

UNIVERSIDADE FEDERAL DE JUÍZ DE FORA  
INSTITUTO DE CIÊNCIAS EXATAS  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

# **Um aplicativo de apoio ao registro e análise de dados de avaliações discentes da UFJF**

**Luís Guilherme da Cunha Cipriani**

JUIZ DE FORA  
FEVEREIRO, 2022

# Um aplicativo de apoio ao registro e análise de dados de avaliações discentes da UFJF

LUÍS GUILHERME DA CUNHA CIPRIANI

Universidade Federal de Juíz de Fora  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Bacharelado em Sistemas de Informação

Orientador: André Luiz de Oliveira

JUIZ DE FORA  
FEVEREIRO, 2022

# UM APLICATIVO DE APOIO AO REGISTRO E ANÁLISE DE DADOS DE AVALIAÇÕES DISCENTES DA UFJF

Luís Guilherme da Cunha Cipriani

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS  
EXATAS DA UNIVERSIDADE FEDERAL DE JUÍZ DE FORA, COMO PARTE INTE-  
GRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE  
BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Aprovada por:

André Luiz de Oliveira  
Doutor em Ciência da Computação

Gleiph Ghiotto Lima de Menezes  
Doutor em Ciência da Computação

Igor de Oliveira Knop  
Doutor em Modelagem Computacional

JUIZ DE FORA  
18 DE FEVEREIRO, 2022

## Resumo

Com a ascensão de tecnologias móveis e a praticidade de *smartphones* em auxiliar tarefas cotidianas, a utilização desses dispositivos tem aumentado e impactado em mudanças culturais na sociedade, seja para receber, produzir ou compartilhar conteúdos entre indivíduos nos mais diversos ambientes. Na área de ensino, a tecnologia móvel também se faz presente, como na UFJF, a qual dispõe de um aplicativo para fornecer informações a seus alunos, como também redes sociais e uma plataforma de educação à distância. No tocante ao processo de avaliação institucional docente, as avaliações discentes são realizadas por meio de formulários, os quais não garantem com exatidão qual usuário realizou o preenchimento, visto que não há garantia ou confirmação da autenticidade de quem o acessa. Para solucionar esses problemas, foi desenvolvido o Sistema de Avaliações Discentes (SAD-UFJF), que consiste em um aplicativo móvel de apoio à coleta e análise de dados das avaliações discentes da UFJF, com base em métodos de análise, avaliação da Interação Humano-Computador e boas práticas de *design*. Testes funcionais e avaliação heurística foram realizados para avaliar a corretude e a usabilidade das funcionalidades desenvolvidas. Como resultado, foi constatado que as *interfaces* do aplicativo foram projetadas em aderência aos princípios de *design*. Dessa forma, contribuindo para a usabilidade e para o aprimoramento da experiência do usuário.

**Palavras-chave:** Usabilidade, tecnologias para dispositivos móveis, *design* de *interfaces*, princípios e diretrizes de *design*, segurança da informação.



# Abstract

With the rise of mobile technologies and the practicality of *smartphones* in supporting everyday tasks, the use of smartphones has increased and impacted on cultural changes in society, whether to receive, produce or sharing content between individuals in the most diverse environments. Mobile technology is also present in the education field. UFJF has a mobile application, as well as social network profiles, and an e-learning platform that provides information to its students. However, with respect to the institutional teaching evaluation process, student evaluations are carried out through electronic forms, which does not guarantee the authenticity of the identity of the users who filled the forms, since there is no guarantee or confirmation of the authenticity of those who access the forms. In order to solve these problems, the Student Evaluation System (SAD-UFJF) was developed as a mobile application to support the collection and analysis of teaching quality evaluation data provided by students. The application is built upon analysis, design, and evaluation methods from Human-Computer Interaction and good user interface *design* practices and principles for mobile devices. Functional tests and heuristic evaluation were performed to evaluate the correctness and the usability of the application's capabilities. The results demonstrate that the application's graphical user interfaces were designed in compliance with design principles, contributing to usability and improve the user's experience.

**Keywords:** Usability, technologies for mobile devices, design of interfaces, design principles and guidelines, information security.

## Agradecimentos

A Deus, pela minha vida, por me ajudar a alcançar meus objetivos e ultrapassar todos os obstáculos encontrados ao longo do caminho.

Aos meus pais e familiares, pelo incentivo e todo apoio necessário. Principalmente ao meu irmão, que sempre esteve ao meu lado nos momentos difíceis.

Ao professor André pela orientação, amizade e ensinamentos, os quais tornaram possível a realização deste trabalho.

Aos colegas e professores do Departamento de Ciência da Computação pelos seus ensinamentos, que durante esses anos, contribuíram para meu enriquecimento pessoal e profissional.

*“Sonhos determinam o que você quer. Ação  
determina o que você conquista”.*

*Aldo Novak*

# Conteúdo

<b>Lista de Figuras</b>	<b>7</b>
<b>Lista de Tabelas</b>	<b>11</b>
<b>Lista de Abreviações</b>	<b>12</b>
<b>1 Introdução</b>	<b>13</b>
<b>2 Fundamentação Teórica</b>	<b>16</b>
2.1 Tecnologias de Informação e Comunicação . . . . .	16
2.2 Interação Humano-Computador . . . . .	17
2.3 Perspectivas de Interação . . . . .	19
2.4 Qualidade de Uso . . . . .	22
2.5 Usabilidade <i>WEB</i> e Móvel . . . . .	23
2.5.1 Usabilidade <i>WEB</i> . . . . .	24
2.5.2 Usabilidade Móvel . . . . .	25
2.6 Princípios e Diretrizes de <i>Design</i> . . . . .	25
2.6.1 Princípios da <i>Gestalt</i> . . . . .	26
2.6.2 Regras de Ouro de Schneiderman . . . . .	29
2.6.3 Processo de Engenharia de Usabilidade de Nielsen . . . . .	30
2.6.4 Heurísticas de <i>Design</i> de Nielsen . . . . .	32
2.7 Engenharia Cognitiva . . . . .	33
2.8 Arquitetura de Aplicações <i>Web</i> e Móveis . . . . .	35
2.9 Tecnologias Utilizadas no Desenvolvimento <i>Web</i> e Móvel . . . . .	36
2.9.1 <i>React Native</i> . . . . .	37
2.9.2 <i>Node.JS</i> . . . . .	38
2.9.3 <i>Express.JS</i> . . . . .	39
2.9.4 <i>MongoDB</i> . . . . .	40
2.9.5 <i>Mongoose</i> . . . . .	41
2.9.6 <i>Heroku</i> . . . . .	41
2.10 Processo de Avaliação Discente no DCC/UFJF . . . . .	43
2.11 Considerações Finais . . . . .	46
<b>3 Metodologia: O Processo</b>	<b>47</b>
3.1 Visão Geral do Processo . . . . .	47
3.2 Fase 1: Análise da Situação Atual . . . . .	48
3.2.1 Escopo do Sistema . . . . .	48
3.2.2 Metas de <i>design</i> . . . . .	49
3.2.3 Caracterização dos Usuários . . . . .	49
3.2.4 Requisitos Funcionais . . . . .	49
3.2.5 Requisitos Não Funcionais . . . . .	50
3.2.6 Especificação dos Requisitos Funcionais . . . . .	51
3.2.7 Especificação dos Requisitos Não Funcionais . . . . .	52
3.3 Fase 2: Projeto da Intervenção . . . . .	53

3.3.1	Especificação dos Cenários para Detalhar os Requisitos Funcionais do Sistema . . . . .	53
3.3.2	Modelagem de Tarefas . . . . .	57
3.3.3	Prototipação dos Cenários . . . . .	58
3.4	Avaliação de IHC . . . . .	59
3.5	Considerações Finais . . . . .	60
<b>4</b>	<b>SAD-UFJF: Sistema de Avaliação Discente</b>	<b>62</b>
4.1	Arquitetura do Aplicativo . . . . .	62
4.2	Modelo Conceitual . . . . .	65
4.3	Visão Geral do <i>Back-End</i> . . . . .	67
4.3.1	Implementação da Camada Modelo . . . . .	70
4.3.2	Implementação da Camada Controle . . . . .	73
4.4	Visão Geral do <i>Front-End</i> . . . . .	75
4.4.1	Descrição das Telas do SAD-UFJF . . . . .	76
4.4.2	Implementação da Camada Visão . . . . .	85
4.5	Cenários de Uso do Aplicativo SAD-UFJF . . . . .	88
4.5.1	Preenchimento de Questionário de Avaliação Discente . . . . .	89
4.5.2	Visualização de Gráfico de Barras de Avaliação de Desempenho Docente . . . . .	91
4.6	Considerações Finais . . . . .	92
<b>5</b>	<b>Fase 3: Avaliação da Intervenção</b>	<b>93</b>
5.1	Teste Funcional . . . . .	93
5.1.1	Preparação . . . . .	93
5.1.2	Execução e Coleta de Dados . . . . .	94
5.1.3	Análise e Conclusão dos Resultados . . . . .	96
5.2	Avaliação Heurística . . . . .	103
5.2.1	Preparação . . . . .	103
5.2.2	Coleta de Dados . . . . .	103
5.2.3	Consolidação dos Resultados . . . . .	108
5.3	Considerações Finais . . . . .	110
<b>6</b>	<b>Conclusão e Trabalhos Futuros</b>	<b>112</b>
	<b>Bibliografia</b>	<b>114</b>
<b>A</b>	<b>Apêndice</b>	<b>117</b>
A.1	Requisitos funcionais do Aplicativo SAD-UFJF . . . . .	117
<b>B</b>	<b>Apêndice</b>	<b>118</b>
B.1	Especificação dos Requisitos Funcionais do Aplicativo SAD-UFJF . . . . .	118
B.2	Especificação dos Requisitos não Funcionais do Aplicativo SAD-UFJF . . . . .	135
<b>C</b>	<b>Apêndice</b>	<b>137</b>
C.1	Prototipação dos cenários com utilização da ferramenta <i>Balsamiq</i> . . . . .	137
C.2	<i>Interfaces</i> desenvolvidas para os cenários da aplicação . . . . .	138
<b>D</b>	<b>Apêndice</b>	<b>140</b>
D.1	Diagrama de Caso de Uso . . . . .	140
D.2	Diagramas de Atividades . . . . .	140

## Lista de Figuras

2.1	Perspectiva de Sistema. Fonte: (KAMMERGAARD, 1988)	20
2.2	Perspectiva de Parceiro do Discurso. Fonte: (KAMMERGAARD, 1988)	21
2.3	Perspectiva de Ferramenta. Fonte: (KAMMERGAARD, 1988)	21
2.4	Perspectiva de Mídia. Fonte: (KAMMERGAARD, 1988)	22
2.5	Princípio da Proximidade. Fonte: Nike.com	26
2.6	Princípio da Continuidade. Fonte: (HOFSTETTER, 2019)	27
2.7	Princípio da Simetria. Fonte: Nike.com	27
2.8	Princípio da Similaridade. Fonte: Nike.com	27
2.9	Princípio do Destino Comum. Fonte: Microsoft Word	28
2.10	Princípio do Fecho. Fonte: (HOFSTETTER, 2019)	28
2.11	Princípio da Região Comum. Fonte: (Aela Editorial, )	29
2.12	Princípio da Conectividade. Fonte: (HOFSTETTER, 2019)	29
2.13	Heurística de Consistência e Padronização no Pacote Microsoft Office.	33
2.14	Estágios de atividade do usuário na travessia dos golfos de execução e de avaliação. Fonte: (BARBOSA; SILVA, 2017)	34
2.15	Exemplo de um cenário com aplicação da Teoria dos Golfos	35
2.16	Sequência de passos na interação do usuário com o sistema para a especificação de um cenário	35
2.17	Padrão de Arquitetura de Software MVC. Fonte: (TreinaWeb, 2021)	36
2.18	Arquitetura <i>React Native</i> . Fonte: (Rocketseat, apud Medium, 2019)	38
2.19	Arquitetura <i>Node.JS</i> . Fonte: (Kinsta, 2021)	39
2.20	Um fluxo de chamada de solicitação/resposta HTTP típico com Node e Express. Fonte: Turing School 2019.	40
2.21	Comparação de termos entre MongoDB e Bancos de Dados Relacionais. Fonte: (Alura, 2021)	41
2.22	Tempo de execução do <i>Heroku</i> : do código aos usuários finais. Fonte: <i>Heroku website</i>	42
2.23	Critérios de avaliação de desempenho docente em disciplinas	44
2.24	Critérios de avaliação de desempenho discente em disciplinas	44
2.25	Exemplo de tabela com dados das avaliações discentes	45
2.26	Formulário de avaliação docente em estágio probatório via <i>Google Forms</i>	46
3.1	Fases das atividades do processo adotadas para desenvolver o aplicativo. Fonte: Autoria Própria	48
3.2	Modelo de tarefas para a meta geral de submeter questionário. Fonte: Autoria Própria.	58
3.3	Protótipo da tela de autenticação. Sistema SAD-UFJF. Fonte: Autoria própria	59
3.4	Protótipo área do discente. Sistema SAD-UFJF. Fonte: Autoria própria	59
3.5	Protótipo área do docente. Sistema SAD-UFJF. Fonte: Autoria própria	59

3.6	Protótipo do menu lateral discente. Sistema SAD-UFJF. Fonte: Autoria própria . . . . .	59
4.1	Arquitetura do sistema para o funcionamento do <i>software</i> . Fonte: Autoria Própria. . . . .	63
4.2	Modelo entidade-relacionamento do SAD-UFJF. . . . .	65
4.3	Formato <i>JSON</i> . Fonte: Autoria Própria. . . . .	69
4.4	Diagrama de classes de entidade do aplicativo. . . . .	70
4.5	Trecho de código NodeJS da classe entidade <i>Questionário</i> do sistema SAD-UFJF. . . . .	73
4.6	Trecho de código NodeJS da classe <i>controller</i> para a entidade <i>Questionário</i> do sistema SAD-UFJF. . . . .	75
4.7	Modelo navegacional das telas do sistema SAD-UFJF. . . . .	76
4.8	Tela de login do sistema SAD-UFJF. . . . .	77
4.9	Área do usuário discente. . . . .	78
4.10	Tela de questionários de avaliação não respondidos. . . . .	79
4.11	Tela de questionários de avaliação submetidos. . . . .	80
4.12	Tela de preenchimento de questionário de avaliação de disciplina. . . . .	81
4.13	Área do usuário docente. . . . .	82
4.14	Tela de seleção de disciplinas usuário docente. . . . .	83
4.15	Tela de exibição de gráfico de barras de avaliação de disciplina. . . . .	84
4.16	Tela de exibição de relatório de avaliação de disciplina. . . . .	85
4.17	Implementação do <i>front-end</i> para a tela de questionário do Sistema SAD-UFJF. . . . .	87
4.18	Implementações do <i>updateState</i> e <i>SaveState</i> no <i>front-end</i> para a tela de questionário do Sistema SAD-UFJF. . . . .	87
4.19	Modelo navegacional das telas do aplicativo. . . . .	89
4.20	Sequência de passos necessários para realizar o envio de um questionário de avaliação discente. . . . .	90
4.21	Sequência de passos necessária para gerar um gráfico de avaliação de desempenho docente em uma disciplina. . . . .	91
5.1	Escala do grau de satisfação. Fonte: (LIKERT, 1932 apud ALMEIDA JUNIOR., 2017). . . . .	94
5.2	Gráfico de barras com destaque nos valores de <i>ranking</i> médio obtidos. SAD-UFJF Fonte: Autoria Própria. . . . .	102
5.3	Gráfico de barras com destaque nos valores de <i>ranking</i> médio obtidos. SAD-UFJF Fonte: Autoria Própria. . . . .	102
5.4	Gráfico de barras com destaque nos valores de <i>ranking</i> médio obtidos. SAD-UFJF Fonte: Autoria Própria. . . . .	102
5.5	Questionário avaliativo. SAD-UFJF Fonte: Autoria Própria. . . . .	105
5.6	Mensagem de confirmação de envio do questionário. SAD-UFJF Fonte: Autoria Própria. . . . .	105
5.7	Menu lateral do discente. SAD-UFJF Fonte: Autoria Própria. . . . .	106

5.8	Tela inicial do docente. SAD-UFJF	
	Fonte: Autoria Própria. . . . .	106
5.9	Tela de seleção de disciplinas. SAD-UFJF	
	Fonte: Autoria Própria. . . . .	107
5.10	Tela de geração do gráfico de barras. SAD-UFJF	
	Fonte: Autoria Própria. . . . .	107
C.1	Protótipo da tela de autenticação. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	137
C.2	Protótipo área do discente.	
	Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	137
C.3	Protótipo área do docente.	
	Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	137
C.4	Protótipo do menu lateral discente. Sistema SAD-UFJF. Fonte: Autoria	
	própria . . . . .	137
C.5	Protótipo consulta disciplinas. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	138
C.6	Protótipo questionário avaliativo. Sistema SAD-UFJF. Fonte: Autoria	
	própria . . . . .	138
C.7	Protótipo área de comentário.	
	Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	138
C.8	Protótipo gráfico barra.	
	Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	138
C.9	Tela de autenticação. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	138
C.10	Área do discente.	
	Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	138
C.11	Seleção de questionário. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	138
C.12	Questionário avaliativo. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	138
C.13	Área do docente. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	139
C.14	Tela consulta de disciplinas. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	139
C.15	Geração de gráfico. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	139
C.16	Perguntas do questionário. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	139
C.17	Geração de relatório. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	139
C.18	Menu lateral discente. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	139
C.19	Questionários concluídos. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	139



C.20	Busca questionário. Sistema SAD-UFJF.	
	Fonte: Autoria própria . . . . .	139
D.1	Diagrama de casos de uso para o sistema. Fonte: Autoria Própria. . . . .	140
D.2	Diagrama de atividades para o requisito funcional 'RF01'. Fonte: Autoria Própria. . . . .	141
D.3	Diagrama de atividades para os requisitos funcionais 'RF04', 'RF05', 'RF06' e 'RF07'. Fonte: Autoria Própria. . . . .	142

## Lista de Tabelas

3.1	Requisitos funcionais do sistema . . . . .	50
3.2	Requisitos não funcionais do sistema . . . . .	50
3.3	Especificação do requisito funcional 'RF07' . . . . .	51
3.4	Especificação do requisito não funcional 'RNF03' . . . . .	53
3.5	Questões referentes ao docente, presentes no questionário avaliativo . . . .	56
3.6	Questões referentes ao discente, presentes no questionário avaliativo . . . .	57
4.1	Comparação dos formatos <i>JSON</i> e <i>BSON</i> . Fonte: Adaptado de (MongoDB, 2022a) . . . . .	70
5.1	Respostas dos alunos matriculados na turma que responderam o questionário	97
5.2	Cálculos realizados para as questões presentes no questionário . . . . .	101
5.3	Heurísticas de Nielsen e Princípios da <i>Gestalt</i> utilizados nas análises . . . .	104
5.4	Relatório com resultados da Avaliação Heurística . . . . .	109
A.1	Requisitos funcionais do sistema . . . . .	117
B.1	Especificação do requisito funcional 'RF01' . . . . .	118
B.2	Especificação do requisito funcional 'RF02' . . . . .	120
B.3	Especificação do requisito funcional 'RF03' . . . . .	121
B.4	Especificação do requisito funcional 'RF04' . . . . .	122
B.5	Especificação do requisito funcional 'RF05' . . . . .	123
B.6	Especificação do requisito funcional 'RF06' . . . . .	124
B.7	Especificação do requisito funcional 'RF07' . . . . .	125
B.8	Especificação do requisito funcional 'RF08' . . . . .	126
B.9	Especificação do requisito funcional 'RF09' . . . . .	127
B.10	Especificação do requisito funcional 'RF10' . . . . .	128
B.11	Especificação do requisito funcional 'RF11' . . . . .	129
B.12	Especificação do requisito funcional 'RF12' . . . . .	130
B.13	Especificação do requisito funcional 'RF13' . . . . .	131
B.14	Especificação do requisito funcional 'RF14' . . . . .	132
B.15	Especificação do requisito funcional 'RF15' . . . . .	133
B.16	Especificação do requisito funcional 'RF16' . . . . .	134
B.17	Especificação do requisito funcional 'RF17' . . . . .	135
B.18	Especificação do requisito não funcional 'RNF01' . . . . .	135
B.19	Especificação do requisito não funcional 'RNF02' . . . . .	136
B.20	Especificação do requisito não funcional 'RNF03' . . . . .	136

## Lista de Abreviações

SAD-UFJF	Sistema de Avaliação Discente
UFJF	Universidade Federal de Juiz de Fora
DCC	Departamento de Ciência da Computação
CPADs	Comissões Permanentes de Avaliação Docente
TCC	Trabalho de Conclusão de Curso
SIGA	Sistema Integrado de Gestão Acadêmica
TICs	Tecnologias de Informação e Comunicação
IHC	Interação Humano-Computador
MVC	<i>Model-View-Controller</i>
SQL	<i>Structured Query Language</i>
NoSQL	<i>Not Only SQL</i>
API	<i>applications protocol interface</i>
JSON	<i>JavaScript Object Notation</i>
SGBD	Sistema Gerenciador de Banco de Dados
UML	<i>Unified Modeling Language</i>

# 1 Introdução

A rápida e crescente evolução tecnológica contribuiu para o aumento da praticidade de muitas atividades cotidianas devido a redução dos custos e consequentemente expansão do acesso à dispositivos móveis com o surgimento e popularização dos *smartphones*. *Smartphones* são utilizados para auxiliar pessoas na realização de diversas tarefas diárias devido ao seu tamanho compacto, que o torna fácil de ser transportado, além de sua alta capacidade de processamento e armazenamento de dados, conectividade com a *Internet* e pela sua possibilidade de utilização nos mais diversos tipos de aplicações, como *Internet Banking*, *e-commerce*, redes sociais, jogos etc (SILVA et al., 2020).

No ambiente universitário os *smartphones* também são utilizados com frequência, seja nas relações sociais ou no acesso à recursos acadêmicos, como materiais de aula, a visualização de enunciados de atividades, realização de aulas *on-line* e consultas à informações acadêmicas, como: disciplinas oferecidas, disciplinas cursadas e matriculadas, e visualização de notas obtidas em avaliações (VEBER et al., 2020). A Universidade Federal de Juiz de Fora (UFJF), desde o ano de 2018, disponibiliza seu próprio aplicativo para dispositivos móveis, o qual visa ampliar os serviços da instituição à comunidade e seus alunos (UFJF, 2018). Entretanto, o processo de avaliação institucional docente, que consiste na coleta e análise dos resultados de avaliações discentes de disciplinas dos cursos de graduação da Universidade são realizados de forma independente em cada Departamento da Instituição e na maioria dos casos com base no preenchimento de formulários eletrônicos, como ocorre no Departamento de Ciência da Computação (DCC).

Os formulários de avaliação disciplinas são disponibilizados na forma de questionários *on-line* e incluem requisitos e critérios de avaliação da qualidade de ensino relacionados à uma disciplina cursada, como também de autoavaliação de desempenho do discente respondente. Ao se aproximar do término de um período letivo, os discentes são convidados a preencher esses questionários através do sistema *Google Forms*, utilizado para a coletar informações relacionadas à avaliação da qualidade do ensino de uma determinada disciplina ministrada por um docente, que posteriormente servem de base para

avaliar o desempenho docente em atividades de ensino.

O preenchimento de formulários de avaliação discente de disciplinas da UFJF, cujas respostas são posteriormente armazenadas em planilhas eletrônicas editáveis, como o *Google Forms*, possui vulnerabilidades tais como a dificuldade de garantir a autenticidade da identidade do discente quem preencheu o formulário. Caso tais vulnerabilidades sejam exploradas, podem violar a confidencialidade e a integridade dos dados obtidos e comprometer todo o processo de avaliação dos docentes da instituição. Além disso, a disponibilização desses dados em planilhas eletrônicas é uma ameaça à confidencialidade e a segurança do processo de avaliação discente, uma vez que não há garantias de que os dados informados pelos discentes não serão alterados por pessoas não autorizadas, violando a integridade dos dados.

O processo de coleta de dados de avaliação discente baseado em formulários demanda um alto esforço de docentes membros de Comissões Permanentes de Avaliação Docente (CPADs), que analisam os dados das avaliações de cada docente e elaboram manualmente, para cada disciplina ministrada, tabelas com as médias das respostas obtidas dos discentes para cada critério avaliativo e gráficos de barras a partir desses dados, e posteriormente disponibilizam essas informações a cada docente da instituição. Tal processo demanda tempo, é altamente propenso a erros e não garante a transparência e a disponibilidade dos dados das avaliações, tanto aos discentes quanto aos docentes avaliados.

Com o objetivo de aumentar a confidencialidade no processo de coleta de dados de avaliações da qualidade de ensino de disciplinas realizadas por alunos e automatizar a geração de relatórios avaliativos de disciplinas ministradas por docentes da UFJF, neste trabalho é proposto um protótipo de aplicativo de apoio a coleta de dados e análise de dados de avaliações discentes. O aplicativo proposto possui compatibilidade com as plataformas operacionais de *smartphones Android* e *IOs*, e poderá contribuir para o fortalecimento do Processo de Avaliação Institucional Docente da UFJF.

O aplicativo foi desenvolvido com base em princípios e diretrizes de *design* da Interação Humano-Computador e no processo de Engenharia de Usabilidade (NIELSEN, 1993), nos quais foram realizadas atividades nas fases de Análise do Domínio do Problema, Projeto da Intervenção e Avaliação da Intervenção de *Design*. Na fase de análise foram

realizadas seções de *brainstorm* para caracterizar os usuários e identificar os requisitos do sistema. Posteriormente, na fase *design*, cada requisito funcional do sistema foi detalhado em um cenário e modelo de tarefas. A ferramenta de prototipação *Balsamiq* também foi utilizada na especificação dos protótipos de cada cenário, que posteriormente foram implementados utilizando-se os *frameworks React Native* para *front-end* e *Node JS* para *back-end*. Na fase de avaliação, a corretude das funcionalidades desenvolvidas foi verificada por meio de Testes Funcionais e a usabilidade das *interfaces* foi avaliada por meio do método de Avaliação Heurística (NIELSEN, 1993). A versão protótipo do aplicativo desenvolvida neste Trabalho de Conclusão de Curso de Graduação tem o potencial de ser posteriormente integrada ao Sistema de Gestão Acadêmica (SIGA) e ao aplicativo institucional da UFJF.

Esta monografia está organizada em seis capítulos. No **Capítulo 2** são apresentados os conceitos relacionados à Interação Humano-Computador, Princípios e Diretrizes de *design*, Engenharia Cognitiva, Arquitetura de Aplicações *Web* e tecnologias de desenvolvimento de aplicações móveis necessários à compreensão das contribuições deste trabalho. No **Capítulo 3** é descrito o processo de desenvolvimento do aplicativo. No **Capítulo 4** é apresentada a arquitetura do Aplicativo de apoio à coleta e análise de dados das avaliações discentes da UFJF. No **Capítulo 5** são descritos os testes funcionais realizados para avaliar a corretude das funcionalidades do protótipo e a aplicação do método de Avaliação Heurística para avaliar a usabilidade de telas desenvolvidas. No **Capítulo 6** são apresentadas as conclusões e as propostas de trabalhos futuros.

## 2 Fundamentação Teórica

Neste capítulo são apresentados os conceitos necessários à compreensão das contribuições deste Trabalho de Conclusão de Curso. Na Seção 2.1 é apresentado o conceito de Tecnologias de Informação e Comunicação. Os conceitos básicos da IHC são descritos na Seção 2.2. Na Seção 2.3 são apresentadas as quatro perspectivas da Interação Humano-Computador. Cada perspectiva descreve uma forma pela qual o usuário pode interagir com um sistema computacional interativo. Na Seção 2.4 é apresentado o conceito e os critérios de qualidade de uso. Na Seção 2.5 é fornecida uma visão geral dos princípios de Usabilidade na *web* e em dispositivos móveis. Na Seção 2.6 são descritas as boas práticas de *design* de *interfaces*. Na Seção 2.7 é fornecida uma visão geral da teoria de IHC da Engenharia Cognitiva. Uma visão geral da arquitetura de aplicações *web* e móveis é apresentada na Seção 2.8, com destaque ao padrão *Model-View-Controller* utilizado no *design* de aplicações *WEB* e de dispositivos móveis. Na Seção 2.9 são apresentadas as tecnologias de desenvolvimento de aplicações web e móveis *React Native*, *NodeJS* e o Banco de Dados *MongoDB* utilizados no desenvolvimento deste trabalho de conclusão de curso.

### 2.1 Tecnologias de Informação e Comunicação

As Tecnologias de Informação e Comunicação (TICs) estão presentes em vários momentos da vida cotidiana das pessoas, e.g., no ensino, transações bancárias, compras em comércio eletrônico etc. Atualmente, diversos dispositivos são capazes de processar informações, não apenas computadores tradicionais, mas também outros dispositivos eletrônicos oriundos dos mais variados ambientes. Exemplos de tais dispositivos incluem consoles de jogos, TV digital interativa, rádios digitais, *smartphones*, dentre outros (BARBOSA; SILVA, 2017).

A introdução de TICs vêm modificando a forma como as pessoas se comunicam. Atualmente, as pessoas costumam recorrer a programas e aplicativos para interagir com

outros indivíduos, seja na troca de mensagens por meio de *e-mails*, redes sociais como *Facebook*, sistemas de comunicação de forma síncrona ou assíncrona como *Whatsapp*, ou simplesmente um telefone celular. Essas novas formas de comunicação permitem que as pessoas se comuniquem através de textos, vídeos e áudios, e compartilhem informações nos mais variados formatos como fotos, músicas e vídeos. A introdução das TICs impactaram a sociedade de forma significativa, modificando diversos campos como, por exemplo, a política, o comércio, os transportes e a saúde. As TICs também tem alterado o ambiente educacional devido a possibilidade de uma comunicação mais dinâmica, na qual a informação contida na *Internet* possibilita um aprendizado que vai além do conhecimento restrito apenas aos livros ou as salas de aula, permitindo a professores e alunos uma comunicação mais interativa por meio do apoio computacional fornecido (BARBOSA; SILVA, 2017).

Como as TICs transformaram os processos de interação e comunicação entre as pessoas, é fundamental que estas sejam desenvolvidas com atenção aos efeitos que sua introdução podem causar nas tarefas cotidianas dos usuários. Dessa forma, os desenvolvedores de TICs devem implementar mecanismos que protejam os usuários quanto a possíveis falhas humanas, devido ao acionamento não intencional de uma funcionalidade que possui uma consequência catastrófica, como formatação de disco ou remoção de arquivos, e falhas computacionais. As TICs devem ser projetadas de modo a prever o comportamento dos usuários, fornecendo respostas explícitas e não ambíguas a cada ação realizada (BARBOSA; SILVA, 2017).

## 2.2 Interação Humano-Computador

A Interação Humano-Computador (IHC) é uma área de conhecimento que possui o enfoque na análise, projeto, implementação e avaliação de sistemas computacionais interativos para uso humano e no estudo dos fenômenos relacionados a este uso (HEWETT et al., 1992). A disciplina de IHC toma como objetos de estudo: a **Natureza da Interação das Pessoas com Sistemas Computacionais**, o **Contexto no qual Sistemas Interativos são Utilizados** (características do contexto como a estrutura do trabalho e organização social que impactam no uso do sistema), as **Características Humanas**



(crenças, preferências, capacidades e limitações cognitivas, perceptivas e motoras, a linguagem, a comunicação e as formas de interação) que impactam no uso de sistemas computacionais interativos, a **Arquitetura de Sistemas e da Interface com o Usuário** (características dos dispositivos de entrada e saída) e **Processos de Desenvolvimento** com o enfoque na qualidade de uso. Os processos de desenvolvimento de sistemas computacionais interativos incluem: abordagens, princípios e diretrizes de *design*, ferramentas e técnicas de implementação e métodos de avaliação da qualidade de uso de interfaces de sistemas computacionais interativos (BARBOSA; SILVA, 2017).

IHC é uma área multidisciplinar que envolve conhecimentos de outras áreas além da Computação para melhor compreender os fenômenos envolvidos no uso de sistemas computacionais interativos. Por exemplo, os conhecimentos da Psicologia, Sociologia e da Antropologia contribuem para adquirir conhecimentos sobre a cultura e comportamentos dos usuários nos ambientes onde desempenham suas atividades individuais e em grupo. Os conhecimentos das áreas do *Design*, Ergonomia, Linguística e Semiótica são utilizados no projeto das interfaces com o usuário.

O estudo dos fenômenos de interação entre seres humanos e sistemas computacionais interativos possibilita uma melhor compreensão do processo de interação humano-computador para que seja possível melhorar a concepção, o *design* e a introdução de TICs no cotidiano das pessoas, com melhorias na experiência de uso (BARBOSA; SILVA, 2017). Conhecimentos de IHC permitem um melhor entendimento das características humanas e dos recursos computacionais para o desenvolvimento de sistemas interativos que possam contribuir para o aumento da produtividade e satisfazer as necessidades dos usuários. O aumento da qualidade de uso de sistemas computacionais interativos contribui para a melhoria da experiência do usuário, compreendendo características de interação e de *interface* que proporcionam maior satisfação durante a utilização de sistemas computacionais interativos. A qualidade de uso tem o potencial de apoiar na redução dos custos de desenvolvimento de sistemas computacionais interativos, bem como os custos de treinamento dos usuários. Para aumentar a qualidade de uso de sistemas computacionais interativos é necessário considerar os elementos envolvidos na interação, uma vez que a qualidade de uso está estritamente relacionada aos conceitos de interação usuário-sistema,

*interface* gráfica do usuário e *affordance*.

A definição do conceito de interação tem evoluído ao longo do tempo (BARBOSA; SILVA, 2017). Inicialmente, a interação foi definida como uma sequência de estímulos e respostas entre usuário e o sistema. A partir de resultados de pesquisas no campo da Psicologia Cognitiva, o conceito de interação passou a ser definido como a comunicação de seres humanos com máquinas, ao invés de um conjunto de operações de máquina (CARD; MORAN; NEWELL, 1983). De acordo com (NORMAN, 1986), a interação pode ser definida como o processo pelo qual o usuário formula uma intenção, planeja as suas ações, atua sob a *interface*, percebe e interpreta a resposta do sistema e avalia se o seu objetivo foi alcançado. Uma outra definição enfatiza a interação como o processo de comunicação entre pessoas, *designer* e usuário, mediado pelo sistema computacional interativo (de SOUZA, 2005a).

A *interface* compreende à toda porção do sistema com a qual o usuário mantém contato físico (motor ou perceptivo) ou conceitual (interpretação e avaliação) durante a interação (MORAN, 1981). O contato físico pode ocorrer através de *interfaces* de *hardware* ou de *software* utilizadas durante a interação. Dispositivos de entrada como teclado, *mouse*, botões de um *joystick*, microfone, caneta e câmera permitem que o usuário participe ativamente da interação. Dispositivos de saída como monitor, impressoras e caixas de som permitem que usuários percebam as respostas do sistema a cada uma de suas ações e participem passivamente da interação. *Affordances* são definidas pelas características físicas das *interfaces* de um sistema, que possibilitam ao usuário compreender as possíveis operações e manipulações que se pode realizar na *interface*. O conceito de *affordance* consiste no conjunto de características físicas de um objeto que determinam ou sugerem ao observador o tipo de ação/uso que se pode fazer com o objeto (GIBSON, 1977).

## 2.3 Perspectivas de Interação

Segundo (KAMMERGAARD, 1988) a interação do usuário com a máquina pode ser definida sob quatro perspectivas: Sistema, Parceiro do Discurso, Ferramenta e Mídia. Cada perspectiva atribui determinados papéis ao usuário e ao sistema e caracteriza a interação usuário-sistema sob um diferente ponto de vista.

Na perspectiva de sistema (Figura 2.1), o usuário é considerado um sistema e a interação humano-computador se aproxima da interação entre sistemas computacionais, ou seja, a interação é vista como uma mera transmissão de dados entre sistemas. Nesta perspectiva, o usuário deve se comportar como uma máquina, interagindo com o sistema de forma sistemática e restrita por formatos de entrada padronizados, e.g., preenchendo campos de texto e selecionando opções de listas pré-definidas (*comboboxes*). *Web-sites* de vendas de passagens aéreas e de reservas de hotel são exemplos de sistemas que se enquadram na Perspectiva de Sistema, uma vez que fornecem mecanismos para restringir a entrada do usuário com o objetivo de minimizar a probabilidade de erros durante a interação.

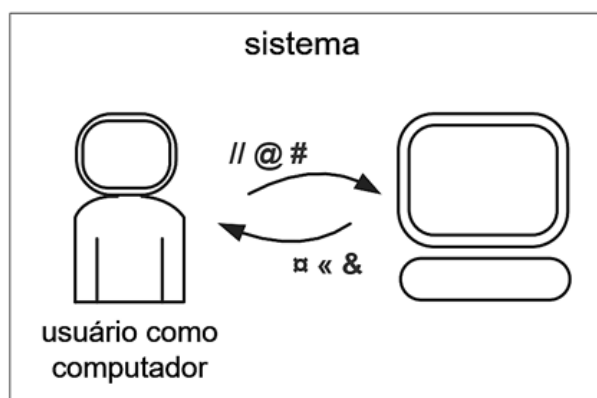


Figura 2.1: Perspectiva de Sistema. Fonte: (KAMMERSGAARD, 1988)

Na perspectiva de parceiro do discurso (Figura 2.2), o sistema interage com o usuário utilizando linguagem natural para auxiliá-lo a solucionar uma dúvida ou atingir determinado objetivo. Os *chatbots* de operadoras de telefonia e de *e-commerces* são exemplos de sistemas que se enquadram na perspectiva de parceiro do discurso. Esses sistemas utilizam recursos de Inteligência Artificial para processar as entradas do usuário e fornecer as respostas adequadas a cada situação (BARBOSA; SILVA, 2017).

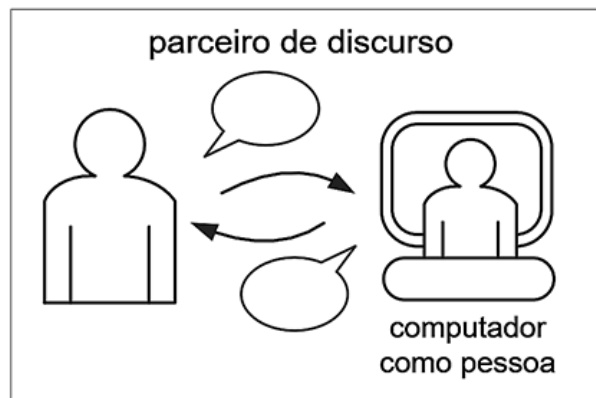


Figura 2.2: Perspectiva de Parceiro do Discurso. Fonte: (KAMMERGAARD, 1988)

Na figura 2.3 é ilustrada a perspectiva de ferramenta, onde o sistema de interação é considerado um instrumento que auxilia o usuário na realização de tarefas, onde a facilidade de uso, proporcionada por funcionalidades relevantes, resulta no fator de qualidade mais evidente. Segundo (KAMMERGAARD, 1988), a interação nessa perspectiva representa um processo no qual uma ferramenta é aplicada a algum material e então tem-se a avaliação do resultado durante a realização de uma atividade. Alguns sistemas de propósito geral e aplicações de escritório, como *Microsoft Office* e *OpenOffice* são exemplos de perspectiva de ferramenta.

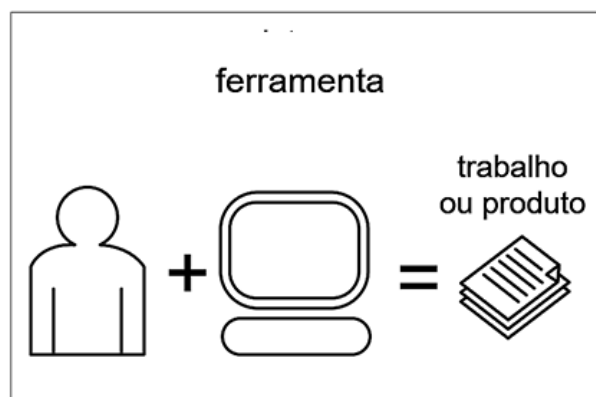


Figura 2.3: Perspectiva de Ferramenta. Fonte: (KAMMERGAARD, 1988)

Na perspectiva de mídia (Figura 2.4), a interação ocorre por sistemas que operam como meio de comunicação, principalmente aqueles que conectam interlocutores através da *internet*, seja entre usuários ou *designer*-usuário. Esses sistemas atuam como mídia, na qual pessoas se comunicam de forma coletiva, como ocorre em sistemas de *e-mail*, fóruns, *chats* e redes sociais. Ou de forma unilateral na atuação de *designers* para usuários, como visto em sistemas de ajuda *on-line*, instruções de *interface* e documentação, ou seleção e

disposição dos elementos de *interface* de sistemas (BARBOSA; SILVA, 2017).

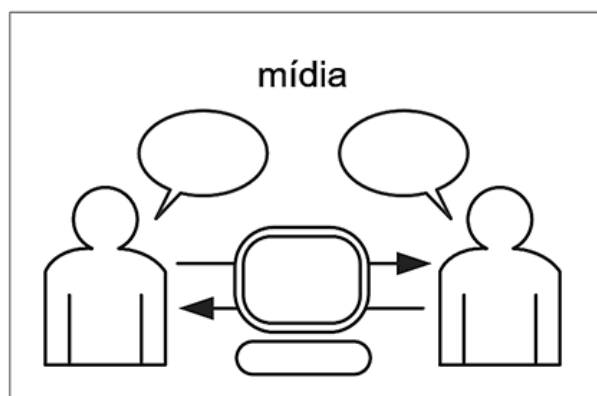


Figura 2.4: Perspectiva de Mídia. Fonte: (KAMMERGAARD, 1988)

## 2.4 Qualidade de Uso

A qualidade de um sistema computacional interativo está relacionada ao quão adequados são os *designs* de interação e de *interface*, que permitam aos usuários aproveitarem o máximo de apoio computacional fornecido pelo sistema. O uso pode ser definido como uma interação com a *interface* de um sistema, a fim de que objetivos sejam alcançados em um determinado contexto (BARBOSA; SILVA, 2017). Para isso têm-se os critérios de qualidade de uso, os quais referem-se a atributos de qualidade da interação e da *interface*, que devem ser atendidos para garantir que usuários tirem o máximo de proveito das capacidades do sistema. Os critérios de qualidade de uso são definidos como: Usabilidade, Acessibilidade, Experiência de Usuário e Comunicabilidade, de acordo com (BARBOSA; SILVA, 2017).

”Usabilidade está relacionada com a facilidade de aprendizado e uso da *interface*, bem como a satisfação do usuário em decorrência desse uso” (NIELSEN, 1993). É composta por cinco atributos: facilidade do aprendizado, facilidade de memorização, eficiência no uso, segurança e experiência do usuário. Segundo (NIELSEN, 1993), a facilidade de aprendizado se caracteriza pela necessidade do usuário em assimilar as características de um sistema e aprender a utilizá-lo, sendo possível aproveitar suas funcionalidades. A facilidade de memorização tem relação com a capacidade de um sistema ser recordado por um usuário, sendo possível a lembrança de seu funcionamento mesmo após algum tempo

inutilizado. Eficiência é a capacidade de um sistema oferecer ao usuário a possibilidade de se atingir uma boa produtividade, por meio de seu uso. A segurança, por sua vez, deve servir de apoio ao usuário, prevendo suas ações e evitando que este cometa possíveis erros, ou permitir que sejam facilmente revertidos caso ocorram.

A experiência do usuário se caracteriza como a satisfação que o sistema proporciona ao ser utilizado, é a possibilidade oferecida ao usuário para que este consiga atingir suas expectativas de forma agradável, por meio de apoio computacional. Hoje em dia os sistemas interativos são utilizados em diversas tarefas humanas, o que consequentemente afeta as emoções e sentimentos das pessoas. A atenção a esses fatores vai além da satisfação do usuário, se tornando um critério de qualidade denominado experiência do usuário, em que aspectos subjetivos são investigados, de formas positiva e negativa, como: satisfação, entretenimento, interesse, surpresa, desafio, entre outros (BARBOSA; SILVA, 2017).

Para que os sistemas possam ser utilizados e capazes de oferecer o máximo de apoio computacional é preciso que barreiras sejam evitadas para que haja perfeita interação do usuário. Esse critério é denominado acessibilidade, que permite acesso ao conteúdo à usuários com diferentes necessidades, evitando obstáculos quanto a movimentação no sistema, capacidade de compreensão, cognição e conhecimento. O intuito é permitir uma interação sem limitações ao sistema também à usuários com necessidades especiais ou deficiências (BARBOSA; SILVA, 2017).

A comunicabilidade permite ao *designer* transmitir suas intenções e propósitos de elaboração. Ao realizar o desenvolvimento o *designer* precisa entender a visão, o propósito, as decisões e o cenário ao qual o usuário pretende utilizar seu sistema, a fim de oferecer a melhor experiência de interação, sem barreiras e de fácil usabilidade (BARBOSA; SILVA, 2017).

## 2.5 Usabilidade *WEB* e Móvel

Atualmente as pessoas utilizam sistemas *web* e móveis para realizar diversas atividades que vão além da troca de informações como, por exemplo, ensino, entretenimento, compras, transações financeiras, entre outras. A usabilidade possui um papel importante na

interação dos indivíduos com esses sistemas, visto que a maneira como o ser humano interage com a máquina influencia em sua experiência de uso e, consequentemente, afeta no grau de sucesso que o apoio computacional é capaz de oferecer. O usuário precisa ser guiado para que consiga obter o máximo de proveito de um sistema e, com isso, ser capaz de concluir uma ação desejada, principalmente usuários inexperientes, que podem não estar familiarizados com um ambiente utilizado.

### 2.5.1 Usabilidade *WEB*

Ao acessar um *website* o usuário observa as características do ambiente e se caso estas o agradarem provavelmente a navegação no sistema será continuada, podendo muitas vezes ser interrompida quando as intenções do usuário não são atendidas. O usuário precisa entender a proposta do desenvolvedor e ser capaz de desfrutar do ambiente de forma objetiva, não devendo ser exposto à conteúdos inúteis em páginas sem motivos que justifiquem sua presença, como nos casos de excesso de modais, *menus pop-up*, aparição de janelas extras no navegador, utilização de elementos que se assemelham a anúncios, entre outros.

O usuário precisa ser capaz de identificar as *affordances* presentes no ambiente e estas corresponderem com as expectativas esperadas para suas funções, sendo que cada página deve ser organizada de maneira clara e bem estruturada quanto a prioridade de seu conteúdo e sua distribuição pela *interface*, bem como atenção ao contraste de cores, características de fontes de textos e mensagens de alertas utilizadas. A preocupação com cada detalhe fornece ao usuário uma boa experiência de uso, permitindo que este realize suas ações com maior rapidez, possibilitando que o ambiente seja visitado novamente (NIELSEN; LORANGER, 2007).

Algumas práticas são úteis no desenvolvimento de sistemas *web*, pois permitem ao usuário desfrutar de boa interação por meio de uma usabilidade bem planejada. Algumas dicas podem ser seguidas na construção de *web sites* para que seja melhorada a experiência do usuário, como: atenção à arquitetura da informação, clareza no *design* da interação, *design* estético e minimalista, usabilidade acessível em diferentes plataformas, interação responsiva, acesso instantâneo à página inicial a qualquer momento, mensa-

gens de erro personalizadas, *layout* limpo, simplicidade e relevância do conteúdo, rápido *feedback*, utilização de tecnologias atuais, entre outras (NIELSEN; BUDIU, 2013).

### 2.5.2 Usabilidade Móvel

Assim como na usabilidade *Web*, a atenção aos detalhes no desenvolvimento de sistemas interativos móveis é o que afeta a experiência do usuário. A limitação presente nas telas destes dispositivos é um dos principais fatores que influenciam a maneira como o conteúdo deve ser apresentado na *interface*. Problemas com tamanho de fontes e imagens podem dificultar o entendimento do usuário, bem como opções de rolagem em *sites* cujo foco é somente *desktop*. Com a aparição dos *smartphones* e a popularização de aplicativos, é importante que *sites* sejam construídos em versões separadas, que possam também atender à dispositivos móveis.

As opções presentes nas *interfaces* devem conter somente informações fundamentais, possibilitando ao usuário expandir a visualização do conteúdo se achar necessário, bem como funções de ampliação de elementos por meio de toques na tela. O *layout* precisa ser limpo, com dimensões que ajustem corretamente a proporção dos elementos observados, além das características adotadas nas fontes utilizadas e cores empregadas no sistema. Deve possuir boa arquitetura da informação, permitir facilidade na navegação, clareza no *design* da interação, conteúdo relevante, *design* estético e minimalista, *feedbacks* sobre as ações realizadas no sistema, entre outras (NIELSEN; BUDIU, 2013).

## 2.6 Princípios e Diretrizes de *Design*

Para que sistemas interativos sejam criados de maneira a possibilitar a melhor experiência e satisfação do usuário, bem como a eficiência das funcionalidades de um produto no apoio computacional à realização de tarefas, é importante utilizar boas práticas que ajudem na implementação. Algumas dessas práticas são bastante utilizadas e ajudam a guiar os *designers* na elaboração das *interfaces*, como os conhecidos "Princípios da *Gestalt*", "Regras de Ouro de Schneiderman" e as "Heurísticas de *Design* de Nielsen", sendo todas essas citadas a seguir.



### 2.6.1 Princípios da *Gestalt*

Segundo (WARE, 2003 apud BARBOSA; SILVA, 2017), parte da nossa inteligência se caracteriza pela capacidade que possuímos de identificar padrões, sendo o sistema visual nosso mecanismo de reconhecimento mais apurado. Um dos objetivos do *design* de representações visuais é realizar o mapeamento dos dados de maneira que estes sejam compatíveis com nossas capacidades de percepção. Em 1912 foi fundada a escola de psicologia da *Gestalt*, pelos psicólogos alemães Max Wertheimer, Kurt Koffka e Wolfgang Kohler. Esses psicólogos definiram leis de percepção de padrões, conhecidas como Princípios da *Gestalt*. Esses princípios apresentam características de como os humanos identificam padrões quando vêem/percebem formas/objetos e, de acordo com (WARE, 2003 apud BARBOSA; SILVA, 2017), podem ser definidos da seguinte maneira:

**Proximidade:** "as entidades visuais que estão próximas umas das outras são percebidas como um grupo ou unidade". Um exemplo é ilustrado na Figura 2.5, onde círculos correspondentes as opções de cores de um produto são percebidos como um único grupo, caracterizando o princípio da proximidade.



Figura 2.5: Princípio da Proximidade. Fonte: Nike.com

**Boa continuidade:** "traços contínuos são percebidos mais prontamente do que contornos que mudam de direção rapidamente". É possível perceber na Figura 2.6 traços que alternam entre tons claros e escuros de forma contínua e gradativa. A boa continuidade também é conhecido com o princípio do alinhamento dos elementos em uma

*interface*, tornando mais intuitiva a identificação dos elementos na *interface*.



Figura 2.6: Princípio da Continuidade. Fonte: (HOFSTETTER, 2019)

**Simetria:** "objetos simétricos são prontamente percebidos mais do que objetos assimétricos". Na ilustração da Figura 2.7 é possível perceber um exemplo de simetria, uma vez que a imagem apresenta opções de escolha de produtos em um site. Essas opções são exibidas de maneira simétrica, com os itens seguindo o mesmo padrão e tamanhos proporcionais.

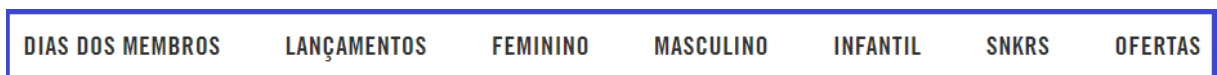


Figura 2.7: Princípio da Simetria. Fonte: Nike.com

**Similaridade:** "objetos semelhantes são percebidos como um grupo". Na Figura 2.8 ocorre um exemplo de similaridade, onde é possível perceber a exibição de dois modelos similares de tênis à venda em um site de compras *on-line*. Os produtos apresentam características semelhantes, como formato, cor e tamanho, sendo percebidos como um grupo.

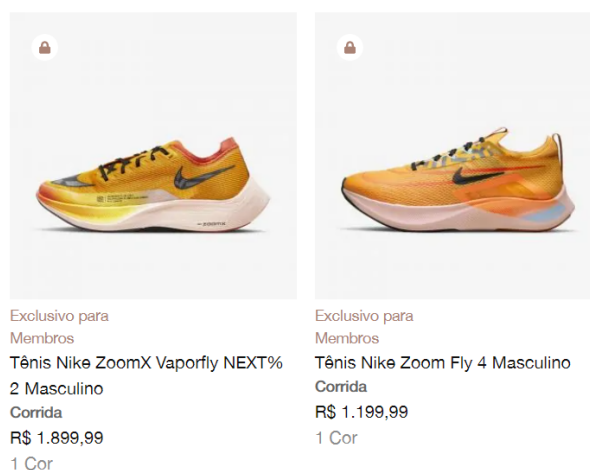


Figura 2.8: Princípio da Similaridade. Fonte: Nike.com

**Destino comum:** "objetos com a mesma direção de movimento são percebidos como um grupo". Os elementos da Figura 2.9 se movem de forma sincronizada e na mesma direção, sendo caracterizados como um exemplo de destino comum. A região do *menu* presente no topo da ferramenta *Microsoft Word* parece fazer parte das opções localizadas na área mais inferior.

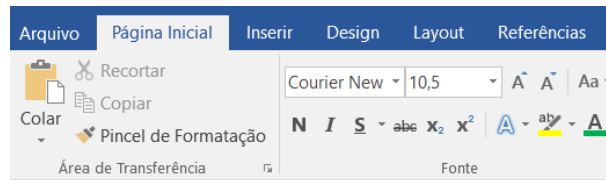


Figura 2.9: Princípio do Destino Comum. Fonte: Microsoft Word

**Fecho:** "a mente tende a fechar contornos para completar figuras regulares, "completando as falhas" e aumentando a regularidade". Um exemplo de fechamento pode ser visualizado na Figura 2.10, uma vez que ao perceber a ilustração nosso cérebro é capaz de fechar ou concluir formas abertas ou inacabadas, prevendo como deveria ser a estrutura. O princípio é utilizado para atrair a atenção do usuário para determinadas regiões da *interface*.

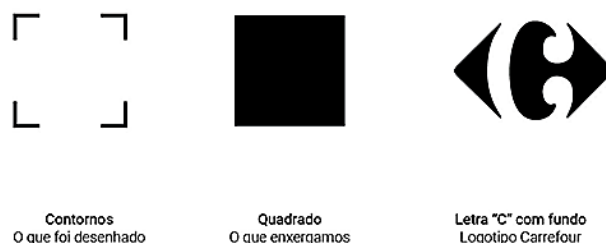


Figura 2.10: Princípio do Fecho. Fonte: (HOFSTETTER, 2019)

**Região comum:** "objetos agrupados dentro de uma mesma região são percebidos como um grupo". Pode até ser considerada um sub-princípio da Proximidade, dado que há uma relação entre elas. Na Figura 2.11 os círculos são dispostos de maneira a serem interpretados como diferentes grupos, uma vez que há uma divisão em que objetos compartilham uma área com limite claramente definido.

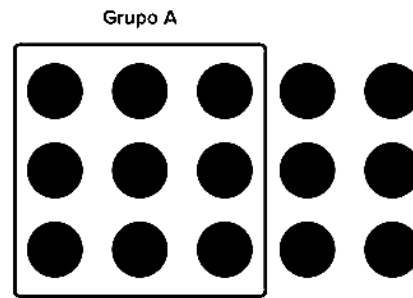


Figura 2.11: Princípio da Região Comum. Fonte: (Aela Editorial, )

**Conectividade:** "objetos conectados por traços contínuos são percebidos como relacionados". A Figura 2.12 apresenta um exemplo de conectividade, em que os quadradinhos são ilustrados de maneira a manterem ligação por meio de "linhas".

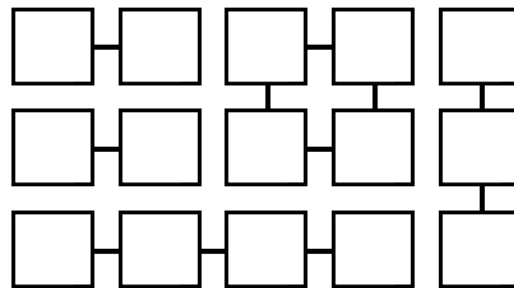


Figura 2.12: Princípio da Conectividade. Fonte: (HOFSTETTER, 2019)

### 2.6.2 Regras de Ouro de Schneiderman

Em 1986 o americano Ben Shneiderman apresentou suas heurísticas que ajudam na elaboração de *interfaces*. Conhecidas como "Regras de ouro", essas heurísticas foram divididas em oito princípios, os quais são descritos a seguir com base nas definições do próprio autor (SHNEIDERMAN et al., 2016).

- **Esforce-se para ter consistência:** ações consistentes devem se repetir em similares situações. Avisos, *menus* e telas de ajuda devem seguir terminologias iguais, havendo consistência nos elementos empregados por toda parte;
- **Busque usabilidade universal:** flexibilidade deve ser empregada no design de interfaces, com base na percepção das necessidades dos diferentes tipos de usuários, com explicações e atalhos para melhorar a qualidade dos sistemas;

- **Ofereça *feedback* informativo:** sistemas devem fornecer *feedback* aos usuários para cada ação realizada. Apresentação visual pode contribuir na percepção de alterações;
- **Projete diálogos que indiquem fechamento:** sequências de ações devem ter seu tempo de duração indicado, com início, meio e fim. Usuários precisam ter *feedback* após conclusão de cada etapa;
- **Previna erros:** sistemas devem ser projetados de forma a evitar que usuários cometam erros graves. Mensagens com instruções devem ser informadas caso hajam erros;
- **Permita a fácil reversão de ações:** sistemas devem permitir a reversão de ações. Usuários devem ser capazes de desfazer erros cometidos;
- **Mantenha os usuários no controle:** usuários experientes esperam obter controle das interfaces, sendo capazes de produzir resultados desejados sem surpresas no comportamento de suas ações;
- **Reduza a carga de memória de curto prazo:** sistemas devem evitar interfaces que obriguem usuários a memorizarem informações de uma tela para serem utilizadas em outras.

### 2.6.3 Processo de Engenharia de Usabilidade de Nielsen

O Processo de Engenharia de Usabilidade de Nielsen (NIELSEN, 1993) prescreve um conjunto de atividades de análise, projeto e avaliação de interfaces do ciclo de desenvolvimento de um sistema computacional interativo. Nielsen prescreve que as seguintes atividades para o desenvolvimento de um sistema computacional interativo:

- **Conheça seu usuário:** deve-se estudar as características de usuários individuais e conhecer seus ambientes físicos e sociais de trabalho, variabilidade de suas tarefas e como lidam com circunstâncias inesperadas;

- **Realize uma análise competitiva:** produtos com funcionalidades semelhantes ou complementares devem ser examinados, resultando em informações sobre o que funciona ou não, o que é cabível de aperfeiçoamento e por quê;
- **Defina as metas de usabilidade:** consiste em determinar fatores de qualidade de uso com alta prioridade, como serão avaliados no processo de *design* e o grau de aceitação para cada indicador de interesse;
- **Faça *designs* paralelos:** diferentes alternativas de *design* devem ser elaboradas, com grupos de *designers* trabalhando de maneira independente, para eleger quais serão detalhadas nas seguintes atividades do processo;
- **Adote o *design* participativo:** a equipe de *design* deve sempre ter acesso a um conjunto de usuários representantes de uma população-alvo;
- **Faça o *design* coordenado da interface como um todo:** consiste em determinar um responsável pelo *design* coordenado da interface, para evitar que esta sofra inconsistências com usuário;
- **Aplique diretrizes e análise heurística:** a equipe de *design* deve seguir diretrizes e princípios conhecidos para elaboração da interface com o usuário e, a medida que esta for projetada uma avaliação heurística deve ser realizada para avaliar se há violação de diretrizes;
- **Faça protótipos:** protótipos são indicados antes da implementação de sistemas finais, que podem ser criados rapidamente, servindo de base para o entendimento junto a usuários, podendo haver modificações com maior facilidade no decorrer do processo de entendimento;
- **Realize testes empíricos:** testes empíricos devem ser realizados a partir de protótipos, com objetivo de se observar o comportamento dos usuários ao utilizá-los na realização de tarefas;
- **Pratique *design* iterativo:** com os resultados de testes empíricos os *designers* produzem uma nova versão da interface, repassando de forma iterativa pelas ativi-

dades do processo, podendo haver correção de problemas ou a introdução de novos, devendo o processo ser repetido até o alcance das metas de usabilidade estabelecidas.

#### 2.6.4 Heurísticas de *Design* de Nielsen

Nielsen propôs em 1993 boas práticas chamadas de heurísticas de *design*. Essas heurísticas constituem com conjunto de regras gerais de *design* comumente observadas na prática, que podem ser utilizadas em todas as etapas de desenvolvimento de um produto, até mesmo antes do desenvolvimento das *interfaces* do sistema. As heurísticas de *design* são definidas em dez tipos, organizadas da seguinte maneira:

1. Correspondência com as expectativas dos usuários;
2. Simplicidade nas estruturas das tarefas;
3. Equilíbrio entre controle e liberdade do usuário;
4. Consistência e padronização;
5. Promoção da eficiência do usuário;
6. Antecipação das necessidades dos usuários;
7. Visibilidade reconhecimento;
8. Conteúdo relevante e expressão adequada;
9. Projeto para erros;
10. Ajuda e documentação.

Um exemplo das heurísticas de *design* pode ser observado na figura 2.13. As ferramentas *Word*, *Excel* e *PowerPoint* do pacote *Microsoft Office* atendem às características de **Consistência e Padronização**, visto que as ferramentas embora apresentem funcionalidades distintas as ações relacionadas em situações similares funcionam da mesma maneira, com ações e resultados das ações padronizados, assim como o *layout* dos diálogos e as visualizações de informações.

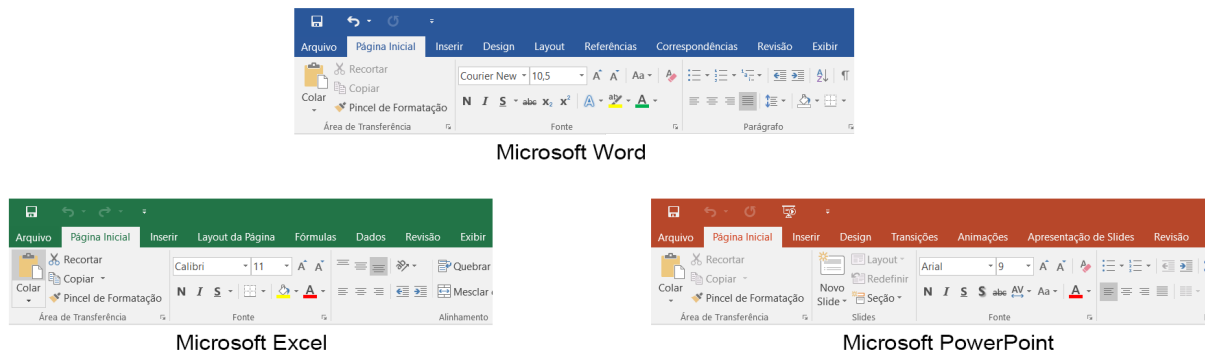


Figura 2.13: Heurística de Consistência e Padronização no Pacote Microsoft Office.

## 2.7 Engenharia Cognitiva

Segundo (NORMAN, 1986) a Engenharia Cognitiva investiga aspectos cognitivos do processo de interação humano-computador. Tende a aplicar conhecimento dos campos da ciência, psicologia cognitiva e fatores humanos ao *design* de sistemas computacionais, envolvendo aspectos psicológicos, como objetivos e intenções do usuário, e físicos, como aqueles realizados pela atuação sobre controles fornecidos pela *interface*. Os sistemas precisam ser desenvolvidos de maneira a proporcionar experiências agradáveis aos usuários, sendo o entendimento dos princípios de ação e desempenho humano relevante nos objetivos da engenharia cognitiva.

A engenharia cognitiva considera IHC como um tipo de ação, onde a teoria se relaciona ao projeto focado no usuário, com base em estudos dos fenômenos que ocorrem durante a interação do humano com um artefato cognitivo, ou seja, o quanto a interação é direta e envolvente (NORMAN, 1991). Norman propôs uma teoria da ação com diversos estágios de atividades que ocorrem numa interação entre usuário-sistema, representando a discrepância entre variáveis psicológicas (objetivos das pessoas) e os controles, e variáveis físicas (mecanismos de interação e estados dos sistema), com representação através do golfo de execução e do golfo de avaliação (BARBOSA; SILVA, 2017).

De acordo com Norman, o golfo de execução se caracteriza como a dificuldade existente na atuação sobre o ambiente, considerando-se também o sucesso no apoio às ações fornecido pelo artefato. Já o golfo de avaliação refere-se a dificuldade de se avaliar o estado do ambiente, bem como o grau de sucesso com que o artefato apoia a detecção e interpretação desse estado (BARBOSA; SILVA, 2017). A Figura 2.14 ilustra os estágios



de travessia dos golfos onde, segundo Norman, a travessia do golfo de execução se torna a primeira fase do ciclo, em que o usuário estabelece um objetivo no qual se deseja alcançar por meio de interação com o sistema. Após estabelecer o objetivo, o usuário formula uma ou mais intenções imediatas necessárias para atingir o objetivo. Posteriormente, o usuário formula um plano de ações mentais que devem ser executadas para atingir a intenção imediata.

Após a especificação do plano de ações mentais, o usuário atua sob a *interface* para executar cada ação mental de acordo com a ordem especificada. Em seguida na execução da ação, o usuário percebe (forma) a resposta do sistema, interpreta o significado dessa resposta e avalia o estado do sistema após atravessar o golfo de avaliação (BARBOSA; SILVA, 2017).

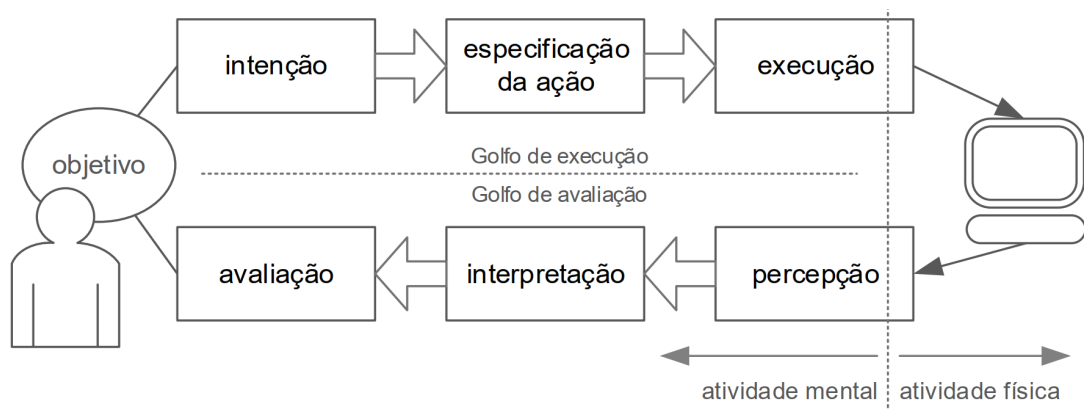


Figura 2.14: Estágios de atividade do usuário na travessia dos golfos de execução e de avaliação. Fonte: (BARBOSA; SILVA, 2017)

A figura 2.15 ilustra o exemplo de um cenário de interação humano-computador com destaque nas atividades que representam as etapas da Travessia dos Golfos. No Golfo de Execução o usuário formula a intenção de procurar por um livro de IHC, especificando em seguida as ações necessárias para atingir o objetivo, que consistem em selecionar o comando de busca e entrar com os dados no sistema. Após passar pela etapa de especificação da ação e saber o que precisa ser realizado no sistema, o usuário inicia a primeira atividade física com a máquina, a execução. Nessa etapa é acionada a opção de busca no *menu* e o nome do livro é informado no campo de pesquisa, sendo o botão "buscar" executado em seguida.

No caminho de retorno (Golfo de Avaliação) o usuário percebe o resultado for-

necido pelo sistema, em que uma nova janela com lista de livros retornados pela busca é apresentada. O usuário após a etapa de percepção interpreta os resultados, para saber se os livros retornados correspondem à busca realizada. Por fim, o usuário na etapa de avaliação compara se o resultado obtido corresponde ao objetivo inicial, avaliando se o livro desejado foi encontrado, sendo a tarefa realizada completada com sucesso. A figura 2.16 apresenta a sequência de interação pelas telas de um navegador.

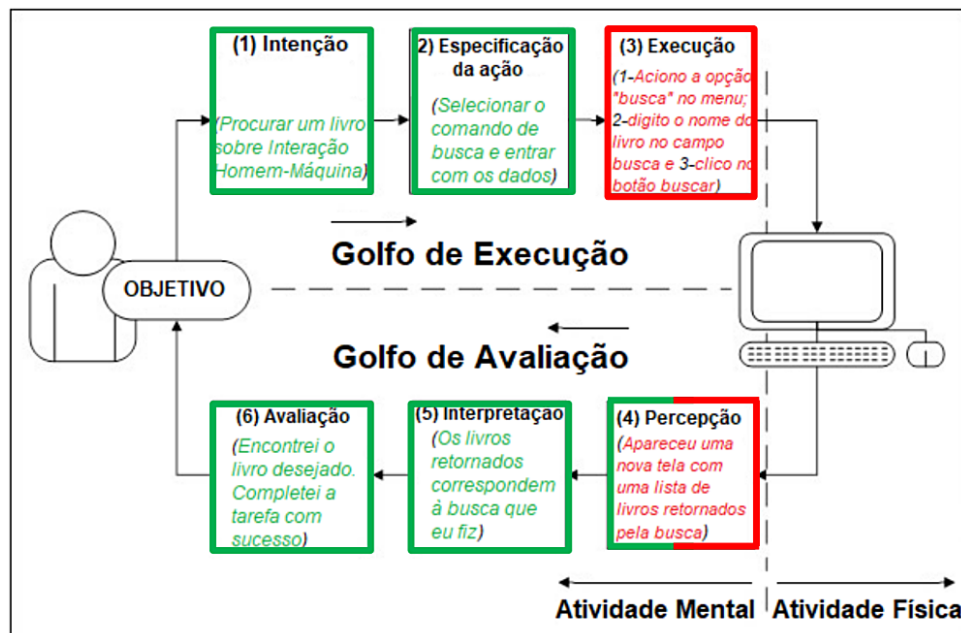


Figura 2.15: Exemplo de um cenário com aplicação da Teoria dos Golfos

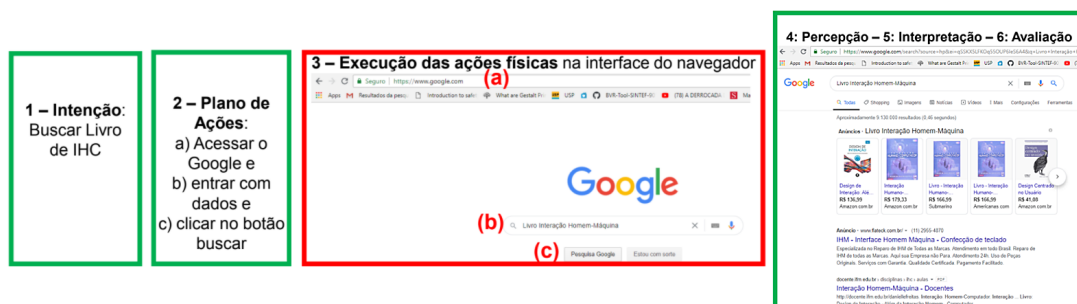


Figura 2.16: Sequência de passos na interação do usuário com o sistema para a especificação de um cenário

## 2.8 Arquitetura de Aplicações Web e Móveis

Como ilustrado na Figura 2.17 o padrão de projetos MVC (*Model-View-Controller*) é muito utilizado em aplicações WEB e móveis. Sua dinâmica é simples, com arquitetura

dividida em três componentes, com recebimento de requisições do sistema (*Controller*), que são posteriormente processadas (*Model*), tendo por fim seu resultado exibido (*View*). As camadas de apresentação são isoladas ao máximo, bem como as de lógica de negócio e as de gerenciamento de fluxo. Esse padrão é aplicado para simplificar o entendimento do sistema no momento de seu desenvolvimento, assim como facilitar na manutenção do projeto e sua reutilização (LUCIANO; ALVES, 2017).

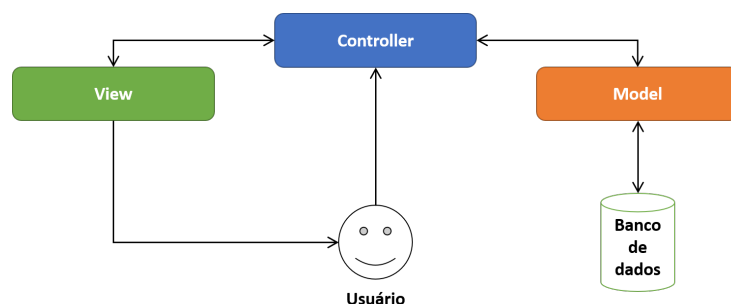


Figura 2.17: Padrão de Arquitetura de Software MVC. Fonte: (TreinaWeb, 2021)

O *Model* compreende a funcionalidade principal e os dados da aplicação. Esses dados são encapsulados e os procedimentos exportados para a realização de procedimentos específicos do sistema, fornecendo também funções para que seus dados sejam acessados por *Views* (Laputan, 1999).

As *Views* do sistema ficam responsáveis por exibir as informações aos usuários. Essas informações são baseadas em dados obtidos do *Model*, onde cada *View* possui um componente *Controller* associado (Laputan, 1999).

Os *Controllers* trabalham com as informações obtidas dos usuários. Recebem os dados de entrada, via toques de tela, *mouse* ou teclado, onde eventos de *Model* e *Views* são traduzidos em serviços. A união de *Views* e *Controllers* resulta na *interface* do usuário (Laputan, 1999).

## 2.9 Tecnologias Utilizadas no Desenvolvimento Web e Móvel

Diversas tecnologias podem ser utilizadas no padrão *Model-View-Controller* para o desenvolvimento Web e Móvel, dentre *frameworks*, bancos de dados e linguagens de pro-

gramação. Algumas tecnologias foram aplicadas na implementação do aplicativo SAD-UFJF e são citadas a seguir:

Para o componente *Model* tem-se a utilização de NoSQL (Banco de Dados Não Relacional), um meio de armazenamento de dados que não se limita à regras do Banco de Dados tradicional, conhecido como SQL. NoSQL é muitas vezes escolhido por proporcionar maior rapidez e flexibilidade, visto que não se prende a uma relação de tabelas e todos os seus dados constam no mesmo registro (CUER, 2015).

*Views* podem ser criadas com ajuda de *frameworks* para facilitar no trabalho de desenvolvimento, tornando o *front-end* do sistema mais rápido e eficiente. Nesse sentido a tecnologia *React Native* é utilizada como forma de melhorar o desempenho e produtividade, por permitir que a implementação seja realizada com maior facilidade, independentemente da plataforma a ser utilizada (MATTOS; KERNITSKEI; JUNIOR, ).

Para *controllers* há a necessidade de tecnologias que apoiem o desenvolvimento de *back-ends* cada vez mais eficientes e produtivos. Para isso, a escolha de um ambiente de execução *JavaScript*, como *NodeJS*, permite a criação de aplicações independentes de *browsers* para execução. Essa tecnologia possui alta capacidade de escala, aliado a uma boa arquitetura e flexibilidade, possibilitando o recebimento de várias requisições, sem a necessidade de espera por tempo de resposta durante uma execução (Lenon, 2018).

### 2.9.1 *React Native*

É uma tecnologia para a criação de aplicativos nativos para as plataformas *Android* e *iOS* com a utilização de *React*. Utiliza *JavaScript* renderizado com código nativo para construir *interfaces* de usuário. Fornece um conjunto de componentes que não dependem da plataforma utilizada, como texto, visualização e imagem. Com o uso de *JavaScript* o *React Native* é capaz de realizar iterações com grande velocidade, permitindo que alterações no código de um projeto sejam visualizadas rapidamente, uma vez que estas são exibidas no momento em que são salvas (React Native, 2022). O *React Native Packager* executa a tarefa de coletar o código *Javascript*, unindo e tornando mais acessível para o *Javascript Core*. Logo após o arquivo ser processado pelo *React Native Package (bundle.js)*, a execução do app ocorre de forma automática, com um fluxo de desenvolvimento

mais claro e limpo (WEIAND, 2019). A arquitetura pode ser visualizada na Figura 2.18.

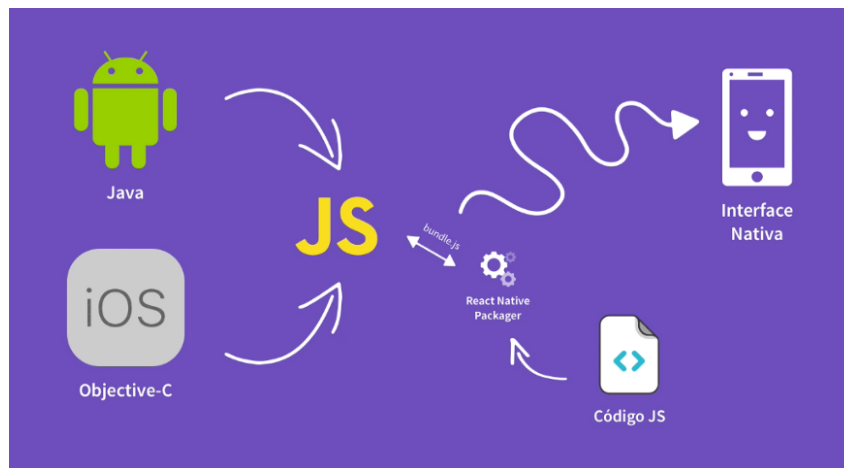


Figura 2.18: Arquitetura *React Native*. Fonte: (Rocketseat, apud Medium, 2019)

### 2.9.2 *Node.JS*

É uma tecnologia responsável pela execução dos códigos *JavaScript* fora do navegador *Web*. Sua comunicação com bases de dados *NoSQL* é bastante intuitiva, uma vez que essas bases se apoiam em *JSON* (*JavaScript Object Notation*), não sendo necessárias modificações no modelo de dados, podendo os objetos *JavaScript* armazenados serem enviados para o *front-end* sem que precise haver conversões ou tratamento. Obtêm boa performance quanto ao consumo de memória, evitando ociosidade e permitindo que o processamento dos servidores seja utilizado ao máximo (NodeJS, 2021). A utilização de *NodeJS* evita que sistemas que o consomem sofram com problemas de *dead-locks*, pois apenas uma única *thread* por processo é utilizada, gerando consequentemente maior praticidade e menos complicação (NodeJS, 2021). Após a execução de um *script* de entrada o *NodeJS* inicia um *loop* de eventos, o qual é encerrado quando não há mais *callbacks* a serem executados, se comportando como o *JavaScript* de um navegador (NodeJS, 2021). Na ilustração da Figura 2.19 é possível visualizar a arquitetura da tecnologia *Node.js*, em que é mantido um conjunto limitado de *threads* que atendem pedidos, sendo cada um armazenado em uma fila. O "loop de eventos" atua como o componente central e espera pelas solicitações indefinidamente, verificando se o pedido ao chegar requer operação de entrada/saída de bloqueio, processando e enviando resposta caso contrário. Se operações de bloqueio forem identificadas o *loop* coloca os pedidos na fila, uma vez que a tarefa de

bloqueio é processada, mantendo a natureza de não-bloqueio.

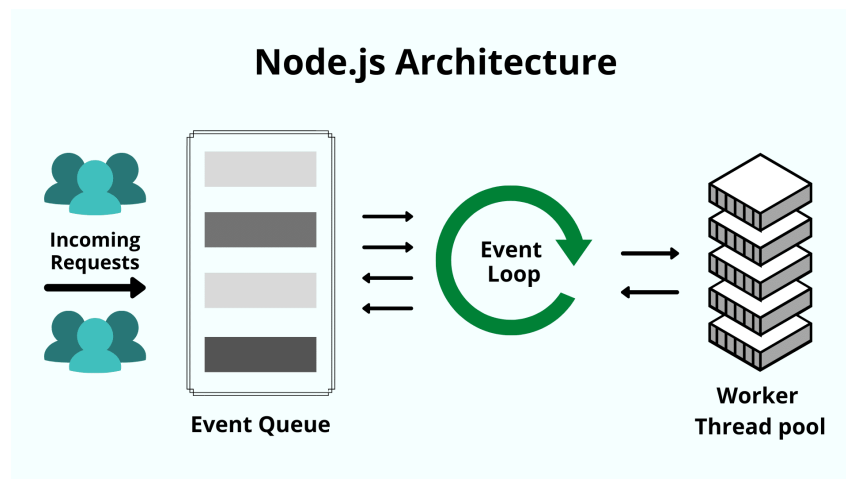


Figura 2.19: Arquitetura *Node.js*. Fonte: (Kinsta, 2021)

### 2.9.3 *Express.js*

É um *framework* para *Node.js* que fornece um conjunto de recursos mínimos que são necessários para a construção de aplicativos *web* e móveis. Dentre vantagens na utilização desse *framework* destacam-se a presença de sistema de rotas completo, possibilidade de tratamento de exceções na própria aplicação, integração de *templates* para facilitar criações de páginas *web* para aplicações, criação rápida de aplicações utilizando-se pequeno conjunto de arquivos e pastas, entre outras (Express, 2022)

O Express.js é utilizado mesmo quando já existe o Node.js. Embora seja possível criar um aplicativo da *Web* ou um serviço de *API* apenas com *Node.js*, o *Express.js* simplifica o processo de desenvolvimento. O próprio *Express* é baseado em *Node*. Esse *framework* fornece uma camada fina de recursos fundamentais de aplicativos da *Web*, sem obscurecer os recursos do *Node.js*.

Por exemplo, enviar uma imagem é complexo no *Node*, mas é feito facilmente no *Express*. No *Node*, o manipulador de rotas é um grande monólito, mas o *Express* permite um *design* mais modular e um código de manutenção. O *Node* é um *runtime JavaScript* para execução no lado do servidor, assim, o *Node* pode ser usado como o servidor de aplicativos *web*. O *Express* é visto como um *framework web* leve, que possui vários *softwares* "middleware" que implementam soluções prontas para uso em requisitos

típicos de aplicativos da *Web*. As solicitações/respostas *HTTP* são retransmitidas do *Node* ao *Express*, cujo *middleware* faz o processamento (ROOM, ).

A figura 2.20 ilustra um fluxo de chamada de solicitação/resposta *HTTP* típico com *Node* e *Express*. Inicialmente o cliente faz requisição ao servidor, em seguida o servidor *Node* envia a solicitação para o *Express*, que posteriormente adiciona recursos para serem usados no *middleware*. Como resposta as funções de *middleware* personalizadas lidam com a solicitação e finalmente o servidor entrega resposta de volta ao cliente.

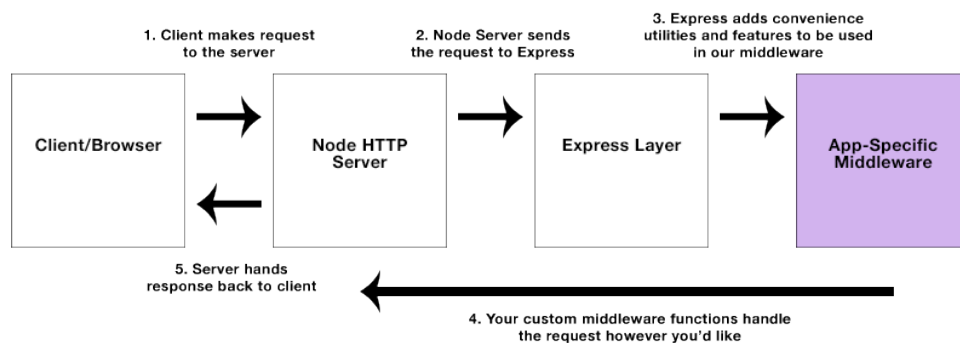


Figura 2.20: Um fluxo de chamada de solicitação/resposta HTTP típico com Node e Express. Fonte: Turing School 2019.

### 2.9.4 *MongoDB*

*MongoDB* é um banco de dados não relacional disponível em nuvem ou servidor. Permite escalabilidade de dados armazenados em documentos do tipo *JSON* flexíveis, onde pode haver variação entre campos e alteração na estrutura de dados no decorrer do tempo. É um banco de dados gratuito e fácil de utilizar, uma vez que seu modelo de documento se baseia em mapas para objetos em seu código. O *MongoDB Atlas* surge como um serviço do *MongoDB*, sendo responsável por tarefas de infraestrutura, manutenção e segurança, facilitando no desenvolvimento do usuário, em que este precisa apenas se preocupar com a administração dos dados armazenados (MongoDB, 2022b). Na ilustração da Figura 2.21 é apresentada uma comparação entre os termos utilizados nos bancos relacionais e as nomenclaturas recebidas em *MongoDB*.

Linhas no *MongoDB* passam a ser conhecidas como documentos (*documents*), colunas como campos (*fields*) e tabelas como coleções de documentos (*collections*).


MongoDB	Bancos de dados relacionais	Exemplo
Document	Linha	{ "nome" : "Guilherme Lima", "email" : "gui@alura.com" }
Field	Coluna	{ "nome" : "Guilherme Lima", "email" : "gui@alura.com" }
Collection	Tabela	

Figura 2.21: Comparação de termos entre MongoDB e Bancos de Dados Relacionais. Fonte: (Alura, 2021)

O *MongoDB* por ser um banco de dados não relacional (*NoSQL*) apresenta algumas vantagens quanto a sua utilização. É mais flexível e escalável quando comparado aos bancos relacionais (*SQL*), já que permite adicionar novos dados sem ter que pré-defini-lo no esquema do banco de dados, tornando possível o processamento rápido de grandes volumes de dados não estruturados, semi-estruturados e estruturados. Aceita o desenvolvimento ágil, com iterações rápidas e significativas. O *MongoDB* é gratuito, de código aberto, e oferece uma arquitetura eficiente e escalonada ao invés de uma arquitetura monolítica cara. Possui velocidade para o mercado, visto que apresenta programação orientada a objetos, o que permite aos desenvolvedores executar consultas com menos preparação e sem ter que navegar por arquiteturas de banco de dados *SQL* (MongoDB, 2022b).

### 2.9.5 Mongoose

É uma biblioteca de programação orientada a objetos *JavaScript* que se faz presente na criação da conexão entre o *MongoDB* e o *Express.JS*. É uma ferramenta de modelagem de objetos *MongoDB* projetada para funcionar em um ambiente assíncrono, com suporte a *promises* e retornos de chamadas (Mongoose, 2022).

### 2.9.6 Heroku

É uma plataforma em nuvem como um serviço que suporta várias linguagens de programação. Utilizado para o *deploy* do *back-end*, com serviços de dados integrados e um



poderoso ecossistema, para implantação e execução de aplicativos modernos. O *Heroku* auxilia no desenvolvimento escalável e possui produtividade como uma de suas vantagens, já que possui integração com a plataforma de hospedagem *GitHub*, facilitando nas atividades de implantação e controle (Heroku, 2022).

A arquitetura *Heroku runtime* é ilustrada na figura 2.22, com a sequência de seu tempo de execução até os usuários finais da plataforma. As etapas são descritas da seguinte maneira:

- Código de implantação do desenvolvedor;
- Aplicativos executados em dynos (contêineres inteligentes);
- Usuários fazem solicitações, que são atendidas pelo aplicativo;
- Desenvolvedor gerencia o aplicativo no painel;
- Dados em serviços de suporte em escala da Web (por meio de complementos).

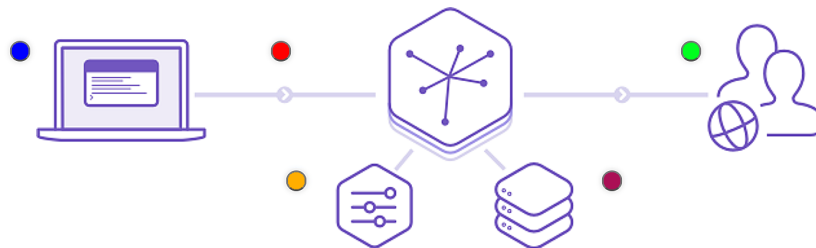


Figura 2.22: Tempo de execução do *Heroku*: do código aos usuários finais. Fonte: *Heroku website*

O *Heroku* apresenta vantagens na sua utilização quando se trata, por exemplo, de produtividade, uma vez que há integração da plataforma com o *GitHub*, que deixa o *deploy* e o controle mais fácil. Além disso, possui uma variedade de *add-ons* interessantes, que facilitam adicionar uma *database*, ou como um novo *plug-in* em um *site WordPress*, apenas com acesso ao *Heroku* e seleção do *add-on* desejado. O *Heroku* foca no desenvolvimento do núcleo da aplicação e não em serviços externos como hospedagem, testes, ou infraestrutura, tornando mais simples o seu uso (Heroku, 2022).

## 2.10 Processo de Avaliação Discente no DCC/UFJF

O processo de avaliação institucional docente da Universidade Federal de Juiz de Fora é resultado da coleta e análise de avaliações discentes. O docente em período de experiência é avaliado por discentes de uma disciplina a qual o mesmo tenha lecionado, sendo o resultado analisado por comissões de avaliação de estágio probatório que decidem o futuro do docente na universidade.

Cada departamento da instituição pode realizar avaliação de um docente em estágio probatório de maneira independente, porém, na maioria dos casos estas optam por realizar avaliações por meio de preenchimento de formulários eletrônicos, como é o caso do Departamento de Ciência da Computação (DCC). Os formulários *online* incluem critérios de avaliação da qualidade de ensino relacionados à uma disciplina cursada. Próximo ao término de um período letivo os discentes com matrícula ativa em uma turma são convidados a preencherem esses formulários e com isso avaliarem a conduta de docentes em estágio probatório, bem como avaliarem questões relacionadas aos seus próprios desempenhos. O DCC aplica as avaliações por meio dos formulários do *Google*, conhecidos como *Google Forms*.

Na figura 2.23 são apresentados os critérios utilizados na avaliação de desempenho docente em disciplinas. Os critérios referentes ao docente são divididos entre treze questões que são apresentadas aos discentes no momento do preenchimento dos formulários. Como forma opcional o discente respondente pode deixar um comentário referente a observações sobre uma disciplina cursada, comentário esse que é posteriormente enviado ao Chefe de Departamento e a Coordenadores de Cursos de Graduação. Os discentes também podem responder à critérios de auto desempenho, os quais são exibidos em dez questões relacionadas ao conteúdo apresentado e ao comportamento do aluno no decorrer de uma disciplina cursada. Os critérios de avaliação de desempenho discente são apresentados na Figura 2.24.

<b>CrITÉrios de Avaliação Docente:</b>	
<b>A</b> - O professor disponibilizou o plano de curso da disciplina na primeira semana de aula?	<b>I</b> - O material utilizado em aula é atualizado?
<b>B</b> - O professor é pontual?	<b>J</b> - O professor apresenta exemplos em aula e faz ou propõe exercícios e/ou trabalhos práticos?
<b>C</b> - O professor é assÍduo Às aulas?	<b>K</b> - As avaliações refletem o conteúdo apresentado em sala?
<b>D</b> - O professor usa todo o tempo de aula?	<b>L</b> - As notas das avaliações foram publicadas até três dias antes da avaliação seguinte?
<b>E</b> - O professor estÁ disponÍvel no horÁrio de atendimento?	<b>M</b> - O conteúdo programático da disciplina foi cumprido?
<b>F</b> - O professor tira dÍvidas em sala de aula?	
<b>G</b> - O professor apresenta os conteúdos da disciplina de forma clara?	
<b>H</b> - O professor demonstra dominar o conteúdo da disciplina?	
<b>Opcional:</b>	
- Comentário ao Chefe de Departamento e Coordenadores de Curso de Graduação.	

Figura 2.23: CritÉrios de avaliação de desempenho docente em disciplinas

<b>N</b> - Você acha que a bibliografia sugerida no Plano de Curso é adequada ao conteúdo da disciplina?
<b>O</b> - Os objetivos da disciplina foram atendidos?
<b>P</b> - Você considera a ementa da disciplina adequada ao seu curso?
<b>Q</b> - Você é pontual?
<b>R</b> - Você é assÍduo Às aulas?
<b>S</b> - Você procurou o professor, tutor ou monitor da disciplina para tirar dÍvidas ao longo do semestre?
<b>T</b> - Você considera que possuía a base teórica necessária para cursar esta disciplina?
<b>U</b> - Você faz as atividades práticas propostas pelo professor?
<b>V</b> - Você considera que se dedicou suficientemente à disciplina?
<b>X</b> - Na sua opinião, suas notas refletem de modo fiel o conhecimento que você adquiriu ao longo da disciplina?

Figura 2.24: CritÉrios de avaliação de desempenho discente em disciplinas

Na Figura 2.25 é apresentado um exemplo de tabela com os dados das avaliações discentes. Cada linha da tabela corresponde a avaliação de um discente, onde a identidade do mesmo não é revelada, sendo exibido apenas o identificador (ID) do respondente, data e hora do preenchimento do formulário, curso que o discente estÁ matriculado, disciplina que estÁ sendo avaliada e respostas para cada uma das questİes referentes a conduta do docente que leciona a disciplina. Cada letra presente na tabela corresponde a um critÉrio utilizado na avaliação, como a letra A, que discorre sobre o plano de curso da disciplina, para saber se este foi disponibilizado pelo docente na primeira semana de aula.

ID	Timestamp	Curso	Disciplina/Critério	A	B	C	D	E	F	G	H	I	J	K	L	M
D1	15/02/2022 16:44:11	CC-N	DCCXXX	Sim	5	5	5	5	4	3	4	3	5	5	5	4
D2	15/02/2022 11:07:14	SI	DCCXXX	Sim	5	5	5	2	3	1	1	1	3	4	2	3

**Legenda:** CC-N – Ciência da Computação Noturno, CC-D – Ciência da Computação Diurno, SI – Sistemas de Informação

**Critérios de Avaliação Docente:**

<p><b>A</b> - O professor disponibilizou o plano de curso da disciplina na primeira semana de aula?</p> <p><b>B</b> - O professor é pontual?</p> <p><b>C</b> - O professor é assíduo às aulas?</p> <p><b>D</b> - O professor usa todo o tempo de aula?</p> <p><b>E</b> - O professor está disponível no horário de atendimento?</p> <p><b>F</b> - O professor tira dúvidas em sala de aula?</p> <p><b>G</b> - O professor apresenta os conteúdos da disciplina de forma clara?</p> <p><b>H</b> - O professor demonstra dominar o conteúdo da disciplina?</p>	<p><b>I</b> - O material utilizado em aula é atualizado?</p> <p><b>J</b> - O professor apresenta exemplos em aula e faz ou propõe exercícios e/ou trabalhos práticos?</p> <p><b>K</b> - As avaliações refletem o conteúdo apresentado em sala?</p> <p><b>L</b> - As notas das avaliações foram publicadas até três dias antes da avaliação seguinte?</p> <p><b>M</b> - O conteúdo programático da disciplina foi cumprido?</p>
--	--

Figura 2.25: Exemplo de tabela com dados das avaliações discentes

O processo de avaliação de docentes em estágio probatório realizado atualmente por formulários eletrônicos apresenta dificuldades quanto a confidencialidade dos dados coletados, uma vez que não há como garantir a autenticidade da identidade do discente que preenche o formulário, já que este pode ser preenchido por qualquer pessoa com acesso ao link, sendo a autenticidade um meio de alcançar a confidencialidade. A falta de garantia da integridade dos dados pode afetar o resultado das avaliações, uma vez que após coletados os dados são armazenados em planilhas, podendo estes sofrerem alterações por terceiros, ou seja, pessoas com acesso aos dados que realizam alterações sem autorização. Os dados coletados precisam ser confiáveis, já que são analisados manualmente e os resultados das avaliações disponibilizados aos docentes, coordenadores de cursos de graduação da UFJF, chefes de departamento e membros de comissão de avaliação de estágio probatório docente.

Na Figura 2.26 é ilustrado um exemplo de formulário (*Google Forms*) utilizado pelo DCC para avaliar um docente em estágio probatório. No formulário constam o nome do docente avaliado e a turma pertencente à disciplina cursada, que são apresentados ao docente que realiza o preenchimento. Além disso, cada critério utilizado para avaliação possui entre cinco opções de resposta, que variam entre discordar totalmente (opção 1) a concordar totalmente (opção 5). Para cada questão é possível marcar apenas uma resposta, que ao final de todo o preenchimento do formulário esse é enviado e utilizado junto a outros preenchimentos como base na avaliação de um docente em estágio probatório.

Perguntas Respostas Configurações

### Avaliação Discente -- Prof. XXXX -- Turma DCCXXX-Y

Avaliação Discente do prof. XXXX refere à turma DCCXXX-Y - Disciplina ministrada no período letivo 20XX-Y.

Esta avaliação é anônima e servirá como apoio à avaliação que será feita pela Coordenação do Curso CCCCC.

As resposta devem ser dadas numa escala de 1 a 5, onde 1 indica "discordo totalmente" e 5 indica "concordo totalmente".

Cumpre o programa e objetivos da disciplina. \*

	1	2	3	4	5	
discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	concordo totalmente

Há coerência entre os objetivos propostos e os conteúdos das aulas. \*

	1	2	3	4	5	
discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	concordo totalmente

Figura 2.26: Formulário de avaliação docente em estágio probatório via *Google Forms*

## 2.11 Considerações Finais

Neste capítulo foram apresentados os conceitos, tecnologias e processos de *design* da Interação Humano-Computador utilizados no desenvolvimento deste trabalho de conclusão de curso. Os Princípios da *Gestalt* e as Heurísticas de *Design* de Nielsen foram apresentadas, que aliados à teoria de IHC da Engenharia Cognitiva possibilitam o *design* de *interfaces* de sistemas que proporcionem uma melhor experiência ao usuário. Tecnologias e padrões arquiteturais adotados no desenvolvimento de aplicações *web* utilizados no desenvolvimento deste trabalho também foram abordados. No próximo capítulo é apresentado o processo seguido para o desenvolvimento deste trabalho de conclusão de curso.

## 3 Metodologia: O Processo

Neste capítulo, é apresentada a estrutura do processo de desenvolvimento do Sistema de Avaliação Discente da UFJF (SAD-UFJF). Na Seção 3.1 é apresentada uma visão geral do processo de desenvolvimento (fases e as atividades). Na Seção 3.2 é apresentada as atividades da fase de análise da situação atual. Na Seção são descritas as atividades da fase de projeto da intervenção. Na Seção 3.4 são detalhadas as atividades da fase de avaliação. Na 3.5 são apresentadas as considerações finais capítulo.

### 3.1 Visão Geral do Processo

O processo de *design* foi utilizado na estruturação das atividades de preparação, desenvolvimento e testes do aplicativo implementado neste trabalho. O processo foi composto por três etapas: Análise da Situação Atual, Síntese da Intervenção e Avaliação da Intervenção. Como ilustrado na figura 3.1, a fase 1, referente a análise da situação atual é aplicada para que seja possível levantar informações que ajudem o desenvolvedor a ter maior conhecimento de um problema no qual o mesmo deseja solucionar. Para essa fase o desenvolvedor realiza algumas atividades, como foi o caso deste trabalho em que foram definidos o escopo do sistema e as metas de *design*, bem como a caracterização dos usuários que irão utilizar o aplicativo SAD-UFJF, levantamento e especificações dos requisitos funcionais e não funcionais presentes no sistema.

Na fase 2 o desenvolvedor realiza atividades para a construção da proposta de intervenção a um problema encontrado, sendo no caso deste trabalho o desenvolvimento de um aplicativo para apoiar na coleta e análise de avaliações discentes de docentes em estágio probatório, visando solucionar o problemas que podem ocorrer na utilização de formulários eletrônicos por parte do DCC da UFJF. As atividades desta etapa incluem a especificação dos cenários correspondentes às funcionalidades do aplicativo SAD-UFJF, modelagem de tarefas, prototipação dos cenários criados e implementação dos códigos de *back-end* e *front-end* do aplicativo.

Após a criação do produto (SAD-UFJF) o mesmo precisou ser testado. Dessa forma na fase 3 foi realizada a avaliação da intervenção. Nesta etapa foi realizado o teste funcional para verificar a conformidade da implementação com os requisitos do aplicativo, e avaliação heurística foi aplicada para avaliar a usabilidade e aspectos da experiência do usuário na interação com as *interfaces* desenvolvidas.

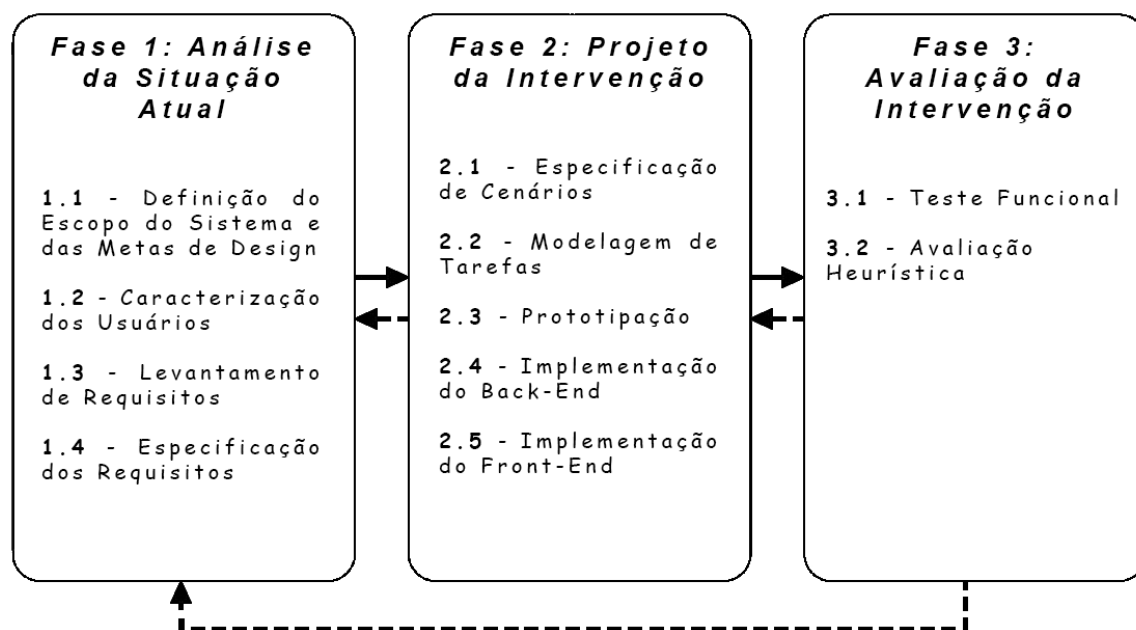


Figura 3.1: Fases das atividades do processo adotadas para desenvolver o aplicativo.  
Fonte: Autoria Própria

## 3.2 Fase 1: Análise da Situação Atual

Para a condução de atividades de levantamento de informações necessárias como base para a construção do aplicativo foram seguidas etapas de análise de IHC. Entre as etapas abordadas estão as seções de *brainstorm*, escopo do sistema e perspectiva do produto.

### 3.2.1 Escopo do Sistema

O sistema deverá ser capaz de auxiliar na coleta de dados de avaliações discentes da UFJF, possibilitando consultas e preenchimento de questionários, bem como análises por meio de gráficos e relatórios. O objetivo com o desenvolvimento do sistema é contribuir no processo avaliativo da UFJF, permitindo que o usuário discente seja capaz de avaliar

docentes em estágio probatório.

### 3.2.2 Metas de *design*

Como metas de *design* tem-se o critério de usabilidade da funcionalidade de preenchimento de formulário avaliativo discente. São seguidos os indicadores de número de erros e tempo de preenchimento do formulário de avaliação. São utilizados também critérios para garantir a confidencialidade no acesso ao formulário avaliativo e a integridade dos dados coletados e armazenados, bem como a redução no tempo para disponibilizar os resultados das avaliações.

### 3.2.3 Caracterização dos Usuários

Os usuários do sistema serão discentes matriculados em algum curso de graduação da UFJF, bem como docentes atuantes em departamentos da universidade. Ambos usuários deverão possuir cadastro no SIGA, para que tenham suas autenticações validadas no sistema e possam executar as funcionalidades presentes no aplicativo.

### 3.2.4 Requisitos Funcionais

Seções de *brainstorming* foram realizadas para o levantamento dos requisitos do sistema, necessários para a construção das funcionalidades do aplicativo SAD-UFJF.

Requisitos são declarações que apresentam, de forma detalhada, o que um sistema é capaz de fazer para cumprir com as necessidades dos usuários. Esses requisitos podem ser definidos como funcionais ou não-funcionais. Como o próprio nome sugere, requisitos funcionais estão relacionados às funcionalidades do *software*, que devem ser projetadas para atender às expectativas das partes interessadas no sistema, bem como a maneira que esse programa irá responder às ações executadas (GUEDES, 2018). Os requisitos funcionais identificados e utilizados para guiar a construção do sistema desenvolvido neste trabalho foram definidos por um identificador único, em que a numeração se inicia com o identificador [RF01] e prossegue sendo incrementada a cada surgimento de um novo requisito. Alguns desses requisitos funcionais podem ser conferidos na tabela 3.1. A tabela com todos os requisitos funcionais pode ser acessada no apêndice A.1.



Tabela 3.1: Requisitos funcionais do sistema

Identificador	Nome
RF01	Efetuar <i>login</i> no sistema.
RF02	Consultar informações de autenticação.
RF03	Consultar disciplinas.
RF04	Selecionar disciplina para o preenchimento do questionário.
RF05	Preencher questionário.
RF06	Salvar progresso.

### 3.2.5 Requisitos Não Funcionais

Os requisitos não-funcionais são implícitos e fazem parte das características do *software*, podendo estes serem determinados como "restrições, condições, consistências e validações que devem ser levadas a efeito sobre os requisitos funcionais" (GUEDES, 2018). Assim como nos requisitos funcionais, os não funcionais foram definidos por um identificador único, em que a numeração se inicia com o identificador [RNF01] e prossegue sendo incrementada a cada surgimento de um novo requisito. Esses requisitos não funcionais podem ser conferidos na tabela 3.2.

Tabela 3.2: Requisitos não funcionais do sistema

Identificador	Nome
RNF01	Segurança.
RNF02	Aspectos legais.
RNF03	Usabilidade.

### 3.2.6 Especificação dos Requisitos Funcionais

Para especificar os requisitos funcionais deste projeto, há a descrição de alguns pontos importantes que contribuem no entendimento das etapas do sistema, como identificador, nome, sumário, atores, pré-condições, descrição, alternativas, pós-condições e exceções. Como exemplo são apresentadas na tabela 3.3 as especificações para o requisito funcional "RF07", que corresponde a etapa final na submissão de um questionário para avaliação. Descrições detalhadas também para os demais requisitos funcionais podem ser conferidas nas tabelas B.1 à B.17, presentes no apêndice.

Tabela 3.3: Especificação do requisito funcional 'RF07'

Identificador	RF07
Nome	Enviar questionário.
Sumário	O caso de uso é responsável por submeter o questionário para avaliação.
Atores	Discente.
Pré-condições	Concluir o requisito funcional RF05.
Descrição	<b>Fluxo principal:</b> 1: O usuário faz clique sobre o botão "Enviar". 2: O sistema analisa as opções preenchidas. 3: O sistema exibe uma caixa de diálogo solicitando uma confirmação, com a opção "Sim", caso o usuário deseje realizar o envio do questionário para avaliação ou "Não", caso o mesmo opte por cancelar a ação. 4: O usuário faz clique sobre a opção "Sim".

Identificador	RF07
Descrição	<p>5: O sistema exibe uma caixa de diálogo informando que o envio foi concluído, com um campo "OK", para que a mensagem seja fechada.</p> <p>6: O usuário faz clique sobre o campo "OK".</p> <p>7: O sistema redireciona para a tela anterior ao preenchimento.</p>
Alternativas	<p><b>Alternativa ao passo 3 do fluxo principal:</b></p> <p>1: O sistema verifica que nem todas as questões possuem respostas. Há um redirecionamento para aquela que necessita ser preenchida, com exibição de uma caixa de diálogo informando ao usuário que todas as questões precisam ser respondidas, com um campo "OK", para que a mensagem seja fechada.</p> <p>2: O usuário faz clique sobre o campo "OK".</p> <p>3: O usuário responde a questão destacada.</p> <p>4: O processo retorna ao passo 1 do fluxo principal.</p> <p><b>Alternativa ao passo 4 do fluxo principal:</b></p> <p>1: O usuário faz clique sobre a opção "Não".</p> <p>2: O sistema não realiza o envio e permanece exibindo a mesma tela.</p>
Pós-condições	<p><b>Pós-condição:</b></p> <p>O status do questionário enviado é alterado, de "Pendente" ou "Incompleto" para "Concluído". Após alterado o status, o questionário é retirado da tela de seleção de disciplinas disponíveis para o preenchimento e enviado para a tela de questionários respondidos, não sendo mais permitidas alterações nas respostas. O sistema redireciona para a tela anterior ao preenchimento do questionário.</p>
Exceções	<p>O envio do questionário não poderá ser concluído caso o usuário deixe de preencher todas questões.</p>

### 3.2.7 Especificação dos Requisitos Não Funcionais

A especificação de requisitos não funcionais, assim como ocorre com requisitos funcionais, se mostra uma etapa essencial que deve ser criada. Com isso é possível detalhar as informações envolvidas em um projeto que são necessárias de serem identificadas, uma vez que podem limitar o funcionamento do sistema, assim como fatores externos ao desenvolvimento que devem ser atendidos no projeto. Dessa forma, para os requisitos não

funcionais, há a descrição de alguns pontos importantes que contribuem no entendimento das etapas do sistema, dentre eles: identificador, nome, categoria, prioridade e descrição. Como exemplo essas informações podem ser conferidas na tabela 3.4, referente a usabilidade. As especificações também para os demais requisitos não funcionais encontram-se descritas nas tabelas B.18 a B.20, presentes no apêndice.

Tabela 3.4: Especificação do requisito não funcional 'RNF03'

Identificador	RNF03
Nome	Usabilidade.
Categoria	Usabilidade.
Prioridade	Importante.
Descrição	Uso de design responsivo nas interfaces gráficas.

### 3.3 Fase 2: Projeto da Intervenção

Nesta etapa são apresentadas as atividades para a construção da proposta de solução ao problema encontrado na coleta de dados de avaliações discentes do DCC da UFJF, o desenvolvimento do aplicativo SAD-UFJF.

#### 3.3.1 Especificação dos Cenários para Detalhar os Requisitos Funcionais do Sistema

Segundo (BARBOSA; SILVA, 2017), diferentes tipos de cenários podem ser utilizados para representar de forma básica e fundamental todas as atividades presentes em uma solução de IHC. Um cenário é "simplesmente uma história sobre pessoas executando uma atividade" (ROSSON; CARROLL, 2002). Seu uso motiva a participação e contribuição dos interessados no sistema em decisões de *design*, de modo direto ou indireto. Com o emprego de cenários na elaboração de um sistema, a equipe de *design* tem a oportunidade

de observar como as atividades dos usuários são afetadas pela tecnologia utilizada. De acordo com (ROSSON; CARROLL, 2002), alguns elementos constituem um cenário, os quais são definidos a seguir:

- Contexto: "detalhes circunstanciais que motivam ou explicam objetivos, ações e reações dos atores do cenário";
- Evento: "ação ou reação externa produzida pelo computador ou outras características do ambiente";
- Avaliação: "atividade mental voltada para interpretar características de uma situação";
- Ação: "comportamento observável";
- Atores: "pessoas que interagem com o computador ou com o contexto";
- Objetivos: "efeitos na situação que motivam as ações que os atores realizam";
- Plano: "atividade mental voltada para converter um objetivo em comportamento".

Dessa forma, cenários foram elaborados como apoio ao desenvolvimento das funcionalidades do sistema SAD. Um desses exemplos é apresentado a seguir, para descrever o requisito funcional "RF05: preencher questionário", cujo ator fictício utilizado "João" atua como *persona* da história.

#### **Cenário para o requisito funcional "RF05: preencher questionário":**

[1] João, aluno de graduação do curso Sistemas de Informação da UFJF (Universidade Federal de Juiz de Fora), está utilizando o sistema SAD pela primeira vez este período. [2] Com esse aplicativo ele pretende compartilhar sua experiência em uma disciplina cursada, avaliando a conduta do docente e seu próprio desempenho. [3] Por meio do SAD, o aluno visualiza disciplinas disponíveis para avaliação e seleciona o questionário que deseja preencher, realizando clique sobre a disciplina desejada. [4] Dessa forma, questões relacionadas ao desempenho na disciplina escolhida são exibidas na tela (Descrição das questões nas tabelas 3.5 e 3.6), uma por vez, sendo possível acessá-las por meio de botões

de avançar e retroceder. [5] João começa a marcar uma opção de resposta para cada questão, que variam de 1 (péssimo) à 5 (excelente), até o final do questionário. Após o preenchimento de todas as questões, o aluno decide escrever uma observação para o coordenador de seu curso no campo exibido, realizando logo em seguida um clique no botão de envio para finalizar o questionário. João percebe que o sistema solicita uma confirmação do usuário, uma vez que após o envio não é mais possível alterar as respostas. O aluno examina suas escolhas e decide submeter o questionário para avaliação.

Cenários podem ser utilizados para responder questões capazes de serem exploradas na elaboração de uma história. No exemplo apresentado com as ações de João é possível perceber trechos do cenário numerados de 1 a 5, a fim de responderem a cada uma das seguintes perguntas:

- Qual o perfil do usuário? [1]
- Para que serve o aplicativo SAD-UFJF? [2]
- Como selecionar um questionário para avaliação? [3]
- Quais informações estão presentes no questionário? [4]
- Como realizar a avaliação do questionário? [5]

Tabela 3.5: Questões referentes ao docente, presentes no questionário avaliativo

1	O professor disponibilizou o plano de curso da disciplina na primeira semana de aula?
2	O professor é pontual?
3	O professor é assíduo às aulas?
4	O professor usa todo o tempo de aula?
5	O professor está disponível no horário de atendimento?
6	O professor tira dúvidas em sala de aula?
7	O professor apresenta os conteúdos da disciplina de forma clara?
8	O professor demonstra dominar o conteúdo da disciplina?
9	O material utilizado em aula é atualizado?
10	O professor apresenta exemplos em aula e faz ou propõe exercícios e/ou trabalhos práticos?
11	As avaliações refletem o conteúdo apresentado em sala?
12	As notas das avaliações foram publicadas até três dias antes da avaliação seguinte?
13	O conteúdo programático da disciplina foi cumprido?
14	Você acha que a bibliografia sugerida no Plano de Curso é adequada ao conteúdo da disciplina?
15	Os objetivos da disciplina foram atendidos?
16	Você considera a ementa da disciplina adequada ao seu curso?

Tabela 3.6: Questões referentes ao discente, presentes no questionário avaliativo

<b>17</b>	Você é pontual?
<b>18</b>	Você é assíduo às aulas?
<b>19</b>	Você procurou o professor, tutor ou monitor da disciplina para tirar dúvidas ao longo do semestre?
<b>20</b>	Você considera que possuía a base teórica necessária para cursar esta disciplina?
<b>21</b>	Você faz as atividades práticas propostas pelo professor?
<b>22</b>	Você considera que se dedicou suficientemente à disciplina?
<b>23</b>	Na sua opinião, suas notas refletem de modo fiel o conhecimento que você adquiriu ao longo da disciplina?

### 3.3.2 Modelagem de Tarefas

Com o objetivo de definir o plano de ações da Engenharia Cognitiva, foi utilizada a modelagem de tarefas, a qual apresenta metas dependentes, em que há uma ordem hierárquica de ações e sub-metas necessárias para que uma meta geral (presente no topo) seja alcançada. São identificadas as ações, os planos e intenções dos usuários para representar o comportamento dinâmico de sua interação com o sistema. Um exemplo pode ser conferido na ilustração da figura 3.2, onde a ação de "realizar *login* no sistema" é necessária para que seja possível atingir os planos, identificados pelas sub-metas "Acessar questionários disponíveis", "Selecionar questionário" e "Preencher questionário". Esses planos, por sua vez, precisam ser atingidos para que a intenção do usuário, caracterizada pela meta geral "Submeter questionário" seja alcançada.



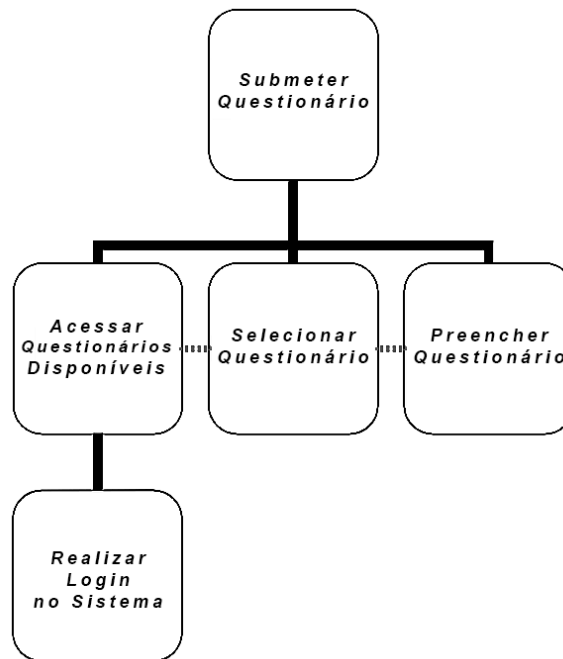


Figura 3.2: Modelo de tarefas para a meta geral de submeter questionário. Fonte: Autoria Própria.

### 3.3.3 Prototipação dos Cenários

Prototipação é um dos princípios-chave necessário em um processo de desenvolvimento para que seja possível cuidar da qualidade de uso de um sistema. Os protótipos em diferentes níveis devem ser elaborados, para que estes sejam visualizados e avaliados, a fim de se obter propostas de soluções (BARBOSA; SILVA, 2017). Existem diferentes métodos que podem ser utilizados na construção de protótipos, podendo variar os níveis de detalhes de acordo com tipo escolhido. A prototipação em papel, por exemplo, utilizada para a avaliação, é uma prática rápida e barata para a identificação de problemas de usabilidade antes de ser construída uma solução de IHC desejável. Esse método é interessante para uma avaliação junto aos usuários, sendo possível avaliar soluções parciais, sem cobrir toda a *interface*, com soluções de baixa e média fidelidade (BARBOSA; SILVA, 2017). *Softwares* de computador também permitem criação de protótipos que ajudam a identificar melhores soluções na elaboração das *interfaces*. Muitos fornecem ferramentas para construção de protótipos com resultados que possibilitam uma visão detalhada das etapas a serem desenvolvidas, por meio de *interfaces* que permitem a elaboração dos protótipos de maneira mais interativa. Alguns desses *softwares* podem ser citados, como: *Balsamiq*, *Marvel*, *Figma*, *Sketch*, entre outros. Para a criação dos protótipos para o sistema deste

trabalho, além da construção em papel, foi utilizado o *software Balsamiq*, que permitiu a elaboração de *interfaces* gráficas simplificadas, tomadas como base no processo de desenvolvimento da aplicação. Alguns dos exemplos gerados pelo *software* podem ser conferidos nas figuras 3.3 a 3.6. Prototipações para mais cenários se encontram ilustradas nas figuras C.1 a C.8, presentes no apêndice.

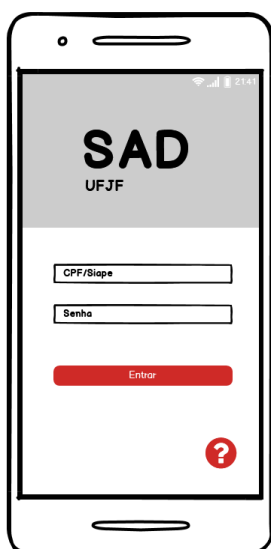


Figura 3.3: Protótipo da tela de autenticação. Sistema SAD-UFJF. Fonte: Autoria própria



Figura 3.4: Protótipo área do discente. Sistema SAD-UFJF. Fonte: Autoria própria



Figura 3.5: Protótipo área do docente. Sistema SAD-UFJF. Fonte: Autoria própria



Figura 3.6: Protótipo do menu lateral discente. Sistema SAD-UFJF. Fonte: Autoria própria

## 3.4 Avaliação de IHC

Para avaliar as *interfaces* utilizadas no aplicativo foram aplicados métodos de Interação Humano-Computador. De acordo com (BARBOSA; SILVA, 2017) uma avaliação de IHC é empregada na verificação de interação e *interface* de um produto, onde são verificadas se estas atendem critérios de qualidade de uso, considerados prioritários na análise de uma situação. A avaliação de IHC pode ser realizada durante o processo de *design* ou depois de finalizado o produto, devendo a qualidade de uso sempre ser avaliada desde o início do

processo de *design*, uma vez que custos para correções de possíveis problemas podem ser menores quando identificados nas fases iniciais.

Para este trabalho os métodos aplicados na avaliação de IHC foram os testes funcionais e avaliações heurísticas. Os testes funcionais basearam-se na execução de funcionalidades do aplicativo, para que fosse verificado se estas cumprem com seus objetivos e retornam para os usuários informações esperadas. A avaliação heurística foi aplicada para que possíveis problemas presentes nas *interfaces* pudessem ser identificados. Com o emprego da avaliação heurística pôde-se observar se boas práticas foram aplicadas, ou não, na criação das telas do aplicativo, por meio de diretrizes seguidas para avaliar a conformidade das *interfaces* com Heurísticas de Nielsen e Princípios da *Gestalt*.

Ambos os métodos de avaliação de IHC utilizados seguiram etapas na aplicação das análises, dentre elas preparação, coletas de dados e consolidação dos resultados. A etapa de preparação foi responsável por permitir a identificação de informações necessárias, como usuários e domínio, assim como a escolha das *interfaces* avaliadas e os métodos utilizados. A coleta de dados ocorreu conforme o planejamento realizado e o método de avaliação selecionado. As atividades envolveram apenas um usuário, o qual examinou as *interfaces* para identificar suas funcionalidades e discrepâncias com o padrão adotado, bem como a percepção de fatores que influenciam na experiência do usuário. Por fim, na consolidação dos resultados as questões que motivaram os estudos foram respondidas ou justificadas, sendo os problemas encontrados relatados, no caso da avaliação heurística, com descrições mais detalhadas que mostram como essas adversidades podem afetar a usabilidade. As descrições das etapas realizadas no teste funcional e avaliação heurística podem ser conferidas no capítulo 5.

## 3.5 Considerações Finais

Foi apresentada a metodologia seguida no processo, com atividades abordadas como base no entendimento dos caminhos necessários na construção da aplicação. Para a obtenção de uma visão geral do processo foram descritas atribuições utilizadas na análise de IHC, com destaque nos elementos que ajudam a elaborar um documento de requisitos, fundamentais no apoio ao período de desenvolvimento. Assim como a elaboração de atividades presentes

---

no *Design* de IHC, com criação de cenário, modelagem de tarefas e prototipações, como auxílio ao desenvolvedor no momento de construção de *interfaces*. Foram mencionados também métodos utilizados da avaliação de IHC para a realização de testes de desempenho e interação. Os temas abordados neste capítulo apoiaram a etapa de desenvolvimento, sendo fundamentais no processo de criação do aplicativo.

## 4 SAD-UFJF: Sistema de Avaliação Discente

Neste capítulo é apresentada uma visão geral da arquitetura e das funcionalidades da versão protótipo do Sistema de Avaliação Discente (SAD-UFJF) desenvolvida na forma de um aplicativo móvel. Na Seção 4.1 é apresentada uma visão geral dos módulos que compõem a arquitetura do sistema. Na Seção 4.2 é apresentado o modelo conceitual da aplicação. Uma descrição mais detalhada das implementações do *back-end* do projeto do aplicativo é apresentada na Seção 4.3. Na Seção 4.4 é descrita a implementação do *front-end*. Na Seção 4.5 é apresentada a descrição de dois cenários de uso do aplicativo, um para o usuário discente e outro para o usuário docente.

### 4.1 Arquitetura do Aplicativo

Na Figura 4.1 é apresentada uma visão geral dos elementos que compõem a arquitetura do aplicativo SAD-UFJF. A arquitetura do aplicativo foi construída com base no padrão *Model-View-Controller* (Seção 2.8). Na camada *Model* foi utilizado o sistema de gerenciamento de banco de dados não-relacional *MongoDB* (MongoDB, 2022b) (Subseção 2.9.4) para armazenamento e recuperação de dados. No *Controller*, foi utilizado o *framework NodeJS* (NodeJS, 2021) (Subseção 2.9.2) implementar as funcionalidades do sistema disponibilizadas na forma de serviços. Além do *NodeJS*, o *framework ExpressJS* (Express, 2022) (Subseção 2.9.3) também foi utilizado na implementação de mecanismos de controle navegacional e tratamento de exceções das *interfaces (front-end)* do aplicativo. Finalmente, na camada *View*, foi utilizado o *framework JavaScript React Native* (React Native, 2022) (Subseção 2.9.1) para o desenvolvimento das *interfaces* gráficas do aplicativo.

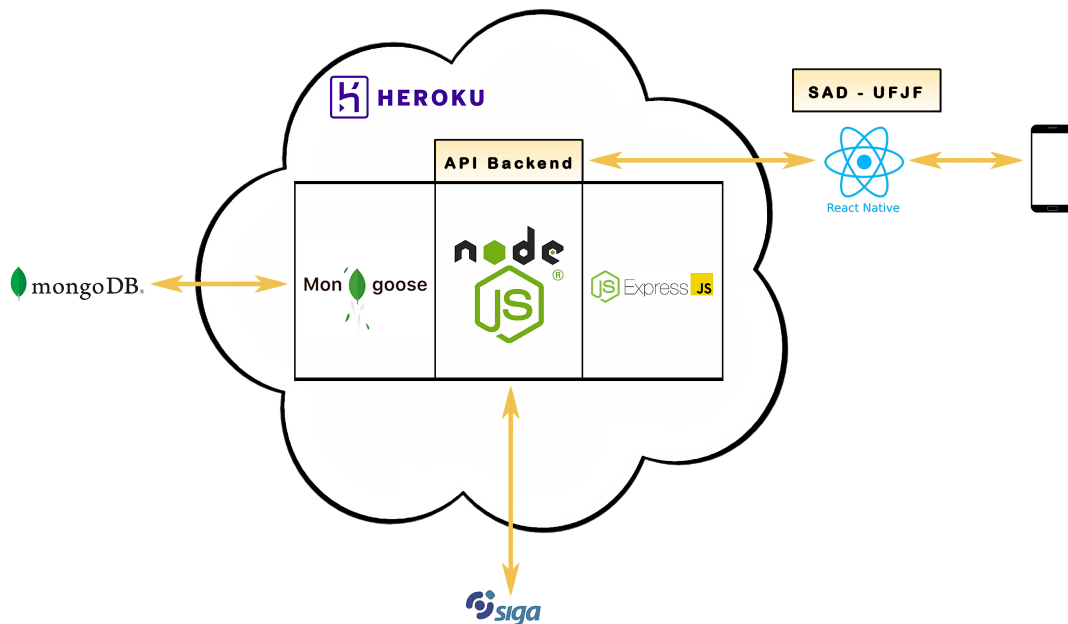


Figura 4.1: Arquitetura do sistema para o funcionamento do *software*. Fonte: Autoria Própria.

A camada *View*, representada pelas *interfaces* implementadas em *React Native*, alteram o modelo (*Model*), ou seja, as informações referentes às entidades do domínio do problema como Usuário e Avaliação Discente armazenadas em uma base de dados *MongoDB*, e notificam o *Controller*. O sistema de gerenciamento de banco de dados não-relacional (*NoSQL*) *MongoDB* foi utilizado na camada *Model* para apoiar o armazenamento e a recuperação de informações em uma base de dados. Um sistema gerenciador de banco de dados não-relacional foi escolhido em virtude de sua eficiência no processo de busca e flexibilidade que facilita futuras modificações na base de dados sem afetar os registros previamente armazenados.

O controlador obtém informações da visão, por exemplo, via formulários submetidos com informações referentes a uma *Avaliação Discente* de uma determinada disciplina, e atualiza o modelo, ou seja, atualiza a base de dados *MongoDB* com a adição de mais um registro. Por outro lado, o *Controller* comunica-se com a camada *Model*, representada pelo sistema gerenciador de banco de dados não-relacional *MongoDB*, para recuperar informações da base de dados do aplicativo SAD-UFJF e atualiza a camada *View* (*interfaces* do sistema) a partir de informações recuperadas do modelo. Isso pode ocorrer em situações nas quais o usuário interage com a *interface* do sistema por meio do acionamento de uma funcionalidade de busca com base em um determinado critério para obter

informações sobre um determinado registro, e.g., consulta a uma avaliação registrada e ainda não submetida ou acesso à informações de perfil.

A biblioteca *Mongoose* (Mongoose, 2022) (Subseção 2.9.5) foi utilizada para estabelecer a conexão entre o código *back-end* dos serviços de persistência e recuperação de dados implementados utilizando o *framework NodeJS* e o sistema de gerenciamento de banco de dados *MongoDB*. A plataforma em nuvem *Heroku* (Heroku, 2022) (Subseção 2.9.6) foi utilizada para a realização do *deploy* do código *back-end* do sistema, por possuir integração com a plataforma de hospedagem *GitHub*, facilitando o controle de versões da aplicação.

O funcionamento do SAD-UFJF requer informações relacionadas às matrículas e disciplinas cursadas pelos discentes fornecidas pelo Sistema de Informação Gestão Acadêmica (SIGA) da UFJF. A autenticação do usuário no SAD-UFJF será realizada por meio de seu cpf e senha do SIGA, garantindo a autenticidade da identidade do discente quem responderá cada questionário de avaliação de disciplina que o mesmo está devidamente matriculado. Bem como o docente quem irá consultar o desempenho referente a uma disciplina avaliada por discentes de uma turma. A autenticação do usuário no SAD-UFJF será realizada via comunicação do código *back-end* dos serviços de persistência *NodeJS* do aplicativo (*Controller*) com o SIGA-UFJF. Dessa forma, evitando que discentes não matriculados (as) na disciplina ou pessoas não autorizadas realizem o preenchimento do questionário de forma indevida. As informações relacionadas às disciplinas nas quais cada discente está matriculado (a) também serão obtidas via comunicação do código *back-end* (serviços *NodeJS* da camada *Controller*) do aplicativo com o SIGA-UFJF. Por esse motivo, a comunicação da versão protótipo do aplicativo SAD-UFJF, apresentada neste trabalho de conclusão de curso, com os serviços de autenticação e consulta de disciplinas fornecidos pelo SIGA-UFJF será posteriormente implementada para viabilizar a implantação do aplicativo.

A introdução do aplicativo na UFJF tem o potencial de apoiar o Processo de Avaliação Institucional da Qualidade de Ensino de Graduação da Pró-Reitoria de Graduação. A versão protótipo do aplicativo apresentada neste trabalho de conclusão de curso ainda não possui comunicação com os serviços de autenticação e consulta de disciplinas do

SIGA-UFJF. Tratativas com Centro de Gestão de Conhecimento Organizacional (CGCO) da UFJF já foram iniciadas com o objetivo de obter acesso aos serviços de autenticação e consulta a serem fornecidos pelo CGCO na forma de uma *Application Programming Interface* (API) para viabilizar a implantação e o início da operação do SAD-UFJF na instituição.

## 4.2 Modelo Conceitual

Nesta seção é apresentado o modelo conceitual que representa as entidades do domínio do sistema de apoio ao registro e análise de dados de avaliações discentes da qualidade do ensino de disciplinas dos cursos graduação da UFJF e seus relacionamentos. Na Figura 4.2 é mostrado o modelo conceitual do aplicativo SAD-UFJF com as entidades de domínio, seus atributos e relacionamentos na notação Entidade-Relacionamento (ER) (CHEN, 2002). Usuário, Disciplina, Turma, Questionário, Pergunta (Questão) e Resposta são as entidades de interesse do domínio do aplicativo SAD-UFJF, cujas informações serão armazenadas e manipuladas.

A entidade *Usuário* armazena as seguintes informações de cada usuário do aplicativo: o número do *Cadastro de Pessoa Física* (CPF), o identificador único de um usuário do sistema, a senha de acesso ao sistema, o nome, o tipo de usuário, que pode ser discente

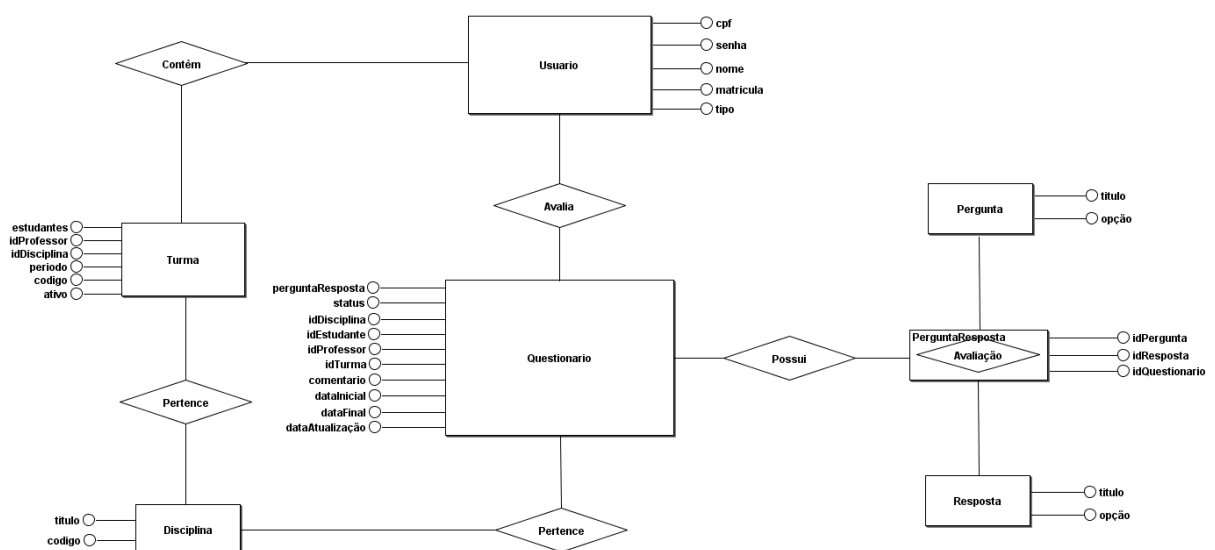


Figura 4.2: Modelo entidade-relacionamento do SAD-UFJF.



(aluno) ou docente, e o número de matrícula do usuário, que é o identificador único do usuário dentro da instituição. Um usuário do tipo *docente* pode estar associado a zero ou muitas *Turmas*, sendo o responsável por ministrar uma determinada *Disciplina* para uma *Turma*. Um usuário discente pode estar matriculado em zero ou muitas turmas.

Cada turma está associada a uma determinada *Disciplina*. É possível que haja mais de uma turma para uma mesma *Disciplina*. Tal situação ocorre na disciplina DCC119 Algoritmos que possui várias turmas, e.g., DCC119 Turma A, DCC119 Turma B etc. Cada *Turma* pode ter um ou mais *Usuários* do tipo *docente* como responsáveis. Por esse motivo, a entidade *Usuário* possui um relacionamento do tipo  $1..N$  com a entidade *Turma*. Este relacionamento também denota que um *Usuário* do tipo *discente* pode estar alocado (matriculado) em zero ou múltiplas *Turmas*.

A entidade *Turma* armazena as seguintes informações: o *código* da *Disciplina* (*idDisciplina*) associada, o *código* da turma (e.g., A), o *período* no qual uma turma de uma determinada disciplina está sendo oferecida, e.g., primeiro semestre (2022-1) ou segundo semestre letivo (2022-3), e um *flag* rotulado como *ativo* para armazenar a situação de uma determinada turma, ou seja, se a turma está ativa ou não. Uma turma não ativa corresponde a uma turma que já foi fechada. A informação relacionada à situação de uma Turma (ativa ou fechada) é utilizada pelo aplicativo para disponibilizar os questionários de avaliação apenas para as *Disciplinas* nas quais os discentes estejam com matrículas ativas no período corrente. Tal verificação evita que questionários de avaliação de *Turmas* de *Disciplinas* oferecidas em períodos anteriores sejam exibidos aos usuários do aplicativo. Caso um determinado *Usuário* do tipo *discente* tenha efetuado o trancamento de sua matrícula em uma determinada *Turma*, o referido discente não será mais vinculado à turma e o mesmo não precisará preencher o *Questionário* de avaliação da referida *Disciplina* associada à *Turma*.

A entidade *Disciplina* armazena o *nome* e o *código*, que é o identificador único de uma disciplina. Mais de uma *Turma* pode estar associada a uma mesma *Disciplina*, denotado pelo relacionamento  $1..N$  rotulado como *pertence* entre *Disciplina* e *Turma*. Cada disciplina pode estar associada a zero ou muitos *Questionários* a depender da quantidade de discentes matriculados respondentes, conforme denotado pelo relacionamento  $1..N$  en-

tre *Disciplina* e *Questionário* rotulado como *pertence*. Um *Usuário* do tipo *discente* pode responder zero ou mais questionários, dependendo da quantidade de disciplinas que o mesmo está matriculado no período corrente, denotado pelo relacionamento 1..N rotulado como *avalia* na Figura 4.2.

Um *Usuário* do tipo *discente* pode responder a zero ou múltiplos questionários de avaliação de disciplinas denotado pelo relacionamento 1..N rotulado como *avalia*. Um usuário discente somente poderá avaliar a qualidade de ensino, via preenchimento de pesquisa, para *Disciplinas* do período corrente que estiver com *matrícula* ativa. A entidade *Questionário* armazena as seguintes informações a respeito de um questionário de avaliação discente: os códigos identificadores da *Disciplina*, *Turma* e do Usuário respondente, as *datas* de *início* do preenchimento, da *última atualização* e de *envio* do questionário, o *status* do questionário, que pode ser submetido ou em preenchimento e o atributo *comentário*, utilizado para armazenar um comentário sobre a conduta docente na disciplina. Um *Questionário* é composto por várias *Questões (Perguntas)*. Cada *Pergunta* possui um código identificador único, um título e uma resposta (*opção*). Uma *Pergunta* de múltipla escolha pode ter várias opções de *Resposta*. O relacionamento entre as entidades *Pergunta* e *Resposta* resultou na entidade *PerguntaResposta*. Esta entidade derivada do relacionamento rotulado como *PerguntaResposta* armazena as respostas a cada questão de múltipla escolha do questionário de avaliação de disciplina.

Nas seções subsequentes são apresentados o projeto detalhado do *back-end* (camada *Controller*) e do *front-end* (camada *View*) do aplicativo SAD-UFJF.

### 4.3 Visão Geral do *Back-End*

Nesta seção é apresentada a implementação da *API back-end* das funcionalidades do aplicativo utilizando o *framework NodeJS* e o sistema gerenciador de banco de dados *MongoDB*. Essa *API* consiste em uma *interface* de programação de aplicações em conformidade com a arquitetura *REST (Representational State Transfer)* (FIELDING; TAYLOR, 2000). O código *back-end* de cada funcionalidade de consulta e persistência de dados do aplicativo foi implementado seguindo o protocolo *HTTP (HyperText Transfer Protocol)* no qual os serviços foram disponibilizados por meio de *interfaces POST, GET*

e *DELETE*.

Os serviços (funcionalidades do aplicativo) são acessíveis via *URLs*. Serviços de inserção e atualização de dados foram implementados utilizando o método *POST*. Serviços de consulta foram implementados utilizando método *GET* e serviços de exclusão de registros utilizam o método *DELETE*. Dessa forma, requisições a cada um desses serviços geram como resposta do servidor *HTTP* um código de *status* (*status code*) que indica a situação do processamento da requisição. Por exemplo, a execução bem sucedida de um serviço retorna o código *HTTP 200*. A execução mal sucedida de um serviço retorna o código *HTTP 400*.

O formato de armazenamento e troca de dados *JSON* (*JavaScript Object Notation*) (JSON, 2022) foi utilizado na implementação das funcionalidades do aplicativo na forma de serviços *REST*. O formato *JSON* foi escolhido por ser amplamente utilizado para troca de dados entre aplicações *web*, por fornecer uma forma de armazenamento de dados simples, de fácil escrita e interpretação. O formato *JSON* é utilizado no envio de dados do servidor para a página *web* e da página *web* para o servidor. Um arquivo *JSON* é composto por duas estruturas: uma coleção de pares nome/valor, utilizada para caracterizar estruturas de objeto, registro, *struct*, dicionário, *hash table*, lista; e uma lista ordenada de valores que caracteriza um *array*, vetor, lista ou sequência. Em *JSON*, os dados são apresentados como objetos. Um objeto é um conjunto desordenado de pares nome/valor. A especificação de um objeto começa com '{', a chave de abertura, e termina com '}', a chave de fechamento. Cada nome é seguido por ':' (dois pontos) e os pares nome/valor são separados por ',' vírgula (JSON, 2022). Na Figura 4.3 é mostrado um exemplo de objeto *JSON* que contém informações de um registro da entidade *Turma*. O objeto contém o *status* da turma (*active*) que possui o valor *true* (linha 9), o código da turma (*id*) com o valor '*6148d5d2...*' (linha 10) o *período*, o código (*code*) e a data de criação da turma (*createdAt*) (linhas 27-29). O objeto *Turma* contém um objeto do tipo *professor* (linhas 11-19), um objeto do tipo *disciplina* (linhas 20-26) e um conjunto de referências a objetos do tipo *student* (linhas 4-8).

```
1 {  
2   "classes": [  
3     {  
4       "students": [  
5         "6148d168c615762908f124df",  
6         "6148d190c615762908f124e0",  
7         "6148d1b4c615762908f124e1"  
8       ],  
9       "active": true,  
10      "_id": "6148d5d2c615762908f124e3",  
11      "idProfessor": {  
12        "_id": "6148d05dc615762908f124dd",  
13        "cpf": "12248966357",  
14        "name": "Fulano da Silva",  
15        "type": "P",  
16        "registration": "202165879",  
17        "createdAt": "2021-09-20T18:18:05.254Z",  
18        "__v": 0  
19      },  
20      "idDiscipline": {  
21        "_id": "6148d0dbc615762908f124de",  
22        "title": "Interação Humano-Computador",  
23        "code": "DCC174",  
24        "createdAt": "2021-09-20T18:20:11.717Z",  
25        "__v": 0  
26      },  
27      "period": "2021/1",  
28      "code": "A",  
29      "createdAt": "2021-09-20T18:41:22.138Z",  
30      "__v": 0  
31    }  
32  ]  
33 }
```

Figura 4.3: Formato *JSON*. Fonte: Autoria Própria.

O formato de serialização binária *BSON* (*Binary JSON*) foi utilizado para armazenar documentos em coleções e fazer chamadas a procedimentos remotos executados no sistema gerenciador de banco de dados *MongoDB*. O formato *BSON* fornece uma estrutura baseada em documentos semelhante a objetos *JSON*, que codifica informações como tipo e tamanho do objeto, permitindo que objetos sejam serializados e deserializados mais rapidamente. A estrutura de objeto *BSON* adiciona suporte a tipos de dados como *Data* e binário não são suportados pelo formato *JSON*. O *MongoDB* armazena dados no formato *BSON* e qualquer representação em *JSON* pode ser armazenada nativamente no *MongoDB* e recuperada com a mesma facilidade do formato *JSON* (MongoDB, 2022a). Na Tabela 4.1 é mostrada uma análise comparativa das características dos formatos *JSON* e *BSON* com respeito à codificação de caracteres adotada, os tipos de dados suportados e a quem esses formatos são legíveis.

Tabela 4.1: Comparação dos formatos *JSON* e *BSON*. Fonte: Adaptado de (MongoDB, 2022a)

	JSON	BSON
<b>Codificação</b>	UTF-8 String	Binary
<b>Suporte de dados</b>	String, Boolean, Number, Array	String, Boolean, Number (Integer, Float, Long, Decimal128...), Array, Date, Raw Binary
<b>Legibilidade</b>	Humano e Máquina	Somente máquina

### 4.3.1 Implementação da Camada Modelo

Para cada entidade conceitual do modelo entidade-relacionamento da Figura 4.2 foi implementada uma classe entidade correspondente utilizando o *framework NodeJS*. As classes de entidade *Usuário*, *Turma*, *Disciplina*, *Questionário*, *Pergunta*, *Resposta* e *Pergunta-Resposta* e seus relacionamentos mostrados na Figura 4.4 representam a camada *Model* do projeto do aplicativo. Essas classes têm a finalidade de apoiar a manipulação de dados captados da camada *View* enviados à camada *Controller* e transportar os dados recuperados a partir de consultas ao banco de dados *MongoDB* da camada *Controller* até a camada *View*.

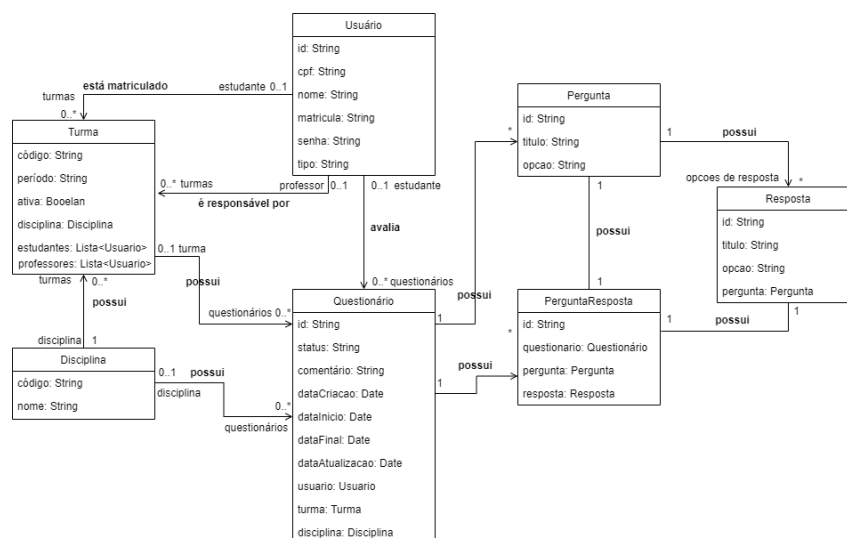


Figura 4.4: Diagrama de classes de entidade do aplicativo.

A entidade *Usuário* possui dois relacionamentos de cardinalidade  $1...^*$  com a

entidade *Turma* (Figura 4.4). O primeiro relacionamento denota que um usuário do tipo estudante pode estar alocado a zero ou muitas *Turmas*. O segundo relacionamento representa a associação entre *Usuários* do tipo *docente* responsáveis por zero ou mais *Turmas*. Um *Usuário* do tipo estudante pode responder a zero ou mais *Questionários* de avaliação de disciplina. Uma *Turma* pode estar associada a zero ou mais *Questionários*. Um *Questionário* é composto por um conjunto de questões (*Perguntas*). Cada *Pergunta* de múltipla escolha pode ter várias opções de *Resposta*. A classe *PerguntaResposta* é oriunda do relacionamento entre as classes *Pergunta* e *Resposta* e tem a finalidade de armazenar informações sobre as perguntas associadas a um determinado *Questionário*, permitindo a criação de questionários personalizados. Outra finalidade da classe de associação *PerguntaResposta* é armazenar as respostas a cada pergunta de um *Questionário* fornecida por um *Usuário estudante*.

Na Figura 4.5 é mostrado um trecho de código da classe de entidade *Questionário* do aplicativo declarado com um schema utilizando a biblioteca Mongoose (Mongoose, 2022). Cada *schema* possui mapeamento para uma coleção (*collection*) do modelo de dados no MongoDB e define a estrutura de cada documento dentro desta coleção (entidade do modelo de dados). Na linha 1 é declarada a constante "*mongoose*" relacionada à biblioteca *Mongoose* utilizada para estabelecer conexão com o sistema gerenciador de banco de dados *MongoDB* e criar a coleção. Posteriormente, a estrutura do documento *Questionário* (*schema*) é declarada na linha 3. Um *Questionário* avaliativo contém os seguintes atributos: *perguntaResposta* (linhas 4-7), que consiste em um *array* que faz referências a um conjunto de documentos do tipo *PerguntaResposta*, *status* (linhas 8-11), que representa a situação de um questionário de avaliação de disciplina, que pode ser: *não respondido* (*N*), *finalizado* (*S*) ou *incompleto* (*I*), *disciplina* e *turma*, que fazem referências a documentos dos tipos *Disciplina* (linhas 12-16) e *Turma* (linhas 27-31) respectivamente. Cada documento *PerguntaResposta* do *Questionário* faz referência a um documento do tipo *Pergunta*, que armazena o *título* e a *opção de resposta* selecionada pelo usuário após o preenchimento do questionário, e referências a um conjunto de possíveis *Respostas* a uma determinada *Pergunta*.

Cada documento do tipo *Resposta* armazena o *título* e um valor numérico (*opção*)

correspondente à resposta uma determinada pergunta. Por exemplo, um documento *Resposta* com o *título* "*Excelente*" e *opção* com valor "*5*". Cada registro (objeto) do tipo *Questionário* possui referências a um documento *Usuário* do tipo *estudante* (linhas 17-21), que corresponde a quem respondeu ao questionário. Cada *Questionário* armazena um texto de opinião (*comentário*) (linhas 32-35) do *Usuário estudante* sobre a qualidade do ensino de uma *Disciplina* oferecida por um *Usuário docente* a uma determinada *turma*. Finalmente, cada *Questionário* também armazena as seguintes informações: *data de criação* (*createdAt*) (linhas 36-39), *data de abertura do Questionário* (*dataInício*) (linhas 40-42), a *data final* (linhas 43-45) na qual o questionário foi encerrado, sem possibilidade de futuras edições, e a *data da última atualização* feita pelo *Usuário estudante* (linhas 46-48).

A função assíncrona *save* é definida na linha 51 para adicionar um novo documento do tipo *Questionario* à respectiva coleção na base de dados *MongoDB*. A constante *Questionário* foi criada na linha 56 para converter o *schema* questionário em um modelo de documento que possibilita manipular dados do tipo *Questionário*. Por fim, o comando de atribuição *module.exports = Questionario* é converter um *schema* recuperado da base de dados *MongoDB* em um objeto no formato *JSON* para tornar possível a manipulação desses dados nas classes *controladoras* codificadas *NodeJS* (NodeJS, 2021).

```
1  const mongoose = require('mongoose')
2
3  const questionarioSchema = mongoose.Schema({
4    perguntaResposta: [{
5      type: mongoose.Schema.Types.ObjectId,
6      ref: 'PerguntaResposta'
7    }],
8    status: {
9      type: String,
10     default: 'N'
11   },
12   idDisciplina: {
13     type: mongoose.Schema.Types.ObjectId,
14     ref: 'Disciplina',
15     required: true,
16   },
17   idEstudante: { ...
21   },
22   idProfessor: { ...
26   },
27   idTurma: { ...
31   },
32   comentario: { ...
35   },
36   createdAt: { ...
39   },
40   dataInicio: { ...
42   },
43   dataFinal: { ...
45   },
46   dataAtualizacao: { ...
48   }
49 })
50
51 questionarioSchema.pre('save', async function (next) { ...
54 })
55
56 const Questionario = mongoose.model('Questionario', questionarioSchema)
57
58 module.exports = Questionario;
```

Figura 4.5: Trecho de código NodeJS da classe entidade *Questionário* do sistema SAD-UFJF.

### 4.3.2 Implementação da Camada Controle

O *controller* é a camada intermediária entre a visão (*View*) e a camada de negócio (*Model*) responsável por atualizar o modelo e renderizar a visão. Após a criação das classes de entidade da camada *modelo*, as classes controladoras que manipulam o modelo foram criadas utilizando o *framework NodeJS*. Classes controladoras foram criadas para as classes de entidade Usuário, Turma, Questionário, Disciplina, Pergunta, Resposta e Pergunta-Resposta (Figura 4.4). Cada classe controladora contém funções de registro, atualização e busca sob uma coleção da base de dados *MongoDB*. Essas funções foram implementadas e disponibilizadas na forma de serviços que são consumidos pelos módulos *front-end* do



SAD-UFJF para cadastrar ou atualizar registros em coleções ou recuperar registros de coleções na base de dados *MongoDB*.

Na Figura 4.6 é mostrado um trecho de código fonte *NodeJS* da função *register* da classe controladora para a entidade *Questionário*. Esta função manipula coleções dos tipos *Questionário*, *PerguntaResposta*, *Turma* e *Pergunta*. Por esse motivo, constantes de cada um desses tipos foram criadas (linhas 3-6). A função *register* possibilita a persistência de um registro do tipo *Questionário* na base de dados utilizando o método de requisição *POST* do protocolo *HTTP*. Esta função é acessível através de uma requisição à *url*: *http://mydomain/registro/*.

A especificação da função assíncrona *register* está contida dentro da instrução *router.post* (linhas 10-33). Primeiramente, o *status* do questionário, os *códigos* da *turma* e do *professor*, juntamente com as *datas de início* e de *submissão* do questionário são passados para a função via requisição *post* (linha 12). Caso já exista um questionário de avaliação de disciplina com o mesmo código de turma e de usuário já armazenado na base de dados, a função retorna uma *mensagem de erro* (*HTTP 400*) informando que o usuário já submeteu um questionário para aquela turma (linhas 14-16). Tal verificação é feita pela chamada à função *findOne* (linha 14) da classe de controle da entidade *Questionário*. Caso contrário, a função *create* da classe *Questionário* é chamada passando os parâmetros: *s código da turma*, *código do professor*, *dataInício* e *dataFinal* para iniciar o procedimento de registro de um novo questionário na base de dados. Posteriormente, as respostas a cada uma das perguntas do questionário são passadas como parâmetro e mapeadas para as respectivas perguntas (linhas 22-26). Finalmente, o questionário é persistido na base de dados (linha 28) e a mensagem de *status HTTP 200* é retornada após o envio do formulário (linha 29). Caso tenha ocorrido algum erro durante o registro do questionário, uma mensagem de *status HTTP 400* é enviada (linha 31).

```

1  const express = require('express');
2
3  const PerguntaResposta = require('../models/perguntaResposta');
4  const Questionario = require('../models/questionario');
5  const Turma = require('../models/turma');
6  const Pergunta = require('../models/pergunta');
7
8  const router = express.Router();
9
10 router.post('/registro', async (req, res) => {
11   try {
12     const { status, idDisciplina, idEstudante, idProfessor, idTurma, dataInicio, dataFinal } = req.body;
13
14     if (await Questionario.findOne({ status, idDisciplina, idEstudante, idProfessor, idTurma })) {
15       return res.status(400).send({ error: 'Questionário já existente!' });
16     }
17
18     const questionario = await Questionario.create({ status, idDisciplina, idEstudante, idProfessor, idTurma, dataInicio, dataFinal });
19
20     const perguntas = await Pergunta.find();
21
22     await Promise.all(perguntas.map(async idPergunta => {
23       const resPerguntaResposta = new PerguntaResposta({ idPergunta: idPergunta._id, idQuestionario: questionario._id });
24       await resPerguntaResposta.save();
25       questionario.perguntaResposta.push(resPerguntaResposta);
26     }));
27
28     await questionario.save();
29     return res.status(201).send({ questionario });
30   } catch (err) {
31     res.status(400).send({ error: 'Erro ao registrar questionário!' + err });
32   }
33 });

```

Figura 4.6: Trecho de código NodeJS da classe *controller* para a entidade *Questionário* do sistema SAD-UFJF.

## 4.4 Visão Geral do *Front-End*

Nesta seção é apresentada a implementação do *front-end* (*interfaces gráficas*) do aplicativo utilizando o *framework React Native* (React Native, 2022). O uso do *framework React Native* possibilitou o desenvolvimento de *interfaces gráficas HTML* e *CSS* ricas com controle de lógica interna em *JavaScript* e que exibem comportamento nativo, ou seja, o comportamento das *interfaces* é automaticamente adaptável às características cada plataforma operacional alvo, e.g., *Android* e *iOS*. O código *front-end* de cada funcionalidade representa a camada de visualização (*View*) do aplicativo, que compreende o único meio pelo qual o usuário interage com sistema, tanto na entrada de dados quando na apresentação da informações nas *interfaces*.

No aplicativo SAD-UFJF, há *interfaces gráficas* (telas) com interação baseada em formulários para a entrada de dados de autenticação, preenchimento de dados de questionário de avaliação discente e consultas a questionários, como ilustrado no modelo navegacional do aplicativo da Figura 4.7. O aplicativo também contém as seguintes *interfaces gráficas* de saída de dados que fornecem respostas às ações do usuário: *Área do Usuário Discente/Docente*, *Listagem de Disciplinas* que possuem questionários a serem respondidos (*Questionários não Concluídos*) e *Listagem de Disciplinas* com questionários

já respondidos (*Questionários Concluídos*) e *Preenchimento de Questionário* disponíveis a usuários discentes e telas de exibição de *Disciplinas para Análise*, de *Gráfico de Barras* das médias de respostas de questionários referentes a avaliação discente de uma disciplina (*Geração de Gráfico*) e *Geração de Relatórios* disponíveis a usuários docentes. As telas de autenticação (*login*), de *Consultas de Disciplinas* e de informações *Sobre o Aplicativo* são comuns a ambos os usuários.

As entradas do usuário captadas pelas *interfaces* da camada *View* são encaminhadas à camada *Controller* via requisições a serviços *POST* das classes controladoras *Usuário* e *Questionário* disponíveis em *urls*, e.g., *questionario/registro*. As classes controladoras atualizam o modelo, notificam e atualizam *View* em resposta à cada ação do usuário.

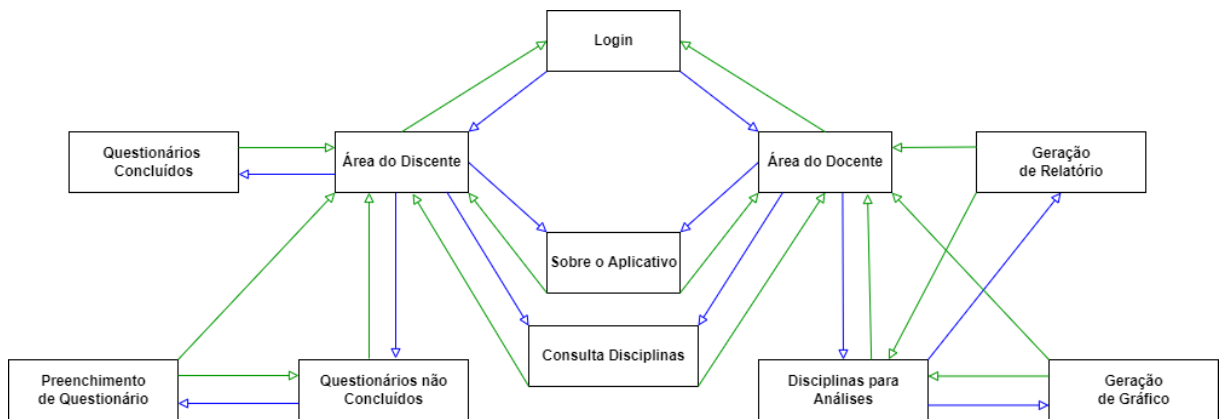


Figura 4.7: Modelo navegacional das telas do sistema SAD-UFJF.

#### 4.4.1 Descrição das Telas do SAD-UFJF

A tela de *Login* é a primeira apresentada pelo aplicativo SAD-UFJF. Esta tela consiste em uma *interface* baseada em formulário para a captura do CPF e da senha para a autenticação do usuário no aplicativo como ilustrado na Figura 4.8. Caso as credenciais do usuário sejam válidas, é liberado o acesso a área do usuário discente ou docente, dependendo do tipo de usuário autenticado.

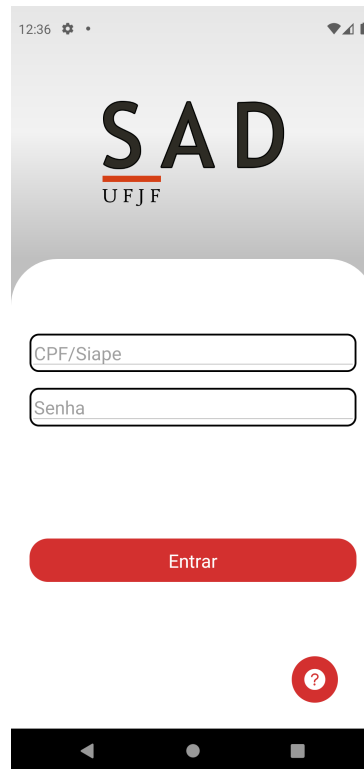


Figura 4.8: Tela de login do sistema SAD-UFJF.

A Área do Discente é a tela principal exibida após a autenticação de um usuário discente (Figura 4.9). Esta tela fornece acesso ao usuário discente às seguintes funcionalidades: preenchimento de questionário de avaliação de disciplina, *Visualização* de questionários que não tiveram o seu preenchimento concluído, *Visualização* de questionários cujos preenchimentos já foram concluídos, *Consultas* à disciplinas nas quais o discente tem matrícula ativa no período letivo e *Visualização* das informações sobre o aplicativo SAD-UFJF. Essas funcionalidades estão acessíveis por meio do acionamento de itens de menu, como ilustrado na Figura 4.9.

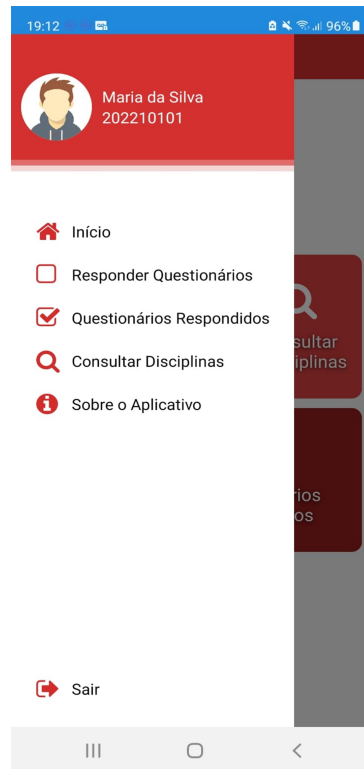


Figura 4.9: Área do usuário discente.

A tela de questionários não concluídos mostrada na Figura 4.10 apresenta a listagem das disciplinas que possuem questionários com situação de não respondidos ou incompletos. Esses questionários referem-se à disciplinas cursadas pelo usuário discente no período letivo corrente. Os questionários são exibidos enquanto o período de avaliação dos mesmos encontra-se aberto, sendo apresentados somente aqueles cuja situação encontra-se com preenchimento em andamento, ou seja, os que ainda não foram submetidos, ou aqueles nos quais ainda não foram respondidos.

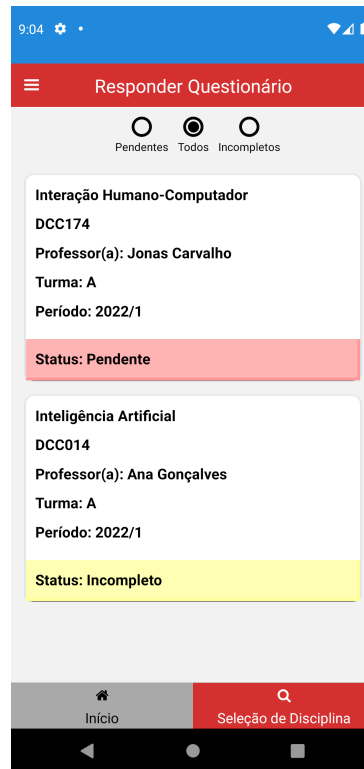


Figura 4.10: Tela de questionários de avaliação não respondidos.

Na Figura 4.11 é mostrada a tela com a listagem dos questionários de avaliação de disciplinas que já foram preenchidos e submetidos (finalizados), ou seja, questionários que já foram submetidos e armazenados na base de dados. As respostas aos questionários são base para a avaliação da qualidade do ensino das disciplinas cursadas e geração de gráficos de barras e tabelas.

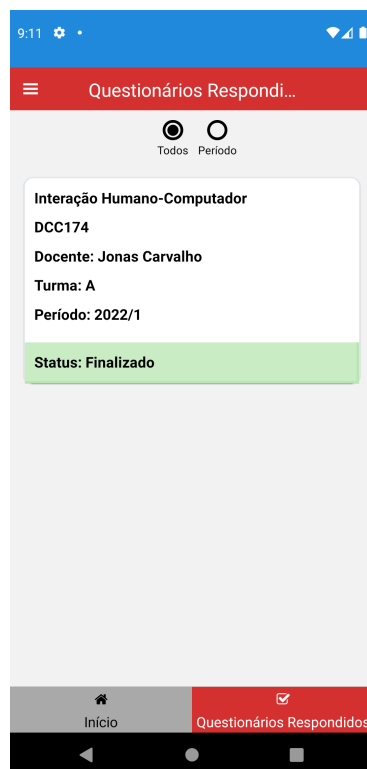
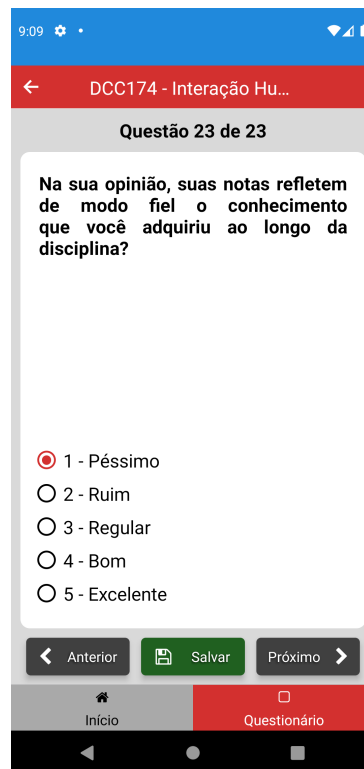


Figura 4.11: Tela de questionários de avaliação submetidos.

Na Figura 4.12 é mostrada a tela de preenchimento de questionário de avaliação de disciplina. Esta tela é acionada após a seleção do questionário com preenchimento pendente ou não inicializado na tela de questionários não respondidos (Figura 4.10). O questionário contém um conjunto de 23 questões para avaliar a qualidade de ensino e a postura do docente na condução da disciplina e um conjunto de questões de autoavaliação do discente com respeito ao seu desempenho na disciplina. As questões são de múltipla escolha e o discente deve selecionar uma das cinco opções: 1 - Péssimo, 2 - Ruim, 3 - Regular, 4 - Bom ou 5 - Excelente. Após fornecer respostas a cada questão, o discente poderá postar um comentário à respeito de sua avaliação pessoal da disciplina a ser encaminhado ao docente e ao chefe de departamento.



The screenshot shows a mobile application interface for a discipline evaluation questionnaire. At the top, there is a status bar with the time 9:09 and various icons. Below it is a red header bar with a back arrow and the text "DCC174 - Interação Hu...". The main content area is titled "Questão 23 de 23". The question text is: "Na sua opinião, suas notas refletem de modo fiel o conhecimento que você adquiriu ao longo da disciplina?". Below the question, there are five radio button options: "1 - Péssimo", "2 - Ruim", "3 - Regular", "4 - Bom", and "5 - Excelente". The first option, "1 - Péssimo", is selected. At the bottom of the question area, there are three buttons: "Anterior", "Salvar", and "Próximo". The bottom navigation bar has two items: "Início" with a home icon and "Questionário" with a document icon and a red background.

Figura 4.12: Tela de preenchimento de questionário de avaliação de disciplina.

Na Figura 4.13 é mostrada a tela da área do usuário docente exibida após a autenticação no sistema. Esta tela fornece acesso às seguintes funcionalidades do usuário docente: *Visualização de disciplinas* a serem selecionadas para a geração de gráficos e relatórios com um *ranking* médio das respostas obtidas dos discentes a cada pergunta do questionário de avaliação de uma determinada disciplina, *Consulta* à disciplinas nas quais o docente está lecionando no período letivo corrente e a *Visualização* de informações sobre o SAD-UFJF. Essas funcionalidades estão acessíveis através de itens de menu (botões) na área do usuário docente.





Figura 4.13: Área do usuário docente.

Na Figura 4.14 é mostrada a tela de seleção de disciplinas para visualizar os gráficos e os relatórios com os resultados das análises das respostas obtidas dos discentes para cada disciplina ministrada no período letivo corrente. Nesta tela é exibida a listagem de disciplinas que o usuário docente está lecionando no período letivo corrente com as seguintes informações: nome, código, turma, e docente responsável. Essas mesmas informações são exibidas na tela de consulta de disciplinas nas quais um usuário discente possui matrícula ativa no período letivo corrente. Ao selecionar uma disciplina, o usuário docente poderá visualizar o gráfico de barras e o relatório de avaliação de seu desempenho em uma determinada disciplina oferecida a uma turma.

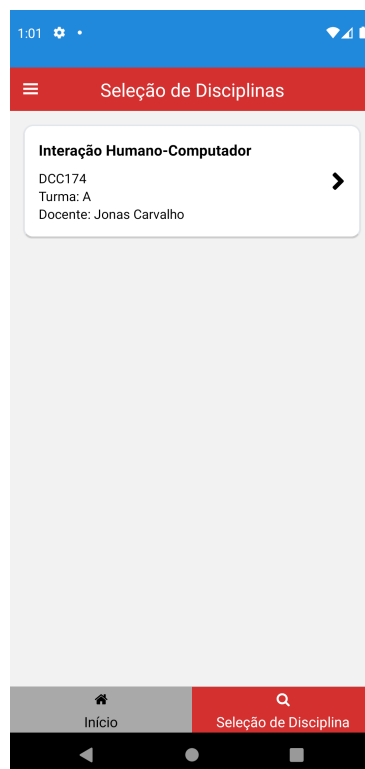


Figura 4.14: Tela de seleção de disciplinas usuário docente.

No exemplo da Figura 4.14 é listada a disciplina Interação Humano-Computador na qual o docente Jonas Carvalho é o responsável no período letivo corrente. Ao selecionar a disciplina Interação Humano-Computador, o docente poderá visualizar o gráfico de barras com as médias das respostas obtidas dos discentes para cada critério avaliativo relacionado a seu desempenho na disciplina como: 1 - *Disponibilidade*, 2 - *Pontualidade*, 3 - *Assiduidade*, 4 - *Utilização do tempo de aula*, 5 - *Disponibilidade para atendimento*, 6 - *Esclarecimento de dúvidas*, 7 - *Clareza na explicação dos conteúdos*, 8 - *Domínio do conteúdo*, 9 - *Atualização do material*, 10 - *Apresentação de exemplos*, dentre outros, como ilustrado na Figura 4.15.

Na tela da Figura 4.15, o docente obtém uma visualização do seu desempenho médio em cada um dos critérios avaliativos em uma escala de 0 - *Péssimo* a 5 - *Excelente* a partir do *ranking* médio das respostas a cada questão do questionário de avaliação de disciplina fornecidas pelos discentes matriculados na turma na qual a disciplina foi oferecida pelo docente. O docente e o chefe de departamento têm a opção de visualizar o gráfico de barras com ou sem a presença de *outliers* nas amostras. Um *outlier* consiste em um questionário preenchido por um discente que contém respostas que apresentam gran-

des diferenças comparado com as respostas obtidas de outros discentes para as mesmas questões. Por exemplo, um questionário com respostas com valor 1 - Péssimo para os 23 itens avaliativos fornecido por um único discente enquanto os demais discentes forneceram respostas iguais ou superiores à 3 para todos os critérios avaliativos. Neste contexto, é possível visualizar o gráfico de barras e o relatório de avaliação discente para uma determinada disciplina considerando a presença ou a ausência de amostras *outliers*. No caso do exemplo da Figura 4.15, é mostrado o resultado da avaliação da disciplina DCC174 Interação Humano-Computador considerando a presença de *outliers*.

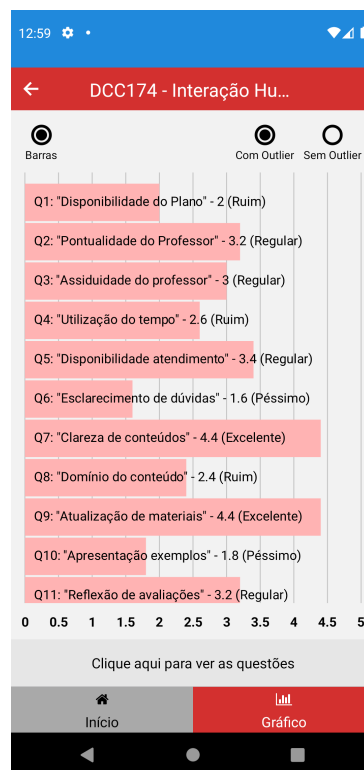
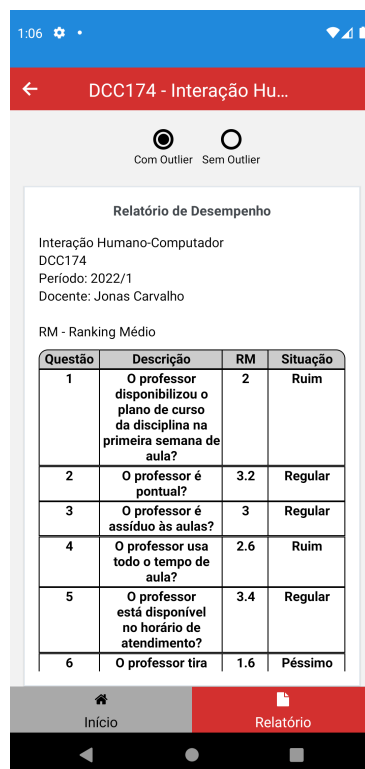


Figura 4.15: Tela de exibição de gráfico de barras de avaliação de disciplina.

Na tela de relatório de avaliação da Figura 4.16 é exibida, além das informações da disciplina como: nome, código, período e docente, uma tabela com a listagem de cada pergunta do questionário de avaliação discente, os valores de *ranking* médio das respostas recebidas para cada pergunta e a classificação de desempenho docente em cada critério avaliativo de acordo com cada valor médio obtido, sendo 1 - *Péssimo*, 2 - *Ruim*, 3 - *Regular*, 4 - *Bom* ou 5 - *Excelente*. Também é possível gerar o relatório considerando ou não respostas *outliers*, ou seja, respostas que apresentam alta discrepância em relação a maioria das respostas obtidas dos discentes.



1:06

← DCC174 - Interação Hu...

Com Outlier Sem Outlier

**Relatório de Desempenho**

Interação Humano-Computador  
DCC174  
Período: 2022/1  
Docente: Jonas Carvalho

RM - Ranking Médio

Questão	Descrição	RM	Situação
1	O professor disponibilizou o plano de curso da disciplina na primeira semana de aula?	2	Ruim
2	O professor é pontual?	3.2	Regular
3	O professor é assíduo às aulas?	3	Regular
4	O professor usa todo o tempo de aula?	2.6	Ruim
5	O professor está disponível no horário de atendimento?	3.4	Regular
6	O professor tira	1.6	Péssimo

Início Relatório

Figura 4.16: Tela de exibição de relatório de avaliação de disciplina.

#### 4.4.2 Implementação da Camada Visão

Nesta subseção é apresentada a implementação *front-end React Native* (React Native, 2022) das telas da funcionalidade de *Preenchimento de Questionário de Avaliação Discente* do aplicativo. Para orquestrar o roteamento e a navegação pelas telas (páginas) do aplicativo SAD-UFJF, rotas foram criadas para mapear as sequências de navegação. Componentes reutilizáveis de navegação como cabeçalhos (*headers*), *menus* e *botões* foram criados para possibilitar a navegação entre as *interfaces* (telas). O uso desses componentes facilita o desenvolvimento por meio da reutilização, evitando a reescrita de código relacionado à navegação pelas telas do aplicativo. Todas as telas do aplicativo SAD-UFJF foram projetadas utilizando o mesmo *layout* e esquema de cores, tamanho e tipos de fonte organizados em folhas de estilos codificadas em arquivos *Cascading Style Sheets (CSSs)*.

O armazenamento de configurações de formatação de cabeçalhos, rodapés, botões e caixas de texto em folhas de estilos *CSS* também favorece o reuso, acelerando o desenvolvimento do *front-end* do aplicativo. A implementação das telas que compõem o *front-end* da aplicação foi realizada com base em protótipos especificados com o apoio da ferramenta de prototipação *Balsamiq*. Boas práticas de *design* de IHC como os princípios da *Gestalt*

(CARD; MORAN; NEWELL, 1983) e as Heurística de Usabilidade de Nielsen (NIELSEN, 1993) foram aplicados na especificação dos protótipos do sistema SAD-UFJF.

Na Figura 4.17 é mostrado o trecho de código fonte *React Native* da tela de preenchimento de questionário de avaliação discente. A classe *Questionario* estende a classe *Component* do *React Native* (linha 11) e o método construtor da classe renderiza o estado da tela de questionário com base em informações sobre as questões e opções de resposta a cada questão relacionadas ao questionário obtidas através de requisições ao *back-end* da aplicação nas funções *emphgetPerguntas*, *getRespostas* e *inicial*.

A função *inicial* (linhas 32-43) é responsável por montar dinamicamente a lista de perguntas e as respectivas opções de resposta de cada pergunta a serem exibidas na tela de questionário com base nas informações retornadas da base de dados. A função *inicial* é invocada dentro da função *componentDidMount* (linhas 25-30) após chamadas às funções *getPerguntas* e *getRespostas*. A função *atualizaEstado* (linhas 76-87 da Figura 4.18) altera o estado atual do questionário com base nas ações do usuário de seleção de opção de resposta (via *radiobuttons*) para cada questão. A função *salvaEstado* (linhas 89-104) persiste o registro de uma resposta ao questionário de avaliação de disciplina fornecidas por um discente na base de dados *MongoDB*, atualiza a tela informando se questionário foi submetido com sucesso ou se houve algum erro, via mensagem de *status* de requisição *HTTP*, e remete o usuário discente para a tela de seleção de disciplinas.

```

11 export default class Questionario extends Component {
12   constructor(props) {
13     super(props);
14     this.state = {
15       perguntas: [],
16       respostas: [],
17       pergunta_resposta: [],
18       index: 0,
19       comentario: '',
20       radioColor: AppColors.radioColor1
21     };
22     this.questionario = this.props.navigation.getParam('questionario');
23   }
24
25   async componentDidMount() {
26     this.setState({ pergunta_resposta: this.questionario.perguntaResposta });
27     await this.getPerguntas();
28     await this.getRespostas();
29     await this.inicial();
30   }
31
32   inicial = async () => {
33     const { pergunta_resposta, perguntas, respostas } = this.state;
34
35     const lista = pergunta_resposta.map(object => {
36       const pergunta = perguntas.filter(obj => obj._id == object.idPergunta)[0];
37       const resposta = respostas.filter(obj => obj._id == object.idResposta)[0];
38       return ({
39         ...object,
40         idResposta: resposta,
41         idPergunta: pergunta
42       })
43     });

```

Figura 4.17: Implementação do *front-end* para a tela de questionário do Sistema SAD-UFJF.

```

76   atualizaEstado = (value) => {
77     const { respostas, index } = this.state;
78     let { pergunta_resposta } = this.state;
79     const resposta = respostas.filter(object => object.option == value)[0];
80
81     pergunta_resposta[index] = {
82       ...pergunta_resposta[index],
83       idResposta: resposta
84     };
85
86     this.setState({ pergunta_resposta, radioColor: AppColors.radioColor1 });
87   }
88
89   salvaEstado = async () => {
90     try {
91       const { pergunta_resposta, comentario } = this.state;
92       const { idQuestionario } = pergunta_resposta[0];
93       const lista = {
94         idQuestionario,
95         comentario,
96         status: 'I',
97         perguntaResposta: pergunta_resposta
98       }
99       const response = await Api.put('/questionario/atualiza', lista)
100       this.props.navigation.navigate('TelaSelecaoDisciplinas')
101     } catch (error) {
102       console.log(error)
103     }
104   }

```

Figura 4.18: Implementações do *updateState* e *SaveState* no *front-end* para a tela de questionário do Sistema SAD-UFJF.

## 4.5 Cenários de Uso do Aplicativo SAD-UFJF

Nesta seção é apresentado o modelo navegacional detalhado do aplicativo SAD-UFJF juntamente com a descrição dos seguintes cenários de uso: *Preenchimento de Questionário de Avaliação Discente para uma Disciplina* do usuário discente (subseção 4.5.1) e *Visualização de Gráfico de Barras de Avaliação de Desempenho Docente em uma Disciplina* do usuário docente (subseção 4.5.2).

O modelo navegacional detalhado do aplicativo SAD-UFJF é mostrado na Figura 4.19. Neste diagrama é ilustrada todas as sequências de transições de telas possíveis de serem percorridas pelos usuários discente e docente em uma seção de navegação pelo aplicativo. Dentre as funcionalidades possíveis de serem acessadas/alcançadas por usuários no papel de discente estão a *autenticação (login)* no sistema, *selecionar disciplina* para o preenchimento de questionário de avaliação que inclui preencher o questionário com respostas à cada questão, salvar alterações no questionário e enviar/submeter o questionário para análise, consultar informações de autenticação (*login* e senha), informações sobre do aplicativo ou disciplinas na qual o discente possui matrícula ativa no período letivo corrente.

Para um usuário docente, as funcionalidades disponíveis são: *autenticação (login)* no sistema, *consulta* à disciplinas na qual o mesmo leciona no período letivo corrente, consulta a informações de autenticação e informações sobre o aplicativo, selecionar disciplinas para a geração de gráficos e relatórios de desempenho docente, análise com base na avaliação da conduta docente em uma determinada turma pelos discentes quanto aos critérios avaliativos da qualidade do ensino de uma disciplina cursada. As setas coloridas em roxo denotam os caminhos de navegação entre as funcionalidades (telas) do aplicativo que podem ser percorridas por usuários do tipo discente e as setas em vermelho representam os caminhos de navegação que podem ser percorridos por usuários docentes.

Após a autenticação no sistema, o usuário discente pode acessar as telas de consulta de disciplinas, de questionários concluídos, de informações sobre o aplicativo e de seleção de disciplina para o preenchimento de questionário de avaliação discente. Na tela de preenchimento de questionário, o usuário discente poderá percorrer cada tela para responder à cada questão do questionário. O discente pode optar por salvar o questionário e

continuar o preenchimento em um outro momento ou submeter o questionário no sistema. O usuário docente pode acessar as seguintes funcionalidades: selecionar disciplina para a visualização de gráficos e relatórios de avaliação de desempenho com e sem *outliers*.

A funcionalidade de visualização de gráfico de barras fornece um *feedback* ao docente sobre a qualidade de ensino e seu desempenho na condução de determinada disciplina oferecida em uma turma. Esses gráficos são gerados a partir das respostas à questionários de avaliação de disciplina preenchidos pelos discentes ao final de cada período letivo. A dependência entre essas duas funcionalidades tornam ambas de grande importância na avaliação da conduta docente nas disciplinas dos cursos de graduação da UFJF. Dessa forma, os cenários de uso relacionados às funcionalidades de preenchimento de questionário de avaliação de disciplina e visualização de gráficos de barras são detalhados nas seções subseqüentes.

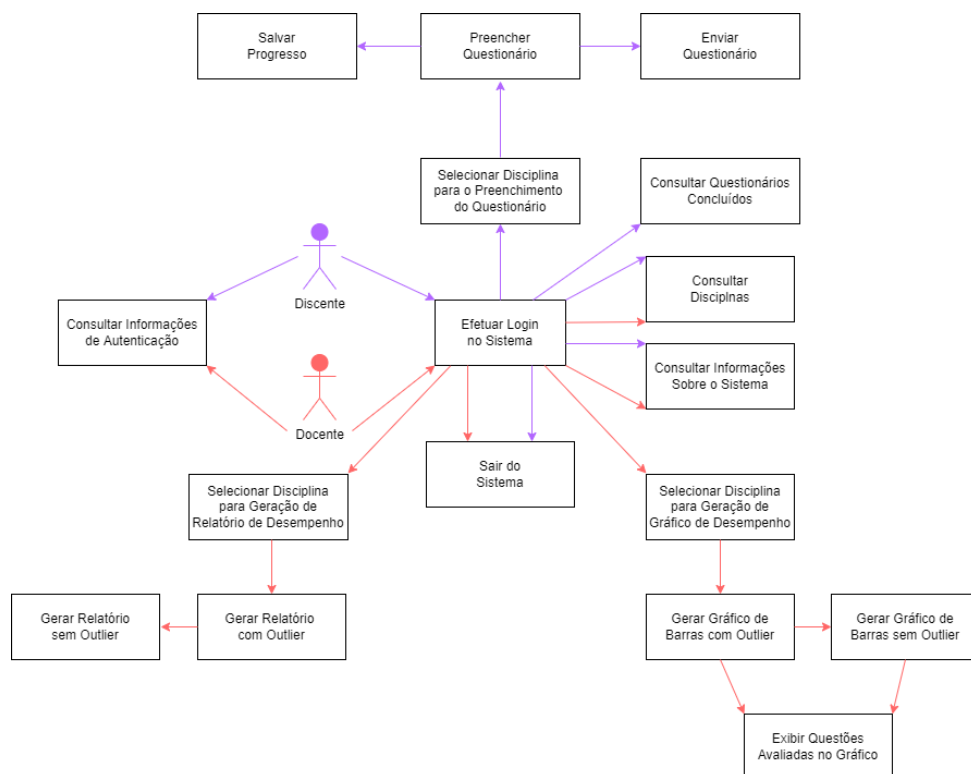


Figura 4.19: Modelo navegacional das telas do aplicativo.

#### 4.5.1 Preenchimento de Questionário de Avaliação Discente

Na Figura 4.20 é mostrada a sequência de transição de telas necessárias para um usuário discente realizar o objetivo de preencher um questionário de avaliação de uma disciplina.



Primeiramente, o usuário precisa autenticar no sistema fornecendo o *login* e a senha (Figura 4.20a), selecionar o item de menu *Responder Questionário* na tela que exibe o *menu* do usuário (Figura 4.20b). Posteriormente, o usuário é redirecionado à tela que exibe as disciplinas que o discente está matriculado no período letivo corrente que possuem questionários de avaliação a serem respondidos (Figura 4.20c), e.g., DCC174 e DCC014. Após o usuário selecionar uma das disciplinas, a tela de preenchimento da primeira questão do questionário de avaliação da referida disciplina, e.g., DCC174, é exibida como ilustrado na Figura 4.20d.

Ao acessar a tela de questionário (figura 4.20 (d)), o discente deve analisar as questões apresentadas e selecionar apenas uma das cinco opções de resposta disponíveis para cada tema exibido. Todas as questões precisam ser respondidas para que o envio do questionário possa ser realizado, devendo o usuário avançar a cada pergunta e marcar a opção que melhor reflete seu grau de satisfação acerca do tema apresentado. Ao concluir o preenchimento, o discente precisa enviar o questionário (figura 4.20 (e)) para que as respostas nele contidas sejam utilizadas, juntamente com as respostas dos demais alunos da turma, na avaliação da qualidade de ensino e do desempenho docente na condução de determinada disciplina de graduação da UFJF.

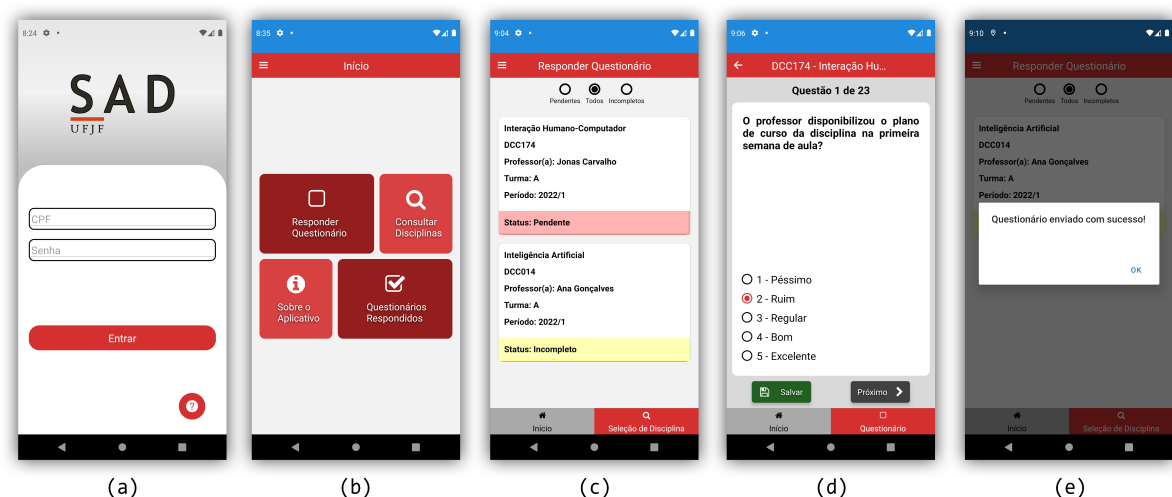


Figura 4.20: Sequência de passos necessários para realizar o envio de um questionário de avaliação discente.

### 4.5.2 Visualização de Gráfico de Barras de Avaliação de Desempenho Docente

Na Figura 4.21 é mostrada a sequência de transição de telas necessária para a realização do cenário de visualizar o gráfico de avaliação de desempenho docente em uma disciplina. Primeiramente, o usuário docente precisa estar autenticado no sistema após fornecer as suas credenciais (CPF e senha) de acesso ao sistema (Figura 4.21a). Após a autenticação, o docente deverá selecionar o item de *menu Consultar Desempenho* (Figura 4.21b). Posteriormente, o usuário será remetido à tela de seleção de disciplinas (Figura 4.21c). Nesta tela, o usuário deverá selecionar uma disciplina na qual deseja visualizar o gráfico de barras que contém informações sobre o desempenho do docente em cada item avaliativo do questionário. Após selecionar uma disciplina, a tela com informações da disciplina como nome, código e o gráfico de barras do desempenho docente é exibida ao usuário (Figura 4.21d). Os resultados da avaliação de desempenho fornecem um *feedback* ao docente sobre a condução da disciplina.

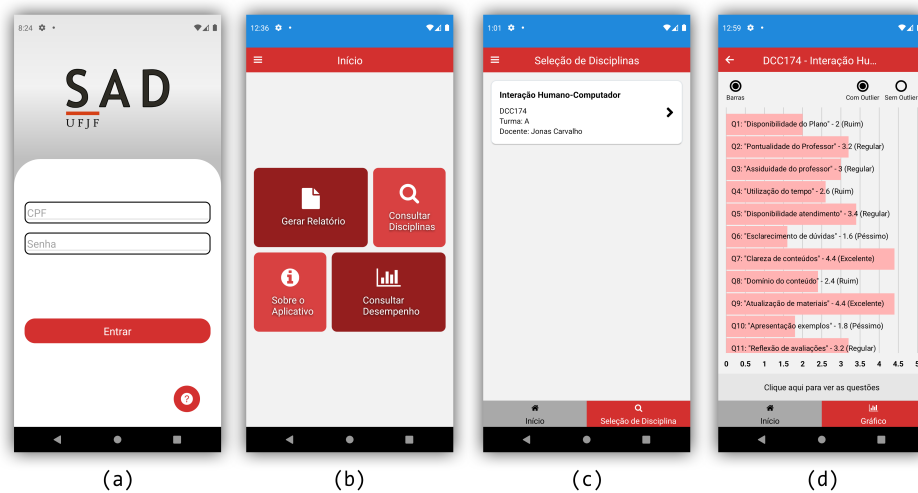


Figura 4.21: Sequência de passos necessária para gerar um gráfico de avaliação de desempenho docente em uma disciplina.

O instalador do aplicativo SAD-UFJF está disponível para *download*<sup>1</sup>, a fim de que funcionalidades possam ser conferidas. Devido ao motivo de ainda não haver integração do aplicativo com o sistema SIGA da universidade, foi necessário cadastrar usuários fictícios. Para autenticar como discente é necessário informar o CPF: "1111111111", e

<sup>1</sup>[https://drive.google.com/drive/folders/19Psa-8FRhIslGUTmU\\_ImtPd606rTHXc\\_?usp=sharing](https://drive.google.com/drive/folders/19Psa-8FRhIslGUTmU_ImtPd606rTHXc_?usp=sharing)

informar "00000000000" para autenticar como docente. Ambos com a senha "123".

## 4.6 Considerações Finais

Neste capítulo foi apresentada uma visão geral da arquitetura *Model-View-Controller* do aplicativo SAD-UFJF. Também foi apresentado o modelo conceitual e descrições da implementação do *back-end* com trechos de código fonte de classes das camadas *Model* e *Controller* utilizando a biblioteca *Mongoose* e o *framework NodeJS*. Além disso, a descrição da implementação das telas desenvolvidas (*front-end*) utilizando o *framework React Native* também foi apresentada, juntamente com a descrição de dois cenários de uso do aplicativo.

O aplicativo foi desenvolvido com base no processo da Engenharia de Usabilidade (NIELSEN, 1993) (Capítulo 3) e em conformidade com os princípios e diretrizes de *design* da IHC. Para verificar a aderência a esses princípios, as *interfaces* gráficas da versão protótipo do aplicativo precisam ser avaliadas. No próximo capítulo é apresentada a avaliação do aplicativo no qual testes funcionais foram aplicados para avaliar a correte de suas funcionalidades e o método de Avaliação Heurística foi aplicado para avaliar a aderência do projeto de *interfaces* aos princípios e heurísticas de *design*.

## 5 Fase 3: Avaliação da Intervenção

Neste capítulo são apresentados métodos aplicados para avaliar a corretude das funcionalidades e o projeto de *interfaces* do aplicativo desenvolvido. Na seção 5.1 são realizados testes funcionais para verificar a conformidade da implementação com os requisitos do sistema. Na seção 5.2 são realizadas avaliações heurísticas para avaliar a usabilidade e aspectos da experiência do usuário ao interagirem com as *interfaces* desenvolvidas no aplicativo.

### 5.1 Teste Funcional

Com o objetivo de realizar uma análise acerca da eficiência de funcionalidades presentes no aplicativo SAD-UFJF desenvolvido neste trabalho, foram implementados testes funcionais, os quais são compostos pelas etapas de Preparação, Execução e Coleta de Dados, Análise e Conclusão dos Resultados.

#### 5.1.1 Preparação

A submissão de questionários preenchidos e a visualização de gráfico de barras com resultados de avaliações discentes foram as funcionalidades testadas do aplicativo. Para a realização dos testes foi necessário a criação de cadastros no banco de dados utilizado, registrando seis usuários fictícios com CPF/Siape e senha, sendo um docente ("Jonas") e cinco discentes ("Maria", "Pedro", "Carolina", "Bruno" e "Marcos"). Foi necessária também a criação de uma disciplina ("Interação Humano-Computador"), com turma identificada como "A" e inicializada no período letivo "2022/1". Os registros fictícios criados para os testes foram realizados pelo motivo do aplicativo SAD-UFJF ainda não possuir integração com o sistema SIGA da universidade. Porém, uma ideia futura de integração com o sistema da instituição seria útil na validação dos usuários reais na etapa de autenticação do aplicativo.

### 5.1.2 Execução e Coleta de Dados

Cada discente registrado com matrícula ativa na turma respondeu à 23 questões referentes à conduta do docente em estágio probatório que ministrou a disciplina cursada, bem como questões de auto desempenho do discente respondente. As opções de resposta obedeceram uma escala de tipo *Likert*, de 5 pontos, onde apenas uma resposta pôde ser marcada por critério avaliativo. A escala *Likert* apresentada por (LIKERT, 1932 apud ALMEIDA JUNIOR., 2017) foi tomada como base para este teste funcional, a qual originalmente foi utilizada para medir o grau de satisfação do usuário, indo de (1) "Totalmente insatisfeito", com porcentagem equivalente a 0%, até (5) "Totalmente satisfeito", correspondente a 100%, como ilustrado na figura 5.1. Mas que para este trabalho teve sua nomenclatura redefinida para: "Péssimo"(1), "Ruim"(2), "Regular"(3), "Bom"(4) e "Excelente"(5).

Para ser considerado "Péssimo" o valor precisa estar inserido no intervalo entre 1 e 1.8, para "Ruim" deve ser maior que 1.8 e menor ou igual a 2.6, "Regular" maior que 2.6 e menor ou igual a 3.4, para ser considerado "Bom" deve ser maior que 3.4 e menor ou igual 4.2, e finalmente para ser "Excelente" o valor necessita ser maior que 4.2 e menor ou igual a 5.

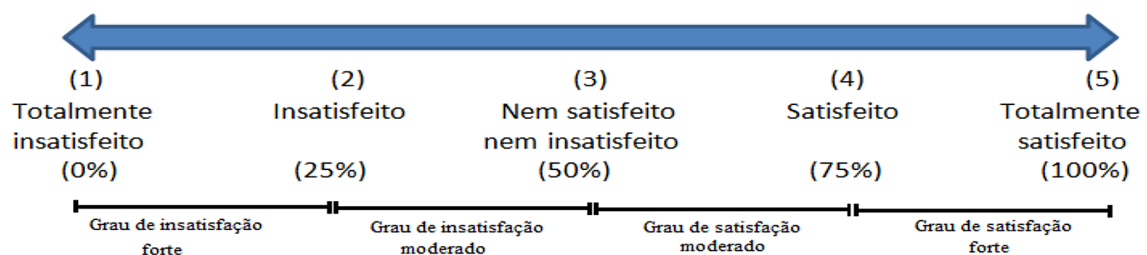


Figura 5.1: Escala do grau de satisfação. Fonte: (LIKERT, 1932 apud ALMEIDA JUNIOR., 2017).

Os testes com usuários fictícios criados na fase de preparação primeiramente ocorreram na etapa de autenticação do sistema, para que as funcionalidades executadas, descritas nas etapas 2 e 3 dos testes, pudessem ser concluídas. Essas etapas são brevemente apresentadas a seguir, onde "Maria" e "Jonas" foram os usuários escolhidos para a demonstração, porém os outros usuários também necessitaram concluir as etapas para alcançar os resultados esperados.

**Teste 1 - Objetivo final: submeter o questionário preenchido para avaliação.**

- Etapa 1: "Maria" informa CPF e senha propositalmente incorretos e tenta acessar o aplicativo. O sistema não permite a entrada da usuária e informa o erro. "Maria" então preenche os dados corretamente e dessa vez consegue entrar na área do discente. Requisito funcional "RF01: efetuar *login* no sistema" concluído.
- Etapa 2: "Maria" opta por responder questionário e seleciona a disciplina "Interação Humano-Computador" para realizar o preenchimento. Requisito Funcional "RF04: Selecionar disciplina para o preenchimento do questionário" concluído.
- Etapa 3: A usuária ao se deparar com as questões começa a marcar resposta para cada uma das perguntas, avançando uma por uma até a última questão. Decide por não utilizar a área de comentário e finaliza o questionário, obtendo confirmação do sistema. Requisitos funcionais "RF05: preencher questionário" e "RF07: enviar questionário" concluídos.

**Teste 2 - Objetivo final: gerar gráfico para consulta de desempenho em uma disciplina.**

- Etapa 1: "Jonas", docente da instituição e professor da turma "A" na disciplina "Interação Humano-Computador" no período letivo "2022/1", informa o SIAPE e uma senha incorreta e tenta acessar o aplicativo. O sistema não permite a entrada do usuário e informa o erro. "Jonas" então preenche a senha corretamente e dessa vez consegue entrar na área do docente. Requisito funcional "RF01: efetuar *login* no sistema" concluído.
- Etapa 2: "Jonas" opta por consultar desempenho e seleciona a disciplina lecionada para a geração de gráfico. Requisito funcional "RF09: selecionar disciplina para geração de gráfico de desempenho" concluído.
- Etapa 3: o usuário ao se deparar com um gráfico de barras consegue analisar o desempenho da disciplina lecionada, onde cada barra representa um *ranking* médio

para cada pergunta do questionário, avaliadas pelos cinco alunos da turma. Requisito funcional "RF10: gerar gráfico de barras com *outliers*" concluído.

Os dados coletados correspondem a todas as respostas fornecidas pelos discentes criados para a simulação, os quais possibilitaram, por meio dos questionários respondidos, que gráficos de barra pudessem ser gerados para a análise do docente fictício.

### 5.1.3 Análise e Conclusão dos Resultados

Para que seja possível a geração do gráfico com valores para cada uma das questões, necessariamente avaliações precisam ter sido realizadas. Dessa forma, como "Jonas" foi capaz de visualizar o desempenho na disciplina a qual leciona, as etapas para o preenchimento e submissão dos questionários por parte dos discentes se mostraram eficazes em atingir seus objetivos. Logo, as funções testadas para simular ações de discentes e docentes do aplicativo cumpriram com suas finalidades. Porém, para que as informações contidas nos gráficos obtidos sejam válidas é preciso analisar os cálculos realizados sobre as respostas marcadas pelos discentes e compará-las com aquelas presentes na geração. Com base nisso, a tabela 5.1 apresenta as respostas dos alunos integrantes da turma para cada uma das perguntas do questionário avaliativo.

Tabela 5.1: Respostas dos alunos matriculados na turma que responderam o questionário

Questão	Descrição	Maria	Pedro	Carolina	Bruno	Marcos
1	O professor disponibilizou o plano de curso da disciplina na primeira semana de aula?	2	3	2	1	2
2	O professor é pontual?	4	1	5	3	3
3	O professor é assíduo às aulas?	5	3	2	2	3
4	O professor usa todo o tempo de aula?	3	1	1	4	4
5	O professor está disponível no horário de atendimento?	4	4	3	3	3
6	O professor tira dúvidas em sala de aula?	1	1	3	1	2
7	O professor apresenta os conteúdos da disciplina de forma clara?	5	5	4	3	5
8	O professor demonstra dominar o conteúdo da disciplina?	1	3	2	2	4
9	O material utilizado em aula é atualizado?	5	5	4	4	4
10	O professor apresenta exemplos em aula e faz ou propõe exercícios e/ou trabalhos práticos?	1	1	1	5	1



Questão	Descrição	Maria	Pedro	Carolina	Bruno	Marcos
11	As avaliações refletem o conteúdo apresentado em sala?	2	3	5	4	2
12	As notas das avaliações foram publicadas até três dias antes da avaliação seguinte?	5	5	5	5	5
13	O conteúdo programático da disciplina foi cumprido?	4	3	4	1	2
14	Você acha que a bibliografia sugerida no Plano de Curso é adequada ao conteúdo da disciplina?	4	3	4	4	3
15	Os objetivos da disciplina foram atendidos?	1	2	2	5	2
16	Você considera a ementa da disciplina adequada ao seu curso?	4	3	1	2	1
17	Você é pontual?	5	2	4	4	4
18	Você é assíduo às aulas?	5	5	5	5	5

Questão	Descrição	Maria	Pedro	Carolina	Bruno	Marcos
19	Você procurou o professor, tutor ou monitor da disciplina para tirar dúvidas ao longo do semestre?	5	4	3	3	4
20	Você considera que possuía a base teórica necessária para cursar esta disciplina?	2	3	3	4	4
21	Você faz as atividades práticas propostas pelo professor?	5	5	3	2	5
22	Você considera que se dedicou suficientemente à disciplina?	4	4	5	5	3
23	Na sua opinião, suas notas refletem de modo fiel o conhecimento que você adquiriu ao longo da disciplina?	1	2	1	3	4

Para analisar os itens da escala Likert foi utilizado o cálculo do *Ranking* Médio (RM) proposto por (OLIVEIRA; TOSTES; MELO, 2005 apud ALMEIDA JUNIOR., 2017). A cada opção é atribuído um valor de 1 a 5 nesse modelo, em que a partir da resposta escolhida é realizado o cálculo da média ponderada para cada item, com base na frequência das respostas. O grau de satisfação dos usuários em relação aos itens correspondentes aumenta a medida que o RM se aproxima de 5 e diminui a medida que se aproxima de 1, ou seja, maior grau de insatisfação. Diante disso, o cálculo do *Ranking* Médio pode ser obtido por meio da equação:

$$RM = \frac{\sum_{i=1}^5 (f_i \cdot v_i)}{NS} \quad (5.1)$$

$f_i$  = Frequência observada de cada resposta para cada afirmação

$v_i$  = Valor de cada resposta

---

$NS$  = Número de sujeitos (usuários discentes respondentes)

Com base nas respostas da tabela 5.1 esse cálculo foi realizado para todas as questões presentes no questionário avaliativo. Porém, para exemplificar, apenas a demonstração do cálculo de RM para a primeira questão é apresentada a seguir:

$$RM = \frac{(1 \cdot 1) + (3 \cdot 2) + (1 \cdot 3) + (0 \cdot 4) + (0 \cdot 5)}{5} = 2 \quad (5.2)$$

Com o valor 2 obtido para o *ranking* médio, a primeira questão é então classificada como "Ruim", ou seja, os discentes que realizaram as avaliações se mostraram insatisfeitos quanto a disponibilidade do plano de curso.

Tabela 5.2: Cálculos realizados para as questões presentes no questionário

Questão	Freq.1	Freq.2	Freq.3	Freq.4	Freq.5	MP	RM
<b>1</b>	1	3	1	0	0	10	<b>2</b>
<b>2</b>	1	0	2	1	1	16	<b>3.2</b>
<b>3</b>	0	2	2	0	1	15	<b>3</b>
<b>4</b>	2	0	1	2	0	13	<b>2.6</b>
<b>5</b>	0	0	3	2	0	17	<b>3.4</b>
<b>6</b>	3	1	1	0	0	8	<b>1.6</b>
<b>7</b>	0	0	1	1	3	22	<b>4.4</b>
<b>8</b>	1	2	1	1	0	12	<b>2.4</b>
<b>9</b>	0	0	0	3	2	22	<b>4.4</b>
<b>10</b>	4	0	0	0	1	9	<b>1.8</b>
<b>11</b>	0	2	1	1	1	16	<b>3.2</b>
<b>12</b>	0	0	0	0	5	25	<b>5</b>
<b>13</b>	1	1	1	2	0	14	<b>2.8</b>
<b>14</b>	0	0	2	3	0	18	<b>3.6</b>
<b>15</b>	1	3	0	0	1	12	<b>2.4</b>
<b>16</b>	2	1	1	1	0	11	<b>2.2</b>
<b>17</b>	0	1	0	3	1	19	<b>3.8</b>
<b>18</b>	0	0	0	0	5	25	<b>5</b>
<b>19</b>	0	0	2	2	1	19	<b>3.8</b>
<b>20</b>	0	1	2	2	0	16	<b>3.2</b>
<b>21</b>	0	1	1	0	3	20	<b>4</b>
<b>22</b>	0	0	1	2	2	21	<b>4.2</b>
<b>23</b>	2	1	1	1	0	11	<b>2.2</b>

Após a realização dos cálculos e obtidos os valores de RM para todas as questões foi montada a tabela 5.2, contendo as frequências (Freq.) referentes a observação de cada uma das cinco opções disponíveis, além de média ponderada (MP) para cada item e *ranking* médio (RM) para cada uma das questões presentes no questionário. A seguir, nas figuras 5.2, 5.3 e 5.4 são apresentadas as telas do aplicativo SAD-UFJF com o gráfico gerado da disciplina, para que sejam realizadas comparações entre os valores obtidos de RM destacados na tabela 5.2 com aqueles mostrados pelo software.

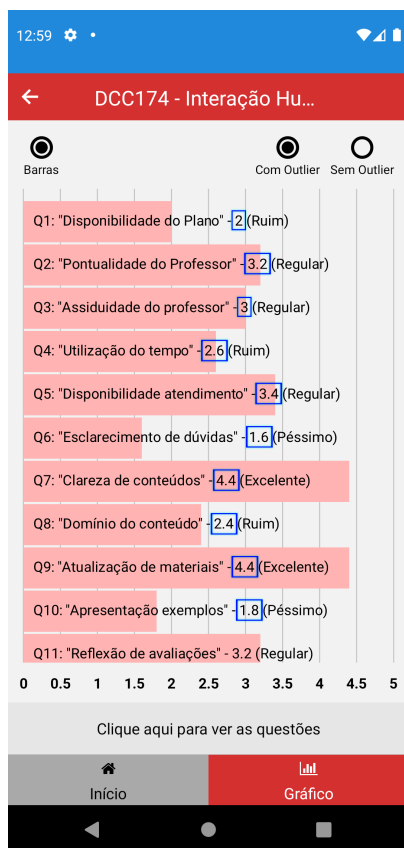


Figura 5.2: Gráfico de barras com destaque nos valores de *ranking* médio obtidos. SAD-UFJF  
Fonte: Autoria Própria.

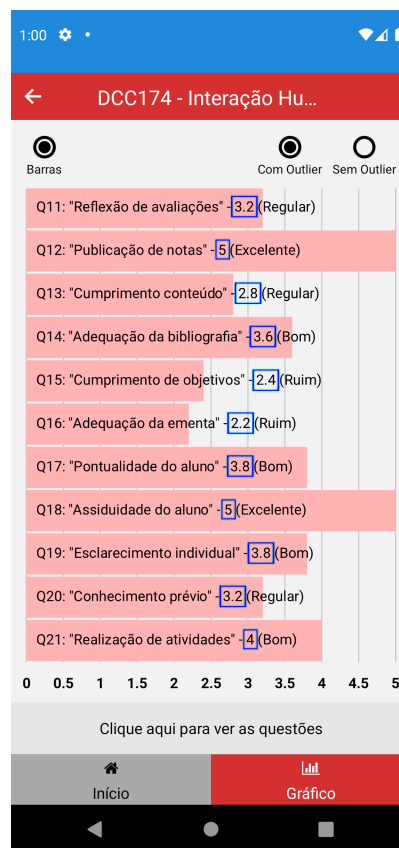


Figura 5.3: Gráfico de barras com destaque nos valores de *ranking* médio obtidos. SAD-UFJF  
Fonte: Autoria Própria.

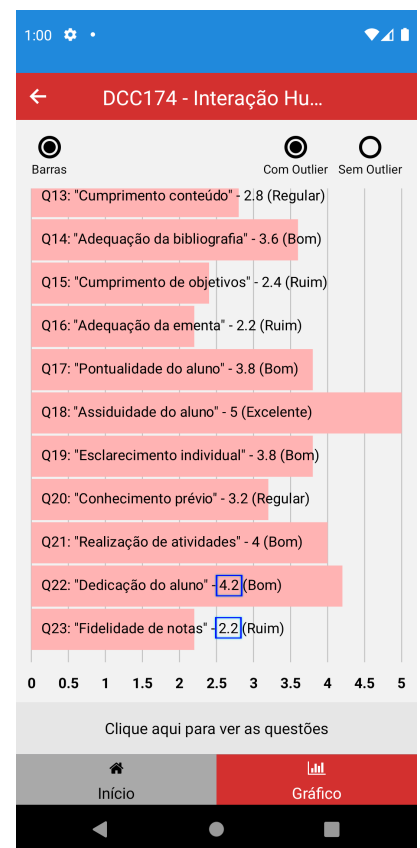


Figura 5.4: Gráfico de barras com destaque nos valores de *ranking* médio obtidos. SAD-UFJF  
Fonte: Autoria Própria.

Comparando-se os resultados calculados da tabela 5.2 com os apresentados pelo gráfico gerado pelo aplicativo é possível perceber que os valores são os mesmos. Logo, são retornados corretamente os valores esperados pela funcionalidade responsável pela exibição de um gráfico com valores correspondentes ao *ranking* médio das avaliações discentes para cada questão presente no questionário, referentes a uma disciplina avaliada.

## 5.2 Avaliação Heurística

”Avaliação heurística é um método de avaliação de IHC criado para encontrar problemas de usabilidade durante um processo de *design* iterativo” (NIELSEN; MOLICH, 1990 apud BARBOSA; SILVA, 2017). Esse método é utilizado como orientação aos avaliadores, para que sejam capazes de analisar *interfaces* a procura de problemas que possam prejudicar a usabilidade. Essa avaliação baseia-se num conjunto de diretrizes de usabilidade, definidas por Nielsen como heurísticas, as quais apresentam características desejáveis de interação e *interface* (BARBOSA; SILVA, 2017). A partir disso, algumas *interfaces* da aplicação foram analisadas, a fim de observar se estas atendem à boas práticas de *design*, por meio dos métodos de avaliação heurística de Nielsen e Princípios da *Gestalt*. As avaliações são apresentadas nas etapas de preparação, coleta de dados e consolidação dos resultados.

### 5.2.1 Preparação

O objetivo foi avaliar a usabilidade das funcionalidades referentes às tela de questionário avaliativo, menu lateral do discente, tela inicial do docente, tela de seleção de disciplinas para preenchimento de questionário e tela de geração de gráfico de barras do SAD-UFJF. Os perfis de usuário discente e docente foram considerados na avaliação. Os Princípios da *Gestalt* (subseção 2.6.1) e as Heurísticas de Design de Nielsen (NIELSEN, 1993) (subseção 2.6.3).

### 5.2.2 Coleta de Dados

Nesta fase, *interfaces* do SAD-UFJF (figuras 5.5 a 5.10) foram inspecionadas para identificar possíveis violações das heurísticas de *design* e dos Princípios da *Gestalt*. Problemas foram encontrados e identificados os locais onde ocorreram, gravidade, justificativa e recomendações de soluções. As Heurísticas de Nielsen e Princípios da *Gestalt* escolhidos para as avaliações são apresentados na tabela 5.3.

Tabela 5.3: Heurísticas de Nielsen e Princípios da *Gestalt* utilizados nas análises

<b>HN</b>	Heurísticas de Nielsen
<b>HN1</b>	Visibilidade do estado do sistema
<b>HN2</b>	Flexibilidade e eficiência de uso
<b>HN3</b>	Projeto estético e minimalista
<b>HN4</b>	Prevenção de erros
<b>HN5</b>	Consistência e padronização
<b>PG</b>	Princípios da <i>Gestalt</i>
<b>PG1</b>	Princípio da Proximidade
<b>PG2</b>	Princípio da Simetria
<b>PG3</b>	Princípio da Similaridade

Na figura 5.5 é possível perceber que a heurística "HN1" é atendida, uma vez que o sistema mantém os usuários informados sobre a tela que estão visitando (Destaques na cor amarelo). A heurística "HN3" também é atendida, já que as informações contidas na tela são todas necessárias e fazem parte do questionário.

O Princípio da *Gestalt* "PG1" pode ser notado, já que entidades visuais próximas são percebidas como grupo, como acontece nos botões (Destaque na cor verde).

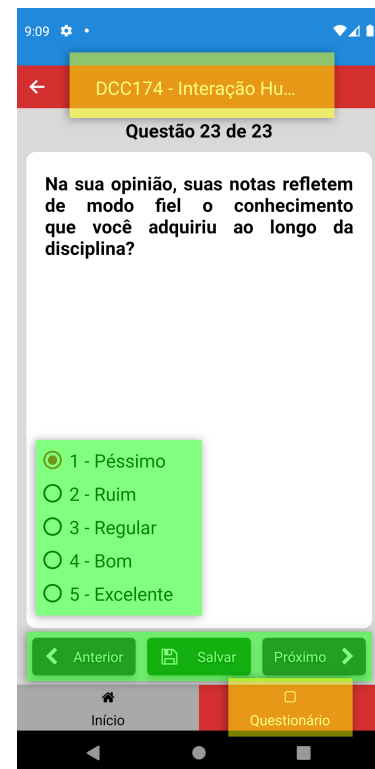


Figura 5.5: Questionário avaliativo. SAD-UFJF  
Fonte: Autoria Própria.

Na figura 5.6 a heurística "HN4" é atendida, uma vez que o sistema alerta o usuário com uma mensagem solicitando confirmação, evitando que a submissão do questionário seja realizada por engano (Destaques na cor amarelo). A heurística "HN3" também é atendida, já que as informações contidas na tela são todas necessárias e fazem parte do questionário.



Figura 5.6: Mensagem de confirmação de envio do questionário. SAD-UFJF  
Fonte: Autoria Própria.



Na figura 5.7 há o atendimento a heurística "HN2", visto que aceleradores, neste caso representados por teclas de atalho para acionar itens do menu, podem tornar mais rápida e eficiente a interação do usuário (Destaques na cor amarelo). A heurística "HN3" também é atendida, já que as informações contidas na tela são todas necessárias e fazem parte do questionário.

O Princípio da *Gestalt* "PG1" está presente na tela, já que os itens do menu podem ser percebidos como grupo (Destaque na cor amarelo). Além disso, esses itens são similares, pois apresentam o mesmo padrão de ícones e cores, atendendo também ao Princípio "PG3".

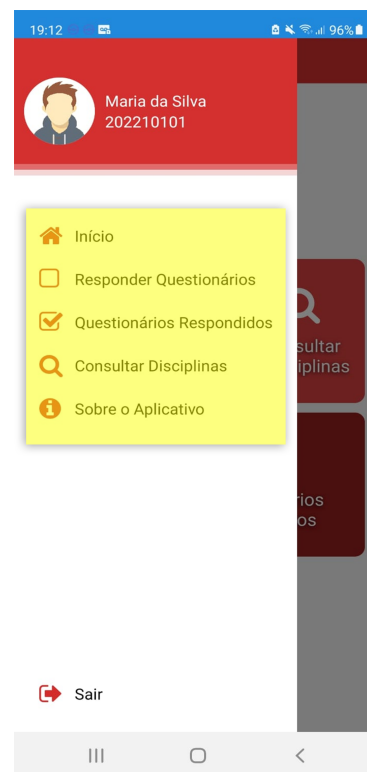


Figura 5.7: Menu lateral do discente. SAD-UFJF  
Fonte: Autoria Própria.

Na figura 5.8 há a presença da heurística "HN2", dado que teclas de atalho podem ser utilizadas na tela principal para acionar funções do aplicativo, tornando mais rápida e eficiente a interação do usuário (Destaques nas cores amarelo e verde).

O Princípio da *Gestalt* "PG1" é atendido pela figura, uma vez que os atalhos da tela podem ser percebidos como grupo (Destaque nas cores amarelo e verde). Está presente também o princípio "PG2" com atalhos relacionados a análises, simétricos entre si (Destacados na cor amarelo) e atalhos relacionados a consultas, simétricos entre si (Destacados na cor verde). Além disso, esses atalhos atendem ao Princípio "PG3", sendo similares, percebidos como grupo.



Figura 5.8: Tela inicial do docente. SAD-UFJF  
Fonte: Autoria Própria.

Na figura 5.9 é atendida a heurística "HN1", visto que informa o usuário da tela que está sendo navegada (Destaques na cor verde). A heurística "HN3" também pode ser observada, as informações contidas na tela são básicas e apenas necessárias para identificar uma disciplina (Destaques na cor amarelo).

O Princípio da *Gestalt* "PG1" é observado na tela, em razão dos botões de rádio serem percebidos como um grupo (Destaques na cor azul), bem como acontece nos campos com disciplinas (Destaques na cor amarelo). Os botões de rádio (Destaques na cor azul) e os campos com disciplinas (Destaques na cor amarelo) por serem similares entre si também configuram o Princípio "PG3". Além disso, percepções de cores diferentes são notadas nos *status* das disciplinas, para que a atenção aos estados dos questionários sejam destacadas.

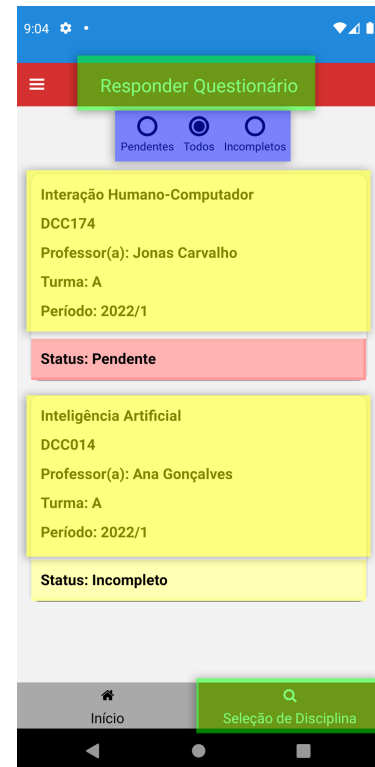


Figura 5.9: Tela de seleção de disciplinas. SAD-UFJF  
Fonte: Autoria Própria.

Na figura 5.10 o usuário consegue identificar a tela que está sendo navegada, fazendo com que a heurística "HN1" seja atendida (Destaques na cor verde). A heurística "HN3" não é bem atendida, pois embora todas as informações relacionadas as questões e suas avaliações sejam úteis no entendimento do gráfico, estas são apresentadas com excesso de conteúdo, competindo com as barras pela atenção do usuário (Destacados na cor amarelo). A heurística "HN5" também apresenta um problema, visto que a letra "Q" não diz ao usuário do que se trata. Embora se refira a palavra "Questão" e há uma descrição completa para cada pergunta em outra tela, essa letra não é exibida de forma clara. Ambos os problemas podem ser considerados como pequenos, localizados no gráfico e podendo ser solucionados com a utilização de *tooltips*, para que as informações fiquem mais organizadas.

O Princípio da *Gestalt* "PG1" é atendido pela figura, uma vez que os botões de rádio (Destacados na cor azul) podem ser percebidos como grupo, bem como similares entre si, como ocorre também com as barras do gráfico (Destacadas na cor amarelo).

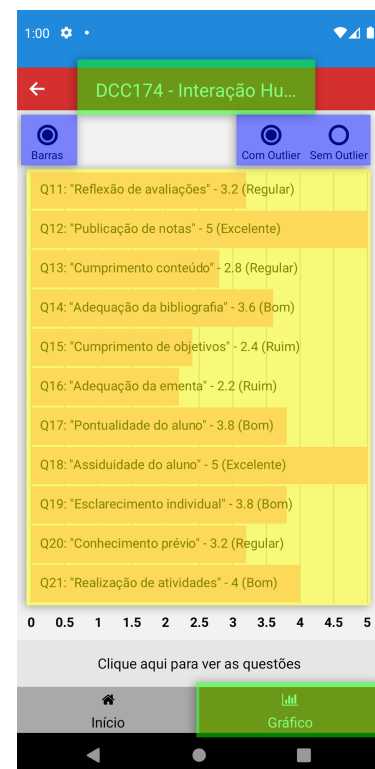


Figura 5.10: Tela de geração do gráfico de barras. SAD-UFJF  
Fonte: Autoria Própria.

### 5.2.3 Consolidação dos Resultados

Os problemas encontrados foram revisados nesta etapa, considerando-se o grau de relevância, gravidade, justificativa e recomendações de solução. Após uma visão mais abrangente, se fez necessário a elaboração de um relatório de resultados, o qual pode ser conferido na tabela 5.4.

Esse relatório apresenta informações importantes avaliadas, que se organizam da seguinte maneira:

- **Objetivos da avaliação:** apresenta a meta geral a ser alcançada na utilização do método de avaliação;
- **Escopo da avaliação:** apresenta o método de avaliação utilizado e seu objetivo;
- **Descrição do método de avaliação heurística:** descreve brevemente o método utilizado;
- **Conjunto de diretrizes utilizado:** descreve as etapas seguidas na avaliação de *interfaces*;
- **Número e perfil dos avaliadores:** quantidade de avaliadores envolvidos e suas características;
- **Lista de problemas encontrados:**
  - **local de ocorrência:** descreve onde está localizado o problema na *interface*;
  - **descrição do problema:** detalha a situação encontrada que pode dificultar na usabilidade do sistema;
  - **diretriz(es) violada(s):** apresenta Heurísticas de Nielsen e/ou Princípios da *Gestalt* que não foram seguidos corretamente no desenvolvimento de *interfaces*;
  - **severidade do problema:** define o nível de gravidade identificado;
  - **sugestões de solução:** apresenta propostas de melhorias para a correção do problema encontrado.

Tabela 5.4: Relatório com resultados da Avaliação Heurística

<b>Objetivos da avaliação</b>	Analisar se <i>interfaces</i> utilizadas na construção do aplicativo seguem boas práticas de <i>design</i>
<b>Escopo da avaliação</b>	Aplicar o método de inspeção Avaliação Heurística para identificar erros na aplicação que possam prejudicar a usabilidade
<b>Descrição do método de Avaliação Heurística</b>	Método de avaliação de IHC utilizado para encontrar problemas de usabilidade e verificar se há violações de diretrizes
<b>Diretrizes utilizadas</b>	Etapas de preparação, coleta de dados e consolidação dos resultados, com avaliações quanto a conformidade de <i>interfaces</i> da aplicação em relação a Heurísticas de Nielsen e Princípios da <i>Gestalt</i>
<b>Número e perfil dos avaliadores</b>	Um usuário responsável pelas análises das <i>interfaces</i> , sendo este o desenvolvedor da aplicação, atuando na simulação de discentes e docentes da universidade
<b>Problema encontrado</b>	P1: referente ao primeiro problema encontrado na figura 5.10
<b>Local onde ocorre</b>	Barras do gráfico gerado pela tela de análise de desempenho
<b>Descrição do problema</b>	Muitos textos presentes em cada barra do gráfico podem gerar excesso de conteúdo na visão do usuário, havendo disputa por atenção aos detalhes
<b>Diretrizes violadas</b>	Heurística de Nielsen "HN3": projeto estético e minimalista
<b>Severidade do problema</b>	2 (problema pequeno). Embora possa haver alguma confusão na visualização das barras o usuário é capaz de guiar-se através da escala presente no gráfico e por meio dos valores e classificações obtidos por cada pergunta do questionário
<b>Sugestões de solução</b>	Opções de <i>tooltips</i> poderiam ser utilizadas em cada barra presente no gráfico, para que as informações sejam apresentadas de forma mais organizada
<b>Problema encontrado</b>	P2: referente ao segundo problema encontrado na figura 5.10
<b>Local onde ocorre</b>	Barras do gráfico gerado pela tela de análise de desempenho

<b>Descrição do problema</b>	Letras "Q" presentes nas barras do gráfico não deixam claro seu significado ao usuário. Embora façam referência a questões do questionário avaliativo essas letras não descrevem seu objetivo na tela
<b>Diretrizes violadas</b>	Heurística de Nielsen "HN5": consistência e padronização
<b>Severidade do problema</b>	2 (problema pequeno). Embora as letras "Q" presentes nas barras do gráfico não deixem claro seu significado, o usuário ainda assim é capaz de identificar as questões presentes no questionário avaliativo, uma vez que há um identificador único para cada questão no intervalo de 1 a 23, além de uma breve descrição de cada assunto abordado
<b>Sugestões de solução</b>	Opções de <i>tooltips</i> poderiam ser utilizadas em cada barra presente no gráfico, para que as informações sejam apresentadas de forma mais organizada, com descrição completa das informações. Embora haja possibilidade de acessar todas as questões por meio da opção "Clique aqui para ver as questões" localizada na parte inferior da <i>interface</i> , seria mais interessante o usuário obter informações navegando na mesma tela

Com aplicação do método de Avaliação Heurística foi possível perceber que a maioria das *interfaces* analisadas seguem boas práticas de *design*, desenvolvidas em conformidade com as Heurísticas de Nielsen e Princípios da *Gestalt* avaliados. Apenas dois problemas foram identificados como exemplos na detecção de violações de usabilidades, os quais foram definidos como "P1" e "P2", com suas descrições completas na apresentação do relatório, apresentado como forma de tabela (5.4).

## 5.3 Considerações Finais

Neste capítulo foi possível perceber a importância existente na realização de avaliações de produtos nas etapas de interação com o usuário, visto que nessa fase funcionalidades podem ser testadas e problemas podem ser identificados para que sejam posteriormente corrigidos, a fim de atenderem melhor a diretrizes adotadas. O teste funcional pôde

---

apresentar a execução de duas funcionalidades presentes no aplicativo, as quais cumpriram com seus objetivos e geraram resultados esperados. Na Avaliação Heurística a comparação de *interfaces* criadas com Heurísticas de Nielsen e Princípios da *Gestalt* possibilitou a observação de conformidades com boas práticas de *design* e percepção de erros que podem prejudicar na usabilidade da aplicação. O teste de usabilidade, porém, não foi realizado, sendo uma ideia de complementação para trabalhos futuros.

## 6 Conclusão e Trabalhos Futuros

Conclui-se que ao utilizar abordagens de IHC, juntamente com técnicas de modelagem aliadas à tecnologias disponíveis para implementação de dispositivos móveis, foi possível desenvolver um *software* como protótipo para algumas atividades, entre elas simular avaliações de desempenho em uma disciplina cursada, por parte de usuários registrados como representação de discentes da Universidade Federal de Juiz de Fora (UFJF). Essas simulações avaliativas ocorreram por meio de questionários presentes no aplicativo, que possibilitaram aferir o grau de concordância ou não dos hipotéticos perfis registrados, em relação a questões referentes à conduta de docentes imaginários, bem como questões de autoavaliação. Por meio desse protótipo também foi possível simular que docentes fictícios da universidade fossem capazes de realizar aparentes análises de desempenho em disciplinas lecionadas, através de geração de gráficos e relatórios que refletem opiniões de supostos discentes respondentes para cada turma. Com aplicação de Análise de IHC foi possível realizar avaliações no aplicativo em que, por meio de teste funcional, pôde-se verificar a execução de funcionalidades e a correspondência dessas com os resultados esperados. Assim como no método de Avaliação Heurística, o qual foi capaz de possibilitar que *interfaces* desenvolvidas para a aplicação fossem verificadas quanto ao atendimento, ou não, de boas práticas adotadas em suas elaborações, bem como a identificação de problemas de interação que prejudicam a usabilidade do sistema.

O aplicativo SAD-UFJF visa apoiar os processos avaliativos discentes da UFJF, por meio de um sistema que surge como uma nova possibilidade aos usuários. Sua *interface* foi desenvolvida para a utilização em *smartphones*, uma vez que estes dispositivos permitem maior agilidade e disponibilidade, por serem móveis e fáceis de serem transportados. Porém, ausência de avaliações mais robustas do aplicativo surgem como limitações do trabalho, necessitando ainda serem executadas, a fim de garantir que todas as funcionalidades atendam perfeitamente aos objetivos propostos. Há carência de testes de usabilidade com usuários reais, alvos do aplicativo (discentes e docentes), para que possa ser aferido, de fato, suas ações, comportamentos, satisfação e grau de eficácia ao execu-

tarem cada funcionalidade do sistema.

Como trabalhos futuros tem-se a ideia de expansão do aplicativo. Após a adoção de *feedbacks* seria possível realizar alterações para que o sistema se adapte aos gostos e necessidades dos usuários, assim como inclusão de novas funcionalidades que permitam agregar mais possibilidades a aplicação. Tem-se também como plano futuro a integração do aplicativo com o sistema SIGA da UFJF, onde seria possível obter acesso à dados necessários da universidade, sem exposição dos mesmos, sendo utilizados apenas para funções que necessitem identificar quem utiliza o sistema, em funções como autenticação do usuário e consultas à disciplinas. Esses dados poderiam ser consumidos diretamente da instituição, para que cada usuário cadastrado no SIGA pudesse, de fato, ter sua autenticação validada. A assimilação do SAD-UFJF ao aplicativo da instituição também surge como uma possibilidade de expansão, para que todos os recursos disponíveis aos usuários da universidade estejam presentes em um único sistema.



## Bibliografia

Aela Editorial. *Os 7 Princípios de Gestalt e Como Utilizá-los em Projetos de UI Design*. Disponível em: <https://medium.com/aela/os-7-principios-de-gestalt-e-como-utiliza-los-em-projetos-de-ui-design-46d6d832abf6>, year="2020".

ALMEIDA JUNIOR., R. C. V. d. Estudo sobre o grau de satisfação dos usuários do serviço de transporte coletivo prestado pela empresa Viação cidade Corumbá ao município de Corumbá/MS. *Revista GeoPantanal*, UFMS/AGB, n. 23, p. 215–230, 2017. Disponível em: <https://periodicos.ufms.br/index.php/revgeo/article/view/4794>.

Alura. *MongoDB: O banco baseado em documentos*. 2021. Disponível em: <https://www.alura.com.br/artigos/mongodb-o-banco-baseado-em-documentos>. Acesso em: 19 janeiro 2022.

BARBOSA, S.; SILVA, B. *Interação Humano-Computador*. [S.l.]: Elsevier Brasil, 2017.

CARD, S.; MORAN, T. P.; NEWELL, A. *The Psychology of Human-Computer Interaction*. [S.l.]: Hillsdale, NJ: Lawrence Erlbaum Associates, 1983.

CHEN, P. Entity-relationship modeling: Historical events, future trends, and lessons learned. In: \_\_\_\_\_. *Software Pioneers: Contributions to Software Engineering*. Berlin, Heidelberg: Springer-Verlag, 2002. p. 296–310. ISBN 3540430814.

CUER, E. d. S. *Comparação de desempenho de bancos de dados sql e nosql*. 2015. Disponível em: <https://aberto.univem.edu.br/handle/11077/999>. Acesso em: 19 janeiro 2022.

de SOUZA, C. *The Semiotic Engineering of Human-Computer Interaction*. [S.l.]: Cambridge, MA: The MIT Press, 2005a.

Express. *Express: Framework web rápido, flexível e minimalista para Node.js*. 2022. Disponível em: <https://expressjs.com/pt-br/>. Acesso em: 08 fevereiro 2022.

FIELDING, R. T.; TAYLOR, R. N. *Architectural Styles and the Design of Network-Based Software Architectures*. [S.l.]: University of California, Irvine, 2000. AAI9980887.

GIBSON, J. Perceiving, acting, and knowing: toward an ecological psychology. In: \_\_\_\_\_. [S.l.]: New Jersey: Lawrence Erlbaum Associates, 1977. cap. The theory of affordance, p. 67–82.

GUEDES, G. T. *UML2 Uma Abordagem Prática.3.ed.* [S.l.]: São Paulo: Novatec Editora Ltda, 2018.

Heroku. *The Heroku Platform*. 2022. Disponível em: <https://www.heroku.com/platform>. Acesso em: 05 janeiro 2022.

HEWETT, T. T. et al. *ACM SIGCHI Curricula for Human-Computer Interaction*. [S.l.]: ACM SIGCHI Report, ACM, NY, 1992. Disponível em: <https://dl.acm.org/doi/book/10.1145/2594128>. Acesso em: 19 janeiro 2022.

- HOFSTETTER, J. *O que é Gestalt? Saiba tudo sobre as lei da Gestalt*. 2019. Disponível em: <https://4ed.cc/gestalt/>. Acesso em: 02 setembro 2021.
- JSON. *Introdução ao JSON*. 2022. Disponível em: <http://www.json.org/json-pt.html>. Acesso em: 10 fevereiro 2022.
- KAMMERSGAARD, J. Four different perspectives on human–computer interaction. *International Journal of Man-Machine Studies*, Elsevier, v. 28, n. 4, p. 343–362, 1988.
- Kinsta. *O Que é Node.js e Por Que Usá-lo?* 2021. Disponível em: <https://kinsta.com/pt/base-de-conhecimento/o-que-e-node-js/>. Acesso em: 19 janeiro 2022.
- Laputan. *Model-View-Controller*. 1999. Disponível em: <http://www.laputan.org/pub/papers/POSA-MVC.pdf>. Acesso em: 30 julho 2021.
- Lenon. *Node.js – O que é, como funciona e quais as vantagens*. 2018. Disponível em: <https://www.opus-software.com.br/node-js/>. Acesso em: 02 julho 2021.
- LIKERT, R. *A Technique for the Measurement of Attitudes*. [S.l.]: Archives of Psychology, 1932. (A Technique for the Measurement of Attitudes, Nº 136-165).
- LUCIANO, J.; ALVES, W. J. B. Padrão de arquitetura mvc: Model-view-controller. *EPeQ Fafibe*, v. 1, n. 3a, p. 102–107, 2017.
- MATTOS, D. de; KERNITSKEI, J. E.; JUNIOR, O. de P. Análise comparativa de frameworks para desenvolvimento front-end em aplicações web.
- Medium. *React-Native - Arquitetura e comparações*. 2019. Disponível em: <https://medium.com/datainfo>. Acesso em: 19 janeiro 2022.
- MongoDB. *JSON and BSON*. 2022. Disponível em: <https://www.mongodb.com/json-and-bson>. Acesso em: 11 fevereiro 2022.
- MongoDB. *O que é MongoDB?* 2022. Disponível em: <https://www.mongodb.com/pt-br/what-is-mongodb>. Acesso em: 05 de janeiro 2022.
- Mongoose. *Mongoose*. 2022. Disponível em: <https://mongoosejs.com/docs/api/schema.html>. Acesso em: 05 janeiro 2022.
- MORAN, T. *The Command Language Grammars: a representation for the user interface of interactive computer systems*. [S.l.]: International Journal of Man-Machine Studies 15, pp. 3-50, 1981, 1981.
- NIELSEN, J. *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 0125184050.
- NIELSEN, J.; BUDI, R. *Usabilidade Móvel*. [S.l.]: GEN LTC, 2013.
- NIELSEN, J.; LORANGER, H. *Usabilidade na WEB - Projetando Websites com qualidade*. [S.l.]: Elsevier, Campus, 2007.
- NIELSEN, J.; MOLICH, R. *Heuristic evaluation of user interfaces*. [S.l.]: Proceedings of ACM CHI'90, pp. 249-256, 1990.
- NodeJS. *About Node.js®*. 2021. Disponível em: <https://nodejs.org/en/about/>. Acesso em: 02 julho 2021.

- NORMAN, D. A. *Cognitive Engineering*. [S.l.]: In: D.A. Norman e S.W. Draper (eds.), User-Centered System Design. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 31-61, 1986.
- NORMAN, D. A. *Cognitive Artifacts*. [S.l.]: J.M.Carroll (ed.), Designing Interaction: Psychology at the Human-Computer Interface. Cambridge, MA: Cambridge University Press, pp. 17-38, 1991, 1991.
- OLIVEIRA, L. H. d.; TOSTES, A. A. F.; MELO, T. V. C. F. Indicadores de desempenho como estratégia operacional: estudo de caso em uma indústria de café. In: *15<sup>o</sup> IFMA - Congresso Internacional de Administração Rural. Simultâneo ao 5<sup>o</sup> Congresso Brasileiro de Administração Rural, 2005*. [S.l.: s.n.], 2005. v. 15.
- React Native. *React Native*. 2022. Disponível em: <https://reactnative.dev/>. Acesso em: 08 janeiro 2022.
- Rocketseat. *Blog da Rocketseat*. Disponível em: <https://blog.rocketseat.com.br/>. Acesso em: 19 janeiro 2022.
- ROOM, C. Express. js.
- ROSSON, M.; CARROLL, J. *Scenario-Based Development of Human-Computer Interaction*. [S.l.]: San Francisco, CA: Morgan Kaufmann Publishers, 2002.
- SHNEIDERMAN, B. et al. *Designing the user interface: strategies for effective human-computer interaction*. [S.l.]: Pearson, 2016.
- SILVA, R. S. et al. O uso de smartphone e seu impacto no cotidiano da vida dos acadêmicos. *Revista Cereus*, v. 12, n. 3, p. 53-67, 2020.
- TreinaWeb. *O que é MVC?* 2021. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-mvc/>. Acesso em: 19 janeiro 2022.
- UFJF. *UFJF lança aplicativo para facilitar acesso aos serviços da instituição*. 2018. Disponível em: <https://www2.ufjf.br/noticias/2018/04/19/ufjf-lanca-aplicativo-para-facilitar-acesso-aos-servicos-da-instituicao/>. Acesso em: 06 julho 2021.
- VEBER, L. S. et al. Dependência de smartphone em estudantes universitários. *ANAIS CONGREGA MIC-ISBN: 978-65-86471-05-2 e ANAIS MIC JR.-ISBN: 978-65-86471-06-9*, v. 16, p. 23-28, 2020.
- WARE, C. *Design as Applied Perception*. In: J.M. Carrol (ed), *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*. [S.l.]: San Francisco, CA: Morgan Kaufmann, pp. 11-26, 2003.
- WEIAND, H. *React-Native - Arquitetura e Comparações*. 2019. Disponível em: <https://medium.com/datainfo/react-native-arquitetura-e-compara%C3%A7%C3%B5es-1663f77e0a57>. Acesso em: 15 novembro 2021.

## A Apêndice

### A.1 Requisitos funcionais do Aplicativo SAD-UFJF

Tabela A.1: Requisitos funcionais do sistema

Identificador	Nome
RF01	Efetuar <i>login</i> no sistema.
RF02	Consultar informações de autenticação.
RF03	Consultar disciplinas.
RF04	Selecionar disciplina para o preenchimento do questionário.
RF05	Preencher questionário.
RF06	Salvar progresso.
RF07	Enviar questionário.
RF08	Consultar questionários concluídos.
RF09	Selecionar disciplina para geração de gráfico de desempenho.
RF10	Gerar gráfico de barras com outliers.
RF11	Gerar gráfico de barras sem outliers.
RF12	Exibir questões avaliadas no gráfico.
RF13	Selecionar disciplina para geração de relatório de desempenho.
RF14	Gerar relatório com outlier.
RF15	Gerar relatório sem outlier.
RF16	Consultar informações sobre o sistema.
RF17	Sair do sistema.

## B Apêndice

### B.1 Especificação dos Requisitos Funcionais do Aplicativo SAD-UFJF

Tabela B.1: Especificação do requisito funcional 'RF01'

Identificador	RF01
Nome	Efetuar login no sistema.
Sumário	O caso de uso é responsável por autenticar a entrada do usuário no sistema.
Atores	(1) Discente, (2) Docente.
Pré-condições	O usuário (1)(2) deve possuir cadastro no SIGA.
Descrição	<p><b>Fluxo principal - usuário (1):</b></p> <ol style="list-style-type: none"> <li>1: O usuário informa o CPF.</li> <li>2: O usuário informa a senha.</li> <li>3: O usuário faz clique sobre o botão "Entrar".</li> <li>4: O sistema analisa as informações fornecidas.</li> <li>5: O sistema exibe a tela principal do aplicativo. Botões "Responder Questionário", "Consultar Disciplinas", "Sobre o Aplicativo" e "Questionários Respondidos" são localizados ao centro, como atalhos para as principais funções do sistema. Um menu lateral à esquerda também fornece as opções de atalho, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada.</li> </ol> <p><b>Fluxo principal - usuário (2):</b></p> <ol style="list-style-type: none"> <li>1: O usuário informa o Siape.</li> <li>2: O usuário informa a senha.</li> <li>3: O usuário faz clique sobre o botão "Entrar".</li> <li>4: O sistema analisa as informações fornecidas.</li> <li>5: O sistema exibe a tela principal do aplicativo. Botões "Gerar Relatório", "Consultar Disciplinas", "Sobre o Aplicativo" e "Consultar Desempenho" são localizados ao centro, como atalhos para as principais funções do sistema. Um menu lateral à esquerda também fornece as opções de atalho, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada.</li> </ol>

RF01	Efetuar login no sistema
Alternativas	<p><b>Alternativa ao passo 4 - usuário (1):</b> 1: O sistema verifica que o CPF e/ou senha estão incorretos, permanecendo na mesma tela de autenticação. Há a exibição de uma caixa de diálogo informando que os dados fornecidos são inválidos, com um campo "OK" para que a mensagem seja fechada. 2: O usuário faz clique sobre o campo "Ok".</p> <p><b>Alternativa ao passo 4 - usuário (2):</b> 1: O sistema verifica que o Siape e/ou senha estão incorretos, permanecendo na mesma tela de autenticação. Há a exibição de uma caixa de diálogo informando que os dados fornecidos são inválidos, com um campo "OK" para que a mensagem seja fechada. 2: O usuário faz clique sobre o campo "Ok".</p>
Pós-condições	<p><b>Pós-condição - usuário (1):</b> A tela correspondente a área do discente é exibida pelo sistema.</p> <p><b>Pós-condição - usuário (2):</b> A tela correspondente a área do docente é exibida pelo sistema.</p>
Exceções	A autenticação no sistema não poderá ser concluída caso o usuário deixe de preencher algum campo.

Tabela B.2: Especificação do requisito funcional 'RF02'

Identificador	RF02
Nome	Consultar informações de autenticação.
Sumário	O caso de uso é responsável por exibir informações sobre como ter acesso ao sistema.
Atores	(1) Discente, (2) Docente.
Pré-condições	
Descrição	<b>Fluxo principal - usuário (1)(2):</b> 1: O usuário faz clique sobre o ícone "?" na tela de autenticação. 2: O sistema exibe uma caixa de diálogo com informações necessárias para a realização da autenticação, com um campo "OK", para que a mensagem seja fechada. 3: O usuário faz clique sobre o campo "OK".
Alternativas	
Pós-condições	<b>Pós-condição - usuário (1)(2):</b> O sistema permanece na mesma tela.
Exceções	

Tabela B.3: Especificação do requisito funcional 'RF03'

Identificador	RF03
Nome	Consultar disciplinas.
Sumário	O caso de uso é responsável por exibir as disciplinas.
Atores	(1) Discente, (2) Docente.
Pré-condições	O usuário (1)(2) deve estar logado no sistema.
Descrição	<p><b>Fluxo principal - usuário (1):</b>  1: O usuário faz clique sobre a opção "Consultar Disciplinas".  2: O sistema exibe uma tela com listagem de todas as disciplinas em que o usuário está matriculado. Cada disciplina apresenta algumas informações básicas, como: nome, código, turma, e docente que a leciona. Um menu lateral à esquerda fornece ao usuário opções de navegação pelos requisitos do sistema, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada.</p> <p><b>Fluxo principal - usuário (2):</b>  1: O usuário faz clique sobre a opção "Consultar Disciplinas".  2: O sistema exibe uma tela com listagem de todas as disciplinas lecionadas pelo usuário. Cada disciplina apresenta algumas informações básicas, como: nome, código, turma, e docente que a leciona. Um menu lateral à esquerda fornece ao usuário opções de navegação pelos requisitos do sistema, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada.</p>
Alternativas	<p><b>Alternativa ao passo 2 do fluxo principal - usuário (1):</b>  1: O usuário não possui matrícula ativa em nenhuma disciplina do período letivo, o sistema exibe uma tela com mensagem informando que não há disciplinas encontradas.</p> <p><b>Alternativa ao passo 2 do fluxo principal - usuário (2):</b>  1: O usuário não leciona em nenhuma disciplina do período letivo, o sistema exibe uma tela com mensagem informando que não há disciplinas encontradas.</p>
Pós-condições	<p><b>Pós-condição - usuário (1)(2):</b>  O sistema permanece na mesma tela.</p>
Exceções	



Tabela B.4: Especificação do requisito funcional 'RF04'

Identificador	RF04
Nome	Selecionar disciplina para o preenchimento do questionário.
Sumário	O caso de uso é responsável por exibir as disciplinas que possuem questionários disponíveis para preenchimento.
Atores	Discente.
Pré-condições	O usuário deve estar logado no sistema.
Descrição	<p><b>Fluxo principal:</b></p> <p>1: O usuário faz clique sobre a opção "Responder Questionários".</p> <p>2: O sistema exibe uma tela listando todas as disciplinas que possuem questionários disponíveis no período letivo, com informações básicas, como: nome, código, turma, docente, período e status. Um menu lateral à esquerda fornece ao usuário opções de navegação pelos requisitos do sistema, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada. Há botões de rádio que permitem ao usuário filtrar o tipo de questionário desejado, cujas opções são:</p> <ul style="list-style-type: none"> <li>• "Todos": exibe todos os questionários abertos para preenchimento no período. Inicia-se com este botão selecionado por padrão;</li> <li>• "Pendentes": exibe os questionários cujos preenchimentos ainda não foram inicializados;</li> <li>• "Incompletos": exibe os questionários cujos preenchimentos foram inicializados, porém não finalizados.</li> </ul> <p>3: O usuário faz clique sobre a disciplina desejada.</p>
Alternativas	<p><b>Alternativa ao passo 2 do fluxo principal:</b></p> <p>1: O usuário não possui matrícula ativa em nenhuma disciplina do período letivo ou não existem questionários abertos para preenchimento. É exibida uma tela com mensagem informando que não há questionários encontrados.</p>
Pós-condições	<p><b>Pós-condição:</b></p> <p>A disciplina é selecionada e o sistema redireciona para o preenchimento do questionário.</p>
Exceções	

Tabela B.5: Especificação do requisito funcional 'RF05'

Identificador	RF05
Nome	Preencher Questionário.
Sumário	O caso de uso é responsável por exibir questões de múltipla escolha, bem como opção de comentário, para que o usuário seja capaz de avaliar uma disciplina cursada.
Atores	Discente.
Pré-condições	Concluir o requisito funcional "RF04".
Descrição	<b>Fluxo principal:</b> 1: O sistema exibe questões relacionadas à disciplina selecionada, assim como questões de auto desempenho do usuário. Essas questões apresentam opções de múltipla escolha, que variam de 1 (Péssimo) à 5 (Excelente), sendo permitido ao usuário marcar apenas uma como resposta. É possível navegar entre as questões por meio dos botões "Anterior" e "Próximo", tal como salvar o progresso atual com o botão "Salvar", ou submeter o questionário para avaliação, recorrendo ao botão "Enviar". Após a última questão é exibido um campo para que o usuário possa deixar alguma observação, se este julgar necessário. Um menu lateral à esquerda fornece ao usuário opções de navegação pelos requisitos do sistema, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada.
Alternativas	
Pós-condições	<b>Pós-condição:</b> O sistema permanece na mesma tela.
Exceções	

Tabela B.6: Especificação do requisito funcional 'RF06'

Identificador	RF06
Nome	Salvar progresso.
Sumário	O caso de uso é responsável por salvar o progresso no questionário.
Atores	Discente.
Pré-condições	Concluir o requisito funcional RF05.
Descrição	<b>Fluxo principal:</b> 1: O usuário faz clique sobre o botão "Salvar". 2: O sistema salva o progresso e exibe uma caixa de diálogo informando que o questionário foi salvo, com um campo "OK", para que a mensagem seja fechada. 3: O usuário faz clique sobre o campo "OK". 4: O sistema redireciona para a tela anterior ao preenchimento.
Alternativas	
Pós-condições	<b>Pós-condição:</b> Se o status do questionário salvo era "Pendente", é realizada alteração para "Incompleto". Se "Incompleto", este permanece com o status inalterado. O sistema redireciona para a tela anterior ao preenchimento do questionário.
Exceções	

Tabela B.7: Especificação do requisito funcional 'RF07'

Identificador	RF07
Nome	Enviar questionário.
Sumário	O caso de uso é responsável por submeter o questionário para avaliação.
Atores	Discente.
Pré-condições	Concluir o requisito funcional RF05.
Descrição	<p><b>Fluxo principal:</b></p> <ol style="list-style-type: none"> <li>1: O usuário faz clique sobre o botão "Enviar".</li> <li>2: O sistema analisa as opções preenchidas.</li> <li>3: O sistema exibe uma caixa de diálogo solicitando uma confirmação, com a opção "Sim", caso o usuário deseje realizar o envio do questionário para avaliação ou "Não", caso o mesmo opte por cancelar a ação.</li> <li>4: O usuário faz clique sobre a opção "Sim".</li> <li>5: O sistema exibe uma caixa de diálogo informando que o envio foi concluído, com um campo "OK", para que a mensagem seja fechada.</li> <li>6: O usuário faz clique sobre o campo "OK".</li> <li>7: O sistema redireciona para a tela anterior ao preenchimento.</li> </ol>
Alternativas	<p><b>Alternativa ao passo 3 do fluxo principal:</b></p> <ol style="list-style-type: none"> <li>1: O sistema verifica que nem todas as questões possuem respostas. Há um redirecionamento para aquela que necessita ser preenchida, com exibição de uma caixa de diálogo informando ao usuário que todas as questões precisam ser respondidas, com um campo "OK", para que a mensagem seja fechada.</li> <li>2: O usuário faz clique sobre o campo "OK".</li> <li>3: O usuário responde a questão destacada.</li> <li>4: O processo retorna ao passo 1 do fluxo principal.</li> </ol> <p><b>Alternativa ao passo 4 do fluxo principal:</b></p> <ol style="list-style-type: none"> <li>1: O usuário faz clique sobre a opção "Não".</li> <li>2: O sistema não realiza o envio e permanece exibindo a mesma tela.</li> </ol>
Pós-condições	<p><b>Pós-condição:</b></p> <p>O status do questionário enviado é alterado, de "Pendente" ou "Incompleto" para "Concluído". Após alterado o status, o questionário é retirado da tela de seleção de disciplinas disponíveis para o preenchimento e enviado para a tela de questionários respondidos, não sendo mais permitidas alterações nas respostas. O sistema redireciona para a tela anterior ao preenchimento do questionário.</p>
Exceções	O envio do questionário não poderá ser concluído caso o usuário deixe de preencher todas questões.

Tabela B.8: Especificação do requisito funcional 'RF08'

Identificador	RF08
Nome	Consultar questionários concluídos.
Sumário	O caso de uso é responsável por exibir os questionários finalizados, entregues para avaliação.
Atores	Discente.
Pré-condições	O usuário deve estar logado no sistema.
Descrição	<p><b>Fluxo principal:</b></p> <p>1: O usuário faz clique sobre a opção "Questionários Respondidos".</p> <p>2: O sistema exibe uma tela com listagem de todas as disciplinas que possuem questionários concluídos. Há botões de rádio que possibilitam a visualização de todos os questionários ou somente aqueles correspondentes a um período específico, informado pelo usuário. Cada questionário apresenta informações básicas, como: nome, código, turma, docente, período e status. Um menu lateral à esquerda fornece ao usuário opções de navegação pelos requisitos do sistema, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada.</p>
Alternativas	<p><b>Alternativa ao passo 2 do fluxo principal:</b></p> <p>1: Não há questionários concluídos ou período informado pelo usuário não é identificado. É exibida uma tela com mensagem informando que não há disciplinas encontradas.</p>
Pós-condições	<p><b>Pós-condição:</b></p> <p>O sistema permanece na mesma tela.</p>
Exceções	A consulta dos questionários por período não poderá ser concluída caso o usuário deixe de preencher o campo.

Tabela B.9: Especificação do requisito funcional 'RF09'

Identificador	RF09
Nome	Selecionar disciplina para geração de gráfico de desempenho.
Sumário	O caso de uso é responsável por exibir as disciplinas que possuem gráficos a serem gerados.
Atores	Docente.
Pré-condições	O usuário deve estar logado no sistema.
Descrição	<b>Fluxo principal:</b> 1: O usuário faz clique sobre a opção "Consultar Desempenho". 2: O sistema exibe uma tela com listagem de todas as disciplinas disponíveis no período letivo para geração de gráficos. Cada disciplina apresenta algumas informações básicas, como: nome, código, turma, e docente que a leciona. Um menu lateral à esquerda fornece ao usuário opções de navegação pelos requisitos do sistema, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada. 3: O usuário seleciona uma disciplina para geração de gráfico de desempenho.
Alternativas	<b>Alternativa ao passo 2 do fluxo principal:</b> 1: O usuário não leciona nenhuma disciplina no período letivo ou não existem disciplinas disponíveis para consulta de desempenho. É exibida uma tela com mensagem informando que não há disciplinas encontradas.
Pós-condições	<b>Pós-condição:</b> A disciplina é selecionada e o sistema realiza o redirecionamento para a tela de geração de gráfico.
Exceções	

Tabela B.10: Especificação do requisito funcional 'RF10'

Identificador	RF10
Nome	Gerar gráfico de barras com outliers.
Sumário	O caso de uso é responsável por gerar um gráfico de barras para uma disciplina, considerando-se os valores de outlier.
Atores	Docente.
Pré-condições	Concluir o requisito funcional "RF09".
Descrição	<p><b>Fluxo principal:</b></p> <p>1: O sistema exibe uma tela contendo um gráfico de barras. As barras correspondem a um ranking médio de cada uma das questões presentes nos questionários avaliados pelos discentes da disciplina. Cada questão é identificada pela letra Q (Questão) mais um identificador único, seguido de uma breve descrição do assunto abordado. O gráfico possui escala de 0 (zero) à 5 (cinco), onde 0 representa o menor aproveitamento e 5 o maior. É possível ter acesso a descrição completa de cada questão ao selecionar o campo "Clique aqui para ver as questões". Botões de rádio são localizados na parte superior da tela, os quais indicam o tipo do gráfico ou a possibilidade de incluir ou não outliers na análise, sendo a opção "Com outlier" marcada inicialmente por padrão. Um menu lateral à esquerda fornece ao usuário opções de navegação pelos requisitos do sistema, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada.</p>
Alternativas	
Pós-condições	<p><b>Pós-condição:</b></p> <p>O sistema permanece na mesma tela.</p>
Exceções	

Tabela B.11: Especificação do requisito funcional 'RF11'

Identificador	RF11
Nome	Gerar gráfico de barras sem outliers.
Sumário	O caso de uso é responsável por gerar um gráfico de barras para uma disciplina, sem considerar os valores de outlier.
Atores	Docente.
Pré-condições	Concluir o requisito funcional "RF09".
Descrição	<p><b>Fluxo principal:</b></p> <p>1: O sistema exibe uma tela contendo um gráfico de barras. As barras correspondem a um ranking médio de cada uma das questões presentes nos questionários avaliados pelos discentes da disciplina. Cada questão é identificada pela letra Q (Questão) mais um identificador único, seguido de uma breve descrição do assunto abordado. O gráfico possui escala de 0 (zero) à 5 (cinco), onde 0 representa o menor aproveitamento e 5 o maior. É possível ter acesso a descrição completa de cada questão ao selecionar o campo "Clique aqui para ver as questões". Botões de rádio são localizados na parte superior da tela, os quais indicam o tipo do gráfico ou a possibilidade de incluir ou não outliers na análise, sendo a opção "Com outlier" marcada inicialmente por padrão. Um menu lateral à esquerda fornece ao usuário opções de navegação pelos requisitos do sistema, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada.</p> <p>2: O usuário seleciona o botão "Sem outlier".</p> <p>3: O sistema exibe o gráfico de barras ignorando os valores de outlier para cada uma das questões avaliadas.</p>
Alternativas	
Pós-condições	<p><b>Pós-condição:</b></p> <p>O sistema permanece na mesma tela.</p>
Exceções	



Tabela B.12: Especificação do requisito funcional 'RF12'

Identificador	RF12
Nome	Exibir questões avaliadas no gráfico.
Sumário	O caso de uso é responsável por exibir a descrição completa de todas questões avaliadas na geração do gráfico.
Atores	Docente.
Pré-condições	Concluir o requisito funcional "RF10".
Descrição	<b>Fluxo principal:</b> 1: O usuário seleciona o campo "Clique aqui para ver as questões", presente na tela de geração de gráfico. 2: O sistema exibe uma tela contendo todas as questões utilizadas no gráfico, listadas com a letra Q (Questão) mais um identificador único, seguido da descrição completa de cada assunto. Há um campo na parte inferior que permite fechar a tela ao realizar o clique sobre "Fechar". 3: O usuário faz clique sobre o campo "Fechar".
Alternativas	
Pós-condições	<b>Pós-condição:</b> O sistema redireciona para a tela anterior.
Exceções	

Tabela B.13: Especificação do requisito funcional 'RF13'

Identificador	RF13
Nome	Selecionar disciplina para geração de relatório de desempenho.
Sumário	O caso de uso é responsável por exibir as disciplinas que possuem relatórios a serem gerados.
Atores	Docente.
Pré-condições	O usuário deve estar logado no sistema.
Descrição	<b>Fluxo principal:</b> 1: O usuário faz clique sobre a opção "Gerar Relatório". 2: O sistema exibe uma tela com listagem de todas as disciplinas disponíveis no período letivo para geração de relatórios. Cada disciplina apresenta algumas informações básicas, como: nome, código, turma, e docente que a leciona. Um menu lateral à esquerda fornece ao usuário opções de navegação pelos requisitos do sistema, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada. 3: O usuário seleciona uma disciplina para geração de relatório de desempenho.
Alternativas	<b>Alternativa ao passo 2 do fluxo principal:</b> 1: O usuário não leciona nenhuma disciplina no período letivo ou não existem disciplinas disponíveis para consulta de desempenho. É exibida uma tela com mensagem informando que não há disciplinas encontradas.
Pós-condições	<b>Pós-condição:</b> A disciplina é selecionada e o sistema realiza o redirecionamento para a tela de geração de relatório.
Exceções	

Tabela B.14: Especificação do requisito funcional 'RF14'

Identificador	RF14
Nome	Gerar relatório com outlier.
Sumário	O caso de uso é responsável por gerar um relatório para uma disciplina, considerando-se os valores de outlier.
Atores	Docente.
Pré-condições	Concluir o requisito funcional "RF13".
Descrição	<p><b>Fluxo principal:</b></p> <p>1: O sistema exibe uma tela contendo um relatório com informações relacionadas a disciplina selecionada, como nome, código, período e docente que a leciona. Além disso, uma tabela é apresentada com informações de aproveitamento da disciplina, com base nas avaliações realizadas por discentes em questionários. Essas informações são organizadas da seguinte maneira:</p> <ul style="list-style-type: none"> <li>• Questão: corresponde a um identificador único para cada questão avaliada;</li> <li>• Descrição: fornece a descrição completa de cada questão avaliada;</li> <li>• RM: corresponde a um ranking médio para cada questão avaliada por discentes da disciplina;</li> <li>• Situação: classificação do aproveitamento da disciplina com base no ranking médio obtido. A classificação varia entre cinco possibilidades: péssimo, ruim, regular, bom e excelente, obedecendo a escala de 1 à 5 respectivamente.</li> </ul> <p>Botões de rádio são localizados na parte superior da tela, os quais indicam a possibilidade de incluir ou não outliers na análise, sendo a opção "Com outlier" marcada inicialmente por padrão. Um menu lateral à esquerda fornece ao usuário opções de navegação pelos requisitos do sistema, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada.</p>
Alternativas	
Pós-condições	<p><b>Pós-condição:</b></p> <p>O sistema permanece na mesma tela.</p>
Exceções	

Tabela B.15: Especificação do requisito funcional 'RF15'

Identificador	RF15
Nome	Gerar relatório sem outlier.
Sumário	O caso de uso é responsável por gerar um relatório para uma disciplina, sem considerar os valores de outlier.
Atores	Docente.
Pré-condições	Concluir o requisito funcional "RF13".
Descrição	<p><b>Fluxo principal:</b></p> <p>1: O sistema exibe uma tela contendo um relatório com informações relacionadas a disciplina selecionada, como nome, código, período e docente que a leciona. Além disso, uma tabela é apresentada com informações de aproveitamento da disciplina, com base nas avaliações realizadas por discentes em questionários. Essas informações são organizadas da seguinte maneira:</p> <ul style="list-style-type: none"> <li>• Questão: corresponde a um identificador único para cada questão avaliada;</li> <li>• Descrição: fornece a descrição completa de cada questão avaliada;</li> <li>• RM: corresponde a um ranking médio para cada questão avaliada por discentes da disciplina;</li> <li>• Situação: classificação do aproveitamento da disciplina com base no ranking médio obtido. A classificação varia entre cinco possibilidades: péssimo, ruim, regular, bom e excelente, obedecendo a escala de 1 à 5 respectivamente.</li> </ul> <p>Botões de rádio são localizados na parte superior da tela, os quais indicam a possibilidade de incluir ou não outliers na análise, sendo a opção "Com outlier" marcada inicialmente por padrão. Um menu lateral à esquerda fornece ao usuário opções de navegação pelos requisitos do sistema, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada.</p> <p>2: O usuário seleciona o botão "Sem outlier".</p> <p>3: O sistema exibe o relatório ignorando os valores de outlier para cada uma das questões avaliadas</p>
Alternativas	
Pós-condições	<p><b>Pós-condição:</b></p> <p>O sistema permanece na mesma tela.</p>
Exceções	

Tabela B.16: Especificação do requisito funcional 'RF16'

Identificador	RF16
Nome	Consultar informações sobre o sistema.
Sumário	O caso de uso é responsável por exibir informações do aplicativo ao usuário.
Atores	(1) Discente, (2) Docente.
Pré-condições	O usuário (1)(2) deve estar logado no sistema.
Descrição	<b>Fluxo principal - usuário (1)(2):</b> 1: O usuário faz clique sobre a opção "Sobre o Aplicativo". 2: O sistema exibe uma tela com informações de desenvolvimento e funcionalidades do aplicativo. Um menu lateral à esquerda fornece ao usuário opções de navegação pelos requisitos do sistema, bem como o rodapé, que possibilita retorno ao início ou exibição da última tela visitada.
Alternativas	
Pós-condições	<b>Pós-condição:</b> O sistema permanece na mesma tela.
Exceções	

Tabela B.17: Especificação do requisito funcional 'RF17'

Identificador	RF17
Nome	Sair do sistema.
Sumário	O caso de uso é responsável por encerrar a sessão do usuário no sistema.
Atores	(1) Discente, (2) Docente.
Pré-condições	O usuário (1)(2) deve estar logado no sistema.
Descrição	<b>Fluxo principal - usuário (1)(2):</b> 1: O usuário faz clique sobre a opção "Sair", presente no menu lateral localizado à esquerda na tela. 2: O sistema encerra a sessão do usuário.
Alternativas	
Pós-condições	<b>Pós-condição:</b> O sistema redireciona para a tela de autenticação.
Exceções	

## B.2 Especificação dos Requisitos não Funcionais do Aplicativo SAD-UFJF

Tabela B.18: Especificação do requisito não funcional 'RNF01'

Identificador	RNF01
Nome	Segurança.
Categoria	Segurança.
Prioridade	Essencial.
Descrição	O sistema deve fornecer mecanismos de segurança e autenticação, alinhados aos adotados pelo SIGA.

Tabela B.19: Especificação do requisito não funcional 'RNF02'

<b>Identificador</b>	<b>RNF02</b>
Nome	Aspectos legais.
Categoria	Legal.
Prioridade	Essencial.
Descrição	O sistema deve seguir orientações e atender à normas da universidade.

Tabela B.20: Especificação do requisito não funcional 'RNF03'

<b>Identificador</b>	<b>RNF03</b>
Nome	Usabilidade.
Categoria	Usabilidade.
Prioridade	Importante.
Descrição	Uso de design responsivo nas interfaces gráficas.

## C Apêndice

### C.1 Prototipação dos cenários com utilização da ferramenta *Balsamiq*

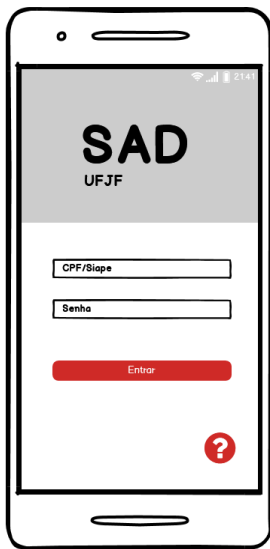


Figura C.1:  
Protótipo da tela  
de autenticação.  
Sistema SAD-  
UFJF.  
Fonte: Autoria  
própria



Figura C.2:  
Protótipo área do  
discente.  
Sistema SAD-  
UFJF.  
Fonte: Autoria  
própria



Figura C.3:  
Protótipo área do  
docente.  
Sistema SAD-  
UFJF.  
Fonte: Autoria  
própria



Figura C.4:  
Protótipo do  
menu lateral  
discente. Sis-  
tema SAD-UFJF.  
Fonte: Autoria  
própria



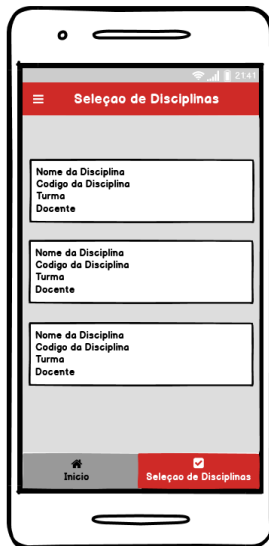


Figura C.5:  
Protótipo con-  
sulta disciplinas.  
Sistema SAD-  
UFJF.  
Fonte: Autoria  
própria

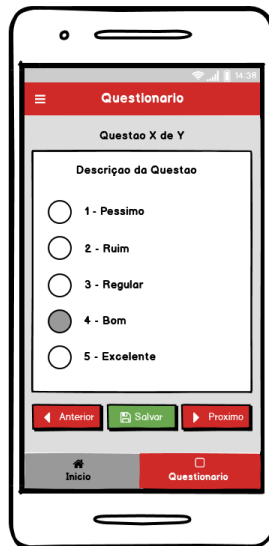


Figura C.6:  
Protótipo ques-  
tionário avali-  
ativo. Sistema  
SAD-UFJF.  
Fonte: Autoria  
própria



Figura C.7:  
Protótipo área de  
comentário.  
Sistema SAD-  
UFJF.  
Fonte: Autoria  
própria

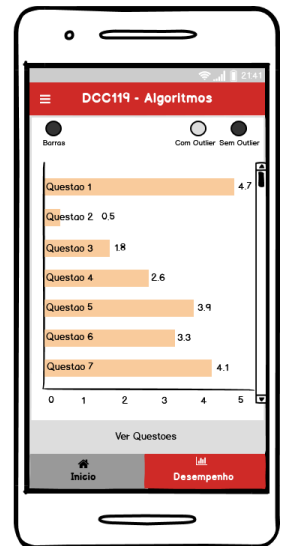


Figura C.8:  
Protótipo gráfico  
barra.  
Sistema SAD-  
UFJF.  
Fonte: Autoria  
própria

## C.2 Interfaces desenvolvidas para os cenários da aplicação

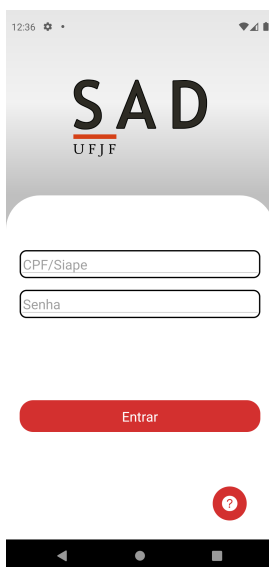


Figura C.9: Tela  
de autenticação.  
Sistema SAD-  
UFJF.  
Fonte: Autoria  
própria



Figura C.10:  
Área do discente.  
Sistema SAD-  
UFJF.  
Fonte: Autoria  
própria

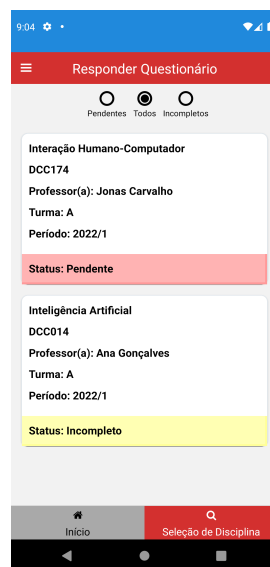


Figura C.11:  
Seleção de ques-  
tionário. Sistema  
SAD-UFJF.  
Fonte: Autoria  
própria

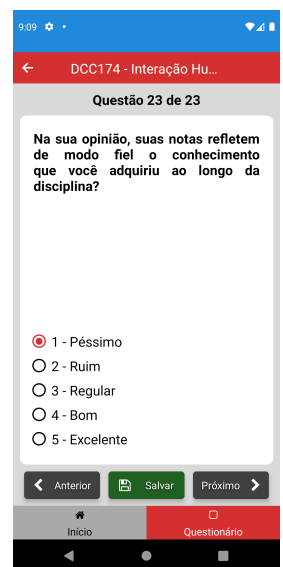


Figura C.12:  
Questionário avali-  
ativo. Sistema  
SAD-UFJF.  
Fonte: Autoria  
própria



Figura C.13: Área do docente. Sistema SAD-UFJF. Fonte: Autoria própria

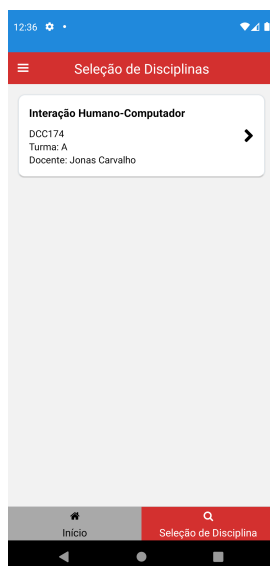


Figura C.14: Tela consulta de disciplinas. Sistema SAD-UFJF. Fonte: Autoria própria

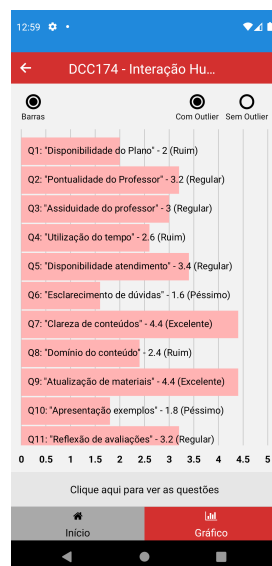


Figura C.15: Geração de gráfico. Sistema SAD-UFJF. Fonte: Autoria própria

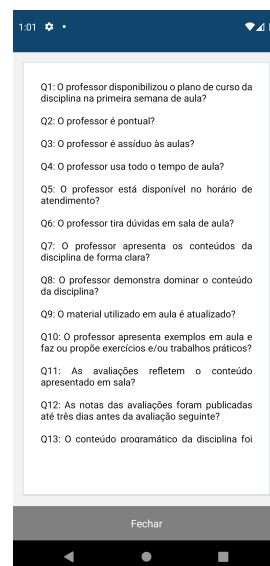


Figura C.16: Perguntas do questionário. Sistema SAD-UFJF. Fonte: Autoria própria

 A interface para geração de relatório, intitulada "DCC174 - Interação Hu...". Ela exibe um relatório de desempenho com o título "Relatório de Desempenho" e o subtítulo "Interação Humano-Computador".

Questão	Descrição	RM	Situação
1	O professor disponibilizou o plano de curso da disciplina na primeira semana de aula?	2	Ruim
2	O professor é pontual?	3.2	Regular
3	O professor é assíduo às aulas?	3	Regular
4	O professor usa todo o tempo de aula?	2.6	Ruim
5	O professor está disponível no horário de atendimento?	3.4	Regular
6	O professor tira dúvidas em sala de aula?	1.6	Péssimo

Figura C.17: Geração de relatório. Sistema SAD-UFJF. Fonte: Autoria própria

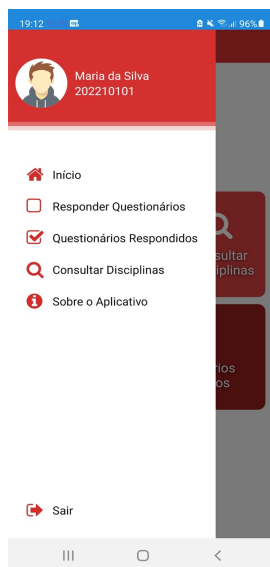


Figura C.18: Menu lateral discente. Sistema SAD-UFJF. Fonte: Autoria própria

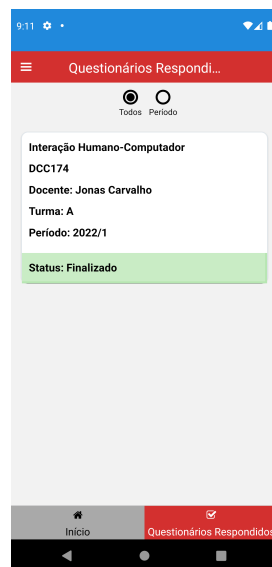


Figura C.19: Questionários concluídos. Sistema SAD-UFJF. Fonte: Autoria própria

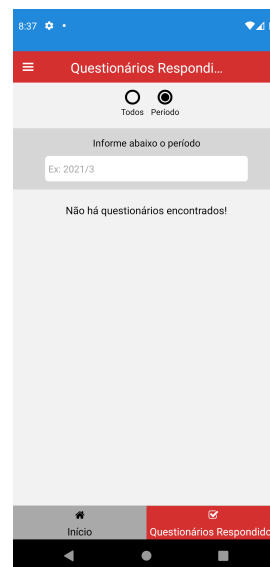


Figura C.20: Busca questionário. Sistema SAD-UFJF. Fonte: Autoria própria

## D Apêndice

### D.1 Diagrama de Caso de Uso

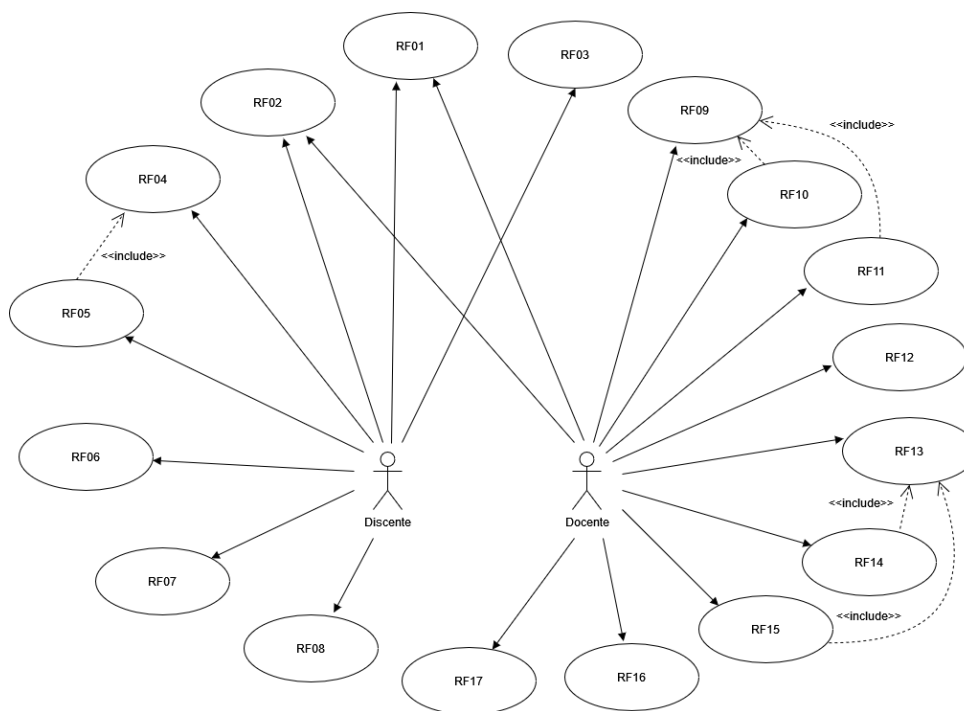


Figura D.1: Diagrama de casos de uso para o sistema. Fonte: Autoria Própria.

### D.2 Diagramas de Atividades

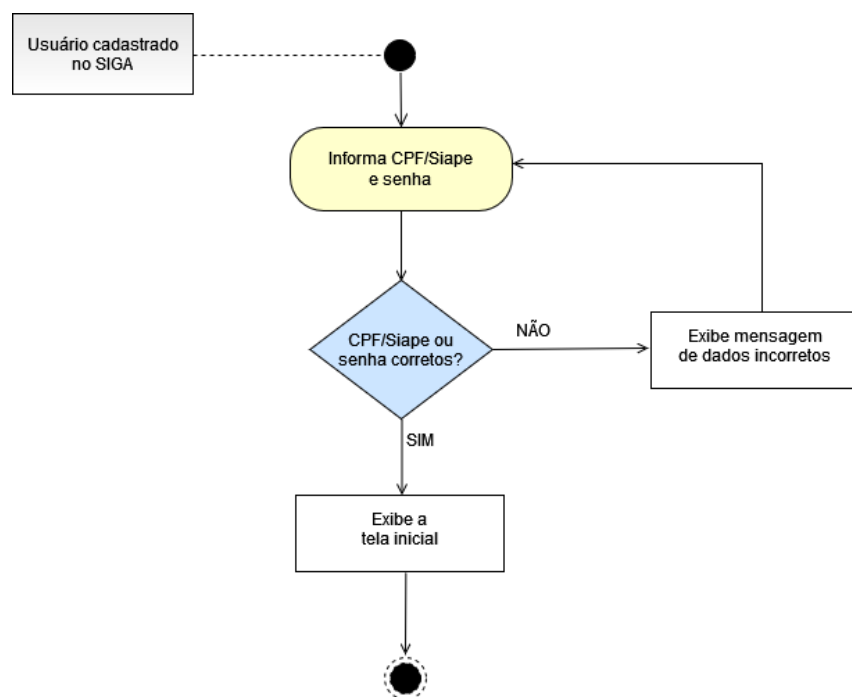


Figura D.2: Diagrama de atividades para o requisito funcional 'RF01'. Fonte: Autoria Própria.

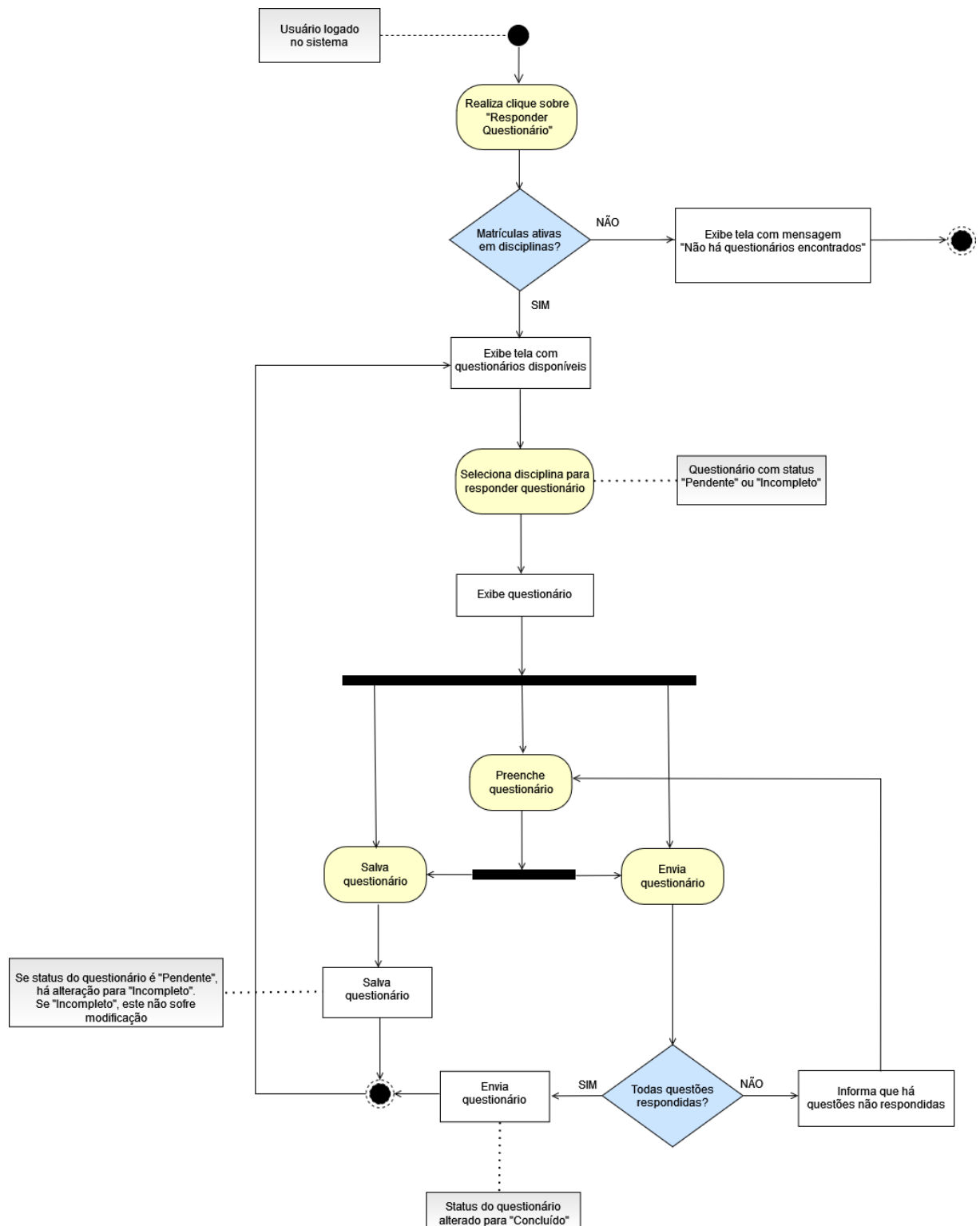


Figura D.3: Diagrama de atividades para os requisitos funcionais 'RF04', 'RF05', 'RF06' e 'RF07'. Fonte: Autoria Própria.