberto criado pela Google em 2017, sendo relativamente recente, com foco em criar modernos aplicativos para Android e iOS.

Flutter tem como base a linguagem Dart para desenvolver as aplicações. Ao criar um aplicativo, Napoli (2020) explica que o código da linguagem Dart é compilado para a linguagem

base do dispositivo, fazendo com que as aplicações sejam nativas, conseguindo acessar recursos do dispositivo sem que terceiros façam as conversões, trazendo um desempenho maior.

2.6.2 Ionic

Neste *framework*, Cheng (2018) diz que o mesmo é de código aberto, e que utiliza tecnologias de desenvolvimento Web, como AngularJS, TypeScript, HTML5 e JavaScript, voltados para o desenvolvimento híbrido. O Ionic é focado em performance, design e otimizações.

Em Cheng (2018), o autor descreve que as aplicações buscam se aproximar o máximo possível das aplicações nativas, e para isso, a estrutura utiliza *plugins* dos *framework*s Apache Cordova e Capacitor para acessar recursos nativos do sistema operacional.

2.6.3 React Native

Na descrição de Boduch & Derks (2020), React é uma estrutura de desenvolvimento de código aberto JavaScript e React.js criada pelo Facebook, em 2011. Tem por características ser declarativa, eficiente e flexível.

Boduch & Derks (2020), ressaltam que o objetivo deste *framework* é de desenvolver interfaces de usuário para aplicativos. Seu funcionamento baseia-se em renderizar os componentes gráficos para os mesmos da plataforma nativa, sendo indicado para aplicações simples onde não são necessárias funcionalidades mais complexas. Ao requerer acessar os recursos nativos, deverá ser utilizado o código nativo da plataforma.

2.6.4 Xamarin

Segundo Bennett (2018), este *framework* pertence à empresa Microsoft, e é caracterizado por ser de código aberto, com foco em criar aplicativos multiplataforma para Android e iOS com as linguagens .NET e C#. O mesmo possui estrutura para acessar recursos nativos da plataforma.

Bennett (2018) aponta que o mesmo oferece rápida compilação, o que favorece para se encontrar uma menor quantidade de erros de tempo de execução. Facilita também a elaboração de interfaces de um aplicativo, pois as mesmas por ele feitas se assemelham às nativas.

Após conhecer um pouco mais sobre os métodos de desenvolvimento, linguagens de programação e *frameworks* para dispositivos móveis, é necessário também preocupar-se com como armazenar e recuperar as informações criadas pelo usuário. Com isso, avaliam-se as diferentes formas de armazenamento de dados. Esse fator é indispensável, pois é com ele que será encontrada a estratégia mais adequada para garantir a consistência das informações que o usuário manipula.

2.7 MÉTODOS DE ARMAZENAMENTO

Para Simas *et al.* (2019), nos dispositivos móveis existem diferentes formas de persistir os dados do usuário. Dentre elas estão a possibilidade de armazená-los de modo local, através do armazenamento interno do dispositivo, e com uso de bancos de dados relacionais. Também há a possibilidade de utilizar a rede de internet, por meio de conexão com servidor próprio, ou com armazenamento em nuvem.

Cada maneira de armazenamento de dados deverá ser escolhida de acordo com a necessidade, e para melhor conhecê-las, as Subseções seguintes irão descrever suas características.

2.7.1 Armazenamento local

Simas *et al.* (2019) diz que no armazenamento local, utiliza-se diretamente o próprio armazenamento interno do dispositivo para guardar e recuperar informações manipuladas pelo usuário. O autor explica que desta forma, armazenam-se as informações por meio de bancos de dados, atributos chave-valor, arquivos e documentos.

Nessa abordagem, do ponto de vista de Huria (2018), por se utilizar o próprio dispositivo para manipular as informações e operações solicitadas pelo aplicativo, o processo é feito de forma mais rápida e com menos riscos, pois não utiliza conexão com a internet. Deste modo, todo o processamento é executado de forma *offline*, com recursos sempre disponíveis.

Existem também desvantagens por meio do emprego deste tipo de armazenamento, pois Huria (2018) ressalta que os dados estão salvos de forma local no próprio dispositivo, fazendo-se necessário haver uma forma de *backup* dos mesmos. Logo, uma vez que haja troca do dispositivo, as informações serão perdidas se não forem possíveis de serem recuperadas, além de impossibilitar também o acesso remoto aos dados.

2.7.2 Armazenamento em rede com servidor próprio

Neste contexto de armazenamento, observa-se através de Diamond (2020), que é preciso possuir a própria infraestrutura de servidores que irão armazenar os dados das aplicações, ficando estes sob responsabilidade do mantenedor.

Nos dispositivos móveis, esses servidores geralmente são acessados por meio de uma API, que "fazem a intermediação entre o aplicativo e o servidor de banco de dados, sem a necessidade de usar consultas SQL, de modo direto, no aplicativo." (SIMAS *et al.*, 2019, pp. 144).

Por meio do armazenamento com emprego de conexão à servidores próprios, também deve-se observar se há vantagens e desvantagens, a fim de auxiliar na decisão do uso deste método.

Diamond (2020) explica que como vantagens, há um maior controle e segurança dos dados, por ter-se a própria infraestrutura de armazenamento. Junto disso, há também a confiabilidade de que as informações estão sob domínio da própria responsabilidade. Desta maneira, mesmo que haja a desinstalação do aplicativo, ela não afetará a existência dos dados, pois eles ficarão guardados nas dependências dos servidores.

Porém, algumas observações precisam ser examinadas antes de se utilizar esta abordagem, visto que é necessário fazer investimentos que trazem custos para o empreendimento, como, dar manutenção aos equipamentos, prevenir falta de energia elétrica para mantê-los sempre disponíveis, e manter cópias de segurança dos dados armazenados em casos de possíveis desastres.

2.7.3 Armazenamento em nuvem

Neste conceito, Diamond (2020) explica que os dados dos usuários são colocados em servidores gerenciados por uma empresa especializada, que assume todos os recursos de suporte, como o *hardware* e infraestrutura.

Para se utilizar esta opção, deve-se buscar organizações que fornecem este tipo de serviço. Deve-se considerar também o plano de pagamento, que normalmente poderá ser anual, mensal, ou de acordo com um limite de armazenamento. Examina-se adicionalmente se o valor do custo do serviço contratado vale mais do que se ter uma infraestrutura própria.

Com este método, Diamond (2020) explica que por necessitar de conexão ativa com a internet, deve-se conhecer as formas de segurança que os serviços empregam para proteger os dados, dando a garantia de não haver riscos de acessos indevidos, ou de vazamentos. Uma boa conexão à internet também é essencial, pois os usuários poderão enfrentar problemas para acessarem seus arquivos caso haja má qualidade da mesma.

Avaliadas as precauções, ainda em Diamond (2020), o autor destaca alguns pontos positivos encontrados ao se utilizar armazenamento em nuvem:

- Não requer a compra de equipamentos próprios de armazenamento.
- Muitas vezes, o dinheiro gasto com um serviço em nuvem é menor do que possuir a própria infraestrutura.
- Permite o acesso à informação por meio de qualquer dispositivo, através da internet.

No mercado, existem diversas ferramentas de armazenamento em nuvem disponíveis, e Huria (2018) evidencia o Firebase por ser da conhecida empresa Google. Este por sua vez, é um banco de dados em nuvem, que oferece sincronização em tempo real para guardar os dados de usuários. Por meio da Seção 2.7.4, compreende-se algumas informações a respeito do serviço.

2.7.4 Firebase

O Firebase é uma ferramenta em nuvem que auxilia no desenvolvimento de aplicações para diversas plataformas, como Android, iOS e Web. Através de Firebase (2021b), encontram-se diversos serviços úteis disponibilizados para aplicativos. Alguns deles destacados em sua página, são:

- *Realtime Database*: faz a sincronização em tempo real dos dados entre o aplicativo do usuário e o banco de dados do Firebase.
- *Cloud Storage*: fornece armazenamento para documentos e arquivos de mídia do usuário.
- *Authentication*: disponibiliza autenticação de usuários através de senhas, números de telefone, e outros serviços como, conta do Google, Facebook e Twitter.
- *Analytics*: utilizado para obter dados de uso analíticos dos usuários, e relatórios de erros nos aplicativos.

A respeito da plataforma, Wieruch (2019) destaca que a mesma se diferencia dos outros métodos de armazenamento por não necessitar a escrita de uma API específica para se conectar a um servidor de dados. Deste modo, o serviço disponibiliza a sua própria interface, que faz o acesso e o gerenciamento dos seus serviços.

Em Firebase (2021a), notam-se planos de uso que oferecem serviços de forma gratuita, até atingir uma certa capacidade estabelecida. Para sistemas com baixo consumo de dados, a plataforma faz-se vantajosa, pois cobranças somente são realizadas após exceder estes limites, conforme a utilização.

Ao escolher o método de persistência dos dados da aplicação, deseja-se também obter qualidade e bons resultados no produto. Para tanto, boas práticas devem ser empregadas no decorrer do desenvolvimento. A fim de planejar e projetar a arquitetura que irá constituir o sistema, utiliza-se uma metodologia de desenvolvimento que segue uma linha de padrões predefinidos. Para o caso deste trabalho, empregou-se o modelo de processo de desenvolvimento de *software* Iconix.

2.8 PROCESSO DE DESENVOLVIMENTO

Por meio do processo Iconix, observa-se em Rosemberg, Stephens & COLLINS-COPE (2005), que é obtida uma metodologia do tipo ágil, fazendo uso de um processo simples e unificado. Esta abordagem visa remover as redundâncias da *Unified Modeling Language (UML)*, com um menor emprego de diagramas de projeto.

De forma resumida, pode-se afirmar que: “O Processo Iconix é uma metodologia de análise e design baseada em casos de uso. Seu foco principal é obter de forma confiável, casos

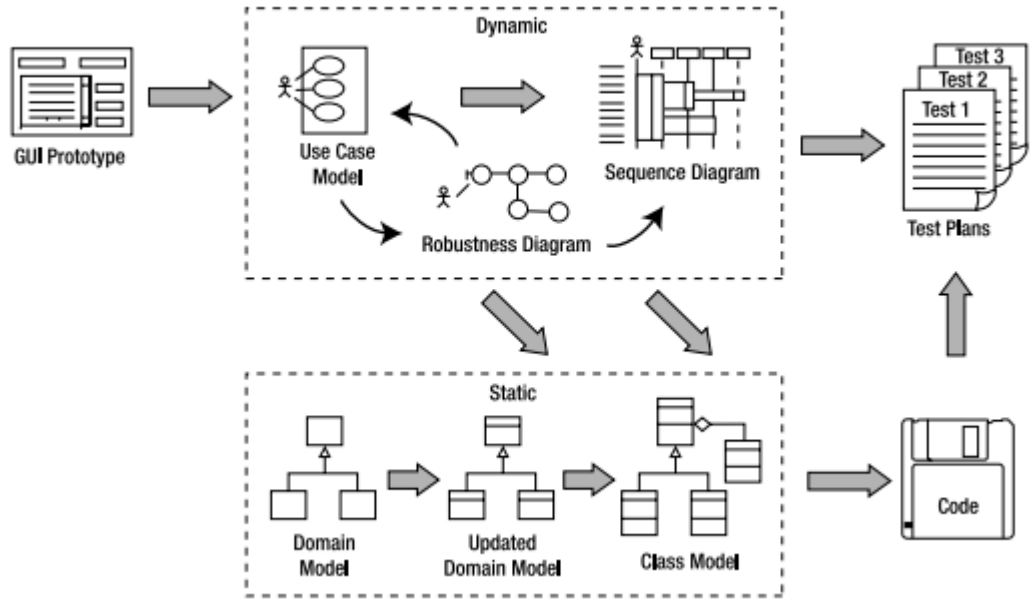
de uso para programação no mínimo de etapas possível." (ROSEMBERG; STEPHENS; COLLINS-COPE, 2005, pp. 3, tradução nossa).

Segundo os autores Rosemberg, Stephens & COLLINS-COPE (2005), o processo faz uso de quatro diagramas:

- Diagramas de classes e domínio: descrevem os objetos e seus relacionamentos.
- Diagramas de casos de uso: definem o comportamento dos atores para cada requisito.
- Diagramas de sequência: atribuem funções para cada classe.
- Diagramas de robustez: eliminam a ambiguidade dos requisitos de comportamento, e vincula aos objetos.

Na Figura 13, representa-se o processo por meio de suas etapas, junto aos artefatos gerados.

Figura 13 – Processo Iconix



Fonte: Rosemberg, Stephens & COLLINS-COPE (2005, pp. 45).

Os benefícios de se utilizar o processo Iconix, vistos em Rosemberg, Stephens & COLLINS-COPE (2005) podem ser apontados por:

- Metodologia prática e simples, com fases objetivas.
- Eliminação de ambiguidade de requisitos com design limpo.
- Rastreabilidade de requisitos sem excesso de documentação, visando apenas o necessário.

3 PROPOSTA DE SOLUÇÃO

Observando a estrutura de um prontuário clínico veterinário, construiu-se um sistema do tipo aplicativo para dispositivos móveis que visa auxiliar médicos veterinários a registrarem seus atendimentos aos pacientes de forma digital. De tal maneira, o aplicativo conta com um cadastro de uma instituição clínica veterinária, com seus respectivos profissionais, clientes, pacientes, e atendimentos.

A fim de executar a proposta principal do projeto, fez-se também um cadastro de usuários que utilizarão o sistema, sendo usualmente veterinários. Com isso, possibilitou-se de serem registrados atendimentos de prontuários odontológicos, e realizar ações específicas sobre os mesmos.

Ao categorizar o nível de acesso das pessoas no sistema, as mesmas têm permissões de uso restringidas ao seu cargo na clínica. Somente um administrador poderá definir para cada usuário, quais operações o mesmo poderá fazer sobre os dados, e somente as permissões fornecidas possibilitarão o acesso à determinadas funções.

O fluxo de uso do sistema pode ser exemplificado como: um administrador previamente definido em uma clínica veterinária irá cadastrar sua própria conta no aplicativo. Em seguida, no sistema, preencherá os dados da clínica em que trabalha. Após isso, o mesmo irá cadastrar os usuários e *login* dos veterinários e secretários que trabalham na clínica. Posteriormente, ele fornecerá esse *login* criado para seus proprietários acessarem o sistema, e alterar sua senha junto aos seus dados pessoais, se necessário. Com um usuário em mãos, os mesmos poderão efetuar suas tarefas no aplicativo.

A pessoa com cargo de veterinário, ao acessar o sistema, poderá iniciar um novo atendimento através da sua respectiva tela. Ao dar início no mesmo, ela irá inserir o animal a ser atendido, e selecionar o que será realizado, que poderá ser uma consulta ou procedimento. Ao preencher estes dados predefinidos, será trazida a ficha de acordo com a espécie do animal inserido. Nela estarão todos os seus dados e opções de representações de odontogramas que estão disponíveis para serem preenchidos por meio de desenhos sobre os mesmos. Existe também a possibilidade de inserção de imagens feitas durante um atendimento.

O veterinário, adicionalmente poderá cadastrar os clientes da clínica com os seus respectivos animais, sendo estes os pacientes. Os mesmos cadastros de clientes e pacientes também poderão ser feitos por um secretário, que na maioria dos casos, atua na recepção de uma clínica.

Os usuários precisam ter acesso à internet para usar o aplicativo e sincronizar as informações criadas até o banco de dados do sistema.

4 PROJETO DO SISTEMA

Como visto na Seção 2.2, o mercado atualmente conta com mais opções de sistemas de prontuário veterinário para a plataforma Web, sendo que estes não apresentam a melhor ergonomia possível para clínicas veterinárias, devido ao fato de nem sempre haver um computador próximo às salas de atendimento. Por isso, decidiu-se por construir uma aplicação para a plataforma Android, pois como evidenciado na Seção 2.4, existe um maior número de dispositivos ativos ao redor do mundo utilizando-a, além da maior parte das vendas de novos dispositivos serem de aparelhos que executam esta mesma plataforma.

Por ser feito exclusivamente de forma nativa para o Android, optou-se pela linguagem Java como linguagem de desenvolvimento. Esta decisão foi tomada pelo motivo do desenvolvedor possuir maior familiaridade com a mesma, e também por ela estar há mais tempo no mercado. O ambiente de produção foi o Android Studio, pela razão de ser a ferramenta oficial do sistema operacional.

Como a filosofia de desenvolvimento é voltada à dispositivos móveis, para fazer o gerenciamento e armazenamento de informações criadas pelo usuário, a plataforma em nuvem Firebase foi empregada para fornecer opções de armazenamento e recuperação de dados e arquivos que estarão vinculados a uma clínica.

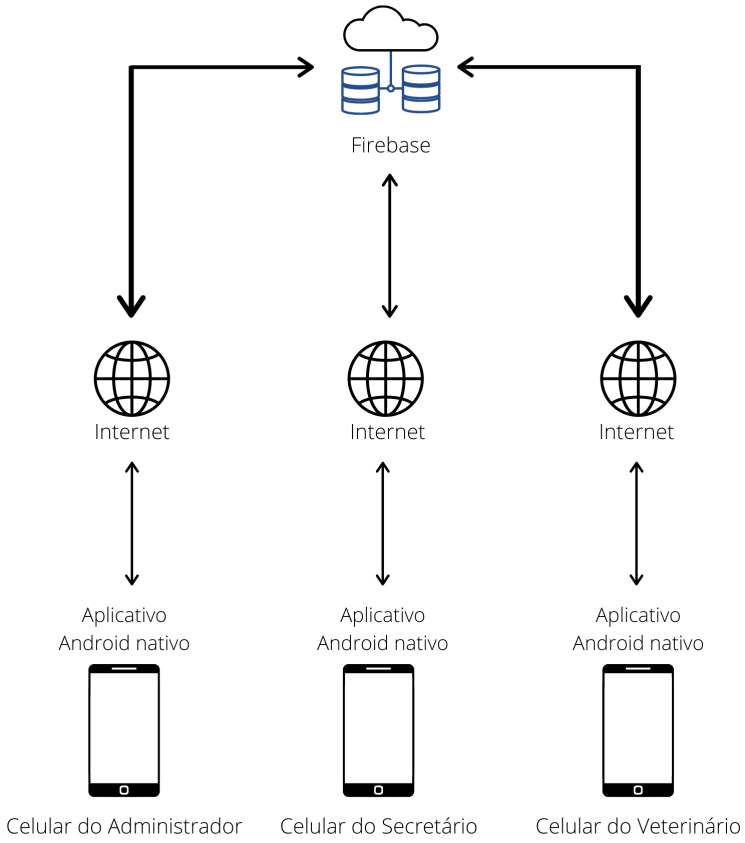
O maior diferencial que justifica a criação deste aplicativo em relação aos já existentes, como o BensVET, é criar algo que seja exclusivo para registros de prontuários odontológicos veterinários através de *smartphones*.

O aplicativo desenvolvido neste trabalho especialmente para o sistema operacional Android, usufrui de todas as características de uma aplicação nativa. Por isso, um usuário que já possui familiaridade com um estilo de interface padrão nas aplicações de sua plataforma, encontrará facilidade em sua utilização, além de contar com um maior desempenho em suas tarefas.

4.1 ARQUITETURA DO SISTEMA

Por meio dos requisitos coletados para o sistema, idealizou-se uma topologia para o aplicativo, mostrada na Figura 14.

Figura 14 – Topologia do aplicativo



Fonte: O Autor (2021).

Tanto para o administrador, quanto para o secretário e veterinário, o aplicativo será o mesmo, e o que restringe as funcionalidades que cada um poderá acessar, são as permissões de acesso. As mesmas estarão armazenadas no banco de dados em tempo real do Firebase, e são validadas no momento em que a sessão é iniciada no sistema.

Ao efetuar operações no aplicativo que geram criação ou modificação de dados, eles são enviados do aplicativo, para o Firebase, através da internet. Já para obter ou visualizá-los, o caminho é inverso, sendo trazidos do Firebase até o celular do usuário, tudo em tempo real.

4.2 REQUISITOS DO SISTEMA

Conforme Sommerville (2019), um sistema precisa oferecer solução e proporcionar benefícios diante das necessidades de um utilizador. Assim, por meio de uma visão de alto nível, listam-se os requisitos funcionais e não-funcionais que o sistema deve conter para garantir que funcionalidades estarão presentes na aplicação. As Seções 4.2.1 e 4.2.2 demonstram quais requisitos serão necessários na aplicação.

4.2.1 Requisitos Funcionais

De acordo com Sommerville (2019), requisitos funcionais são declarações dos serviços que o sistema deve fornecer, e como deve reagir à determinadas entradas. Também define seu comportamento em determinadas situações. No Quadro 2 são exibidos quais requisitos estão presentes no sistema.

Quadro 2 – Requisitos Funcionais do sistema

ID	Requisito
RF-01	Criar usuários
RF-02	Gerenciar perfis de usuário
RF-03	Editar conta de usuário
RF-04	Cadastrar clínica
RF-05	Cadastrar clientes
RF-06	Cadastrar animais
RF-07	Realizar atendimento
RF-08	Preencher ficha de exame clínico e de procedimento
RF-09	Demonstrar a ilustração do odontograma do animal
RF-10	Demonstrar a ilustração de cavidade oral aberta
RF-11	Demonstrar a ilustração de cavidade oral esquerda, direita e crânios
RF-12	Desenhar sobre a ilustração do odontograma, cavidades orais e crânios
RF-13	Inserir observações de atendimento
RF-14	Inserir imagens no atendimento
RF-15	Exportar atendimento

Fonte: O Autor (2021).

4.2.2 Requisitos Não-Funcionais

Assim como em um sistema existem requisitos funcionais que definem o que o mesmo deve fornecer, também existem os requisitos não-funcionais, onde Sommerville (2019) explica que os mesmos são restrições sobre os serviços oferecidos pelo sistema, aplicados nele como um todo. Através do Quadro 3, listam-se os requisitos não-funcionais do sistema.

Quadro 3 – Requisitos Não-Funcionais do sistema

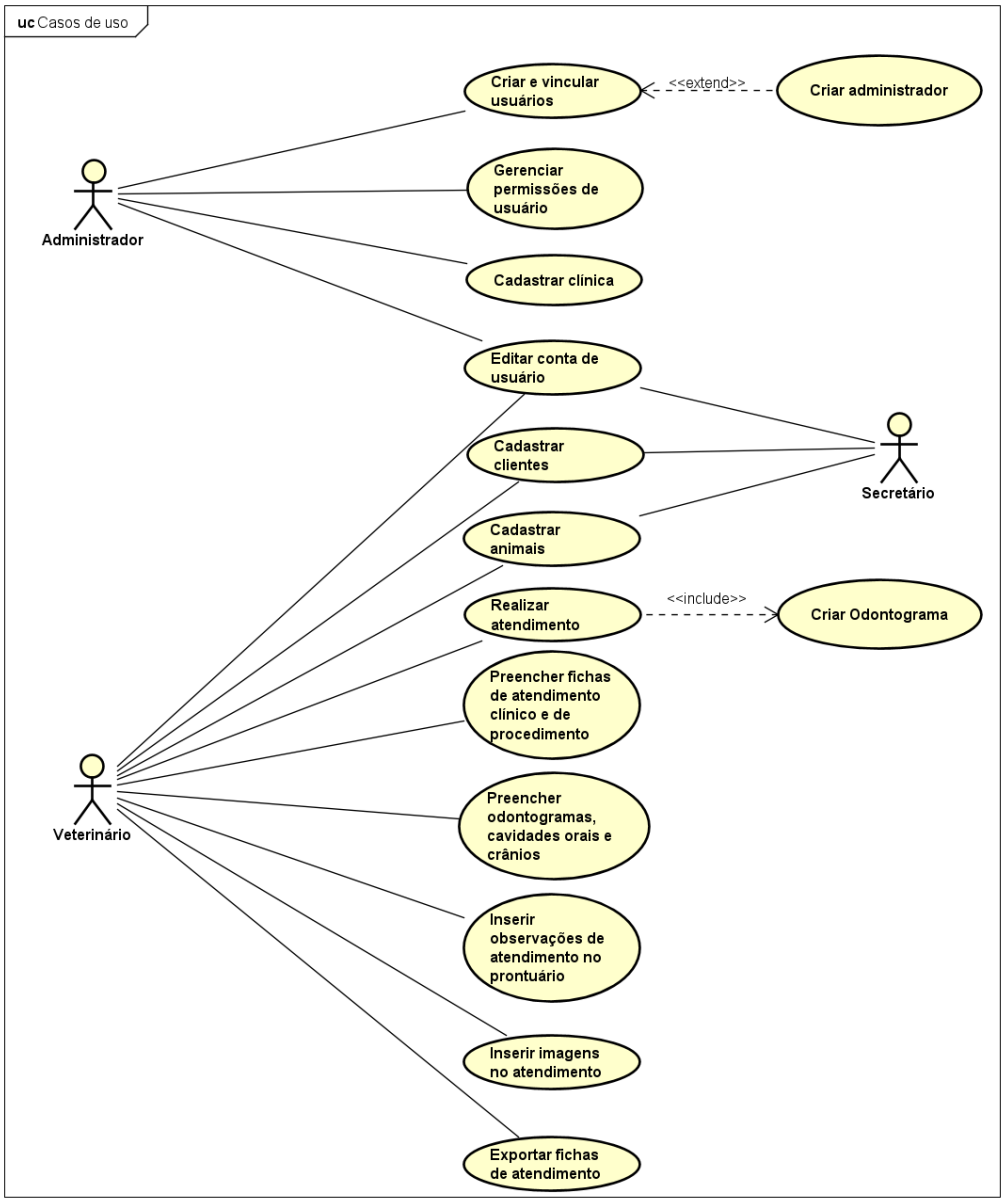
ID	Requisito
RNF-01	Ser simples de utilizar
RNF-02	O sistema deve estar disponível 24 horas por dia, todos os dias da semana
RNF-03	Ser uma aplicação nativa

Fonte: O Autor (2021).

4.3 CASOS DE USO

Por meio dos requisitos funcionais coletados, são formalizados os casos de uso que representam como os atores utilizarão o sistema, junto ao que cada um deles pode fazer.

Figura 15 – Diagrama de Casos de Uso



Fonte: O Autor (2021).

Mediante o diagrama apresentado, detalha-se cada um dos casos de uso para indicar o que os mesmos irão desempenhar no sistema.

Quadro 4 – Caso de Uso 01 - Criar e vincular usuários

Caso de Uso 01 - Criar e vincular usuários	
Descrição	O administrador após fazer o seu próprio cadastro de usuário, poderá cadastrar e vincular novos usuários no sistema.
Requisitos atendidos	RF-01
Ator	Administrador
Pré-condição	1. Possuir o aplicativo instalado no dispositivo, com acesso à internet. 2. Possuir um usuário administrador criado. 3. Acessar o sistema.
Fluxo principal	1. Administrador definido cria o seu próprio usuário, se não criado, e acessa o sistema. 2. Administrador acessa a tela de cadastro de usuários. 3. Administrador preenche e confirma os dados obrigatórios do novo usuário: a) Dados cadastrais: nome, *e-mail*, *login* e senha. b) Cargo: Secretário, Veterinário ou outro. c) Permissões de acesso: cadastrar clientes, cadastrar animais, realizar atendimento, administrador.
Fluxo alternativo e exceções	Item 3: 1. Administrador não preenche os campos obrigatórios, e tenta confirmar. a) Sistema exibe mensagem de erro solicitando preenchimento dos dados.
Pós-condição	Usuário cadastrado e vinculado.

Fonte: O Autor (2021).

Quadro 5 – Caso de Uso 02 - Gerenciar permissões de usuário

Caso de Uso 02 - Gerenciar permissões de usuário	
Descrição	O administrador poderá conceder ou restringir permissões de acesso de funções à determinados usuários no sistema, com grupo de permissões.
Requisitos atendidos	RF-02
Ator	Administrador
Pré-condição	Possuir usuários vinculados ao sistema.
Fluxo principal	1. Administrador acessa a tela de usuários. 2. Seleciona um usuário. 3. Atribui ou remove permissão específica ao usuário selecionado.
Fluxo alternativo e exceções	Item 3: 1. Administrador remove todas as permissões de um usuário, e tenta confirmar. a) Sistema exibe alerta para marcar ao menos uma permissão.
Pós-condição	Permissões de usuário atualizadas.

Fonte: O Autor (2021).

Quadro 6 – Caso de Uso 03 - Editar conta de usuário

Caso de Uso 03 - Editar conta de usuário	
Descrição	O usuário poderá alterar os dados da sua conta.
Requisitos atendidos	RF-03
Ator	Administrador, Secretário, Veterinário.
Pré-condição	1. Possuir uma conta de usuário. 2. Estar logado no aplicativo.
Fluxo principal	1. Acessar o menu de opções do aplicativo. 2. Abrir a tela de Conta de usuário. 3. Preencher ou alterar os dados contidos no formulário: nome, cpf, rg, telefone 1, telefone 2, crmv, cidade, cep, rua, número, bairro, uf, *e-mail*, senha, nova senha.
Fluxo alternativo e exceções	Item 3: 1. Ao salvar os dados, o usuário não preenche os dados obrigatórios: nome, cpf, *e-mail*. a) Sistema emite aviso para preencher os campos. 2. Veterinário não informa o seu número crmv. a) Sistema emite aviso para preencher o campo.
Pós-condição	Dados de usuário atualizados.

Fonte: O Autor (2021).

Quadro 7 – Caso de Uso 04 - Cadastrar clínica

Caso de Uso 04 - Cadastrar clínica	
Descrição	O administrador poderá preencher ou editar no sistema os dados da clínica onde trabalha.
Requisitos atendidos	RF-04
Ator	Administrador
Pré-condição	1. Ser um usuário administrador. 2. Ter logado no sistema. 3. Acessar a tela de dados da clínica.
Fluxo principal	1. Administrador acessa a tela de dados da clínica. 2. Administrador preenche os dados obrigatórios da clínica: razão social, cnpj, *e-mail*, telefone 1, cidade, cep, rua, número, bairro e uf.
Fluxo alternativo e exceções	Item 2: 1. Administrador não preenche os campos obrigatórios, e tenta confirmar. a) Sistema exibe mensagem de erro solicitando preenchimento dos dados.
Pós-condição	Dados da clínica atualizados.

Fonte: O Autor (2021).

Quadro 8 – Caso de Uso 05 - Cadastrar clientes

Caso de Uso 05 - Cadastrar clientes	
Descrição	O secretário, ou veterinário poderão cadastrar e editar clientes no sistema.
Requisitos atendidos	RF-05
Ator	Secretário, veterinário.
Pré-condição	1. Estar logado no sistema. 2. Possuir permissão de cadastro de clientes.
Fluxo principal	1. Usuário acessa o sistema 2. Usuário acessa a tela de cadastro de clientes. 3. Usuário acessa o formulário e preenche os dados obrigatórios: nome, cpf, rg, telefone 1, cidade, cep, rua, número, bairro, uf e indicação. 4. Sistema exibe confirmação.
Fluxo alternativo e exceções	Item 3: 1. Usuário não preenche os campos obrigatórios. a) Sistema exibe mensagem de erro para preencher os campos. 2. Usuário preenche os dados de cpf e rg com formato inválido. a) Sistema através da validação do formato dos campos, emite mensagem de erro para revisar os dados e inserir novamente.
Pós-condição	Cliente cadastrado.

Fonte: O Autor (2021).

Quadro 9 – Caso de Uso 06 - Cadastrar animais

Caso de Uso 06 - Cadastrar animais	
Descrição	O secretário ou veterinário poderão cadastrar e editar animais no sistema.
Requisitos atendidos	RF-06
Ator	Secretário, veterinário.
Pré-condição	1. Possuir um cliente cadastrado. 2. Possuir permissão de cadastro de animais
Fluxo principal	1. Usuário acessa o sistema. 2. Usuário acessa a tela de cadastro de animais. 3. Usuário preenche o formulário, junto aos dados obrigatórios do animal: proprietário, nome, idade, peso, sexo, espécie, raça, cor da pelagem, e temperamento. 4. Sistema exibe confirmação.
Fluxo alternativo e exceções	Item 3: 1. Usuário não preenche os campos obrigatórios. a) Sistema exibe mensagem de erro para preencher os campos obrigatórios.
Pós-condição	Animal cadastrado.

Fonte: O Autor (2021).

Quadro 10 – Caso de Uso 07 - Realizar atendimento

Caso de Uso 07 - Realizar atendimento	
Descrição	O veterinário poderá cadastrar e iniciar um novo atendimento no sistema.
Requisitos atendidos	RF-07
Ator	Veterinário.
Pré-condição	1. Estar logado no sistema. 2. Possuir um cliente cadastrado. 3. Possuir um animal cadastrado. 4. Possuir permissão com cargo de veterinário.
Fluxo principal	1. Usuário acessa o sistema. 2. Usuário acessa a tela de novo atendimento. 3. Usuário preenche os campos obrigatórios: animal, e tipo de atendimento. 4. Sistema prossegue e exibe tela com a ficha específica do animal a ser atendido.
Fluxo alternativo e exceções	Item 3: 1. Usuário não preenche os campos obrigatórios e tenta prosseguir para as fichas. a) Sistema exibe mensagem de erro para preencher os campos obrigatórios.
Pós-condição	Atendimento prossegue para etapa de preenchimento de ficha.

Fonte: O Autor (2021).

Quadro 11 – Caso de Uso 08 - Preencher fichas de atendimento clínico e de procedimento

Caso de Uso 08 - Preencher fichas de atendimento clínico e de procedimento	
Descrição	O veterinário, no decorrer de um atendimento no sistema, deverá preencher as fichas de atendimento de exames clínicos ou de procedimento.
Requisitos atendidos	RF-08
Ator	Veterinário.
Pré-condição	Ter cadastrado um atendimento.
Fluxo principal	1. Usuário inicializa um novo atendimento. 2. Sistema exibe a tela de atendimento com a respectiva ficha. 3. Usuário preenche os dados pertencentes à ficha de exame clínico, exibidos no Quadro 17, e da ficha de procedimento, representados no Quadro 18.
Fluxo alternativo e exceções	Não possui.
Pós-condição	Ficha de atendimento preenchida.

Fonte: O Autor (2021).

Quadro 12 – Caso de Uso 09 - Preencher odontogramas, cavidades orais e crânios

Caso de Uso 09 - Preencher odontogramas, cavidades orais e crânios	
Descrição	O veterinário poderá desenhar à mão livre sobre as representações gráficas do odontograma, cavidade oral e crânio do animal para sinalizar achados e procedimentos.
Requisitos atendidos	RF-09, RF-10, RF-11, RF-12
Ator	Veterinário.
Pré-condição	1. Ter cadastrado um atendimento. 2. Estar na tela de preenchimento de ficha.
Fluxo principal	1. Usuário realiza um novo atendimento. 2. Sistema exibe as opções para exibir o odontograma, cavidades orais e crânios do animal. 3. Usuário desenha na representação selecionada.
Fluxo alternativo e exceções	Item 2: 1. Sistema perde a conexão e não consegue carregar o odontograma. a) Sistema aguarda conexão à rede para obter o odontograma
Pós-condição	Odontograma preenchido.

Fonte: O Autor (2021).

Quadro 13 – Caso de Uso 10 - Inserir observações de atendimento no prontuário

Caso de Uso 10 - Inserir observações de atendimento no prontuário	
Descrição	O veterinário, ao realizar um atendimento no sistema, poderá inserir observações ao prontuário do animal.
Requisitos atendidos	RF-13
Ator	Veterinário.
Pré-condição	1. Ter cadastrado um atendimento. 2. Estar na tela de preenchimento de ficha.
Fluxo principal	1. Usuário realiza um novo atendimento. 2. Sistema exibe a tela com a ficha do animal. 3. Ao final da ficha, o usuário insere dados referentes ao atendimento, no campo de observações.
Fluxo alternativo e exceções	Item 3: 1. Usuário excede o limite de caracteres do campo de observações. a) Campo interrompe a inserção de caracteres.
Pós-condição	Observações inseridas.

Fonte: O Autor (2021).

Quadro 14 – Caso de Uso 11 - Inserir imagens no atendimento

Caso de Uso 11 - Inserir imagens no atendimento	
Descrição	O veterinário ao realizar um atendimento no sistema, poderá vincular e visualizar imagens no prontuário do animal.
Requisitos atendidos	RF-14
Ator	Veterinário.
Pré-condição	1. Ter cadastrado um atendimento. 2. Estar na tela de preenchimento de ficha.
Fluxo principal	1. Usuário realiza um novo atendimento. 2. Sistema exibe a tela do atendimento. 3. Usuário acessa a tela de imagens, e insere uma foto no prontuário, buscando do dispositivo, ou registrando com a câmera.
Fluxo alternativo e exceções	Item 3: 1. Sistema perde a conexão e não consegue enviar a imagem para o servidor. a) Sistema aguarda conexão à rede para tentar o reenvio.
Pós-condição	Imagens inseridas.

Fonte: O Autor (2021).

Quadro 15 – Caso de Uso 12 - Exportar fichas de atendimento

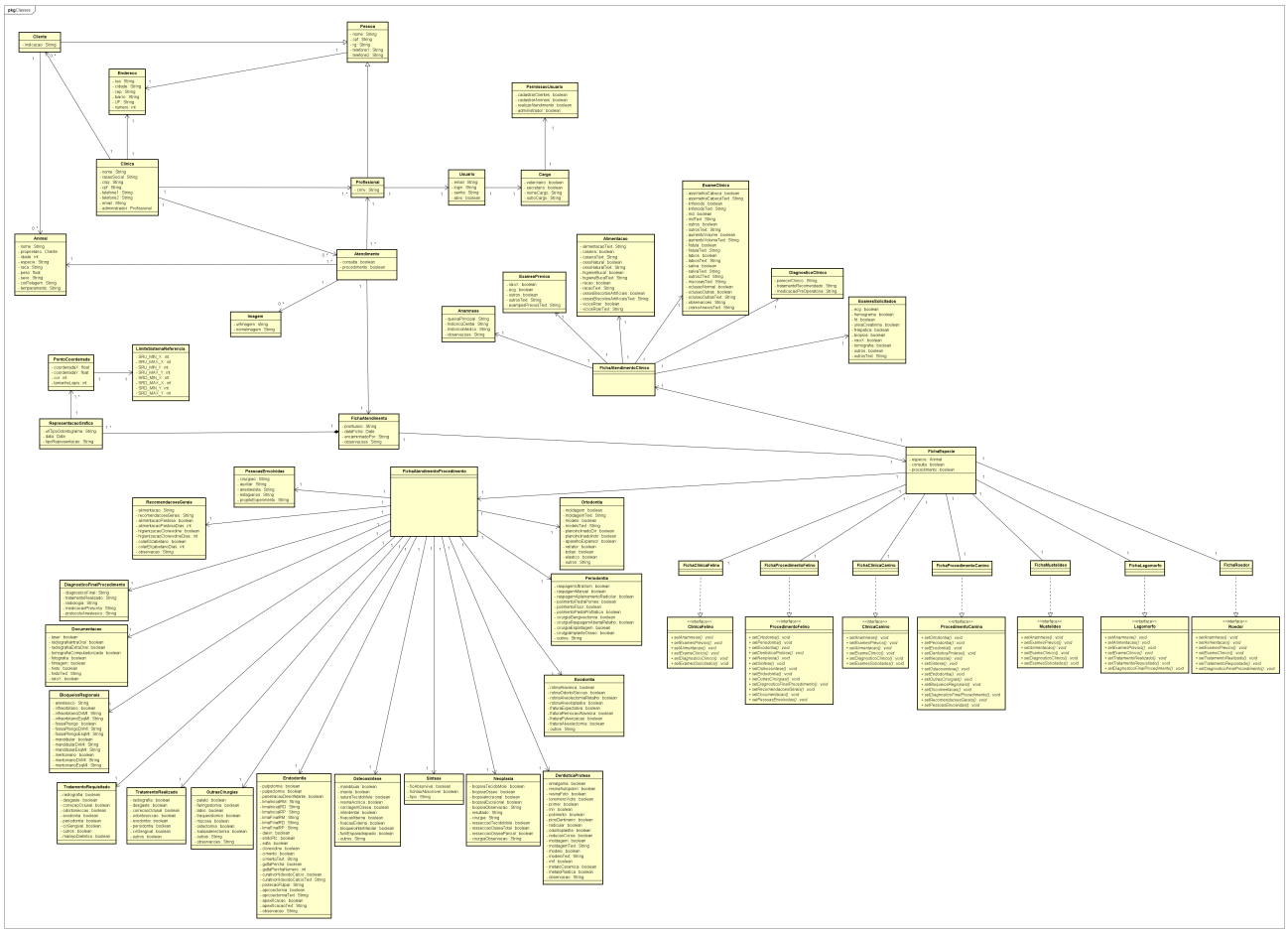
Caso de Uso 12 - Exportar fichas de atendimento	
Descrição	O veterinário quando necessitar disponibilizar uma ficha de atendimento, precisará exportar a mesma para um formato de arquivo de leitura.
Requisitos atendidos	RF-15
Ator	Veterinário.
Pré-condição	1. Ter realizado um atendimento
Fluxo principal	1. Usuário acessa a aba de atendimentos. 2. Seleciona o atendimento pretendido. 3. Seleciona opção de exportar ficha. 4. Seleciona para onde deseja exportar.
Fluxo alternativo e exceções	Item 3: 1. Sistema perde a conexão e não consegue buscar os dados para montar a ficha. a) Sistema aguarda conexão à rede para tentar a exportação novamente.
Pós-condição	Ficha de atendimento exportada.

Fonte: O Autor (2021).

4.4 MODELO DE DOMÍNIO

Com o objetivo de mapear a estrutura de organização do sistema, construiu-se o modelo de domínio com emprego das classes, atributos e relacionamentos, que estão representados na Figura 16.

Figura 16 – Diagrama de Domínio do sistema

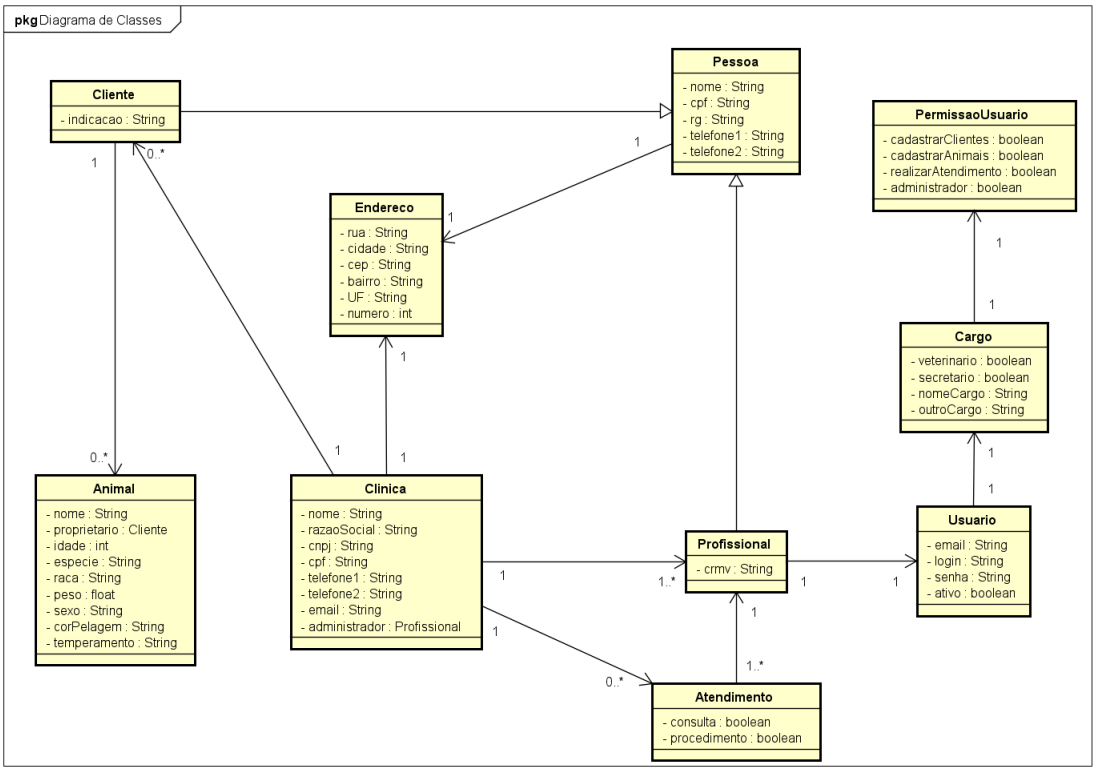


Fonte: O Autor (2021).

A partir da Figura 16, reduziu-se a mesma em cinco fragmentos de imagens, para melhor visualizar as classes que compõem o modelo.

Na Figura 17, mostra-se o recorte onde compreendem-se as classes associadas ao gerenciamento organizacional da clínica.

Figura 17 – Diagrama de Domínio do sistema - Parte 1

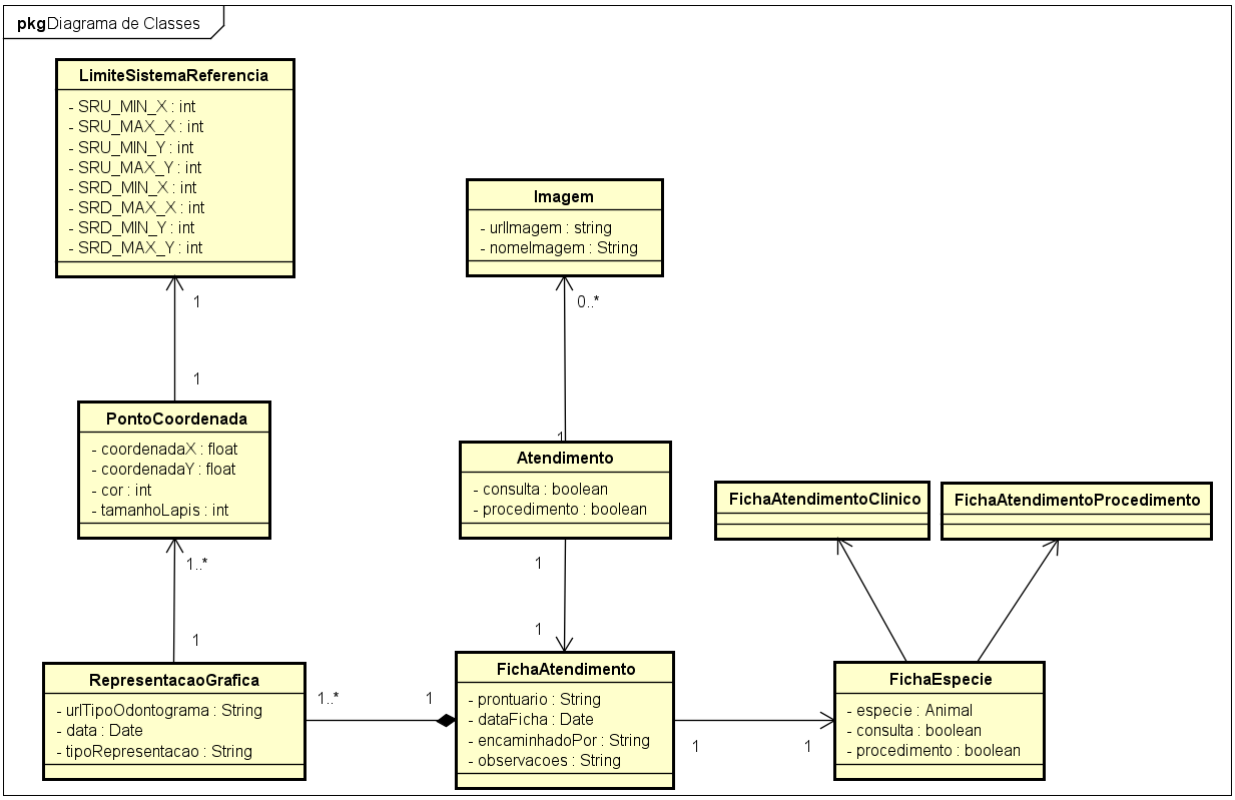


Fonte: O Autor (2021).

Junto, é referenciada a classe de atendimento, pertencente à estrutura das fichas, continuada pela Figura 18.

Na Figura 18 prossegue-se mostrando as classes que formam a estrutura de um atendimento. Nesta, especificam-se os tipos de fichas de atendimento clínico e de procedimento, que são continuadas pelas Figuras 19, 20 e 21.

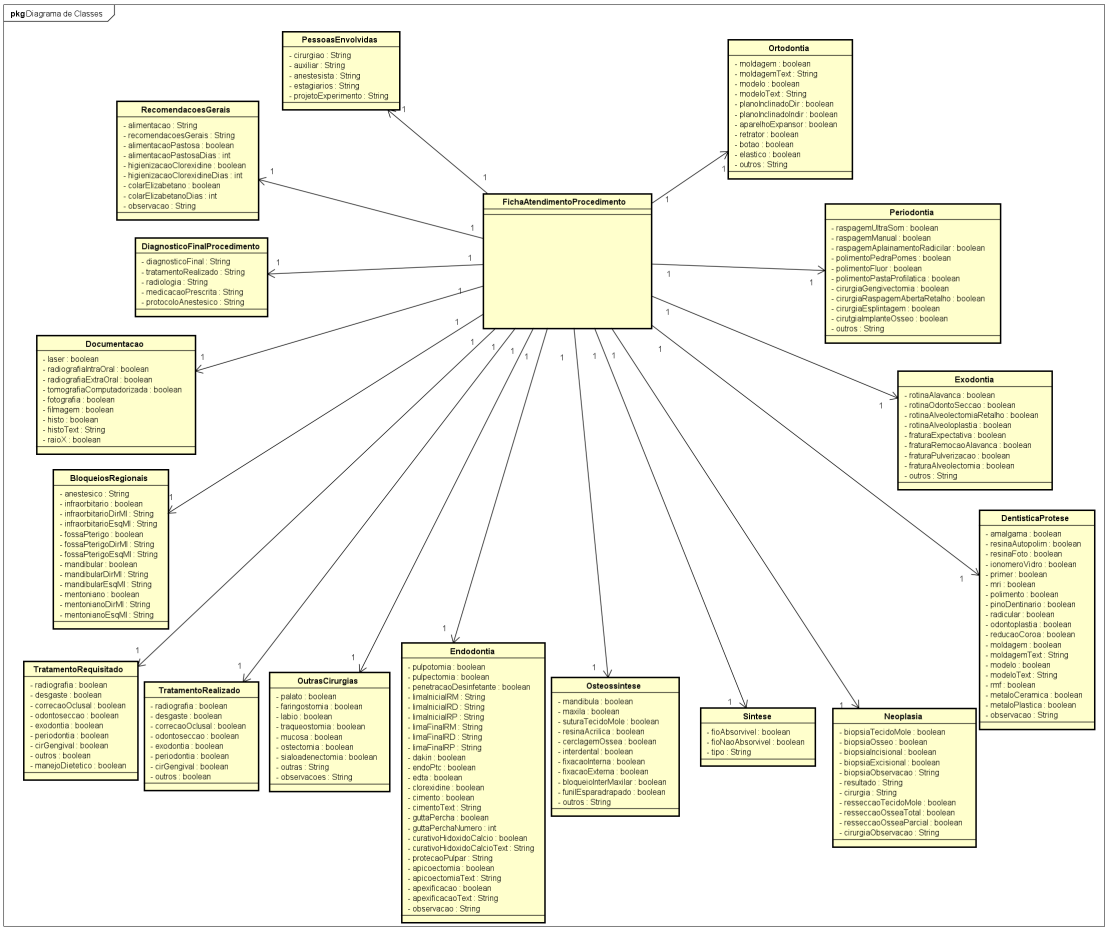
Figura 18 – Diagrama de Domínio do sistema - Parte 2



Fonte: O Autor (2021).

Na Figura 19, mostra-se a classe da ficha de procedimentos, que comporta os atributos das classes que equivalem aos grupos de dados de procedimentos cirúrgicos da mesma.

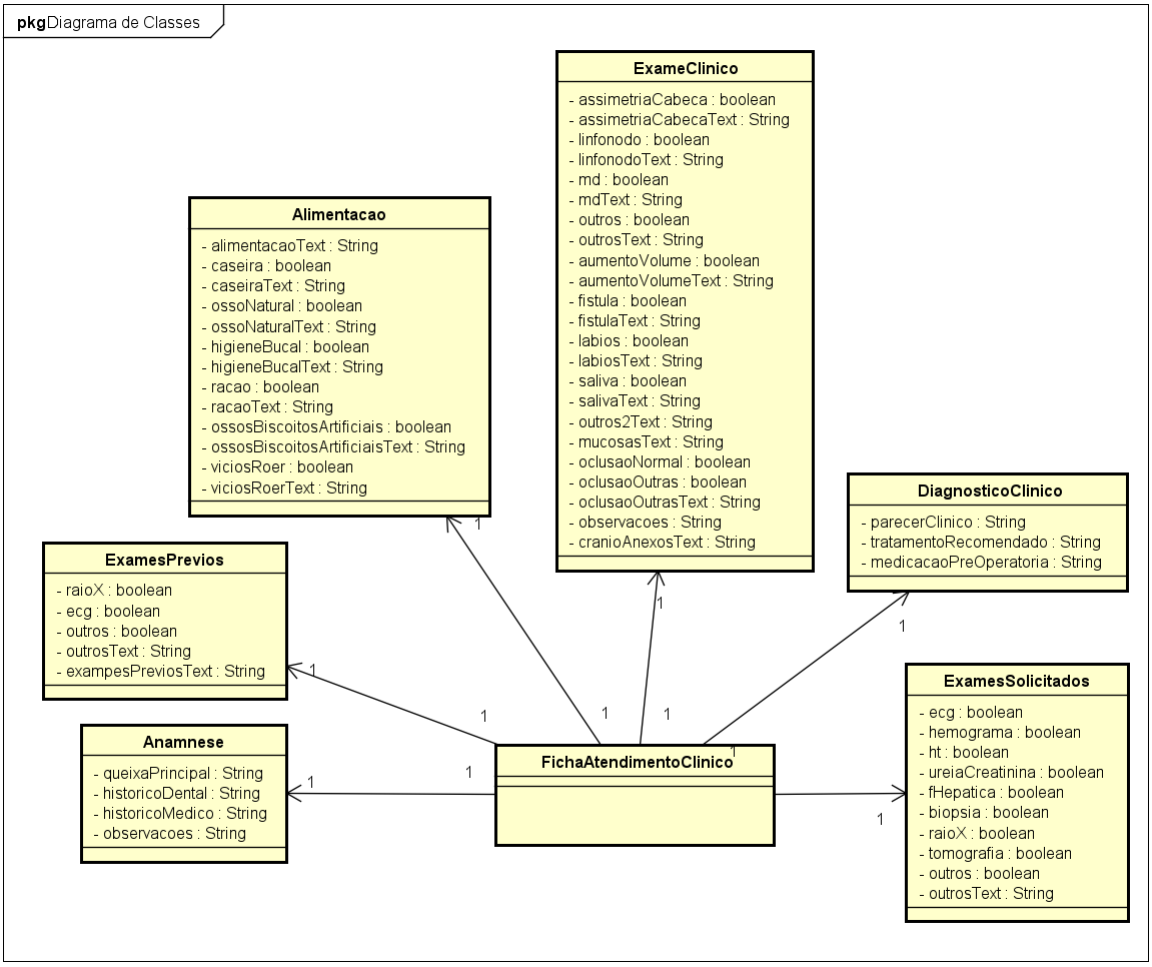
Figura 19 – Diagrama de Domínio do sistema - Parte 3



Fonte: O Autor (2021).

Na Figura 20, mostra-se a classe da ficha de atendimento clínico, que comporta os atributos das classes que equivalem aos grupos de dados de condutas da mesma.

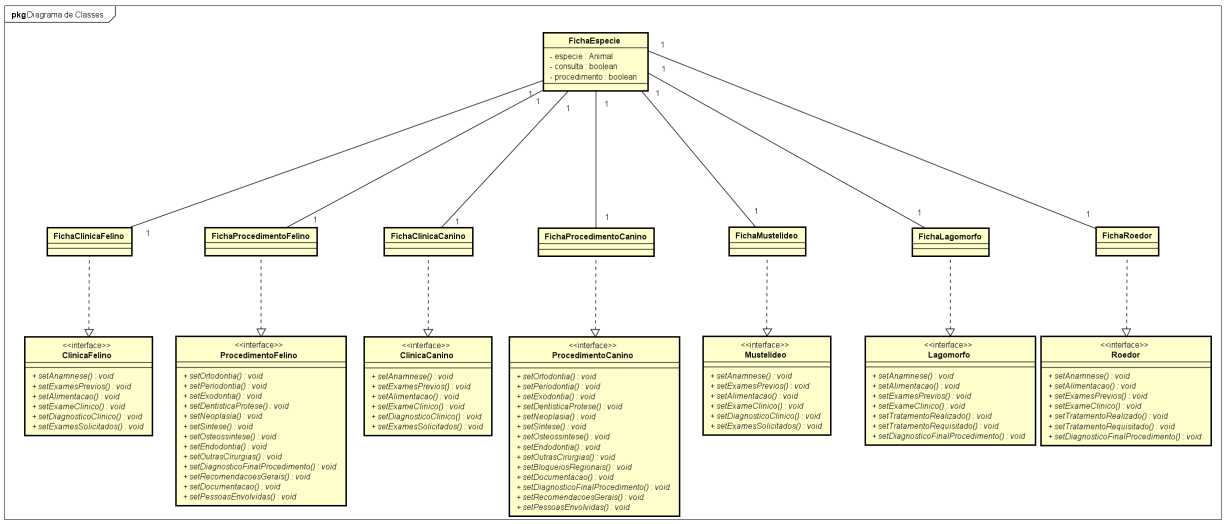
Figura 20 – Diagrama de Domínio do sistema - Parte 4



Fonte: O Autor (2021).

Na Figura 21, mostra-se a classe “FichaEspecie”, que comporta os atributos das classes que compreendem os tipos de fichas específicas de cada espécie de animal. Junto às mesmas, situam-se as interfaces atribuídas a cada uma, que indicam os métodos que devem estar presentes.

Figura 21 – Diagrama de Domínio do sistema - Parte 5



Fonte: O Autor (2021).

Para entender o significado das classes presentes no diagrama, o Quadro 16 descreve a função de cada uma delas.

Quadro 16 – Descrição das classes do modelo de domínio

(continua)

Nome da classe	**Descrição**
Alimentacao	Dados de preenchimento da alimentação do paciente na ficha de exame clínico.
Anamnese	Dados de preenchimento da anamnese do paciente na ficha de exame clínico.
Animal	Dados de cada animal que é paciente da clínica. Um animal pertence à um cliente.
Atendimento	Dados de atendimento de um paciente, onde enquadra-se o profissional, e o tipo de atendimento que contém uma ficha específica para o mesmo.
BloqueiosRegionais	Preenchimento de dados de procedimentos referentes aos bloqueios regionais da ficha de procedimento.
Cargo	Cargo que um profissional possui na clínica.
Cliente	Classe secundária que armazena os dados de cada cliente da clínica.
Clinica	Informações referentes à clínica, como seus próprios dados, profissionais, clientes, pacientes e atendimentos.
DentisticaProtese	Preenchimento de dados de procedimentos referentes à Dentística/Prótese da ficha de procedimento.
DiagnosticoClinico	Dados de preenchimento do diagnóstico feito na ficha de exame clínico do paciente.
DiagnosticoFinalProcedimento	Dados para preenchimento referentes ao diagnóstico final da ficha de procedimento.
Documentacao	Documentos utilizados no procedimento, listados na ficha de procedimento.
Endereco	Atributos de endereço da clínica e das pessoas envolvidas.
Endodontia	Preenchimento de dados na ficha de procedimento, referentes à Endodontia.
ExameClinico	Dados de preenchimento na ficha de exame clínico, referentes aos exames clínicos realizados no paciente.
ExamesPrevios	Dados de preenchimento dos exames previos do paciente na ficha de exame clínico.

Quadro 16 – Descrição das classes do modelo de domínio

(continuação)

Nome da classe	**Descrição**
ExamesSolicitados	Dados de preenchimento dos exames solicitados na ficha de exame clínico do paciente.
Exodontia	Preenchimento de dados de procedimentos referentes à Exodontia na ficha de procedimento.
FichaAtendimento	Cabeçalho de informações de um atendimento, como data, número de prontuário e observações.
FichaAtendimentoClinico	Classe que contém os atributos das classes que formam os grupos de dados das fichas de exame clínico.
FichaClinicaCanina	Monta uma ficha com os atributos pertencentes à ficha de consulta canina.
FichaClinicaFelino	Monta uma ficha com os atributos pertencentes à ficha de exame clínico felina.
FichaEspecie	Traz uma ficha com os grupos e campos de dados próprios de um paciente de determinada espécie.
FichaLagomorfo	Monta uma ficha com os atributos pertencentes à ficha da espécie dos lagomorfos.
FichaMustelideo	Monta uma ficha com os atributos pertencentes à ficha da espécie dos mustelídeos.
FichaRoedor	Monta uma ficha com os atributos pertencentes à ficha da espécie dos roedores.
FichaAtendimentoProcedimento	Classe que contém os atributos das classes que formam cada grupo de dados pertencentes à uma ficha de procedimento.
FichaProcedimentoCanino	Monta uma ficha com os atributos pertencentes à ficha de procedimento canina.
FichaProcedimentoFelino	Monta uma ficha com os atributos pertencentes à ficha de procedimento felina.
Imagem	Permite o veterinário vincular imagens a um atendimento.
LimiteSistemaReferencia	Atribui os valores de limites mínimos e máximos dos sistemas de coordenadas do universo e do dispositivo.
Neoplasia	Preenchimento de dados de procedimentos referentes à Neoplasia na ficha de procedimento.
Ortodontia	Preenchimento de dados de procedimentos referentes à Ortodontia na ficha de procedimento.

Quadro 16 – Descrição das classes do modelo de domínio

(conclusão)

Nome da classe	**Descrição**
Osteossintese	Preenchimento de dados de procedimentos referentes à Osteossíntese na ficha de procedimento.
OutrasCirurgias	Preenchimento de dados de procedimentos referentes à outros tipos de cirurgias na ficha de procedimento.
Periodontia	Preenchimento de dados de procedimentos referentes à Periodontia na ficha de procedimento.
PermissaoUsuario	Grupo de permissões de acesso para um usuário no sistema.
Pessoa	Classe primária com os atributos em comum dos clientes e profissionais.
PessoasEnvolvidas	Profissionais envolvidos na cirurgia, listados na ficha de procedimento.
PontoCoordenada	Armazena os atributos dos pontos de coordenadas do desenho sobre um odontograma, junto às cores e tamanhos destes.
Profissional	Classe secundária onde estão os atributos dos profissionais da clínica.
RecomendacoesGerais	Dados para preenchimento referentes às recomendações gerais prescritas na ficha de procedimento.
RepresentacaoGrafica	Representação gráfica de um tipo, podendo ser odontograma, cavidade oral, e crânio de um animal.
Sintese	Preenchimento de dados de procedimentos referentes à Síntese na ficha de procedimento.
TratamentoRealizado	Preenchimento de dados referentes aos tratamentos realizados, listados na ficha de procedimento.
TratamentoRequisitado	Preenchimento de dados referentes aos tratamentos requisitados.
Usuario	Dados de um usuário para acessar o sistema.

Fonte: O Autor (2021).

4.5 INTERFACES DE TELA DO APLICATIVO

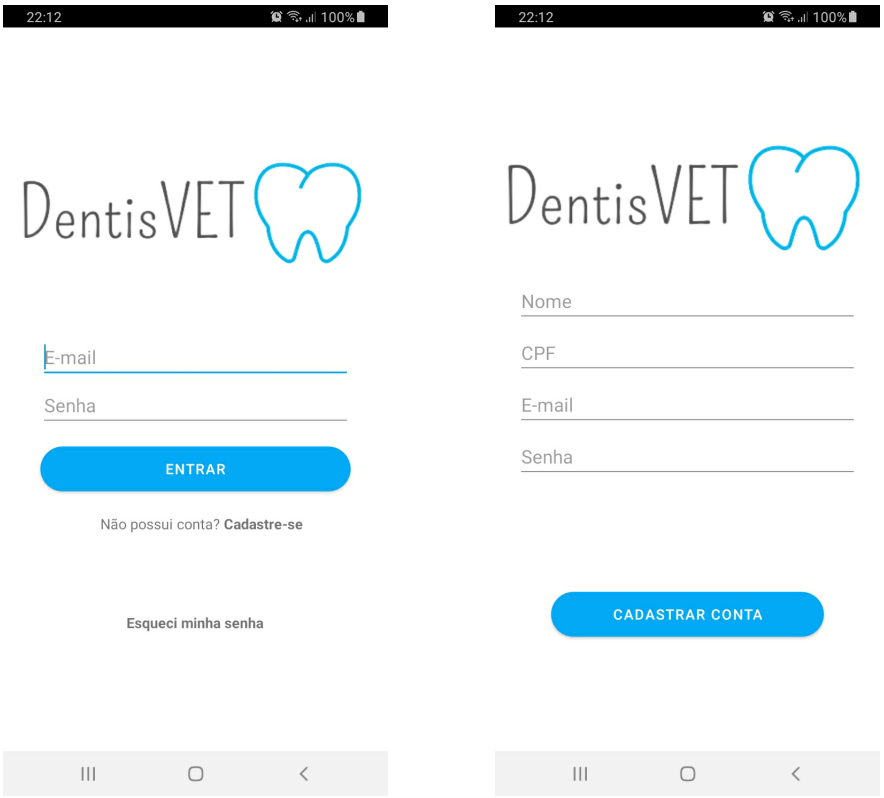
Nas seguintes subseções, serão demonstradas as capturas de tela das interfaces de usuário no aplicativo. Elas estabelecem como são exibidas cada funcionalidade dos itens especificados nos requisitos do projeto.

4.5.1 Criação de usuário administrador

O administrador previamente definido na clínica em que trabalha, para poder gerenciá-la no sistema junto aos usuários e respectivas permissões, criará o seu próprio usuário através da opção “Cadastre-se”, mostrado na Figura 22a.

Ao ser levado para a tela de cadastro, ele irá preencher os seus dados cadastrais, solicitados no formulário representado na Figura 22b.

Figura 22 – Interface gráfica - Criação de usuário administrador



(a) Tela de acesso ao sistema (b) Dados cadastrais do usuário

Fonte: O Autor (2021).

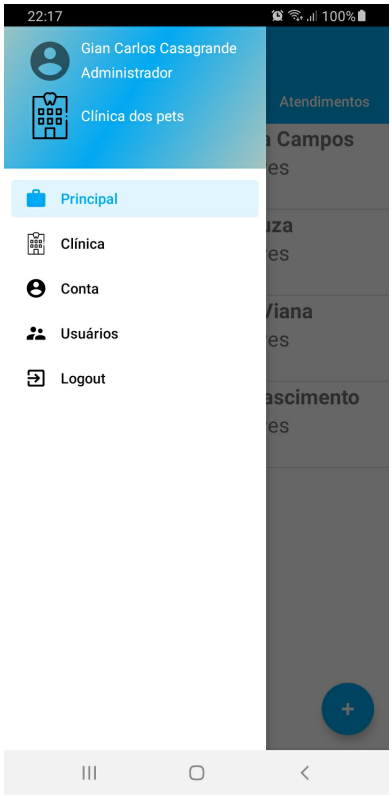
4.5.2 Menu de opções do administrador

O administrador tem no menu do sistema, as opções de gerenciar os dados de usuários, conta, e clínica. Estas opções estão representadas pela Figura 23.

Desta forma, o administrador irá acessar o gerenciamento dos usuários da clínica, utilizando o botão “Usuários”. Para cadastrar e editar os dados da clínica, usará o botão “Clínica”. Para editar os dados da sua própria conta, utilizará o botão “Conta”.

As opções de gerenciamento de usuários, e dados da clínica, são exclusivas do administrador, e somente ele poderá efetuar estas ações para garantir um melhor controle e consistência dos dados.

Figura 23 – Interface gráfica - Menu de opções do administrador



Fonte: O Autor (2021).

4.5.3 Dados da clínica

O administrador tem controle sobre os dados da clínica onde trabalha, podendo preenchê-los e editá-los, como demonstra-se no formulário exibido pela Figura 24.

Figura 24 – Interface gráfica - Dados da clínica



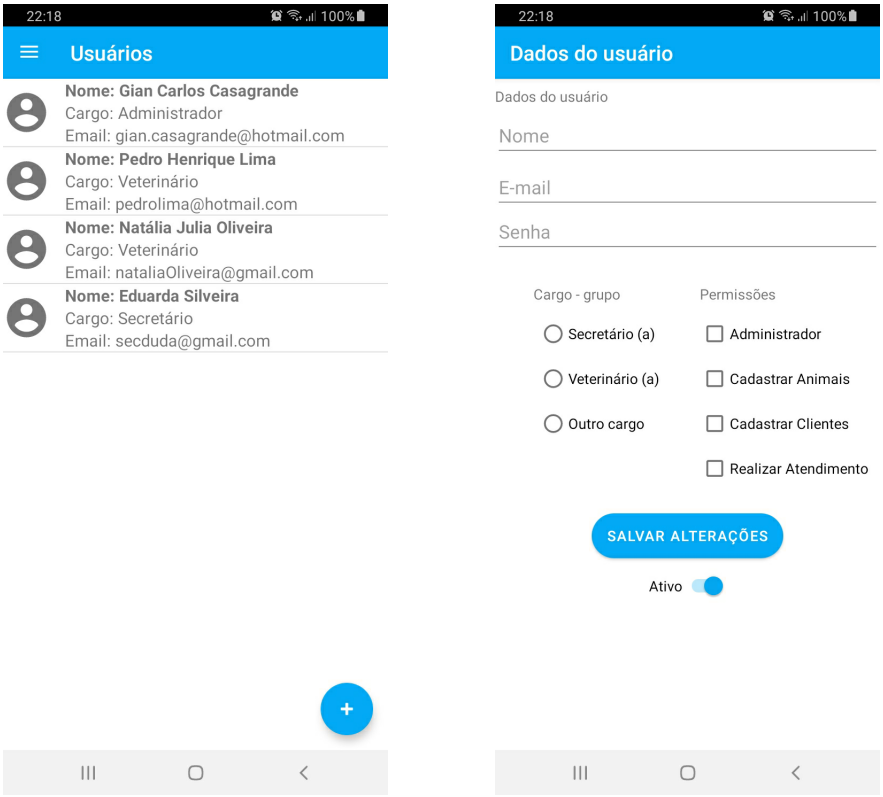
Fonte: O Autor (2021).

4.5.4 Gerenciamento de usuários

Para utilizar o aplicativo e suas funcionalidades, as pessoas precisam ter um usuário em mãos. Este é criado e fornecido pelo administrador.

O administrador acessa o sistema, e por meio do menu deslizante do aplicativo, encontra a opção de usuários. Ele irá selecionar a mesma, e visualizará os já existentes, como mostrado na Figura 25a.

Figura 25 – Interface gráfica - Gerenciamento de usuários no sistema



(a) Usuários do sistema (b) Criação/edição de usuário

Fonte: O Autor (2021).

Caso necessário, ele irá criar, editar, ou desativar os usuários das pessoas que trabalham na clínica.

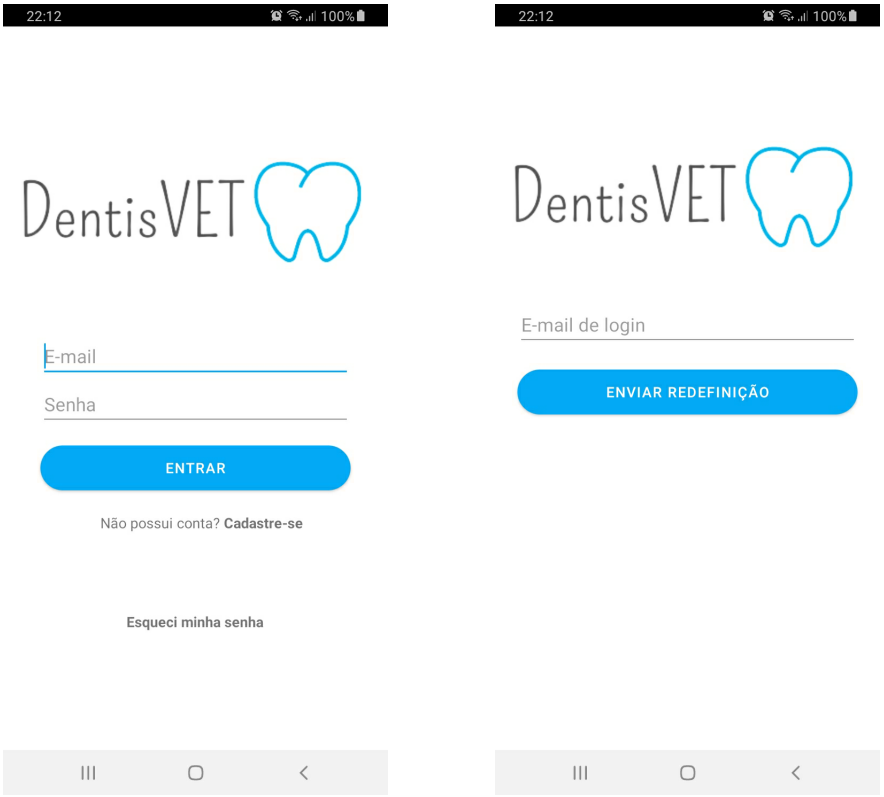
Na edição ou criação, é necessário atribuir um cargo ao usuário. Este cargo, por sua vez, possui um grupo de permissões predefinidas, que não podem ser desmarcadas. Porém, as demais permissões que apresentam-se disponíveis, podem ser atribuídas ao mesmo. Caso necessário, cria-se um novo cargo personalizado através da opção “Outro cargo”, e preenche-se o nome do mesmo na caixa de inserção, marcando também suas respectivas permissões, que são mostradas na Figura 25b.

4.5.5 Autenticação de usuário no sistema

Com um usuário criado e em mãos, o mesmo pode ser autenticado pela tela de *login* mostrada na Figura 22a.

Caso a pessoa tenha problemas com a senha, a mesma pode ser alterada através da opção “Esqueci minha senha”, ainda da tela de *login*. Para isso, ela irá informar seu endereço de *e-mail*, indicado no campo mostrado pela Figura 26b, para receber um *e-mail* de redefinição.

Figura 26 – Interface gráfica - Autenticação do usuário



(a) Tela de *login* (b) Redefinição de senha

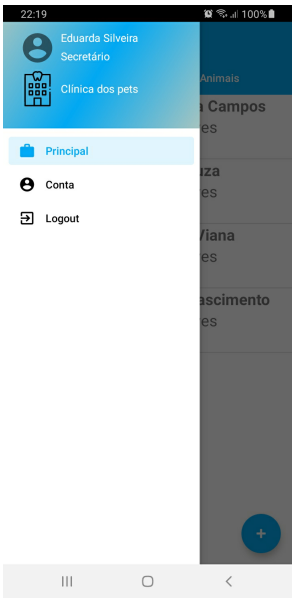
Fonte: O Autor (2021).

4.5.6 Dados de usuário

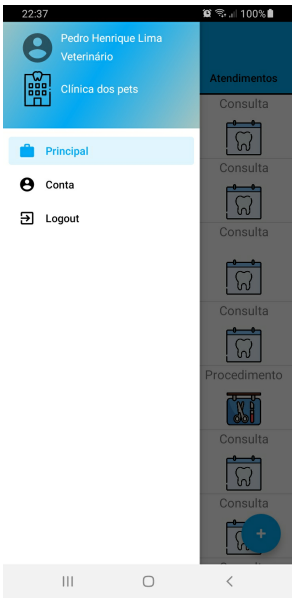
Acessando o sistema, um usuário também conta com um menu de opções. Nas figuras 27a e 27b podem ser vistas as opções disponíveis para o secretário e veterinário, respectivamente.

Nesse menu, ao contrário do administrador, o usuário pode acessar a tela para somente alterar seus dados cadastrais, demonstrados na Figura 27c. Estes são acessados por meio da opção “Conta”.

Figura 27 – Interface gráfica - Menu com opções disponíveis para os usuários



(a) Opções para o secretário



(b) Opções para o veterinário



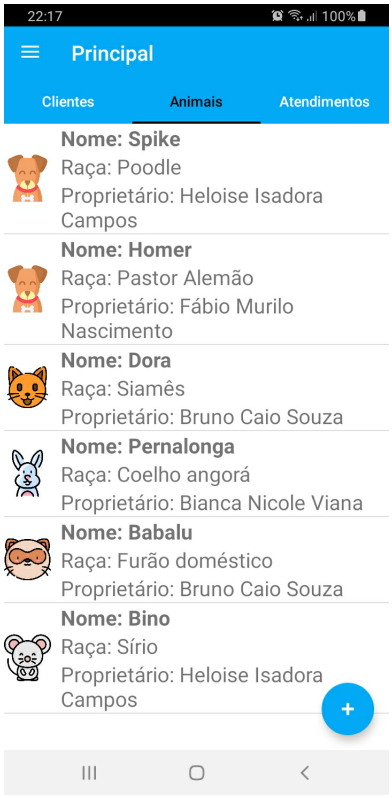
(c) Edição de conta do usuário

Fonte: O Autor (2021).

4.5.7 Abas do sistema

O aplicativo em sua tela principal, conta com três abas, conhecidas como *tabs*. São elas: Clientes, Animais, e Atendimentos, respectivamente. Todas elas, ao acessá-las, exibem uma lista com seus dados já inseridos, caso contrário, pode ser criado um novo registro através do botão de adicionar, representado pelo símbolo de "+", na parte inferior. Também é possível editar estes registros, selecionando-os, e sendo dirigido para a tela dos dados do mesmo. A Figura 28 exibe uma representação das abas contidas no aplicativo, aberta na aba de cadastro de animais.

Figura 28 – Interface gráfica - Abas do sistema



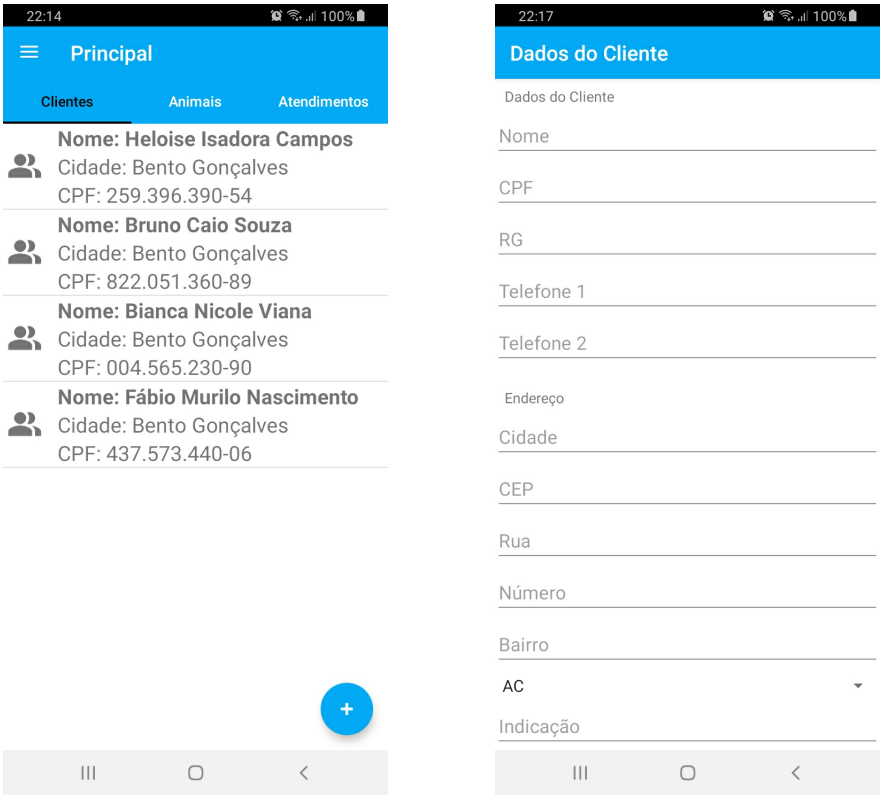
Fonte: O Autor (2021).

4.5.8 Cadastro de clientes

A primeira aba possui finalidade de cadastrar clientes, estando disponível ao secretário, veterinário, ou a quem possuir a permissão de cadastro de clientes.

A Figura 29a ilustra a lista de clientes que é exibida quando há registros. Na parte inferior, está a opção de inserir um novo. Já a Figura 29b demonstra a tela de inserção e edição com os campos dos dados de um cliente.

Figura 29 – Interface gráfica - Cadastro de clientes



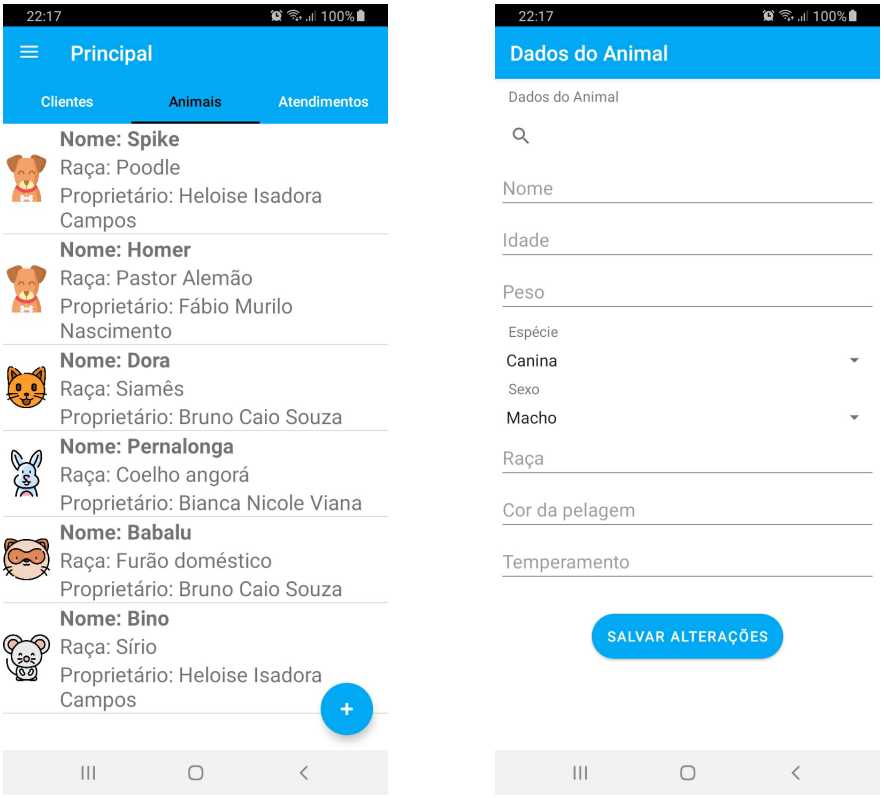
(a) Lista de clientes (b) Dados do cliente

Fonte: O Autor (2021).

4.5.9 Cadastro de animais

Após um cliente ser cadastrado no sistema, os animais a ele pertencentes podem ser inseridos por meio da segunda aba, com nome de “Animais”. Observa-se esta funcionalidade na Figura 30a, onde exibe-se a lista dos animais já inseridos, junto ao botão de adicionar um novo, na parte inferior. Para adicionar, são preenchidos os campos exibidos pela Figura 30b.

Figura 30 – Interface gráfica - Cadastro de animais



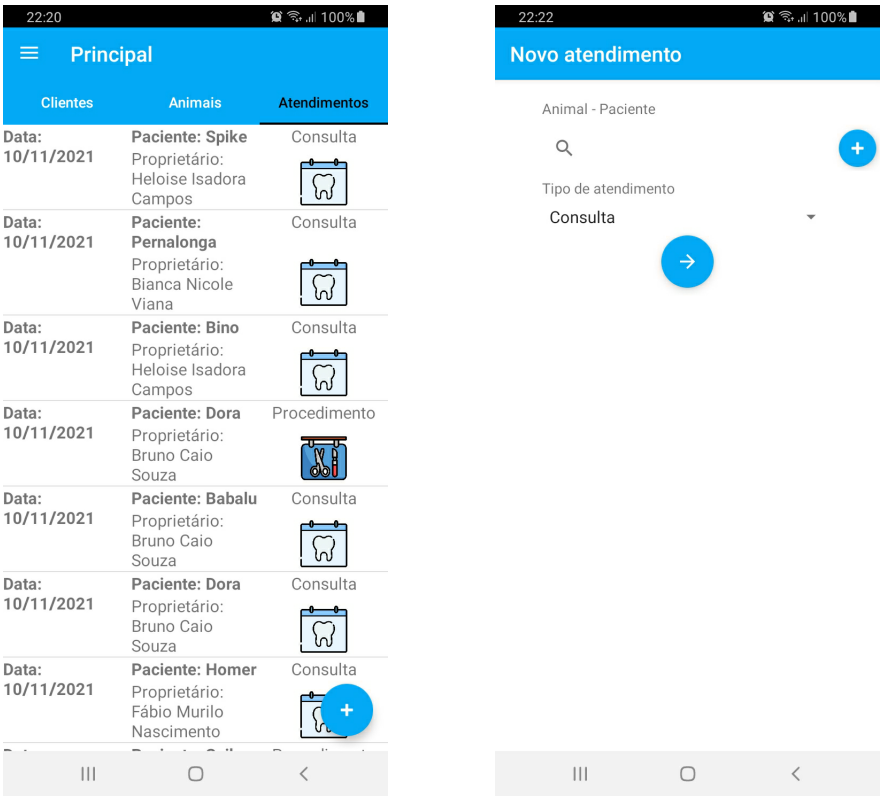
(a) Lista de animais (b) Dados do animal

Fonte: O Autor (2021).

4.5.10 Realização de atendimento

A terceira aba à direita exibe a lista de atendimentos já realizados na clínica, como pode ser visto na Figura 31a. Somente o veterinário pode realizar um atendimento.

Figura 31 – Interface gráfica - Atendimentos



(a) Aba de atendimentos (b) Dados iniciais

Fonte: O Autor (2021).

Ao criar um novo atendimento, o profissional veterinário preencherá dois dados iniciais. Neles, são inseridos o animal a ser atendido, e qual será o tipo de atendimento realizado, como representado pela Figura 31b. Estes dados são necessários, pois é por meio deles que uma ficha específica do animal inserido é trazida, pois uma ficha de consulta difere de uma ficha de procedimento, conforme sua espécie.

4.5.11 Preenchimento de ficha de atendimento

Na Figura 32, é ilustrado como estão organizados os dados das fichas de atendimento no aplicativo. Neste exemplo, foram utilizados os dados da ficha de exame clínico canina, porém, a ficha pode variar conforme a espécie do animal.

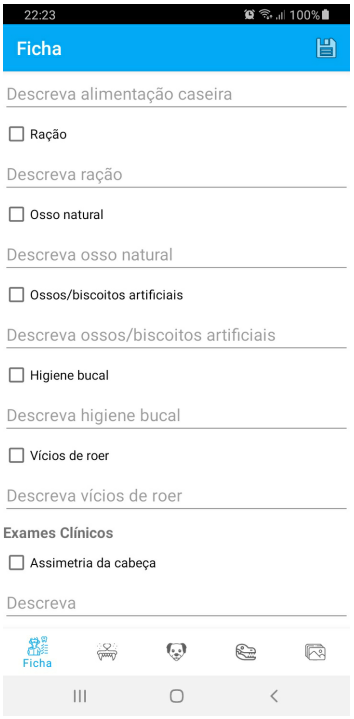
Na parte inferior há uma barra de navegação com botões que levam o usuário a diferentes janelas que exibem as representações gráficas do animal, como odontogramas, cavidades orais, crânios, ou para a galeria de imagens do atendimento, de acordo com o selecionado.

Ao fim da rolagem da tela de preenchimento das fichas, observa-se na Figura 32d, que há um campo de observações, onde o veterinário pode inserir um texto para descrever o seu atendimento.

Figura 32 – Interface gráfica - Tela de atendimento com uma ficha clínica



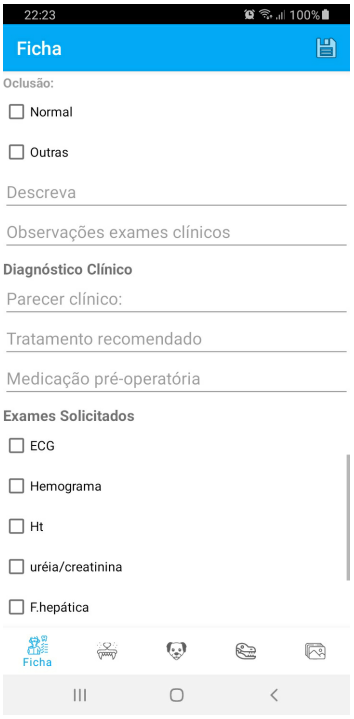
(a) Parte 1



(b) Parte 2



(c) Parte 3



(d) Parte 4

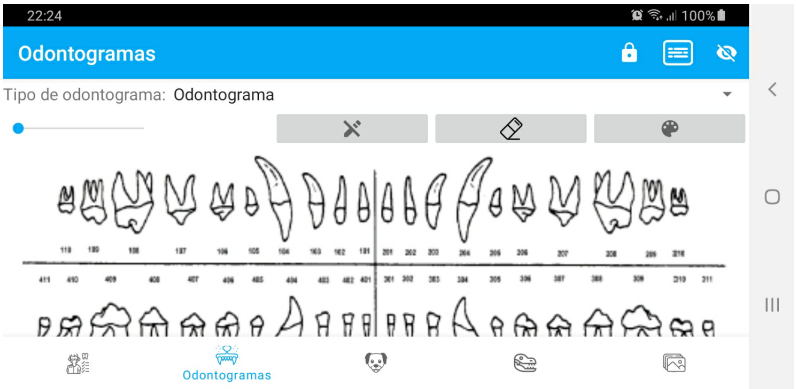
Fonte: O Autor (2021).

4.5.12 Representações gráficas

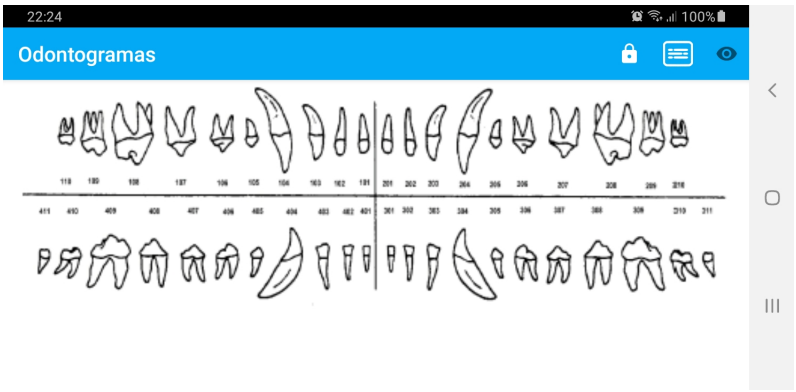
Foram inseridas telas que exibem uma visualização exclusiva de cada representação gráfica do animal tratado, como odontogramas, cavidades orais, e crânio. Junto às mesmas, existe uma caixa de ferramentas de desenho para o veterinário desenhar sobre elas, a fim de indicar problemas ou procedimentos. No desenho, é possível determinar o tamanho do lápis, cor, ou apagar o mesmo.

Na ilustração das representações, como pode-se observar na Figura 33, devido ao tamanho da mesma e por esta necessitar de um maior espaço de área na tela, para facilitar o desenho, utilizou-se a orientação horizontal do dispositivo. Juntamente, na barra de ações superior, situam-se três ícones que aplicam diferentes ações sobre as representações. O ícone de cadeado serve para travar a rolagem da tela, nos casos em que a representação ocupe maior espaço vertical. Também há a opção de legenda para exibir as abreviações e siglas utilizadas nos atendimentos, que identificam anormalidades, como observa-se no exemplo da Figura 33c que demonstra as abreviações utilizadas nos odontogramas. Já o ícone mais à direita, representado pelo ícone de um olho, auxilia na melhor visualização da representação, ocultando as ferramentas de desenho e o menu de navegação inferior, fazendo assim, com que fique maior espaço disponível na tela, como demonstrado na Figura 33b

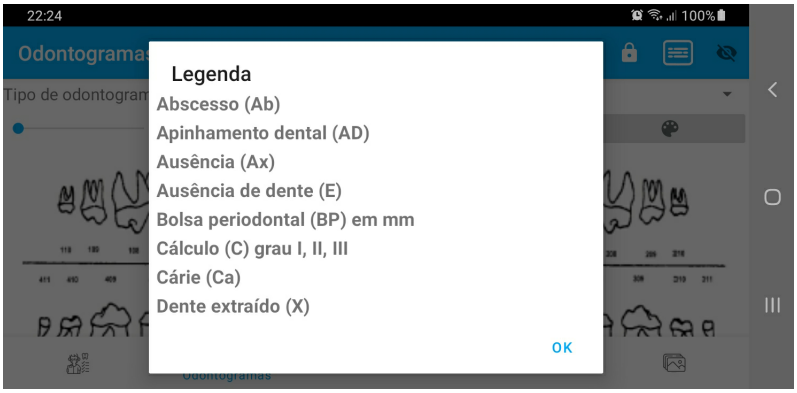
Figura 33 – Interface gráfica - Atividade dedicada ao odontograma



(a) Odontograma



(b) Odontograma com componentes ocultados

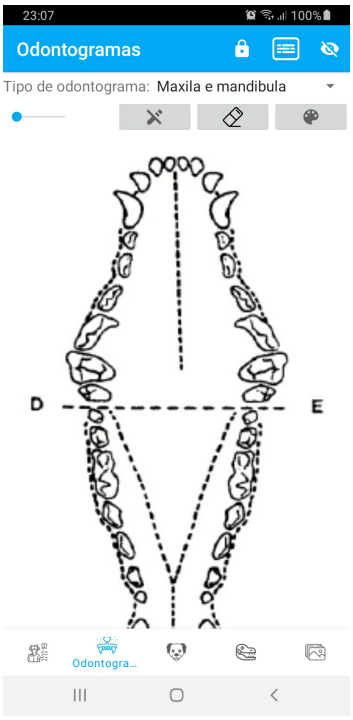


(c) Odontograma com quadro de legenda

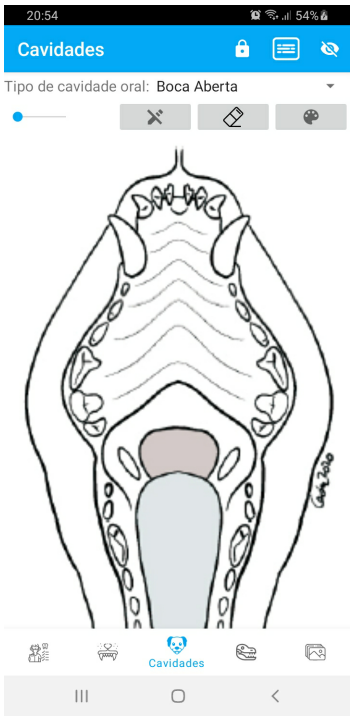
Fonte: O Autor (2021).

Para as ilustrações da maxila e mandíbula, cavidades orais, e crânio, por as mesmas terem um formato de desenho diferente, foi utilizada a orientação vertical do dispositivo, como ilustrado na Figura 34.

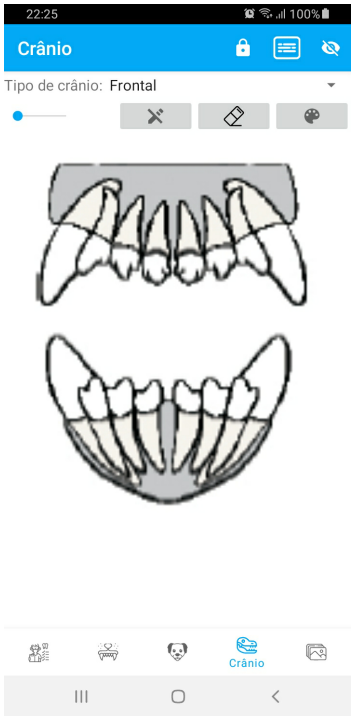
Figura 34 – Interface gráfica - Tela com representações gráficas da maxila e mandíbula, cavidades orais e crânios



(a) Maxila Mandíbula



(b) Cavidade oral aberta



(c) Crânio frontal

Fonte: O Autor (2021).

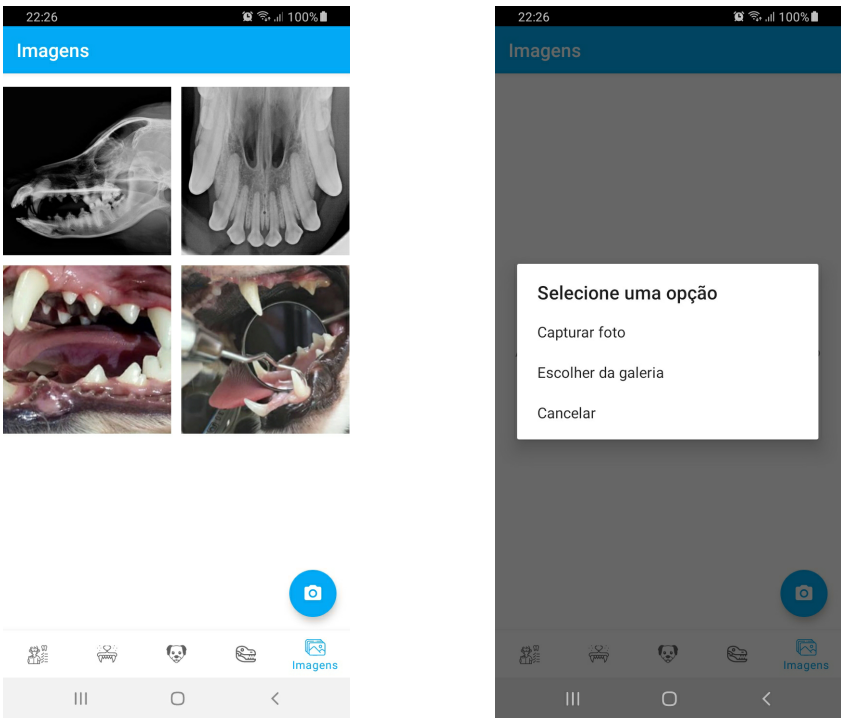
Para desenhar sobre as representações gráficas, utilizou-se uma biblioteca externa que faz uso do componente *Canvas* do Android, que é explicada na Seção 5.3.

4.5.13 Inserção e visualização de imagens no atendimento

Como pode ser visto, na parte inferior da tela de atendimento, há uma barra de navegação com opções que levam o usuário a realizar uma ação.

Na opção de imagens, o usuário conta com uma galeria de imagens do atendimento, que podem ser inseridas e visualizadas, como demonstra-se na Figura 35.

Figura 35 – Interface gráfica - Imagens do atendimento



(a) Imagens do atendimento (b) Opções de inserção

Fonte: O Autor (2021).

4.5.14 Exportação de ficha de atendimento

Após um atendimento ser finalizado e salvo, o mesmo é exibido na lista contida na aba de atendimentos.

Ao manter selecionado um elemento da lista de atendimentos, surge um ícone na barra de ações superior, que possibilita a exportação do atendimento para um documento visualizável, ilustrado na Figura 36. Após a exportação ser feita, é aberta uma lista de destinatários do usuário, onde pode ser exportado o documento com as informações.

Figura 36 – Interface gráfica - Seleção de um atendimento com opção de exportação



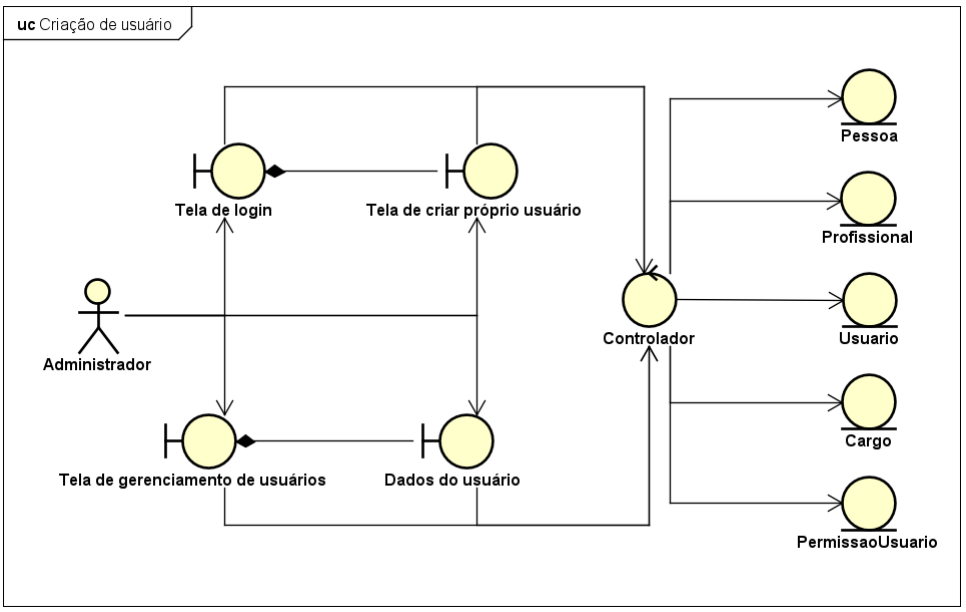
Fonte: O Autor (2021).

4.6 DIAGRAMAS DE ROBUSTEZ

Nesta seção são apresentados os diagramas de robustez do projeto, que segundo Rosemberg, Stephens & COLLINS-COPE (2005), através deles são eliminadas as ambiguidades e lacunas dos casos de uso.

A Figura 37 ilustra o diagrama de robustez que atende o caso de uso “Criar e vincular usuários”.

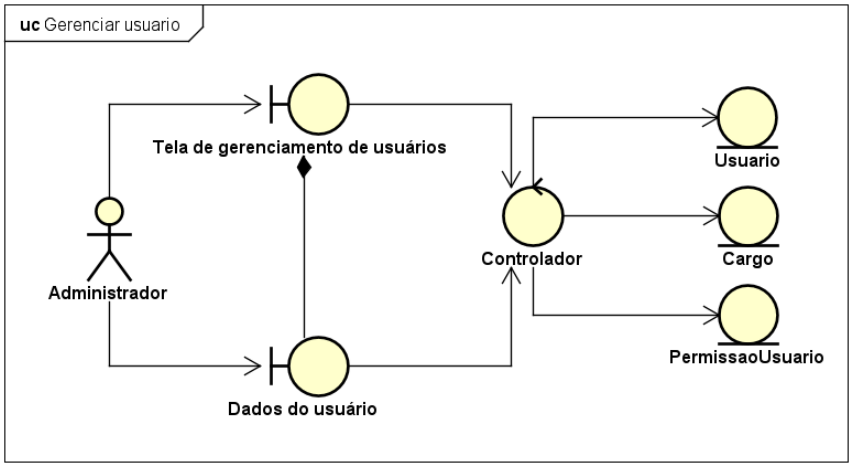
Figura 37 – Robustez - Criar usuário



Fonte: O Autor (2021).

A Figura 38 ilustra o diagrama de robustez que atende o caso de uso “Gerenciar permissões de usuário”.

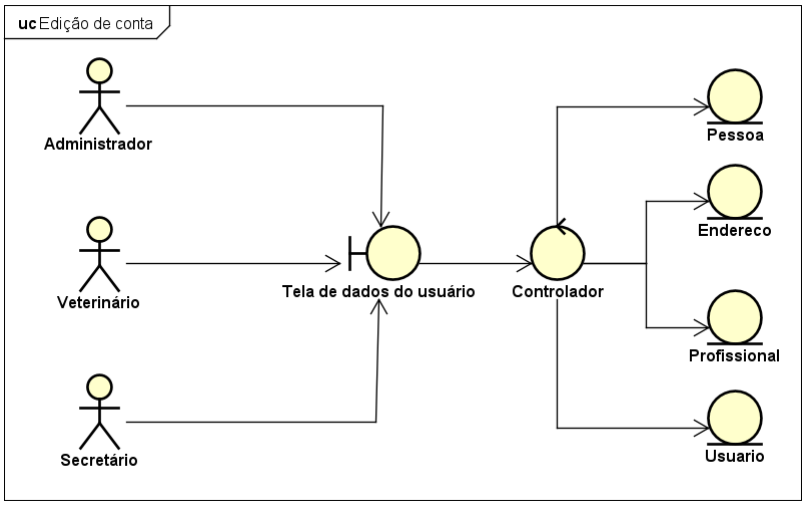
Figura 38 – Robustez - Gerenciar permissões de usuário



Fonte: O Autor (2021).

A Figura 39 ilustra o diagrama de robustez que atende o caso de uso “Editar conta de usuário”.

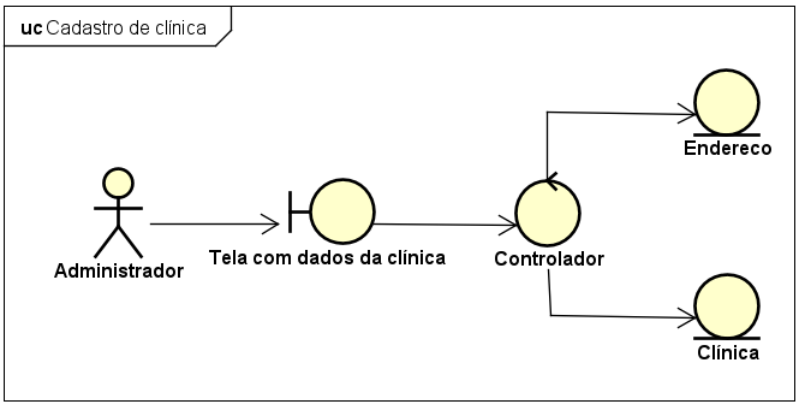
Figura 39 – Robustez - Editar conta de usuário



Fonte: O Autor (2021).

A Figura 40 ilustra o diagrama de robustez que atende o caso de uso “Cadastrar clínica”.

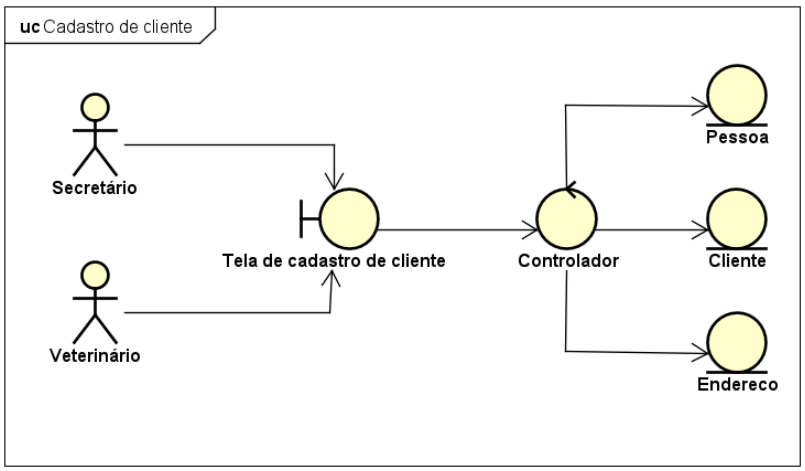
Figura 40 – Robustez - Cadastrar clínica



Fonte: O Autor (2021).

A Figura 41 ilustra o diagrama de robustez que atende o caso de uso “Cadastrar clientes”.

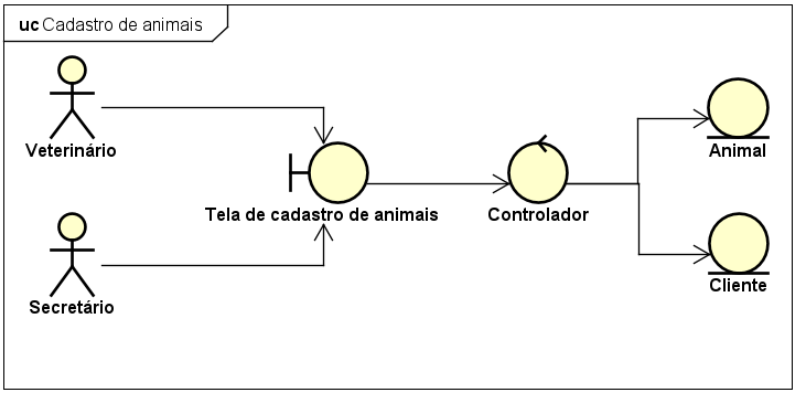
Figura 41 – Robustez - Cadastrar clientes



Fonte: O Autor (2021).

A Figura 42 ilustra o diagrama de robustez que atende o caso de uso “Cadastrar animais”.

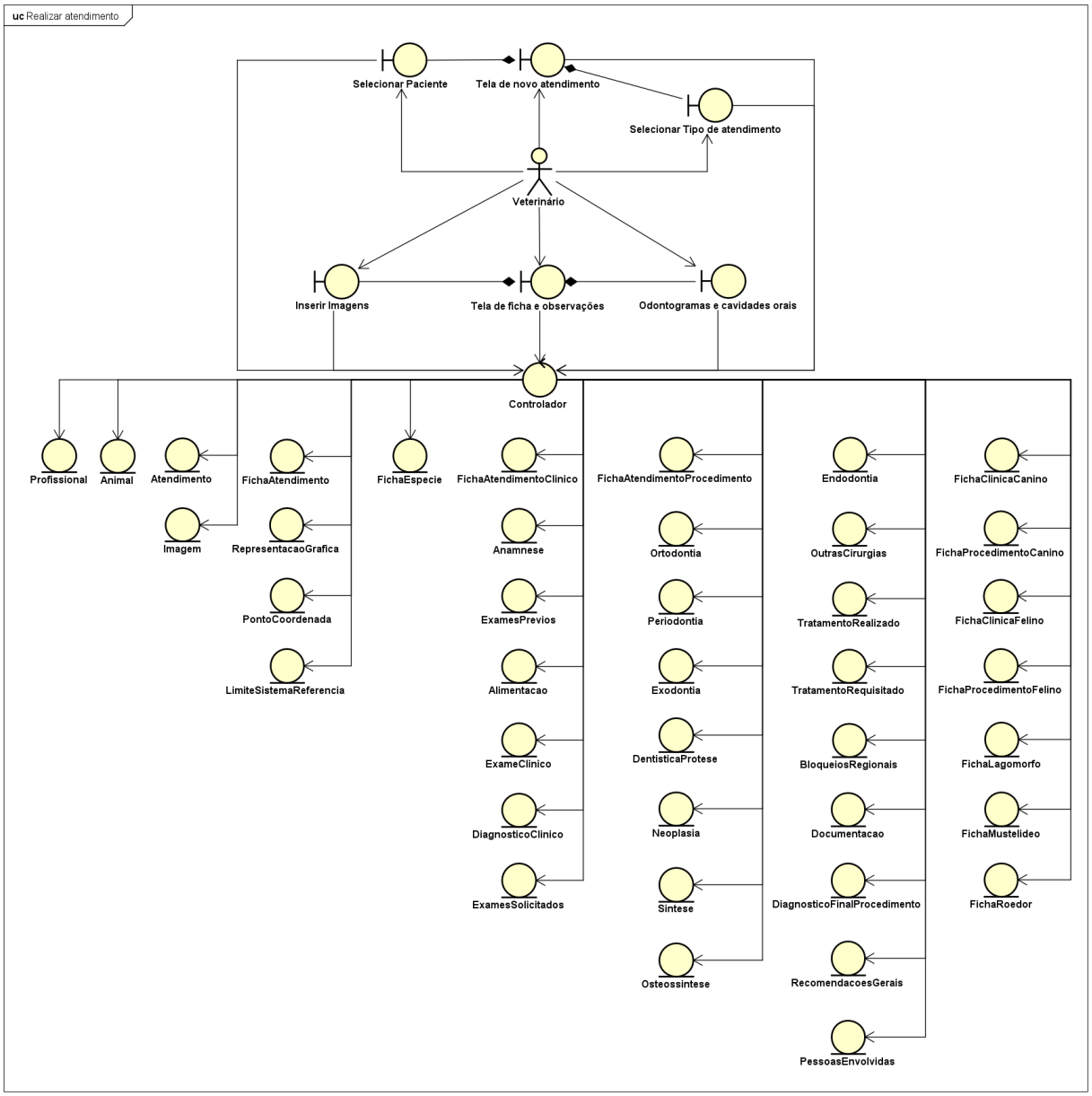
Figura 42 – Robustez - Cadastrar animais



Fonte: O Autor (2021).

A Figura 43 ilustra o diagrama de robustez que atende os casos de uso “Realizar atendimento”, “Preencher fichas de atendimento clínico e de procedimento”, “Preencher odontogramas, cavidades orais e crânios”, e “Inserir observações de atendimento no prontuário”.

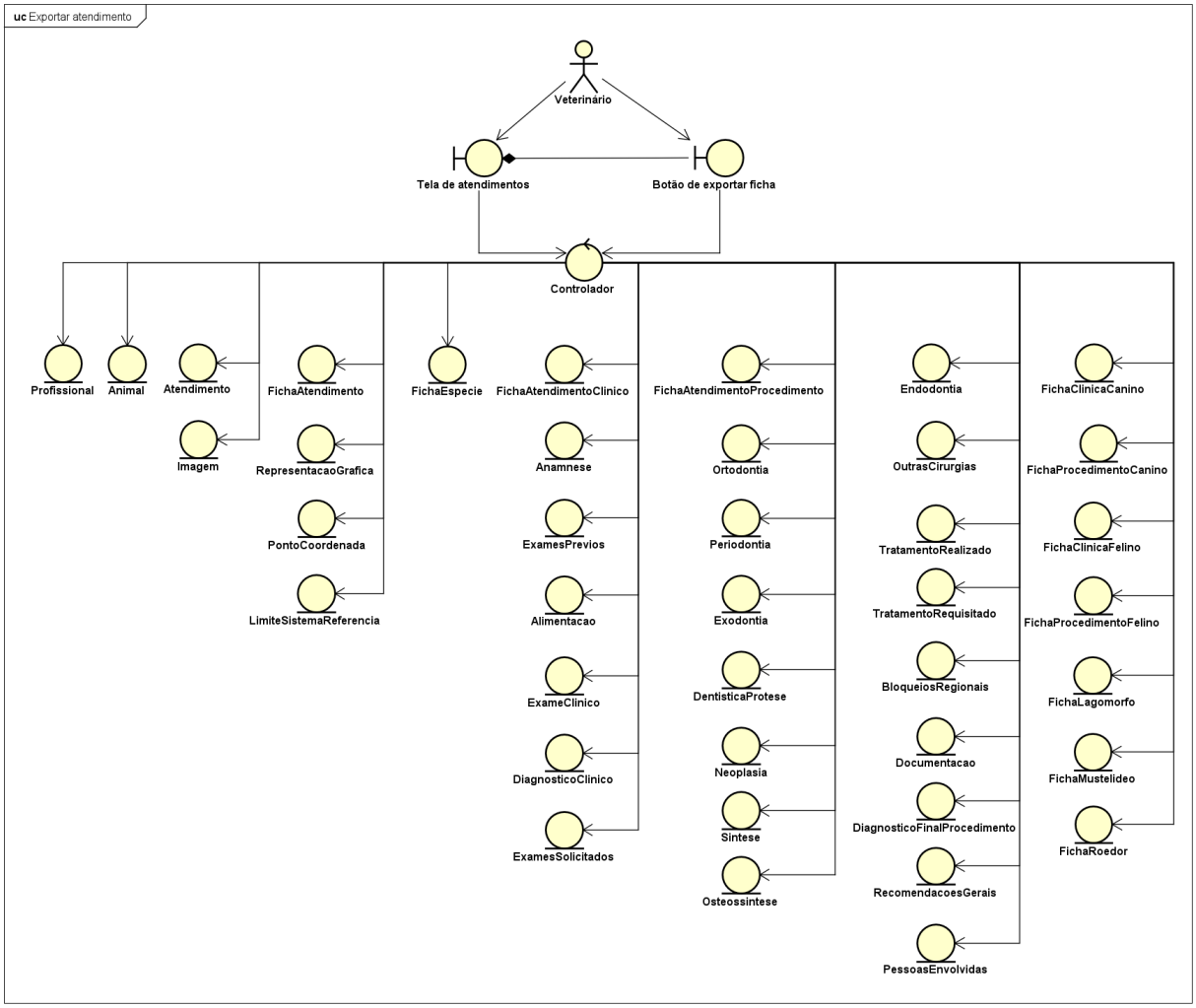
Figura 43 – Robustez - Realizar atendimento



Fonte: O Autor (2021).

A Figura 44 ilustra o diagrama de robustez que atende o caso de uso “Exportar fichas de atendimento”.

Figura 44 – Robustez - Exportar fichas de atendimento



Fonte: O Autor (2021).

5 IMPLEMENTAÇÃO

Este capítulo expõe de que modo foi desenvolvida a aplicação móvel apresentada. Na Seção 5.1, são explicadas as ferramentas e configurações dos *softwares* instalados, juntamente às especificações técnicas da máquina utilizada para produzir o aplicativo. Na Seção 5.2, é abordada a arquitetura de projeto utilizada, e os componentes que integram a mesma. Na Seção 5.3 explica-se como se deu o processo de desenvolvimento dos casos de uso.

5.1 INSTALAÇÃO DAS FERRAMENTAS DE DESENVOLVIMENTO

Todo o projeto foi desenvolvido em um computador *Desktop*, configurado com 8,00 *Gigabyte (GB)* de memória *Random Access Memory (RAM)* instaladas, e operado pelo sistema operacional Windows 10, na versão Pro. Não fez-se necessário o uso de um sistema operacional específico, devido ao conjunto de desenvolvimento Android ser suportado pela maioria dos sistemas.

No primeiro momento, instalou-se na máquina, a IDE Android Studio, juntamente ao emulador para Android que acompanha o pacote de instalação. O intuito deste emulador, é simular um dispositivo Android, a fim de executar os projetos e realizar testes dentro da máquina onde está sendo realizado o desenvolvimento, para verificar se tudo está ocorrendo conforme o planejado. Desse modo, pretendeu-se utilizar este emulador, porém, o mesmo fez alto uso de memória RAM, não sendo suficiente o já instalado na máquina. Para isso então, substituiu-se este método pelo modo de depuração por vínculo em um *smartphone* modelo Samsung Galaxy S9+, de uso pessoal. Através desta alteração, ao efetuar testes de execução do aplicativo no dispositivo, foram mostrados resultados satisfatórios.

Após a instalação do ambiente, criou-se um novo projeto em branco, e em seguida, realizou-se a integração da plataforma Firebase no mesmo. Este processo foi feito de forma rápida e eficiente, bastando inserir no serviço, o certificado de assinatura presente no projeto do aplicativo. Posteriormente, a partir dos parâmetros fornecidos, foi feito o *download* de um arquivo de configuração gerado, e este foi adicionado nas dependências do projeto. Através destas dependências, serviços do Firebase foram vinculados para auxiliar no desenvolvimento das funcionalidades, tais como:

- *Firebase Authentication*: foi usado para criar contas de usuário, onde para fazer o cadastro e acesso, os mesmos utilizam a opção de *login* por *e-mail* e senha.
- *Firebase Realtime Database*: foi usado para armazenar e recuperar os dados em tempo real.

- *Cloud Storage*: foi usado para armazenar documentos de mídia, como as representações gráficas das espécies, e as imagens capturadas nos atendimentos.

Depois dos ambientes serem configurados, e ter feito a integração com o banco de dados, foi iniciada a codificação propriamente dita. Devido ao tamanho do projeto ser de porte consideravelmente grande, fez-se necessário manter um histórico de alterações do mesmo, a fim de evitar possíveis perdas, e registrar tudo o que foi feito. Para isso, utilizou-se a plataforma de controle de versões GitHub[1] já integrada no conjunto de ferramentas do Android Studio, para hospedagem dos arquivos.

5.2 COMPONENTES E ARQUITETURA DO PROJETO

Durante o processo de desenvolvimento do sistema, aplicou-se métodos de organização arquitetural de *software*, a fim de garantir uma maior qualidade na produção, e facilitar o entendimento em futuras manutenções. Como a linguagem utilizada detém um paradigma de programação Orientado a Objetos, aplicou-se boas práticas de programação, fazendo uso de padrões de nomenclaturas de variáveis, e de padrões de projeto. A arquitetura de *software* escolhida foi a *Model-view-viewmodel (MVVM)*, sendo esta considerada a arquitetura padrão de desenvolvimento pela plataforma. A mesma corresponde em separar a camada de negócio, denominada *model*, da camada de interface de usuário, *view*, para assim serem conectadas pela terceira camada, a *viewmodel*.

Além da arquitetura MVVM, foram empregues os componentes *Data Binding* e *LiveData*, presentes no pacote nativo da plataforma. Através do *Data Binding*, é feita uma melhor referência entre os itens de *layout* pertencentes a uma interface gráfica, que podem ser de uma *activity*, ou de um *fragment*. Nisso, são listados apenas os componentes que pertencem ao código fonte da mesma.

Já o componente *LiveData*, é um componente que atende o padrão *Observer*, sendo este útil para transferir os dados em tempo real do banco de dados Firebase. Logo, ao ser detectada uma alteração nos dados, os mesmos são instantaneamente exibidos nos componentes do *Data Binding*.

5.3 ETAPAS DE DESENVOLVIMENTO

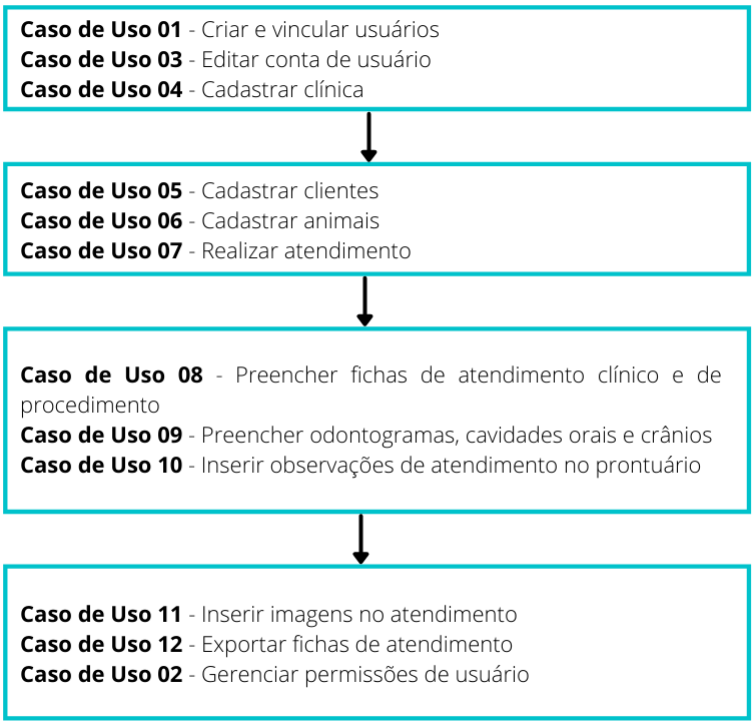
O desenvolvimento das funcionalidades da aplicação foi baseado nos casos de uso e requisitos funcionais apresentados no Capítulo 4.

Iniciou-se elaborando toda a parte visual do aplicativo, mais conhecida como *front-end*, que consiste nas interfaces gráficas de interação com o usuário. Esta decisão foi tomada

[1] <https://github.com/>

com a finalidade de deixar preparada toda a estrutura de utilização do aplicativo, como telas de cadastros, *login*, e fichas, para em seguida, desenvolver a lógica de programação por trás dos casos de uso descritos. A Figura 45 demonstra em grupos, a prioridade e sequência de desenvolvimento dos casos de uso.

Figura 45 – Etapas de desenvolvimento dos casos de uso

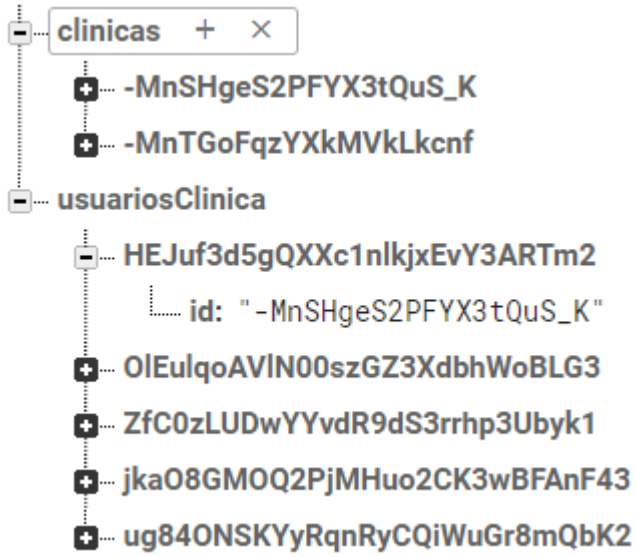


Fonte: O Autor (2021).

Optou-se por iniciar pelos cadastros de clínicas e usuários, para definir em qual clínica cada usuário estará inserido. De modo, os mesmos podem prosseguir cadastrando no local correto, os clientes, animais, e atendimentos. Pelo fato do banco de dados utilizar uma árvore JSON em sua estrutura, e esta ser do tipo não relacional, a forma de validar onde cada usuário irá registrar as informações criadas, é por meio de sua autenticação no aplicativo, que é verificada no momento do acesso. Para isso, foi criado separadamente um nó de clínicas, e outro para usuários das clínicas, como mostrado na Figura 46.

No interior do nó “clinicas”, são criados outros nodos com um identificador único para armazenar os dados de uma clínica, à medida que forem cadastradas. Nesta etapa, houve um problema para saber em qual nó um usuário autenticado deve ser direcionado para poder salvar os seus dados, pois o acesso a um valor é feito por meio da referência à um filho de um nó. Nisso, criou-se o nodo auxiliar “usuariosClinica”, onde em cada filho do mesmo, é inserido um registro único que distingue cada usuário, nomeado com um identificador gerado na sua criação. Este nodo possui como filho, o identificador da clínica em que está vinculado. Nisso, o conteúdo contido neste respectivo usuário, direciona o acesso do mesmo até o nó da clínica

Figura 46 – Organização estrutural do banco de dados



Fonte: O Autor (2021).

onde serão salvos ou recuperados os dados.

Posteriormente, após definir a estrutura de organização dos dados para o banco, foram implementadas as possibilidades de cadastrar clientes, animais, e atendimentos, respectivamente, pois para haver um atendimento, é necessário ter um animal cadastrado, e este, deve pertencer a um cliente. Nas telas de cadastros de dados, utilizou-se *layout* com rolagem (*scroll*), no intuito de comportar todos os campos.

Nesta etapa de preparação de cadastros, pelo fato de não possuir muita familiaridade no uso da plataforma Firebase, encontrou-se algumas dificuldades na utilização dos dados obtidos pelas consultas na mesma, devido ao uso da filosofia de programação assíncrona. Ao se requisitar um dado, existe um tempo de espera para o serviço retorná-lo. Deste modo, ao chamar um método no código fonte que necessita de uma requisição, haviam problemas durante a execução, por esta processar instruções com um dado que ainda não estava disponível, causando assim, falhas em tempo de execução. Entretanto, após buscar soluções, fez-se uso de interfaces com *listeners*, para que somente após receber uma resposta, uma ação fosse tomada. Um exemplo deste acontecimento, foi ao inicializar o aplicativo, onde o carregamento das abas com os dados cadastrais de clientes, animais e atendimentos eram concluídos antes de receber os dados do Firebase. Para resolver este problema, criou-se uma *Splash screen*, que consiste em exibir uma tela de carregamento enquanto o aplicativo não recebe a resposta com os dados. Após recebê-los, esta tela é fechada, e o aplicativo é aberto em seu menu principal, com todos os seus dados já disponíveis para serem manipulados.

Finalizados os cadastros, dentro da tela de atendimento desenvolveu-se as fichas de atendimento de consulta e de procedimento. Nestas estão incorporados todos os campos e atributos pertencentes à cada tipo de ficha. Devido à grande quantidade de dados contidos em uma ficha,

também se utilizou a estratégia de *layout* com rolagem (*scroll*), a fim de comportar todos os campos em uma única tela. Na mesma exibição, no menu de navegação inferior, localizam-se os botões para o usuário acessar as interfaces para desenhar sobre as representações gráficas, como os odontogramas, cavidades orais, crânios, e para inserir imagens de atendimento na galeria de imagens. Todas estas representações gráficas estão armazenadas no serviço *Storage* do Firebase, que também foi responsável por guardar as imagens dos atendimentos, pois o mesmo tem como propósito, armazenar arquivos de mídia.

No uso do componente de menu inferior, percebeu-se que ao manipular os campos de uma interface, e mudar o foco de visualização para outra, os mesmos eram redefinidos, perdendo assim todos os seus dados de preenchimento. Nisso, pesquisando-se a respeito sobre o ciclo de vida do Android, compreendeu-se que era necessário persistir os dados para após recuperá-los. Nisso, empregou-se o componente *ViewModelProvider* para quando pausada a visualização em uma tela, os dados fossem enviados para a sua *viewmodel* específica, e recuperados, para serem inseridos nos campos ao resumir a visualização novamente.

As representações gráficas, em cada atendimento, são filtradas de acordo com o tipo selecionado em um componente do tipo *Spinner*, e estas validadas conforme a espécie do animal. No caso do odontograma, que ocupa a orientação horizontal do dispositivo, também foi necessário guardar o estado de visualização da tela, pois quando o dispositivo tem sua orientação alterada por rotação de tela, todos os componentes têm os seus valores redefinidos. Neste caso, também se utilizou o *viewmodel* do seu respectivo *fragment* para resolver este problema.

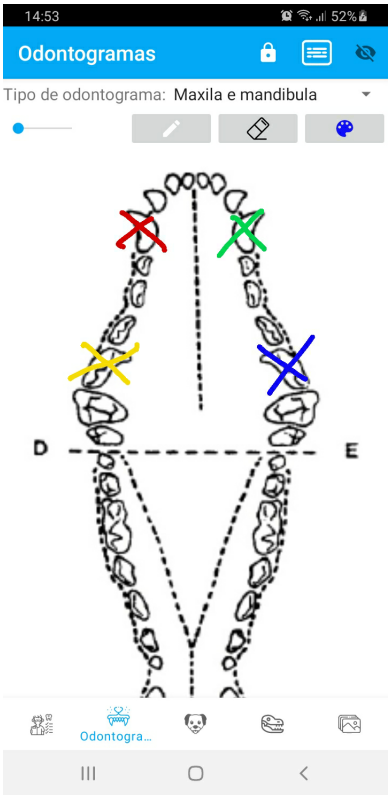
Por possibilitarem desenho sobre as mesmas, as representações gráficas compartilham de um mesmo *layout*, onde utilizou-se um componente da biblioteca *PhotoView*[2] para exibir as imagens. Ao invés de empregar o componente nativo *ImageView*, que traz somente uma simples visualização de imagem, a biblioteca permite várias operações adicionais, como a possibilidade de uso de *zoom*, que poderá ser implementado futuramente. Posicionou-se acima da representação, um componente de desenho, obtido através de uma outra biblioteca chamada *SignatureView*[3]. Esta biblioteca demonstrou-se muito útil, pois seu pacote de funções permite desenhar em uma área específica da tela, além de possibilitar escolher o tamanho, cor e também apagar os desenhos feitos. Dessa forma, optou-se por empregá-la no projeto, devido à fácil integração, e simplificação de todo o processo de criação de métodos de desenho. Junto ao componente, para facilitar ao usuário escolher uma cor para o lápis, usou-se uma biblioteca chamada *AmbilWarnaDialog*[4], que consiste em exibir uma janela de diálogo para o usuário escolher uma cor desejada. Quando o mesmo escolhe, o valor referente a cor é enviado ao *SignatureView*, para assim, os desenhos ficarem de acordo com a cor escolhida, como mostra a Figura 47, que enfatiza o resultado da junção dos três componentes.

[2] <https://github.com/Baseflow/PhotoView>
[3] <https://github.com/zahid-ali-shah/SignatureView>
[4] <https://github.com/yukuku/ambilwarna>

Figura 47 – Odontograma com possibilidade de desenho e escolha de cor



(a) Diálogo de seleção de cor



(b) Odontograma preenchido com coloração

Fonte: O Autor (2021).

Em virtude de cada tipo de representação gráfica ter formatos diferentes, consequentemente, seu tamanho também varia. Para isso, a fim de posicionar corretamente o componente de desenho sobre a imagem, necessitou-se alterar o posicionamento do *layout* em tempo de execução. Dessa forma, a classe *ConstraintSet* auxiliou na correta conexão em cada borda de um componente e de outro, no *ConstraintLayout* utilizado.

Após um desenho ser feito sobre a representação gráfica, necessita-se persistir o mesmo juntamente a todos os dados de uma ficha, para assim, quando necessário visualizá-los, encontrar também os pontos desenhados. Nisso, buscou-se uma forma de capturar estes pontos através dos gestos de toque, utilizando as coordenadas da tela. Estas coordenadas remetem ao local X (horizontal) e Y (vertical) do dispositivo, e os mesmos, enquanto desenhados, foram inseridos em uma lista de objetos que guardam estes atributos, junto ao tamanho em Densidade de pixel (DP), e a cor do ponto desenhado. Após salvos, procurou-se uma maneira de redesenhar estes pontos no componente *SignatureView*. Pesquisando o seu funcionamento, descobriu-se que o mesmo faz uso de um objeto do tipo *Bitmap* que pode ser gerado a partir do desenho feito, e exportado, além de poder ser inserido no componente. Criou-se então um objeto *Bitmap* auxiliar, inicialmente vazio, onde foram inseridos todos os pontos capturados, junto aos seus outros atributos, para posteriormente este objeto ser injetado no componente. Encontrou-se as-

sim, uma adversidade, pois o local de desenho feito em um dispositivo pode não ser o mesmo em outro, devido ao tamanho de telas serem diferentes. Consequentemente, fez-se necessário realizar o processo de mapeamento de coordenadas, abordagem pertencente da área de Computação Gráfica, onde encontram-se o Sistema de Referência do Universo (SRU) e o Sistema de Referência do Dispositivo (SRD). O SRU consiste em um plano genérico universal, com limites mínimos e máximos para o tamanho de altura e largura. Admitiu-se o valor -1 e +1 para largura, e -1 e +1 para altura.

Já o SRD especifica os limites mínimos e máximos para o plano do tamanho de tela do dispositivo. Neste, tomou-se como referência o tamanho da imagem, que deve ocupar o maior espaço disponível da largura e da altura da tela. Logo o tamanho da largura foi de mínimo 0, até o valor máximo obtido, sendo armazenado em uma variável. A mesma metodologia foi aplicada para a altura.

Com essa estratégia, os pontos capturados do dispositivo SRD foram convertidos para as coordenadas do universo SRU, para assim, serem convertidos para qualquer tamanho de tela, quando necessário. Logo, este processo foi feito utilizando as seguintes fórmulas, extraídas de Junior (2006), onde especifica-se:

Para o ponto de coordenada X, utilizou-se a fórmula

$$\frac{Xd - XdMin}{Xu - XuMin} = \frac{XdMax - XdMin}{XuMax - XuMin}, \tag{5.1}$$

Para as coordenadas horizontais do dispositivo SRD, define-se Xd como o ponto X clicado ou a ser encontrado; XdMin, como o ponto mínimo de X, e XdMax, o ponto máximo.

Para as coordenadas horizontais do universo SRU, denota-se a variável Xu como o ponto existente ou a ser encontrado; XuMin o valor mínimo de X, e XuMax o valor máximo de X.

Já para a coordenada Y, utilizou-se a fórmula

$$\frac{Yd - YdMax}{Yu - YuMin} = \frac{YdMin - YdMax}{YuMax - YuMin}, \tag{5.2}$$

Para as coordenadas verticais do dispositivo SRD, define-se Yd como o ponto Y clicado, ou a ser encontrado; YdMin, como o ponto mínimo de Y, e YdMax, o ponto máximo.

Para as coordenadas verticais do universo SRU, denota-se a variável Yu como o ponto existente ou a ser encontrado; YuMin o valor mínimo de Y, e YuMax o valor máximo de Y.

Concluída a implementação das representações gráficas, iniciou-se o desenvolvimento da inserção de imagens no atendimento. Criou-se então, uma galeria de imagens, com uso do componente *GridView* do Android. Na estratégia, permitiu-se que o usuário decida entre capturar uma imagem através da câmera, ou buscá-la da galeria de fotos. Para realizar estas operações no dispositivo, foi necessário que o aplicativo solicite ao usuário, permissões para obter acesso

à câmera e ao armazenamento do dispositivo, a fim de que possam ser realizadas estas ações corretamente. Estas solicitações são feitas na primeira vez que o usuário utiliza o aplicativo, após seu *login* ser validado no acesso. Após selecionar e inserir as imagens desejadas durante o atendimento, as mesmas são listadas na galeria, e ao clicar sobre alguma em específico, a mesma é aberta em uma nova janela para poder ser visualizada em tamanho maior, com suporte a *zoom*. Ao salvar um atendimento, é criado um nodo de atendimentos na clínica, onde cada um de seus filhos terá seu identificador, com todos os atributos de fichas, pontos de coordenadas de desenho, e os *links* de acesso das imagens inseridas no serviço *Storage*.

No decorrer do desenvolvimento das fichas de atendimento de cada espécie, devido a não ter obtido tempo suficiente, foram implementadas duas únicas fichas de procedimento e de consulta para serem utilizadas em todos tipos de espécies. Entretanto nestas, as representações gráficas de cada espécie estão disponíveis para cada uma. Dessa forma, mesmo que no diagrama de domínio da Seção 4.4 estejam representadas as classes das fichas com as interfaces de cada tipo de espécie de animal, como por exemplo, ficha de mustelídeos, lagomorfos, e roedores, as mesmas ainda estão em processo de desenvolvimento. Sendo assim, até o momento, utilizou-se apenas as classes “FichaAtendimentoClinico”, e “FichaAtendimentoProcedimento”, para assim, serem utilizados todos os campos de dados.

Após concluir o desenvolvimento de todos os casos de uso relacionados ao atendimento, tendo em vista que os dados do mesmo já podem ser extraídos e visualizados, restou somente realizar a exportação das fichas. Nisso, gerou-se um *layout* simples em formato *Portable Document Format (PDF)* para as fichas de consulta e de procedimento, com visualização das representações gráficas. Esta exportação não necessitou de uso de bibliotecas externas, devido ao Android fornecer uma ferramenta nativa, de fácil utilização, chamada *PdfDocument*[5], que permite criar páginas uma a uma, gravando conteúdos nela.

Deixou-se a implementação do gerenciamento de usuários por último, visto que as permissões de acesso dos mesmos seriam melhor e mais facilmente elaboradas após todas as funcionalidades estarem presentes no aplicativo, o que facilitaria também no desenvolvimento e na realização de testes. A fim de possibilitar que o usuário com cargo de administrador consiga criar e adicionar um novo usuário, ao buscar soluções na internet, encontrou-se que não é possível realizar esta operação a partir do usuário logado, pois fazendo este procedimento, o mesmo teria sua sessão encerrada e substituída pela criada, necessitando que inicie novamente. Sendo assim, a plataforma recomenda o uso do *Firebase SDK Admin*, uma biblioteca específica para gerenciamento de usuários, destinada a servidores, e sem oferecer suporte para o Android, pelo motivo de haver classes e métodos com o mesmo nome, inviabilizando-os. Fez-se necessário então obter no aplicativo, uma nova instância do banco de dados, para assim, criar uma nova classe de autenticação, e assim, ser possível de registrar um novo usuário sem o atual ser desconectado.

[5] <https://developer.android.com/reference/android/graphics/pdf/PdfDocument>

6 CONCLUSÃO

Neste trabalho foi realizada uma pesquisa bibliográfica a respeito da especialidade de Odontologia Veterinária, e também sobre a programação para dispositivos móveis, assuntos estes essenciais para desenvolver um aplicativo para *smartphones* que irá auxiliar profissionais veterinários a melhorar a forma de realizar seus atendimentos. Para isso, fez-se também a modelagem e planejamento arquitetural do projeto do *software* proposto, utilizando a metodologia de desenvolvimento Iconix, a fim de documentar e manter a rastreabilidade dos requisitos que compõem o sistema, além das classes, casos de uso, e interfaces de telas.

Implementou-se assim, um protótipo de aplicativo dirigido por todas estas etapas de modelagem de casos de uso e prototipação de telas, que contribuíram adequadamente para desenvolver sequencialmente cada funcionalidade do aplicativo, em razão de tudo estar previamente planejado.

As ferramentas de desenvolvimento empregadas, como linguagem de programação, ambiente de desenvolvimento e banco de dados demonstraram resultados satisfatórios durante o seu uso, dando destaque ao Firebase, por permitir uma fácil integração com o projeto, e também por disponibilizar diversas funcionalidades que proporcionaram ganho de tempo, e auxílio no desenvolvimento. A abordagem de desenvolvimento nativa assegurou que o aplicativo fosse desenvolvido com maior facilidade pelo desenvolvedor, devido ao mesmo possuir maior familiaridade com esta, e também por ter encontrado soluções satisfatórias em pesquisas para problemas encontrados no decorrer do desenvolvimento.

As bibliotecas utilizadas para poder efetivar os desenhos sobre as representações gráficas também auxiliaram no ganho de tempo, pois encontrou-se a funcionalidade já disponível, sem necessitar de que tudo fosse desenvolvido do início. Contudo, a fim de melhor usufruir de suas funcionalidades, fez-se necessário dedicar estudos sobre as mesmas para compreender o seu funcionamento, e também quais eram os componentes utilizados em suas estruturas.

Como trabalhos futuros, atividades de melhorias podem ser realizadas na aplicação, como otimizações, que irão manter no armazenamento interno as imagens das representações gráficas, o que diminuiria no tempo de carregamento e exportação de fichas. Outros elementos de implementação também seriam convenientes, como emprego de *zoom* na visualização das representações gráficas, no intuito de realizar um desenho mais acurado, além de concluir todas as fichas, deixando-as mais restritivas a cada espécie.

Deste modo, pelo fato do aplicativo demandar aprimoramentos em sua utilização, o mesmo não foi submetido à avaliação por profissionais da área de Odontologia Veterinária. Entretanto, os demais objetivos do trabalho mostraram-se alcançados, pois através do estudo da área de Odontologia Veterinária, conseguiu-se desenvolver um protótipo de aplicativo para o

problema apresentado.

Por fim, considera-se que o aplicativo tem características de uso promissoras para auxiliar em consultórios veterinários, e apesar do curto tempo de desenvolvimento, as funcionalidades foram implementadas, com possibilidade de serem aperfeiçoadas e aprimoradas por meio de tarefas futuras.

REFERÊNCIAS

ANIMAL DENTAL CHART INC. **Animal Dental Chart**. Erin, ON, 2021.

BARROS, A. L. B. L. **Anamnese e exame físico**: avaliação diagnóstica de enfermagem no adulto. Porto Alegre: Artmed, 2016.

BENNETT, J. **Xamarin in action**: Creating native cross-platform mobile apps. Shelter Island: Manning Publications Co., 2018.

BENSVET. Sistema veterinário bensvet. 2021. Disponível em: <https://bensvet.com.br/>. Acesso em: 19 abr. 2021.

BODUCH, A.; DERKS, R. **React and React Native**: A complete hands-on guide to modern web and mobile development with react.js. Birmingham: Packt Publishing, 2020.

BRASIL. **Resolução Nº 625**. Registro de título de especialista no âmbito dos Conselhos Regionais de Medicina Veterinária. Brasília: Conselho Federal de Medicina Veterinária, 1995.

CHENG, F. **Build Mobile Apps with Ionic 4 and Firebase**: Hybrid mobile app development. Auckland: Apress, 2018.

CLARO, D. B.; SOBRAL, J. B. M. **PROGRAMAÇÃO EM JAVA.** Florianópolis: Copyleft Pearson Education., 2008.

DIAMOND, P. Armazenamento em nuvem vs. servidores locais: nove considerações importantes. **Microsoft**, Setembro 2020. Disponível em: <https://www.microsoft.com/pt-br/microsoft-365/business-insights-ideas/resources/cloud-storage-vs-on-premises-servers>. Acesso em: 17 mai. 2021.

FIREBASE. Firebase pricing. 2021. Disponível em: <https://firebase.google.com/pricing?hl=pt-br>. Acesso em: 02 jun. 2021.

____. Firebase products. 2021. Disponível em: <https://firebase.google.com/products-build?hl=pt-br>. Acesso em: 02 jun. 2021.

GIOSO, M. A. **Odontologia**: Para o clínico de pequenos animais. São Paulo: iEditora, 2003.

____. **Odontologia veterinária**: para o clínico de pequenos animais. [S.l.]: Manole, 2007.

GOOGLE. Arquivos e dados do app. 2021. Disponível em: <https://developer.android.com/guide/topics/data?hl=pt-br>. Acesso em: 09 abr. 2021.

GORREL, C. **Odontología de pequeños animales**. Barcelona: Elsevier, 2010.

____. **Veterinary Dentistry for the General Practitioner**. [S.l.]: Elsevier, 2013.

HURIA, R. Top 10 ways you should know to store data in android. **Loginworks Softwares**, March 2018. Disponível em: <https://www.loginworks.com/blogs/top-10-ways-know-store-data-android/>. Acesso em: 24 mai. 2021.

JUNIOR, A. H. **Fundamentos de Informática**: Computação gráfica. Rio de Janeiro: LTC, 2006.

KACZMAREK, S. *et al.* **Objective-C for Absolute Beginners**: iphone, ipad and mac programming made easy. New York: Apress, 2018.

KERTESZ, P. **A Colour Atlas of Veterinary Dentistry and Oral Surgery**. London: Wolfe Publishing, 1993.

KOTLIN. Kotlin language documentation. 2021. Disponível em: <https://kotlinlang.org/docs/kotlin-reference.pdf>. Acesso em: 25 abr. 2021.

LEON, M. Odontogramas. 2020. Disponível em: <http://www.abov.org.br/link-odontogramas.html>. Acesso em: 10 mai. 2021.

LOBPRISE, H. B.; DODD, J. R. **Wiggs's Veterinary Dentistry**: Principles and practice. [S.l.]: John Wiley & Sons, 2019.

MEIER, R.; LAKE, I. **Professional Android**. [S.l.]: John Wiley & Sons, 2018.

MONTEIRO, J. B. **Google Android. Crie Aplicações Para Celulares e Tablets**. São Paulo: Casa do Código, 2013.

NAPOLI, M. L. **Beginning Flutter®**: A hands on guide to app development. Indianapolis, Indiana: John Wiley & Sons, 2020.

NIEMIEC, B. A. **Small Animal Dental, Oral and Maxillofacial Disease**: A colour handbook. London: Manson Publishing, 2011.

PANHALE, M. **Beginning Hybrid Mobile Application Development**. [S.l.]: Apress, 2016.

REITER, A. M.; GRACIS, M. **BSAVA Manual of Canine and Feline Dentistry and Oral Surgery**. Quedgeley: British Small Animal Veterinary Association, 2018.

ROSEMBERG, D.; STEPHENS, M.; COLLINS-COPE, M. **Agile Development with ICONIX Process**: People, process, and pragmatism. [S.l.]: Apress, 2005.

SILVA, C. M. R. A importância do odontologista veterinário e o conhecimento de conceitos básicos. **Portal Educação**, São Paulo, 2013. Disponível em: <https://www.portaleducacao.com.br/conteudo/artigos/veterinaria/a-importancia-do-odontologista-veterinario-e-o-conhecimento-de-conceitos-basicos/30804>. Acesso em: 15 abr. 2021.

SILVEIRA, G.; JARDIM, J. **Swift:Programe para iPhone e iPad**. [S.l.]: Casa do Código, 2014.

SIMAS, V. L. *et al.* **Desenvolvimento para dispositivos móveis**. Porto Alegre: Grupo A, 2019.

SMITH, R. Top 10 best cross platform app development frameworks in 2021. Janeiro 2021. Disponível em: <https://www.codenameone.com/blog/top-10-best-cross-platform-app-development-frameworks-in-2021.html>. Acesso em: 07 mai. 2021.

SOMMERVILLE, I. **Engenharia de software**. São Paulo: Pearson Universidades, 2019.

STATCOUNTER. Desktop vs mobile vs tablet market share worldwide. 2021. Disponível em: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>. Acesso em: 02 abr. 2021.

____. Operating system market share worldwide. 2021. Disponível em: <https://gs.statcounter.com/os-market-share>. Acesso em: 02 abr. 2021.

STEIL, R. **Ios - Programe Para Iphone E Ipad**. São Paulo: Casa do Código, 2012.

SWIFT. Linguagem de programação swift. 2021. Disponível em: <https://swift.org/>. Acesso em: 25 abr. 2021.

USP. Odontogramas do loc. 2014. Disponível em: <http://www.usp.br/locfmvz/link-download.html>. Acesso em: 20 abr. 2021.

VET DENTAL CHARTS. **Easy Vet Dental Charts**. Melbourne, VIC, 2021.

WIERUCH, R. **The Road to React with Firebase**: Your journey to master advanced react for business web applications. [S.l.]: Independently Published, 2019.

ANEXO A – MANUAL DE UTILIZAÇÃO DO APLICATIVO

Este anexo apresenta um manual de usuário para utilização do sistema.

DentisVET: Manual de utilização.

Antes de utilizar o aplicativo, observar os seguintes passos:

Passo 1: Certificar-se de que o smartphone possua conexão à internet.
Passo 2: Na clínica, definir o cargo de usuário das pessoas que utilizarão o sistema.

Cargos

Administrador: possui acesso total ao sistema, além de poder gerenciar os dados da clínica, criar e editar usuários.

Veterinário: cadastra clientes e animais, e também realiza atendimentos.

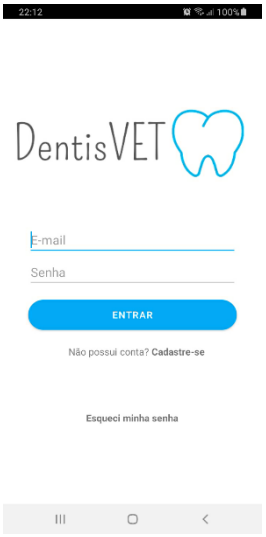
Secretário: realiza cadastro de clientes e animais.

Instruções para o usuário administrador

Criar login

O usuário administrador será quem irá criar o seu próprio cadastro para dar início a uma nova clínica, que também terá seus dados e usuários cadastrados por ele.

1. Acessar o aplicativo, e na tela de login, clicar em "Cadastre-se".





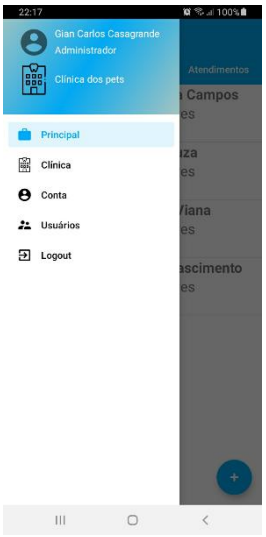
2. Na tela de dados cadastrais, preencher todos os campos exibidos, e clicar em "Cadastrar conta".

Após o cadastro ser concluído, o aplicativo irá prosseguir para a inicialização, validando o login.

Editar dados da clínica

Para completar as informações cadastrais da clínica, realizar os seguintes passos:

1. Acessar o menu deslizante à esquerda, e clicar na opção "Clínica".





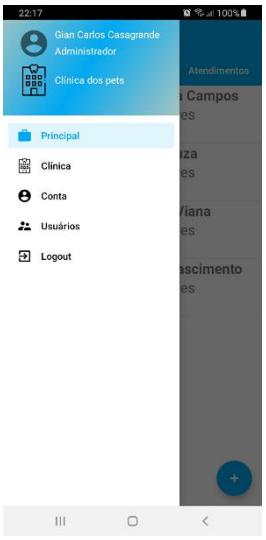
2. Preencher nos campos exibidos, os dados da clínica em que trabalha.

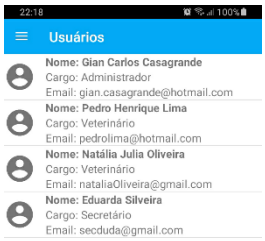
3. Concluído o preenchimento, clicar em "Salvar alterações".

Criar e editar dados e permissões de usuário

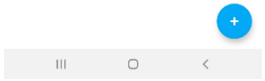
Criar novo usuário

1. Acessar o menu deslizante à esquerda, e clicar na opção "Usuários".





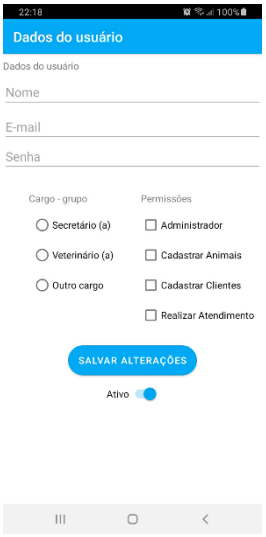
2. Na tela exibida, para criar um novo usuário, clicar no botão com ícone de “+” no canto inferior direito.



3. Nos campos exibidos, preencher o nome da pessoa, seu endereço de e-mail e senha de acesso provisória.

4. No grupo de cargos, indicar qual será o cargo da pessoa. Por padrão, o cargo de secretário possui a permissão de realizar atendimento desabilitada, estando disponível para quem tiver o cargo de veterinário. Há também a opção de outro cargo, onde nessa é possível definir um nome personalizado, e atribuir permissões personalizadas.

5. Após preenchido, clicar em "Salvar alterações".



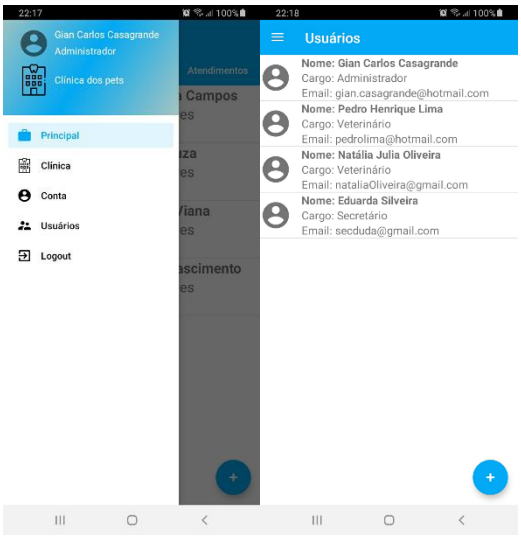
6. Fornecer este usuário com e-mail e senha para a pessoa a quem pertencerá.

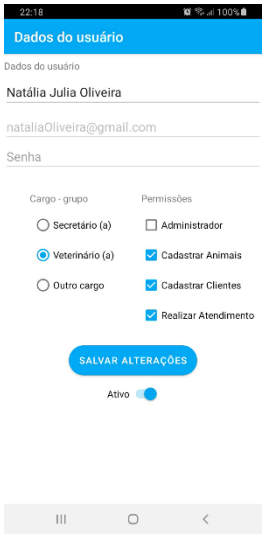
Editar usuário existente

1. Acessar o menu deslizante, e clicar na opção "Usuários".

2. Caso existam usuários inseridos na clínica, os mesmos serão listados na mesma tela.

3. Selecionar o usuário desejado. Será aberta uma nova tela com seus respectivos dados cadastrais.





4. Editar os dados desejados, como cargo, permissão e ativar ou desativar o usuário no sistema, representado por um botão do tipo interruptor.

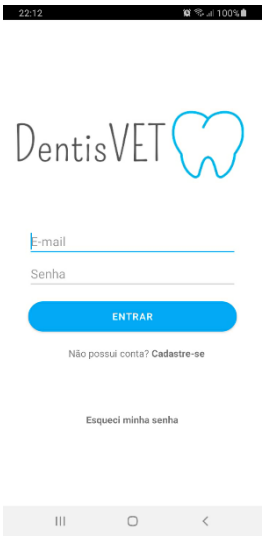
5. Clicar em "Salvar alterações".

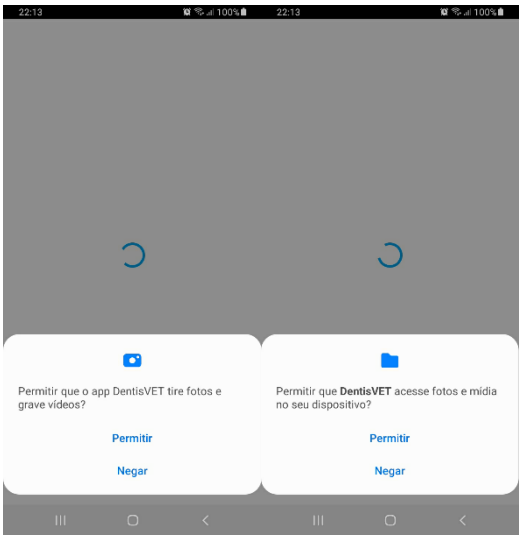
Instruções para os usuários secretário, veterinário e administrador

Fazer Login

1. Iniciar o aplicativo, e na tela de login, preencher os dados de e-mail e senha fornecido pelo administrador.

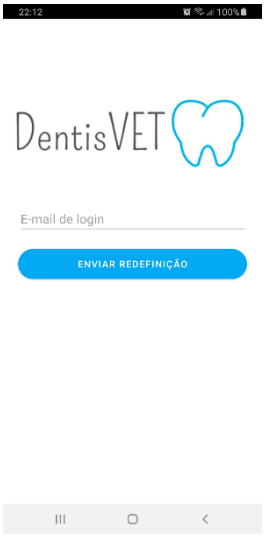
2. Após preencher os dados, clicar no botão “Entrar”.





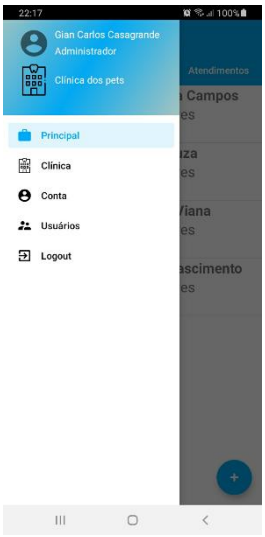
Ao validar o usuário, o aplicativo irá solicitar algumas permissões obrigatórias e essenciais para permitir algumas funcionalidades do aplicativo, como registrar e armazenar imagens de atendimento. Deve-se permitir ambas para o aplicativo ser utilizado.

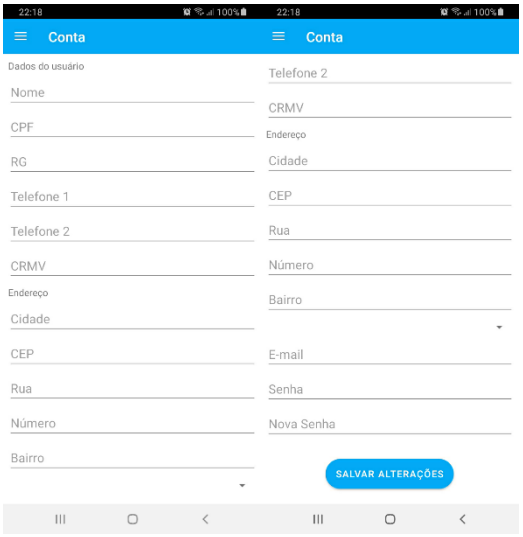
3. Caso a senha for esquecida, a mesma pode ser recuperada através da opção "Esqueci minha senha", da tela de login, preenchendo o e-mail de acesso, e clicando no botão "Enviar redefinição".



Alterar dados da conta

1. Acessar o menu deslizante à esquerda, e clicar no botão "Conta".



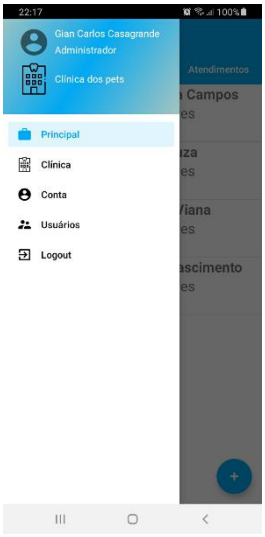


2. Na tela aberta, preencher os dados pessoais pertencentes à própria conta.

3. Clicar em "Salvar alterações".

Encerrar sessão de usuário

1. Acessar o menu deslizante à esquerda, e clicar no botão "Logout".





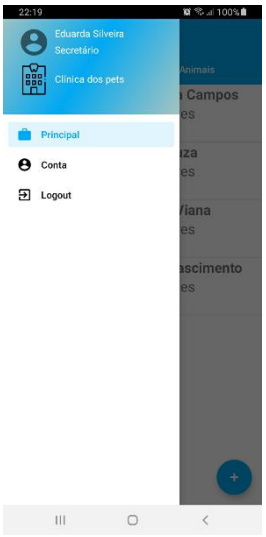
2. Será exibido um alerta de confirmação.

3. Clicar na opção desejada.

Instruções para o usuário Secretário

Cadastrar clientes

1. Acessar o menu deslizante à esquerda, e clicar no botão "Principal".

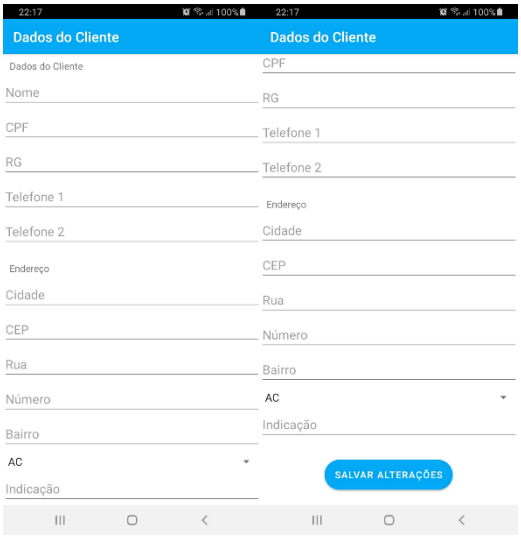




2. Na aba "Clientes", para cadastrar um novo cliente, clicar no botão com ícone de “+” no canto inferior direito.

3. Na tela exibida, preencher os campos com os dados cadastrais do cliente.

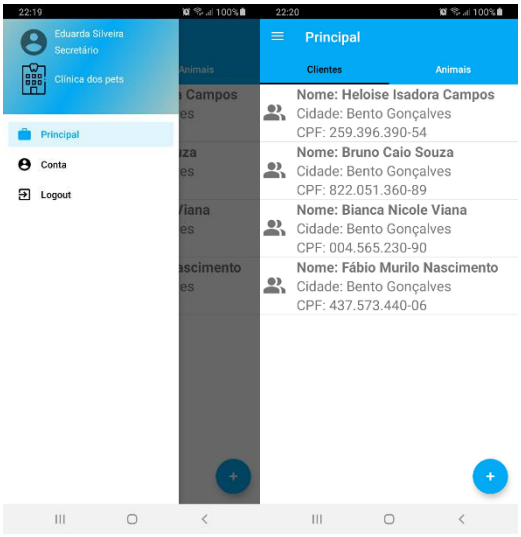
4. Ao terminar de preencher, clicar em "Salvar alterações".

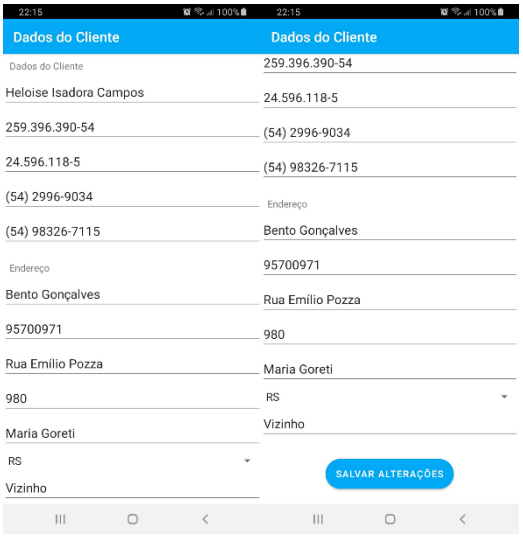


Editar cadastro de clientes

1. Acessar o menu deslizante à esquerda, e clicar no botão "Principal".

2. Na aba "Clientes", para editar o cadastro de um cliente, clicar no registro a ser editado.



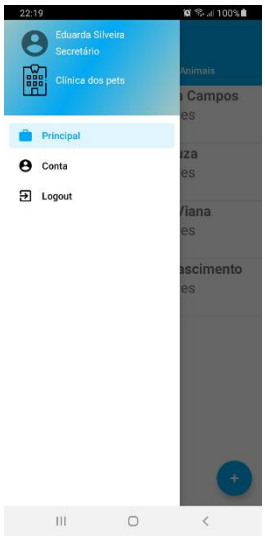


3. Na tela exibida, alterar os campos com os dados cadastrais do cliente a ser editado.

4. Ao terminar de preencher, clicar em "Salvar alterações".

Cadastrar animais

1. Certificar-se que o animal a ser cadastrado já possui cadastro de seu proprietário.

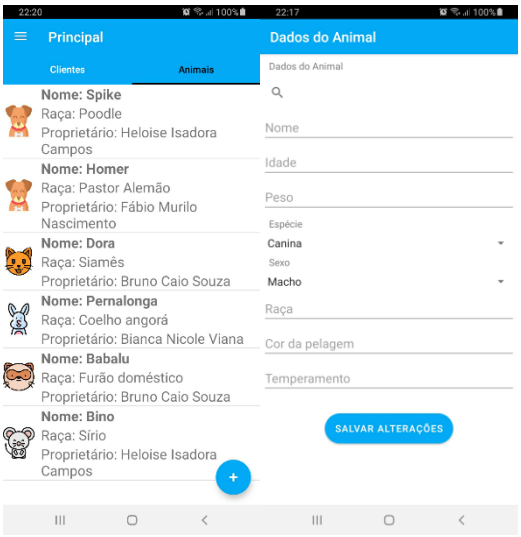


2. Acessar o menu deslizante à esquerda, e clicar no botão "Principal".

3. Na aba "Animais", para cadastrar um novo animal, clicar no botão com ícone de “+” no canto inferior direito.

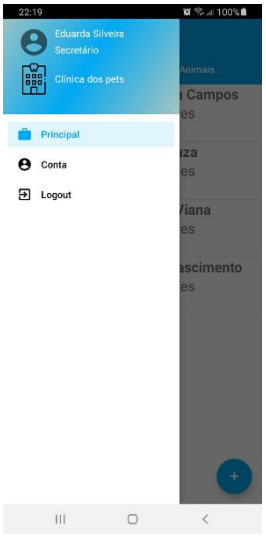
4. Na tela exibida, preencher os campos com os dados cadastrais do animal.

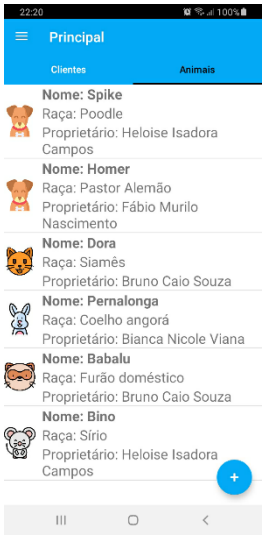
5. Ao terminar de preencher, clicar em "Salvar alterações".



Editar animais

1. Acessar o menu deslizante à esquerda, e clicar no botão "Principal".

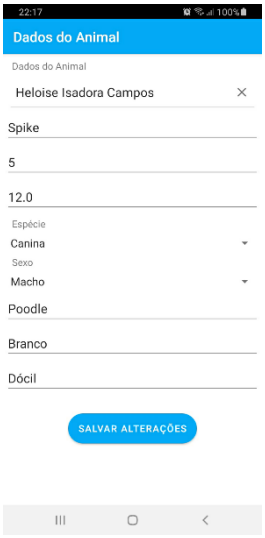




2. Na aba "Animais", para editar um animal, clicar em cima do registro do animal a ser editado.

3. Na tela exibida, editar os campos desejados com os dados cadastrais do animal.

4. Ao terminar de editar, clicar em "Salvar alterações".

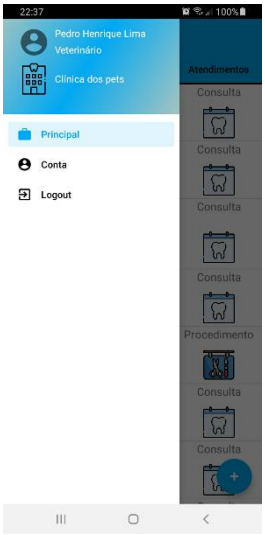


Instruções para o usuário Veterinário

Realizar atendimento

1. Certificar-se de que há um animal cadastrado.

2. Acessar o menu deslizante à esquerda, e clicar no botão "Principal".



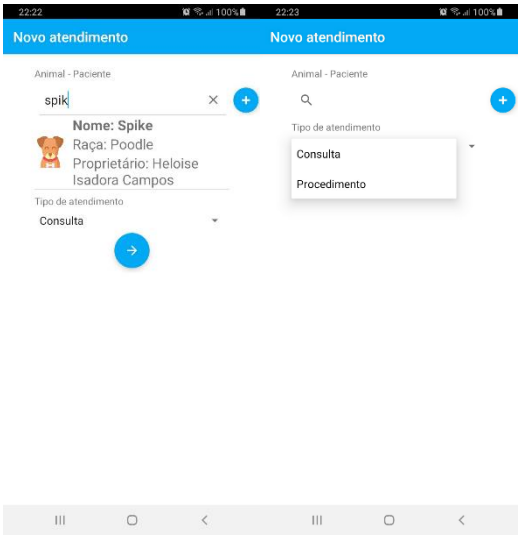


3. Acessar a aba "Atendimentos", e clicar no botão com ícone de “+”, no canto inferior direito, para iniciar um novo atendimento.

4. Na tela aberta, preencher o nome do animal no campo de pesquisa, e ao encontrá-lo, selecioná-lo na lista exibida.

5. No tipo de atendimento, selecionar se o mesmo é do tipo consulta ou procedimento.

6. Após preencher os dados, clicar no botão de prosseguir, representado por uma seta à direita.





7. Na tela seguinte, através do menu de navegação inferior, a mesma é segmentada em cinco itens: ficha, odontogramas, cavidades orais, crânios e imagens. Selecionar o item desejado.

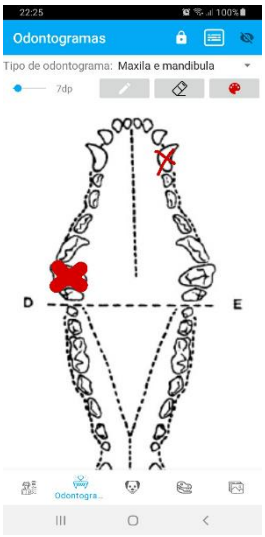
Ficha

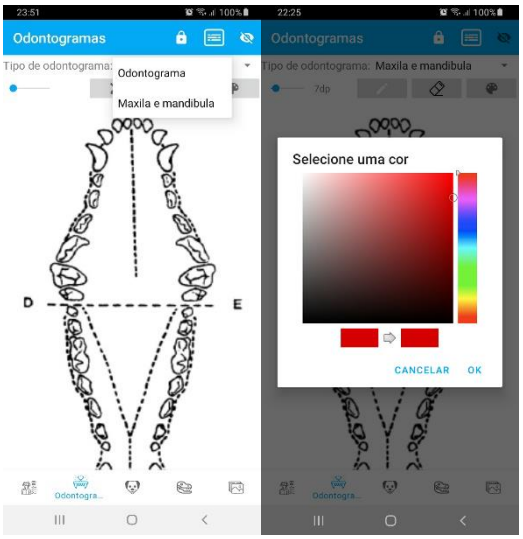
Na tela de ficha, preencher nos campos, os dados conforme as condutas realizadas.

Odontogramas, cavidades orais, e crânio

Acessar no menu inferior, a tela dedicada à cada tipo de representação. Em cada uma delas, existe uma caixa de ferramentas com ações de habilitar ou desabilitar desenho, definir tamanho do lápis, cor, ou apagar desenho.







1. Selecionar o tipo de representação desejada através da caixa de listagem exibida.

2. Após a imagem ser carregada, para desenhar, clicar no botão com um ícone representado por um lápis.

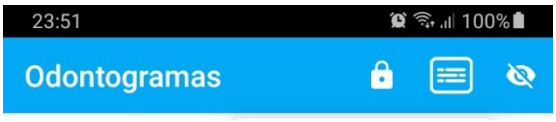
3. Para aumentar ou diminuir o tamanho do lápis, arrastar a barra situada no canto esquerdo até o tamanho desejado.

4. Para apagar o desenho, clicar no ícone de borracha.

5. Para selecionar a cor do lápis, clicar no ícone representado por uma paleta de cores, selecionar a cor desejada, e clicar em "Ok".

6. Acima das ferramentas de desenho, existem três botões, sendo eles:

- Um botão de cadeado para travar a rolagem da tela.
- Um botão de legenda, para exibir as siglas utilizadas nos odontogramas.
- Um botão representado por um olho, para aumentar o espaço de visualização de uma representação gráfica, ocultando os demais componentes.



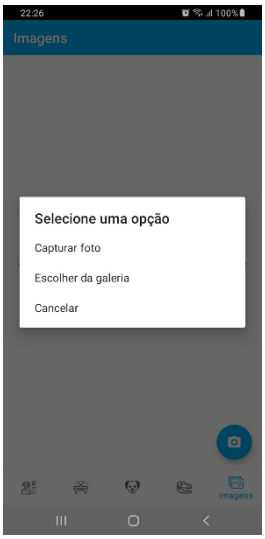
Galeria de imagens

Acessar no menu inferior, a tela dedicada às imagens do atendimento.

Para inserir uma nova imagem

1. Clicar no botão com ícone de câmera no canto inferior direito para inserir uma nova imagem.

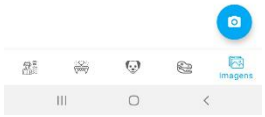
2. Escolher de qual local será escolhida a imagem.





3. Caso for escolhida a câmera, capturar a foto, e clicar em “Ok”.

4. Caso for escolhida da galeria, encontrar a foto, e selecioná-la.



Para visualizar uma imagem

Após uma foto ser inserida, a mesma será exibida em uma galeria, na mesma tela.

1. Selecionar a imagem a ser visualizada.

2. Será aberta uma nova tela, com a imagem em tamanho maior, com suporte a zoom.

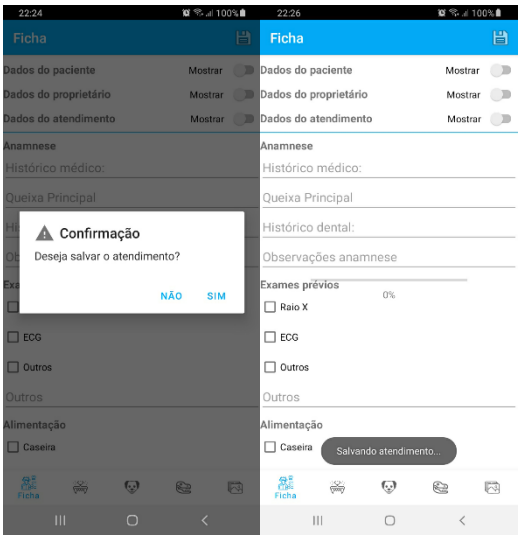
Salvar o atendimento

1. Após preencher todos os dados do atendimento, clicar no botão de ficha, no menu inferior.

2. Na barra superior, clicar no ícone representado por um disquete.

3. No alerta exibido, selecionar “Sim” para salvar o atendimento.

4. Aguardar até a conclusão do processo de salvar.



Exportar atendimento

1. Após haver atendimentos já registrados, acessar a aba de atendimentos.

2. Manter selecionado o registro do atendimento que deseja exportar, até surgir um ícone na barra superior.





3. Clicar no ícone exibido na barra superior, e aguardar até o atendimento finalizar sua exportação.

ANEXO B – DADOS DE PREENCHIMENTO DA FICHA DE EXAME CLÍNICO

Este anexo apresenta os dados presentes na ficha de exame clínico.

Quadro 17 – Grupos de dados da ficha de exame clínico

(continua)

Grupo	**Campo**	**Tipo de preenchimento**
1 Anamnese		
	Queixa principal	Texto
	Histórico dental	Texto
	Histórico médico	Texto
	Observações	Texto
2 Exames prévios		
	( ) Raio-x	Checkbox
	( ) ECG	Checkbox
	( ) Outros	Checkbox e Texto
	Exames Prévios	Texto
3 Alimentação		
	Alimentação	Texto
	( ) Caseira	Checkbox e Texto
	( ) Ração	Checkbox e Texto
	( ) Osso natural	Checkbox e Texto
	( ) Ossos/biscoitos artificiais	Checkbox e Texto
	( ) Higiene bucal	Checkbox e Texto
	( ) Vícios de roer	Checkbox e Texto
4 Exame clínico		
	( ) Assimetria da cabeça	Checkbox e Texto
	( ) Linfonodo	Checkbox e Texto
	( ) Md	Checkbox e Texto
	( ) Outros	Checkbox e Texto
	( ) Aumento de volume	Checkbox e Texto
	( ) Fístula	Checkbox e Texto

Quadro 17 – Grupos de dados da ficha de exame clínico

(conclusão)

Grupo	**Campo**	**Tipo de preenchimento**
	( ) Lábios	Checkbox e Texto
	( ) Saliva	Checkbox e Texto
	( ) Outros	Checkbox e Texto
	Mucosas	Texto
	Oclusão: ( ) Normal	Checkbox
	Oclusão: ( ) Outras	Checkbox e Texto
	Observações	Texto
	Crânio anexos	Texto
5 Diagnóstico clínico		
	Parecer clínico	Texto
	Tratamento recomendado	Texto
	Medicação pré-operatória	Texto
6 Exames solicitados		
	( ) ECG	Checkbox
	( ) Hemograma	Checkbox
	( ) Ht	Checkbox
	( ) Ureia/creatinina	Checkbox
	( ) F.hepática	Checkbox
	( ) Biópsia	Checkbox
	( ) Raio-x	Checkbox
	( ) Tomografia	Checkbox
	( ) Outros	Checkbox e Texto

Fonte: O Autor (2021).

ANEXO C – DADOS DE PREENCHIMENTO DA FICHA DE PROCEDIMENTO

Este anexo apresenta os atributos presentes na ficha de procedimento.

Quadro 18 – Grupos de dados da ficha de procedimento

(continua)

Grupo	**Campo**	**Tipo de preenchimento**
1 Ortodontia		
	( ) Moldagem	Checkbox e Texto
	( ) Modelo	Checkbox e Texto
	( ) Plano Inclinado dir.	Checkbox
	( ) Plano Inclinado indir.	Checkbox
	( ) Aparelho expansor	Checkbox
	( ) Aparelho retrator	Checkbox
	( ) Botão	Checkbox
	( ) Elástico	Checkbox
	Outros	Texto
2 Periodontia		
2.1 Raspagem	( ) Ultrassom	Checkbox
	( ) Manual	Checkbox
	( ) Aplainamento radicular	Checkbox
2.2 Polimento	( ) Pedra pomes	Checkbox
	( ) Flúor	Checkbox
	( ) Pasta profilática	Checkbox
2.3 Cirurgia	( ) Gengivectomia	Checkbox
	( ) Raspagem aberta / retalho	Checkbox
	( ) Esplintagem	Checkbox
	( ) Implante ósseo	Checkbox
	Outros	Texto
3 Exodontia		
3.1 Rotina	( ) Alavanca	Checkbox

Quadro 18 – Grupos de dados da ficha de procedimento

(continuação)

Grupo	**Campo**	**Tipo de preenchimento**
	( ) Odonto-secção	Checkbox
	( ) Alveolectomia / retalho	Checkbox
	( ) Alveoloplastia	Checkbox
3.2 Fratura de raiz	( ) Expectativa	Checkbox
	( ) Remoção/alavanca	Checkbox
	( ) Pulverização	Checkbox
	( ) Alveolectomia	Checkbox
	Outros	Texto
4 Dentística / Prótese		
4.1 Restauração	( ) Amálgama	Checkbox
	( ) Resina autopolim.	Checkbox
	( ) Resina foto	Checkbox
	( ) Ionômero de vidro	Checkbox
	( ) Primer	Checkbox
	( ) M.R.I.	Checkbox
	( ) Polimento	Checkbox
	( ) Pino dentinário	Checkbox
	( ) Radicular	Checkbox
	( ) Odontoplastia	Checkbox
	( ) Redução de coroa	Checkbox
	( ) Moldagem	Checkbox e Texto
	( ) Modelo	Checkbox e Texto
	( ) R.M.F.	Checkbox
	( ) Metalo-Cerâmica	Checkbox
	( ) Metalo-Plástica	Checkbox
	Observação	Texto
5 Neoplasia		
5.1 Biópsia	( ) Tecido mole	Checkbox
	( ) Ósseo	Checkbox
	( ) Incisional	Checkbox
	( ) Excisional	Checkbox

Quadro 18 – Grupos de dados da ficha de procedimento

(continuação)

Grupo	**Campo**	**Tipo de preenchimento**
	Biópsia observação	Texto
	Resultado	Texto
	Cirurgia	Texto
	( ) Ressecção tecido mole	Checkbox
	( ) Ressecção óssea total	Checkbox
	( ) Parcial	Checkbox
	Observação	Texto
6 Síntese	( ) Fio absorvível	Checkbox
	( ) Fio não absorvível	Checkbox
	Tipo	Texto
7 Osteossíntese	( ) Mandíbula	Checkbox
	( ) Maxila	Checkbox
	( ) Sutura tecido mole	Checkbox
	( ) Resina acrílica	Checkbox
	( ) Cerclagem óssea	Checkbox
	( ) Interdental	Checkbox
	( ) Fixação interna	Checkbox
	( ) Fixação externa	Checkbox
	( ) Bloqueio intermaxilar	Checkbox
	( ) Funil esparadrapado	Checkbox
	Outros:	Texto
8 Endodontia	( ) Pulpotomia	Checkbox
	( ) Pulpectomia	Checkbox
	( ) Penetração Desinfetante	Checkbox
8.1 Lima inicial nº / comprimento do canal:	RM __/__ mm	Texto
	RD __/__ mm	Texto
	RP __/__ mm	Texto

Quadro 18 – Grupos de dados da ficha de procedimento

(continuação)

Grupo	**Campo**	**Tipo de preenchimento**
8.2 Lima final nº / comprimento do canal:	RM __/__ mm	Texto
	RD __/__ mm	Texto
	RP __/__ mm	
	( ) Dakin	Checkbox
	( ) Endo-PTC	Checkbox
	( ) EDTA	Checkbox
	( ) Clorexidine	Checkbox
	( ) Cimento	Checkbox e Texto
	( ) Guta-percha nº__	Checkbox e Texto
	( ) Curativo Hidróxido Cálcio	Checkbox e Texto
	Proteção pulpar	Texto
	( ) Apicoectomia	Checkbox e Texto
	( ) Apexificação	Checkbox e texto
	Observação	Texto
9 Outras cirurgias	( ) Palato	Checkbox
	( ) Faringostomia	Checkbox
	( ) Lábio	Checkbox
	( ) Traqueostomia	Checkbox
	( ) Mucosa	Checkbox
	( ) Ostectomia	Checkbox
	( ) Sialoadenectomia	Checkbox
	Outras	Texto
	Observações	Texto
10 Bloqueios regionais		
	Anestésico	Texto
10.1 Infraorbitário ( )		Checkbox
	dir___ ml.	Texto
	esq___ml.	Texto

Quadro 18 – Grupos de dados da ficha de procedimento

(continuação)

Grupo	**Campo**	**Tipo de preenchimento**
10.2 Fossa pterigo ( )		Checkbox
	dir___ml.	Texto
	esq___ml.	Texto
10.3 Mandibular ( )		Checkbox
	dir___ ml.	Texto
	esq___ml.	Texto
10.4 Mentoniano ( )		Checkbox
	dir___ml.	Texto
	esq___ml.	Texto
11 Documentação		
	( ) Laser	Checkbox
	( ) Radiografia intraoral	Checkbox
	( ) Radiografia extraoral	Checkbox
	( ) Tomografia computadorizada	Checkbox
	( ) Fotografia	Checkbox
	( ) Filmagem	Checkbox
	( ) Histo #	Checkbox e Texto
	( ) Raio-X	Checkbox
12 Diagnóstico final		
	Diagnóstico Final	Texto
	Tratamento Realizado	Texto
	Medicação Prescrita	Texto
	Radiologia	Texto
	Protocolo Anestésico	Texto
13 Recomendações gerais		
	Recomendações Gerais	Texto
	Alimentação	Texto

Quadro 18 – Grupos de dados da ficha de procedimento

(continuação)

Grupo	**Campo**	**Tipo de preenchimento**
	( ) Alimentação pastosa __ dias	Checkbox e Texto
	( ) Higienização clorexidine __ dias	Checkbox e Texto
	( ) Colar elizabetano __ dias	Checkbox e Texto
	Observação	Texto
14 Tratamento realizado	Radiografia	Checkbox
	Desgaste	Checkbox
	Correção Oclusal	Checkbox
	Odontosecção	Checkbox
	Exodontia	Checkbox
	Periodontia	Checkbox
	Cirurgia Gengival	Checkbox
	Outros	Checkbox
15 Tratamento requisitado		
	Radiografia	Checkbox
	Desgaste	Checkbox
	Correção Oclusal	Checkbox
	Odontosecção	Checkbox
	Exodontia	Checkbox
	Periodontia	Checkbox
	Cirurgia Gengival	Checkbox
	Outros	Checkbox
	Manejo Dietético	Checkbox
16 Pessoas envolvidas		
	Cirurgião	Texto
	Auxiliar	Texto
	Anestesista	Texto
	Estagiários	Texto

Quadro 18 – Grupos de dados da ficha de procedimento

(conclusão)

Grupo	**Campo**	**Tipo de preenchimento**
	Projeto / Experimento:	Texto

Fonte: O Autor (2021).