

# **Documentação de um Produto de Software**

## **BICHO EXPRESS**

### **Nome dos Alunos:**

Eduardo Baptistella Gonçalves- 824147595  
Erick Domingues Soares- 82414486  
Gabriel Prieto Lima- 824142064  
Maria Eduarda Medeiro Porto- 824144948  
Matheus Alves Santana- 824144952  
Matheus Henrique da Costa e Silva – 82410661  
Wellington de Oliveira Sousa - 825240209

## RESUMO

O presente projeto visa o desenvolvimento de uma plataforma digital de mobilidade especializada para animais de estimação, composta por um aplicativo móvel, sendo uma parte voltada para os usuários (tutores de pets), responsáveis pela solicitação dos serviços de transporte, outro para os motoristas, encarregados de atender às demandas recebidas. Além disso, contará com um portal administrativo para gestão do serviço.

O sistema tem como propósito oferecer uma solução segura, prática e inovadora para o deslocamento de animais, integrando funcionalidades específicas como cadastro detalhado do pet, monitoramento em tempo real da corrida, comunicação direta entre usuário e motorista e métodos de pagamento digitais. Para os motoristas, a solução amplia o leque de oportunidades de trabalho, permitindo a construção de um portfólio profissional e o acompanhamento de ganhos. Para os tutores, a plataforma oferece comodidade, confiança e centralização na solicitação de serviços, garantindo o bem-estar do animal durante o transporte.

O desenvolvimento do projeto contemplou a definição de requisitos funcionais e não funcionais, elaboração de modelos de estrutura e arquitetura de software baseados em microsserviços, etapas de prototipagem, critérios de qualidade, além de fases de testes e planejamento de implantação. O sistema foi projetado para ser escalável, multiplataforma e em conformidade com legislações como a LGPD, priorizando a segurança da informação e a experiência do usuário. Este trabalho tornou-se viável graças à aplicação dos conceitos apresentados na unidade curricular de Modelos, métodos e técnicas de engenharia de software, aliados a pesquisas sobre soluções similares no mercado de transporte e mobilidade pet-friendly.

**Palavras-chave:** Software, Transporte Pet, Mobilidade, Plataforma Digital, Prestador-Cliente.

## LISTA DE FIGURAS

Figura 1- Nicho de Mercado Pet Friendly no Brasil. ....	10
Figura 2- Demonstrativo de Funcionamento Scrum .....	11
Figura 3- Diagrama de Casos de Uso do Usuário .....	12
Figura 4- Diagrama de Casos de Uso do Prestador de Serviços/Motorista .....	13
Figura 5- Diagrama de Casos de Uso do Gestor.....	14
Figura 6- Diagrama de Casos de Uso do Setor Administrativo.....	15
Figura 7- Diagrama de Funcionamento do Hardware.....	57
Figura 8- Diagrama de Classe Usuário/ Tutor de Pets. ....	61
Figura 9- Diagrama de Classe Motorista/Prestador de Serviços. ....	62
Figura 10- Diagrama de Classe Gestor e Setor Administrativo.....	63
Figura 11- Diagrama de Atividade Perfil do Pet.....	64
Figura 12- Diagrama da Atividade Definir Disponibilidade do Motorista .....	65
Figura 13- Diagrama de Atividade Definir Status da Corrida Atual. ....	66
Figura 14- Diagrama de Atividade Avaliar Motorista.....	67
Figura 15- Diagrama de Atividade Avaliar Usuário .....	68
Figura 16- Diagrama de Atividade Perfil do Motorista .....	69
Figura 17—Diagrama de Atividade Pagamento de Corridas. ....	70
Figura 18- Diagrama de Atividade Solicitar Nova Corrida .....	71
Figura 19- Diagrama de Atividade Cadastro de Novo Motorista.....	72
Figura 20- Diagrama de Atividade Perfil do Usuário e Cadastro de Novo Pet. ....	73
Figura 21- Diagrama de Atividade Cadastro de Novo Usuário .....	74
Figura 22- Diagrama de Atividade Histórico de Ganhos.....	75
Figura 23- Diagrama de Atividade Histórico de Corridas.....	76

Figura 24- Cobertura de Funcionalidades dos Módulos. ....	77
Figura 25- Desempenho do Software.....	88

## LISTA DE TABELAS

Tabela 1- Comparativo de Concorrentes Diretos e Indiretos.....	10
Tabela 2- Catálogo de Atores do Software.....	16
Tabela 3- Base de Preços Calculada Com Softwares Concorrentes.....	17
Tabela 4- Amazon EC2- t3 large .....	18
Tabela 5- Componentes Físicos do Software.....	57
Tabela 6- Funcionalidades de Qualidade do Software .....	85
Tabela 7- Confiabilidade do Software.....	87
Tabela 8- Critérios de usabilidade do Software .....	87
Tabela 9- Métricas de Eficiência de Desempenho.....	88
Tabela 10- Aspectos de Manutenibilidade do Software.....	89
Tabela 11- Portabilidade do Software .....	89
Tabela 12- Medidas de Segurança do Software.....	90
Tabela 13- Tabela de Prioridade do Software .....	93
Tabela 14- Requisitos por prioridade RF01 .....	94
Tabela 15- Fluxo de Requisitos RF01 cadastro.....	95
Tabela 16- Fluxo de Requisitos RF01 inserir e-mail.....	96
Tabela 17- Fluxo de Requisitos RF01 validação de senha.....	97
Tabela 18- Fluxo de Requisitos RF01 validação de CPF .....	98
Tabela 19- Fluxo de Requisitos RF01 validação de telefone.....	99
Tabela 20- Fluxo de Requisitos RF01 validação de restrição de idade .....	100
Tabela 21- Fluxo de Requisitos RF01 realizar login via Google .....	101
Tabela 22- Fluxo de Requisitos RF01 já possui cadastro.....	102
Tabela 23- Fluxo de Requisitos RF01 login não validado.....	103
Tabela 24- Fluxo de Requisitos RF01 recuperar senha .....	104
Tabela 25- Fluxo de requisitos RF01 MFA .....	105
Tabela 26- Fluxo de Requisitos RF01 excluir conta .....	106
Tabela 27- Prioridades RF02.....	107
Tabela 28- Fluxo de Requisitos RF02 criar perfil pet.....	108
Tabela 29- Fluxo de Requisitos RF02 salvar histórico de vacinas.....	109
Tabela 30- Fluxo de Requisitos RF02 Campos obrigatórios vazios .....	110
Tabela 31- Fluxo de Requisitos RF02 Editar perfil pet .....	111
Tabela 32- Fluxo de Requisitos RF02 visualizar perfil motorista .....	112
Tabela 33- Fluxo de Requisitos RF02 inserir nova imagem .....	113
Tabela 34- Fluxo de Requisitos RF02 inserir descrição médica .....	113
Tabela 35- Prioridades RF03.....	114
Tabela 36- Fluxo de requisitos RF03 solicitação de corrida imediata .....	115
Tabela 37- Fluxo de requisitos RF03 solicitação de corrida agendada.....	116
Tabela 38- Fluxo de requisitos RF03 cálculo de corrida .....	117
Tabela 39- Fluxo de requisitos RF03 motoristas não disponíveis.....	117
Tabela 40- Fluxo de requisitos RF03 sistema redireciona motorista por timeout.....	118
Tabela 41- Fluxo de Requisito RF03 solicitação de opções especiais .....	118

Tabela 42- Fluxo de Requisito RF03 solicitação de cancelamento por usuário .....	119
Tabela 43- Fluxo de Requisito RF03 solicitação de cancelamento por motorista.....	120
Tabela 44- Fluxo de Requisito RF03 solicitação de corrida com pagamento pré- autorizado.....	120
Tabela 45- Fluxo de Requisito RF03 falha no sistema de API de mapas .....	121
Tabela 46- Fluxo de Requisito RF03 surgimento de pricing .....	121
Tabela 47- Fluxo de Requisito RF03 solicitação de pet com necessidades .....	122
especiais	
Tabela 48- Fluxo de Requisito RF03 geração de históricos de corridas.....	122
Tabela 49- Fluxo de Requisito RF03 agendamento de corrida e cancelamento.....	123
posterior	
Tabela 50- Prioridades RF04.....	124
Tabela 51- Fluxo de Requisito RF04 busca motorista mais próximo.....	125
Tabela 52- Fluxo de Requisito RF04 filtragem de motorista por necessidades especiais .....	126
Tabela 53- Fluxo de Requisito RF04 Ignorar telemetria desatualizada .....	127
Tabela 54- Fluxo de Requisito RF04 Tie-Braeker entre dois motoristas.....	127
Tabela 55- Prioridades RF05.....	128
Tabela 56- Fluxo de Requisito RF05 atualização de posição em tempo real.....	129
Tabela 57- Fluxo de Requisito RF05 atualização de posição ≤5s.....	130
Tabela 58- Fluxo de Requisito RF05 indicar ETA.....	130
Tabela 59- Fluxo de Requisito RF05 lidar com telemetria desatualizada .....	131
Tabela 60- Fluxo de Requisito RF05 mostrar últimos eventos .....	131
Tabela 61- Fluxo de Requisito RF05 privacidade .....	132
Tabela 62- Fluxo de Requisito RF05 histórico de rastreamento .....	132
Tabela 63- Fluxo de Requisito RF05 Modo offline.....	133
Tabela 64- Prioridades RF06.....	134
Tabela 65- Fluxo de Requisito RF06 pagamento via Pix.....	136
Tabela 66- Fluxo de Requisito RF06 pagamento via cartão.....	137
Tabela 67- Fluxo de Requisito RF06 pagamento via cartão.....	137
Tabela 68- Fluxo de Requisito RF06 cartão expirado/recusado .....	138
Tabela 69- Fluxo de Requisito RF06 falha na autorização .....	138
Tabela 70- Fluxo de Requisito RF06 reembolso .....	139
Tabela 71- Fluxo de Requisito RF06 emissão de recibo .....	139
Tabela 72- Fluxo de Requisito RF06 segurança .....	140
Tabela 73- Fluxo de Requisito RF06 estorno .....	140
Tabela 74- Fluxo de Requisito RF06 pagamento offline.....	141
Tabela 75- Prioridades RF07.....	142
Tabela 76- Fluxo de Requisito RF07 cancelamento.....	143
Tabela 77- Fluxo de Requisito RF07 cancelamento pelo .....	144
motorista antes da coleta	
Tabela 78- Fluxo de Requisito RF07 cancelamento durante o trajeto .....	145
Tabela 79- Fluxo de Requisito RF07 aplicar taxa de cancelamento.....	146
Tabela 80- Prioridades RF08.....	147
Tabela 81- Fluxo de Requisito RF08 exibir tela de avaliação .....	148
Tabela 82- Fluxo de Requisito RF08 submeter à avaliação .....	148

Tabela 83- Fluxo de Requisito RF08 recalcular média .....	149
Tabela 84- Fluxo de Requisito RF08 avaliação duplicada .....	149
Tabela 85- Fluxo de Requisito RF08 geração de alertas.....	150
para notas baixas	
Tabela 86- Prioridades RF09.....	151
Tabela 87- Fluxo de Requisito RF09 exibir histórico de corridas .....	152
Tabela 88- Fluxo de Requisito RF09 filtrar históricos por .....	153
período status	
Tabela 89- Fluxo de Requisito RF09 exibir detalhes do pagamento .....	153
Tabela 90- Fluxo de Requisito RF09 validar integridade dos dados.....	154
Tabela 91- Prioridades do RF10.....	155
Tabela 92- Fluxo de Requisito RF10 enviar mensagem.....	156
via chat suporte	
Tabela 93- Fluxo de Requisito RF10 receber resposta aos do suporte em tempo real .....	157
Tabela 94- Fluxo de Requisito RF10 anexar imagem .....	157
Tabela 95- Fluxo de Requisito RF10 registrar histórico .....	158
Tabela 96- Fluxo de Requisito RF10 reabrir ticket .....	158
Tabela 97- Prioridades RF11.....	159
Tabela 98- Fluxo de Requisito RF11 cadastro básico de motorista.....	160
Tabela 99- Fluxo de Requisito RF11 upload de .....	160
documentos obrigatórios	
Tabela 100- Fluxo de Requisito RF11 cadastro incompleto .....	161
Tabela 101- Prioridades RF12.....	162
Tabela 102- Fluxo de Requisito RF12 revisar cadastro pendente .....	163
Tabela 103- Fluxo de Requisito RF12 aprovar ou rejeitar motorista.....	163
Tabela 104- Fluxo de Requisito RF12 solicitar nova documentação .....	164
Tabela 105- Prioridades RF13.....	165
Tabela 106- Fluxo de Requisito RF13 alternar status.....	166
Tabela 107- Fluxo de Requisito RF13 atualização .....	166
automática de status	
Tabela 108- Prioridades RF14.....	167
Tabela 109- Fluxo de Requisito RF14 receber solicitação de corrida.....	168
Tabela 110- Fluxo de Requisito RF14 aceitar solicitação .....	169
dentro do tempo	
Tabela 111- Fluxo de Requisito RF14 recusar solicitação.....	169
Tabela 112- Prioridades RF15.....	170
Tabela 113- Fluxo de Requisito RF15 exibir informações do pet .....	171
Tabela 114- Fluxo de Requisito RF15 ver detalhes.....	171
adicionais e cuidados	
Tabela 115- Prioridades RF16.....	172
Tabela 116- Fluxo de Requisito RF16 atualizar o.....	173
status manualmente	
Tabela 117- Prioridades RF17.....	174
Tabela 118- Fluxo de Requisito RF17 exibir histórico completo .....	175

Tabela 119- Prioridades RF18.....	176
Tabela 120- Fluxo de Requisito RF18 receber .....	177
notificação de corrida agendada	
Tabela 121- Prioridades RF19.....	178
Tabela 122- Fluxo de Requisito RF19 registraram ocorrência.....	179
Tabela 123- Prioridades RF20.....	180
Tabela 124- Fluxo de Requisito RF20 cadastro de motorista.....	181
Tabela 125- Prioridades RF21.....	182
Tabela 126- Fluxo de Requisito RF21 listar .....	183
corridas em andamento no Painel	
Tabela 127- Fluxo de Requisito RF21 filtrar .....	183
por área status e prioridade	
Tabela 128- Fluxo de Requisito RF21 atualização em tempo real .....	184
Tabela 129- Fluxo de Requisito RF21 abrir detalhes da corrida.....	184
Tabela 130- Fluxo de Requisito RF21 alertas automáticos .....	185
Tabela 131- Prioridades RF22.....	186
Tabela 132- Fluxo de Requisito RF22 registrar disputa.....	187
Tabela 133- Fluxo de Requisito RF22 coletar evidências.....	187
Tabela 134- Fluxo de Requisito RF22 comunicação moderada .....	187
Tabela 135- Prioridades RF23.....	188
Tabela 136- Fluxo de Requisito RF23 relatórios.....	189
de corridas por período	
Tabela 137- Fluxo de Requisito RF23 relatório financeiro .....	189
Tabela 138- Prioridades RF24.....	190
Tabela 139- Fluxo de Requisito RF24 iniciar chat interno .....	191
Tabela 140- Fluxo de Requisito RF24 anexo de Imagens.....	191
Tabela 141- Prioridades RF25.....	192
Tabela 142- Fluxo de Requisito RF25 nota média e total.....	193
Tabela 143- Prioridades RF26.....	194
Tabela 144- Prioridades RF27.....	195
Tabela 145- Prioridades RF28.....	196
Tabela 146- Fluxo de Requisito RF28 configurar regras .....	197
Tabela 147- Fluxo de Requisito RF28 aplicar multiplicadores .....	197
Tabela 148- Prioridades RF28.....	198
Tabela 149- Fluxo de Requisito RF29 criar cupom .....	199
Tabela 150- Prioridades RF30.....	200
Tabela 151- Prioridades RF31.....	201
Tabela 152- Fluxo de Requisito RF31 mapa de calor.....	202
Tabela 153- Fluxo de Requisito RF31 revisão temporal .....	202



## ÍNDICE DETALHADO

<b>Introdução .....</b>	<b>10</b>
Tema .....	11
Objetivos a serem alcançados .....	11
Escopo Principal.....	12
Modelos e Objetivo de Negócio .....	12
Serviço e Mercado do Projeto .....	13
Concorrência de Mercado .....	13
 <b>Definição do Modelo de Processo .....</b>	 <b>15</b>
 <b>Desenvolvimento do Software.....</b>	 <b>16</b>
Diagrama de Casos de Uso- Funcionamento do Aplicativo do Usuário .....	16
Diagrama de Casos de Uso- Funcionamento do Aplicativo do Prestador de Serviço.....	17
Diagrama de Casos de Uso- Funcionamento do Aplicativo do Gestor .....	18
Diagrama de Casos de Uso- Funcionamento do Aplicativo do Setor Administrativo.....	19
Catálogo de Atores.....	20
Base de Preços .....	21
Requisitos Funcionais.....	22
Requisitos do Usuário .....	22
Requisitos do Prestador de Serviços .....	22
Requisitos do Setor Administrativo.....	23
Requisitos do Gestor .....	23
Requisitos Não Funcionais .....	24
Requisitos do Externos.....	24
Interoperabilidade .....	24
Éticos.....	24
Legais .....	24
Requisitos Organizacionais .....	24
Entrega.....	24
Implementação.....	25
Padrões .....	25
Requisitos de Facilidade de Uso .....	25
Requisitos de Confiabilidade.....	26
Requisitos de Portabilidade.....	26
Requisitos de Eficiência.....	26
Requisitos de Espaço .....	26
Fluxo de Requisitos do Software.....	27
Aplicativo 1- Usuário/ Tutor de Pets.....	27
RF01- Cadastro/ Login .....	27
RF02- Cadastro de pets .....	30
RF03- Solicitar Transporte.....	31
RF04- Exibir Motorista.....	32
RF05- Acompanhar Corrida em tempo Real. ....	33
RF06 – Pagamento.....	34
RF07- Cancelamento de Corridas.....	35
RF08- Avaliação de Motoristas. ....	36
RF09- Histórico de Corridas e Pagamentos .....	37
RF010- Suporte via Chat .....	38

Aplicativo 1- Requisitos do Prestador de Serviço/Motorista.....	39
RF011- Cadastro de Motorista.....	39
RF012- Validação pelo Gestor .....	40
RF013- Indicar Status de Disponibilidade .....	41
RF014- Receber Solicitação de Corridas. ....	42
RF015- Acesso a Informações do Pet. ....	43
RF016- Atualizar Status da Corrida.....	44
RF017- Histórico de Corridas e Ganhos .....	45
RF018- Notificação sobre Corridas Agendadas. ....	46
RF019- Registrar Ocorrências durante a Corrida.....	46
Aplicativo 1- Requisitos do Setor Administrativo.....	48
RF020- Gerenciar Motoristas e Usuários.....	48
RF021- Monitorar Corridas em Andamento .....	49
RF022- Intervir em Disputas entre Usuários e Motoristas. ....	49
RF023- Emitir Relatórios Detalhados.....	50
RF024- Enviar Mensagens de Suporte via Chat Interno .....	51
RF025- Acompanhar Estatísticas de Avaliações.....	52
Aplicativo 1- Requisitos do Gestor.....	53
RF026- Validar Motoristas. ....	53
RF027- Suspende/Bloquear Motoristas e Usuários .....	53
RF028- Configurar Tarifas Dinâmicas.....	54
RF029- Criar Campanhas. ....	55
RF030- Acessar Relatórios Estratégicos .....	56
RF031- Acessar Dashboards.....	57
<b>Projeto.....</b>	<b>58</b>
Arquitetura Lógica.....	58
Arquitetura do Software.....	58
Layers do Sistema.....	58
Justificativa da Arquitetura. ....	59
Tier .....	60
Armazenamento Cloud Computer.....	60
Linguagem de Programação.....	61
Arquitetura Física.....	61
Configurações do Hardware .....	61
Diagrama da Funcionamento Do Software. ....	62
Configurações da Rede. ....	63
Topologia Lógica.....	63
Inventário de Componentes e Configurações. ....	64
Diagramas de Classe e Atividade do Software.....	66
Diagrama de Classe Usuário/Tutor de Pets.....	66
Diagrama de Classe Prestador de Serviços.....	67
Diagrama de Classe Setor Administrativo e Gestores. ....	68
Diagrama de Atividade Perfil do Pet.....	69
Diagrama de Atividade Disponibilidade do Motorista .....	70
Diagrama de Atividade Definir Status da Corrida .....	71
Diagrama de Atividade Avaliar Motorista.....	72
Diagrama de Atividade Avaliar Usuário.....	73
Diagrama de Atividade Perfil do Motorista .....	74
Diagrama de Atividade Pagamento.....	75
Diagrama de Atividade Solicitar Nova Corrida .....	76

Diagrama de Atividade Cadastro de Motorista .....	77
Diagrama de Atividade Perfil de Usuário e Cadastro de Pet.....	78
Diagrama de Atividade Cadastro de Novo Usuário .....	79
Diagrama de Atividade Histórico de Ganhos.....	80
Diagrama de Atividade Histórico de Corridas .....	81
<b>Protótipo de Interface .....</b>	<b>82</b>
Fase de Levantamento.....	82
Detalhamento Aplicativo Usuários.....	85
Detalhamento Aplicativo Motoristas.....	87
Detalhamento Aplicativo Gestores e Setor Administrativo .....	89
Protótipo da Interface.....	91
<b>Critérios de Qualidade de Software .....</b>	<b>93</b>
Funcionalidades .....	93
Confiabilidade.....	94
Usabilidade.....	95
Eficiência de Desempenho.....	95
Manutenibilidade.....	96
Portabilidade.....	98
Segurança.....	98
Compatibilidade e Integração.....	98
Conclusão Critérios de Segurança.....	99
<b>Testes de Requisitos Funcionais .....</b>	<b>100</b>
Tabela de Prioridades.....	100
Plano de Testes RF01 Cadastro/Login.....	101
Plano de Testes RF02 Cadastro e Perfil de Pets.....	112
Plano de Testes RF03 Solicitação de Nova Corrida.....	118
Plano de Testes RF04 Exibir Motorista Mais Próximo .....	128
Plano de Testes RF05 Acompanhar o Transporte em tempo Real .....	132
Plano de Testes RF06 Pagamento .....	138
Plano de Testes RF07 Cancelamento de Corridas.....	146
Plano de Testes RF08 Avaliação de Motoristas .....	151
Plano de Testes RF09 Histórico de Corridas e Pagamento .....	157
Plano de Testes RF10 Suporte Via Chat .....	161
Plano de Testes RF11 Cadastro de Motorista.....	165
Plano de Testes RF12 Validação por Gestor.....	167
Plano de Testes RF13 Indicar Status.....	170
Plano de Testes RF14 Receber Solicitação de Corrida.....	172
Plano de Testes RF15 Acesso a Informações de Pets.....	175
Plano de Testes RF16 Atualizar Status de Corrida Atual .....	177
Plano de Testes RF17 Histórico de Corridas e Ganhos .....	179
Plano de Testes RF18 Notificação sobre Corridas Agendadas.....	181
Plano de Testes RF19 Registrar Ocorrências durante Corridas .....	181
Plano de Testes RF20 Gerenciar Motoristas e Usuários.....	184
Plano de Testes RF21 Monitorar Corridas Em Andamento .....	186
Plano de Testes RF22 Intervir em Disputas .....	190
Plano de Testes RF23 Emitir Relatórios.....	191
Plano de Testes RF24 Enviar Mensagens de Suporte.....	193
Plano de Testes RF25 Acompanhar Estatísticas.....	195

Plano de Testes RF26 Validar Motoristas .....	197
Plano de Testes RF27 Suspende/Bloquear Motoristas .....	198
Plano de Testes RF28 Configurar Tarifas .....	198
Plano de Testes RF29 Criar Campanhas .....	200
Plano de Testes RF30 Acessar Relatórios Estratégicos .....	201
Plano de Testes RF31 Acessar Dashboards.....	202
<b>Conclusão .....</b>	<b>211</b>
<b>Referências Bibliográficas .....</b>	<b>213</b>

## 1. Introdução

Com o crescimento do uso da tecnologia e a expansão de soluções digitais voltadas para mobilidade, surge a necessidade de desenvolver plataformas que atendam nichos específicos da sociedade, oferecendo praticidade, segurança e inovação. Dentro desse contexto, os serviços pet-friendly vêm ganhando destaque, acompanhando o aumento do número de animais de estimação e a valorização do bem-estar animal nas grandes cidades.

O presente trabalho propõe o desenvolvimento de uma plataforma digital de transporte especializado para animais de estimação, denominada Bicho Express. O sistema será composto por um aplicativo móvel – uma parte voltada para os tutores de pets (usuários) e outra para os motoristas responsáveis pelo transporte – além de um portal administrativo para a gestão do serviço.

A solução contempla funcionalidades como cadastro detalhado do animal, solicitação e agendamento de corridas, cálculo automático de rotas, monitoramento em tempo real, comunicação direta entre usuário e motorista, avaliações pós-serviço e múltiplas formas de pagamento digitais. Com isso, pretende-se garantir a segurança, o conforto e a transparência no deslocamento dos animais.

Do ponto de vista dos motoristas, o sistema oferece novas oportunidades de trabalho, possibilitando ganhos extras, organização de corridas e construção de portfólio profissional. Para os tutores, representa uma alternativa confiável e prática para o transporte de seus pets, centralizando todas as etapas em uma única plataforma digital.

O desenvolvimento do projeto seguirá metodologias ágeis (Scrum), visando entregas incrementais e escalabilidade da solução. Além disso, será considerada a conformidade com legislações como a LGPD e normas de segurança da informação, assegurando a proteção dos dados e a confiabilidade da plataforma.

### **1.1. Tema**

O projeto visa o desenvolvimento de um sistema de transporte especializado para animais de estimação, que funcione como uma plataforma digital de mobilidade pet-friendly. A solução será composta por um aplicativo móvel para usuários (donos de pets) e motoristas, além de um portal administrativo para gestão do serviço. O sistema deverá contemplar funcionalidades específicas para o transporte seguro e confortável dos animais, como cadastro detalhado do pet, requisitos de segurança, monitoramento em tempo real e comunicação direta entre usuário e motorista.

A escolha de São Paulo como sede é estratégica, pois a cidade possui uma grande concentração de animais de estimação e uma demanda crescente por serviços especializados, além de desafios logísticos típicos de grandes metrópoles, como trânsito intenso e diversidade de perfis de usuários.

A arquitetura do sistema será baseada em microserviços para garantir escalabilidade e facilidade de manutenção, utilizando APIs RESTful para comunicação entre front-end e back-end. A interface será desenvolvida com foco em usabilidade e acessibilidade, seguindo padrões modernos de design responsivo

### **1.2. Objetivos a serem alcançados**

O principal objetivo do projeto é criar uma plataforma digital intuitiva e confiável para o transporte de animais de estimação, que atenda às necessidades dos donos e garanta o bem-estar dos pets durante o trajeto. Especificamente, pretende-se:

- Desenvolver um aplicativo móvel multiplataforma (iOS e Android) e um portal web para cadastro, solicitação e gerenciamento de corridas.
- Implementar funcionalidades de geolocalização em tempo real, cálculo automático de rotas e estimativa de preços.
- Garantir a segurança dos animais com requisitos técnicos, como motoristas treinados, veículos adaptados e monitoramento via câmera.
- Automatizar processos de agendamento, pagamento e avaliação dos serviços.
- Oferecer suporte ao cliente integrado via chat e notificações push.
- Atender ao público-alvo composto por donos de animais de estimação residentes em São Paulo e região metropolitana, que buscam uma alternativa prática e segura para transportar seus pets.
- Promover inovação no mercado pet, integrando tecnologias de IoT para monitoramento do bem-estar animal durante o transporte.

### 1.3. Escopo principal

A proposta principal do projeto é desenvolver um sistema completo de transporte de animais que englobe as seguintes ações e implementações:

- Cadastro e perfil do usuário e do pet: inclusão de informações detalhadas sobre o animal (raça, porte, necessidades especiais) e do dono.
- Solicitação e agendamento de corridas: interface para o usuário solicitar transporte imediato ou agendado, com opções de tipos de veículos adaptados.
- Geolocalização e roteirização: integração com APIs de mapas para rastreamento em tempo real e otimização de rotas.
- Sistema de pagamento integrado: suporte a múltiplas formas de pagamento digitais, com segurança e transparência.
- Avaliação e feedback: sistema de avaliação para motoristas e usuários, promovendo a qualidade do serviço.
- Painel administrativo: dashboard para gestão de motoristas, corridas, faturamento e suporte.
- Notificações e comunicação: envio de alertas via SMS, e-mail e push para status da corrida e comunicação direta entre partes.
- Segurança e conformidade: implementação de protocolos para garantir a segurança dos animais e conformidade com legislações locais.

O desenvolvimento seguirá metodologias ágeis, com entregas incrementais e foco na experiência do usuário, garantindo que o sistema seja escalável e adaptável às necessidades futuras.

### 1.4. Modelo e Objetivo de Negócio

O **Bicho Express** visa ser uma plataforma digital de mobilidade pet-friendly, composta por dois aplicativos – um voltado para os tutores de animais de estimação (usuários) e outro para os motoristas autorizados – além de um portal administrativo para gestão dos serviços. A solução promove a intermediação entre cliente e prestador, garantindo transporte seguro, ágil e confortável para os animais, a partir de um cadastro detalhado e solicitações diretas via aplicativo.

Temos como objetivo facilitar a busca por um transporte de qualidade para os tutores, gerar novas oportunidades de renda para motoristas qualificados e assegurar o bem-estar animal em cada viagem, entregando autonomia e confiança para ambos os lados.

### **1.5. Serviço e Mercado do Projeto**

Nosso foco é atingir tutores de animais de estimação por meio de uma plataforma digital intuitiva, prática e segura, oferecendo um serviço especializado de transporte pet-friendly. A solução garante o bem-estar dos animais durante o trajeto, proporcionando comodidade para os clientes e uma experiência diferenciada em comparação a transportes comuns.

Em relação ao mercado, observa-se um crescimento constante do setor pet no Brasil, especialmente em serviços voltados para mobilidade e cuidados com animais. Muitos tutores enfrentam dificuldades em deslocar seus pets de forma segura, seja para consultas veterinárias, pet shops ou viagens, e carecem de alternativas adequadas. Ao mesmo tempo, existem motoristas interessados em novas oportunidades de renda, mas que ainda não encontram plataformas que conectem de forma estruturada esse nicho de mercado.

Com essas informações, o Bicho Express direciona seu foco na demanda crescente por transporte especializado, oferecendo um leque maior de possibilidades de atendimento e consolidando-se como uma plataforma inovadora, confiável e adaptada às necessidades do setor pet.

### **1.6. Concorrência de Mercado**

O mercado pet no Brasil é um dos que mais cresce no mundo, movimentando mais de R\$ 60 bilhões anuais e ocupando a terceira posição mundial em faturamento, ficando atrás apenas dos Estados Unidos e da China. No entanto, a área de mobilidade pet ainda é pouco explorada, com poucas soluções que ofereçam transporte seguro, confortável e dedicado exclusivamente a animais de estimação.

Atualmente, os tutores que precisam deslocar seus pets contam com três principais alternativas:

1. Aplicativos de transporte convencionais (Uber, 99, InDrive): permitem o transporte de pets, mas não possuem estrutura especializada, gerando insegurança para os animais e desconforto para motoristas que não são preparados.
2. Táxis pet e empresas locais de transporte: presentes em grandes cidades, mas com alcance limitado, alto custo e pouca praticidade para agendamento imediato.
3. Serviços informais (redes sociais, grupos de WhatsApp): embora comuns, carecem de profissionalização, segurança e padronização nos preços.



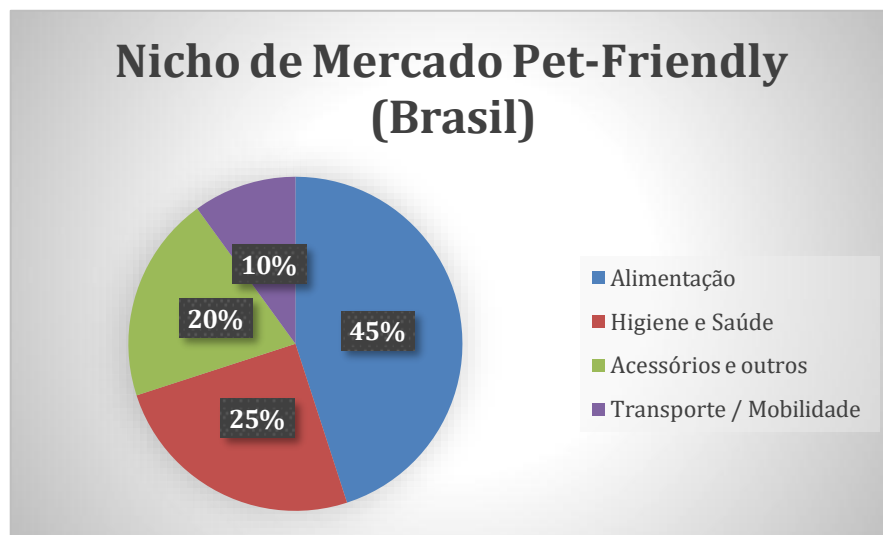
4. Diante disso, o Bicho Express se diferencia por unir a praticidade de um aplicativo de mobilidade urbana com a especialização em transporte animal, atendendo a uma dor real dos tutores e explorando uma oportunidade de mercado em expansão.

TABELA 1- Comparativo de Concorrentes Diretos e Indiretos

Concorrente / Solução	Cobertura	Foco no Pet	Preço Médio	Segurança	Diferencial
Uber / 99 (Pet friendly)	Nacional	Baixo	R\$ 20-60	Limitada	Alta capilaridade
Táxis Pet locais	Regional	Alto	R\$ 60-150	Alto	Transporte dedicado
Serviços informais (WhatsApp)	Regional	Médio	Negociável	Baixo	Flexibilidade
<b>Bicho Express (Proposto)</b>	Nacional (expansão escalável)	Alto	R\$ 30-80	Alto	Plataforma 100% pet-friendly com rastreamento em tempo real

Fonte: Base de pesquisas dos autores

GRÁFICO 1- Nicho De Mercado Pet-Friendly no Brasil



Fonte: Base de pesquisas dos autores

O **Bicho Express** tem como diferencial a **especialização em transporte pet**, aliado à praticidade de aplicativos já consolidados. Seu modelo escalável e digital o coloca em posição de vantagem competitiva, com potencial de conquistar rapidamente a confiança dos tutores e expandir para grandes centros urbanos.

## 2. Definição do Modelo de Processo

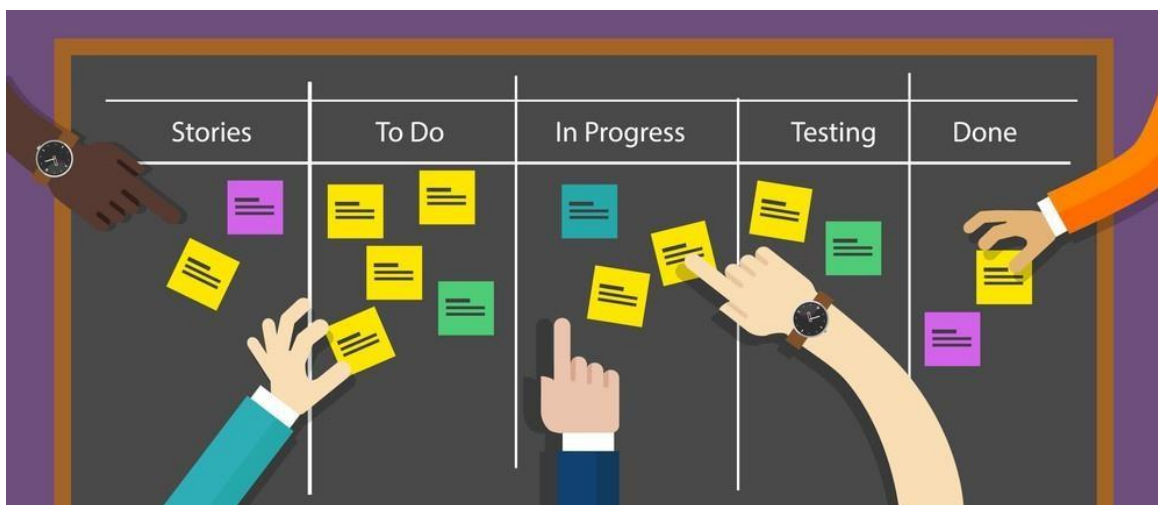
A empresa adotou o Scrum como modelo de desenvolvimento para a plataforma de transporte Bicho Express, aproveitando sua flexibilidade e agilidade para projetos complexos e dinâmicos. Com ciclos curtos de trabalho (sprints), estamos entregando funcionalidades rapidamente, permitindo ajustes contínuos baseados no feedback dos usuários e nas demandas do mercado.

Essa metodologia garante foco constante na experiência do usuário, priorizando usabilidade, segurança e inovação. Além disso, facilita a coordenação eficiente entre as equipes multidisciplinares, desde o desenvolvimento mobile até o suporte, assegurando que a arquitetura baseada em microserviços seja implementada de forma escalável e modular.

Com o Scrum, o projeto conta com total transparência no progresso, redução de riscos e entregas contínuas de valor. Isso acelera o lançamento no mercado, garantindo que a solução atenda às expectativas reais dos clientes e fortaleça a posição da empresa no mercado de mobilidade pet.

Optamos por não utilizar modelos tradicionais como Cascata, Iterativo, Prototipação ou Espiral devido às suas limitações em lidar com a complexidade, necessidade de adaptação rápida e constante evolução do projeto. Esses modelos são mais rígidos ou burocráticos, o que poderia dificultar a incorporação de feedbacks e atrasar entregas essenciais, enquanto o Scrum oferece a agilidade e colaboração necessárias para garantir o sucesso dessa iniciativa inovadora.

FIGURA 2- Demonstrativo de Funcionamento Método Scrum



### 3. Desenvolvimento do Software

#### 3.1. Diagrama de Casos de Uso- Funcionamento do Aplicativo do Usuário/Tutor de Pets

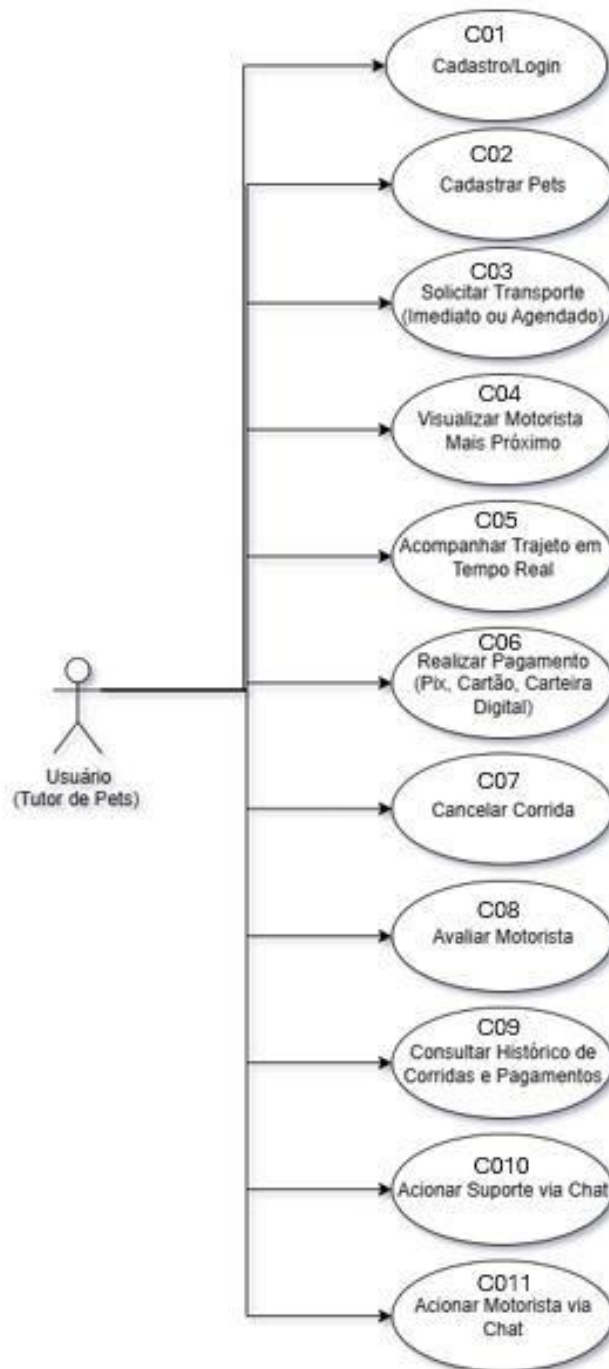


FIGURA 3- Diagrama caso de uso do cliente, (Fonte: Autores realizado no Draw.io, 2025)

### 3.2. Diagrama de Casos de Uso- Funcionamento Aplicativo Funcionário/Motorista

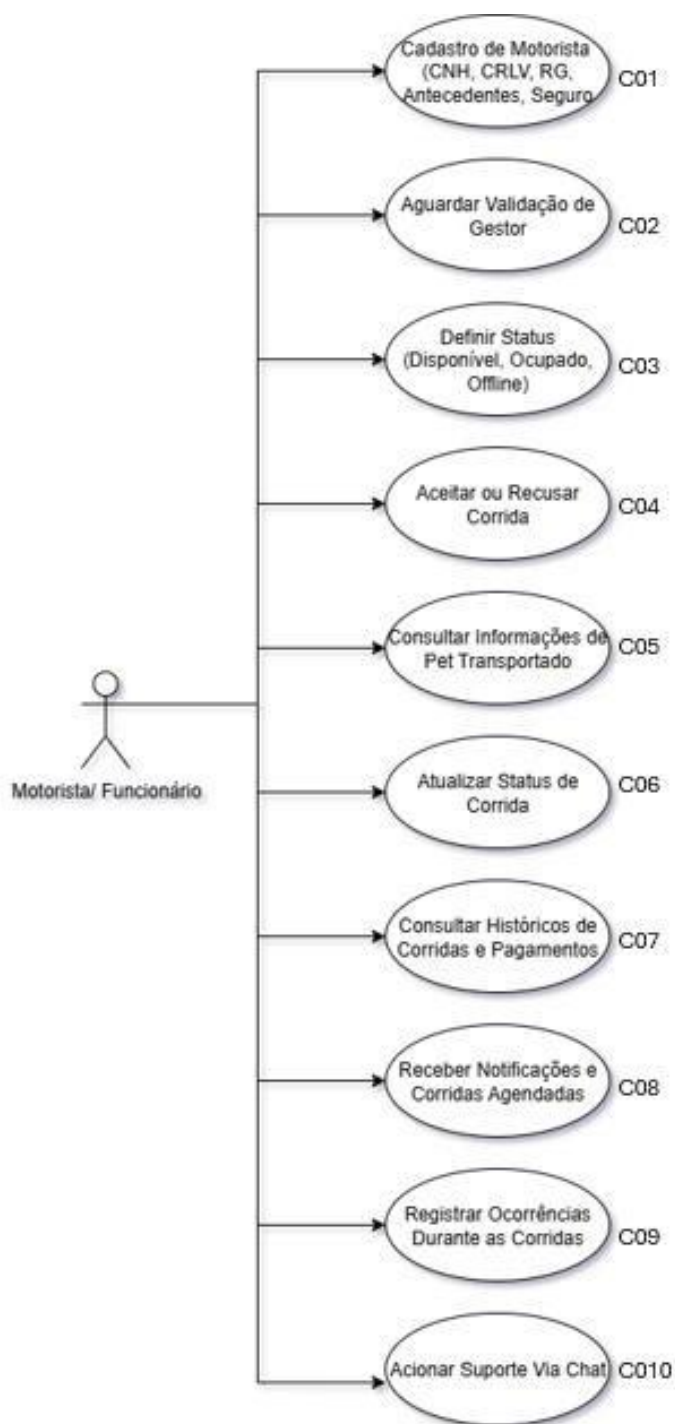


FIGURA 4- Diagrama caso de uso do Funcionário/Motorista, (Fonte: Autores realizado no Draw.io, 2025)

### 3.3. Diagrama de Casos de Uso- Funcionamento Gestor do Software

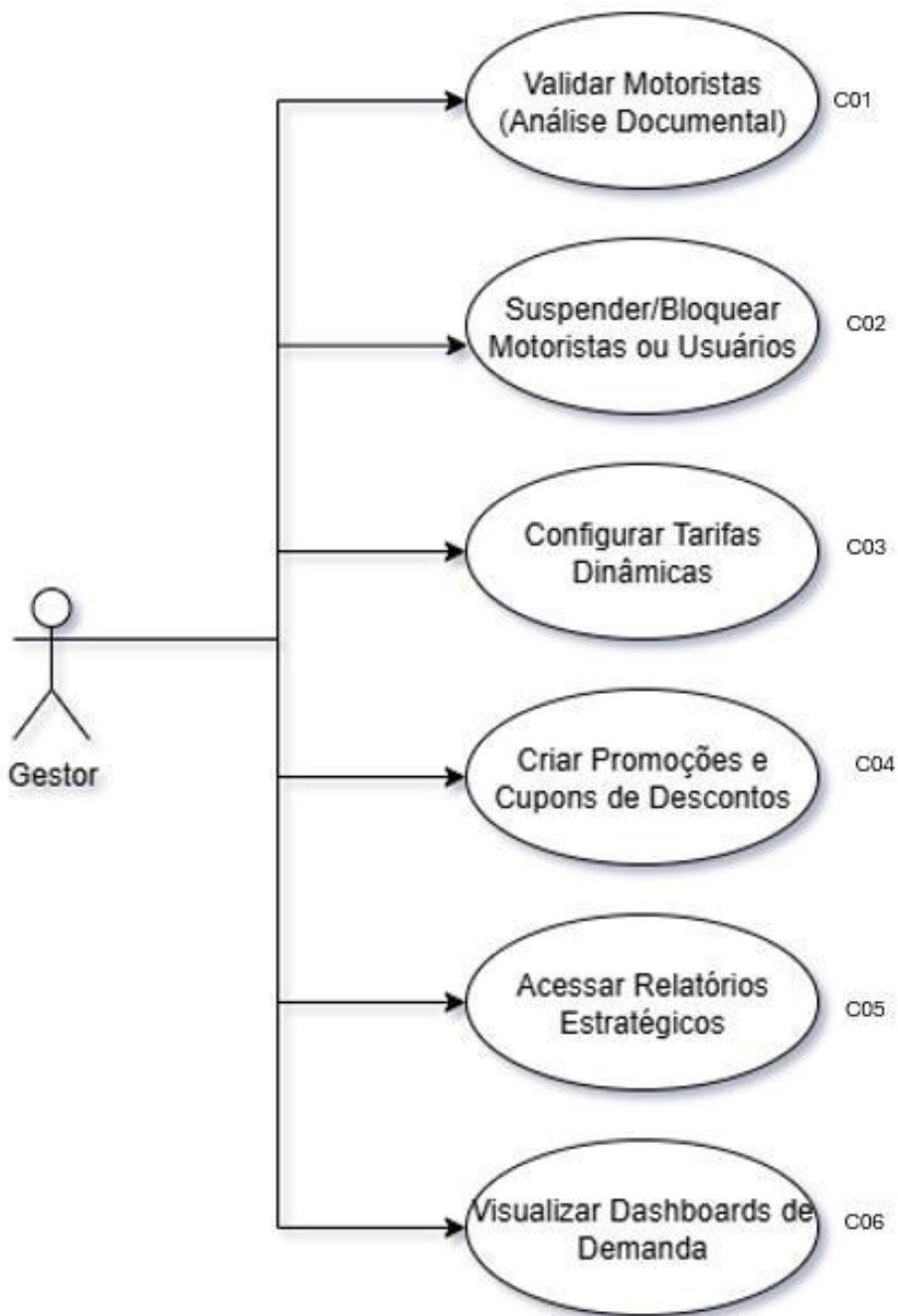


FIGURA 5- Diagrama caso de uso do Gestor, (Fonte: Autores realizado no Draw.io, 2025)

### 3.4. Diagrama de Casos de Uso- Funcionamento do Software Setor Administrativo

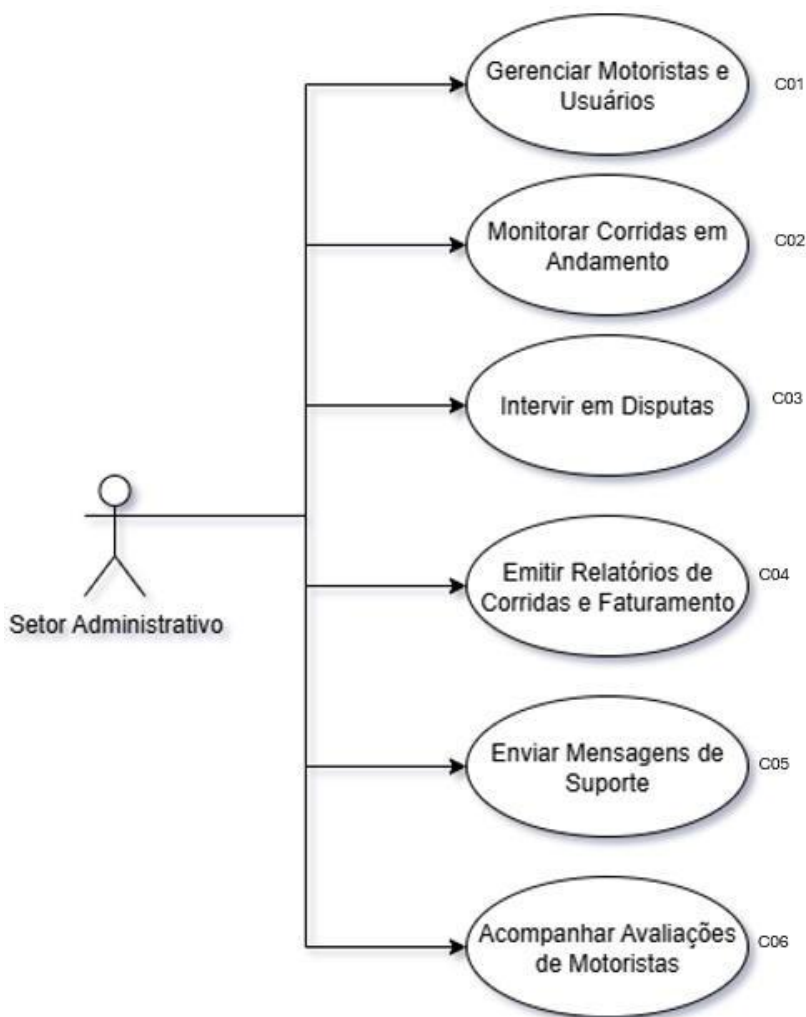


FIGURA 6- Diagrama caso de uso do Setor Administrativo, (Fonte: Autores realizado no Draw.io, 2025)

### 3.5. Catálogo de Atores

**TABELA 2- Catálogo de Atores do Software**

Ator	Tipo	Descrição	Principais Interações com o Sistema
[AT1] Usuário (Tutor de Pets)	Humano	Pessoa que utiliza o aplicativo para solicitar o transporte para seus animais.	Cadastro/login, cadastro de pets, solicitar transporte, pagamento, avaliação de motoristas, acompanhamento em tempo real, solicitar auxílio via chat.
[AT2] Motorista/ Funcionário	Humano	Profissional responsável por realizar o transporte dos pets.	Cadastro/ login e validação documental, aceitar corridas, atualizar status da corrida, consultar informações de pets e corridas, registrar ocorrência de pets, solicitar auxílio via chat.
[AT3] Setor Administrativo	Humano	Equipe/ sistema de suporte e monitoramento do software.	Gerenciar usuários e motoristas, monitorar corridas em andamento, emitir relatórios, suporte via chat.
[AT4] Gestor	Humano	Responsável/Sistema pela validação de motoristas e definição de políticas do sistema.	Validar documentos de motoristas, gerenciar usuários e motoristas, suspender contas, gerenciar tarifas, criar campanhas promocionais, acessar relatórios estratégicos.
[AT5] Sist. de Pagamentos	Externo API	Serviço de pagamento digital Integrado no Software.	Processar pagamentos, validar transações, gerar comprovantes.
[AT6] Google Login (OAuth2)	Externo API	Serviço de Autenticação social para login simplificado.	Permitir login via Google com segurança.
[AT7] Google Maps/ Geolocalização	Externo API	Serviço de mapas e rotas (GPS)	Calcular rotas, exibir localização em tempo real, estimar tempo de chegada.
[AT8] FireBase (Banco de Dados)	Externo Serviço	Banco de dados em Nuvem para a persistência de dados.	Armazenar informações de usuários, pets, motoristas, corridas e históricos.
[AT9] Sist. Notificações (FireBase Cloud)	Externo API	Serviço de envio de mensagens push para dispositivos móveis.	Notificar motoristas e usuários sobre corridas, pagamentos e status.
[AT10] Serviço de Validação (CEP, CPF, Telefone)	Externo API	Serviço público/ privado externo para a veracidade de dados pessoais.	Validar Informações de usuários e motoristas no momento do cadastro.
[AT11] Sist. de Monitoramento e Logs (AWS CloudWatch)	Externo Serviço	Serviço de monitoramento de métricas	Garantir disponibilidade, registrar falhas e analisar desempenho do sistema.

Fonte: Autores, 2025

### 3.6. Base de Preços

TABELA 3- Base de Preços Calculada Com Softwares concorrentes

Tipo de Corrida	Tarifa Base R\$	Valor por Km (R\$/Km)	Taxa de Espera (R\$/min)	Macrossazonal	Observações
Corrida Curta (Até 5Km)	R\$10,00	R\$3,00	R\$0,50	-	Médias Urbana; similar a PetDriver
Corrida Média (5-15Km)	R\$8,00	R\$2,80	R\$0,50	-	R\$2,50-3,50/Km praticados por táxi Dog
Corrida Longa (>15Km)	R\$6,00	R\$2,50	R\$0,40	-10\$ em horários de baixa (manhã e tarde)	Descontos em trechos mais longos
Taxa extra (+3 animal)	-	-	-	+R\$15,00 Fixa	Cobrança fixa por mais de 2 pets
Taxa de Limpeza	-	-	-	R\$50,00	Limpeza em caso de incidentes
Horário de Pico	-	+20%	-	+20% Tarifa/Km	Finais de Semana, feriados e noites.
Tempo de Espera >5 min	-	-	R\$1,00	-	Após 5min grátis, começa a cobrança
Viagem Interestadual/extra-municipal	R\$100,00	R\$3,80	-	+15%	Valores entre R\$3,80-4,00/Km

Fonte: Autores, 2025



### **3.7. Requisitos Funcionais**

Os requisitos funcionais definem o comportamento esperado do sistema sob a perspectiva dos usuários, descritos de forma simples para facilitar a compreensão. São organizados por tipos de usuários.

#### **3.7.1 Requisitos do Usuário (Tutor de Pets)**

RF1 O sistema deve permitir cadastro/login via e-mail, senha, Google e autenticação multifator (MFA).

RF2 O usuário deve poder cadastrar múltiplos pets, com informações detalhadas como espécie, porte, raça, peso, idade, vacinas e necessidades especiais.

RF3 O usuário pode solicitar transporte imediato ou agendado para seus pets.

RF4 Deve exibir o motorista mais próximo, com estimativas de tempo e valor da corrida.

RF5 O tutor pode acompanhar o transporte em tempo real via mapa.

RF6 O sistema deve oferecer múltiplas formas de pagamento, incluindo Pix, cartão de crédito/débito e carteira digital.

RF7 O usuário pode cancelar corridas conforme regras definidas no sistema.

RF8 Deve possibilitar avaliação de motoristas após corrida.

RF9 Usuários podem consultar histórico de corridas e pagamentos realizados.

RF10 Deve permitir acionamento de suporte ao cliente via chat integrado.

#### **3.7.2 Requisitos do Motorista/Funcionário**

RF11 O sistema deve permitir cadastro de motoristas, com documentos obrigatórios (CNH, antecedentes criminais, CRLV, seguro).

RF12 Apenas motoristas validados por um gestor autorizado podem realizar corridas.

RF13 Motoristas devem indicar seu status: Disponível, Ocupado ou Offline.

RF14 Devem receber solicitações de transporte podendo aceitar ou recusar.

RF15 Motoristas têm acesso a informações do pet transportado, incluindo porte, cuidados e restrições.

RF16 Devem atualizar o status da corrida (a caminho, embarcado, em andamento, concluído).

RF17 Motoristas podem consultar histórico de corridas e seus ganhos.

RF18 Devem receber notificações sobre corridas agendadas.

RF19 Possibilidade de registrar ocorrências durante a corrida.

### **3.7.3. Requisitos do Setor Administrativo**

RF20 Deve poder cadastrar, alterar e excluir motoristas e usuários em situações de suporte.

RF21 Monitorar corridas em andamento via painel de controle.

RF22 Intervir em disputas entre usuários e motoristas.

RF23 Emitir relatórios detalhados de corridas, faturamento e cancelamentos.

RF24 Enviar mensagens de suporte via chat interno.

RF25 Acompanhar estatísticas de avaliações de motoristas.

### **3.7.4 Requisitos do Gestor**

RF26 Validar motoristas após análise documental.

RF27 Suspender ou bloquear motoristas ou usuários denunciados.

RF28 Configurar tarifas dinâmicas baseadas em preço por km, porte do pet e taxa mínima.

RF29 Criar campanhas promocionais e cupons de desconto.

RF30 Acessar relatórios estratégicos sobre receitas, clientes ativos e cancelamentos.

RF31 Acessar dashboards com previsões de demanda por região e horários de pico.

### **3.8. Requisitos Não Funcionais**

Os requisitos não funcionais especificam restrições gerais do sistema, garantindo qualidade e eficiência.

#### **3.8.1 Requisitos Externos**

##### **3.8.1.1 Interoperabilidade**

RNF1 Permitir login integrado com Google.

RNF2 Integrar APIs bancárias, CPF, CEP e telefone para validações.

##### **3.8.1.2 Éticos**

RNF3 Garantir transparência total nos preços antes da corrida.

RNF4 Apenas motoristas aprovados podem realizar transporte de pets.

RNF5 Respeitar princípios de bem-estar animal durante o transporte.

##### **3.8.1.3 Legais**

RNF6 Estar em conformidade com a Lei Geral de Proteção de Dados (LGPD).

RNF7 Utilizar criptografia para proteger dados sensíveis.

RNF8 Seguir a norma ISO/IEC 27001 para segurança da informação.

#### **3.8.2 Requisitos Organizacionais**

##### **3.8.2.1 Entrega**

RNF9 Apenas motoristas validados devem realizar corridas.

RNF10 Filtrar motoristas por localização para otimização das corridas.

RNF11 Permitir pagamento apenas por meios digitais.

RNF12 Bloquear perfis denunciados quando confirmadas irregularidades.

RNF13 Permitir comunicação via chat com criptografia de ponta a ponta.

### **3.8.2.2 Implementação**

RNF14 Todas as páginas devem carregar em até 2 segundos.

RNF15 Notificações devem ser entregues em até 5 segundos.

RNF16 Campos de texto devem acionar teclado automaticamente na interface móvel.

RNF17 Textos devem usar fonte padrão, com títulos destacados em negrito.

### **3.8.2.3 Padrões**

RNF18 Todas as telas devem seguir design consistente.

RNF19 Botões e campos interativos devem ter comportamentos previsíveis.

RNF20 Código do sistema deve seguir boas práticas de desenvolvimento seguro.

### **3.8.3 Requisitos de Facilidade de Uso**

RNF21 A interface deve ser intuitiva e acessível a pessoas com diferentes perfis e necessidades.

RNF22 Cadastro e login devem ser concluídos em até 30 segundos.

RNF23 A busca por motoristas deve permitir filtros por localização, porte do animal, preço e avaliações.

RNF24 Sistema deve oferecer acessibilidade, incluindo alto contraste, leitores de tela e fontes ajustáveis.

### **3.8.4 Requisitos de Confiabilidade**

RNF25 Garantir disponibilidade do sistema em 99,9% do tempo.

RNF26 Realizar backups diários com retenção mínima de 12 meses.

RNF27 Tempo máximo para recuperação do sistema após falhas críticas é de 15 minutos.

RNF28 Notificar automaticamente equipe técnica em caso de falhas graves.

### **3.8.5 Requisitos de Portabilidade**

RNF29 Aplicativo deve ser compatível com Android 8.0+ e iOS 12+.

RNF30 Aplicativo não deve ultrapassar 100MB e deve consumir pouca bateria.

RNF31 Desenvolver em framework multiplataforma (React Native ou Flutter) para facilitar manutenção e escalabilidade.

### **3.8.6 Requisitos de Eficiência**

RNF32 Consultas e operações devem responder em até 2 segundos.

RNF33 Suportar até 100 mil usuários simultâneos.

RNF34 O desempenho do sistema deve ser monitorado continuamente para identificação e mitigação de gargalos.

### **3.8.7 Requisitos de Espaço**

RNF35 O sistema deve suportar crescimento da base de dados de corridas, pets e motoristas.

RNF36 Possibilitar escalabilidade horizontal, adicionando servidores conforme demanda.

RNF37 Compatibilidade com armazenamento em nuvem híbrida para garantir disponibilidade e segurança de dados.

### **3.9. Fluxo de Requisitos do Software**

#### **3.9.1. Aplicativo 1- Bicho Express (Usuário/Tutor de Pets)**

##### **3.9.1.1. RF01- Cadastro (Login)**

Este requisito do Cliente na plataforma. O ator pode fazer seu cadastro pelo Google ou pela plataforma inserindo seus dados manualmente.

Fluxo de Requisitos:

1. Após o Cliente instalar e abrir o aplicativo ele é direcionado para a página inicial.
2. Nesta página existem três opções: "Continue pelo Google", fazer o login já possuindo conta, sendo solicitado o e-mail e a senha já cadastrados no momento da criação da conta, e a opção "registrar-se", para cadastrar um novo usuário.
3. Crie seu cadastro direciona o cliente a uma nova página com as seguintes solicitações de campos pessoais do tutor do pet a serem preenchidos:
  - a) E-mail;
  - b) Data de nascimento (sendo o mínimo de 18 anos)
  - c) Nome completo;
  - d) CPF;
  - e) Telefone
  - f) Senha
  - g) Confirmação de senha
4. Um botão de "Próximo" finaliza o processo
5. O aplicativo aceita somente uma inscrição por CPF
6. O CPF, telefone e e-mail são validados pelo banco de dados correspondentes (APIs)
7. Após a validação o sistema salva todas as informações do ator cliente
8. Com os dados salvos a mensagem "Cadastro Concluído" é imprimido na tela
9. Adicionado a página 'perfil' do aplicativo
10. O cliente pode consultar os seus dados
11. O cliente pode escolher sair da sua conta a partir do botão "sair" no seu perfil
12. O cliente não pode excluir diretamente os seus dados
13. O cliente pode excluir seu cadastro encerrando sua participação no aplicativo

##### **3.9.3.1.2. Subfluxos [SUB1]: Login pelo Google**

1. O cliente tem a possibilidade de criar seu cadastro através da plataforma do Google
2. Um botão localizado na parte inferior da tela escrito "continuar com o Google"
3. Após ser selecionado o botão o cliente deve ser direcionado para uma aba do Google de autorização de cadastro na plataforma
4. Cliente autoriza

5. Google transfere as informações do cliente para o banco de dados da plataforma
6. Após as informações serem transferidas inserir a mensagem de "cadastro concluído"
7. O cliente é direcionado para a nova página "perfil" do aplicativo

#### **3.9.3.1.3. Subfluxo [SUB2]: Cliente já possui cadastro**

1. O cliente tem a possibilidade de inserir o cadastro já criado em um acesso anterior
2. As caixas de texto no centro da tela direcionam o usuário a inserir seu e-mail e senha já cadastrados anteriormente
3. Após a validação do cliente, o sistema salva todas as informações do ator cliente
4. Após as informações transferidas, inserir a mensagem "Login concluído"
5. O cliente é transferido diretamente para a aba principal de chamada de nova corrida.

#### **3.9.3.1.4. Subfluxo [SUB3]: Telefone**

1. O campo telefone e CPF são obrigatórios de serem preenchidos
2. O número de telefone e CPF são vinculados ao inserido no cadastro e na conta Google
3. Par atualizar o telefone é necessário clicar no campo "telefone", apagar o antigo e inserir um novo
4. Não é permitido alterar o CPF de uma conta já existente
5. É obrigatório pelo menos um telefone por conta e usuário
6. Quando o novo telefone for escrito o cliente deve pressionar o botão ao lado "salvar as informações"
7. Após vinculado, uma nova deverá ser impressa "Novo telefone cadastrado com sucesso!"

#### **3.9.3.1.5. Subfluxo [SUB4]: Saída do aplicativo**

1. Ao entrar na aba "perfil" do aplicativo, o cliente tem a possibilidade de pressionar o botão "sair"
2. Ao pressionar este botão uma mensagem de confirmação aparece
3. Confirmando a sua saída, o cliente tem a possibilidade de se cadastrar novamente com a mesma conta, ou logar com um registro diferente

#### **3.9.3.1.6. Subfluxo[SUB5]: Exclusão de cadastro**

1. Possível apenas depois de concluída a etapa de criação de perfil
2. O cliente seleciona a opção de "excluir conta"
3. Ao pressionar este botão uma mensagem de confirmação aparece, onde o administrador também alerta sobre as consequências causadas ao aceitar essa exclusão
4. Após a validação, o sistema exclui o perfil assim como o solicitado

5. Com os dados deletados a mensagem "cadastro deletado" é imprimida na tela

### **3.9.1.2. Fluxos alternativos [FA]:**

#### **3.9.1.2.1. Fluxo alternativo [FA1]: Restrição de idade**

1. Ocorre quando o usuário tenta realizar a criação de um cadastro , registrando a sua idade como sendo de alguém com menos de 18 anos
2. Ao tentar, uma mensagem automática aparece:" cadastro não permitido para menores de 18 anos"
3. O cadastro só poderá ser concluído se a data de nascimento informada atender ao requisito de maioridade

#### **3.9.1.2.2. Fluxo Alternativo [FA2]: Nome incompatível**

1. Este fluxo é acionado quando o nome informado não coincide com o registrado no CPF
2. Ao tentar, uma mensagem automática aparece. "inserir nome registrado no CPF"
3. O cadastro só poderá ser concluído se o nome informado for o mesmo encontrado no CPF

#### **3.9.1.2.3. Fluxo Alternativo [FA3]: Desenvolvimento de senha**

1. Ocorre quando o cliente não procede com as regras delimitadas para a construção de uma senha fortificada
2. Enquanto não obedecer esses itens, uma mensagem de alerta será emitida, avisando que o login não pode ser concluído até que essa etapa seja concluída
3. Quando as regras forem seguidas e as outras informações forem preenchidas, é possível concluir o cadastro

#### **3.9.1.2.4. Fluxo alternativo [FA4] : Confirmação de senha incorreta**

1. O cliente insere uma senha no campo "confirmação de senha", a qual não está correlacionada com a adicionada no campo "senha"
2. Enquanto as senhas descritas não forem compatíveis, uma mensagem de alerta será emitida, alertando o erro e solicitando a inserção das informações corretamente
3. Quando as regras forem seguidas e as outras informações forem preenchidas, é possível concluir o cadastro

#### **3.9.1.2.5. Fluxo alternativo [FA5]: Login ou senha incorretos**

1. Ao realizar a tentativa de inserir um cadastro já criado previamente e o mesmo não for localizado no banco de dados, uma mensagem será emitida avisando o preenchimento incorreto do login e ou senha



2. O cadastro só poderá ser concluído quando as informações forem inseridas corretamente

#### **3.9.1.2.6. Fluxo Alternativo [FA6]: Recuperar senha**

1. Caso o cliente esqueça a senha utilizada em sua conta, é possível que ele realize a recuperação da mesma
2. Um e-mail de recuperação será enviado onde o usuário poderá criar uma nova senha ao clicar no botão "criar nova senha"
3. Esse botão levará a uma página onde pode ser inserida e aprovada para poder ser utilizada
4. O login deve ser realizado novamente no aplicativo

**3.9.1.2. RF02- Cadastro de Pets:** Este requisito é voltado para a criação do perfil do(s) Pet(s) que serão transportados pelo serviço de aplicativo

#### **Fluxo de Requisitos**

1. Após o cadastro do tutor concluído com sucesso, o usuário é direcionada para uma página onde poderá realizar o cadastro de um ou mais pets
2. O usuário tem a opção de visualizar, editar e adicionar os campos
  - a. Nome do animal
  - b. Espécie
  - c. Raça
  - d. Porte
  - e. Peso
  - f. Idade
  - g. Se é vacinado, esterilizado/castrado
  - h. Condições médicas
  - i. Outras necessidade especiais
3. Esta etapa deve ser concluída para a solicitação de uma nova corrida
4. Avaliações e comentários recebidos ficarão expostos nesta página
5. Perfil do usuário e do animal ficará visível ao prestador de serviços/motorista, quando a corrida estiver sendo solicitada
6. Após a edição completa do perfil do pet, o usuário será direcionado a aba de solicitar nova corrida
7. O perfil poderá ser acessado a qualquer momento pelo usuário para a edição e consulta

#### **3.9.1.2.1. Subfluxo [SUB1]:**

##### **Campos não inseridos**

1. Ocorre quando algum dos campos obrigatórios(nome, estado de vacinação, porte, espécie e peso) do cadastro do pet está incompleto, inválido ou não preenchido
2. O sistema deve imprimir uma mensagem de "dados incompletos" até que os campos sejam plenamente inseridos

3. O sistema após os dados inseridos corretamente deve concluir o cadastro do pet e direcionar o usuário a aba de solicitação de nova corrida

#### 3.9.1.2.3. Subfluxo [SUB2]:

Visibilidade ao Prestador

1. Quando o usuário solicitar um serviço de corrida através do formulário de solicitação, quando o motorista receber a solicitação de corrida, o perfil do usuário e do pet será visível
2. O prestador poderá acessar o perfil do usuário e do pet com mais detalhes clicando no botão "mais detalhes"

#### 3.9.1.3. RF03 — Solicitar Transporte (Imediato ou Agendado)

O usuário pode solicitar transporte para um ou mais pets, escolhendo entre corrida imediata (ao vivo) ou corrida agendada (data/hora futura). Deve informar origem, destino, pet(s) e opções (veículo adaptado, necessidade especial).

##### Pré-condições

- Usuário cadastrado e logado (RF01).
- Pelo menos 1 pet cadastrado (RF02) antes de solicitar corrida.
- Localização (GPS) habilitada no dispositivo ou endereço digitado.

##### Fluxo de Requisitos

1. Usuário abre app e seleciona "Solicitar Corrida".
2. App exibe tela com opções: **Imediato / Agendado**.
3. Usuário escolhe pet(s) a transportar (checkbox), origem (GPS ou endereço), destino (endereço), e pode incluir observações/necessidades especiais.
4. Usuário seleciona tipo de veículo, se disponível (ex.: veículo para animais grandes, caixa de transporte).
5. App calcula **estimativa de preço** (Tarifa base + R\$/km + adicionais por porte/animal extra) e **ETA** (tempo estimado de chegada).
6. Usuário confirma solicitação; o sistema envia solicitação a motoristas compatíveis (por localização e requisitos do pet).
7. Motorista aceita a corrida (RF14) — sistema notifica usuário com dados do motorista (nome, avaliação, veículo, tempo estimado).
8. App passa para fluxo de acompanhamento (RF05) até conclusão.
9. Ao finalizar, app direciona para pagamento (RF06) e avaliação (RF08).

##### 3.9.1.3.1. Subfluxos

- **SUB1 — Agendamento com Google Calendar:** se usuário optar, o agendamento pode ser sincronizado com Google Calendar (com permissão).
- **SUB2 — Seleção de múltiplos pets:** usuário marca mais de um pet; sistema recalcula preço e verifica capacidade do veículo.

- **SUB3 — Escolha de método de pagamento pré-aprovado:** se o usuário tiver carteira cadastrada, pode confirmar pagamento automático.

#### 3.9.1.3.2. Fluxos Alternativos

- **FA1 — Nenhum motorista disponível:** caso não haja motoristas compatíveis/online, app informa “Sem motoristas disponíveis” e sugere agendamento posterior ou notificação quando houver disponibilidade.
- **FA2 — Aceite recusado / timeout:** se nenhum motorista aceitar em X segundos (configurável), sistema reprova a busca e notifica o usuário.
- **FA3 — Endereço inválido:** se o endereço informado for inconsistente/fora de área atendida, app exibe “Endereço fora da área de cobertura”.
- **FA4 — Requisitos especiais não compatíveis:** se o pet tem necessidades que exigem veículo/treinamento específico e não há motoristas compatíveis, exibe mensagem e opções (reprogramar / contatar suporte).

#### Pós-condições

- Solicitação criada no banco de dados com status: PENDENTE → ACEITA → A CAMINHO → EMBARCADO → CONCLUÍDO / CANCELADO.
- Notificações enviadas ao motorista e usuário.
- Registro de estimativa de preço e detalhe do pet.

#### Regras de negócio

- Máximo de X pets por corrida (definir política).
- Taxa extra por animal a partir do 3º pet.
- Multa de cancelamento se cancelado após motorista sair para coleta (ver política do gestor — RF28).

#### 3.9.1.4. RF04 — Exibir Motorista Mais Próximo / Estimativa (Driver Matching)

Mostrar ao usuário os motoristas disponíveis mais próximos, estimativa de chegada (ETA) e preço estimado antes da confirmação.

#### Pré-condições

- GPS/serviço de localização ativo.
- Motoristas online com status “Disponível” (RF13).

#### Fluxo de Requisitos

1. Após inserir origem/destino, app chama serviço de **matching** (API) que busca motoristas por proximidade e compatibilidade (porte do pet, veículo).
2. Service retorna lista ordenada por distância/tempo/nota.
3. App exibe o motorista sugerido (ou múltiplas opções se houver) com ETA e preço estimado.
4. Usuário pode visualizar perfil do motorista (avaliação, histórico) antes de confirmar.

#### 3.9.1.3.1. Subfluxos

- **SUB1 — Filtrar por requisitos:** usuário solicita somente motoristas com veículos adaptados ou experiência com pets agressivos; sistema aplica filtro no matching.
- **SUB2 — Mostrar alternativos:** botão “Ver outras opções” lista mais motoristas com ETAs e preços.

#### 3.9.1.3.2. Fluxos Alternativos

- **FA1 — Motorista indisponível no instante do clique:** sistema atualiza lista em tempo real; se motorista perder disponibilidade, sugere próxima opção.
- **FA2 — Diferença entre estimativa e preço final:** se percurso real divergir do estimado (ex.: desvio), o preço final será recalculado no final da corrida (detalhar na política).

#### Pós-condições

- Motorista indicado e requisitado; tela de espera enquanto aguarda aceitação.

#### Regras de negócio

- Priorizar motoristas por ETA e por avaliação média (configurável).
- Limite de tentativas de chamada a motoristas (p.ex. 3 motoristas em sequência).

#### 3.9.1.5. RF05 — Acompanhar Transporte em Tempo Real (Tracking)

Mostrar em mapa a localização do motorista, trajeto, status da corrida e tempo estimado de chegada.

#### Pré-condições

- Corrida aceita por motorista (RF14).
- Permissão de localização concedida pelo motorista e usuário (para rastreamento).

#### Fluxo de Requisitos

1. Ao aceitar a corrida, motoristas compartilham localização em tempo real via WebSocket / service de localização.
2. App do usuário exibe mapa com:
  - Posição atual do motorista;
  - Rota prevista e ETA;
  - Status textual (A caminho / Em trânsito / Concluído).
3. Usuário pode abrir painel com detalhes do motorista e pet (RF02 visível).
4. Durante transporte, o app atualiza estimativas e status.
5. Ao final da corrida, status muda para “Concluído” e segue fluxo de pagamento/avaliação.

#### 3.9.1.3.4. Subfluxos

- **SUB1 — Compartilhar localização temporária para suporte:** usuário pode permitir link de rastreamento por tempo limitado para outra pessoa.
- **SUB2 — Notificações automáticas:** push quando motorista a X minutos de chegada, quando embarcou, quando concluiu.

#### 3.9.1.3.5. Fluxos Alternativos

- **FA1 — Perda temporária de sinal GPS:** app exibe última posição conhecida e notifica o usuário; tenta reconexão automática.
- **FA2 — Desvio de rota:** se motorista fizer desvio maior que Y%, sistema solicita justificativa ao motorista e notifica usuário.

#### Pós-condições

- Histórico de rastreamento armazenado (logs) para auditoria.

#### Regras de negócio

- Políticas de privacidade sobre armazenamento do trajeto (retention conforme LGPD).
- Tempo máximo de exposição do tracking para terceiros (link temporário).

#### 3.9.1.6. RF06 — Pagamento (Pix, Cartão, Carteira Digital)

Permitir que o usuário efetue pagamento da corrida através de meios digitais integrados (Pix, cartão, carteira digital). Deve suportar pagamento pré-autorizado ou no final da corrida.

#### Pré-condições

- Usuário com método de pagamento cadastrado ou pronto para inserir no momento.
- Integração com gateway de pagamento ativa.

#### Fluxo de Requisitos

1. Ao confirmar corrida, usuário escolhe método de pagamento:
  - **Pagamento automático (pré-aprovado):** cartão salvo / wallet;
  - **Pagamento no app (ao final):** selecionar Pix, Cartão ou Carteira.
2. Ao término da corrida, app calcula valor final (tarifa base + km + tempo + adicionais).
3. Se método automático: gateway captura o valor; usuário recebe comprovante.
4. Se Pix: app gera QR / chave e confirma recebimento; ou redireciona para app de banco.
5. Se carteira: processa via API correspondente.
6. App gera recibo e atualiza histórico (RF09). Motorista recebe confirmação de pagamento na sua interface.

#### 3.9.1.3.7. Subfluxos

- **SUB1 — Pagamento dividido/vale:** possibilidade futura de dividir pagamento (não obrigatório no MVP).
- **SUB2 — Promoções/cupons:** aplicar cupom antes do pagamento (RF29 configurado pelo gestor).

#### 3.9.1.3.8. Fluxos Alternativos

- **FA1 — Falha na captura do cartão:** app solicita método alternativo; usuário pode tentar novamente.
- **FA2 — Chargeback / disputa:** caso de disputa, transações retidas e suporte notificado (RF20).
- **FA3 — Cancelamento com pagamento pré-aprovado:** política de reembolso conforme RF7 e regras do gestor (RF28).

#### Pós-condições

- Transação registrada, recibo gerado, extrato do usuário atualizado.

#### Regras de negócio

- Compliance PCI-DSS para tratamento de cartão (guardar somente token, não dados completos).
- Taxa de serviço da plataforma aplicada automaticamente (definida por RF28).

#### 3.9.1.7. RF07 — Cancelamento de Corrida

O usuário pode cancelar uma corrida, com regras de cancelamento que podem gerar multa caso o cancelamento ocorra após determinado evento (ex.: após motorista iniciar deslocamento).

#### Pré-condições

- Corrida solicitada e em status PENDENTE/ACEITA/A CAMINHO.

#### Fluxo de Requisitos

1. Usuário acessa a corrida em andamento e pressiona “Cancelar”.
2. App exibe política de cancelamento (tempo grátis, multa aplicável).
3. Usuário confirma cancelamento.
4. Sistema:
  - Se dentro do período sem multa: atualiza status para CANCELADO sem cobrança.
  - Se multa aplicável: aplica taxa e processa cobrança (RF06), notifica motorista e setor administrativo.

5. Motorista recebe notificação do cancelamento e solicitação disponibilizada a outros motoristas.

#### **3.9.1.4.1. Subfluxos**

- **SUB1 — Cancelamento pelo motorista:** similar para motorista, com regras para o motorista (suspensão por recusas repetidas).

#### **3.9.1.4.2. Fluxos Alternativos**

- **FA1 — Usuário contesta multa:** inicia-se processo de disputa via suporte (RF20/24).
- **FA2 — Corrida em agendamento cancelada com prazo X antes:** sem multa.

#### **Pós-condições**

- Status da corrida atualizado e registro de multa (se houver).

#### **Regras de negócio**

- Definir janela grátis (ex.: cancelamento até 5 minutos sem custo).
- Política diferenciada para corridas agendadas.

#### **3.9.1.8. RF08 — Avaliação de Motoristas**

Usuário pode avaliar e deixar comentário após finalização da corrida (nota 1–5 e comentário opcional). Avaliação impacta perfis dos motoristas.

#### **Pré-condições**

- Corrida concluída (RF05 finalizada e pagamento realizado).

#### **Fluxo de Requisitos**

1. Após conclusão, app exibe tela de avaliação.
2. Usuário escolhe nota (1 a 5) e escreve comentário opcional.
3. Submete avaliação; o sistema atualiza média do motorista e armazena comentário.
4. Motorista pode visualizar avaliação em seu app (RF17) e setor admin pode monitorar (RF25).

#### **3.9.1.5.1. Subfluxos**

- **SUB1 — Avaliação anônima:** política de privacidade define se comentário mostra nome do usuário.
- **SUB2 — Denúncia associada:** opção de marcar incidentes graves que acionam RF20 (suporte).

#### **3.9.1.5.2. Fluxos Alternativos**

- **FA1 — Usuário não avalia:** é possível pular a avaliação; sistema pode enviar lembrete após X horas.

#### **Pós-condições**

- Nota e comentário salvos; média do motorista recalculada.

#### **Regras de negócio**

- Sistema de pesagem de avaliações (ex.: avaliações recentes têm peso maior).
- Mecanismo de moderação para comentários ofensivos.

#### **3.9.1.9. RF09 — Histórico de Corridas e Pagamentos**

Usuário pode consultar o histórico completo de corridas, recibos, detalhes do trajeto, data, valor e motorista atendente.

#### **Pré-condições**

- Histórico populado após corridas concluídas.

#### **Fluxo principal**

1. Usuário acessa menu “Histórico”.
2. App lista corridas por data, mostrando resumo (origem, destino, valor, status).
3. Ao clicar em um registro, usuário visualiza detalhes: mapa da rota, tempo, taxa aplicada, recibo (opção de exportar PDF/email).
4. Histórico financeiro exibe totais mensais e filtros (por período).

#### **3.9.1.5.1 Subfluxos**

- **SUB1 — Exportar recibo:** gerar PDF e enviar por e-mail.
- **SUB2 — Solicitar estorno / disputa:** iniciar processo de disputa para cobrança indevida (encaminha para RF20).

#### **3.9.1.5.2. Fluxos Alternativos**

- **FA1 — Registro ausente:** se corrida não aparece, usuário contata suporte; log de auditoria verificado.

#### **Pós-condições**

- Acesso ao registro e possibilidade de acionar suporte.

#### **Regras de negócio**

- Retenção de registros conforme LGPD e políticas (ex.: 12 meses mínimo, anonimização posterior conforme regras).



### **3.9.1.10. RF10 — Suporte via Chat Integrado**

Serviço de suporte ao cliente via chat in-app, para dúvidas, denúncias, problemas técnicos ou disputas de pagamento.

#### **Pré-condições**

- Usuário logado (pode também permitir acesso limitado para não-logados com dados básicos).

#### **Fluxo principal**

1. Usuário acessa “Suporte” / “Chat”.
2. Seleciona tópico (Ex.: Problema com cadastro, pagamento, corrida, denúncia).
3. Chat abre com atendente ou bot (triagem automática).
4. Para problemas simples, bot responde; para casos complexos, chat é encaminhado a atendente humano (Setor Administrativo — RF24).
5. Atendente visualiza contexto da corrida/histórico e pode solicitar ações (reembolsos, investigações).
6. Conversa arquivada no histórico do usuário e setor admin.

#### **3.9.1.6.1. Subfluxos**

- **SUB1 — Anexar evidências:** usuário envia foto, vídeo, prints (limite de tamanho).
- **SUB2 — Escalar para gestor:** para casos críticos, agente escalona para gestor (RF26/RF27).

#### **3.9.1.6.2 Fluxos Alternativos**

- **FA1 — Off-hours:** fora do horário de atendimento humano, bot registra ticket e informa tempo estimado de retorno.
- **FA2 — Spam / abuso:** sistema de moderação trata mensagens ofensivas, aplicando sanções se necessário.

#### **Pós-condições**

- Ticket de suporte criado, solução registrada, métricas de atendimento atualizadas.

#### **Regras de negócio**

- SLA de resposta definido (ex.: resposta inicial em 2 horas; resolução em X dias).
- Armazenamento das conversas conforme LGPD e política de retenção.

### **3.9.2. Aplicativo 1- Requisitos do Motorista / Funcionário**

#### **3.9.2.1. RF11 — Cadastro de Motorista (CNH, CRLV, antecedentes, seguro)**

Permitir que um motorista se cadastre na plataforma fornecendo dados pessoais e documentos obrigatórios para verificação.

##### **Pré-condições**

- Motorista possui dispositivo móvel com o app instalado.
- Motorista possui documentos digitais (fotos/scan) prontos.

##### **Fluxo de Requisitos**

1. Motorista abre app e seleciona “Quero ser motorista / cadastrar”.
2. Preenche dados pessoais: nome completo, data de nascimento, CPF, telefone, e-mail.
3. Anexa documentos exigidos: CNH (foto frente/verso), CRLV do veículo, comprovante de seguro, comprovante de antecedentes criminais.
4. Informa dados do veículo: marca, modelo, placa, capacidade (kg/pets), se possui itens de segurança para pets (cintos, caixas).
5. Envia e confirma cadastro.
6. Sistema realiza validações automáticas básicas (formato CPF, telefone) e envia documentos para fila de análise do gestor (RF26).
7. Motorista recebe tela com status: **Em análise** e notificação por push/email.

##### **3.9.2.1.1. Subfluxos**

- **SUB1 — Upload de documentos via câmera:** captura direta da câmera com validação de borda (imagem legível).
- **SUB2 — Salvamento temporário:** possibilidade de salvar rascunho do cadastro e continuar depois.

##### **3.9.2.1.2. Fluxos Alternativos**

- **FA1 — Documento ilegível/incompleto:** sistema solicita novo envio com mensagem explicativa.
- **FA2 — Dados inválidos (ex.: CPF já cadastrado):** alerta e direcionamento para recuperar conta ou suporte.

##### **Pós-condições**

- Cadastro criado com status **PENDENTE\_VALIDACAO**.
- Notificação enviada ao gestor para análise documental.

## Regras de negócio

- Motorista com CPF já vinculado a conta não pode se cadastrar novamente.
- Documentos expirados (ex.: seguro vencido) não são aceitos — sistema requer upload válido.

### 3.9.2.2. RF12 — Validação por Gestor (liberação para realizar corridas)

Fluxo de validação documental e aprovação por gestor para liberação do motorista na plataforma.

#### Pré-condições

- Cadastro do motorista concluído (RF11) e documentos anexados.

#### Fluxo de Requisitos

1. Gestor recebe notificação/posto de trabalho com lista de cadastros pendentes.
2. Gestor acessa painel de verificação documental (visualiza CNH, CRLV, seguros e antecedentes).
3. Gestor marca **Aprovar** ou **Rejeitar** com justificativa.
4. Se **Aprovar**: status do motorista atualiza para **Ativo** e motorista recebe notificação de liberação.
5. Se **Rejeitar**: motorista recebe notificação com justificativa e instruções para reenvio.

#### 3.9.2.2.1. Subfluxos

- **SUB1 — Solicitar documentação adicional**: gestor pede documentos extras via chat interno; motorista envia e processo reavalia.
- **SUB2 — Aprovação condicional**: gestor aprova com restrição (ex.: só pode atender pets pequenos) — tag de restrição aplicada ao perfil.

#### 3.9.2.2.2. Fluxos Alternativos

- **FA1 — Divergência nos documentos**: gestor abre ticket para investigação com setor administrativo (RF20).

#### Pós-condições

- Motorista fica **Ativo** (pode receber corridas) ou **Rejeitado** (não pode operar até regularização).

## Regras de negócio

- Aprovação manual é obrigatória antes de aceitar corridas.
- Histórico da decisão (quem aprovou, data) deve ser registrado para auditoria.

### 3.9.2.3. RF13 — Indicar Status (Disponível / Ocupado / Offline)

Motorista deve gerenciar seu status operacional no app.

#### Pré-condições

- Motorista logado e perfil ativo.

#### Fluxo principal

1. Motorista abre app e acessa painel de controle.
2. Seleciona um dos status: **Disponível**, **Ocupado**, **Offline**.
3. App atualiza status no backend e transmite alteração via websocket para o sistema de matching.
4. Usuários e sistema visualizam disponibilidade atualizada.

#### 3.9.2.3.1. Subfluxos

- **SUB1 — Status automático:** ao aceitar corrida, status muda automaticamente para **Ocupado**; ao concluir, volta para **Disponível**.
- **SUB2 — Horário de trabalho:** motoristas podem definir janelas de disponibilidade futura.

#### 3.9.2.3.2. Fluxos Alternativos

- **FA1 — Conexão perdida:** se app perde conexão, status define como **Offline** após timeout.

#### Pós-condições

- Motorista passa a ser elegível (ou não) para receber solicitações.

## Regras de negócio

- Motoristas que ficam repetidamente **Offline** após aceitar corridas podem ser penalizados (configurável).

#### 3.9.2.4. RF14 — Receber/aceitar/recusar solicitações de transporte

Motorista recebe solicitações compatíveis e decide aceitar ou recusar.

##### Pré-condições

- Motorista com status **Disponível**.

##### Fluxo de Requisitos

1. Motorista recebe notificação push com detalhes resumidos da corrida (origem, destino, tipo de pet, distância estimada, preço estimado).
2. No app, botão **Aceitar** e **Recusar** são apresentados.
3. Motorista tem X segundos para responder (ex.: 30s).
4. Ao **Aceitar**, app altera status para **A caminho** e direciona para o fluxo de navegação; usuário recebe confirmação com dados do motorista.
5. Ao **Recusar**, sistema tenta chamar próximo motorista elegível.

#### 3.9.3.4.1. Subfluxos

- **SUB1 — Aceite automático para corridas agendadas:** para corridas agendadas, motorista pode aceitar com antecedência.
- **SUB2 — Motivo da recusa:** opção de informar motivo (ex.: veículo incompatível) para métricas.

#### 3.9.3.4.2. Fluxos Alternativos

- **FA1 — Timeout:** se o motorista não responde em X segundos, é considerado **Recusar** e próximo motorista é acionado.
- **FA2 — Aceitar e depois cancelar:** se motorista cancela após aceitar, regras de penalidade aplicam-se.

##### Pós-condições

- Corrida atualizada para **Aceita** com link de navegação e dados trocados.

##### Regras de negócio

- Limite de recusas por período; excesso resulta em advertência/suspensão temporária.

### 3.9.2.5. RF15 — Acesso a informações do pet

Motorista pode visualizar, antes de aceitar (resumo) e após aceitar (detalhe), informações do pet (porte, necessidades, restrições).

#### Pré-condições

- Usuário cadastrou pet (RF02).
- Privacidade e consentimento respeitados (usuario autorizou compartilhar).

#### Fluxo de Requisitos

1. Ao notificação de corrida, motorista vê resumo do pet (espécie, porte, restrições).
2. Ao aceitar, motorista pode abrir perfil completo do pet (vacinação, condições médicas, observações).
3. Motorista confirma que possui condições para realizar o transporte (opcional checklist).

### 3.9.3.5.1. Subfluxos

- **SUB1 — Solicitar esclarecimento ao usuário:** botão para enviar mensagem antes de aceitar (ex.: confirmar se o pet está com guia).

### 3.9.3.5.2. Fluxos Alternativos

- **FA1 — Informação sensível faltante:** se dados críticos não estiverem preenchidos (ex.: risco de agressividade), motorista pode recusar sem penalidade.

#### Pós-condições

- Motorista tem ciência das exigências e pode preparar veículo/equipamentos.

#### Regras de negócio

- Motoristas não podem negar transporte por critério discriminatório; motivos para recusa devem ser registrados.

### 3.9.2.6. RF16 — Atualizar status da corrida (a caminho, embarcado, em andamento, concluído)

Motorista atualiza o ciclo de vida da corrida através de status.

#### Pré-condições

- Corrida aceita.

#### Fluxo de Requisitos

1. Motorista inicia navegação e pressiona **A caminho**.
2. Ao chegar ao ponto de coleta, pressiona **Cheguei / Embarcando**.
3. Após pet embarcado, atualiza para **Em transporte**.
4. Ao chegar no destino e finalizar, pressiona **Concluído**.
5. Cada alteração dispara notificações push para o usuário e atualiza o histórico.

#### 3.9.3.6.1. Subfluxos

- **SUB1 — Fotos de embarque/desembarque:** opção de registro fotográfico (para disputas).
- **SUB2 — Confirmação via código:** usuário confirma recebimento com código PIN para segurança.

#### 3.9.3.6.2. Fluxos Alternativos

- **FA1 — Ocorrência durante transporte:** motorista registra ocorrência (RF19) e altera status para **Em ocorrência**; suporte e usuário são notificados.

#### Pós-condições

- Registro de timestamps de cada status armazenado.

#### Regras de negócio

- Ordem de status deve ser respeitada (não pode ir de PENDENTE → CONCLUÍDO sem passar por A CAMINHO).
- Falta de atualização de status sem justificativa pode gerar penalidade.

### 3.9.2.7. RF17 — Histórico de corridas e ganhos

Motorista pode consultar histórico de corridas, valores recebidos, taxas e extratos.

#### Pré-condições

- Motorista ativo com corridas realizadas.

#### Fluxo de Requisitos

1. Motorista acessa menu **Histórico / Ganhos**.
2. Sistema lista corridas por data e exibe detalhes (valor bruto, taxa da plataforma, valor líquido, data).
3. Motorista pode filtrar por período (semana, mês) e exportar extrato em PDF/CSV.
4. Motorista visualiza saldo disponível e histórico de pagamentos (transferências efetuadas).

#### 3.9.2.7.1. Subfluxos

- **SUB1 — Solicitar pagamento:** botão para solicitar saque/transferência para conta bancária cadastrada.
- **SUB2 — Verificar disputas:** opção de visualizar corridas com disputas/chargebacks.

#### 3.9.2.7.2. Fluxos Alternativos

- **FA1 — Divergência no valor:** iniciar disputa para revisão do cálculo (RF20).

#### Pós-condições

- Extratos gerados e solicitações de saque registradas.

#### Regras de negócio

- Período de pagamento (ex.: semanal/mensal) e taxas de transferência devem estar explícitos.



### 3.9.2.8. RF18 — Notificações sobre corridas agendadas

Motorista recebe notificações para corridas futuras agendadas.

#### Pré-condições

- Motorista aceitou/agendou corrida com antecedência.

#### Fluxo de Requisitos

1. Sistema agenda notificações: 24h antes, 1h antes e 10 min antes (configurável).
2. Motorista recebe push / email com resumo e opção de confirmar presença.
3. Se motorista confirmar, sistema marca como **Confirmado**; se recusar, reabre busca.

#### 3.9.2.8.1. Subfluxos

- **SUB1 — Remarcar**: motorista pode propor novo horário (sujeito à aceitação do usuário).

#### 3.9.2.8.2. Fluxos Alternativos

- **FA1 — Não confirmação**: se motorista não confirmar até X horas antes, sistema tenta substituir.

#### Pós-condições

- Notificações e confirmações registradas.

#### Regras de negócio

- Política de substituição e penalidades por faltas em corridas agendadas.

### 3.9.2.9. RF19 — Registrar ocorrências durante a corrida

Motorista pode registrar eventos/ocorrências (acidentes, comportamento do pet, danos) durante ou após corrida.

#### Pré-condições

- Corrida em andamento ou finalizada.

## Fluxo principal

1. Motorista acessa opção **Registrar Ocorrência** no app da corrida.
2. Seleciona tipo de ocorrência (Saúde, Agressividade, Acidente, Danos ao veículo, Limpeza).
3. Anexa fotos/descrição e indicação de urgência.
4. Submete; sistema cria ticket e notifica setor administrativo e usuário simultaneamente.
5. Caso necessário, aciona políticas (reembolso, taxa de limpeza) e inicia fluxo de suporte (RF24).

### 3.9.2.9.2. Subfluxos

- **SUB1 — Acionamento de emergência:** opção para ligar para serviços de emergência/veterinário (se necessário).

### 3.9.2.9.2. Fluxos Alternativos

- **FA1 — Ocorrência falsa / disputa:** inicia investigação com logs, fotos e declarações.

## Pós-condições

- Ticket criado e rastreado no painel administrativo.

## Regras de negócio

- Todos os tickets devem ter SLA de resposta; ocorrências graves podem levar à suspensão provisória do motorista até investigação.

### **3.9.3 Aplicativo 1- Requisitos do Setor Administrativo**

#### **3.9.3.1. RF20 — Gerenciar (cadastrar, alterar, excluir) motoristas e usuários**

Painel administrativo para suporte com CRUD completo sobre perfis de motoristas e usuários (com logs e justificativas).

##### **Pré-condições**

- Usuário administrativo autenticado com privilégios adequados.

##### **Fluxo de Requisitos**

1. Admin acessa painel e busca usuário/motorista por filtros (nome, CPF, status).
2. Admin pode editar dados cadastrais, anexar documentos, atualizar status ou excluir (com justificativa e registro).
3. Exclusão lógica (soft delete) por padrão — para conformidade LGPD, opção para exclusão definitiva mediante solicitação e auditoria.
4. Alterações geram logs de auditoria (quem, quando, o quê).

##### **3.9.3.1.1 Subfluxos**

- **SUB1 — Reativar perfil:** restaurar contas suspensas mediante análise.
- **SUB2 — Forçar validação:** solicitar nova documentação.

##### **3.9.3.1.2. Fluxos Alternativos**

- **FA1 — Tentativa de exclusão sem justificativa:** sistema nega até que justificativa seja informada.

##### **Pós-condições**

- Alterações persistidas e logadas.

##### **Regras de negócio**

- Alterações críticas (ex.: exclusão) requerem duplo fator de autenticação e justificativa.

### **3.9.3.2. RF21 — Monitorar corridas em andamento via painel**

Painel em tempo real com mapa e lista de corridas para monitoramento operacional.

#### **Pré-condições**

- Admin autenticado; sistemas de rastreamento funcionando.

#### **Fluxo de Requisitos**

1. Admin abre painel “Operações” e visualiza mapa com pins das corridas ativas.
2. Seleciona corrida para visualizar detalhes (usuário, motorista, rota, status, histórico de mensagens).
3. Pode enviar mensagens para motorista/usuário via chat interno e intervir (ex.: reatribuir corrida).

#### **3.9.3.2.1. Subfluxos**

- **SUB1 — Filtros operacionais:** filtrar por região, status, tipo de ocorrência.

#### **3.9.3.2.2. Fluxos Alternativos**

- **FA1 — Perda de telemetria:** se perda de sinal, painel mostra última posição e alerta equipe.

#### **Pós-condições**

- Ações tomadas e registradas (mensagens, reassign).

#### **Regras de negócio**

- Logs de intervenção são obrigatórios para auditoria.

### **3.9.3.3. RF22 — Intervir em disputas entre usuário e motorista**

Ferramentas para investigação e resolução de conflitos (reembolsos, advertências, bloqueios temporários).

#### **Pré-condições**

- Ticket de disputa criado (RF19 ou RF10).

### Fluxo de Requisitos

1. Admin acessa ticket com evidências (fotos, logs, avaliações).
2. Avalia contexto e decide: Reembolso total/ parcial, advertência ao motorista, suspensão temporária, bloqueio definitivo.
3. Aplica ação no sistema (processa reembolso via gateway, altera status do motorista) e notifica partes.
4. Registra decisão e motivo.

#### 3.9.3.3.1. Subfluxos

- **SUB1 — Pedido de informações:** solicitar provas adicionais ao motorista/usuário.

#### 3.9.3.3.2. Fluxos Alternativos

- **FA1 — Caso não resolvido:** escalar para gestor (RF26) com resumo.

### Pós-condições

- Disputa encerrada; ações aplicadas; registro de auditoria.

### Regras de negócio

- Reembolsos seguem política financeira e devem passar por aprovação gerencial dependendo do valor.

#### 3.9.3.4. RF23 — Emitir relatórios detalhados

Gerar relatórios de corridas, faturamento, cancelamentos e KPIs para análise.

### Pré-condições

- Admin autenticado com permissão de relatório.

### Fluxo de Requisitos

1. Admin acessa módulo de relatórios e define período/ filtros (região, motorista).
2. Gera relatório (tabela/CSV/PDF) com métricas: número de corridas, receita bruta, cancelamentos, avaliação média.
3. Pode agendar envio automático para e-mails executivos.

#### 3.9.3.4.1. Subfluxos

- **SUB1 — Relatórios customizados:** combinar múltiplos filtros e métricas.

#### 3.9.3.4.2. Fluxos Alternativos

- **FA1 — Dados incompletos:** alertar sobre ausência de dados e sugerir janela diferente.

#### Pós-condições

- Relatório exportado e armazenado.

#### Regras de negócio

- Relatórios financeiros sensíveis devem ter controle de acesso e logs.

#### 3.9.3.5. RF24 — Enviar mensagens de suporte via chat interno

Ferramenta de chat interno para atendimento ao cliente e comunicação com motoristas.

#### Pré-condições

- Admin autenticado; integração de chat funcionando.

#### Fluxo de Requisitos

1. Admin abre chat do ticket ou conversa direta com motorista/usuário.
2. Envia mensagens, solicita evidências e aplica ações.
3. Encerrado o atendimento, registra resolução e tempo de atendimento.

#### 3.9.3.5.1. Subfluxos

- **SUB1 — Templates de resposta:** respostas rápidas para casos comuns.

#### 3.9.3.5.2. Fluxos Alternativos

- **FA1 — Escala para gestor:** caso crítico, chat é marcado e escalado.

### **Pós-condições**

- Conversa arquivada e vinculada ao ticket.

### **Regras de negócio**

- SLA de atendimento deve ser medido e mantido.

#### **3.9.3.6. RF25 — Acompanhar estatísticas de avaliações**

Painel com métricas de avaliação de motoristas (média, tendências, flags).

### **Pré-condições**

- Dados de avaliações disponíveis.

### **Fluxo de Requisitos**

1. Admin acessa dashboard de avaliações.
2. Visualiza motoristas com média baixa, histórico de reclamações e volume de avaliações.
3. Pode aplicar ações (treinamento, advertência, bloqueio).

#### **3.9.4.6.1. Subfluxos**

- **SUB1 — Alertas automáticos:** motoristas com média  $< X$  por período geram alerta automático.

#### **3.9.4.6.2. Fluxos Alternativos**

- **FA1 — Contestação de avaliação:** motoristas podem solicitar revisão; admin avalia.

### **Pós-condições**

- Ações tomadas e registradas.

### **Regras de negócio**

- Sistema de peso para avaliações (mais recentes podem ter maior peso).

### **3.9.4. Aplicativo 1 - Requisitos do Gestor**

#### **3.9.4.1. RF26— Validar motoristas (Análise documental)**

Gestor realiza análise documental profunda e decide aprovação/suspensão.

##### **Pré-condições**

- Cadastro enviado (RF11).

##### **Fluxo de Requisitos**

1. Gestor abre fila de cadastros pendentes.
2. Realiza checagens externas se necessário (ex.: checar antecedentes em base pública).
3. Aprova ou rejeita com justificativa.
4. Decide restrições (ex.: operar somente em determinadas regiões) se aplicável.

##### **3.9.4.1.1. Subfluxos**

- **SUB1 — Solicitar auditoria:** em caso de inconsistência, pedir suporte jurídico.

##### **3.9.4.1.2. Fluxos Alternativos**

- **FA1 — Aprovação por exceção:** gestor pode aprovar temporariamente sob condição (ex.: reavaliação em 30 dias).

##### **Pós-condições**

- Decisão gravada e comunicado ao motorista.

##### **Regras de negócio**

- Registrar data e responsável pela aprovação (compliance).

#### **3.9.4.2. RF27 — Suspende / bloquear motoristas ou usuários denunciados**

Poder de ação para suspender/ bloquear contas conforme investigações e políticas.

##### **Pré-condições**

- Ticket de denúncia ou investigação concluída.



## Fluxo de Requisitos

1. Gestor acessa caso e determina sanção (advertência, suspensão temporária, bloqueio permanente).
2. Executa ação no sistema; notifica partes com justificativa.
3. Define prazo de suspensão ou condições de reativação.

### 3.9.4.2.1. Subfluxos

- **SUB1 — Suspensão administrativa provisória:** bloqueio enquanto investiga.

### 3.9.4.2.2. Fluxos Alternativos

- **FA1 — Apelação:** usuário/motorista pode solicitar revisão; gestor reavalia.

## Pós-condições

- Conta atualizada e ações registradas.

## Regras de negócio

- Sanções baseadas em política de conduta e documentação do caso.

### 3.9.4.3. RF28 — Configurar tarifas dinâmicas

Gestor define regras de tarifação (preço por km, taxa mínima, adições por porte, horário de pico).

## Pré-condições

- Gestor autenticado com permissão de preço.

## Fluxo de Requisitos

1. Gestor acessa módulo de tarifas.
2. Define parâmetros: tarifa base, R\$/km, taxa por porte, taxa mínima, multiplicador de pico.
3. Salva regras; sistema aplica em tempo real para cálculo de estimativas.
4. Versões anteriores armazenadas (audit trail).

#### **3.9.5.3.1. Subfluxos**

- **SUB1 — Tarifas por região:** definir zonas com preços diferenciados.
- **SUB2 — Promoções temporárias:** ligar/desligar promoções (RF29).

#### **3.9.5.3.2. Fluxos Alternativos**

- **FA1 — Teste de tarifa:** simular cenários antes de aplicar.

#### **Pós-condições**

- Novas regras ativadas e aplicadas em cálculos.

#### **Regras de negócio**

- Mudanças de tarifa comunicadas com antecedência e registradas.

#### **3.9.5.4. RF29 — Criar campanhas prom. e cupons de desconto**

Gestor cria e gerencia campanhas promocionais para atrair usuários.

#### **Pré-condições**

- Gestor com permissão de marketing.

#### **Fluxo de Requisitos**

1. Gestor acessa módulo de promoções.
2. Define cupom (código, desconto fixo/percentual, validade, regras de elegibilidade).
3. Define público alvo (ex.: novos usuários, região X).
4. Ativa campanha; sistema aplica desconto em cálculo de pagamento.

#### **3.9.4.4.1. Subfluxos**

- **SUB1 — Relatório de campanha:** métricas de uso e ROI.

#### **3.9.4.4.2. Fluxos Alternativos**

- **FA1 — Cancelar campanha:** encerrar antes da validade.

### **Pós-condições**

- Campanha ativa e métricas sendo coletadas.

### **Regras de negócio**

- Limites de uso por usuário e políticas de cumulatividade.

#### **3.9.4.5. RF30 — Acessar relatórios estratégicos**

Gestor obtém relatórios estratégicos (receita, churn, KPIs regionais).

### **Pré-condições**

- Gestor autenticado.

### **Fluxo de Requisitos**

1. Gestor seleciona métricas e período.
2. Sistema gera dashboards interativos (gráficos, tabelas).
3. Possibilidade de exportar e agendar envios.

#### **3.9.4.5.1. Subfluxos**

- **SUB1 — Drill-down:** detalhamento por região/motorista.

#### **3.9.4.5.2. Fluxos Alternativos**

- **FA1 — Dados inconsistentes:** alertar equipe de dados.

### **Pós-condições**

- Insights obtidos e decisões tomadas.

### **Regras de negócio**

- Controle de acesso e logs; relatórios financeiros possuem camada extra de segurança.

#### **3.9.4.6. RF31 — Acessar dashboards com previsões de demanda**

Dashboards com previsões (machine learning / heurísticas) por região e horário.

##### **Pré-condições**

- Modelos de previsão treinados e integrados ao sistema.

##### **Fluxo de Requisitos**

1. Gestor abre painel de previsão.
2. Visualiza heatmaps por região e previsão de demanda por hora/dia.
3. Utiliza dados para ajustar tarifas, ofertas e alocação de motoristas.

##### **3.9.4.6.1. Subfluxos**

- **SUB1 — Simulação de cenários:** testar impacto de ajuste de tarifas.

##### **3.9.4.6.2. Fluxos Alternativos**

- **FA1 — Modelo indisponível:** fallback para heurística simples (média histórica).

##### **Pós-condições**

- Ajustes operacionais realizados com base nas previsões.

##### **Regras de negócio**

- Periodicidade de retraining do modelo e validação de acurácia.

## 4. Projeto

### 4.1. Arquitetura Lógica

Este capítulo especifica a arquitetura do sistema de transporte especializado para animais de estimação, detalhando sua composição lógica e física, além das justificativas para a escolha dos componentes, garantindo escalabilidade, manutenção facilitada e atendimento às necessidades do usuário.

#### 4.1.1 Arquitetura de Software (Lógica)

A arquitetura do sistema Bicho Express é baseada no modelo de microserviços, que divide o sistema em componentes independentes, cada um responsável por funcionalidades específicas, facilitando a manutenção, escalabilidade e a incorporação de novas funcionalidades futuras. Essa abordagem modular permite uma comunicação eficiente entre os componentes via APIs RESTful, favorecendo a interoperabilidade entre o aplicativo móvel (usuário e motorista), portal administrativo e serviços de terceiros (como APIs de mapas, pagamentos e notificações).

Cada microserviço é responsável por um domínio do sistema, o que garante que o impacto de mudanças em uma parte do sistema seja isolado. Isso facilita atualizações contínuas adotando a metodologia ágil Scrum, com entregas incrementais.

#### 4.1.2. Layers do Sistema

As camadas do sistema foram organizadas para distribuir funcionalidades específicas de acordo com o tipo de usuário, garantindo clareza lógica e eficiência na implementação:

**Login e Autenticação:** Controle do cadastro, login via e-mail, Google e autenticação multifator, além da gestão de sessões e segurança.

**Cadastro e Perfil:** Gestão de perfis dos usuários e pets, incluindo detalhes como raça, porte, vacinas e necessidades especiais.

**Solicitação e Agendamento:** Interface para solicitar corridas imediatas ou agendadas, tipo de veículo adaptado, cálculo e otimização de rotas.

**Geolocalização e Rastreamento:** Integração com APIs de mapas para localização em tempo real de motoristas e acompanhamento da corrida.

**Pagamento:** Processamento de pagamentos via Pix, cartões e carteira digital, assegurando transparência e segurança.

**Avaliações e Feedback:** Registro e exibição de avaliações entre usuários e motoristas para estimular confiança e qualidade.

**Histórico de Corridas:** Gerenciamento do status das corridas, pagamentos e cancelamentos.

**Painel Administrativo:** Gestão de motoristas, corridas, faturamento, suporte e relatórios para o setor administrativo.

**Notificações e Comunicação:** Sistema de mensagens push, chat criptografado e alertas para manter usuários informados.

**Segurança e Conformidade:** Tratamento seguro dos dados, conformidade com LGPD e normas ISO/IEC 27001, além de protocolos de bem-estar animal.

#### **4.1.3. Justificativa da Arquitetura**

A escolha da arquitetura baseada em microserviços permite:

- Escalabilidade horizontal, com adição de serviços conforme crescimento do sistema;
- Independência entre módulos, facilitando atualizações e manutenção sem impactar toda a aplicação;
- Implementação ágil, alinhada à metodologia Scrum, com entregas incrementais e feedback contínuo;
- Integração com APIs externas (pagamento, mapas, notificações) sem comprometer a estabilidade do sistema;
- Modularidade, permitindo futuras implementações de IoT ou novos serviços relacionados ao transporte pet.

#### 4.1.4. Tier

O **Tier** é o modelo de arquitetura que realiza a separação do aplicativo em partes, combinando conceitos físicos e lógicos. Essa divisão permite que diferentes equipes trabalhem de forma independente em cada camada, garantindo que alterações em uma camada não causem impacto nas demais.

A estrutura física e lógica do sistema segue o modelo **N-Tier**, permitindo divisão e administração independentes das camadas, aumentando a flexibilidade e eficiência:

**Apresentação:** camada visível ao usuário, responsável por coletar informações e interagir diretamente com o aplicativo.

**Aplicativo:** camada responsável por processar as informações recebidas da camada de apresentação, realizando lógica de negócio e regras de transporte.

**Banco de dados:** camada onde os dados são armazenados em um servidor NoSQL (Firebase), garantindo persistência e segurança das informações.

**Cache:** banco de memória temporário usado para armazenar informações acessadas com frequência, acelerando o acesso sem necessidade de consultar o banco de dados principal. Deve ser utilizado para itens que não requerem atualizações constantes, como avaliações, portfólios profissionais, lista de serviços, mensagens automáticas e FAQ.

**API Gateway:** camada que gerencia a comunicação entre o aplicativo principal e outras plataformas, controlando o tráfego de informações externas.

#### 4.1.5. Armazenamento Cloud Computer

Para garantir alta disponibilidade, segurança e escalabilidade, o sistema utilizará a infraestrutura de **cloud computing AWS (Amazon Web Services)**, escolhida pela sua robustez e serviços inovadores:

Utilização do **AWS EC2**, com instância configurada para alta capacidade de memória e processamento, capaz de suportar apps móveis e backend, além do banco de dados Firebase.

A instância EC2 oferece recursos para hospedagem dos microserviços, com capacidade para múltiplas máquinas virtuais independentes, permitindo isolamento e escalabilidade planejada.

Configuração de alta velocidade de rede (clusters em zona de disponibilidade) para baixa latência e alta velocidade de acesso ao banco de dados e APIs.

Backups automáticos e políticas de recuperação acelerada em caso de falhas para garantir a segurança dos dados.

#### 4.1.6. Linguagem de Programação

Para acelerar o desenvolvimento e garantir compatibilidade multiplataforma, o sistema utilizará principalmente:

**JavaScript/TypeScript** para desenvolvimento do backend (Node.js) e frontend (React Native para apps móveis), garantindo código unificado e fácil manutenção.

Frameworks modernos que suportam desenvolvimento rápido, com vastas bibliotecas para integração com APIs externas e atendimento a requisitos de performance, segurança e usabilidade.

Uso de **React Native** para desenvolvimento multiplataforma (iOS e Android) para oferecer uma experiência de usuário consistente e otimizada.

### 4.2. Arquitetura Física

#### 4.2.1. Configurações de Hardware

O sistema Bicho Express será executado em uma infraestrutura de cloud computing, para garantir escalabilidade, alta disponibilidade e segurança para operações. A seguir está a configuração proposta:

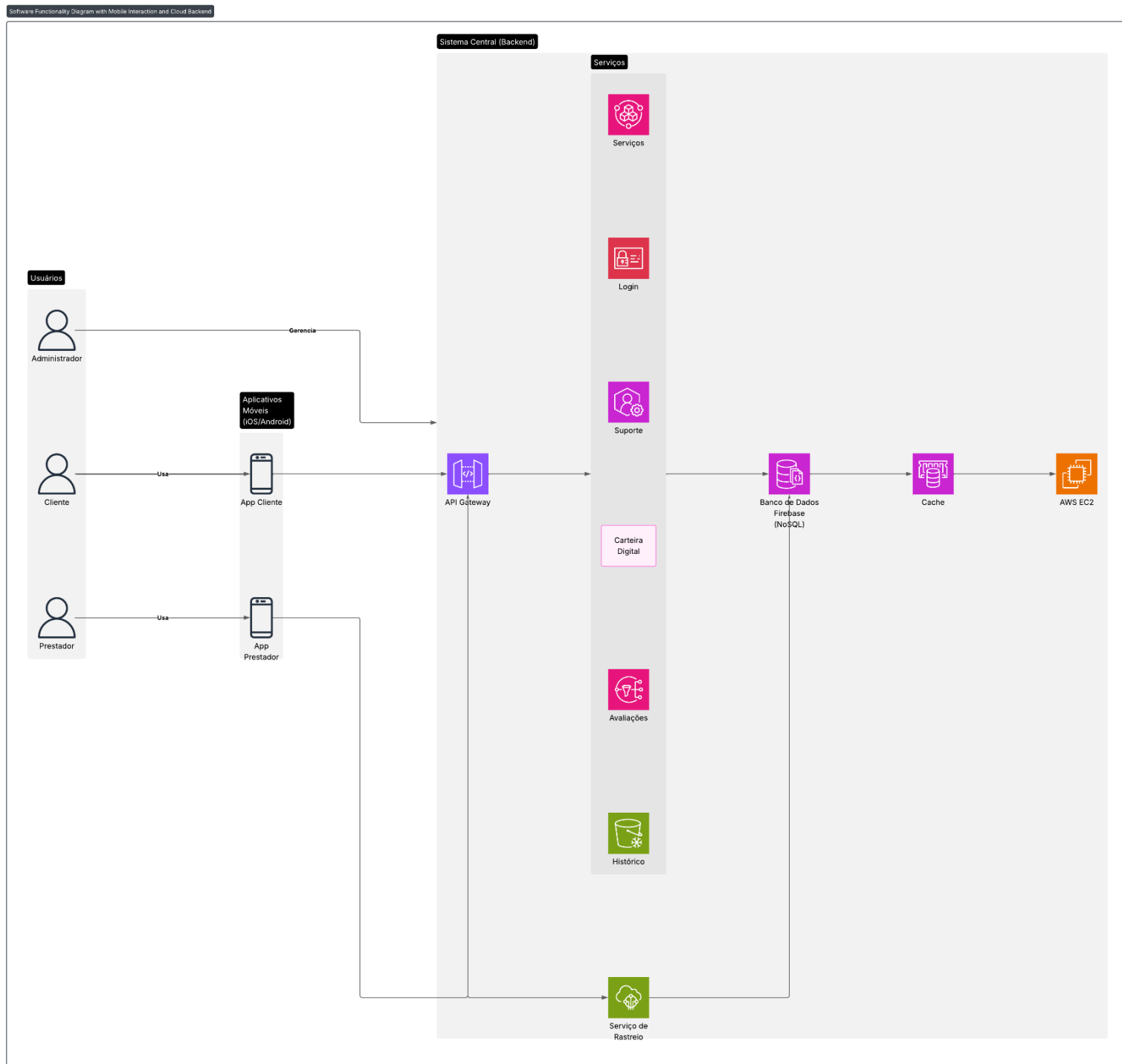
Tabela 4- Amazon EC2- t3.large

Item	Unidade / Fórmula	Unit	Qtde	Custo mensal (USD)	Custo anual (USD)	Custo mensal (BRL)	Custo anual (BRL)
EC2 (t3.large) — Linux	\$0.0835/h × 730h	\$ 0,08	730h	\$ 60,96	\$ 731,52	R\$ 327,96	R\$ 3.935,58
EC2 (t3.large) — Windows (estim.)	\$0.10855/h × 730h	\$ 0,11	730h	\$ 79,24	\$ 950,88	R\$ 426,31	R\$ 5.115,73
Armazenamento EBS (gp3)	100 GB × \$0.10/GB-mês	\$ 0,10	100 GB	\$ 10,00	\$ 120,00	R\$ 53,80	R\$ 645,60
Snapshots EBS	30 GB × \$0.05/GB-mês	\$ 0,05	30 GB	\$ 1,50	\$ 18,00	R\$ 8,07	R\$ 96,84
Data transfer out	200 GB × \$0.12/GB	\$ 0,12	200 GB	\$ 24,00	\$ 288,00	R\$ 129,12	R\$ 1.549,44
NAT Gateway	0.045/h × 730h + \$0.045/GB × 200GB	\$ 0,05	730h + 200GB	\$ 41,85	\$ 502,20	R\$ 225,15	R\$ 2.701,84
ALB (Load Balancer)	\$0.0225/h + LCU \$0.008 × 730h	\$ 0,02	730h	\$ 22,27	\$ 267,24	R\$ 119,81	R\$ 1.437,75
RDS (db small - estim.)	\$0.05/h × 730h	\$ 0,05	730h	\$ 36,50	\$ 438,00	R\$ 196,37	R\$ 2.356,44
Monitoramento / CloudWatch	Pacote básico	\$ 5,00	1	\$ 5,00	\$ 60,00	R\$ 26,90	R\$ 322,80
TOTAL (Linux stack)				\$ 281,32	\$ 3.375,84	R\$ 1.513,50	R\$ 18.162,02

**OBS:** A infraestrutura poderá ser dimensionada para instâncias maiores ou adicionais conforme o crescimento do sistema.



## 4.2.2. Diagrama de Funcionamento de Hardware



(Figura 7 - Diagrama de funcionamento do Hardware, Fonte: Autores realizado no Lucidchart, 2025)

#### 4.2.3. Configurações de Rede

Para garantir a segurança e eficiência da rede usada pelo sistema, as seguintes configurações serão adotadas:

**AWS Virtual Private Cloud (VPC):** Para isolar a rede do sistema do tráfego público, garantindo segurança e controle do tráfego interno.

**Grupos de segurança (Security Groups):** Definição restrita de portas e protocolos permitidos para comunicação, minimizando riscos externos.

**Balanceador de Carga (Load Balancer):** Distribui as requisições dos usuários entre as instâncias de backend automaticamente, assegurando alta disponibilidade e escalabilidade.

**Cluster de Posicionamento:** Para despesas críticas, como banco de dados, serviços que demandam alta velocidade e baixa latência utilizam clusters na mesma zona de disponibilidade, reduzindo latência e aumentando a largura de banda interna da AWS.

**Protocolos Seguros:** Comunicação entre componentes do sistema e usuários garante HTTPS com TLS/SSL, protegendo a troca de dados.

**Redes de Distribuição de Conteúdo (CDN):** Para entrega rápida de conteúdo estático, como imagens, ícones e recursos do aplicativo no front-end.

#### 4.2.4. Topologia lógica (resumo do fluxo)

- Usuário (app móvel iOS/Android) → HTTPS → CloudFront (opcional CDN) → DNS (Route 53) → ALB / API Gateway → API (microserviços em containers/EC2) → Firebase (dados), S3 (arquivos), Redis/ElastiCache (cache/sessões) → Serviços externos (Payments, Google Maps, FCM).
- Administração / Painel Web → VPN / Bastion → Console/Admin UI → Mesma API interna.

Logs & monitoramento: CloudWatch / OpenSearch / S3. Backups e casnapshots automáticos.

#### 4.2.5. Inventário de componentes e configuração recomendada

Componente	Função	Recomendação / Especificação	Observações
VPC	Rede privada	VPC com 3 sub-redes públicas + 6 sub-redes privadas (3 AZs ideal)	NACLs, roteamento entre subnets
Subnets	Isolamento	Public (ALB, NAT, Bastion) + Private (microserviços, cache, jobs)	Alta disponibilidade via múltiplas AZs
ALB	Balanceamento HTTP(S)	Application Load Balancer (HTTPS 443) com ACM (certificado)	Terminação TLS no ALB
Auto Scaling	Escalabilidade	ASG mínimo 2 instâncias t3.large (MVP), escalonamento baseado em CPU/latência	Min 2 (AZ redundancy), Max sugerido 50 (veja cálculo)
EC2 (app)	Compute	t3.large (2 vCPU, 8GB) — para MVP; recomenda-se migrar para ECS Fargate ou EKS para containers	Em produção, prefira instâncias de família M ou C se CPU-bound
Containers	Isolamento	Docker + Node.js/Typescript microservices	Use ECR para imagens
Firebase	Banco de Dados	Firestore / Realtime Database (NoSQL)	Conexão segura via service account; exportações regulares p/ S3
S3	Armazenamento objetos	100 GB+ (arquivos, imagens, backup exports)	Ciclo de vida, versioning e cross-region replication (se DR)
EBS gp3	Root / storage local	100 GB gp3 por EC2	IOPS ajustável conforme necessidade
EFS (opcional)	Arquivos compartilhados	NFS compartilhado entre instâncias	Se necessário para uploads/processamento
ElastiCache (Redis)	Cache / sessão	2 nós (multi-AZ) — para rastreamento em tempo real / sessão / rate limiting	Diminui latência / carga no DB
NAT Gateway	Acesso saída	1 por AZ (redundância)	Custo por hora + GB

<b>Bastion / SSM</b>	Acesso admin	SSM Session Manager (recomendado) + bastion para emergency	Não abrir SSH ao mundo
<b>CloudFront</b>	CDN (opcional)	Distribuição para frontend estático e mapas cache	Reduz latência / custos da API
<b>Route53</b>	DNS	DNS primário, health checks	Failover e routing policies
<b>ACM</b>	Certificados TLS	Cert manager para ALB / CloudFront	TLS 1.2+
<b>WAF + Shield</b>	Segurança	Proteção contra OWASP / DDoS	Regras de IP blocking
<b>CloudWatch G OpenSearch</b>	Logs / métricas	Métricas + alarms + dashboard + OpenSearch para logs	Retenção 90 dias (logs)
<b>Backup</b>	Snapshot / export	EBS snapshots, S3 backups, Firestore export diário	Retenção 12 meses (RNF)
<b>IAM / KMS</b>	Segurança	IAM roles least privilege + KMS para at-rest encryption	Rotação de chaves, logs de uso
<b>CI/CD</b>	Deploy	GitHub Actions / CodePipeline + ECR + Terraform	Deploy automatizado, rollback
<b>Monitoring / APM</b>	Observability	Datadog / NewRelic / AWS X-Ray (opcional)	Tracing de requests

Tabela 5- Componentes físicos do software( Fonte, autores, 2025)

### 4.3. Diagramas de Classe

Representação visual do Funcionamento do Software

#### 4.3.1. Diagrama do Usuário/ Tutor de Pets

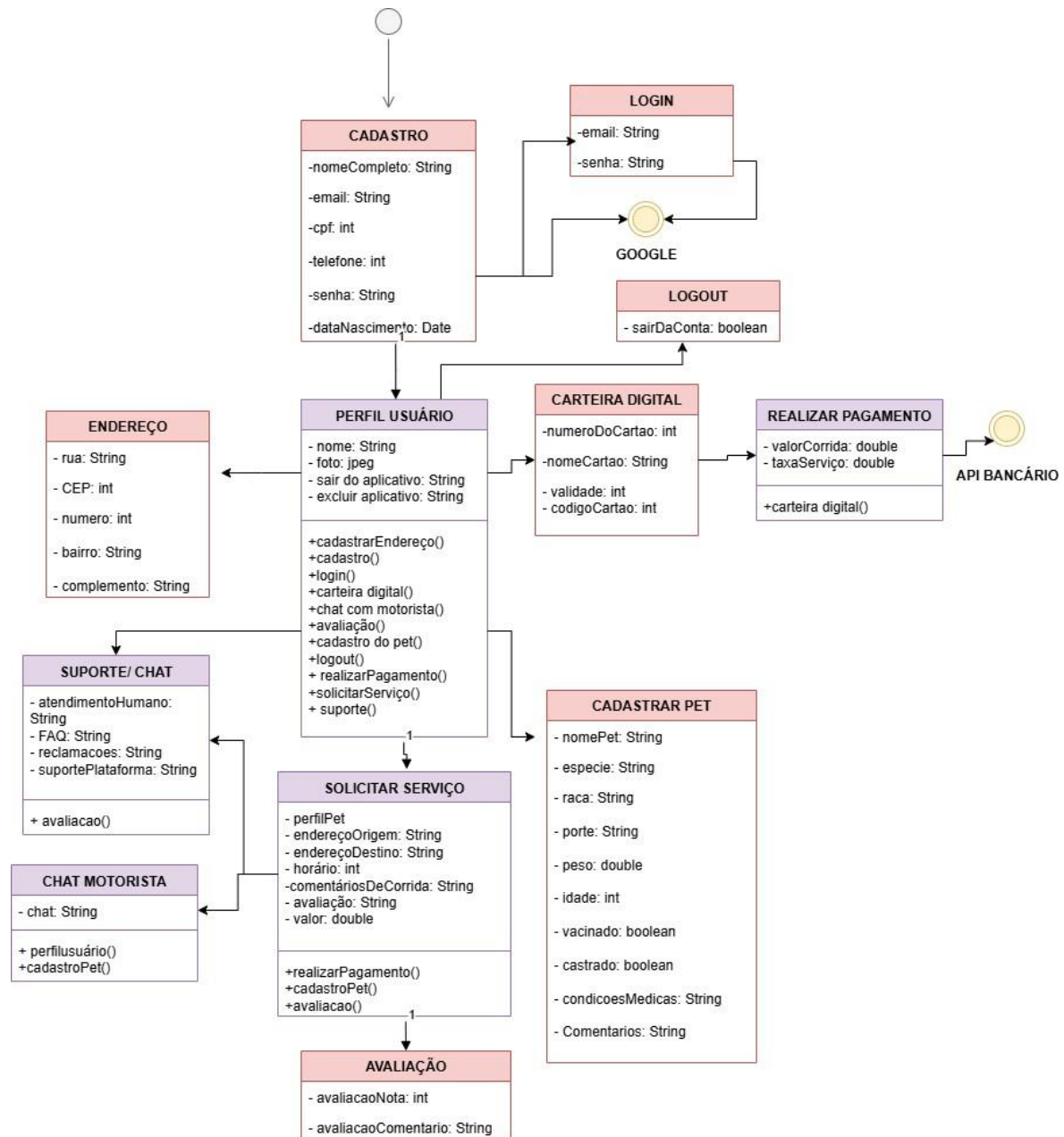


Figura 8- Diagrama de Classe Usuário/ Tutor de Pets  
(Fonte: Autores, Draw.io, 2025)

#### 4.3.2. Diagrama do Prestador de Serviços/ Motorista

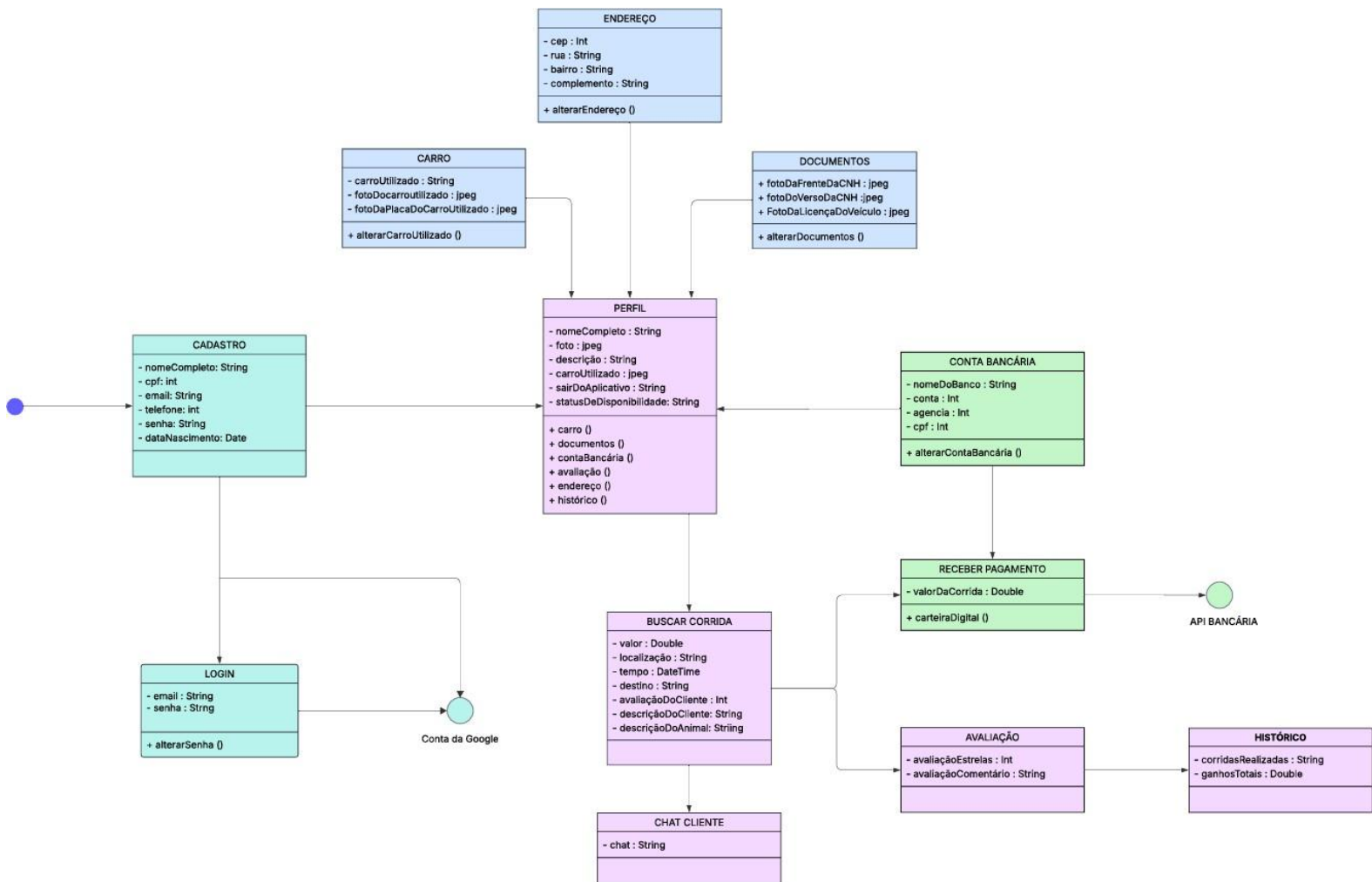


Figura 9- Diagrama de Classes Motorista/Prestador de Serviços  
(Fonte: Autores, Draw.io, 2025)

### 4.3.3. Diagrama de Classes Dos Administradores e Gestores

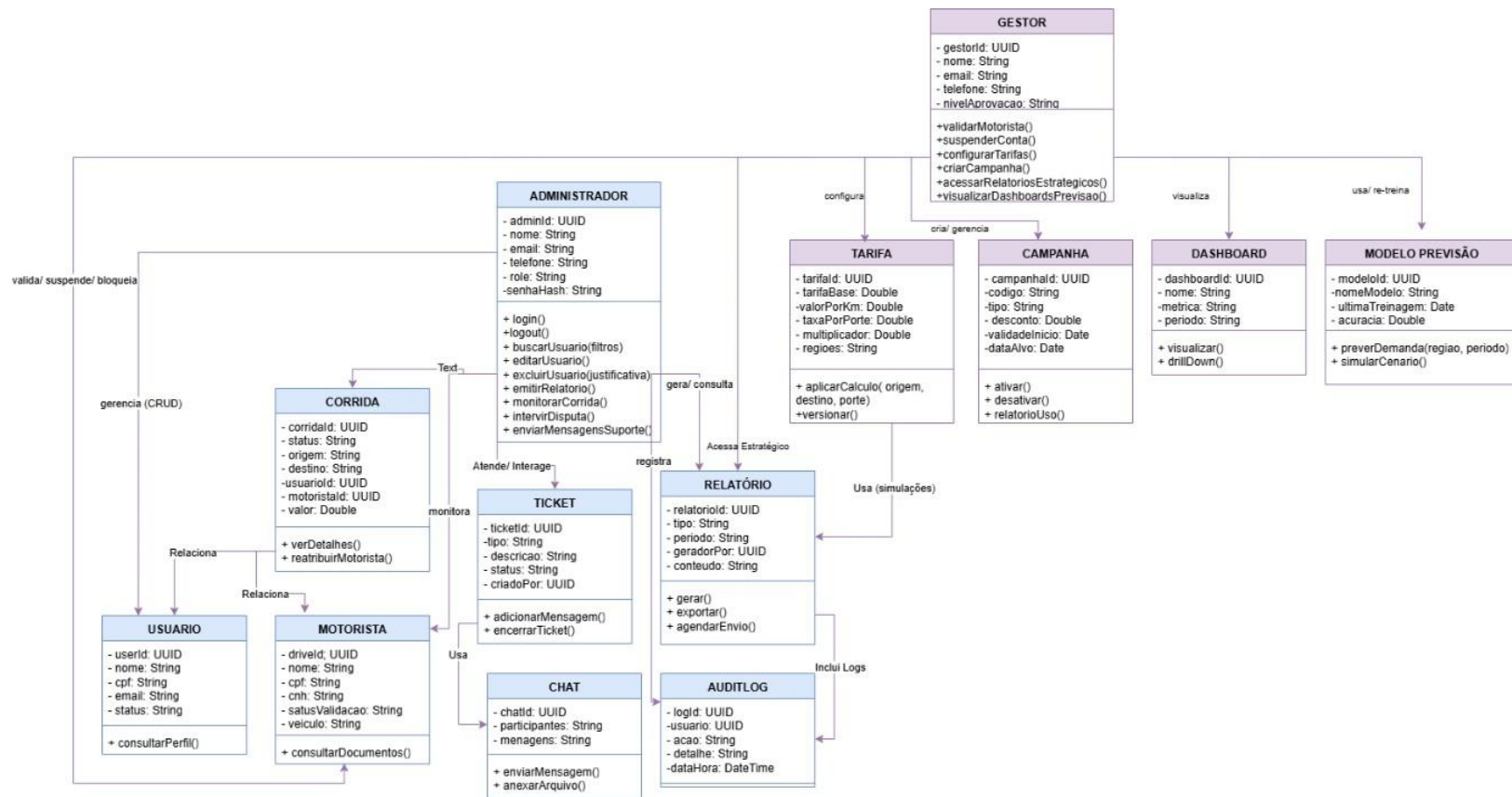


Figura 10- Diagrama de Classes Gestor e Setor Administrativo  
(Fonte: Autores, Draw.io, 2025)

#### 4.3.4. Diagrama de Atividade- Perfil do Pet

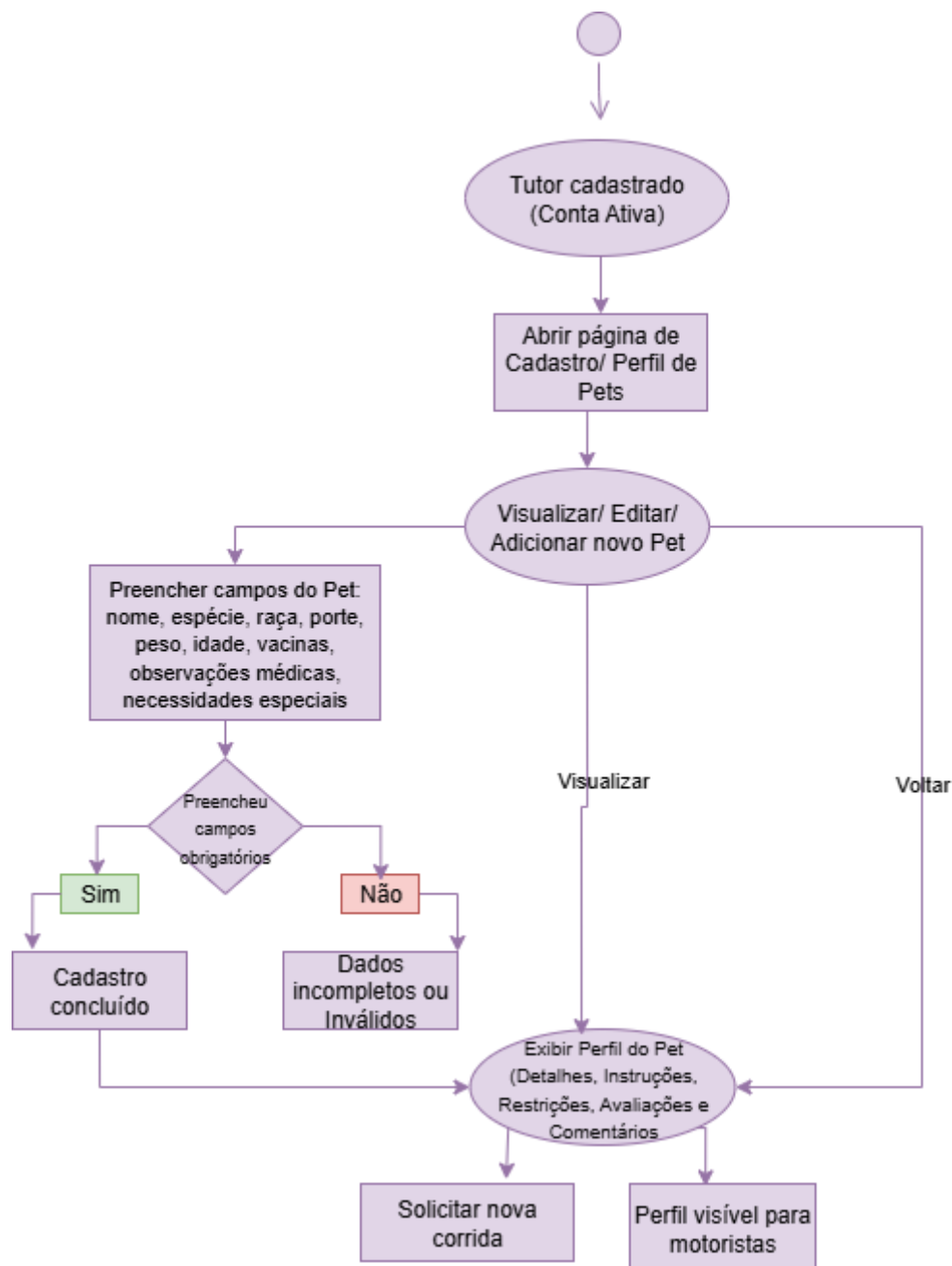


Figura 11- Diagrama de Atividade Perfil do Pet  
(Fonte: Autores, Draw.io, 2025)



#### 4.3.5. Diagrama de Atividade- Disponibilidade do Motorista

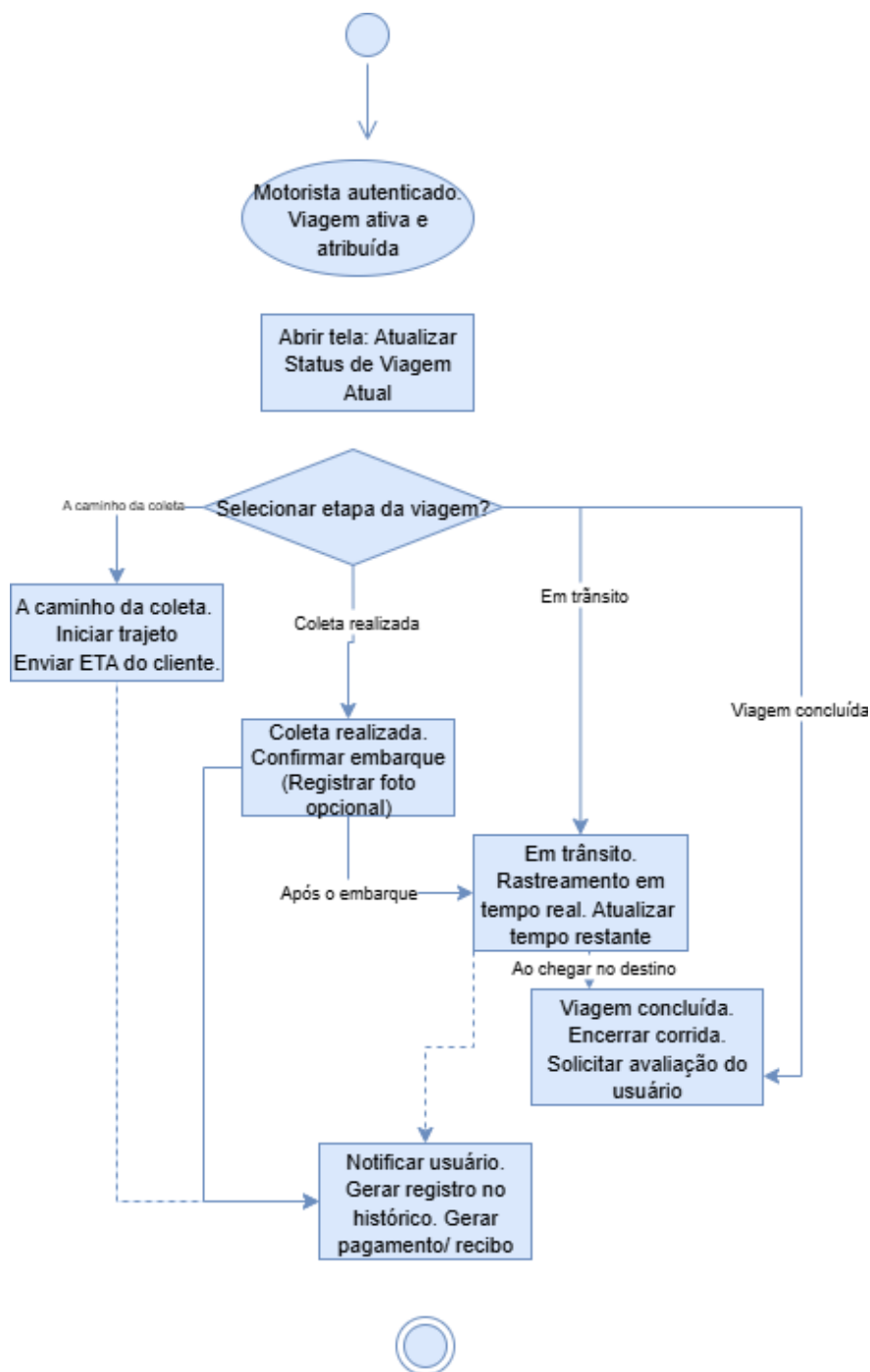


Figura 12- Diagrama de Atividade Definir disponibilidade do Motorista  
(Fonte: Autores, Draw.io, 2025)

#### 4.3.6. Diagrama de Atividade- Definir status da corrida atual

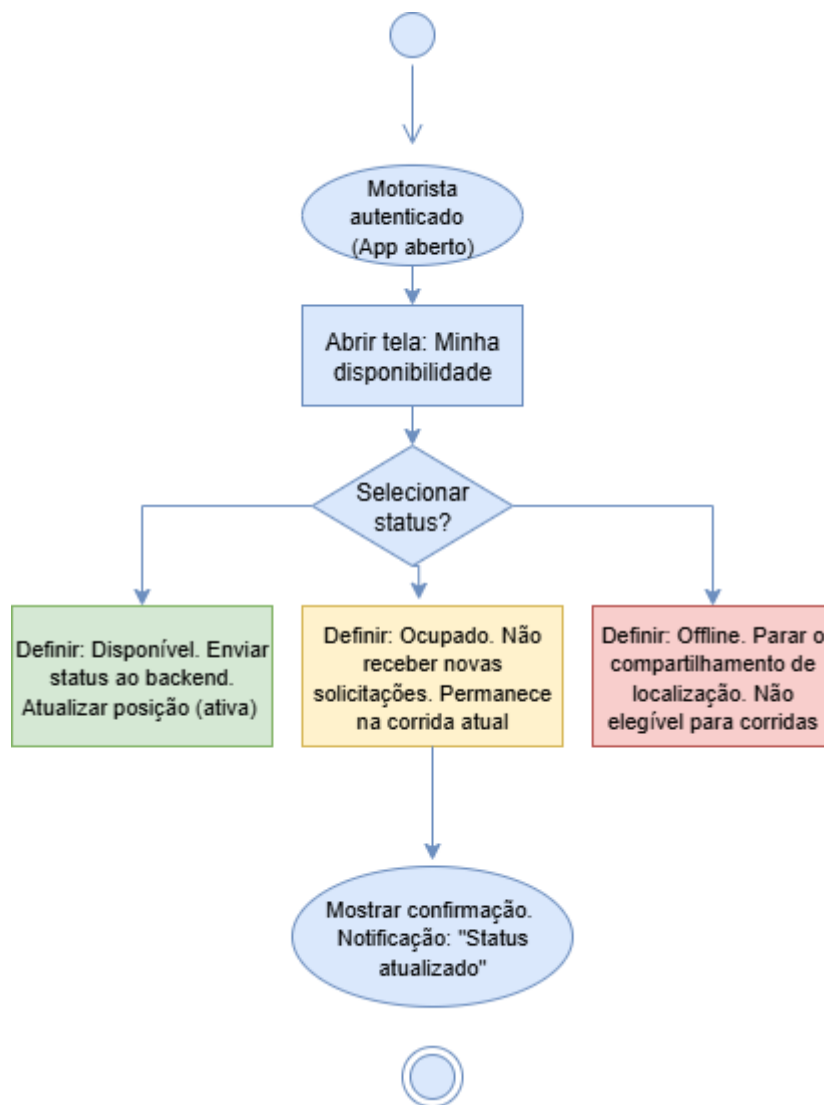


Figura 13- Diagrama de Atividade Definir Status da corrida atual  
(Fonte: Autores, Draw.io, 2025)

#### 4.3.7. Diagrama de Atividade- Avaliar Motorista/ Prestador de Serviços

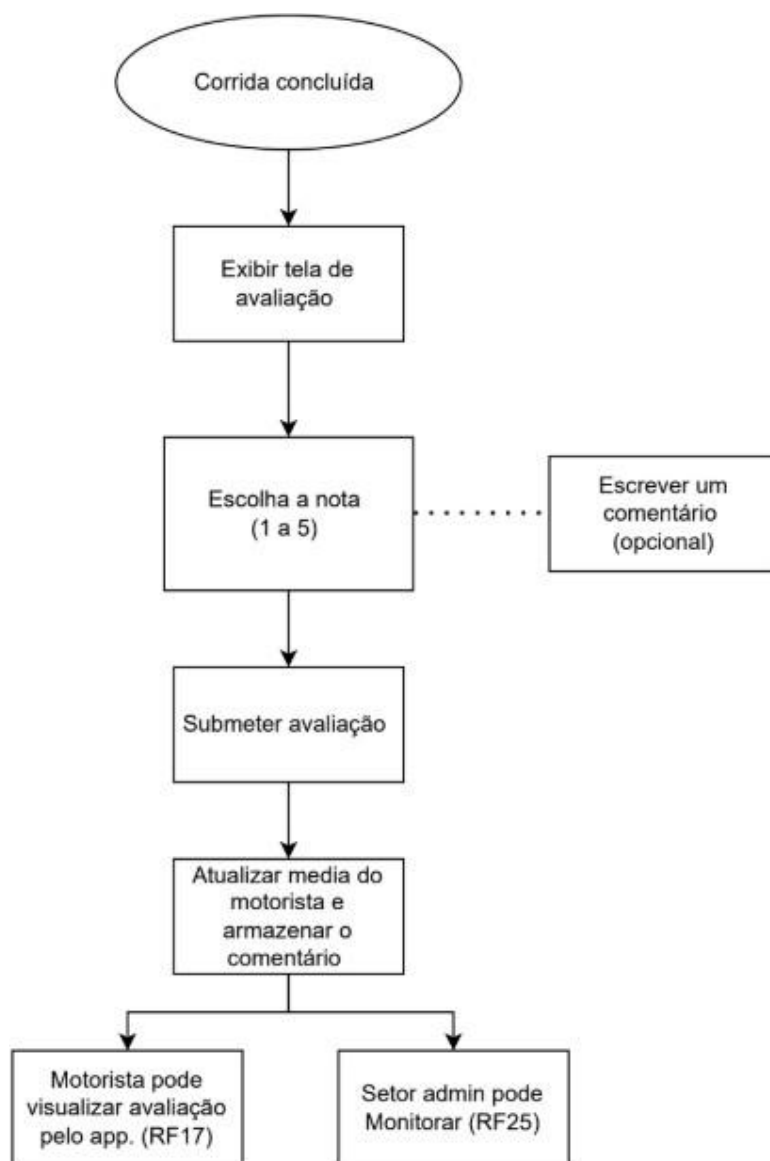


Figura 14- Diagrama de Atividade Avaliar Motorista  
(Fonte: Autores, Draw.io, 2025)

#### 4.3.8. Diagrama de Atividade- Avaliar Usuário

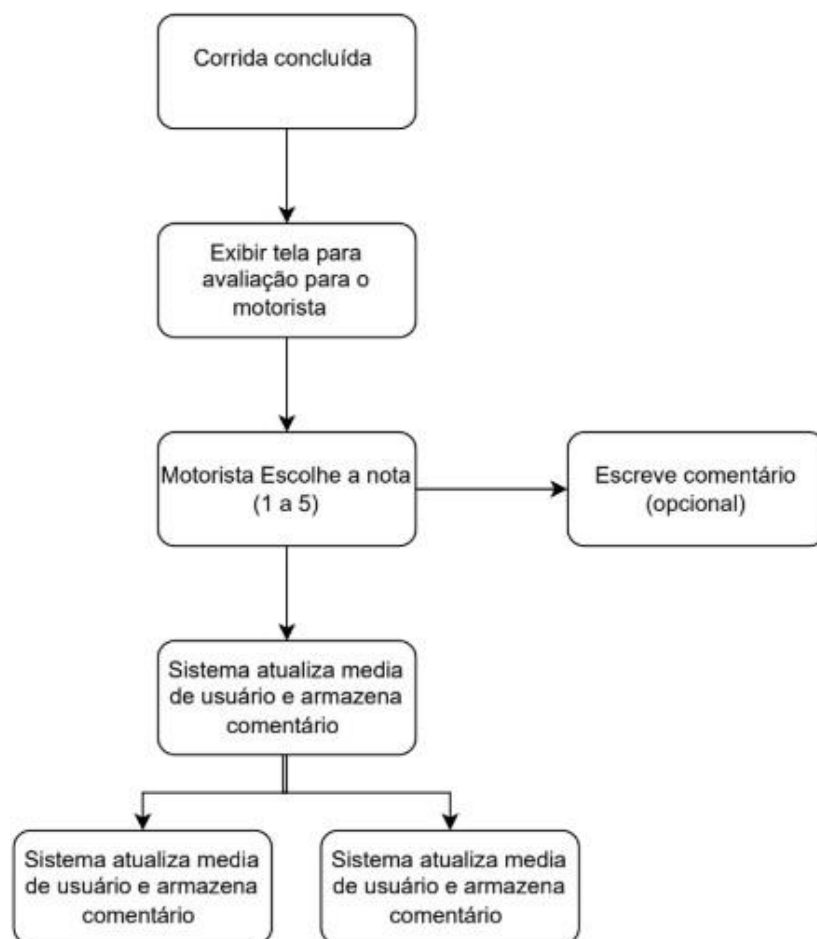


Figura 15- Diagrama de Atividade Avaliar Usuário  
(Fonte: Autores, Draw.io, 2025)

#### 4.3.9. Diagrama de Atividade- Perfil do Motorista

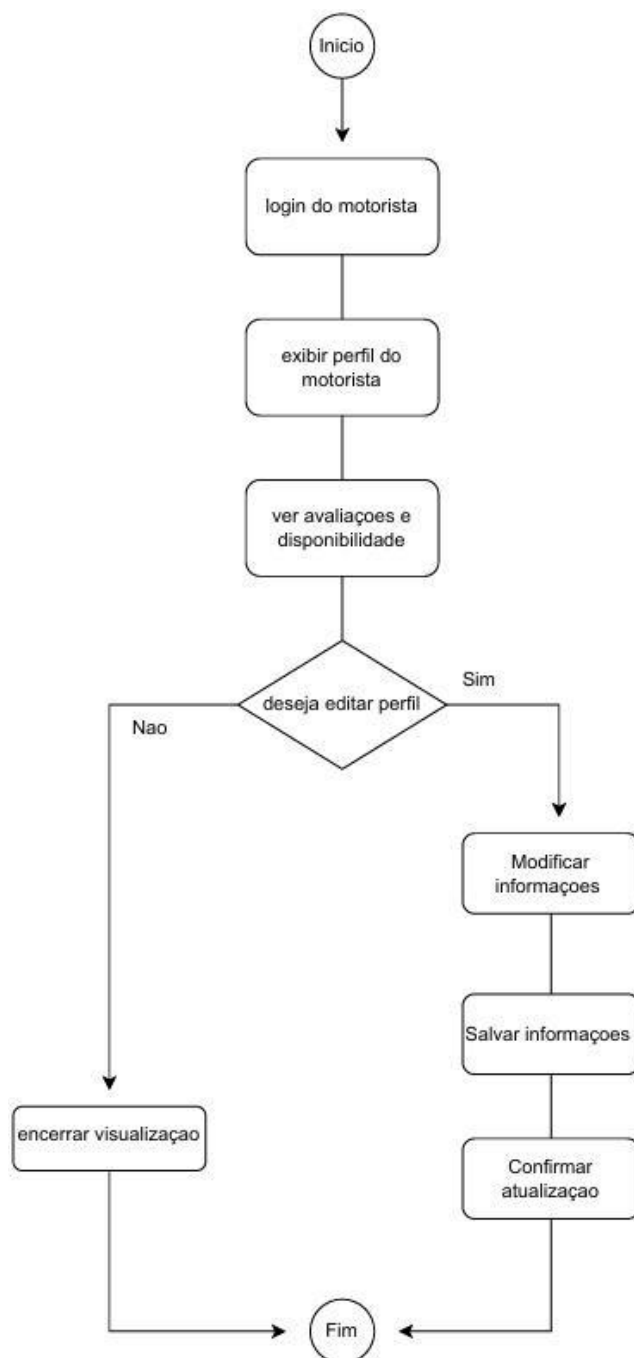


Figura 16- Diagrama de Atividade Perfil Motorista  
(Fonte: Autores, Draw.io, 2025)

#### 4.3.10. Diagrama de Atividade- Pagamento de Corridas

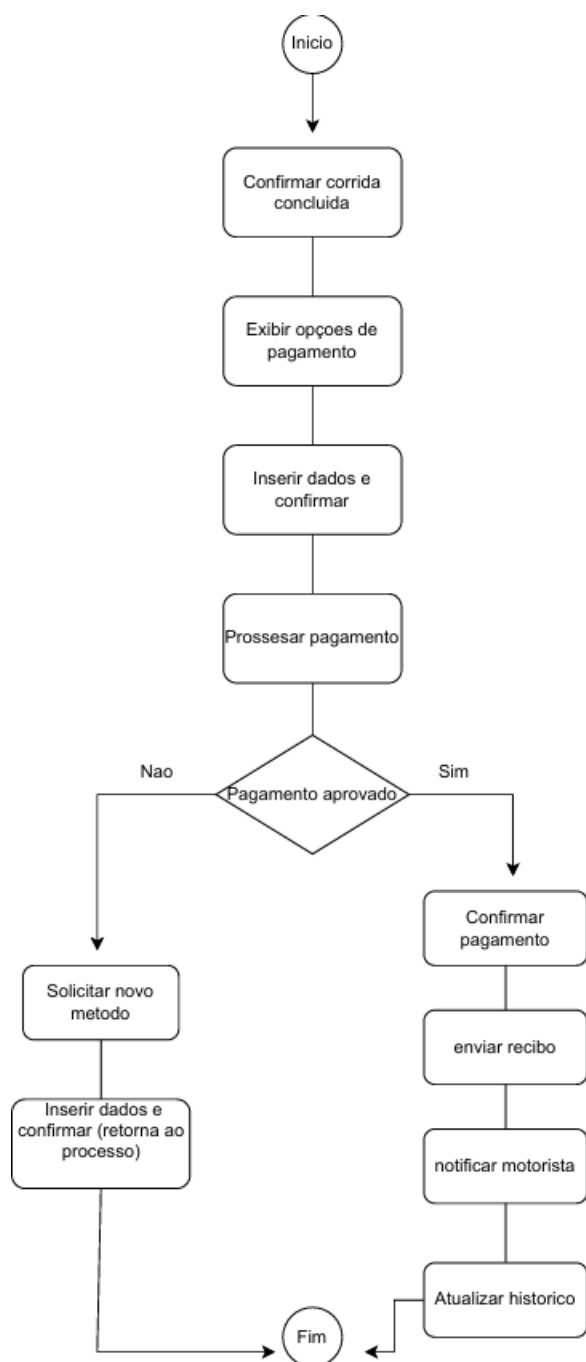


Figura 17- Diagrama de Atividade Pagamento das Corridas  
(Fonte: Autores, Draw.io, 2025)

#### 4.3.11. Diagrama de Atividade- Solicitar Corrida

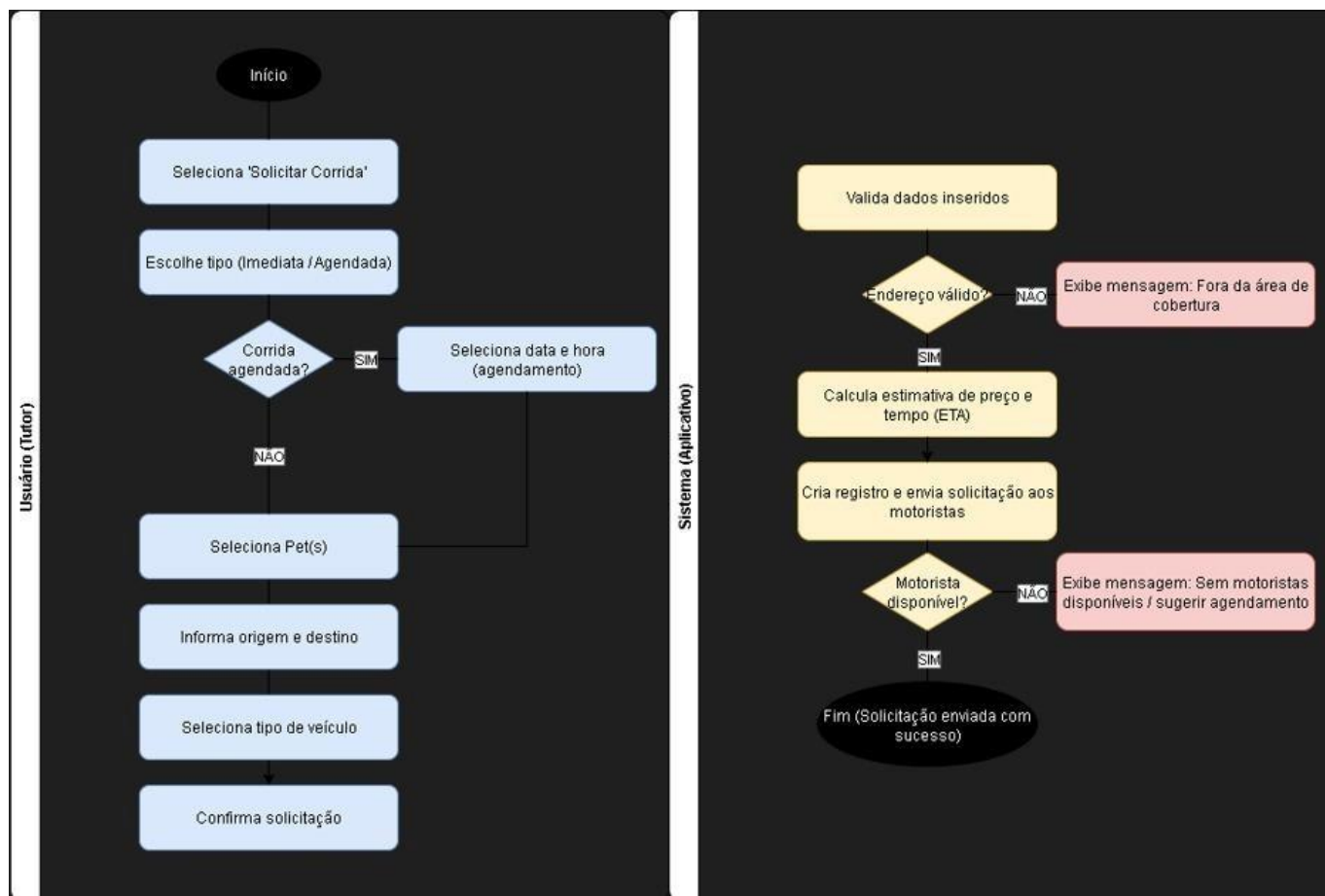


Figura 18- Diagrama de Atividade Solicitar Corrida  
(Fonte: Autores, Draw.io, 2025)

#### 4.3.12. Diagrama de Atividade- Cadastro do Motorista/ Prestador de Serviços

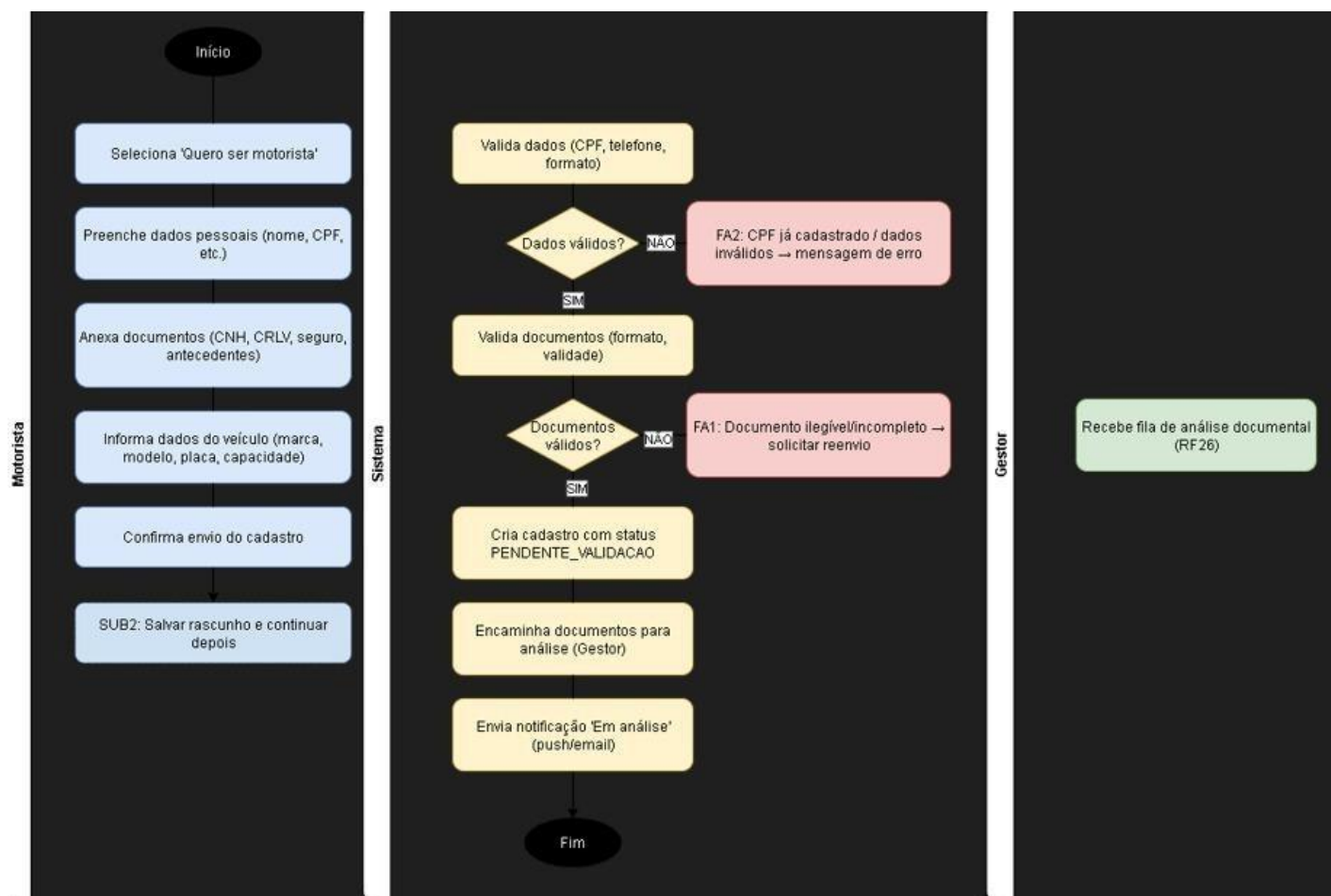


Figura 19- Diagrama de Atividade Cadastro do Motorista  
(Fonte: Autores, Draw.io, 2025)



#### 4.3.13. Diagrama de Atividade - Perfil de Usuário e Cadastro de Novo Pet

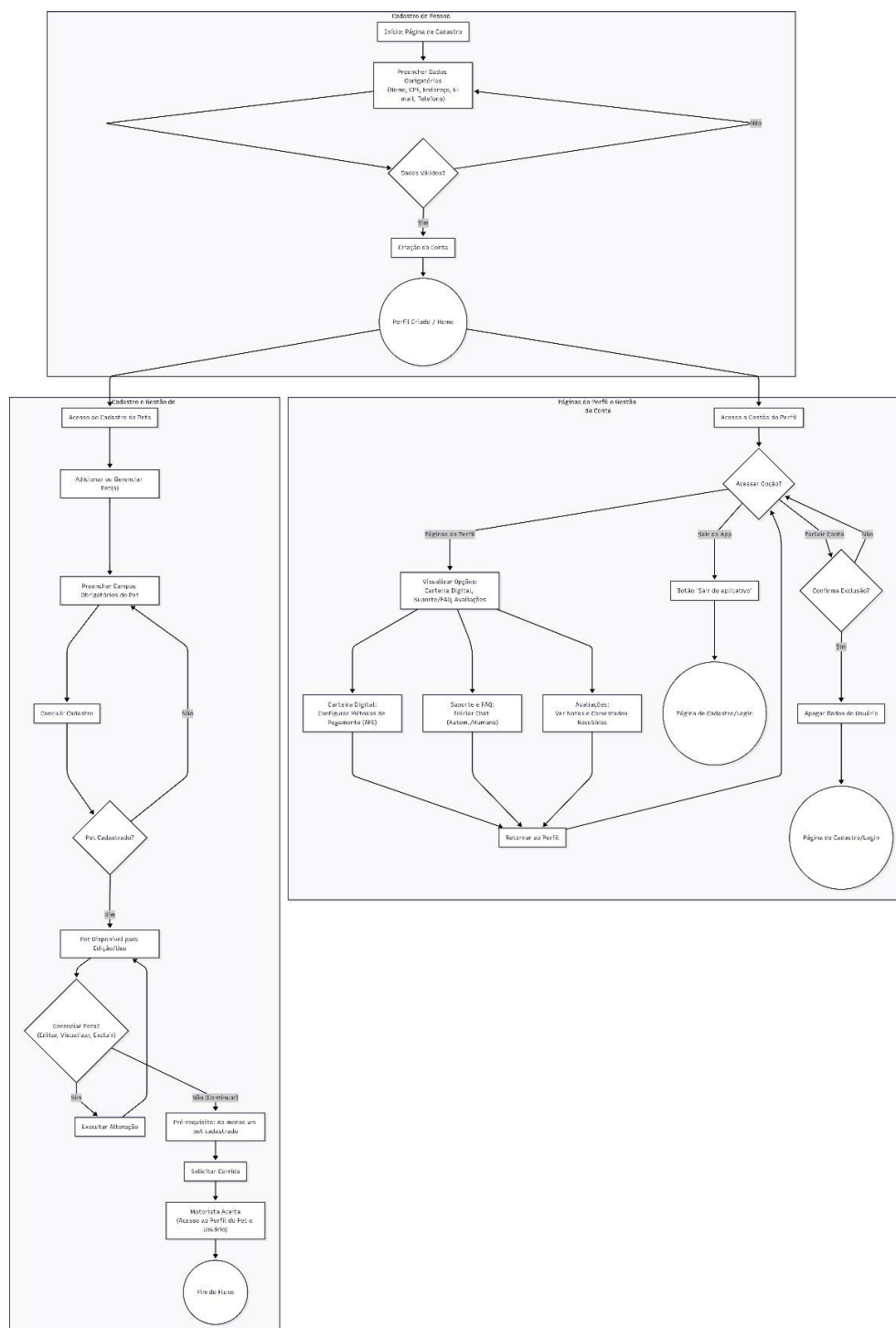


Figura 20- Diagrama de Atividade Cadastro do Motorista (Fonte: Autores, Draw.io, 2025)

#### 4.3.14. Diagrama de Atividade- Cadastro de Novo Usuário

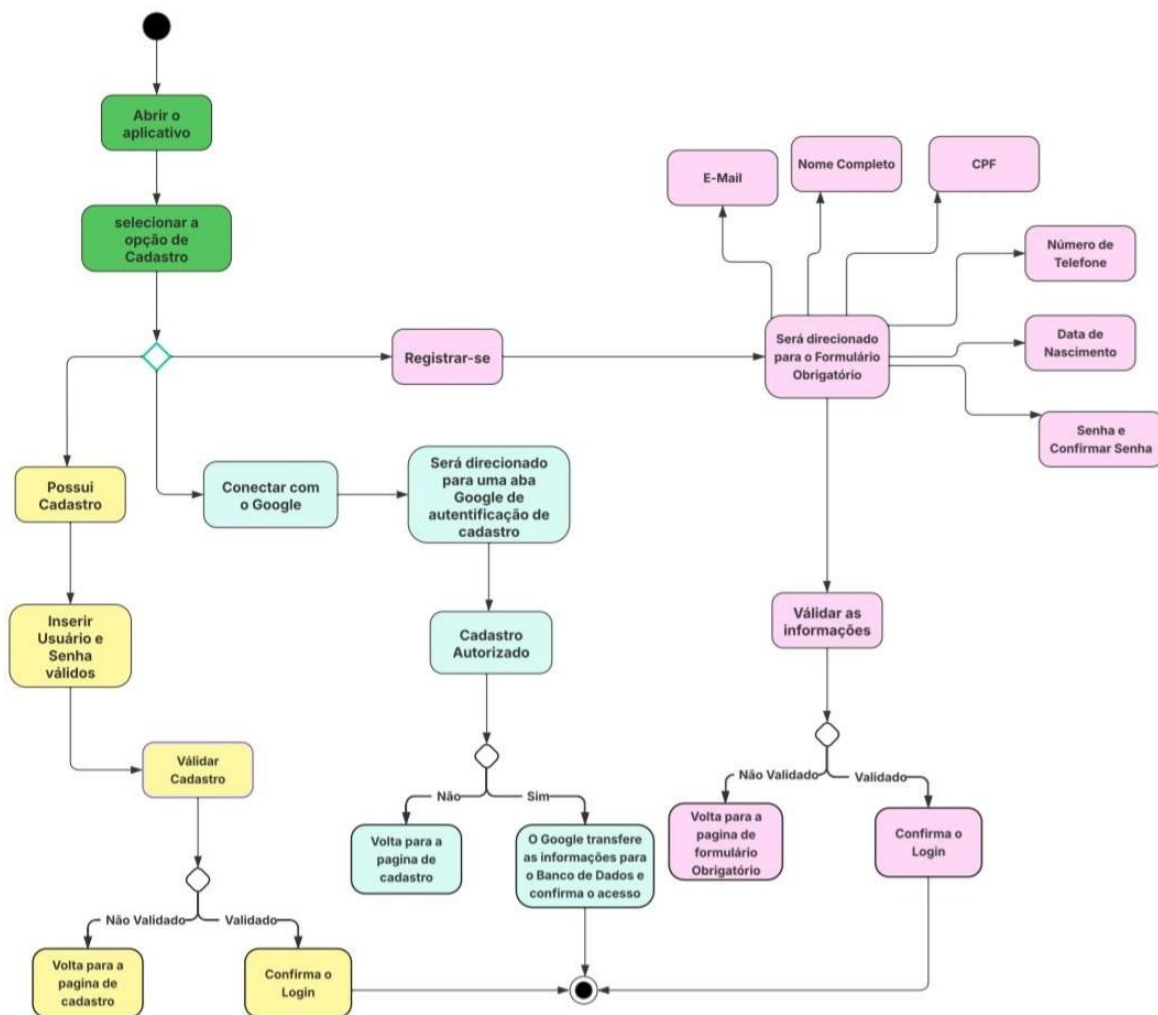


Figura 21- Diagrama de Atividade Cadastro de novo usuário (Fonte: Autores, Draw.io, 2025)

#### 4.3.15. Diagrama de Atividade- Histórico de Ganhos

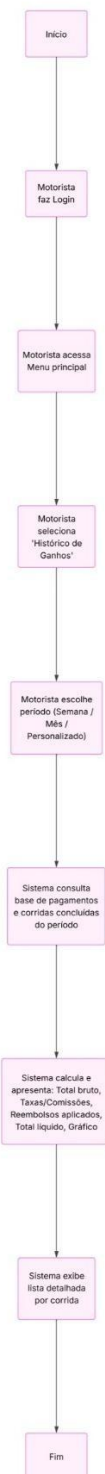


Figura 22- Diagrama de Atividade Histórico de Ganhos (Fonte: Autores, Draw.io, 2025)

#### 4.3.16. Diagrama de Atividade- Histórico de Corridas

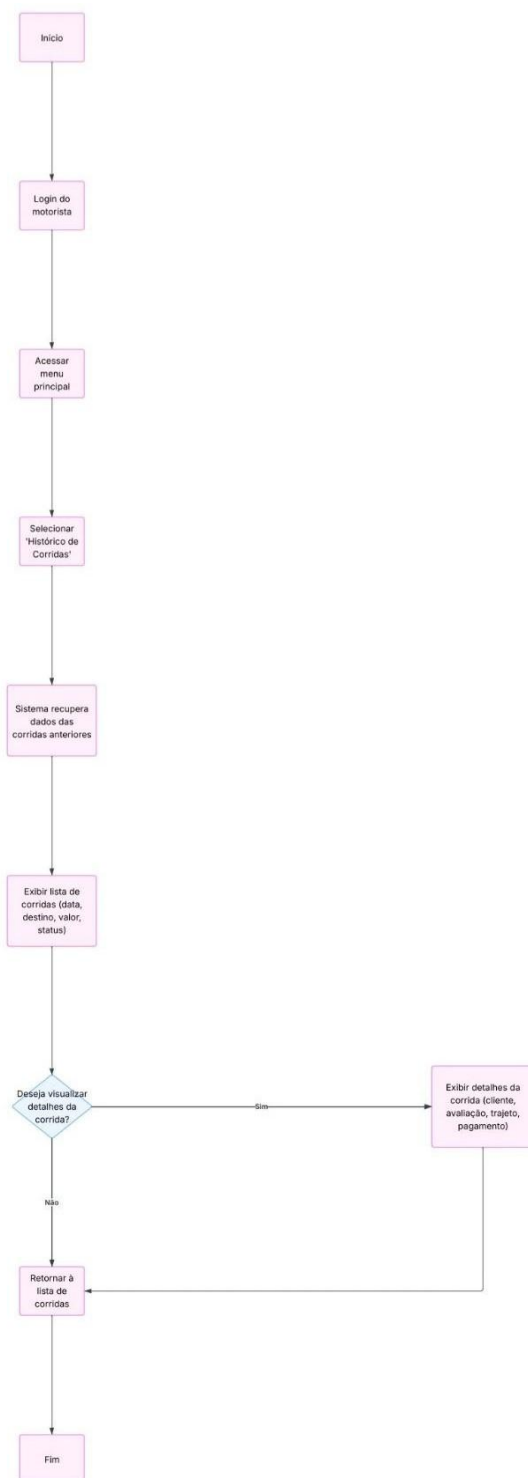


Figura 23- Diagrama de Atividade Histórico de Corridas (Fonte: Autores, Draw.io, 2025)

## 5. Protótipo de Interface

O presente capítulo é direcionado ao processo de prototipagem do sistema **Bicho Express**, um aplicativo e plataforma web voltado para o transporte seguro e confortável de animais de estimação. Nesta etapa, realizamos o ciclo de levantamento, detalhamento, validação e modelagem, com base em requisitos de usuários, motoristas, gestores e setor administrativo. O protótipo foi desenvolvido no Figma, simulando todas as principais telas e fluxos do aplicativo.

### 5.1 Fase de Levantamento

O levantamento constitui a etapa inicial de construção do software, englobando coleta de informações sobre as dificuldades dos tutores de pets e motoristas, bem como análise de mercado sobre soluções similares. Nesse processo, o protótipo foi concebido para mapear fluxos de navegação, prever erros de usabilidade e validar as funcionalidades junto ao público-alvo. Também foram levantados dados de comportamento de usuários com base em técnicas de etnografia digital.

#### 5.1.1 Problemática a ser solucionada

1. **Dificuldade no transporte seguro de animais**
  - Solução: Plataforma com cadastro detalhado do pet (porte, saúde, necessidades especiais) e motoristas treinados.
2. **Falta de alternativas especializadas para deslocamento de pets**
  - Solução: Aplicativo exclusivo para transporte de animais, com corridas imediatas ou agendadas.
3. **Pouca confiança na segurança do serviço**
  - Solução: Sistema de monitoramento em tempo real, avaliações mútuas e validação documental rigorosa de motoristas.
4. **Problemas de acessibilidade em meios tradicionais de transporte**
  - Solução: Interface intuitiva com opções de filtros (porte do animal, cuidados especiais, motorista mais próximo).
5. **Ausência de flexibilidade em pagamentos e cancelamentos**
  - Solução: Carteira digital integrada, múltiplas formas de pagamento (Pix, cartão, carteira digital) e regras claras de cancelamento.

### 5.1.2 Etnografia

Foi conduzida observação do comportamento de tutores de pets durante deslocamentos para consultas veterinárias, pet shops e viagens. Essa técnica permitiu identificar hábitos e preocupações principais, como ansiedade dos animais, exigências de segurança, restrições alimentares e confiança no motorista. A etnografia contribuiu para alinhar a visão técnica dos desenvolvedores com as necessidades reais dos usuários.

### 5.1.3 Público-Alvo

- **Clientes (Tutores de Pets):**
  - Donos de cães, gatos e outros animais domésticos.
  - Pessoas que precisam levar seus pets a clínicas, parques, pet shops ou viagens.
  - Tutores que priorizam **segurança, conforto e agilidade** no transporte.
- **Motoristas/Funcionários:**
  - Prestadores autônomos que desejam gerar renda extra.
  - Motoristas treinados para lidar com pets, respeitando protocolos de segurança.
- **Gestores e Administradores:**
  - Responsáveis pela validação de motoristas.
  - Gerenciamento de tarifas, campanhas promocionais e relatórios financeiros.

### 5.1.4 Funcionalidades do Aplicativo

#### 5.1.4.1 Funcionalidades: Usuário/Tutor

- Cadastro/Login (Google, e-mail, MFA).
- Cadastro de múltiplos pets.
- Solicitar transporte imediato ou agendado.
- Acompanhamento em tempo real via mapa.
- Histórico de corridas e pagamentos.
- Cancelamento de corridas (conforme regras).
- Avaliação de motoristas.
- Chat de suporte integrado.

#### 5.1.4.2 Funcionalidades: Motorista

- Cadastro validado por gestor (CNH, CRLV, antecedentes).
- Definir disponibilidade (Disponível, Ocupado, Offline).
- Receber e aceitar/recusar corridas.
- Atualizar status da corrida.
- Visualizar perfil do pet transportado.

- Histórico de corridas e ganhos.
- Registro de ocorrências.

#### **5.1.4.3. Funcionalidades: Setor Administrativo**

- Gerenciar cadastros de motoristas e usuários.
- Monitorar corridas em tempo real.
- Intervir em disputas.
- Emitir relatórios e estatísticas.
- Suporte via chat interno.

#### **5.1.4.4. Funcionalidades: Gestor**

- Validar documentos de motoristas.
- Configurar tarifas dinâmicas (porte do pet, distância, taxa mínima).
- Gerar relatórios estratégicos.
- Criar campanhas e cupons de desconto.
- Dashboards de demanda e horários de pico.

#### **5.1.5. Erros**

Os erros serão tratados após a fase de testes com usuários reais e motoristas parceiros, utilizando controle de qualidade. Exemplos previstos:

- Falha no cadastro de pets.
- Problemas de integração com API bancária.
- Inconsistência na geolocalização.
- Erros de usabilidade no fluxo de pagamento.

#### **5.1.6. Conclusão**

A prototipagem possibilitou uma validação antecipada das principais funcionalidades do sistema, permitindo ajustes ainda na fase inicial. Isso reduz riscos, melhora a experiência do usuário e garante que o sistema final atenda às expectativas de clientes, motoristas e gestores. O protótipo validado será base para a implementação dos requisitos funcionais e não funcionais, conduzindo à próxima fase de desenvolvimento.

## **5.2. Detalhamento Aplicativo- Usuário/Tutor de Pets**

### **5.2.1. Login**

#### **5.2.1.1. Crie seu cadastro:**

- Botão “Registrar-se” direciona para usuário para preencher formulário de cadastro de novo usuário
- Nova página para formulário de informações PESSOAIS - página 2
- Após apertar botão de "concluir", direcionar para tela de perfil pessoal

#### **5.2.1.2. Continuar pelo Google**

- Nova página que direciona para autorização do google obrigatória
- Após autorizar, usuário é transferido para página de perfil pessoal.

#### **5.2.1.3. Já possuo cadastro**

- Campos de EMAIL e SENHA são apresentados na mesma e após inserir os dados corretos e apertar em "ENTRAR" o usuário é direcionado para página de perfil.

### **5.2.2. Perfil**

#### **5.2.2.1. Criar Perfil**

- Será uma página onde deve ser inserido todas as informações solicitadas em campos.
- Nome, CPF, endereço, e-mail e telefone serão campos obrigatórios a serem preenchidos

#### **5.2.2.1. Páginas do Perfil**

- “Páginas do Perfil” será introduzido após informado os dados da conta e apresentará a Carteira Digital, Suporte e FAQ e Avaliações. Cada uma destas será opções será uma nova página.
  - Carteira Digital: Página onde o usuário preencherá seus dados bancários para pagamento. Vinculação com API e seleção de métodos de pagamento
  - Suporte e FAQ: Chat de atendimento ao usuário automatizado ou humano.
  - Avaliações: Página onde mostra todas as notas e comentários recebidos pelos prestadores
- Sair do aplicativo: botão que direciona o usuário para a primeira página do app de cadastro/login



- Excluir Conta: opção com confirmação onde o usuário terá seus dados apagados e será direcionado para a primeira página do aplicativo de cadastro/login

### 5.2.3. Solicitar Corrida

- Solicitar uma nova corrida agendada ou imediata: O usuário é direcionado à um formulário a ser preenchido para que seja concluído o processo de chamada de um motorista para iniciar a corrida. O usuário deve preencher obrigatoriamente os campos: endereço de origem, endereço de destino, selecionar dentre as opções cadastradas anteriormente, qual o pet que será transportado, método de pagamento e o horário (se a corrida for agendada) do pedido.
- Antes de o pedido ser concluído e a solicitação ser enviada para a procura do motorista disponível, o usuário deve concluir o pagamento pela carteira digital.
- **Chat com o motorista:** Após a corrida ser solicitada e o motorista disponível der o seu aceite, o usuário tem acesso ao perfil e as avaliações do motorista, bem como o acesso ao chat com o motorista.
- **Deixar uma avaliação:** após o término da corrida, tanto o usuário quanto o motorista são direcionados a página para deixarem uma avaliação em forma de nota ou comentário, que posteriormente após aprovado pelo sistema, será disponibilizada no perfil pessoal.
- Cada corrida será uma subpágina e quando selecionada se apresentará com todas as informações daquele trabalho realizado
- Durante todo o trajeto, é disponibilizado para o usuário no mapa da corrida o trajeto em tempo real, bem como o chat com o motorista e o suporte com a plataforma para a resolução de problemas e conflitos.
- Após a finalização da corrida, o recibo do transporte é gerado pelo sistema e disponibilizado para o usuário
- Após a corrida concluída, a mesma é disponibilizada no histórico de corridas do motorista e no histórico de gastos do usuário.

#### **5.2.4. Cadastrar um novo Animal**

- O usuário deverá, obrigatoriamente, antes de solicitar uma nova corrida, adicionar a sua tutoria no aplicativo, ao menos um pet.
- O usuário deve incluir os campos obrigatórios para concluir o cadastro do pet
- O usuário tem a qualquer momento a opção de editar, visualizar, editar e excluir o perfil dos pets previamente cadastrados.
- O motorista apenas tem acesso ao perfil do animal ou do usuário quando a corrida for solicitada e ele der o aceite no transporte.

### **5.3. Detalhamento Aplicativo- Motorista/ Prestador de Serviços**

#### **5.3.1. Login**

##### **5.3.1.1. Crie seu cadastro:**

- Botão criar cadastro direciona para usuário para preencher formulário de
- Nova página para formulário de informações Pessoais e envio de documentação para a aprovação
- Após apertar botão de "concluir", direcionar para criação de perfil.

##### **5.3.1.2. Continue pelo Google**

- Nova página que direciona para autorização do google obrigatória
- após autorizar, usuário é transferido para página de criação de perfil

##### **5.3.1.3. Já possuo Conta**

- Campos de EMAIL e SENHA são apresentados na mesma tela e após inserir os dados corretos e apertar em "ENTRAR" o usuário é direcionado para página de perfil

##### **5.3.1.4. Envio de Documentação**

- Após o preenchimento dos dados pessoais, o motorista é direcionado a uma tela na qual deve inserir todos os documentos necessários para efetivar a criação da sua conta profissional aprovada no aplicativo.
- O motorista deve anexar obrigatoriamente os documentos de RG e CPF, carteira de motorista regularizada, CRLV e inserir a sua conta bancária.
- O aplicativo deve informar que a análise da documentação está pendente, e o motorista deve aguardar ao menos 24h para a aprovação, e uma notificação será enviada quando a análise for concluída.

### 5.3.2. Perfil Motorista/ Prestador de Serviços

"Páginas do Perfil" será introduzido após informado os dados da conta e apresentará a carteira digital, suporte e FAQ, perfil profissional e avaliações. Cada uma destas será opções será uma nova página.

- **Carteira Digital:** Página onde o usuário preencherá seus dados bancários para pagamento. Vinculação com API e seleção de métodos de pagamento.
- **Suporte e FAQ:** Chat de atendimento ao usuário automatizado ou humano.
- **Perfil:** Página onde o usuário terá acesso ao seu perfil profissional pessoal na plataforma, seu histórico de corridas, suas avaliações e editar, visualizar ou excluir seus dados pessoais.
- **Avaliações:** Página onde mostra todas as notas e comentários recebidos pelos prestadores.
- Sair do aplicativo: botão que direciona o usuário para a primeira página do app de cadastro/login.
- **Excluir Conta:** opção com confirmação onde o usuário terá seus dados apagados e será direcionado para a primeira página do aplicativo de cadastro/login
- Após concluir o seu cadastro e ter as suas documentações aceitas pela plataforma o motorista terá acesso a tela onde poderá mudar o seu status de disponibilidade para corridas e dar início a um novo transporte.
- O motorista ao ser solicitado a uma nova corrida, tem a opção de analisar o perfil do usuário solicitante e do Animal para tomar a decisão de aceite da corrida.
- Após o aceite, tem acesso ao chat com o solicitante e o mapa do trajeto a ser concluído.
- Durante o trajeto, possui acesso ao chat com o solicitante, o mapa e ao chat com o suporte do aplicativo.
- Após a conclusão da corrida o motorista pode realizar a avaliação do usuário e do animal em seus perfis e mudar o status de sua disponibilidade de trabalho, para offline ou disponível.
- Após a finalização da corrida, o recibo da corrida é enviado ao motorista e atualizada em seu histórico de ganhos e histórico de corridas.
- O motorista pode aceitar corridas previamente agendadas pelos usuários.

#### 5.4. Detalhamento Aplicativo- Gestor/ Setor Administrativo

- É trabalho do setor administrativo realizar a análise das documentações dos motoristas e responsabilizar-se pela veracidade e segurança dos usuários do aplicativo
- É dever do setor administrativo e dos gestores manterem a segurança durante as corridas, garantindo o chat como maior suporte entre usuário e software.
- É dever do setor administrativo fiscalizar as demandas e chamadas do aplicativo e usuários

#### 5.5. Protótipo da Interface

[Protótipo Figma do Projeto](#)



## 6. Critérios de Qualidade de Software

Os critérios de qualidade do sistema Bicho Express foram definidos com base nas normas ISSO/IEC 25010 e nas melhores práticas de engenharia de software, garantindo desempenho, segurança e confiabilidade.

Esses critérios avaliam o quanto o sistema atende aos requisitos funcionais e não funcionais descritos ao longo deste documento.

Os critérios de qualidade asseguram que o software atenda às necessidades de tutores, motoristas e administradores, mantendo a segurança, usabilidade, portabilidade e confiabilidade.

### 6.1. Funcionalidade

Aspecto	Descrição
Adequação	O Bicho Express atende às necessidades de transporte para animais, com funções específicas de cadastro de agendamento, pagamento digital, rastreamento e avaliação de motoristas.
Acurácia	As informações são validadas por API's externas (CPF, CEP, Google Maps, gateways de pagamento), garantindo dados corretos e transações seguras.
Interoperabilidade	Comunicação via API's RESTful, integrando módulos de usuário, motorista e setor administrativo, além de serviços externos (mapas, pagamentos, notificações)
Conformidade funcional	Cada requisito foi especificado, testado e vinculado a fluxos lógicos, conforme o documentado nos RFs e RNFs.

**Tabela 3- Funcionalidades de Qualidade do Software( Fonte, autores, 2025)**

Exemplo prático:

Quando o tutor solicita uma corrida, o sistema realiza automaticamente o matching com motoristas disponíveis, calcula tarifas dinâmicas, apresenta o tempo estimado e confirma o transporte do pet, garantindo fluidez no serviço.

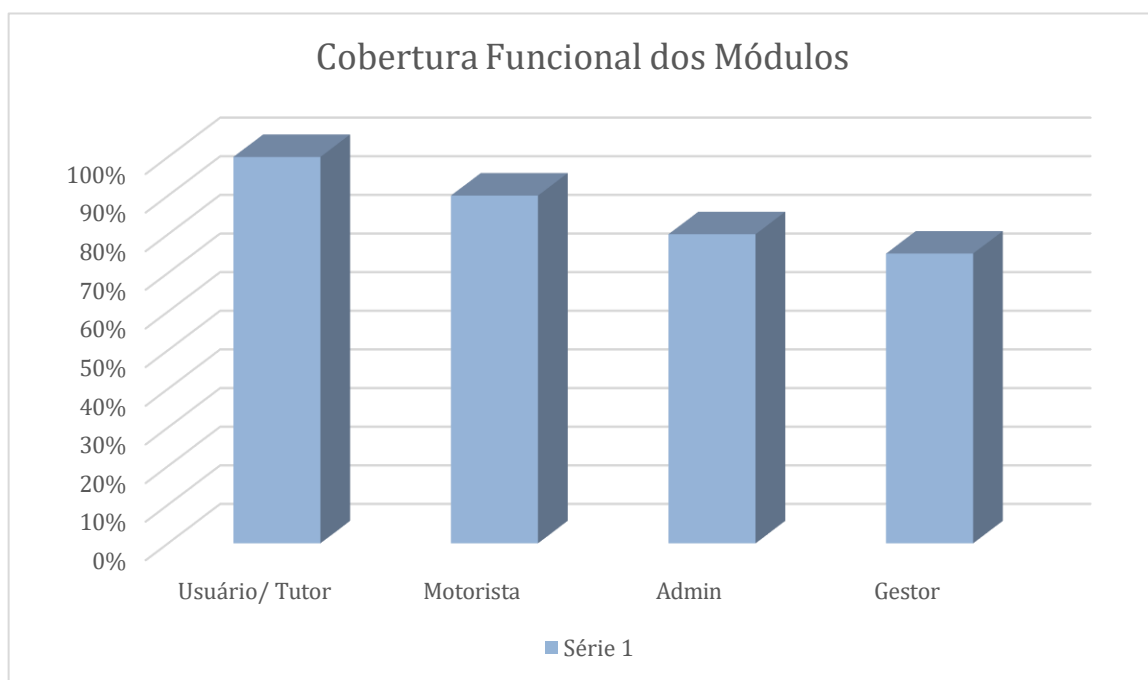


Figura 23- Ilustrativo- Cobertura funcional dos módulos

## 6.2. Confiabilidade

Critério	Implementação
Maturidade	A arquitetura em micro serviços isola falas e permite reinicialização de módulos sem afetar o sistema como um todo.
Disponibilidade	Garantia de uptime de 99,9% com infraestrutura AWS e backups automáticos diários.
Tolerância a falhas	Balanceador de carga (Load Balancer) distribui requisições, evitando sobrecarga em instâncias.
Recuperabilidade	Em falhas técnicas, o sistema é restaurado em até 15 minutos, conforme RNF27

Tabela 7- Confiabilidade do Software (Fonte, autores,2025)

Mecanismos de Confiabilidade:

- Logs centralizados e alertas automáticos em caso de falha de API
- Backups diários e restauração em nuvem (Firebase + AWS S3)
- Failover automático entre zonas de disponibilidade AWS.

### 6.3. Usabilidade

O software foi projetado com foco em simplicidade e acessibilidade, facilitando a navegação tanto para tutores quanto para motoristas.

<b>Critério</b>	<b>Implementação no Software</b>
Facilidade de aprendizagem	Interface intuitiva, ícones autoexplicativos, textos curtos e diretos.
Eficiência de uso	As principais ações (solicitar corrida, alterar status, acompanhar pet) são realizadas em até cliques
Estética e Design	Baseado em princípios Material Design? Human Interface Guidelines
Acessibilidade	Compatível com modo alto contraste, leitores de tela e fontes ajustáveis

Tabela 8- Critérios de Usabilidade do Software

### 6.4. Eficiência de Desempenho

<b>Métrica</b>	<b>Valor Alvo</b>	<b>Descrição</b>
Tempo de resposta	$\leq 2$ segundos	Carregamento de páginas e consultas em APIs
Tempo de Notificação	$\leq 5$ segundos	Envio de notificações push e atualizações em tempo real
Usuários simultâneos	100.000	Capacidade máxima estimada em carga de pico.
Uso de CPU/ memória	Otimizado por containerização	Serviços leves e escaláveis em AWS EC2 e Firebase.

Tabela 9- Métricas de Eficiência de Desempenho

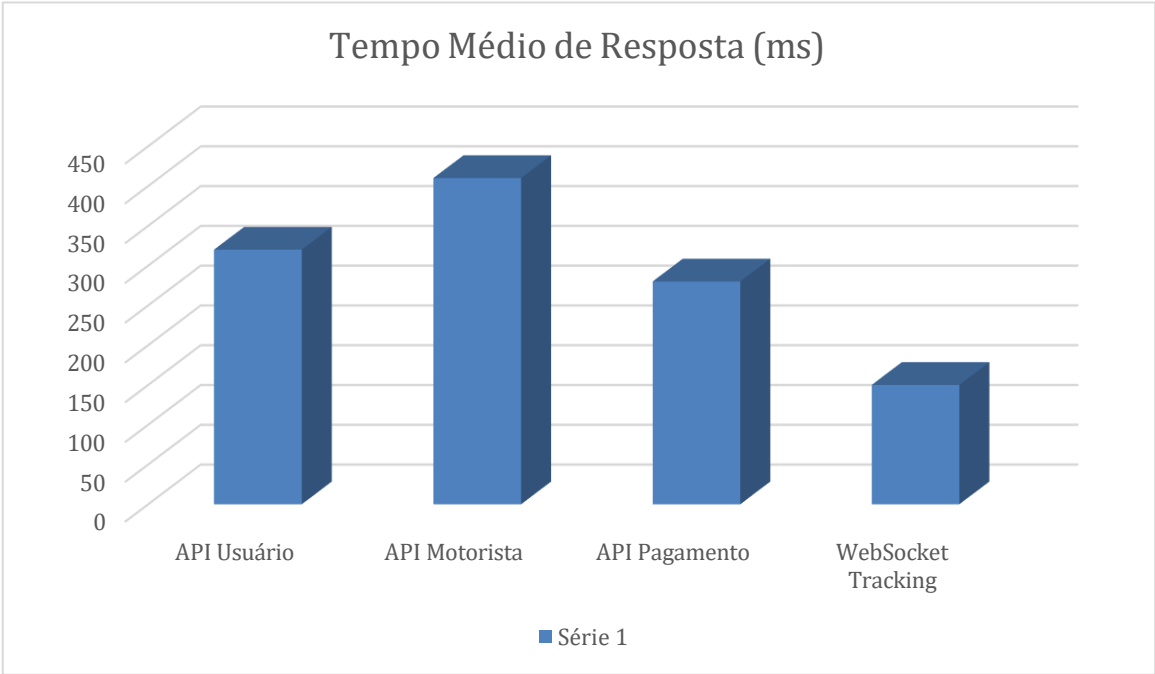


Figura 24- Desempenho do Software (Autores, 2025)

6.5. Manutenibilidade

A modularidade e uso de micro serviços permitem atualizações e correções sem impacto global.

Aspecto	Descrição
Modularidade	Módulos independentes: Usuário, Motorista, Corridas, Pagamento, Chat e Relatórios
Reusabilidade	Componentes de interface (React Native) e back-end reutilizados em múltiplos módulos.
Analisabilidade	Logs estruturados e monitoramento contínuo facilitam identificação de falhas
Modificabilidade	Alterações localizadas em serviços sem interferir nos demais
Testabilidade	Testes automatizados via CD/CD, incluindo testes unitários e de integração.

Tabela 10- Aspectos de Manutenibilidade do Software ( Autores, 2025)



Atualizações frequentes podem ser lançadas sem interromper o funcionamento do aplicativo.

## 6.6. Portabilidade

<b>Critério</b>	<b>Detalhamento</b>
Adaptabilidade	Framework React Native garante compatibilidade com Android(8.0+) e iOS(12+)
Instabilidade	Instalação simplificada pela Play Store e App Store
Substitutibilidade	Pode ser reinstalado sem perda de dados (histórico armazenado em nuvem)
Escalabilidade horizontal	Novos servidores podem ser adicionados automaticamente via AWS Auto Scaling

Tabela 11- Portabilidade do Software (autores, 2025)

## 6.7. Segurança

<b>Medida de Segurança</b>	<b>Implementação</b>
Proteção de dados	Criptografia AES-256 e TLS 1.3 Atende à LGPD e ISSO/IEC 27001
Autenticação	Login multifator (MFA) e OAuth2 via Google
Conformidade Legal	LGPD, ISSO/IEC 27001 e PCI-DSS ( para pagamentos)
Auditoria e Logs	Rastreamento completo de ações críticas (login, exclusão, cancelamento)
Privacidade do Usuário	Compartilhamento limitado de dados durante a corrida

Tabela 12- Medidas de segurança do Software (autores, 2025)

## 6.8. Compatibilidade e Integração

O sistema mantém compatibilidade com APIs externas para garantir comunicação eficiente com serviços de terceiros:

- Google Maps API-> Cálculo de rotas e geolocalização
- Gateway de Pagamento -> Pix, cartões e carteiras
- Firebase Authentication -> Login seguro e gerenciamento de usuários
- ASW S3 -> Armazenamento de documentos e imagens

## **6.9. Conclusão dos Critérios de Qualidade**

O software apresenta alto nível de qualidade técnica e operacional, destacando-se pela:

- Usabilidade e eficiência nas principais tarefas
- Segurança e conformidade legal com a LGPD
- Escalabilidade garantida por uma arquitetura em microsserviços
- Portabilidade e confiabilidade, permitindo uso multiplataforma estável

O conjunto de critérios evidencia que o sistema está preparado para a operação contínua em larga escala, assegurando excelência técnica, experiência, fluída e bem-estar animal durante todo o processo de transporte.

## 7. Testes Requisitos Funcionais

Este capítulo é direcionado para a fase de testes do Software. O objetivo é garantir que o sistema cumpra os requisitos funcionais (RF) e não-funcionais (RNF) definidos, garantindo qualidade, segurança e usabilidade.

### 7.1. Tabela de Prioridade

TABELA DE PRIORIDADE		
Nível	Nome	Descrição
1	Crucial	Falhas que impedem uso básico ( ex: cadastro/login, pagamento). Exige correção imediata.
2	Muito Alto	Funcionalidade crítica com degradação severa (ex: agendamento, segurança)
3	Alto	Importante, mas sistema continua utilizável (ex: filtros de busca, relatórios)

**Tabela 13- Tabela de prioridades do Software ( Fonte: Autores, 2025)**

### 7.1.1. Testes- Aplicativo 1 (Usuário/ Tutor de Pets)

#### 7.1.1.1. Plano de Testes – RF01 Cadastro/ Login/ MFA

**Objetivo:** Validar criação de conta, login ( e-mail e senha), login via Google, MFA, recuperação de senha, logout e exclusão.

ID	RF	Título	Prioridade	Tipo
TC-RF01-01	RF01	Criar Cadastro	CRUCIAL	Funcional
TC-RF01-02	RF01	Inserir e validar email	CRUCIAL	Funcional/ Integração (API)
TC-RF01-03	RF01	Criar senha (política)	CRUCIAL	Segurança
TC-RF01-04	RF01	Inserir CPF (validação)	CRUCIAL	Funcional/ Integração (API)
TC-RF01-05	RF01	Inserir Telefone	MUITO ALTO	Funcional
TC-RF01-06	RF01	Restrição de Idade	MUITO ALTO	Funcional/ Regra de Negócio
TC-RF01-07	RF01	Login pelo Google	MUITO ALTO	Integração
TC-RF01-08	RF01	Já possui cadastro (login)	MUITO ALTO	Funcional
TC-RF01-09	RF01	Login não Validado	ALTO	Usabilidade
TC-RF01-10	RF01	Recuperação de senha	MUITO ALTO	Funcional/ Segurança
TC-RF01-11	RF01	MFA ( setut & valid)	CRUCIAL	Segurança
TC-RF01-12	RF01	Excluir conta	ALTO	Funcional/ LGPD

Tabela 14- Requisitos por prioridade RF01 (Fonte: Autores, 2025)

#### 7.1.1.2. Roteiro de Testes- TC-RF01-01 (Criar Cadastro)

**Procedimento:** Criar conta pelo formulário do app.

**Pré-condições:** App instalado; acesso à internet; API de verificação disponível (CPF, telefone)

Fluxo RF01- Cadastro (Login)

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Abrir app na página login	-	Tela inicial de Login/Cadastro exibida
2	Selecionar "Registrar-se"	-	Formulário de cadastro exibido
3	Preencher campos (email, senha, conf. Senha, data nascimento, nome, CPF, telefone)	email: user@exemplo.com; senha: Abc!2345; conf: Abc!2345; nascimento 1990-01-01; CPF: 48410622836; tel: +5511967779535	Validações locais OK (campos obrigatórios)
4	Submeter	-	API valida CPF/telefone/email; se aprovado resposta OK
5	Receber resposta	-	"Cadastro concluído" + redirect para perfil

Tabela 15- Fluxo de Requisitos Cadastro, Fonte: Autores, 2025

### 7.1.1.3. Roteiro de testes – TC-RF01-02 Cadastro (Login)

**Procedimento:** Inserir E-mail.

**Pré-condições:** App instalado; acesso à internet; API de verificação disponível; iniciar primeiro cadastro.

Fluxo RF01- Cadastro (Login)- email

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Validação de e-mail	Usuário insere e-mail existente no campo descrito como "e-mail"	Sistema de API confirma e-mail e usuário pode prosseguir para a próxima etapa
2	Validação de Email	e-mail: user@exemplo.com	Mensagem "E-mail incorreto! Tente novamente" será impressa na tela
3	Validação de Email	-	Mensagem "E-mail correto! Tente novamente" será impressa na tela

Tabela 16- Fluxo de Requisitos RF01- inserir e-mail

#### 7.1.1.4. Roteiro de Testes – TC-RF01-03

**Procedimento:** Inserir e Validar Senha

**Pré-condições:** App instalado; acesso à internet; API de validação disponível; Iniciar Login.

Fluxo RF01- Cadastro (Login)- Validar Senha

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Criação de senha de acesso	Mínimo oito dígitos, máximo doze dígitos, uma letra maiúscula e um caractere senha: Abc!2345	Senha aprovada e usuário prossegue para próxima etapa
2	Criação de senha de acesso	Usuário insere senha com menos de oito dígitos	Senha recusada e usuário é impedido de prosseguir enquanto não seguir o padrão pré estabelecido
3	Criação de senha de acesso	Usuário insere senha com mais de doze dígitos	Senha recusada e usuário é impedido de prosseguir enquanto não seguir o padrão pré estabelecido
4	Criação de senha de acesso	Usuário insere senha sem letras maiúsculas	Senha recusada e usuário é impedido de prosseguir enquanto não seguir o padrão pré estabelecido
5	Criação de senha de acesso	Usuário insere senha sem caráter	Senha recusada e usuário é impedido de prosseguir enquanto não seguir o padrão pré estabelecido

Tabela 17- Fluxo de Requisitos RF01- validação de senha

#### 7.1.1.5. Roteiro de Testes TC-RF01-04

**Procedimento:** Inserir CPF

**Pré-condições:** App instalado; acesso à internet; API validação do CPF; iniciar cadastro.

Fluxo RF01- Cadastro (Login)- Validar CPF

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Validação de CPF	Usuário insere CPF existente no campo descrito como "CPF" CPF: 48410622836	API correspondente confirma CPF e usuário pode prosseguir para a próxima etapa
2	Validação de CPF	Usuário insere CPF inexistente no campo descrito como "CPF"	Mensagem "CPF incorreto! Tente novamente" será impressa na tela
3	Validação de CPF	Usuário insere CPF com menos de 11 dígitos no campo descrito como "CPF"	Mensagem "CPF incorreto! Tente novamente" será impressa na tela
4	Validação de CPF	Usuário insere CPF já cadastrado no campo descrito "CPF"	Mensagem "CPF incorreto! Tente novamente"

Tabela 18- Fluxo de Requisitos RF01- Validação do CPF



#### 7.1.1.6. Roteiro de Testes TC-RF01-05

Procedimento: Inserir Telefone.

Pré-condições: App instalado; acesso à internet; API disponível; iniciar cadastro

Fluxo RF01- Cadastro (Login)- Validar Telefone

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Quando o usuário selecionar "Criar Cadastro" e for redirecionado para a página de inserção de dados	Inserir número com DDD e nove dígitos +5511967779535	Se a API de telefone confirmar a existência do número o usuário pode seguir para a próxima etapa
2	Quando o usuário selecionar "Criar Cadastro" e for redirecionado para a página de inserção de dados	Número inserido sem DDD	Mensagem "Número invalidado! Insira DDD + número de celular" será impressa na tela quando o usuário tentar prosseguir.
3	Quando o usuário selecionar "Criar Cadastro" e for redirecionado para a página de inserção de dados	Número inserido com menos de nove dígitos	Mensagem "Número invalidado! Insira DDD + número de celular" será impressa na tela quando o usuário tentar prosseguir.

Tabela 19- Fluxo de Requisitos RF01- Validar Telefone

### 7.1.1.7. Roteiro de Testes TC-RF01-06

**Procedimento:** Restrição de Idade

**Pré-condições:** App instalado; acesso à internet; API disponível; cadastro iniciado.

Fluxo RF01- Cadastro (Login)- Restrição de Idade

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário insere a data de nascimento na etapa de criar cadastro	Data de nascimento inserida equivalente a maior de 18 anos	Sistema confirma maioridade e usuário pode seguir para próxima etapa
2	Usuário insere a data de nascimento na etapa de criar cadastro	Data de nascimento inserida não é equivalente a maior de 18 anos	Mensagem " Cadastro não permitido para menores de idade" é impressa na tela e usuário não pode prosseguir para próxima etapa

Tabela 20- Fluxo de Requisitos RF01- Restrição de idade

#### 7.1.1.8. Roteiro de Testes TC-RF01-07

**Procedimento:** Realizar Login via Google

**Pré-condições:** Conexão com API Google habilitada

Fluxo RF01- Cadastro (Login)- Login via Google

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Quando o usuário clicar em "continuar com Google"	—	Redirecionamento para Oauth Google
2	Quando o usuário consentir a autorização do login via Google	Número inserido sem DDD	Redirecionamento para o app
3	verificar dados mapeamento dos dados	—	Conta criada/ associada

Tabela 21- Fluxo de Requisitos RF01- Realizar Login via Google

### 7.1.1.9. Roteiro de Testes TC-RF01-08

**Procedimento:** Já possui cadastro, realizar login

**Pré-condições:** App instalado e cadastro concluído

Fluxo RF01- Cadastro (Login)- Já possui cadastro, realizar login

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Após instalar o aplicativo o usuário deve abri-lo sendo direcionado para página inicial de Login/Cadastro	Selecionar botão "Já possuo um cadastro"	Campos de Email e Senha são impressos na tela
2	Inserir Email e Senha	Email e senha correspondentes aos registrados no banco de dados	Se os dados informados estiver corretos o login será concluído e usuário será redirecionado para página de perfil do aplicativo
3	Inserir Email e Senha	Email e senha informados não correspondem aos salvos no banco de dados	Mensagem descrita como "Email ou senha incorretos" é impressa na tela, sendo obrigado a digitar informações novamente.

Tabela 22- Fluxo de Requisitos RF01- Já possui cadastro, realizar login

#### 7.1.1.10. Roteiro de Testes TC-RF01-09

**Procedimento:** Login não validado

**Pré-condições:** Cadastro concluído

Fluxo RF01- Cadastro (Login)- Login não validado

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Inserir corretamente e-mail e senha já cadastrados no sistema	Email e senha do usuário	Login concluído
2	Inserir e-mail não cadastrado no sistema	Email incorreto e senha do usuário	Mensagem "Login e/ou senha incorreto(os), tente novamente!" é impressa na tela e login não é concluído
3	Inserir senha não cadastrada no sistema	Email do usuário e senha incorreta	Mensagem "Login e/ou senha incorreto(os), tente novamente!" é impressa na tela e login não é concluído
4	Logar sem e-mail e senha	—	Mensagem "Login e/ou senha incorreto(os), tente novamente!" é impressa na tela e login não é concluído

Tabela 23- Fluxo de Requisitos RF01- login não validado

### 7.1.1.11. Roteiro de Testes TC-RF01-10

**Procedimento:** Recuperar senha

**Pré-condições:** Cadastro concluído

Fluxo RF01- Cadastro (Login)- Recuperação de senha

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Recuperação de senha	Usuário seleciona botão "esqueci minha senha" na etapa de login	Email de recuperação de senha é enviado para o email indicado pelo usuário
2	Criação de nova senha através do e-mail recebido	Usuário define e confirma nova senha através do link enviado pelo e-mail	Mensagem "Recuperação de senha aprovada. Realize o login novamente!" é impressa na tela
3	Teste de nova senha	Usuário insere e-mail e nova senha	Login efetuado com sucesso

Tabela 24- Fluxo de Requisitos RF01- Recuperar senha

### 7.1.1.12. Roteiro de Testes TC-RF01-11

**Procedimento:** MFA

**Pré-condições:** Cadastro concluído

Fluxo RF01- Cadastro (Login)- MFA

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Quando o usuário habilitar a MFA nas configurações	—	Enviar código por SMS/ gerar TOTP
2	Quando inserir código	—	Autenticação bem sucedida
3	Quando inserir código inválido	—	Erro ao autenticar

Tabela 25- Fluxo de Requisitos RF01- MFA

### 7.1.1.13. Roteiro de Testes TC-RF01-12

**Procedimento:** Excluir conta.

**Pré-condições:** Cadastro e login concluídos.

Fluxo RF01- Cadastro (Login)- Excluir conta

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Login executado	Entrar na aba de "perfil"	Redirecionar usuário para aba de perfil onde ser apresentado todas as opções de perfil, inclusive a de "excluir conta"
2	Selecionar botão "excluir conta"		Aba de confirmação descrito como "Você realmente deseja excluir sua conta?"
3	Confirmar exclusão		Usuário seleciona "Confirmar" e mensagem descrita como "cadastro deletado" é imprimida na tela por 3 segundos. Em seguida o usuário é direcionado para página inicial de Cadastro/Login
4	Cancelar exclusão		Usuário seleciona "Cancelar" e aba de confirmação é fechada

Tabela 26- Fluxo de Requisitos RF01- Excluir conta



### 7.1.2. Plano de Testes – RF02- Perfil/ Cadastro de Pets

**Objetivo:** Garantir a criação, edição, remoção e visibilidade do perfil do pet (obrigatórios: nome, espécies, porte, peso, vacinas)

ID	RF	Título	Prioridade	Tipo
TC-RF02-01	RF02	Criar Perfil de Pet	CRUCIAL	Funcional
TC-RF02-02	RF02	Salvar histórico de vacinas	MUITO ALTO	Funcional
TC-RF02-03	RF02	Campos obrigatórios vazios	MUITO ALTO	Funcional
TC-RF02-04	RF02	Editar perfil de pet	ALTO	Funcional
TC-RF02-05	RF02	Visualizar perfil de motorista	CRUCIAL	Funcional
TC-RF02-06	RF02	Upload de foto	MUITO ALTO	Funcional
TC-RF02-07	RF02	Inserir condição médica com anexo	ALTO	Funcional

Tabela 27- Prioridade por RF02

### 7.1.2.1. Roteiro de Testes TC-RF02-01

**Procedimento:** Criar perfil de pet

**Pré-condições:** Ter cadastro pessoal concluído, login concluído

Fluxo RF02- Criar perfil de Pet

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário o app e navega até perfil	"meus pets"	Página que criação do perfil novo de pet
2	Cria perfil do pet	Insere todos os campos corretamente	Perfil criado
3	Anexo da carteira de vacinação do pet e foto do animal	—	Pet adicionado

Tabela 28- Fluxo de Requisitos RF02- Criar perfil de pet

### 7.1.2.2. Roteiro de Testes TC-RF02-02

**Procedimento:** Salvar Histórico de vacinas

**Pré-condições:** Login e cadastro pessoal concluídos; pelo menos um pet cadastrado no perfil

Fluxo RF02- Salvar histórico de vacinas

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário o app e navega até perfil	Seleciona "meus pets"	Página de edição do perfil do pet
2	Clica em anexar histórico de vacinas do animal	Insere o arquivo corretamente suportado	Perfil atualizado com sucesso
3	Anexa o arquivo	Arquivo inválido ou não suportado	Erro ao anexar o arquivo

Tabela 29- Fluxo de Requisitos RF02- Salvar Histórico de Vacinas do animal

### 7.1.2.3. Roteiro de Testes TC-RF02-03

**Procedimento:** Campos obrigatórios vazios

**Pré-condições:** Login e cadastro concluídos

Fluxo RF02- Campos obrigatórios vazios

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário clica em adicionar novo pet ou editar pet existente	—	Página de edição ou criação de pet
2	Usuário preenche todos os campos corretamente	—	Perfil atualizado/criado com sucesso
3	Usuário deixa campos obrigatórios vazios	—	Erro ao anexar salvar e impossibilita a criação final do perfil

Tabela 30- Fluxo de Requisitos RF02- Campos obrigatórios vazios

#### 7.1.2.4. Roteiro de Testes TC-RF02-04

**Procedimento:** Editar perfil de pet

**Pré-condições:** Login e cadastro concluído

Fluxo RF02- Editar perfil de pet

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário clica em editar perfil de pet já existente	—	Página de edição ou criação de pet
2	Usuário preenche todos os campos corretamente	—	Perfil atualizado/criado com sucesso
3	Usuário deixa campos obrigatórios vazios	—	Erro ao anexar salvar e impossibilita a criação final do perfil

Tabela 31- Fluxo de Requisitos RF02- Editar perfil de pet

#### 7.1.2.5. Roteiro de Testes TC-RF02-05

**Procedimento:** Visualizar perfil do motorista

**Pré-condições:** Estar cadastrado e logado, ter o cadastro de pelo menos um pet, ter solicitado uma nova corrida.

Fluxo RF02- Visualizar perfil do motorista

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário clica em solicitar nova corrida	—	Página de formulário de nova corrida
2	Usuário preenche todos os campos corretamente para realizar a solicitação	Campos de origem e destino, selecionar pet para o transporte e horário da corrida. Pagamento da corrida via carteira digital	Viagem solicitada e pagamento concluído
3	Motorista aceita corrida solicitada pelo usuário	—	Perfil do usuário é exibido para motorista e perfil do motorista é exibido para usuário

Tabela 32- Fluxo de Requisitos RF02- Visualizar perfil do motorista

#### 7.1.2.6. Roteiro de Testes TC-RF02-06

**Procedimento:** Upload de nova foto

**Pré-condições:** Cadastro concluído, cadastro de novo pet iniciado.

Fluxo RF02- Fazer upload de nova foto no perfil do pet

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário clica em adicionar ou editar pet	Preenche todos os campos obrigatórios corretamente	Campos aceitos
2	Clica em anexar nova foto do pet	Anexa uma foto do formato adequado	Imagem incluída no perfil do pet

Tabela 33- Fluxo de Requisitos RF02- Inserir nova imagem

#### 7.1.2.7. Roteiro de Testes TC-RF02-07

**Procedimento:** Inserir condição médica com anexo

**Pré-condições:** Cadastro do usuário concluído, cadastro do pet iniciado.

Fluxo RF02- Inserir condições médicas do animal com anexo

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário clica em adicionar ou editar pet	Preenche todos os campos obrigatórios corretamente	Campos aceitos
2	Clica em campo não obrigatório de condições médicas do pet	Preenche a descrição com o máximo de detalhes e anexa o laudo médico do pet	Descrição médica do animal incluída no perfil

Tabela 34- Fluxo de Requisitos RF02- Inserir descrição médica do pet com anexo

### 7.1.3. Plano de Testes – RF03- Solicitação de Nova Corrida

**Objetivo:** Validar o fluxo de solicitação de transporte de pets pelo tutor (usuário), tanto no modo imediato quando agendado, incluindo cálculo de estimativa (ETA e valor), seleção de tipo de veículo/ adaptação, seleção de pet, etc.

ID+A1:EG	RF	Título	Prioridade	Tipo
TC-RF03-01	RF03	Solicitação imediata- pet pequeno (corrida curta)	CRUCIAL	Funcional
TC-RF03-02	RF03	Solicitação agendada + integração com Google Calendar	MUITO ALTO	Funcional/Integração
TC-RF03-03	RF03	Estimativa de valor e ETA (precisão)	CRUCIAL	Funcional/Integração
TC-RF03-04	RF03	Sem motoristas disponíveis (fila de espera)	MUITO ALTO	Alternativo/ UX
TC-RF03-05	RF03	Rejeição automática por timeout do motorista	CRUCIAL	Funcional
TC-RF03-06	RF03	Escolha de opções especiais (caixa de transporte/medicação/alimentação)	ALTO	Funcional
TC-RF03-07	RF03	Cancelamento pelo usuário (dentro/fora do prazo)	MUITO ALTO	Funcional/Negócio
TC-RF03-08	RF03	Cancelamento pelo motorista (reatribuição da corrida)	MUITO ALTO	Funcional
TC-RF03-09	RF03	Solicitação com pagamento pré-aprovado	CRUCIAL	Integração/Financeiro
TC-RF03-10	RF03	Falha na API de mapas	ALTO	Resiliência
TC-RF03-11	RF03	Solicitação em área com alta demanda (surge pricing)	ALTO	Negócio/Funcional
TC-RF03-12	RF03	Solicitação para pet com necessidades especiais	CRUCIAL	Regras de Negócio
TC-RF03-13	RF03	Histórico de solicitação criado corretamente (registro)	ALTO	Persistência
TC-RF03-14	RF03	Agendamento com alteração/ cancelamento posterior	ALTO	Funcional

Tabela 35- Fluxo de Requisitos RF03- Solicitação de uma nova corrida



### 7.1.3.1. Roteiro de Testes TC-RF03-01- Solicitação imediata

**Procedimento:** Solicitação de uma nova corrida imediata

**Pré-condições:** usuário autenticado, e pelo menos um pet cadastrado

Fluxo RF03- Solicitar corrida imediata

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet	Página de formulário de requisição de nova corrida
2	Usuário preenche o formulário obrigatório para a corrida	Insere origem e destino, opção de transporte	Cálculo do ETA (estimativa do preço da corrida)
3	Confirma a corrida	—	Sistema inicia o matching com o motorista
4	Motorista aceita a corrida	Usuário visualiza o perfil do motorista	Inicia o rastreamento em tempo real

Tabela 36- Fluxo de Requisitos RF03- Solicitação de corrida imediata

### 7.1.3.2. Roteiro de Testes TC-RF03-02- Solicitação Agendada

**Procedimento:** Solicitar um anova corrida agendada

**Pré-condições:** Usuário autenticado, pet cadastrado, calendário e integração habilitada

Fluxo RF03- Solicitar uma nova corrida agendada

Nº	Passo	Ddos de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet e a opção "agendar"	Página de formulário de requisição de nova corrida
2	Usuário preenche o formulário obrigatório para a corrida	Insere origem e destino, opção de transporte	Cálculo do ETA (estimativa do preço da corrida)
3	Confirma a corrida e enviar para o calendário	—	Sistema grava a corrida com status de aguardando e agenda a notificação 15 min antes da corrida
4	No horário agendado	—	O sistema dispara o matching e notifica os motoristas como se fosse uma corrida imediata, e notifica o tutor de que o motorista está a caminho

Tabela 37- Fluxo de Requisitos RF03- Solicitação de corrida agendada

### 7.1.3.3. Roteiro de Testes TC-RF03-03- Estimativa de valor e ETA

**Procedimento:** Estimativa de valor e ETA (precisão)

**Pré-condições:** Usuário autenticado, serviço de mapas funcionando, tarifas configuradas

Fluxo RF03- Estimativa de valor e ETA

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário clica em "cálculo de estimativa de corrida"	Seleciona a origem e o destino, seleciona o porte do pet	O sistema calcula o valor estimado da corrida, tendo em vista as tarifas e a distância da corrida

Tabela 38- Fluxo de Requisitos RF03- Cálculo de corrida

### 7.1.3.4. Roteiro de Testes TC-RF03-04- Sem motoristas disponíveis

**Procedimento:** Sem motoristas disponíveis para a corrida

**Pré-condições:** Usuário autenticado, solicitação de corrida iniciada

Fluxo RF03- Motoristas indisponíveis para corrida por motivos de área remota ou horário sem motoristas online

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet e preenche o formulário de solicitação	Sistema inicia o matcing com o motorista
2	Após o timeout	—	Sistema exibe mensagem "nenhum motorista disponível no momento- tentar novamente ou agendar?"

Tabela 39- Fluxo de Requisitos RF03- Motoristas não disponíveis para uma nova corrida

### 7.1.3.5. Roteiro de Testes TC-RF03-05

**Procedimento:** Timeout do motorista e re-match

**Pré-condições:** Usuário autenticado, solicitar corrida iniciado

Fluxo RF03- Motorista notificado não aceita a corrida em X segundos

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet e preenche o formulário de solicitação	Sistema inicia o matcing com o motorista
2	Após o timeout com determinado motorista	—	Sistema redireciona corrida para outro motorista e notifica atraso para o usuário

Tabela 40- Fluxo de Requisitos RF03- Sistema redireciona motorista por timeout

### 7.1.3.6. Roteiro de Testes TC-RF03-06- Opções especiais para o transporte de pets

**Procedimento:** Opções especiais para transporte

**Pré-condições:** Usuário autenticado, solicitação de corrida iniciada

Fluxo RF03- Solicitação de opções especiais para transporte de animais

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet e preenche o formulário de solicitação	Sistema inicia o matcing com o motorista
2	Usuário adiciona descrição de necessidade de uma opção de especial para o transporte	Insere na caixa de texto ou seleciona entre as opções a necessidade para o pet	Sistema filtra entre os motoristas disponíveis os que se enquadram na necessidade do pet

Tabela 41- Fluxo de Requisitos RF03- Solicitação de opções especiais

### 7.1.3.7. Roteiro de Testes TC-RF03-07- Cancelamento pelo usuário (dentro/fora de prazo)

**Procedimento:** Cancelamento pelo usuário sendo a corrida dentro ou fora do prazo estipulado

**Pré-condições:** Usuário autenticado, solicitação de corrida iniciada

Fluxo RF03- Solicitação de cancelamento de corrida pelo usuário

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet e preenche o formulário de solicitação	Sistema Inicia a procura do motorista disponível
2	Usuário cancela antes do motorista aceitar a corrida		Sem taxa
3	Usuário seleciona a opção de cancelamento antes de X minutos antes do embarque		Cancelamento grátis
4	Usuário solicita cancelamento após aceite do motorista e muito próximo ao embarque		Cobrar taxa de cancelamento

Tabela 42- Fluxo de Requisitos RF03- Solicitação de cancelamento por parte do usuário

### 7.1.3.8. Roteiro de Testes TC-RF03-08- Cancelamento da corrida por parte do motorista

**Procedimento:** Cancelamento da corrida por parte do motorista

**Pré-condições:** Usuário autenticado, solicitação de corrida iniciada.

Fluxo RF03- Solicitação de cancelamento de corrida por parte do motorista

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet e preenche o formulário de solicitação	Sistema Inicia a procura do motorista disponível
2	Motorista cancela a corrida após aceitar		Sistema notifica o usuário do cancelamento e atraso na corrida. E inicia o re-match com outro motorista

Tabela 43- Fluxo de Requisitos RF03- Solicitação de cancelamento de corrida pelo motorista

### 7.1.3.9. Roteiro de Testes TC-RF03-09- Solicitação com pagamento pré-autorizado

**Procedimento:** Solicitação de nova corrida com o pagamento pré-autorizado

**Pré-condições:** Usuário autenticado, pagamento realizado, solicitação de corrida iniciada

Fluxo RF03- Solicitação de nova corrida com o pagamento pré-autorizado

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet e preenche o formulário de solicitação	Sistema Inicia a procura do motorista disponível
2	Usuário confirma a corrida e cartão é pré-autorizado (tokenização + hold)		Sistema captura valor, em caso de cancelamento em janela sem cobrança, liberar hold

Tabela 44- Fluxo de Requisitos RF03- Solicitação de corrida com pagamento pré-autorizado

### 7.1.3.10. Roteiro de Testes TC-RF03-10- Falha na API de mapas (fallback)

**Procedimento:** Simular erro 500 na API de mapas durante cálculo de rota

**Pré-condições:** Usuário autenticado, sistema de API

Fluxo RF03- Sistema de API de mapas apresenta erro durante simulação de mapas

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "simulação de corrida"	Preenche as informações para o cálculo	Sistema realiza o fluxo de estimativa aproximada, usuário informado, logs de erro

Tabela 45- Fluxo de Requisitos RF03- Falha no sistema API de mapas

### 7.1.3.11. Roteiro de Testes TC-RF03-11- Surge pricing/ alta demanda

**Procedimento:** Surgimento de pricing nas corridas

**Pré-condições:** Quando alta demanda > oferta na região, multiplicador de pico pode ser aplicado

Fluxo RF03- Surgimento de Pricing na plataforma

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Simular alta demanda (vários requests) na mesma região		Sistema aplica multiplicador configurado e exibe ao usuário
2	Usuário solicita nova corrida		Sistema exibe o valor da corrida com etiqueta "horário de pico", política de aceite pelo usuário antes de confirmar a corrida

Tabela 46- Fluxo de Requisitos RF03- Surgimento de pricing por alta demanda no software

### 7.1.3.12. Roteiro de Testes TC-RF03-12- Solicitação para pet com necessidades especiais

**Procedimento:** Solicitação para pet com necessidades especiais

**Pré-condições:** Usuário autenticado, solicitação de nova corrida

Fluxo RF03- Solicitação de corrida com necessidades especiais

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet e preenche o formulário de solicitação	
2	Usuário seleciona a opção de pet com necessidades especiais		Sistema faz a busca dentre os motoristas disponíveis o que se adequa as necessidades especiais do animal

Tabela 47- Fluxo de Requisitos RF03- Solicitação para pet com necessidades especiais

### 7.1.3.13. Roteiro de Testes TC-RF03-13- Histórico de solicitação criado corretamente (registro)

**Procedimento:** Histórico de solicitações de corridas criadas para o usuário

**Pré-condições:** Usuário autenticado, corrida finalizada

Fluxo RF03- Histórico de corridas do usuário

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário finaliza uma corrida	confirma encerramento da corrida	Sistema adiciona ao histórico do motorista e do usuário a corrida e envia a nota fiscal da corrida ocorrida por email
2	Usuário Pode conferir em seu perfil o seu histórico de corrida		Sistema realiza a atualização do histórico a cada corrida ocorrida

Tabela 48- Fluxo de Requisitos RF03- Geração de histórico de corrida



### 7.1.3.14. Roteiro de Testes TC-RF03-14- Agendamento com alteração/ cancelamento posterior

**Procedimento:** Agendamento de corrida com alteração ou cancelamento posterior

**Pré-condições:** Usuário autenticado, solicitação de corrida agendada iniciada.

Fluxo RF03- Solicitação de corrida agendada e cancelamento e alteração posterior

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida agendada"	Seleciona o pet e preenche o formulário de solicitação	Sistema grava a corrida e envia notificação para o usuário
2	Usuário seleciona cancelamento da corrida agendada X minutos antes do início do matching		Sistema cancela corrida sem taxas
3	Usuário seleciona cancelamento da corrida agendada após o início da busca por motorista		Sistema cancela corrida com taxa

Tabela 49- Fluxo de Requisitos RF03-Agendamento de corrida e cancelamento posterior

#### 7.1.4. Roteiro de Testes RF04- Exibir o Motorista mais próximo

**Objetivo:** Validar que o sistema identifica corretamente o motorista mais próximo do ponto de coleta informado pelo tutor, calcula e exibe a estimativa de tempo de chegada (ETA) e o motorista candidato com as informações mínimas necessárias.

ID	RF	Título	Prioridade	Tipo
TC-RF04-01	RF04	Selecionar motorista mais próximo- urbano	CRUCIAL	Funcional
TC-RF04-02	RF04	Filtrar por caixa/porte	CRUCIAL	Funcional
TC-RF04-03	RF04	Ignorar telemetria desatualizada	MUITO ALTO	Resiliência
TC-RF04-04	RF04	Tie-braker entre drivers	ALTO	Algorítmico
TC-RF04-05	RF04	Sem motoristas próximos	MUITO ALTO	UX
TC-RF04-06	RF04	Tempo de cálculo <= 2s	CRUCIAL	Performance

Tabela 50- Fluxo de Requisitos RF04

#### 7.1.4.1. Roteiro de Testes TC-RF04-1- Selecionar motorista mais próximo

**Procedimento:** Selecionar motorista mais próximo (urbano)

**Pré-condições:** Usuário autenticado, solicitação de corrida iniciada

Fluxo RF04- Encontrar motorista mais próximo

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet e preenche o formulário de solicitação	Sistema Inicia a procura do motorista disponível
2	ETA calcula. Motorista mais próximo recebe a solicitação de corrida	Motorista aceita a corrida	Corrida inicia, rastreamento é iniciado
3	ETA calcula. Motorista mais próximo recebe a solicitação de corrida	Motorista recusa a corrida	Sistema reinicia a busca por outro motorista

Tabela 51- Fluxo de Requisitos RF04- Busca de motorista mais próximo

#### 7.1.4.2. Roteiro de Testes TC-RF04-02- Filtrar por caixa/porte

**Procedimento:** Sistema deve filtrar motorista por necessidade de caixa de transporte e/ou porte de animal

**Pré-condições:** Usuário autenticado, solicitação de corrida iniciada

Fluxo RF04- Filtrar por caixa/porte do animal

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet e preenche o formulário de solicitação	Aba de necessidades abre
2	Usuário seleciona necessidade de caixa de transporte e porte do animal		Sistema inicia a busca por motoristas que se enquadram nos filtros selecionados pelo usuário
3	Usuário recebe motorista que se enquadra em suas necessidades	Motorista aceita a corrida	Sistema inicia a corrida

Tabela 52- Fluxo de Requisitos RF04- Filtragem de motoristas por necessidades dos usuários

#### 7.1.4.3. Roteiro de Testes TC-RF04-03- Ignorar telemetria desatualizada

**Procedimento:** Ignorar telemetria desatualizada

**Pré-condições:** Usuário autenticado, solicitação de corrida iniciada

Fluxo RF04- Ignorar telemetria desatualizada

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet e preenche o formulário de solicitação	Sistema inicia busca por motoristas mais próximos
2	Motorista não disponível		Sistema ignora motorista não disponível para o serviço

Tabela 53- Fluxo de Requisitos RF04- Ignorar telemetria desatualizada

#### 7.1.4.4. Roteiro de Testes TC-RF04-04- Tie-braker entre drivers

**Procedimento:** O tie-braker entre 2 motoristas

**Pré-condições:** Usuário autenticado, solicitação de corrida iniciada

Fluxo RF04- Tie-braker entre dois motoristas

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a página "solicitar corrida"	Seleciona o pet e preenche o formulário de solicitação	Sistema inicia busca por motoristas mais próximos
2	D1/D2 quase empatados		Tie-break definido por ETA-> rating

Tabela 54- Fluxo de Requisitos RF04- Tie-brakear entre 2 motoristas

### 7.1.5. Roteiro de Testes RF05- Acompanhar o Transporte em Tempo Real (Tracking)

**Objetivo:** Validar que o sistema fornece rastreamento em tempo real da corrida, com atualização contínua da posição do motorista, ETA recalculado, eventos de status (motorista a caminho, embarcado, concluído), e exibição segura de informações ao tutor. Testar robustez em perda de telemetria, precisão de ETA, e desempenho sob carga.

ID	RF	Título	Prioridade	Tipo
TC-RF05-01	RF05	Exibir posição do motorista em mapa (tempo real)	CRUCIAL	Funcional
TC-RF05-02	RF05	Atualização de posição $\leq 5s$ (telemetria)	CRUCIAL	RNF/Performance
TC-RF05-03	RF05	Indicar ETA e recalculado em movimento	CRUCIAL	Funcional/Integração
TC-RF05-04	RF05	Lidar com telemetria desatualizada / fallback	MUITO ALTO	Resiliência
TC-RF05-05	RF05	Mostrar últimos eventos (chegou, embarcou, concluído)	MUITO ALTO	Funcional
TC-RF05-06	RF05	Privacidade: mascaramento de dados do motorista	ALTO	Segurança/Legislação
TC-RF05-07	RF05	Histórico de rastreamento (replay)	ALTO	Funcional/Forense
TC-RF05-08	RF05	Modo offline / reconciliação de posições	MUITO ALTO	Robustez
TC-RF05-09	RF05	Escalar alertas quando perda > X segundos	ALTO	Operacional

Tabela 55- Fluxo de Requisitos RF05

### 7.1.5.1. Roteiro de Testes TC-RF05-01- Atualização de posição em tempo real

**Procedimento:** Validar que as posições do motorista aparecem para o tutor com atualização a cada ~5s.

**Pré-condições:** Corrida ativa; motorista com app em foreground/background simulando telemetria.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre a tela "Acompanhar corrida"	—	Mapa carregado, camada de rota visível
2	Sistema solicita telemetria para motorista	rideld	Recebe posição atual (lat,lon), timestamp, accuracy
3	Exibir ícone do motorista na posição recebida	lat/lon	Ícone aparece no mapa; carimbo horário visível
4	Exibir ETA calculado	coords motorista + origem	ETA mostrado (minutos) e distância (m/km)
5	Verificar atualização contínua (ver TC-RF05-02)	—	Posição é atualizada no mapa conforme telemetria

Tabela 56- Fluxo de Requisitos RF05- Atualização de posição em tempo real

### 7.1.5.2. Roteiro de Testes TC-RF05-02- Atualização de posição $\leq 5s$

**Procedimento:** Testar frequência e latência das atualizações de posição.

**Pré-condições:** Motorista envia telemetria com frequência configurável (ideal 3–5s).

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Simular telemetria do motorista ( $t=0$ , $t=+3s$ , $+6s$ )	pacotes GPS	Aplicativo do tutor atualiza ícone $\leq 5s$ após chegada do pacote
2	Medir tempo entre chegada do pacote no backend e render no cliente	timestamps	Tempo médio $\leq 2s$ ; nenhum pacote ignorado indevidamente
3	Testar com 50% de perda de pacotes	pacotes perdidos	Sistema aplica suavização e não "teleporta" o ícone

Tabela 57- Fluxo de Requisitos RF05- Atualização de posição  $\leq 5s$

### 7.1.5.3. Roteiro de Testes TC-RF05-03- Indicar ETA e recalcular em movimento

**Procedimento:** Validar cálculo e atualização de ETA usando Directions API (ou mock).

**Pré-condições:** Serviço de rotas disponível.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Receber posição inicial do motorista	coords	Calcular rota até origem; exibir ETA
2	Motorista movimenta-se (mudança de rota)	coords atualizados	Recalcular ETA em $\leq 3s$ e atualizar UI
3	Simular tráfego/obstrução (mock)	Directions API responde maior tempo	ETA ajustado e notificação opcional ao usuário

Tabela 58- Fluxo de Requisitos RF05- Indicar ETA



#### 7.1.5.4. Roteiro de Testes TC-RF05-04- Lidar com telemetria desatualizada / fallback

**Procedimento:** Garantir que posições antigas sejam identificadas e manejadas.

**Pré-condições:** Configuração staleThreshold = 30s (exemplo).

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista para de enviar telemetria (> staleThreshold)	lastSeen > 30s	Aplicativo do tutor exibe indicador "Posição desatualizada"
2	Nova telemetria chega após 1 min	coords nova	Mapa atualiza e sistema mostra mensagem "Atualizado"
3	Se telemetria ausente por X min (config)	lastSeen > 5 min	Sistema aciona alerta para suporte (admin)

Tabela 59- Fluxo de Requisitos RF05- Lidar com telemetria desatualizada

#### 7.1.5.5. Roteiro de Testes TC-RF05-05- Mostrar últimos eventos (chegou, embarcou, concluído)

**Procedimento:** Verificar exibição e sincronização de eventos.

**Pré-condições:** Motorista aciona eventos no app.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista aciona "Cheguei"	evento: arrived	Tutor recebe push + badge no mapa; timestamp registrado
2	Motorista marca "Embarcado"	evento: boarded	Status muda para "Embarcado"; ETA recalculada para destino
3	Motorista marca "Concluído"	evento: finished	Tela final aparece; opção de avaliação surge

Tabela 60- Fluxo de Requisitos RF05- Mostrar últimos eventos

#### 7.1.5.6. Roteiro de Testes TC-RF05-06- Privacidade: mascaramento de dados do motorista

**Procedimento:** Garantir que dados sensíveis não apareçam indevidamente.

**Pré-condições:** Motorista cadastrado com CPF, telefone.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Visualizar perfil do motorista antes da aceitação	—	Somente nome parcial, foto e avaliação mostrados (telefone mascarado)
2	Após motorista aceitar e estar a caminho	—	Telefone completo liberado ao tutor somente quando permitido (política)
3	Exportar log para auditoria	—	CPF e dados sensíveis somente em logs administrativos com acesso controlado

Tabela 61- Fluxo de Requisitos RF05- Privacidade

#### 7.1.5.7. Roteiro de Testes TC-RF05-07- Histórico de rastreamento (replay)

**Procedimento:** Validar que rota histórica da corrida é salva e pode ser reproduzida.

**Pré-condições:** Corrida concluída; logs persistidos.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Abrir corrida concluída no histórico	rideld	Exibir mapa com rota percorrida e timestamps
2	Reproduzir movimento (replay)	botão "reproduzir"	Linha animada reproduz trajeto em velocidade configurável
3	Exportar arquivo para auditoria	export CSV/JSON	Arquivo contém pontos (lat,lon,timestamp,accuracy)

Tabela 62- Fluxo de Requisitos RF05- Histórico de rastreamento

#### 7.1.5.8. Roteiro de Testes TC-RF05-08- Modo offline / reconciliação

**Procedimento:** Simular perda temporária de conexão no app do motorista e posterior reconciliação.

**Pré-condições:** Motorista entra em túnel (sem sinal).

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista perde sinal por 2 min	sem pacotes	Tutor vê última posição conhecida e aviso "Conexão perdida"
2	Motorista reconecta e manda sequência de pontos bufferizados	pontos buffer	Backend aceita pontos, reconcilia e mapa do tutor atualiza com suavização
3	Verificar integridade do histórico	—	Não há gaps falsos; replay mostra trajetória aproximada

Tabela 63- Fluxo de Requisitos RF05- Modo Offline

### 7.1.6. Roteiro de Testes — RF06- Pagamento (Pix, Cartão, Carteira Digital)

**Objetivo:** Verificar o processo de pagamento ao finalizar corrida ou na solicitação (adiantamento), contemplando Pix, Cartão (tokenização, 3DS), e Carteira Digital (saldo), além de fluxos de erro, reembolso e segurança (PCI/LGPD).

ID	RF	Título	Prioridade	Tipo
TC-RF06-01	RF06	Pagamento via Pix (instantâneo)	CRUCIAL	Funcional/Integração
TC-RF06-02	RF06	Pagamento via Cartão (tokenização + 3DS)	CRUCIAL	Segurança/Integração
TC-RF06-03	RF06	Pagamento via Carteira Digital (saldo)	CRUCIAL	Funcional
TC-RF06-04	RF06	Cartão expirado / recusado	MUITO ALTO	Robustez
TC-RF06-05	RF06	Falha na autorização (gateway) e retry	MUITO ALTO	Resiliência
TC-RF06-06	RF06	Reembolso total/parcial via gateway	MUITO ALTO	Funcional/Financeiro
TC-RF06-07	RF06	Emissão de recibo / nota fiscal eletrônica	ALTO	Conformidade

TC-RF06-08	RF06	Segurança: não salvar PAN, usar token	CRUCIAL	Segurança/PCI
TC-RF06-09	RF06	Estorno por fraude / chargeback	ALTO	Operacional/Financeiro
TC-RF06-10	RF06	Pagamento offline (agendamento)	ALTO	UX/Resiliência

Tabela 64- Fluxo de Requisitos RF06

#### 7.1.6.1. Roteiro de Testes TC-RF06-01 — Pagamento via Pix (instantâneo)

**Procedimento:** Validar fluxo de pagamento por Pix (QR code ou copia-e-cola) e confirmação instantânea.

**Pré-condições:** Conta do usuário com app bancário; gateway de pagamentos com suporte a Pix em sandbox.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário escolhe Pix ao finalizar corrida	rideld, valor R\$	Gerar QR code OR copy-paste (payload Pix)
2	Usuário realiza pagamento no app bancário	payload Pix	Gateway confirma pagamento (status: aprovado)
3	Sistema recebe webhook de pagamento	webhook Pix	Atualiza corrida como "pago"; envia recibo ao usuário
4	Verificar tempo de confirmação	—	Confirmação em $\leq 30s$ (ideal) / $\leq 2$ min aceitável no sandbox

Tabela 65- Fluxo de Requisitos RF06- Pagamento via pix

### 7.1.6.2. Roteiro de Testes TC-RF06-02- Pagamento via Cartão (tokenização + 3DS)

**Procedimento:** Testar tokenização do PAN, fluxo 3DS (autenticação do emissor) e charge.

**Pré-condições:** Integração com gateway PCI-compliant; ambiente de testes com cartões válidos, recusados e com 3DS.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário seleciona Cartão e insere dados	PAN, validade, CVV, nome	PAN não armazenado; gateway retorna token
2	Gateway solicita 3DS (se aplicável)	cartão emissor 3DS	Redirecionar para challenge; usuário completa autenticação
3	Autorização do cartão	token, valor	Se aprovado: status "Pago"; gerar recibo
4	Se recusado (ex.: saldo insuf.)	resposta do gateway	Apresentar mensagem clara: "Pagamento recusado: motivo" e opção trocar forma

Tabela 66- Fluxo de Requisitos RF06- Pagamento via cartão

### 7.1.6.3. Roteiro de Testes TC-RF06-03- Pagamento via Carteira Digital (saldo interno)

**Procedimento:** Usar saldo interno (wallet) para pagar corrida.

**Pré-condições:** Usuário com saldo  $\geq$  valor.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário escolhe "Carteira"	confirmar saldo	Sistema valida saldo disponível
2	Confirma pagamento	—	Deduzir saldo e marcar corrida "pago"
3	Gerar recibo e histórico	—	Extrato da carteira atualizado; notificação enviada

Tabela 67- Fluxo de Requisitos RF06- Pagamento via carteira digital

#### 7.1.6.4. Roteiro de Testes TC-RF06-04- Cartão expirado / recusado

**Procedimento:** Confirmar tratamento de cartões inválidos.

**Pré-condições:** Cartão de teste expirado/declined no gateway.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Inserir cartão expirado	PAN vencido	Gateway retorna error; UI mostra "Cartão expirado"
2	Inserir cartão com limite insuficiente	response declined	UI oferta trocar forma; não criar cobrança pendente
3	Re-tentativa com novo cartão	—	Fluxo de tokenização + 3DS executado com sucesso

Tabela 68- Fluxo de Requisitos RF06- Cartão expirado/ recusado

#### 7.1.6.5. Roteiro de Testes TC-RF06-05- Falha na autorização (gateway) e retry

**Procedimento:** Simular timeout do gateway e confirmar retry/rollback.

**Pré-condições:** Gateway em modo timeout/erro.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Enviar autorização ao gateway (simular timeout)	—	Sistema registra timeout e tenta N retries (config 3)
2	Se todos os retries falharem	—	Marcar pagamento como "erro"; notificar usuário com opção pagar novamente
3	Logs devem conter requestId e motivos	—	Auditoria possível com requestId

Tabela 69- Fluxo de Requisitos RF06- Falha na autorização



#### 7.1.6.6. Roteiro de Testes TC-RF06-06- Reembolso total / parcial

**Procedimento:** Efetuar reembolso via painel administrativo / gateway.

**Pré-condições:** Pagamento confirmado (Pix/Cartão/Wallet).

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Abrir ticket e solicitar reembolso	rideld, motivo	Admin solicita reembolso
2	Sistema solicita API do gateway	amount	Gateway processa reembolso; status "reembolsado"
3	Notificar usuário e gerar comprovante	—	Saldo ou estorno no cartão refletido (segundo política do gateway)

Tabela 70- Fluxo de Requisitos RF06- Reembolso

#### 7.1.6.7. Roteiro de Testes TC-RF06-07- Emissão de recibo / nota fiscal

**Procedimento:** Gerar recibo e (se aplicável) NF-e.

**Pré-condições:** Pagamento concluído.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Após pagamento, gerar recibo	rideld	Recibo com ID, valor, taxa, método e timestamp
2	Se requisitado, emitir NF-e (integração com ERP)	dados fiscais	NF-e gerada e enviada por e-mail

Tabela 71- Fluxo de Requisitos RF06- Emissão de recibo

#### 7.1.6.8. Roteiro de Testes TC-RF06-08- Segurança: tokenização e não persistência de PAN

**Procedimento:** Auditar armazenamento de dados sensíveis.

**Pré-condições:** Logs e banco em ambiente de teste.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Inserir cartão e efetuar pagamento	PAN, CVV	PAN não aparece no banco; apenas token armazenado
2	Auditar logs	—	Logs não contêm PAN/CVV; apenas masked PAN (**** **** 1234)
3	Testar ataque de exposição	tentativa de dump	Sistema não retorna dados sensíveis; controles de acesso aplicados

Tabela 72- Fluxo de Requisitos RF06- Segurança

#### 7.1.6.9. Roteiro de Testes TC-RF06-09- Estorno / chargeback

**Procedimento:** Simular chargeback e checar processos (notificação, bloqueio, investigação).

**Pré-condições:** Pagamento por cartão.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Simular contestação no emissor	chargeback event	Gateway notifica sistema; ticket aberto
2	Sistema bloqueia verba do motorista até resolução	—	Status financeiro atualizado; logs de auditoria
3	Se investigação confirma fraude	—	Reembolso / reversão aplicado; ações disciplinares tomadas

Tabela 73- Fluxo de Requisitos RF06- Estorno

#### 7.1.6.10. Roteiro de Testes TC-RF06-010- Pagamento offline / agendamento

**Procedimento:** Solicitar corrida agendada e efetuar pagamento posterior/garantia.

**Pré-condições:** Corrida agendada; política de pré-autorização configurada

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário agenda corrida com pré-autorização de cartão	cartão válido	Realizar pré-autorização (hold) no valor estimado
2	No dia, confirmar cobrança final com ajuste	valor final	Captura do valor correto; release do hold remanescente
3	Se captura falha	—	Notificar usuário e tentar outras formas; cancelar corrida se não resolvido

Tabela 74- Fluxo de Requisitos RF06- Pagamento Offline

### 7.1.7. Roteiro de Testes — RF07- Cancelamento de Corrida

**Objetivo:** Validar que o sistema permite o cancelamento de uma corrida tanto por parte do usuário (tutor) quanto pelo motorista, respeitando as regras de negócio, aplicando taxas conforme o tempo e motivo, e registrando logs para auditoria.

ID	RF	Título	Prioridade	Tipo
TC-RF07-01	RF07	Cancelamento pelo usuário antes do início	CRUCIAL	Funcional
TC-RF07-02	RF07	Cancelamento pelo motorista antes da coleta	ALTO	Funcional
TC-RF07-03	RF07	Cancelamento durante o trajeto	ALTO	Operacional
TC-RF07-04	RF07	Aplicar taxa de cancelamento conforme política	CRUCIAL	Financeiro
TC-RF07-05	RF07	Registro e notificação do cancelamento	CRUCIAL	Auditoria / UX

Tabela 75- Prioridades RF07

### 7.1.7.1. Roteiro de Testes — TC-RF07-01- Cancelamento pelo usuário antes do início

**Procedimento:** Verificar se o usuário consegue cancelar uma corrida ainda não iniciada, e se o sistema processa corretamente sem penalização indevida.

**Pré-condições:** Corrida aceita, motorista ainda não a caminho ou sem status “Embarcado”.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário acessa tela “Minhas corridas”	—	Corridas listadas com status atual
2	Seleciona corrida “Agendada”	rideld válido	Exibe botão “Cancelar corrida”
3	Confirma cancelamento e insere motivo	“Mudança de planos”	Sistema registra motivo, libera motorista
4	Verifica se taxa de cancelamento se aplica	horário < tempo limite	Cancelamento sem custo; corrida removida do painel
5	Sistema envia notificação ao motorista	push/canal interno	Motorista é informado imediatamente

Tabela 76- Fluxo de Requisitos RF07- Cancelamento pelo usuário

### 7.1.7.2 Roteiro de Testes — TC-RF07-02- Cancelamento pelo motorista antes da coleta

**Procedimento:** Verificar cancelamento por parte do motorista e comunicação ao usuário.

**Pré-condições:** Corrida aceita, motorista autenticado.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista acessa corrida ativa	rideId	Botão "Cancelar corrida" disponível
2	Informa motivo do cancelamento	"Problema mecânico"	Sistema registra motivo e bloqueia reatribuição automática
3	Sistema busca novo motorista	filtro região	Novo motorista recebe solicitação
4	Usuário recebe aviso de troca	push notification	Mensagem: "Motorista cancelou, buscando novo motorista"

Tabela 77- Fluxo de Requisitos RF07- Cancelamento pelo motorista antes da coleta

### 7.1.7.3. Roteiro de Testes — TC-RF07-03- Cancelamento durante o trajeto

**Procedimento:** Simular cancelamento com corrida em andamento e validar que o sistema ajusta valores e status corretamente.

**Pré-condições:** Corrida com status “Em andamento”.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário pressiona “Cancelar corrida” durante transporte	rideld, posição atual	Sistema exibe alerta “Cancelamento durante o trajeto gera custo parcial”
2	Usuário confirma cancelamento	—	Corrida finaliza; cálculo proporcional do valor
3	Motorista recebe notificação	—	App do motorista mostra status “Cancelada pelo cliente”
4	Sistema registra logs e transação	motivo + valor	Dados gravados para auditoria e reembolso parcial

Tabela 78- Fluxo de Requisitos RF07-

Cancelamento durante o trajeto

#### 7.1.7.4. Roteiro de Testes — TC-RF07-04- Aplicar taxa de cancelamento

**Procedimento:** Testar regras de negócio para cálculo de taxas conforme tempo, tipo de corrida e parte responsável.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Corrida cancelada após motorista a caminho	$t > 2\text{min}$	Taxa de cancelamento = R\$ valor fixo / percentual
2	Corrida cancelada antes de aceitação	$t < 1\text{min}$	Sem taxa
3	Cancelamento pelo motorista sem justificativa	—	Penalização interna ou alerta administrativo
4	Registrar logs de cobrança	—	Histórico mostra "taxa aplicada" com timestamp

Tabela 79- Fluxo de Requisitos RF07- Aplicar taxa de cancelamento



### 7.1.8. Roteiro de Testes RF08- Avaliação de Motoristas

**Objetivo:** Validar que o sistema permite ao tutor avaliar o motorista após o término da corrida, armazenando nota, comentários e eventuais flags (ex: segurança, limpeza, atraso), e que as médias são recalculadas corretamente.

ID	RF	Título	Prioridade	Tipo
TC-RF08-01	RF08	Exibir tela de avaliação após corrida	CRUCIAL	UX / Funcional
TC-RF08-02	RF08	Submeter avaliação (nota + comentário)	CRUCIAL	Funcional
TC-RF08-03	RF08	Recalcular média do motorista	ALTO	Algorítmico
TC-RF08-04	RF08	Deteção de avaliação duplicada	MÉDIO	Resiliência
TC-RF08-05	RF08	Geração de alertas para notas baixas	ALTO	Operacional

Tabela 80- Prioridades RF08

### 7.1.8.1. Roteiro de Testes- TC-RF08-01- Exibir tela de avaliação

**Procedimento:** Verificar que, após o encerramento da corrida, o sistema apresenta a tela de avaliação automaticamente.

**Pré-condições:** Corrida concluída; usuário autenticado.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Corrida finalizada com status "Concluída"	rideld	Sistema redireciona usuário à tela "Avalie seu motorista"
2	Tela exibe campos de nota (1 a 5 estrelas) e comentário	—	Campos obrigatórios bem formatados
3	Usuário opta por não avaliar agora	—	Sistema permite pular, mas armazena flag "pendente"

Tabela 81- Fluxo de Requisitos RF08- Exibir tela de avaliação

### 7.1.8.2. Roteiro de Testes- TC-RF08-02- Submeter avaliação

**Procedimento:** Verificar o envio da nota e comentário ao servidor.

**Pré-condições:** Usuário autenticado; corrida finalizada.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário seleciona 4 estrelas e escreve "Motorista pontual e educado"	nota=4, comentário	Dados enviados via API com sucesso
2	Backend salva avaliação vinculada ao motorista	motoristald	Registro criado em banco; status "avaliado"
3	Motorista visualiza feedback agregado no painel	—	Atualização refletida no dashboard de avaliações

Tabela 82- Fluxo de Requisitos RF08- Submeter avaliação

### 7.1.8.3. Roteiro de Testes- TC-RF08-03- Recalcular média

**Procedimento:** Garantir que a média de avaliação seja recalculada após cada nova nota.

**Pré-condições:** Motorista já possui avaliações anteriores.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Inserir nova avaliação (nota 5)	motoristald, nota	Sistema recalcula média ponderada (últimos X dias)
2	Verificar resultado no painel administrativo	—	Nova média exibida corretamente; arredondamento ok
3	Atualização refletida na tela do motorista	—	Média visível no app imediatamente (ou após refresh)

Tabela 83- Fluxo de Requisitos RF08- Recalcular Média

### 7.1.8.4. Roteiro de Testes- TC-RF08-04- Avaliação duplicada

**Procedimento:** Evitar que o mesmo usuário avalie a mesma corrida duas vezes.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário tenta enviar nova avaliação para corrida já avaliada	rideld duplicado	Sistema bloqueia e retorna mensagem "Corrida já avaliada"
2	Logar tentativa	—	Log salvo com IP, usuário e horário

Tabela 84- Fluxo de Requisitos RF08- Avaliação duplicada

#### 7.1.8.5. Roteiro de Testes- TC-RF08-05- Geração de alertas para notas baixas

**Procedimento:** Garantir que o sistema gere alertas automáticos para notas menores que o limite (ex: < 3).

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário envia nota 2 com comentário negativo	nota=2	Sistema registra alerta para equipe de suporte
2	Motorista acumula 3 notas < 3 seguidas	—	Geração automática de alerta "Motorista sob revisão"
3	Painel administrativo exibe status "Atenção"	—	Gerente pode aplicar medidas (RF25 - Advertência)

Tabela 85- Fluxo de Requisitos RF08- Geração de alertas para notas baixas

### 7.1.9. Roteiro de Testes- RF09 — Histórico de Corridas e Pagamentos

**Objetivo:** Garantir que o sistema exiba corretamente o histórico de corridas e pagamentos realizados pelo usuário, permitindo consultas filtradas por período, tipo de pagamento, status e valores, mantendo a integridade dos dados e a rastreabilidade das transações.

ID	RF	Título	Prioridade	Tipo
TC-RF09-01	RF09	Exibir histórico de corridas concluídas	CRUCIAL	Funcional
TC-RF09-02	RF09	Filtrar histórico por período e status	ALTO	UX / Funcional
TC-RF09-03	RF09	Exibir detalhes do pagamento	CRUCIAL	Financeiro
TC-RF09-04	RF09	Validar integridade dos dados (corrida ↔ pagamento)	CRUCIAL	Auditoria
TC-RF09-05	RF09	Corrida sem pagamento associado	MÉDIO	Resiliência

Tabela 86- Prioridades RF09

### 7.1.9.1. Roteiro de Testes- TC-RF09-01- Exibir histórico de corridas concluídas

**Procedimento:** Verificar se o usuário consegue visualizar todas as corridas concluídas com suas respectivas informações básicas (data, motorista, valor, forma de pagamento e status).

**Pré-condições:** Usuário autenticado, com histórico de corridas concluídas.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário acessa o menu "Histórico"	—	Página exibe lista de corridas anteriores
2	Sistema consulta API /bichoexpress/history	userId	Retorna lista ordenada por data (mais recente primeiro)
3	Cada corrida exibe resumo: data, motorista, valor e método de pagamento	—	Dados correspondem às transações válidas no banco
4	Clique em uma corrida	rideId	Abre tela de detalhes da corrida e pagamento

Tabela 87- Fluxo de Requisitos RF09- Exibir histórico de corridas

### 7.1.9.2. Roteiro de Testes- TC-RF09-02- Filtrar histórico por período e status

**Procedimento:** Verificar funcionamento dos filtros de pesquisa no histórico.

**Pré-condições:** Usuário autenticado; histórico com múltiplas corridas.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário seleciona filtros "últimos 30 dias" e "concluídas"	filtros: período/status	Lista atualizada de acordo com critérios selecionados
2	Seleciona "canceladas"	—	Exibe apenas corridas canceladas
3	Limpa filtros	—	Exibe todas novamente

Tabela 88- Fluxo de Requisitos RF09- Filtrar histórico por período e status

### 7.1.9.3. Roteiro de Testes- TC-RF09-03- Exibir detalhes do pagamento

**Procedimento:** Validar exibição das informações financeiras de uma corrida.

**Pré-condições:** Corrida concluída e paga.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre corrida no histórico	rideld	Exibe data, valor, forma de pagamento, status "Pago"
2	Clique em "Ver comprovante"	—	Sistema exibe tela com recibo digital e QRCode Pix (quando aplicável)
3	Opção "Baixar recibo"	—	PDF gerado e baixado corretamente

Tabela 89- Fluxo de Requisitos RF09- Exibir detalhes do pagamento

#### 7.1.9.4. Roteiro de Testes- TC-RF09-04- Validar integridade dos dados

**Procedimento:** Confirmar que as corridas exibidas no histórico têm correspondência com as transações de pagamento registradas no sistema.

**Pré-condições:** Corridas registradas e banco de pagamentos sincronizado.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Acessar API de auditoria /payment-log	userId	Retorna logs de pagamentos associados
2	Comparar IDs de corrida	rideId	IDs coincidem e status "Pago" é consistente
3	Corridas sem transação	—	Exibem status "Pagamento pendente" com alerta visual

Tabela 90- Fluxo de Requisitos RF09- Validar integridade dos dados



### 7.1.10. Roteiro de Testes — RF10: Suporte via Chat Integrado

**Objetivo:**

Validar que o sistema permite comunicação em tempo real entre o usuário e o suporte administrativo ou motorista, garantindo que as mensagens sejam enviadas, recebidas, armazenadas e exibidas corretamente.

ID	RF	Título	Prioridade	Tipo
TC-RF10-01	RF10	Enviar mensagem via chat suporte	CRUCIAL	Funcional
TC-RF10-02	RF10	Receber resposta do suporte em tempo real	CRUCIAL	Comunicação
TC-RF10-03	RF10	Envio de anexo (imagem ou comprovante)	ALTO	Funcional
TC-RF10-04	RF10	Encerrar atendimento e registrar histórico	ALTO	Auditoria
TC-RF10-05	RF10	Notificação offline	MÉDIO	UX / Resiliência

Tabela 91- Prioridades RF010

### 7.1.10.1. Roteiro de Testes TC-RF10-01 — Enviar mensagem via chat suporte

**Procedimento:** Testar envio de mensagem do tutor para o suporte via chat.

**Pré-condições:** Usuário autenticado, sessão de chat iniciada

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário abre menu "Ajuda" e seleciona "Falar com suporte"	—	Janela de chat aberta
2	Digita mensagem "Preciso de ajuda com meu pagamento"	texto	Mensagem exibida na tela com status "Enviando..."
3	Sistema confirma envio (ícone ✓)	—	Mensagem salva no servidor e visível no painel do suporte
4	Suporte responde	texto resposta	Usuário visualiza mensagem recebida instantaneamente

Tabela 92- Fluxo de Requisitos RF010- Enviar mensagem via chat suporte

### 7.1.10.2. Roteiro de Testes TC-RF10-02 — Receber resposta do suporte em tempo real

**Procedimento:** Verificar a comunicação bidirecional entre usuário e suporte.

**Pré-condições:** Conexão ativa com servidor WebSocket ou API em tempo real.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário envia mensagem "Minha corrida atrasou"	texto	Mensagem aparece instantaneamente no chat
2	Suporte responde "Verificando, por favor aguarde"	texto	Mensagem recebida pelo usuário com status "entregue"
3	Mensagens antigas persistem após reabertura	—	Histórico mantido mesmo após sair do app

Tabela 93- Fluxo de Requisitos RF010- Receber resposta do suporte em tempo real

### 7.1.10.3. Roteiro de Testes TC-RF10-03 — Anexar imagem

**Procedimento:**

Verificar upload e envio de imagens no chat.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário pressiona ícone de anexo	arquivo.jpg	Sistema abre galeria do dispositivo
2	Seleciona imagem	<2MB	Upload realizado e imagem exibida no chat
3	Suporte visualiza imagem	—	Visualização preserva proporção e legenda

Tabela 94- Fluxo de Requisitos RF010- Anexar imagem

#### 7.1.10.4. Roteiro de Testes TC-RF10-04 — Registrar histórico

**Procedimento:** Verificar persistência das mensagens após o encerramento.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário encerra chat	—	Sistema exibe alerta “Deseja encerrar atendimento?”
2	Após confirmação, ticket é fechado	ticketId	Status = “Encerrado”
3	Reabre chat pelo histórico	—	Mensagens anteriores disponíveis em modo leitura

Tabela 95- Fluxo de Requisitos RF010- Registrar histórico

#### 7.1.10.5. Roteiro de Testes TC-RF10-05 — Reabrir ticket

**Procedimento:** Validar reabertura de um chat encerrado.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Usuário acessa aba “Histórico de Suporte”	—	Lista tickets encerrados
2	Seleciona ticket e pressiona “Reabrir”	ticketId	Novo chat criado vinculado ao anterior
3	Mensagens antigas copiadas para novo ticket	—	Suporte visualiza histórico completo

Tabela 96- Fluxo de Requisitos RF010- Reabrir ticket

### 7.1.11. Roteiro de Testes — RF11: Cadastro de Motorista (CNH, CRLV, antecedentes, seguro)

**Objetivo:** Garantir que o sistema permita o cadastro completo e seguro de motoristas, com upload de documentos obrigatórios e verificação automática de autenticidade.

ID	RF	Título	Prioridade	Tipo
TC-RF11-01	RF11	Cadastro básico de motorista	CRUCIAL	Funcional
TC-RF11-02	RF11	Upload de documentos obrigatórios	CRUCIAL	Segurança
TC-RF11-03	RF11	Validação automática de documentos	ALTO	Integração
TC-RF11-04	RF11	Tentativa de cadastro incompleto	ALTO	UX
TC-RF11-05	RF11	Verificação de duplicidade de CPF	CRUCIAL	Regras de negócio

Tabela 97- Prioridades RF011

#### 7.1.11.1. Roteiro de Testes TC-RF11-01 — Cadastro básico de motorista

**Pré-condições:** Aplicativo instalado; motorista acessando tela “Cadastrar Motorista”.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista acessa aba “Sou Motorista”	—	Exibe formulário de cadastro
2	Preenche nome, CPF, CNH, telefone, e-mail e senha	—	Dados validados localmente
3	Pressiona “Avançar”	—	Sistema salva dados básicos no banco

Tabela 98- Fluxo de Requisitos RF011- Cadastro básico de motorista

#### 7.1.11.2. Roteiro de Testes TC-RF11-02 — Upload de documentos obrigatórios

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista faz upload da CNH	arquivo.pdf	Documento armazenado
2	Upload de CRLV e seguro	3 arquivos	Todos os uploads concluídos
3	Sistema exibe checklist concluído	—	“Cadastro completo — aguardando validação”

Tabela 99- Fluxo de Requisitos RF011- Upload de documentos obrigatórios

### 7.1.11.3. Roteiro de Testes TC-RF11-04 — Cadastro incompleto

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista tenta avançar sem CRLV	—	Alerta “Envie todos os documentos obrigatórios”
2	Corrige e envia CRLV	arquivo	Cadastro prossegue normalmente

Tabela 100- Fluxo de Requisitos RF011- Cadastro incompleto

### 7.1.12. Roteiro de Testes — RF12: Validação por Gestor (liberação para realizar corridas)

**Objetivo:** Validar que o gestor consegue revisar e aprovar motoristas com base em documentação enviada, seguindo as regras de compliance e auditoria.

ID	RF	Título	Prioridade	Tipo
TC-RF12-01	RF12	Revisar cadastro pendente	CRUCIAL	Funcional
TC-RF12-02	RF12	Aprovar ou rejeitar motorista	CRUCIAL	Administrativo
TC-RF12-03	RF12	Solicitar nova documentação	ALTO	Fluxo alternativo
TC-RF12-04	RF12	Aprovação condicional	MÉDIO	Regras de negócio
TC-RF12-05	RF12	Log de decisão do gestor	CRUCIAL	Auditoria

Tabela 101- Prioridades RF012



### 7.1.12.1. Roteiro de Testes TC-RF12-01 — Revisar cadastro pendente

**Pré-condições:** Motorista com status “Em validação” cadastrado previamente.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Gestor acessa painel “Motoristas Pendentes”	—	Lista de cadastros pendentes exibida
2	Seleciona motorista	userId	Exibe documentos enviados
3	Analisa dados e documentos	—	Opções “Aprovar”, “Rejeitar”, “Solicitar revisão” disponíveis

Tabela 102- Fluxo de Requisitos RF012- Revisar cadastro pendente

### 7.1.12.2. Roteiro de Testes TC-RF12-02 — Aprovar ou rejeitar motorista

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Gestor pressiona “Aprovar”	—	Sistema atualiza status → “Aprovado”
2	Gestor pressiona “Rejeitar”	motivo=“documento ilegível”	Status → “Rejeitado” e mensagem enviada ao motorista

Tabela 103- Fluxo de Requisitos RF012- Aprovar ou rejeitar motorista

### 7.1.12.3. Roteiro de Testes TC-RF12-03 — Solicitar nova documentação

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Gestor identifica documento inválido	CRLV	Pressiona "Solicitar nova versão"
2	Sistema envia notificação ao motorista	—	Status atualizado para "Em correção"

Tabela 104- Fluxo de Requisitos RF012- Solicitar nova documentação

### 7.1.13. Roteiro de Testes — RF13: Indicar Status (Disponível / Ocupado / Offline)

**Objetivo:**

Validar que o motorista pode alterar seu status operacional (Disponível, Ocupado, Offline) no aplicativo, e que o sistema reflete corretamente essa informação para o backend e para os usuários que solicitam transporte.

ID	RF	Título	Prioridade	Tipo
TC-RF13-01	RF13	Alternar status "Disponível / Ocupado / Offline"	CRUCIAL	Funcional
TC-RF13-02	RF13	Atualização automática de status durante corrida	CRUCIAL	Lógica de negócio
TC-RF13-03	RF13	Bloqueio de corrida se motorista estiver Offline	ALTO	Regras de uso
TC-RF13-04	RF13	Sincronização de status com servidor	CRUCIAL	Performance
TC-RF13-05	RF13	Exibição incorreta de status (fallback)	MÉDIO	Resiliência

Tabela 105- Prioridades RF013

### 7.1.13.1. Roteiro de Testes TC-RF13-01 — Alternar status “Disponível / Ocupado / Offline”

**Pré-condições:**

Motorista autenticado e com acesso ao painel principal do aplicativo.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista acessa a tela “Meu Status”	—	Exibe os três botões: Disponível, Ocupado, Offline
2	Seleciona “Disponível”	—	Status atualizado; motorista aparece no mapa dos tutores
3	Seleciona “Ocupado”	—	Status muda; sistema bloqueia novas solicitações
4	Seleciona “Offline”	—	Motorista removido do mapa e sem receber corridas

Tabela 106- Fluxo de Requisitos RF013- Alternar status

### 7.1.13.2. Roteiro de Testes TC-RF13-02 — Atualização automática de status durante corrida

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Corrida é iniciada	—	Sistema muda status automaticamente para “Ocupado”
2	Corrida é concluída	—	Sistema redefine status para “Disponível”

Tabela 107- Fluxo de Requisitos RF013- Atualização automática de status

#### 7.1.14. Roteiro de Testes — RF14: Receber, Aceitar e Recusar Solicitações de Transporte

**Objetivo:**

Validar o processo em que o motorista recebe, analisa e aceita ou recusa solicitações de transporte, garantindo tempo de resposta, notificações adequadas e redistribuição automática da corrida em caso de recusa.

ID	RF	Título	Prioridade	Tipo
TC-RF14-01	RF14	Receber solicitação de corrida	CRUCIAL	Funcional
TC-RF14-02	RF14	Aceitar solicitação dentro do tempo limite	CRUCIAL	UX / Tempo real
TC-RF14-03	RF14	Recusar solicitação	ALTO	Fluxo alternativo
TC-RF14-04	RF14	Timeout automático (não resposta)	ALTO	Sistema
TC-RF14-05	RF14	Redistribuir corrida recusada	CRUCIAL	Lógica de negócio

Tabela 108- Fluxo de Requisitos RF014

#### 7.1.14.1. Roteiro de Testes TC-RF14-01 — Receber solicitação de corrida

**Pré-condições:**

Motorista autenticado e com status “Disponível”.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Tutor solicita corrida	dados do pet + rota	Solicitação enviada para o motorista mais próximo
2	Motorista recebe notificação push	—	Exibe tela com informações básicas (pet, origem, destino, valor estimado)
3	Cronômetro inicia (10s)	—	Sistema aguarda resposta do motorista

Tabela 109- Fluxo de Requisitos RF014- Receber solicitação de corrida

#### 7.1.14.2. Roteiro de Testes TC-RF14-02 — Aceitar solicitação dentro do tempo limite

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista pressiona “Aceitar Corrida”	—	Corrida atribuída; status → “A Caminho”
2	Tutor recebe notificação de confirmação	—	Mapa atualizado com rota e motorista designado

Tabela 110- Fluxo de Requisitos RF014- Aceitar solicitação dentro do tempo

### 7.1.14.3. Roteiro de Testes TC-RF14-03 — Recusar solicitação

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista pressiona "Recusar Corrida"	—	Solicitação é devolvida à fila
2	Sistema busca novo motorista disponível	—	Outro motorista recebe a corrida

Tabela 111- Fluxo de Requisitos RF014- Recusar solicitação

### 7.1.15. Roteiro de Testes — RF15: Acesso a Informações do Pet

**Objetivo:**

Garantir que o motorista tenha acesso às informações completas e atualizadas do pet a ser transportado, garantindo segurança e conforto durante o deslocamento.

ID	RF	Título	Prioridade	Tipo
TC-RF15-01	RF15	Exibir informações do pet após aceitar corrida	CRUCIAL	Funcional
TC-RF15-02	RF15	Ver detalhes adicionais (saúde, vacinas, cuidados)	ALTO	Informativo
TC-RF15-03	RF15	Ocultar informações pessoais do tutor	CRUCIAL	Privacidade
TC-RF15-04	RF15	Atualizar informações do pet em tempo real	ALTO	Integração
TC-RF15-05	RF15	Exibir aviso de cuidado especial	CRUCIAL	Segurança Animal

Tabela 112- Prioridades Requisitos RF015



### 7.1.15.1. Roteiro de Testes TC-RF15-01 — Exibir informações do pet após aceitar corrida

**Pré-condições:**

Corrida aceita e vinculada ao motorista.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Corrida confirmada	petId vinculado ao pedido	Sistema exibe perfil do pet
2	Exibe dados: nome, espécie, porte, peso, observações	—	Informações visíveis e legíveis
3	Tutor altera informação (ex: observação médica)	—	Atualização sincronizada em tempo real para o motorista

Tabela 113- Fluxo de Requisitos RF015- Exibir informações do pet

### 7.1.15.2. Roteiro de Testes TC-RF15-02 — Ver detalhes adicionais e cuidados

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista clica em "Ver detalhes do pet"	—	Exibe vacinação, castração, restrições alimentares
2	Caso pet tenha condição médica especial	—	Alerta visual e sonoro exibido: "Cuidado especial necessário"

Tabela 114- Fluxo de Requisitos RF015- Ver detalhes adicionais e cuidados

### 7.1.16. Roteiro de Testes — RF16: Atualizar Status da Corrida (A Caminho, Embarcado, Em Andamento, Concluído)

**Objetivo:**

Verificar se o motorista pode atualizar corretamente o status da corrida em cada etapa do transporte do pet, e se o sistema sincroniza essas alterações em tempo real com o tutor e o painel administrativo.

ID	RF	Título	Prioridade	Tipo
TC-RF16-01	RF16	Atualizar status manualmente	CRUCIAL	Funcional
TC-RF16-02	RF16	Sincronização automática com o tutor	CRUCIAL	Integração
TC-RF16-03	RF16	Envio de notificação a cada atualização	ALTO	Comunicação
TC-RF16-04	RF16	Status inconsistente (rollback)	MÉDIO	Resiliência
TC-RF16-05	RF16	Conclusão automática da corrida	ALTO	Lógica de negócio

Tabela 115- Prioridades Requisitos RF016

### 7.1.16.1. Atualizar status manualmente

**Pré-condições:**

Motorista autenticado e corrida aceita.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista abre tela da corrida ativa	—	Exibe botões: "A caminho", "Embarcado", "Em andamento", "Concluído"
2	Seleciona "A caminho"	—	Status atualizado e notificação enviada ao tutor
3	<b>Seleciona "Embarcado"</b>	—	<b>Atualiza status e adiciona registro no histórico</b>
4	Seleciona "Concluído"	—	Corrida encerrada, cálculo de valor final e avaliação habilitada

Tabela 116- Fluxo de Requisitos RF016- Atualizar status manualmente

### 7.1.17. Roteiro de Testes — RF17: Histórico de Corridas e Ganhos

**Objetivo:**

Verificar se o motorista pode acessar corretamente seu histórico de corridas realizadas e ganhos, com filtros por data, status e forma de pagamento.

ID	RF	Título	Prioridade	Tipo
TC-RF17-01	RF17	Exibir histórico completo	CRUCIAL	Funcional
TC-RF17-02	RF17	Filtrar corridas por período	ALTO	Usabilidade
TC-RF17-03	RF17	Exibir ganhos por corrida	CRUCIAL	Financeiro
TC-RF17-04	RF17	Exportar relatório de ganhos	MÉDIO	Integração

Tabela 117- Fluxo de Requisitos RF017

### 7.1.17.1. Exibir histórico completo

**Pré-condições:**

Motorista autenticado com corridas concluídas no sistema.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista acessa aba "Histórico de Corridas"	—	Exibe lista de corridas anteriores
2	Seleciona uma corrida	—	Exibe data, horário, distância, valor recebido e avaliação do tutor
3	Acessa aba "Ganhos"	—	Exibe total acumulado e filtros de data

Tabela 118- Fluxo de Requisitos RF017- Exibir histórico completo

### 7.1.18. Roteiro de Testes — RF18: Notificações sobre Corridas Agendadas

**Objetivo:**

Verificar se o motorista recebe notificações automáticas e lembretes sobre corridas agendadas, com horário, local e dados do pet.

ID	RF	Título	Prioridade	Tipo
TC-RF18-01	RF18	Receber notificação de corrida agendada	CRUCIAL	Funcional
TC-RF18-02	RF18	Lembrete 30 min antes	ALTO	Automação
TC-RF18-03	RF18	Falha na notificação	MÉDIO	Resiliência
TC-RF18-04	RF18	Sincronização com calendário	ALTO	Integração

Tabela 119- Fluxo de Requisitos RF018

#### 7.1.18.1. Roteiro de Testes TC-RF18-01 — Receber notificação de corrida agendada

**Pré-condições:**

Corrida agendada pelo tutor e confirmada pelo sistema.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Corrida é agendada	data, hora, destino	Motorista recebe notificação "Nova corrida agendada"
2	30 min antes da corrida	—	Notificação de lembrete enviada
3	Corrida iniciada	—	Notificação "Inicie a rota para o ponto de coleta"

Tabela 120- Fluxo de Requisitos RF018- Receber notificação de corrida agendada

### 7.1.19. Roteiro de Testes- RF19: Registrar Ocorrências Durante a Corrida

**Objetivo:**

Permitir que o motorista registre incidentes durante a corrida (ex.: atraso, condição do animal, problema no percurso), com descrição e, opcionalmente, fotos.

ID	RF	Título	Prioridade	Tipo
TC-RF19-01	RF19	Registrar ocorrência com texto	CRUCIAL	Funcional
TC-RF19-02	RF19	Anexar imagem à ocorrência	ALTO	Integração
TC-RF19-03	RF19	Envio automático ao setor administrativo	CRUCIAL	Comunicação
TC-RF1G-04	RF1G	Ocorrência sem descrição	MÉDIO	UX

Tabela 121- Prioridades RF019



### 7.1.19.1. Roteiro de Testes TC-RF19-01 — Registrar ocorrência

**Pré-condições:**

Corrida em andamento.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista acessa botão "Registrar Ocorrência"	—	Exibe formulário de descrição e upload de fotos
2	Preenche "Animal estressado, vomitou durante o trajeto"	+ foto	Ocorrência registrada e enviada
3	Admin recebe notificação	—	Ocorrência disponível no painel administrativo

Tabela 122- Fluxo de Requisitos RF019- Registrar ocorrência

### 7.1.20. Roteiro de Testes- RF20: Gerenciar (Cadastrar, Alterar, Excluir) Motoristas e Usuários

**Objetivo:**

Verificar se o setor administrativo pode realizar CRUD completo sobre motoristas e usuários, com logs e validações de segurança.

ID	RF	Título	Prioridade	Tipo
TC-RF20-01	RF20	Cadastrar motorista	CRUCIAL	Funcional
TC-RF20-02	RF20	Alterar dados de usuário	ALTO	Atualização
TC-RF20-03	RF20	Excluir perfil com justificativa	CRUCIAL	Regras de negócio
TC-RF20-04	RF20	Reativar conta suspensa	ALTO	Suporte
TC-RF20-05	RF20	Registrar logs de alterações	CRUCIAL	Auditoria

Tabela 122- Prioridades RF20

### 7.1.20.1. Roteiro de Testes TC-RF20-01 — Cadastrar motorista

**Pré-condições:**

Admin autenticado e com permissão de cadastro.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Admin acessa painel "Motoristas"	—	Exibe lista e botão "Cadastrar novo motorista"
2	Preenche dados: nome, CPF, CNH, CRLV, telefone	—	Validação de campos e APIs externas
3	Confirma cadastro	—	Motorista adicionado ao sistema e pendente de validação pelo gestor

Tabela 124- Fluxo de Requisitos RF20- Cadastrar motorista

### 7.1.21. Roteiro de Testes — RF21: Monitorar corridas em andamento via painel

**Objetivo:**

Garantir que operadores e gestores consigam visualizar, filtrar e acompanhar corridas em tempo real através do painel administrativo, com dados precisos (localização, status, tempo estimado, histórico parcial) e ações rápidas (entrar em contato, sinalizar incidente).

ID	RF	Título	Prioridade	Tipo
TC-RF21-01	RF21	Listar corridas em andamento no painel	CRUCIAL	Funcional
TC-RF21-02	RF21	Filtrar por área, status e prioridade	ALTO	Funcional
TC-RF21-03	RF21	Atualização em tempo real de localização e status	CRUCIAL	Tempo-Real
TC-RF21-04	RF21	<b>Abrir detalhe da corrida (mapa, ETA, motorista/usuário)</b>	ALTO	<b>Informativo</b>
TC-RF21-05	RF21	Alertas automáticos (desvios, parada longa, emergência)	CRUCIAL	Segurança/Operação

Tabela 125- Fluxo de Requisitos RF21

### 7.1.21.1. Roteiro de Testes TC-RF21-01 — Listar corridas em andamento no painel

**Pré-condições:**

Usuário operador com permissão de visualização de corridas; existem corridas com status “a caminho”, “embarque”, “em andamento”.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Operador abre painel de corridas	—	Lista é exibida com corridas ativas
2	Verifica colunas: ID, origem, destino, status, motorista, ETA	—	Todas as colunas aparecem corretamente e legíveis
3	Ordenar por tempo de início	—	Ordenação aplicada corretamente

Tabela 126- Fluxo de Requisitos RF21- Listar corridas em andamento no painel

### 7.1.21.2. Roteiro de Testes TC-RF21-02 — Filtrar por área, status e prioridade

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Operador aplica filtro “status = em andamento”	—	Apenas corridas em andamento exibidas
2	Filtra por zona/área (ex: Centro)	—	Somente corridas naquela zona aparecem
3	Combina filtros (status + área + prioridade)	—	Resultado correspondente à combinação

Tabela 127- Fluxo de Requisitos RF21- Filtrar por área, status e prioridade

### 7.1.21.3. Roteiro de Testes TC-RF21-03 — Atualização em tempo real

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Motorista atualiza localização via app (simular movimento)	GPS updates	Painel reflete nova posição em < X s (conforme SLA)
2	Motorista muda status para "embarque concluído"	status change	Painel mostra novo status imediatamente
3	Simular perda de sinal por 30s	—	Indicador "conexão instável" e último ponto conhecido exibidos

Tabela 128- Fluxo de Requisitos RF21- Atualizar em tempo real

### 7.1.21.4. Roteiro de Testes TC-RF21-04 — Abrir detalhe da corrida

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Clicar em corrida na lista	—	Painel mostra mapa, rota, histórico de posições, ETA, dados de motorista/usuário
2	Verificar botão "contatar"	—	Opção para ligar/enviar mensagem/abrir chat disponível
3	Verificar logs de eventos (paradas, desvios)	—	Logs exibidos em ordem cronológica

Tabela 129- Fluxo de Requisitos RF21- Abrir detalhe da corrida

#### 7.1.21.5. Roteiro de Testes TC-RF21-05 — Alertas automáticos

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Simular parada > X minutos fora de local previsto	—	Alerta visual e notificação enviada ao operador
2	Simular rota com desvio > Y metros	—	Alerta por desvio e opção de investigar disponível
3	Simular sinal de emergência do motorista	—	Alerta crítico e fluxo de atendimento acionado

Tabela 130- Fluxo de Requisitos RF21- Alertas automáticas

### 7.1.22. Roteiro de Testes — RF22: Intervir em disputas entre usuário e motorista

**Objetivo:**

Assegurar que a plataforma permita intervenção rápida, documentada e auditável em disputas entre usuário e motorista, incluindo coleta de evidências, comunicação, resolução temporária e escalonamento para suporte.

ID	RF	Título	Prioridade	Tipo
TC-RF22-01	RF22	Registrar disputa e abrir caso no painel	CRUCIAL	Funcional
TC-RF22-02	RF22	Coletar evidências (chat, foto, GPS, recibo)	CRUCIAL	Auditoria
TC-RF22-03	RF22	Comunicação moderada (mensagens padrões / manual)	ALTO	Comunicação
TC-RF22-04	RF22	Aplicar ações temporárias (suspensão, reembolso parcial)	CRUCIAL	Operacional
TC-RF22-05	RF22	Escalonar para níveis superiores e gerar relatório de caso	ALTO	Procedural

Tabela 131- Prioridades RF22



### 7.1.22.1. Roteiro de Testes TC-RF22-01 — Registrar disputa

**Pré-condições:** Usuário ou motorista abriu reclamação via app ou suporte; operador com permissão de intervenção.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Receber sinal de disputa (ticket automático)	Ticket recebido	Caso criado com ID único e status "aberto"
2	Vincular corrida e partes envolvidas	corridald, userld, driverld	Caso mostra informações das partes

Tabela 132- Fluxo de Requisitos RF22- Registrar disputa

### 7.1.22.2. Roteiro de Testes TC-RF22-02 — Coletar evidências

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Anexar chat / histórico de mensagens	logs	Evidência anexada no caso
2	Solicitar fotos/vídeos das partes	arquivos	Upload aceito e visível para operador
3	Incluir trajeto GPS e timestamps	GPS trace	Trajeto anexado e reproduzível no mapa

Tabela 133- Fluxo de Requisitos RF22- Coletar evidências

### 7.1.22.3. Roteiro de Testes TC-RF22-03 — Comunicação moderada

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Operador envia mensagem padrão para motorista	template	Mensagem enviada e logada no caso
2	Envia pedido de retratação ao usuário	texto livre	Mensagem entregue; recebimento confirmado
3	Registrar todas as comunicações no caso	—	Conversas armazenadas com carimbo de hora e autor

Tabela 134- Fluxo de Requisitos RF22- Comunicação moderada

### 7.1.23. Roteiro de Testes — RF23: Emitir relatórios detalhados

**Objetivo:**

Garantir que o sistema gere relatórios precisos (operações, financeiras, incidentes) com opções de filtros, formatos (PDF/CSV), agendamento e permissões.

ID	RF	Título	Prioridade	Tipo
TC-RF23-01	RF23	Gerar relatório de corridas por período	CRUCIAL	Relatório Operacional
TC-RF23-02	RF23	Gerar relatório financeiro (ganhos, comissões)	CRUCIAL	Financeiro
TC-RF23-03	RF23	Gerar relatório de incidentes e disputas	ALTO	Compliance
TC-RF23-04	RF23	Exportar em CSV / PDF / JSON	ALTO	Exportação
TC-RF23-05	RF23	Agendar relatórios periódicos e enviar por e-mail	ALTO	Automação

Tabela 135- Fluxo de Requisitos RF23

### 7.1.23.1. Roteiro de Testes TC-RF23-01 — Relatório de corridas por período

**Pré-condições:** Usuário com permissão de geração de relatórios; existem corridas no período solicitado.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Selecionar intervalo de datas (ex: 01/09/2025 - 30/09/2025)	dateFrom, dateTo	Relatório com corridas no período
2	Aplicar filtros (driverId, tipo de corrida)	filtros	Relatório filtrado corretamente
3	Visualizar resumo (total corridas, média tempo)	—	Resumo exibido com métricas corretas

Tabela 136- Fluxo de Requisitos RF23- Relatório de corridas por período

### 7.1.23.2. Roteiro de Testes TC-RF23-02 — Relatório financeiro

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Gerar relatório de ganhos por motorista	driverId, período	Valores de ganho, comissão e repasse exibidos
2	Conferir cálculos com amostra manual	amostra	Totais conferem com cálculo independente
3	Exportar relatório	formato PDF/CSV	Arquivo baixável e compatível

Tabela 137- Fluxo de Requisitos RF23- Relatório financeiro

#### 7.1.24. Roteiro de Testes — RF24: Enviar mensagens de suporte via chat interno

**Objetivo:**

Validar o chat interno entre operadores, motoristas e usuários (quando aplicável), templates de resposta, anexos, notificações e registros de conversas para auditoria.

ID	RF	Título	Prioridade	Tipo
TC-RF24-01	RF24	Iniciar chat interno com motorista/usuário	CRUCIAL	Comunicação
TC-RF24-02	RF24	Enviar/receber anexos e imagens	ALTO	Mídia
TC-RF24-03	RF24	Usar templates de resposta	ALTO	Produtividade
TC-RF24-04	RF24	Notificações push e badge de novas mensagens	CRUCIAL	Tempo-Real
TC-RF24-05	RF24	Armazenamento de transcrições para auditoria	ALTO	Auditoria

Tabela 138- Prioridades RF24

#### 7.1.24.1. Roteiro de Testes TC-RF24-01 — Iniciar chat interno

**Pré-condições:** Operador logado; motorista/usuário ativo.

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Operador clica em “Abrir chat” de uma corrida	corridaId	Janela de chat abre com histórico pré-carregado
2	Enviar mensagem de texto	texto	Mensagem enviada e aparece com timestamp
3	Receber resposta do motorista (simular)	texto	Mensagem recebida; notificação ao operador

Tabela 139- Fluxo de Requisitos RF24- Iniciar chat interno

#### 7.1.24.2. Roteiro de Testes TC-RF24-02 — Anexos e imagens

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Enviar imagem (comprovante/Foto)	arquivo imagem	Upload realizado e exibido no chat
2	Enviar arquivo grande (limitar)	25 MB	Se acima do limite, erro amigável; se dentro, aceita
3	Fazer download de anexo	—	Arquivo baixado corretamente

Tabela 140- Fluxo de Requisitos RF24- Anexo de imagens

### 7.1.25. Roteiro de Testes — RF25: Acompanhar estatísticas de avaliações

**Objetivo:**

Garantir que as métricas de avaliação (nota média, NPS, distribuição, tendências) sejam calculadas corretamente, filtráveis por período/driver/área e apresentadas em painéis e relatórios.

ID	RF	Título	Prioridade	Tipo
TC-RF25-01	RF25	Calcular nota média e total de avaliações	CRUCIAL	Métrica
TC-RF25-02	RF25	Gerar distribuição de notas e gráfico de tendência	ALTO	Visualização
TC-RF25-03	RF25	Filtrar avaliações por motorista, período e área	ALTO	Funcional
TC-RF25-04	RF25	Detectar anomalias (surto de avaliações negativas)	ALTO	Monitoramento
TC-RF25-05	RF25	Exportar métricas e integração com BI	ALTO	Integração

Tabela 141- Prioridades RF25

#### 7.1.25.1. Roteiro de Testes TC-RF25-01 — Nota média e total

**Pré-condições:** Base de avaliações com diferentes notas e comentários

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Calcular nota média de um motorista	driverId, período	Média correta (soma notas / quantidade) exibida
2	Conferência manual com amostra	amostra	Valores batem com cálculo offline

Tabela 142- Fluxo de Requisitos RF25- Nota média e total

### 7.1.26. Roteiro de Testes — RF26: Validar motoristas (Análise documental)

**Objetivo:**

Garantir que o sistema realize corretamente a análise de documentos enviados por motoristas (CNH, CRLV, antecedentes, selfie, comprovante de residência), validando autenticidade, validade e consistência das informações antes da aprovação.

ID	RF	Título	Prioridade	Tipo
TC-RF26-01	RF26	Upload de documentos obrigatórios	CRUCIAL	Funcional
TC-RF26-02	RF26	Verificação automática de validade e formato	CRUCIAL	Validação
TC-RF26-03	RF26	Validação manual por operador	ALTO	Operacional
TC-RF26-04	RF26	Notificação de pendências ou reprovação	ALTO	Comunicação
TC-RF26-05	RF26	Registro e auditoria da decisão	ALTO	Compliance

Tabela 141- Prioridades RF26



### 7.1.27. Roteiro de Testes — RF27: Suspende / bloquear motoristas ou usuários denunciados

**Objetivo:**

Assegurar que a plataforma permita suspender, bloquear e reativar motoristas ou usuários denunciados, mantendo histórico, motivo, período e notificação automática.

Tabela 141- Fluxo de Requisitos RF26

ID	RF	Título	Prioridade	Tipo
TC-RF27-01	RF27	Registrar denúncia e abrir caso	CRUCIAL	Funcional
TC-RF27-02	RF27	Aplicar suspensão temporária	CRUCIAL	Operacional
TC-RF27-03	RF27	Aplicar bloqueio permanente	CRUCIAL	Segurança
TC-RF27-04	RF27	Enviar notificações automáticas	ALTO	Comunicação
TC-RF27-05	RF27	Histórico e reversão de ações	ALTO	Auditoria

Tabela 142- Prioridade RF27

### 7.1.28. Roteiro de Testes — RF28: Configurar tarifas dinâmicas

**Objetivo:**

Validar que o sistema permita definir, aplicar e revisar tarifas dinâmicas com base em horário, demanda, região e condições climáticas, de forma transparente e auditável.

ID	RF	Título	Prioridade	Tipo
TC-RF28-01	RF28	Configurar regras de tarifa dinâmica	CRUCIAL	Funcional
TC-RF28-02	RF28	Aplicar multiplicadores automáticos	CRUCIAL	Cálculo
TC-RF28-03	RF28	Exibir tarifa dinâmica no app	ALTO	UI/Comunicação
TC-RF28-04	RF28	Registrar histórico de ajustes	ALTO	Auditoria
TC-RF28-05	RF28	Reverter regra ou ajustar limite	ALTO	Operacional

Tabela 145- Fluxo de Requisitos RF28

#### 7.1.28.1. Roteiro de Testes TC-RF28-01 — Configurar regras

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Definir regra: "+50% em horário de pico"	região, hora	Regra salva e ativa
2	Definir condição climática: "+20% chuva"	clima=chuva	Regra adicionada corretamente

Tabela 146- Fluxo de Requisitos RF28- Configurar regras

#### 7.1.28.2. Roteiro de Testes TC-RF28-02 — Aplicar multiplicadores

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Corrida iniciada durante pico	—	Tarifa com multiplicador exibida
2	Fim do pico	—	Tarifa retorna ao valor base

Tabela 147- Fluxo de Requisitos RF28- Aplicar multiplicadores

### 7.1.29. Roteiro de Testes — RF29: Criar campanhas promocionais e cupons de desconto

**Objetivo:**

Garantir que administradores possam criar, editar e aplicar campanhas e cupons promocionais com regras, validade e limites de uso corretamente controlados.

ID	RF	Título	Prioridade	Tipo
TC-RF29-01	RF29	Criar nova campanha promocional	CRUCIAL	Funcional
TC-RF29-02	RF29	Criar cupom com regras de uso	CRUCIAL	Funcional
TC-RF2G-03	RF2G	Aplicar cupom na corrida	ALTO	Integração
TC-RF29-04	RF29	Verificar limite de uso e validade	ALTO	Validação
TC-RF29-05	RF29	Gerar relatório de desempenho	ALTO	Análise

Tabela 146- Prioridades RF29

#### 7.1.29.1. Roteiro de Testes TC-RF29-01 — Criar campanha

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Criar campanha "Semana Pet Feliz"	nome, período	Campanha salva e ativa
2	Associar desconto de 20%	regra	Aplicação configurada

Tabela 148- Fluxo de Requisitos RF29- Criar campanha

#### 7.1.29.2. Roteiro de Testes TC-RF29-02 — Criar cupom

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Criar cupom "PET20" com validade 30 dias	código	Cupom criado e visível
2	Limitar uso a 2 por usuário	—	Regra armazenada corretamente

Tabela 149- Fluxo de Requisitos RF29- Criar cupom

## Roteiro de Testes — RF30: Acessar relatórios estratégicos

### Objetivo:

Validar que gestores possam gerar relatórios estratégicos (lucro, crescimento, retenção, eficiência operacional) com indicadores-chave e visualizações agregadas.

ID	RF	Título	Prioridade	Tipo
TC-RF30-01	RF30	Acesso a relatórios estratégicos	CRUCIAL	Funcional
TC-RF30-02	RF30	Filtros por período, região e categoria	ALTO	Interativo
TC-RF30-03	RF30	Exibir indicadores estratégicos (KPIs)	CRUCIAL	Analítico
TC-RF30-04	RF30	Exportar relatórios e gráficos	ALTO	Exportação
TC-RF30-05	RF30	Acesso controlado por perfil	CRUCIAL	Segurança

Tabela 150- Prioridades RF30

### 7.1.31. Roteiro de Testes — RF31: Acessar dashboards com previsões de demanda

**Objetivo:**

Assegurar que os painéis exibam previsões de demanda baseadas em dados históricos e variáveis externas (clima, eventos, horário), permitindo decisões antecipadas de alocação de motoristas.

ID	RF	Título	Prioridade	Tipo
TC-RF31-01	RF31	Exibir mapa de calor de demanda	CRUCIAL	Visualização
TC-RF31-02	RF31	Exibir previsão temporal (próximas horas/dias)	ALTO	Analítico
TC-RF31-03	RF31	Filtrar por cidade, bairro ou categoria	ALTO	Interativo
TC-RF31-04	RF31	Atualização automática em tempo real	CRUCIAL	Tempo-Real
TC-RF31-05	RF31	Exportar previsões e dados	ALTO	Exportação

Tabela 151- Fluxo de Requisitos RF31

#### 7.1.31.1. Roteiro de Testes TC-RF31-01 — Mapa de calor

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Abrir dashboard de demanda	—	Mapa com gradiente de cores exibido
2	Passar o mouse sobre uma região	—	Exibe tooltip com volume previsto e hora

Tabela 151- Fluxo de Requisitos RF31- Mapa de calor

#### 7.1.31.2. Roteiro de Testes TC-RF31-02 — Previsão temporal

Nº	Passo	Dados de Entrada	Resultado Esperado
1	Selecionar "próximas 6h"	—	Gráfico temporal exibido
2	Alterar para "próximos 3 dias"	—	Dados ajustados conforme período

Tabela 152- Fluxo de Requisitos RF31- Previsão temporal



## 8. TESTES NÃO FUNCIONAIS (RNF)

### 8.1. Testes de Confiabilidade / Disponibilidade

- **Meta:** 99.9% disponibilidade.
- **Plano:** executar testes de stress e failover (simular queda de instância EC2 e validar failover e recuperação em <15 minutos).

### 8.2. Testes de Performance / Carga

- **Cenários:**  
Pico normal: 10k usuários simultâneos.  
Evento: 100k usuários simultâneos (meta RNF33).
- **Métricas:** Latência de API < 200ms p/ endpoints críticos; consultas de busca < 2s; mensagens push < 5s.
- **Ambiente:** EC2 t3.large (homolog), Firebase em modo test, rede simulada com latência.

### 8.3. Testes de Segurança

- **Verificações:** TLS/SSL, criptografia dos dados sensíveis, no-storage de PAN (cartão), tokenização com gateway, injeção SQL/NoSQL, XSS, CSRF, autenticação forte (MFA).
- **Checklist de Pentest:** autenticação, autorização, endpoints expostos, rate limiting, logs de auditoria.
- **Conformidade:** LGPD — testes de anonimização e políticas de retenção; verificações de acesso a dados pessoais.

### 8.4. Testes de Interoperabilidade

- **Integrações:** OAuth Google, Google Calendar, API bancária, API CPF/CEP/telefone.
- **Testes:** simular respostas válidas e inválidas de cada API, timeout e retry, mapeamento de campos.

### 8.5. Testes de Compatibilidade / Portabilidade

- **Dispositivos alvo:** Android 8.0+; iOS 12+; diferentes tamanhos de tela.
- **Versões:** validar no mínimo em 3 modelos Android e 3 modelos iOS (incl. versões antigas e recentes).
- **Métrica:** app ≤ 100MB; consumo de bateria aceitável (monitorar com profiler).

#### **8.6. Testes de Recuperação / Backup**

- Simular perda de dados; validar backups diários e retenção 12 meses; testar restore ( $RTO \leq 15$  minutos).

#### **8.7. Testes de Usabilidade / Acessibilidade**

- Testes com usuários reais (etnografia) para fluxo de cadastro em  $\leq 30s$  (meta RNF22); suporte a alto contraste e leitores de tela.

## 9. Conclusão

O desenvolvimento deste projeto teve como principal objetivo a criação de uma plataforma digital inovadora para gestão, intermediação e acompanhamento de serviços de transporte e assistência voltados a pets e seus tutores, integrando recursos tecnológicos modernos e uma arquitetura sólida para garantir eficiência, segurança e confiabilidade em todas as etapas do processo.

Durante as fases de levantamento e análise de requisitos, foi possível compreender as principais dores e necessidades dos usuários — tanto dos tutores de animais quanto dos prestadores de serviço e motoristas parceiros. A partir dessas informações, definiram-se os requisitos funcionais e não funcionais, estruturando o sistema de modo que ele atendesse plenamente às demandas de acessibilidade, agilidade e transparência. Essa base conceitual orientou todas as decisões de projeto, desde o desenho da interface até a definição da arquitetura de software e infraestrutura de rede.

Na fase de prototipagem, elaborada com o apoio de ferramentas gráficas e fluxos de interação (Figma), o projeto consolidou suas principais telas e processos, como cadastro de usuários e pets, solicitação de corridas, acompanhamento em tempo real, pagamento digital, histórico de viagens e sistema de avaliações. A prototipagem validou o fluxo de uso e demonstrou a viabilidade de uma experiência intuitiva e responsiva, fator essencial para um aplicativo voltado a um público diversificado, que inclui desde motoristas profissionais até tutores que utilizam o sistema ocasionalmente.

A arquitetura de infraestrutura foi concebida com ênfase em escalabilidade e segurança. O sistema adota uma arquitetura cliente-servidor com APIs RESTful, bancos de dados relacionais para controle de usuários, pets e corridas, e integração com serviços externos, como gateways de pagamento e sistemas de geolocalização. Essa abordagem permite alta disponibilidade e modularidade, facilitando futuras expansões e integrações com novas tecnologias, como inteligência artificial e análise preditiva de demanda (RF31).

Os critérios de qualidade de software foram amplamente observados, contemplando aspectos como funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade, portabilidade e segurança. A aplicação foi projetada para operar de forma contínua, com tolerância a falhas, suporte a múltiplas plataformas (Android e iOS) e utilização de técnicas de criptografia e autenticação segura para proteção de dados pessoais e financeiros dos usuários, conforme exigências da LGPD.

Na fase de testes, elaborou-se um conjunto completo de Roteiros de Testes (RF01–RF31), abordando desde o cadastro de usuários e pets até o acompanhamento de corridas e gestão administrativa. Cada teste foi definido com procedimentos, dados de entrada, resultados esperados e critérios de aceitação, garantindo a rastreabilidade dos requisitos e o cumprimento integral das funcionalidades descritas no documento de especificação. Essa etapa demonstrou a robustez e a consistência do sistema, bem como a clareza dos fluxos operacionais.

Do ponto de vista técnico e social, o projeto contribui de forma significativa para a valorização do trabalho autônomo e da prestação de serviços com foco em segurança, empatia e bem-estar animal. Ao proporcionar uma plataforma que conecta tutores a motoristas e prestadores de confiança, o sistema amplia as oportunidades de geração de renda e promove uma relação mais transparente e humanizada entre os envolvidos.

O setor administrativo e de gestão foi igualmente fortalecido com módulos de controle avançado (RF20–RF31), que incluem monitoramento de corridas, análise documental, emissão de relatórios, campanhas promocionais, dashboards preditivos e acompanhamento de desempenho. Essas ferramentas asseguram uma visão estratégica da operação e contribuem para a tomada de decisões baseadas em dados.

Por fim, conclui-se que o desenvolvimento deste sistema representa uma solução completa, escalável e socialmente relevante, construída com base em princípios sólidos de engenharia de software e boas práticas de projeto. O trabalho atingiu plenamente seus objetivos, entregando uma aplicação funcional, confiável e aderente às necessidades identificadas no estudo inicial.

## Referências Bibliográficas

**ALMEIDA**, Marcelo de; **PEREIRA**, João Victor. *Engenharia de Software Moderna: Princípios e Práticas para o Desenvolvimento de Aplicações Escaláveis*. São Paulo: Novatec, 2022.

**BEZERRA**, Eduardo. *Princípios de Análise e Projeto de Sistemas com UML*. 3. ed. Rio de Janeiro: LTC, 2021.

**BOOCH**, Grady; **RUMBAUGH**, James; **JACOBSON**, Ivar. *UML: Guia do Usuário*. 2. ed. Porto Alegre: Bookman, 2006.

**BRASIL**. *Lei nº 13.709, de 14 de agosto de 2018*. Lei Geral de Proteção de Dados Pessoais (LGPD). Diário Oficial da União, Brasília, DF, 15 ago. 2018. Disponível em: [https://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2018/lei/l13709.htm](https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm). Acesso em: 10 out. 2025.

**COAD**, Peter; **YOURDON**, Edward. *Análise Orientada a Objetos*. Rio de Janeiro: Campus, 2000.

**ISO/IEC** 25010:2011. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. Geneva: International Organization for Standardization, 2011.

**ISO/IEC/IEEE** 29119-1:2022. *Software and Systems Engineering — Software Testing — Concepts and Definitions*. Geneva: ISO/IEC/IEEE, 2022.

**KOSCIANSKI**, André; **SOARES**, Marcelo. *Qualidade de Software: Aprenda as Metodologias e Técnicas mais Modernas de Teste, Medição e Garantia da Qualidade*. 3. ed. São Paulo: Novatec, 2021.

**PRESSMAN**, Roger S.; **MAXIM**, Bruce R. *Engenharia de Software: Uma Abordagem Profissional*. 9. ed. Porto Alegre: AMGH, 2020.

**SOMMERVILLE**, Ian. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2020.

**WAZLAWICK**, Raul Sidnei. *Análise e Projeto de Sistemas de Informação Orientados a Objetos*. 3. ed. Rio de Janeiro: Elsevier, 2020.

**FOWLER**, Martin. *Patterns of Enterprise Application Architecture*. Boston: Addison-Wesley, 2003.

**IEEE**. *IEEE Std 830-1998 — Recommended Practice for Software Requirements Specifications*. New York: IEEE Computer Society, 1998.

**GOOGLE CLOUD**. *Firebase Documentation: Authentication, Firestore, Cloud Functions and Realtime Database*. 2024. Disponível em: <https://firebase.google.com/docs>. Acesso em: 10 out. 2025.

**FIGMA**. *Figma Design and Prototyping Documentation*. 2024. Disponível em: <https://help.figma.com>. Acesso em: 10 out. 2025.

**PLANTUML**. *PlantUML Language Reference Guide*. 2024. Disponível em: <https://plantuml.com>. Acesso em: 10 out. 2025.

**GOOGLE MAPS PLATFORM**. *API Documentation: Directions, Distance Matrix and Geolocation*. 2024. Disponível em: <https://developers.google.com/maps/documentation>. Acesso em: 10 out. 2025.

**IBGE.** *Estatísticas de Lares com Animais Domésticos no Brasil — Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD)*. Brasília: IBGE, 2024. Disponível em: <https://www.ibge.gov.br>. Acesso em: 9 out. 2025.

**SEBRAE.** *Mercado Pet no Brasil: Tendências e Oportunidades para Empreendedores*. Brasília: SEBRAE Nacional, 2024. Disponível em: <https://www.sebrae.com.br>. Acesso em: 9 out. 2025.