Yves Lima Silva

Ethicalsoft Compliance: Plataforma *Web* para Autoavaliação Ética, Rastreabilidade e Geração de Evidências de Conformidade Declarada Em Projetos de *Software*

Alegre
2026

Yves Lima Silva

Ethicalsoft Compliance: Plataforma *Web* para Autoavaliação Ética, Rastreabilidade e Geração de Evidências de Conformidade Declarada Em Projetos de *Software*

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Universidade Federal do Espírito Santo
Centro de Ciências Exatas, Naturais e da Saúde
Departamento de Computação

Orientador: Prof. Dr. Clayton Vieira Fraga Filho

Alegre
2026

Yves Lima Silva

Ethicalsoft Compliance: Plataforma Web para Autoavaliação Ética, Rastreabilidade e Geração de Evidências de Conformidade Declarada Em Projetos de Software

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado em 20 de julho de 2026.

COMISSÃO EXAMINADORA

Prof. Dr. Clayton Vieira Fraga Filho
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Antonio Almeida de Barros Junior
Universidade Federal do Espírito Santo

Prof. Msc. Giuliano Prado de Morais Giglio
Universidade Federal do Espírito Santo

Agradecimentos

Agradeço aos meus pais, Rita de Fátima e Alberto Oliveira, pelo apoio incondicional, pelos sacrifícios feitos e por sempre acreditarem na importância da educação. Sem o suporte de vocês, esta conquista não seria possível.

Agradeço ao meu irmão, Nicolas Lima, a pessoa com quem sei que sempre posso contar.

Agradeço a Lavínia Souza, minha namorada e futura esposa, que esteve comigo durante a maior parte desta jornada. Minha companheira de vida, que me apoiou incondicionalmente, inclusive nos momentos mais difíceis.

Ao meu orientador, Prof. Dr. Clayton Vieira Fraga Filho, pela paciência, confiança e pelo direcionamento preciso. Suas orientações foram indispensáveis para a idealização e o desenvolvimento do EthicalSoft Compliance.

Aos professores do curso de Sistemas de Informação da Universidade Federal do Espírito Santo (UFES) - campus Alegre, por todos os ensinamentos e por contribuírem para a minha formação profissional.

Aos meus amigos e colegas de curso, em especial a Ayler Francisco, Lucas Damaso e Thobias Dantas, pelas trocas de conhecimento e pelo companheirismo ao longo de todos estes anos.

Resumo

Este trabalho apresenta o EthicalSoft Compliance, plataforma *web* desenvolvida para apoiar a autoavaliação ética, a rastreabilidade de decisões e a geração de evidências de conformidade declarada em projetos de *software*, incluindo contextos de desenvolvimento assistido por inteligência artificial. A pesquisa é aplicada, exploratória e descritiva e adota a *Design Science Research* como procedimento principal. O artefato operacionaliza elementos de modelos éticos aplicados ao ciclo de desenvolvimento por meio de questionários configuráveis, indicadores determinísticos, painéis, notificações, documentos e registros de emissão. A aplicação foi implementada com Angular, Spring Boot, PostgreSQL e MongoDB, com arquitetura hexagonal no *backend* e integração assistiva com modelos de linguagem. A avaliação combinou verificação técnica e demonstração estruturada no cenário simulado SelecIA RH, composto por quatro iterações e cinco papéis. Os questionários produziram ISEQs de 5,36%, 92,14%, 80,00% e 100,00%, e o projeto obteve ISEP de 75,91%, classificado na faixa B segundo o limiar operacional configurado. Os resultados demonstram a execução dos fluxos, a aplicação dos cálculos e a geração das evidências previstas. Por utilizarem dados sintéticos e não envolverem participantes externos, não comprovam efetividade organizacional, mudança de comportamento ou conformidade legal ou normativa.

Palavras-chave: ética em engenharia de *software*; governança de TI; rastreabilidade; dívida ética; Desenvolvimento de *Software* Orientado pela Ética; conformidade declarada.

Abstract

This work presents EthicalSoft Compliance, a web platform developed to support ethical self-assessment, decision traceability, and the generation of declared compliance evidence in software projects, including AI-assisted development contexts. The study is applied, exploratory, and descriptive and adopts Design Science Research as its main procedure. The artifact operationalizes elements of ethical models applied to the software lifecycle through configurable questionnaires, deterministic indicators, dashboards, notifications, documents, and issuance records. The application was implemented with Angular, Spring Boot, PostgreSQL, and MongoDB, using a hexagonal backend architecture and an assistive integration with language models. Evaluation combined technical verification with a structured demonstration in the simulated SelecIA RH scenario, comprising four iterations and five roles. The questionnaires produced ISEQ values of 5.36%, 92.14%, 80.00%, and 100.00%, while the project achieved an ISEP of 75.91%, classified in band B according to the configured operational threshold. The results demonstrate execution of the implemented flows, application of the calculations, and generation of the expected evidence. Because the evaluation used synthetic data and no external participants, it does not establish organizational effectiveness, behavioral change, or legal or normative compliance.

Keywords: software engineering ethics; IT governance; traceability; ethical debt; Ethical-Driven Software Development; declared compliance.

Lista de ilustrações

Figura 1 – Exemplo preenchido de cartão de índice para avaliação da adequação da solução aos usuários. 35
Figura 2 – Fluxo integral da pesquisa. 57
Figura 3 – Arquitetura implementada no EthicalSoft Compliance. 65
Figura 4 – Diagrama de casos de uso do EthicalSoft Compliance. 66
Figura 5 – Modelo conceitual de domínio do EthicalSoft Compliance. 68
Figura 6 – Modelo Entidade-Relacionamento do núcleo relacional do EthicalSoft Compliance. 70
Figura 7 – *Collections* do MongoDB presentes na instância utilizada. 73
Figura 8 – Fluxo de autenticação e recuperação de acesso. 77
Figura 9 – Listagem e filtragem dos projetos acessíveis ao usuário. 78
Figura 10 – Assistente de criação e configuração do projeto. 79
Figura 11 – Detalhes do projeto e ações administrativas disponíveis. 79
Figura 12 – Resposta ao questionário e registro condicionado de evidência ou justificativa. 80
Figura 13 – Organização da interface do painel consolidado do projeto. 81
Figura 14 – Consulta e acesso aos documentos de conformidade gerados. 82
Figura 15 – Resultado da análise estática da *branch* principal no SonarQube Cloud. 86
Figura 16 – Pesos das *sprints* e etapas configurados para o cenário simulado. . . 91
Figura 17 – Distribuição das perguntas por escopo no cenário simulado. 92
Figura 18 – Registro de resposta e justificativa durante a execução do cenário. . 93
Figura 19 – Síntese dos indicadores do questionário referente à *Sprint* 2. 96
Figura 20 – Resultados individuais que compõem o ISEQ da *Sprint* 2. 96
Figura 21 – Distribuição da conformidade por papel e etapa no mapa de calor. . 97
Figura 22 – Termos recorrentes nas justificativas de não conformidade. 98
Figura 23 – Indicadores disponibilizados ao participante no cenário simulado. . 99
Figura 24 – Resultado consolidado do projeto SelecIA RH. 100
Figura 25 – Interpretação assistida dos indicadores do cenário simulado. 101
Figura 26 – Certificado emitido para o resultado final do cenário simulado. . . . 102
Figura 27 – Consulta às respostas, justificativas e evidências consolidadas. . . . 103
Figura 28 – Tela de *login* com destaque para o *link* de cadastro. 169
Figura 29 – Formulário de cadastro preenchido com e-mail já existente. 170
Figura 30 – *Modal* de erro exibido ao tentar cadastrar e-mail duplicado. 171
Figura 31 – *Modal* de erro exibido ao informar credenciais inválidas no *login*. . . 171

Figura 32 – Seletor de contas Google acionado pelo sistema após clique no botão de autenticação federada. 172
Figura 33 – Tela inicial exibida após autenticação bem-sucedida via Google OAuth. 172
Figura 34 – Tela de *login* com destaque para o *link* de recuperação de senha. . . 174
Figura 35 – Tela de recuperação de conta com campo de e-mail preenchido. . . 174
Figura 36 – E-mail de recuperação recebido com código de verificação. 175
Figura 37 – Tela de registro de nova senha após validação do código. 175
Figura 38 – Tela de *login* com *prompt* do navegador para atualização de senha, confirmando o sucesso da redefinição. 176
Figura 39 – Tela de listagem de projetos com destaque para o botão de criação. 177
Figura 40 – *Modal* de seleção do tipo de projeto com opção Iterativo Incremental selecionada. 178
Figura 41 – Formulário de criação com *template* selecionado e número de iterações calculado automaticamente. 178
Figura 42 – Seção de etapas pré-configuradas pelo *template* com pesos atribuídos. 179
Figura 43 – Seção de representantes pré-configurados pelo *template* com papéis e pesos. 179
Figura 44 – Seção de questionários com quatro *sprints* geradas e faixas de aplicação calculadas. 179
Figura 45 – Edição do questionário *Sprint* 1 exibindo as 18 perguntas vinculadas pelo *template* com classificações e papéis. 180
Figura 46 – *Modal* de confirmação de criação do projeto informando geração de *template* e envio de e-mails. 180
Figura 47 – Listagem de projetos com o projeto National Geograph recém-criado em *status* Aberto. 180
Figura 48 – E-mail de vinculação ao projeto recebido pelo representante com *link*, papéis e situação atual. 181
Figura 49 – E-mail de notificação do questionário *Sprint* 1 disponível para resposta. 181
Figura 50 – *Modal* de seleção do tipo de projeto com opção Cascata selecionada. 182
Figura 51 – Formulário de criação com *template* Base Cascata selecionado e dados do projeto preenchidos. 183
Figura 52 – Seção de etapas com três fases pré-configuradas e faixas de aplicação calculadas automaticamente. 183
Figura 53 – Seção de representantes com três participantes configurados e múltiplos encargos por participante. 183
Figura 54 – Seção de questionários com três itens vinculados às etapas e datas sequenciais calculadas. 184

Figura 55 – Listagem de projetos com o projeto Extensão PopTop recém-criado em *status* Aberto do tipo Cascata. 184
Figura 56 – Painel de notificações internas exibindo vinculação ao projeto e questionário pendente. 185
Figura 57 – Tela de detalhes do projeto 14 com o questionário *Sprint* 1 em andamento e todos os representantes com *status* Pendente. 185
Figura 58 – Tela de resposta ao questionário com botão "Enviar respostas" desabilitado e campo de evidência obrigatório em branco. 186
Figura 59 – Tela de resposta com todas as perguntas respondidas e campos de evidência preenchidos, botão de envio habilitado. 186
Figura 60 – *Modal* de confirmação exibido após envio bem-sucedido das respostas. 186
Figura 61 – Tela de resposta ao questionário com duas perguntas respondidas e destaque para o botão "Salvar Rascunho". 187
Figura 62 – *Modal* de confirmação exibido após salvamento do rascunho. 188
Figura 63 – Tela de *login* com as credenciais do representante para retomada da sessão. 188
Figura 64 – Questionário retomado com *status* "Em andamento" e progresso de 2/4 respostas preservado. 189
Figura 65 – Tela de detalhes do projeto 16 com *Sprint* 1 com 3/3 respondentes e destaque para o botão de encerramento forçado. 190
Figura 66 – *Modal* de confirmação de encerramento do questionário com aviso de cálculo do ISEQ. 190
Figura 67 – *Modal* de sucesso confirmando o encerramento do questionário e o cálculo do ISEQ. 191
Figura 68 – *Dashboard* do questionário *Sprint* 1 com ISEQ de 68,37%, distribuição por faixa e métricas da equipe. 191
Figura 69 – *Dashboard* do questionário *Sprint* 2 com ISEQ de 92,86%, Faixa A e desvio padrão de 11,79%. 192
Figura 70 – *Dashboard* consolidado do projeto com ISEP de 80,62%, Faixa B e evolução por *sprint*. 193
Figura 71 – E-mail de boletim de não conformidade recebido pelo representante com ISEQ, faixa obtida e faixa exigida. 194
Figura 72 – Histórico de *downloads* do navegador exibindo o arquivo PDF do certificado em *download*. 194
Figura 73 – Certificado de Ciência Ética e Conformidade Declarada emitido para o projeto SelecIA RH com ISEP 75,91% e código. 195
Figura 74 – *Dashboard* do questionário 20 com gráfico de distribuição por faixa e *heatmap* de risco por encargo e etapa. 196

Figura 75 – *Dashboard* do questionário 20 com nuvem de palavras das justificativas e *widgets* de análise de IA. 196
Figura 76 – Painel individual do representante com ICP 100%, média pessoal e média da equipe. 197
Figura 77 – Gráfico de evolução de conformidade do representante e interpretação textual do desempenho. 197
Figura 78 – *Dashboard* do projeto com destaque para o botão de exportação CSV e *download* iniciado no navegador. 198
Figura 79 – Arquivo CSV aberto em planilha com os dados analíticos do projeto por representante e por *sprint*. 198
Figura 80 – Histórico de *downloads* do navegador exibindo o arquivo JSON do questionário em *download*. 199
Figura 81 – *Dashboard* do questionário com *widget* de Explicação dos Resultados ISEQ. 200
Figura 82 – Tela de configuração do *token* de IA com *status* cadastrado e opções de salvar/remover. 200
Figura 83 – Tela de detalhes do projeto 16 com *Sprint* 2 pendente e destaque para o botão de reagendamento. 201
Figura 84 – *Modal* de sucesso confirmando o reagendamento do questionário *Sprint* 2 com o novo período. 201
Figura 85 – Tela de detalhes do projeto 16 após reagendamento, com *Sprint* 2 em andamento e novo período exibido. 202
Figura 86 – E-mail de notificação de reagendamento recebido pelo representante com período anterior e novo período. 202
Figura 87 – Seleção da estrutura do projeto antes da configuração. 212
Figura 88 – Configurações da conta e gerenciamento da credencial de IA. 213

Lista de quadros

Quadro 1 – Principais perguntas por fase. 36
Quadro 2 – Ética para Lacunas de Conhecimento. 39
Quadro 3 – Ética para Identificação de Requisitos. 40
Quadro 4 – Ética para Qualidade dos Requisitos. 42
Quadro 5 – Ética para Relutância. 43
Quadro 6 – Ética para Atividades Proibidas. 45
Quadro 7 – Faixas de classificação dos indicadores. 60
Quadro 8 – Modelo documental das *collections* do MongoDB. 71
Quadro 9 – Responsabilidade dos armazenamentos no modelo poliglota. . 75
Quadro 10 – Componentes de visualização e contextos de uso. 81
Quadro 11 – Rastreabilidade entre objetivos, execução e resultados observados. 88
Quadro 12 – Representantes, papéis e pesos do cenário simulado. 91
Quadro 13 – Matriz dos resultados consolidados por *sprint*. 94
Quadro 14 – Atendimento aos objetivos específicos. 106
Quadro 15 – Matriz de rastreabilidade entre objetivos, artefatos e evidências. 164
Quadro 16 – Inventário e classificação dos testes automatizados. 167
Quadro 17 – Representantes, papéis e pesos. 203
Quadro 18 – Questionários e pesos. 203
Quadro 19 – Peso por Etapas. 204
Quadro 20 – Resultados esperados e obtidos no cenário SelecIA RH. 205
Quadro 21 – Identificação da versão entregue. 207
Quadro 22 – Variáveis de ambiente obrigatórias do *backend*. 207
Quadro 23 – Autenticação (/api/auth). 209
Quadro 24 – Projetos (/api/projects). 209
Quadro 25 – Questionários (/api/projects/{projectId}/questionnaires). . . 209
Quadro 26 – Respostas (/api/questionnaires/{questionnaireId}/responses). 209
Quadro 27 – *Dashboard* (`/api/dashboard`). 210
Quadro 28 – Documentos (/api/documents). 210
Quadro 29 – Inteligência artificial (/api/ai). 210
Quadro 30 – *Templates* (`/api/templates`). 210
Quadro 31 – Exportação (/api/export). 211

Lista de abreviaturas e siglas

ACM	Association for Computing Machinery
API	Application Programming Interface
APO	Align, Plan and Organise
COBIT	Control Objectives for Information and Related Technologies
CORS	Cross-Origin Resource Sharing
CSV	Comma-Separated Values
DSR	Design Science Research
EDSD	Ethical-Driven Software Development
EDM	Evaluate, Direct and Monitor
EUS	Ethical User Stories
FA	Fluxo Alternativo
FE	Fluxo de Exceção
HTML	HyperText Markup Language
IA	Inteligência Artificial
ICP	Indicador de Conformidade do Participante
IEM	Indicador de Ética por Etapa
IEEE-CS	Institute of Electrical and Electronics Engineers — Computer Society
ISEQ	Indicador Sistêmico Ético do Questionário
ISEP	Indicador Sistêmico Ético do Projeto
ITIL	Information Technology Infrastructure Library
JPA	Jakarta Persistence API
JWT	JSON Web Token
LLM	Large Language Model
MVC	Model-View-Controller

PDF	Portable Document Format
QVD	Quality Value Drivers
REST	Representational State Transfer
RH	Recursos Humanos
TCC	Trabalho de Conclusão de Curso
TI	Tecnologia da Informação
UC	Use Case

Sumário

1 INTRODUÇÃO 21

2 OBJETIVOS 23
2.1 Objetivo Geral 23
2.2 Objetivos Específicos 23

3 FUNDAMENTAÇÃO TEÓRICA 25
3.1 Ética Profissional e Qualidade no Desenvolvimento de *Software* 25
3.1.1 O Código de Ética da Engenharia de *Software* da ACM/IEEE-CS 26
3.1.2 Omissão de Práticas de Desenvolvimento de *Software* de Qualidade 27
3.1.2.1 Fatores que Contribuem para a Omissão de Práticas de Qualidade 28
3.1.2.2 Instâncias e Consequências da Omissão de Práticas de Qualidade 29
3.2 Conformidade Ética no Ciclo de Vida do *Software* 30
3.2.1 Definição de Conformidade Ética 30
3.2.2 Desafios na Implementação da Conformidade Ética . . 30
3.2.3 Mecanismos de Avaliação e Certificação Ética 33
3.3 Modelos e Quadros de Ética Aplicados ao Desenvolvimento . 33
3.3.1 Desenvolvimento de *Software* Orientado pela Ética . . 34
3.3.1.1 A Crise do *Software* e o Papel da Ética 34
3.3.1.2 Integração da Ética no Ciclo de Vida do Desenvolvimento de *Software* 34
3.3.2 Ética como Motor da Qualidade em Projetos Ágeis de *Software* 37
3.3.3 Quadro Ético para Engenharia de Requisitos 38
3.3.3.1 Identificação dos Problemas na Engenharia de Requisitos 38
3.3.3.2 Problemas de Identificação de Requisitos 39
3.3.3.3 Qualidade dos Requisitos e Impacto Ético 41
3.3.3.4 Relutância em Fornecer Requisitos 42
3.3.3.5 Atividades Proibidas e Não Permitidas 44
3.3.3.6 Validação do *Framework* Proposto 46
3.3.4 Comparação entre os Modelos Éticos 46
3.4 Governança, Transparência e Dívida Técnica e Ética 47
3.4.1 Governança de TI e Transparência em Projetos de *Software* 47

3.4.2 Dívida Técnica, Dívida Ética e Opacidade nas Decisões 48
3.4.3 Integração entre Governança, Modelos Éticos e Prática de Projeto . . . 49
3.5 Desenvolvimento de *Software* Assistido por Inteligência Artificial 51
3.6 Conceitos de Arquitetura e Modelagem para o Artefato . . . 51
3.7 Síntese da Literatura e Lacuna de Pesquisa . . . 52

4 METODOLOGIA . . . 54
4.1 Caracterização da Pesquisa . . . 54
4.2 Tecnologias Adotadas . . . 55
4.3 Etapas da Pesquisa . . . 56
4.3.1 Identificação do Problema e Revisão da Literatura . . . 57
4.3.2 Elicitação de Requisitos e Operacionalização dos Modelos Éticos . . . 58
4.3.3 Projeto e Prototipação do Artefato . . . 58
4.3.4 Implementação e Verificação Técnica . . . 58
4.4 Modelo de Avaliação e Indicadores . . . 59
4.4.1 Indicador de Conformidade do Participante (ICP) . . . 59
4.4.2 Indicador Sistêmico Ético do Questionário (ISEQ) . . . 59
4.4.3 Indicador Sistêmico Ético do Projeto (ISEP) . . . 59
4.4.4 Indicador de Ética por Etapa (IEM) . . . 60
4.4.5 Pesos, Faixas e Limiar Operacional . . . 60
4.5 Protocolo de Demonstração e Avaliação . . . 61
4.6 Ameaças à Validade e Limitações . . . 62
4.6.1 Validade de Construção . . . 62
4.6.2 Validade Interna . . . 62
4.6.3 Validade Externa . . . 62
4.6.4 Confiabilidade . . . 62
4.7 Disponibilidade e Reprodutibilidade do Artefato . . . 63

5 DESENVOLVIMENTO DO ARTEFATO . . . 64
5.1 Visão Geral e Arquitetura Implementada . . . 64
5.1.1 Casos de Uso Centrais . . . 65
5.2 Modelo de Domínio e Persistência . . . 67
5.2.1 Modelo de Domínio . . . 67
5.2.2 Modelo Entidade-Relacionamento . . . 69
5.2.3 Modelo Documental do MongoDB . . . 71
5.2.4 Justificativa da Persistência Poliglota . . . 73
5.3 *Front-end* . . . 76
5.3.1 Organização da Interface e Navegação . . . 76

5.3.2 Fluxos de Interação da Aplicação 77
5.3.2.1 Autenticação e Acesso aos Projetos 77
5.3.2.2 Configuração e Gerenciamento do Projeto 78
5.3.2.3 Aplicação dos Questionários e Registro de Evidências 80
5.3.2.4 Painéis, Documentos e Configurações 80
5.4 *Back-end* 83
5.4.1 API REST 83
5.4.2 Exemplo de Fluxo de Requisição 83
5.4.3 Autenticação e Autorização 84
5.5 Integração Assistiva de Inteligência Artificial 84
5.6 Geração de Documentos, Notificações e Exportações 84
5.7 Infraestrutura e Implantação 85
5.8 Verificação e Qualidade do Código 85

6 RESULTADOS E DISCUSSÃO 87
6.1 Artefato Produzido e Rastreabilidade dos Requisitos 87
6.2 Configuração do Cenário Simulado 91
6.3 Execução Funcional do Cenário 93
6.4 Resultados dos Indicadores 94
6.5 Painéis e Visualizações Analíticas 95
6.5.1 Painel do Questionário 95
6.5.2 Painel Individual do Participante 98
6.5.3 Painel Consolidado do Projeto 99
6.5.4 Interpretação Assistida por Inteligência Artificial . . . 100
6.6 Documentos Gerados e Trilha de Auditoria 101
6.7 Discussão dos Resultados 103
6.8 Limitações das Evidências 104

7 CONCLUSÃO 106
7.1 Atendimento aos Objetivos Específicos 106
7.2 Contribuições do Trabalho 108
7.3 Limitações 108
7.4 Trabalhos Futuros 109

REFERÊNCIAS 110

APÊNDICES 116

APÊNDICE A – NARRATIVAS DE CASO DE USO . . 117

A.1 UC01 — Registrar-se no Sistema . . . 117
A.2 UC02 — Realizar *Login* com E-mail e Senha . . . 118
A.3 UC03 — Realizar *Login* com Google . . . 118
A.4 UC04 — Recuperar Conta e Redefinir Senha . . . 119
A.5 UC05 — Alterar Senha e Concluir Primeiro Acesso . . . 120
A.6 UC06 — Listar e Filtrar Projetos . . . 121
A.7 UC07 — Criar Projeto . . . 122
A.8 UC08 — Editar Projeto . . . 123
A.9 UC09 — Publicar Projeto . . . 124
A.10 UC10 — Excluir Projeto . . . 125
A.11 UC11 — Visualizar Detalhes do Projeto . . . 125
A.12 UC12 — Manter Etapas . . . 126
A.13 UC13 — Manter Representantes . . . 127
A.14 UC14 — Manter Questionários . . . 128
A.15 UC15 — Responder Questionário . . . 128
A.16 UC16 — Visualizar Questionário em Modo Administrativo . . . 129
A.17 UC17 — Reagendar Questionário . . . 130
A.18 UC18 — Notificar Representantes Pendentes . . . 131
A.19 UC19 — Encerrar Questionário . . . 132
A.20 UC20 — Visualizar *Dashboard* Consolidado do Projeto . . . 132
A.21 UC21 — Encerrar Projeto . . . 133
A.22 UC22 — Visualizar *Dashboard* do Questionário . . . 134
A.23 UC23 — Visualizar *Dashboard* Individual . . . 134
A.24 UC24 — Visualizar Respostas Consolidadas . . . 135
A.25 UC25 — Exportar Dados Analíticos . . . 136
A.26 UC26 — Utilizar e Gerar *Templates* de Projeto . . . 136
A.27 UC27 — Baixar e Registrar Emissão de Certificado . . . 137
A.28 UC28 — Baixar, Registrar e Emitir Boletim de Não Conformidade 138
A.29 UC29 — Configurar *Token* Pessoal de IA . . . 139
A.30 UC30 — Consultar análises assistidas por inteligência artificial 140
A.31 UC31 — Gerenciar Expiração de Sessão . . . 140

APÊNDICE B – REQUISITOS E REGRAS DE NEGÓCIO 142
B.1 BR01 – Questionário . . . 142
B.1.1 Elaboração e Atribuição de Perguntas . . . 142
B.1.2 Reutilização e Alteração de Questionários . . . 142
B.1.3 Rascunho . . . 143
B.1.4 Modos de Visualização . . . 143
B.1.5 Funcionalidades do Sistema . . . 143
B.1.6 Barra de Progresso . . . 143

B.1.7 Validações . . . 143
B.1.8 Notificações . . . 144
B.1.9 Exportação de Dados . . . 144
B.1.10 Usabilidade . . . 144
B.2 BR02 – Projeto . . . 144
B.2.1 Tipos de Estrutura . . . 144
B.2.2 *Status* do Projeto . . . 144
B.2.3 *Status* de Linha do Tempo dos Questionários . . . 145
B.2.4 Dados Necessários para Criação do Projeto . . . 145
B.2.5 Publicação do Projeto . . . 145
B.2.6 Edição de Projeto Aberto . . . 146
B.2.7 Encerramento do Projeto . . . 146
B.2.8 Exclusão do Projeto . . . 146
B.2.9 Configuração das Iterações . . . 146
B.2.10 Funcionalidades do Sistema . . . 146
B.2.11 Validações . . . 146
B.2.12 Notificações . . . 147
B.3 BR03 – Membro da Equipe . . . 147
B.3.1 Encargos . . . 147
B.3.2 Perfis de Acesso . . . 147
B.3.3 Cadastro de Membros . . . 147
B.3.4 Primeiro Acesso . . . 147
B.3.5 Vinculação a Projetos . . . 147
B.3.6 Peso por Representante . . . 148
B.4 BR04 – Classificação Ética do Projeto . . . 148
B.4.1 Descrição . . . 148
B.4.2 Elementos . . . 148
B.4.2.1 Projeto . . . 148
B.4.2.2 Tipo de projeto . . . 149
B.4.2.3 Iteração . . . 149
B.4.2.4 Etapa . . . 149
B.4.2.5 Questionário . . . 149
B.4.2.6 Pergunta . . . 149
B.4.2.7 Representante . . . 150
B.4.2.8 Papel . . . 150
B.4.2.9 Resposta . . . 150
B.4.2.10 ICP . . . 150
B.4.2.11 ISEQ . . . 150
B.4.2.12 ISEP . . . 151

B.4.2.13 Certificação . . . 151
B.4.2.14 Regras gerais . . . 151
B.4.2.15 Regras de elegibilidade . . . 152
B.4.2.16 Regras de cobertura por papel e etapa . . . 152
B.4.2.17 Regras de validade das respostas . . . 152
B.4.2.18 Cálculo dos indicadores . . . 152
B.4.2.19 Regras para certificação . . . 153
B.4.2.20 Redação formal . . . 153
B.4.2.21 Regras de interface e interação . . . 154
B.4.2.21.1 Regras gerais de edição do projeto . . . 154
B.4.2.21.2 Regras para questionários em projetos iterativos . 154
B.4.2.21.3 Regras para questionários em projetos cascata . . 154
B.5 BR04.1 — Classificação e Tipo das Perguntas . . . 155
B.6 BR05 – Controle de Acesso ao Sistema . . . 156
B.6.1 Perfil ADMIN (Analista de Qualidade) . . . 156
B.6.2 Perfil USER (Representante) . . . 157
B.7 BR06 – Boletim de Não Conformidade Ética . . . 157
B.7.1 Conteúdo do Boletim . . . 157
B.7.2 Processo de Revisão . . . 157
B.7.3 Permissões . . . 157
B.7.4 Histórico e Notificações . . . 157
B.8 BR07 – Certificado de Ciência Ética . . . 158
B.8.1 Conteúdo do Certificado . . . 158
B.8.2 Permissões . . . 158
B.8.3 Certificado Consolidado . . . 158
B.8.4 Armazenamento . . . 158
B.9 BR08 – Previsão de Data Limite de Resposta dos Questionários 158
B.9.1 Projeto Cascata . . . 158
B.9.2 Projeto Iterativo . . . 159
B.9.3 Reagendamento . . . 159
B.9.4 Edição das Datas . . . 159
B.10 BR09 – *Template* . . . 159
B.10.1 Criação de *Template* . . . 159
B.10.2 Estrutura do *Template* . . . 159
B.10.3 Tipos de Importação Parcial . . . 160
B.10.4 Uso na Criação de Projeto . . . 160
B.10.5 Listagem de *Templates* . . . 160
B.11 BR10 – Notificações Internas . . . 160
B.11.1 Eventos de Notificação . . . 160

B.11.2 Estrutura da Notificação 160
B.11.3 Gerenciamento 161
B.12 BR11 – Inteligência Artificial 161
B.12.1 Funcionalidades de IA 161
B.12.2 Disponibilidade 161
B.13 BR12 – Indicadores de Dívida Ética e Técnica 161
B.13.1 Dívida Ética 162
B.13.2 Dívida Técnica 162
B.13.3 Disponibilidade 162
B.14 BR13 – Dimensões de Governança 162
B.15 BR14 – *Preview* de Alterações 162
B.15.1 Fluxo de Edição com Simulação Prévia 163
B.15.2 Bloqueios de Alteração 163

APÊNDICE C – MATRIZ DE RASTREABILIDADE . . 164

APÊNDICE D – CASOS DE TESTE E EVIDÊNCIAS DE EXECUÇÃO 166
D.1 Testes automatizados 166
D.2 Testes funcionais manuais 169
D.2.1 TF01 — Cadastro de usuário com e-mail duplicado (UC01) 169
D.2.2 TF02 — *Login* com credenciais inválidas (UC02) 169
D.2.3 TF03 — *Login* via Google OAuth (UC03) 170
D.2.4 TF04 — Recuperação de conta por código de e-mail (UC04) 173
D.2.5 TF05 — Criação de projeto iterativo com *template* (UC07, UC26) 173
D.2.6 TF06 — Criação de projeto cascata com *template* (UC07, UC26) 177
D.2.7 TF07 — Resposta ao questionário com campo obrigatório ausente (UC15) 184
D.2.8 TF08 — Salvamento de rascunho durante preenchimento (UC15) 187
D.2.9 TF09 — Encerramento manual de questionário (UC19) 189
D.2.10 TF10 — Cálculo do ISEQ da *Sprint* 1 (UC19, BR04) . 189
D.2.11 TF11 — Cálculo do ISEQ da *Sprint* 2 (UC19, BR04) . 192
D.2.12 TF12 — Cálculo do ISEP consolidado (UC21, BR04) . 192
D.2.13 TF13 — Geração do boletim de não conformidade (UC28, BR06) 193

D.2.14 TF14 — Emissão do Certificado de Ciência Ética e Conformidade Declarada (UC27, BR07) 193
D.2.15 TF15 — *Dashboard* do questionário com *heatmap* e nuvem de palavras (UC22) 194
D.2.16 TF16 — *Dashboard* individual do representante (UC23) 195
D.2.17 TF17 — Exportação de CSV do *dashboard* do projeto (UC25) 197
D.2.18 TF18 — Exportação de JSON do *dashboard* do questionário (UC25) 198
D.2.19 TF19 — Configuração e uso do *token* de IA (UC29, UC30) 199
D.2.20 TF20 — Reagendamento de questionário pendente (UC17) 199

APÊNDICE E – ESPECIFICAÇÃO DO CENÁRIO DEMONSTRATIVO SELECIA RH 203
E.1 Descrição do cenário 203
E.2 Configuração do projeto 203
E.3 Distribuição e classificação das perguntas 204
E.4 Respostas configuradas por *sprint* 205
E.5 Resultados esperados e obtidos 205

APÊNDICE F – CONTRATO DA API, INSTRUÇÕES DE EXECUÇÃO E IDENTIFICAÇÃO DA VERSÃO 207
F.1 Pré-requisitos de execução 207
F.2 Instruções de execução local 208
F.3 Contrato resumido da API REST 208

APÊNDICE G – TELAS COMPLEMENTARES DA APLICAÇÃO 212
G.1 Seleção do Tipo de Projeto 212
G.2 Configurações da Conta 212

1 INTRODUÇÃO

O desenvolvimento de *software* exerce influência crescente sobre atividades sociais, econômicas e institucionais. À medida que sistemas passam a apoiar decisões sensíveis, amplia-se a necessidade de combinar qualidade técnica, segurança, transparência e responsabilidade sobre os impactos produzidos. Entretanto, decisões de requisitos, projeto e implementação ainda podem permanecer dispersas em ferramentas técnicas ou dependentes do conhecimento de poucos participantes, dificultando a reconstrução de suas justificativas e consequências (LURIE; MARK, 2016; BIABLE; GARCIA; MIDEKSO, 2023; GHANBARI; VARTIAINEN; SIPONEN, 2018).

Metodologias ágeis favorecem entregas incrementais e adaptação a mudanças, mas também concentram decisões em ciclos curtos. Quando pressões por prazo e escopo levam à omissão de análise, teste, documentação ou revisão, podem ser acumuladas dívida técnica e dívida ética. Nesse contexto, qualidade não deve ser tratada apenas como propriedade do produto final, mas como compromisso distribuído pelo ciclo de desenvolvimento (SCHWABER; SUTHERLAND, 2020; GHANBARI; VARTIAINEN; SIPONEN, 2018; HALME et al., 2024).

Três dimensões precisam ser distinguidas. A primeira é a ética no processo de desenvolvimento, relacionada aos deveres profissionais, à consideração dos impactos e ao julgamento exercido por quem participa do projeto. A segunda é a conformidade ética declarada, entendida neste trabalho como o registro verificável de respostas, justificativas e evidências diante de critérios previamente configurados. Essa conformidade não equivale à certificação moral ou legal do *software*. A terceira dimensão é o desenvolvimento assistido por inteligência artificial, que introduz riscos específicos de erro, viés, exposição de dados, opacidade e perda de rastreabilidade, mas não constitui o mecanismo decisório da avaliação realizada pela plataforma (ACM; IEEE COMPUTER SOCIETY, 1999; HOU et al., 2024; VAKKURI et al., 2022).

O Código de Ética e Prática Profissional da Engenharia de *Software* estabelece princípios relacionados ao interesse público, aos clientes, ao produto, ao julgamento e à gestão (ACM; IEEE COMPUTER SOCIETY, 1999). Modelos como o Desenvolvimento de *Software* Orientado pela Ética, proposto por Lurie e Mark (2016), e o quadro de Biable, Garcia e Midekso (2023) aproximam esses princípios das fases e das atividades de engenharia de requisitos. Referenciais de governança, como COBIT 2019 e ITIL 4, enfatizam criação de valor, gestão de riscos, visibilidade e prestação de contas (ISACA, 2018; AXELOS, 2019).

Apesar dessas contribuições, permanece uma lacuna operacional entre princípios, modelos éticos e mecanismos cotidianos de acompanhamento. Os códigos profissionais indicam deveres, os modelos formulam perguntas e cláusulas e os referenciais de governança

orientam controle e monitoramento, mas esses elementos nem sempre são registrados continuamente com vínculo a projetos, etapas, papéis e evidências. Essa lacuna dificulta transformar reflexão ética em informação recuperável para equipes e partes interessadas.

Como resposta a esse problema, foi desenvolvido o EthicalSoft Compliance, plataforma *web* para configurar questionários éticos, registrar respostas e evidências, calcular indicadores, apresentar painéis, emitir documentos de conformidade declarada e manter registros de emissão. A aplicação utiliza elementos do EDSD e do quadro de Biable, Garcia e Midekso (2023), associados a conceitos de transparência e governança. A inteligência artificial integrada possui função assistiva de interpretação e não participa dos cálculos determinísticos nem define a classificação do projeto.

Diante desse contexto, a questão de pesquisa é formulada da seguinte maneira: *como uma plataforma web pode operacionalizar critérios de autoavaliação ética e produzir registros rastreáveis de conformidade declarada ao longo de projetos de software?*

O objetivo geral consiste em desenvolver uma plataforma *web* que apoie autoavaliação ética, rastreabilidade de decisões e geração de evidências de conformidade declarada em projetos de *software*, incluindo contextos de desenvolvimento assistido por IA. Os objetivos específicos abrangem a especificação dos instrumentos, a implementação dos fluxos, o cálculo dos indicadores, a apresentação das visualizações, a geração dos documentos, o registro das emissões, a integração de notificações, o mapeamento de governança e a avaliação do artefato.

O escopo do trabalho está limitado à construção e à demonstração funcional do artefato. A plataforma não certifica a correção moral de um produto, não substitui julgamento profissional, auditoria externa ou avaliação jurídica e não comprova conformidade com legislação ou norma. Os pesos, faixas e critérios são parâmetros operacionais configuráveis, e não medidas universais de comportamento ético.

A avaliação foi realizada por verificação técnica e por um cenário simulado denominado SelecIA RH. As entradas foram elaboradas para exercitar regras e produzir situações contrastantes. Não houve participantes externos nem implantação em organização real. Assim, os resultados permitem examinar funcionamento e coerência interna, mas não demonstram efetividade organizacional, mudança de comportamento ou generalização estatística.

Este trabalho está organizado em sete capítulos. O Capítulo 2 apresenta os objetivos. O Capítulo 3 reúne a fundamentação teórica. O Capítulo 4 descreve a metodologia, os indicadores e o protocolo. O Capítulo 5 documenta a implementação. O Capítulo 6 apresenta e discute as evidências do cenário simulado. Por fim, o Capítulo 7 retoma os objetivos, as contribuições, as limitações e os trabalhos futuros.

2 OBJETIVOS

Este capítulo apresenta o objetivo geral e os objetivos específicos definidos para orientar o desenvolvimento e a avaliação do EthicalSoft Compliance.

2.1 Objetivo Geral

Desenvolver uma plataforma *web* para apoiar a autoavaliação ética, a rastreabilidade de decisões e a geração de evidências de conformidade declarada em projetos de *software* desenvolvidos sob metodologias ágeis, por meio da operacionalização dos quadros éticos propostos por Biable, Garcia e Midekso (2023) e pelo *framework Ethical-Driven Software Development* (EDSD) de Lurie e Mark (2016), integrados a conceitos de governança de TI inspirados no COBIT 2019 e no ITIL 4, considerando também os riscos éticos associados ao desenvolvimento assistido por IA.

2.2 Objetivos Específicos

OE1: Especificar questionários configuráveis para avaliação ética de projetos de *software*, organizados por etapa, iteração, papel e domínio de governança, incluindo critérios relacionados aos riscos do desenvolvimento assistido por IA;

OE2: Implementar a parametrização dos questionários éticos conforme as necessidades de cada projeto, abrangendo pesos, papéis, etapas, categorias de análise e ativação condicional de perguntas relacionadas ao uso de IA;

OE3: Calcular os indicadores ICP, ISEQ, ISEP e IEM como medidas operacionais de conformidade ética declarada nos níveis de participante, questionário, etapa e projeto;

OE4: Apresentar painéis e visualizações interativas para diferentes perfis, contemplando a evolução do ISEP, a distribuição por faixas de classificação, a análise por etapa e papel e a comparação entre percepções dos participantes;

OE5: Gerar boletins de não conformidade ética quando os resultados ficarem abaixo do limiar configurado, indicando os participantes, as perguntas, os domínios e os pontos que requerem revisão;

OE6: Gerar certificados de ciência ética e conformidade declarada para projetos encerrados que atinjam o limiar configurado, esclarecendo que o documento não substitui auditoria externa, certificação normativa independente ou avaliação jurídica;

OE7: Registrar metadados das emissões de documentos, incluindo código de autenticidade, tipo, escopo avaliado, data e hora, usuário emissor, índice, faixa e *hash* dos dados utilizados;

OE8: Integrar notificações e registros auditáveis associados a eventos do ciclo de avaliação, como atrasos, encerramentos, cálculo de indicadores e emissão de documentos;

OE9: Mapear o alinhamento das funcionalidades da plataforma com objetivos de governança do COBIT 2019 e com práticas de visibilidade e melhoria contínua do ITIL 4;

OE10: Avaliar o funcionamento e a coerência interna da plataforma por meio de uma demonstração estruturada em cenário simulado, verificando os fluxos implementados, os cálculos dos indicadores e os documentos gerados.

3 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos necessários à compreensão do problema de pesquisa e do artefato desenvolvido. A exposição parte da ética profissional e da qualidade em engenharia de *software*, delimita a conformidade ética, examina os modelos adotados, relaciona governança, transparência e dívida e caracteriza os riscos do desenvolvimento assistido por inteligência artificial. Ao final, são apresentados os conceitos de arquitetura e modelagem utilizados e a lacuna que orienta o EthicalSoft Compliance.

3.1 Ética Profissional e Qualidade no Desenvolvimento de *Software*

A ética, segundo VÁZQUEZ (2017), é o campo da filosofia que estuda o comportamento moral dos seres humanos, com foco nas escolhas conscientes e livres que orientam suas ações. No contexto da engenharia de *software*, essa reflexão se desloca para o cotidiano dos profissionais que tomam decisões envolvendo uso de código, privacidade de dados, segurança e integridade de sistemas. A moralidade não se reduz a distinguir "certo" e "errado" de forma simplista, mas exige uma avaliação cuidadosa das consequências das soluções técnicas para usuários, organizações e sociedade.

Com o avanço tecnológico, surgiu a ética computacional, campo que explora dilemas específicos da computação. Questões como privacidade, acesso a dados e propriedade intelectual são centrais para os desenvolvedores de *software*, que devem equilibrar eficiência técnica com considerações morais (JOHNSON, 2008). Uma revisão sistemática conduzida por Vakkuri et al. (2022) sobre o estado da arte da ética em engenharia de *software* confirmou que, apesar do crescente volume de pesquisas, ainda persiste uma lacuna significativa entre os princípios éticos formulados na academia e sua adoção prática nas equipes de desenvolvimento, evidenciando a necessidade de instrumentos mais concretos de operacionalização ética.

Além disso, o desenvolvimento de *software* muitas vezes impõe decisões baseadas em valores pragmáticos, como a escolha de uma solução eficiente versus outra que atenda a padrões éticos mais elevados. Esse tipo de dilema ressalta a complexidade do julgamento ético na engenharia de *software*, em que a eficiência operacional precisa ser equilibrada com o impacto moral das soluções desenvolvidas. Pesquisa conduzida com desenvolvedores de *software* de diferentes perfis demográficos identificou que percepções éticas variam significativamente conforme o nível de senioridade, o contexto organizacional e o setor de atuação, o que reforça a importância de diretrizes estruturadas e não apenas de orientações genéricas (CHEN, 2024).

Apesar da importância das leis e regulamentos, estar em conformidade legal não garante a moralidade das ações (MASIERO, 2008; VÁZQUEZ, 2017). A ética vai além das leis, permitindo que os profissionais ajam de acordo com princípios morais, em vez de apenas cumprir regras. Uma revisão de literatura sobre a presença da ética nas publicações brasileiras de engenharia de *software* identificou que, entre 1529 artigos analisados de 2010 a 2022, apenas 0,4% abordavam aspectos éticos ou morais de forma substantiva, expondo um cenário ainda incipiente no contexto nacional (CARVALHO et al., 2024).

3.1.1 O Código de Ética da Engenharia de *Software* da ACM/ IEEE-CS

O Código de Ética e Prática Profissional da Engenharia de *Software*, desenvolvido conjuntamente pela Association for Computing Machinery (ACM) e pela IEEE Computer Society (IEEE-CS), estabelece diretrizes fundamentais para a conduta ética de profissionais envolvidos no desenvolvimento de *software*, abrangendo engenheiros, gestores, educadores e estudantes. Esse código foi adotado como padrão para orientar a prática e o ensino da engenharia de *software*, promovendo a responsabilidade ética no ciclo de vida do desenvolvimento e nas interações profissionais. Ele é estruturado em oito princípios que tratam das relações dos engenheiros com o público, clientes, empregadores, produtos, julgamento profissional, gestão, colegas e desenvolvimento individual (ACM; IEEE COMPUTER SOCIETY, 1999).

O primeiro princípio enfatiza a atuação segundo o interesse público, destacando a necessidade de priorizar a segurança, a privacidade e o bem-estar dos usuários. Os engenheiros de *software* devem avaliar e aprovar produtos apenas quando estiverem convencidos de que estes atendem aos requisitos de segurança e não geram danos ambientais ou à qualidade de vida das pessoas. O segundo princípio orienta que os engenheiros devem agir no melhor interesse de seus clientes e empregadores, desde que isso esteja alinhado com o bem-estar público, o que inclui garantir a integridade dos produtos e reportar riscos conhecidos que possam comprometer o uso seguro do *software* (ACM; IEEE COMPUTER SOCIETY, 1999).

A responsabilidade pelo produto, prevista no Princípio 3 do Código de Ética e Prática Profissional da Engenharia de *Software*, não é compreendida neste trabalho como a busca abstrata por um “alto padrão”. Ela corresponde ao atendimento de obrigações que podem ser observadas no processo e nos artefatos do projeto. Entre essas obrigações estão definir objetivos apropriados e alcançáveis; utilizar método compatível com o projeto; compreender, documentar e obter a aprovação das especificações; tornar explícitas ao cliente as decisões de compromisso entre qualidade, custo e prazo; apresentar estimativas realistas e suas incertezas; realizar testes, depuração e revisão; documentar problemas relevantes e as soluções adotadas; preservar a privacidade; utilizar dados obtidos de forma lícita e au-

torizada; e manter a integridade dos dados. As cláusulas 3.01 a 3.14 convertem, portanto, o enunciado geral do princípio em responsabilidades verificáveis por registros, aprovações, resultados de testes, revisões e evidências do projeto (ACM; IEEE COMPUTER SOCIETY, 1999).

Nesse sentido, um produto profissionalmente responsável é aquele para o qual a equipe consegue demonstrar, com evidências, que os requisitos foram compreendidos e aprovados, que os critérios de qualidade foram definidos antes da aceitação, que riscos e limitações foram comunicados, que as decisões relevantes possuem justificativa e autoria e que o *software* foi submetido às verificações adequadas ao seu contexto. O padrão considerado não é um nível absoluto ou idêntico para todos os projetos: ele depende do domínio, do risco, das normas aplicáveis e das necessidades dos usuários. O que deve permanecer constante é a obrigação de explicitar os critérios adotados, justificar eventuais desvios e tornar os resultados passíveis de inspeção.

O princípio voltado para a profissão enfatiza a necessidade de os engenheiros contribuírem para a integridade da engenharia de *software*, promovendo um ambiente de trabalho que encoraje a ética e o desenvolvimento contínuo. A interação com colegas deve ser guiada por equidade e apoio mútuo, incentivando a colaboração e a disseminação do conhecimento entre profissionais. O código também aborda o compromisso com o aprendizado contínuo, incentivando os engenheiros a se manterem atualizados sobre práticas e padrões que regem a profissão e a integrarem uma postura ética nas atividades diárias (ACM; IEEE COMPUTER SOCIETY, 1999).

Cada um dos princípios é complementado por cláusulas que especificam obrigações práticas e exemplos de como as diretrizes éticas podem ser aplicadas no contexto profissional. Essas cláusulas funcionam como guias para a tomada de decisões, ajudando os engenheiros a considerar os impactos de suas ações sobre a sociedade e os *stakeholders* envolvidos. O código não pretende funcionar como um algoritmo capaz de gerar decisões éticas automáticas; em vez disso, serve como base para que os profissionais avaliem as complexidades e conflitos éticos que surgem durante o desenvolvimento de *software* (ACM; IEEE COMPUTER SOCIETY, 1999).

3.1.2 Omissão de Práticas de Desenvolvimento de *Software* de Qualidade

A omissão de práticas de desenvolvimento de *software* de qualidade surge como uma preocupação relevante na engenharia de *software*, pois compromete a integridade e a confiabilidade dos produtos gerados. Essa prática está associada a múltiplos fatores, incluindo pressões de mercado, limitações organizacionais e decisões individuais. Esses fatores, isolados ou combinados, levam equipes de desenvolvimento a priorizar a entrega rápida de funcionalidades em detrimento de procedimentos que garantem a qualidade do *software*.

Estudos indicam que essa omissão está associada a problemas recorrentes, gerando defeitos que impactam a operação e a segurança dos sistemas implementados (GHANBARI; VARTIAINEN; SIPONEN, 2018).

3.1.2.1 Fatores que Contribuem para a Omissão de Práticas de Qualidade

A revisão da literatura identifica três categorias principais que influenciam a decisão de omitir práticas de qualidade: fatores de mercado, fatores organizacionais e fatores individuais. Cada uma dessas categorias opera em diferentes níveis contextuais e exerce pressão sobre os processos de desenvolvimento de *software*, comprometendo a execução de práticas estabelecidas.

Pressões de Mercado: As características do ambiente de negócios, como a alta competitividade e as demandas dinâmicas, estimulam as empresas a buscar vantagens estratégicas imediatas, o que muitas vezes leva à omissão de práticas de qualidade. O foco em reduzir o tempo de entrada no mercado e atender rapidamente a mudanças nos requisitos dos clientes influencia negativamente a adesão aos padrões de qualidade. Nesses cenários, a agilidade na entrega é priorizada, o que frequentemente resulta na redução de testes e na supressão de etapas de revisão fundamentais para assegurar a confiabilidade do *software* (GHANBARI; VARTIAINEN; SIPONEN, 2018).

Fatores Organizacionais: Internamente, a escassez de recursos, como tempo, orçamento e pessoal qualificado, limita a capacidade das equipes de desenvolvimento em seguir práticas de qualidade. A relutância da organização em investir em capacitação, automação, revisões sistemáticas e melhoria contínua também contribui para a manutenção de processos frágeis, nos quais a qualidade é tratada como atividade acessória e não como parte estruturante do ciclo de desenvolvimento. O planejamento inadequado, somado a estimativas imprecisas de custo e tempo, frequentemente gera decisões que eliminam atividades críticas, como a análise detalhada de requisitos, a validação contínua e as revisões de código. A literatura revela que, em ambientes onde faltam diretrizes claras e mecanismos de controle, a responsabilidade pela qualidade é deslocada para os desenvolvedores, o que aumenta o risco de práticas ineficazes e omissões que comprometem o produto final. Estudo realizado com equipes ágeis identificou que a segurança psicológica favorece comportamentos relacionados à qualidade de *software*, como admitir erros, comunicar preocupações e tomar iniciativas para prevenir ou corrigir problemas. Esses comportamentos ampliam a aprendizagem da equipe e reduzem a possibilidade de que falhas permaneçam ocultas por receio de exposição ou responsabilização indevida (ALAMI; ZAHEDI; KRANCHER, 2024).

Comportamentos e Fatores Psicológicos: No nível individual, as atitudes, crenças e vieses dos desenvolvedores e gerentes desempenham um papel decisivo. A omissão de práticas de qualidade pode ser influenciada por fatores psicológicos, como a tendência a buscar atalhos ou minimizar riscos percebidos. O viés de confirmação é um exemplo claro:

desenvolvedores tendem a preferir testes que validam suas expectativas, evitando aqueles que poderiam evidenciar falhas no código (GHANBARI; VARTIAINEN; SIPONEN, 2018). Além disso, a falta de políticas de fiscalização eficazes e a ausência de consequências para o descumprimento de padrões técnicos criam um ambiente permissivo à manutenção de práticas inadequadas. Nessa perspectiva, a qualidade não se restringe à fase de testes, mas deve ser entendida como um compromisso distribuído ao longo de todo o processo de desenvolvimento, desde o levantamento de requisitos até a manutenção do *software* (GHANBARI; VARTIAINEN; SIPONEN, 2018; PRESSMAN; MAXIM, 2021; SOMMERVILLE, 2011).

3.1.2.2 Instâncias e Consequências da Omissão de Práticas de Qualidade

As omissões ocorrem com frequência nas atividades de controle de qualidade e testes, além de serem comuns nos estágios de *design*, implementação e documentação técnica. A ausência de testes abrangentes, a postergação de refatorações necessárias e a documentação insuficiente são instâncias recorrentes de abandono de padrões de qualidade. Essas escolhas resultam em aumento da complexidade do *software*, maior ocorrência de defeitos e acúmulo de dívida técnica, o que eleva os custos de manutenção e impacta a estabilidade do sistema a longo prazo.

Embora a omissão de práticas de qualidade possa trazer benefícios imediatos, como redução aparente de custos e agilidade na entrega, esses ganhos são frequentemente superados por custos adicionais relacionados à correção de defeitos não detectados e à necessidade de revisões subsequentes. Levantamento realizado com 97 ferramentas de gestão de dívida técnica mostrou que a maior parte delas, 59%, atua apenas no plano gerencial, sem apoiar efetivamente a identificação e o pagamento da dívida, evidenciando a fragilidade dos mecanismos disponíveis para tratar as consequências dessas omissões (GOMES; LOIOLA; SANTOS, 2024). Além disso, essa dinâmica afeta a cultura organizacional, promovendo um ambiente onde a velocidade é priorizada em detrimento da integridade técnica, o que compromete o desenvolvimento futuro e estabelece precedentes negativos difíceis de reverter.

A revisão aponta que a omissão de práticas de qualidade reflete uma interação de fatores contextuais em diferentes níveis. No plano de mercado, as demandas competitivas incentivam decisões que favorecem a velocidade de entrega em detrimento da conformidade com padrões técnicos. No plano organizacional, as limitações de recursos e a ausência de processos formais robustos facilitam o abandono de práticas recomendadas. No nível individual, os vieses cognitivos e as atitudes dos desenvolvedores contribuem para a perpetuação dessas práticas. O conjunto dessas variáveis cria um cenário em que o compromisso com a qualidade é comprometido, afetando não apenas o projeto em questão, mas também a cultura técnica dentro das organizações.

3.2 Conformidade Ética no Ciclo de Vida do *Software*

Esta seção delimita o conceito de conformidade ética empregado neste trabalho e sua relação com o processo de desenvolvimento de *software*. Inicialmente, apresenta-se uma definição operacional que permita converter princípios éticos em critérios observáveis. Em seguida, discutem-se os principais obstáculos para incorporar esses critérios às atividades de desenvolvimento. Por fim, são examinados mecanismos de avaliação, monitoramento e certificação capazes de registrar evidências e apoiar a prestação de contas. Essa organização distingue a fundamentação conceitual dos procedimentos específicos implementados no EthicalSoft Compliance.

3.2.1 Definição de Conformidade Ética

A conformidade ética no desenvolvimento de *software* refere-se à adesão a princípios e diretrizes morais que garantem que o *software* esteja alinhado com valores sociais como justiça, privacidade e transparência. Ela assegura que os aplicativos de *software* não sejam apenas legalmente compatíveis, mas também eticamente responsáveis, considerando seu impacto sobre os usuários e a sociedade em geral. A conformidade ética abrange aspectos como privacidade de dados, prevenção de viés em algoritmos e acessibilidade para todos os usuários (STAHL et al., 2014).

A conformidade ética desempenha também um papel significativo na melhoria da qualidade e reputação das empresas de *software*. No ambiente digital atual, em que o *software* permeia quase todos os aspectos da vida, e em que ferramentas de inteligência artificial passam a integrar ativamente o processo de desenvolvimento, manter registros estruturados de decisões, justificativas e evidências torna-se componente central da responsabilidade ética, e não apenas um diferencial competitivo.

3.2.2 Desafios na Implementação da Conformidade Ética

Os desafios para implementar a conformidade ética no desenvolvimento de *software* envolvem a evolução das referências normativas, a fragmentação das responsabilidades profissionais e a dificuldade de converter princípios gerais em práticas verificáveis. Antes da apresentação dos obstáculos organizacionais, técnicos e culturais, é necessário delimitar as particularidades do contexto profissional brasileiro.

No Brasil, a conformidade ética no desenvolvimento de *software* precisa ser compreendida a partir de uma estrutura profissional heterogênea. A Constituição Federal assegura o livre exercício de qualquer trabalho, ofício ou profissão, condicionando eventuais restrições às qualificações estabelecidas em lei. A presença de uma ocupação na Classificação Brasileira de Ocupações não representa sua regulamentação, pois a CBO possui finalidade classificatória e administrativa. A regulamentação profissional depende de lei específica

e pode envolver requisitos de formação, registro e fiscalização (BRASIL, 1988; BRASIL. MINISTÉRIO DO TRABALHO E EMPREGO, 2023a).

As ocupações gerais ligadas à computação, como desenvolvimento de sistemas, programação, análise de sistemas e administração de dados, não estão submetidas a um conselho profissional único e obrigatório para todos os trabalhadores da área. O debate sobre sua regulamentação ocorre no Brasil há várias décadas e envolve posições divergentes. De um lado, são apresentados argumentos relacionados à proteção da sociedade, à definição de responsabilidades e ao reconhecimento profissional. De outro, aparecem preocupações com reserva de mercado, restrições ao exercício profissional e valorização do diploma em detrimento da competência demonstrada. Essa tensão contribuiu para que o setor não alcançasse uma normatização profissional uniforme (BRASIL. MINISTÉRIO DO TRABALHO E EMPREGO, 2023b; NUNES; SOUSA, 2017; SILVEIRA; TONINI, 2021).

A situação do título de engenheiro de *software* exige tratamento específico. A Resolução Confea nº 1.156, de 24 de outubro de 2025, atribui ao engenheiro de *software* registrado no Sistema Confea/Crea competências relacionadas a requisitos, sistemas, soluções, evolução e integração de *software*. O próprio normativo determina, entretanto, que essas atribuições não prejudiquem os direitos dos demais profissionais da computação. Assim, a regulamentação do título de engenheiro de *software* não transforma todas as atividades de desenvolvimento de *software* em atribuições exclusivas desse profissional nem institui um conselho obrigatório para todas as ocupações da computação (CONSELHO FEDERAL DE ENGENHARIA E AGRONOMIA, 2025a).

Essa distinção também influencia a aplicabilidade dos códigos de ética. Os engenheiros de *software* registrados no Sistema Confea/Crea estão submetidos ao Código de Ética Profissional adotado pela Resolução Confea nº 1.002/2002 e aos correspondentes procedimentos disciplinares. Para os demais profissionais da computação, códigos como o *ACM Code of Ethics and Professional Conduct* e o *Software Engineering Code of Ethics and Professional Practice*, da ACM e da IEEE Computer Society, constituem referências técnicas e morais relevantes, mas não equivalem, por si próprios, a uma norma estatal obrigatória para todos os trabalhadores brasileiros. Os códigos profissionais podem orientar decisões, explicitar responsabilidades perante usuários, clientes, empregadores e sociedade e servir como parâmetros de profissionalismo, mas precisam ser diferenciados das obrigações jurídicas e contratuais aplicáveis a cada contexto (GOTTERBARN, 2016; ACM; IEEE COMPUTER SOCIETY, 1999; CONSELHO FEDERAL DE ENGENHARIA E AGRONOMIA, 2025b).

A ausência de um código disciplinar único aplicável a todos os profissionais da computação não significa ausência de responsabilidade. O desenvolvimento de *software* continua sujeito às leis relacionadas ao produto e ao domínio de aplicação, aos contratos, às políticas organizacionais, às normas técnicas e às responsabilidades civis, administrativas ou

penais decorrentes de cada situação. O principal desafio é a fragmentação das referências utilizadas para definir a conduta esperada. Uma equipe pode estar simultaneamente submetida a obrigações legais, políticas internas, requisitos contratuais, normas técnicas e códigos profissionais voluntariamente adotados. Por essa razão, a conformidade ética não deve ser reduzida à verificação de filiação a um conselho ou à aceitação formal de um código.

As Diretrizes Curriculares Nacionais da área de Computação procuram enfrentar parcialmente esse problema no âmbito educacional. A Resolução CNE/CES nº 5/2016 determina que os cursos da área proporcionem conhecimento sobre questões sociais, profissionais, legais, éticas, políticas e humanísticas. Para Engenharia de *Software*, a norma estabelece que a construção de sistemas de alta qualidade deve considerar questões éticas, sociais, legais e econômicas e que o egresso deve compreender os impactos diretos e indiretos do *software* sobre as pessoas e a sociedade (BRASIL. CONSELHO NACIONAL DE EDUCAÇÃO. CÂMARA DE EDUCAÇÃO SUPERIOR, 2016).

A formação acadêmica, entretanto, não deve limitar a ética à memorização de regras. Uma formação exclusivamente prescritiva pode não preparar o profissional para reconhecer conflitos entre valores, interesses e consequências em situações novas. A educação ética em computação deve desenvolver capacidade de argumentação, julgamento contextual, responsabilidade moral e consideração pelas pessoas afetadas pelas decisões técnicas (SILVA; CARVALHO, 2024).

A necessidade dessa formação é reforçada pelo cenário da pesquisa brasileira em Engenharia de *Software*. Carvalho et al. (2024) examinaram 1.529 publicações brasileiras produzidas entre 2010 e 2022. Embora 175 trabalhos apresentassem alguma ocorrência relacionada à ética, apenas sete continham tratamento considerado relevante ou aprofundado. Os autores concluem que os aspectos éticos ainda ocupam espaço reduzido na produção brasileira da área, indicando a necessidade de mecanismos mais explícitos para formação, discussão e registro das decisões éticas.

Nesse contexto, o EthicalSoft Compliance pode contribuir ao organizar referências que atualmente se encontram dispersas. A plataforma não deve apresentar um código profissional específico como se fosse juridicamente obrigatório para todos os participantes. Cada critério de avaliação deve identificar sua origem, distinguindo obrigação legal, norma técnica, código profissional, política institucional, requisito contratual ou decisão do projeto. Também deve registrar o responsável pela avaliação, a evidência utilizada, a justificativa da resposta e as alterações realizadas ao longo do desenvolvimento. Dessa maneira, a plataforma não substitui conselhos profissionais, autoridades públicas ou julgamento humano. Sua função é apoiar transparência, rastreabilidade e prestação de contas em um contexto no qual as responsabilidades éticas existem, mas não são organizadas por um único regime profissional.

A fragmentação normativa observada no contexto brasileiro soma-se aos desafios orga-

nizacionais, técnicos e culturais apresentados a seguir. A implementação da conformidade ética depende não apenas da existência de princípios ou códigos, mas também da capacidade de convertê-los em responsabilidades, critérios, evidências e decisões verificáveis durante o ciclo de desenvolvimento.

A implementação de práticas éticas enfrenta dificuldades relacionadas à tradução de princípios gerais em atividades verificáveis, à ausência de mecanismos organizacionais de apoio e à necessidade de incorporar responsabilidades éticas ao processo de engenharia de *software*. Vakkuri et al. (2022) identificam lacunas entre a discussão acadêmica sobre ética e as práticas adotadas pelas empresas, enquanto Mitchell, Balasubramaniam e Fletcher (2022) discutem desafios para incorporar considerações éticas às atividades de engenharia de *software*. Com a difusão de assistentes baseados em modelos de linguagem, torna-se necessário registrar seu uso, manter revisão humana e avaliar os efeitos dessas ferramentas conforme a tarefa e o contexto de desenvolvimento, como discutem Hou et al. (2024) e Mohamed, Assi e Guizani (2025).

3.2.3 Mecanismos de Avaliação e Certificação Ética

Avaliar a conformidade ética no desenvolvimento de *software* requer ferramentas tanto qualitativas quanto quantitativas. Auditorias éticas periódicas permitem que as organizações verifiquem se seus produtos atendem a padrões predeterminados, como privacidade do usuário, inclusão e justiça. Essas auditorias podem ser complementadas por ferramentas voltadas à detecção de vieses em algoritmos ou à verificação de acessibilidade para usuários com deficiências.

Todavia, a literatura aponta a dificuldade de transformar princípios éticos amplos em práticas operacionais verificáveis dentro do ciclo de vida de *software* (LURIE; MARK, 2016; MITCHELL; BALASUBRAMANIAM; FLETCHER, 2022). Por isso, mecanismos de autoavaliação com rastreabilidade, que registrem respostas, justificativas, evidências e trilhas de emissão, oferecem uma abordagem mais factível e defensável do que sistemas que prometem certificação formal ou validação independente. Uma proposta recente de *framework privacy- and security-aware* para IA sistematizou que as dimensões de dados, tecnologia, pessoas e processo precisam ser cobertas por mecanismos de conformidade para que as considerações éticas sejam mantidas ao longo de todo o ciclo de vida do *software*, e não apenas no momento de entrega (KOROBENKO; NIKIFOROVA; SHARMA, 2024).

3.3 Modelos e Quadros de Ética Aplicados ao Desenvolvimen

Os modelos éticos no desenvolvimento de *software* têm ganhado relevância significativa, especialmente à medida que o impacto social da tecnologia se torna cada vez mais evidente. Dois quadros éticos se destacam na literatura recente: o *framework* proposto por Biable, Garcia e Midekso (2023) e o modelo de Desenvolvimento de *Software* Orientado à

Ética (EDSD), proposto por Lurie e Mark (2016). Ambos têm como objetivo a incorporação de práticas éticas no processo de desenvolvimento de *software*, mas se diferenciam em seus enfoques, aplicação prática e metodologia. Uma revisão de práticas adequadas para a engenharia de *software ethics-aware* identificou que *frameworks* que operacionalizam a ética por meio de artefatos concretos, como *checklists*, cartões e histórias de usuário éticas, têm maior probabilidade de adoção do que aqueles que se limitam a princípios genéricos, reforçando o valor das abordagens de Lurie e Mark (2016) e Biable, Garcia e Midekso (2023) (HALME et al., 2022).

3.3.1 Desenvolvimento de *Software* Orientado pela Ética

3.3.1.1 A Crise do *Software* e o Papel da Ética

A "Crise do *Software*", termo cunhado durante uma conferência da OTAN em 1968, descreve o crescente descompasso entre a capacidade de desenvolver *software* de alta qualidade e a rápida evolução do poder computacional. Essa crise se manifesta em projetos que frequentemente extrapolam prazos e orçamentos, ou que simplesmente não atendem aos requisitos dos clientes. Como contextualização histórica, estudos da época indicavam que apenas cerca de um terço dos projetos era concluído com sucesso dentro do prazo e do orçamento (STANDISH GROUP, 2009).

Diversas causas foram identificadas ao longo do tempo: complexidade dos sistemas, expectativas irrealistas, planejamento inadequado, mudanças de requisitos durante o projeto, falhas de comunicação e falta de habilidades (NEIMAT, 2005). A proposta de Lurie e Mark (2016) foi adotar um *framework* ético inspirado nos próprios conceitos da engenharia de *software*, integrando a ética diretamente ao ciclo de vida do desenvolvimento, em vez de mantê-la isolada em códigos genéricos, frequentemente distantes da prática profissional.

3.3.1.2 Integração da Ética no Ciclo de Vida do Desenvolvimento de *Software*

O EDSD propõe que, ao longo de todas as fases do desenvolvimento, da definição de requisitos à fase de testes e manutenção, considerações éticas específicas sejam abordadas de forma prática e integrada. Para facilitar essa integração, o *framework* utiliza cartões de indexação (Figura 1) com perguntas éticas pertinentes a cada etapa do ciclo de vida. Cada cartão contém uma única pergunta do tipo "sim/não", que deve ser avaliada por todas as partes interessadas: clientes, administradores, líderes de equipe, desenvolvedores, *designers* e equipes de controle de qualidade. Esses cartões ajudam a conscientizar os envolvidos sobre os riscos e as responsabilidades éticas associados ao desenvolvimento, promovendo uma postura que prioriza a reflexão ética em conjunto com as decisões técnicas.

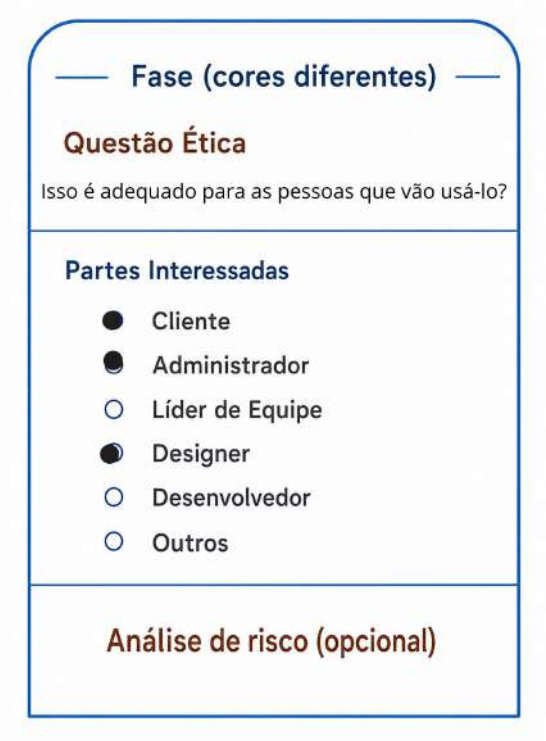

Figura 1 – Exemplo preenchido de cartão de índice para avaliação da adequação da solução aos usuários.

Fonte: Adaptado de Lurie e Mark (2016).

No exemplo, a questão "Isso é adequado para as pessoas que vão usá-lo?" é associada às partes interessadas selecionadas pelos marcadores preenchidos: cliente, administrador e *designer*. Líder de equipe, desenvolvedor e outros participantes permanecem sem seleção, e o campo opcional de análise de risco não foi preenchido. Os marcadores identificam quem deve participar da reflexão; eles não constituem, por si sós, uma resposta afirmativa ou negativa à questão. Na aplicação prática, os participantes selecionados devem discutir a adequação da solução, registrar os critérios considerados e relacionar a decisão às justificativas ou evidências produzidas no projeto.

A principal função dessas fichas é transformar a ética, muitas vezes percebida como abstrata, em uma série de verificações práticas e aplicáveis. O objetivo, segundo os autores, não é fornecer respostas prontas para dilemas éticos, mas sim induzir reflexão nos engenheiros de *software* e nos demais *stakeholders*. Pesquisa recente confirmou essa abordagem ao demonstrar que o uso de histórias de usuário éticas, estruturadas de maneira similar aos cartões do EDSD, facilita a integração de considerações éticas em equipes ágeis, criando artefatos de fácil compreensão e incorporação nas cerimônias de desenvolvimento (HALME et al., 2024).

Lurie e Mark (2016) destacam que o *framework* oferece exemplos claros de perguntas específicas para cada estágio do ciclo de desenvolvimento, conforme apresentado no Quadro 1.

Quadro 1 – Principais perguntas por fase.

Fase	**Pergunta**
Fase de Iniciação	Existem critérios para ajustar a organização para examinar a iniciação?
	Existem critérios para aprovação da iniciação do produto, que devem ser particularmente exigentes para áreas nas quais a organização ou desenvolvedor não é especialista?
	Existem critérios segundo os quais o fornecedor concorda em oferecer um produto que está sendo desenvolvido para um novo campo?
	O fornecedor tem conhecimento suficiente sobre os recursos com os quais deve trabalhar?
	Existe um mecanismo para garantir que o acordo do fornecedor relacionado ao projeto em questão foi obtido somente após determinar se a organização ou desenvolvedor possui a capacidade de executar o projeto?
	Foi determinado quem toma decisões em nome do cliente?
	Foi determinado quem aprova a flexibilidade orçamentária para a gestão do projeto?
	Foi determinado quem é a autoridade que lida com anormalidades no escopo do trabalho?
Fase de Requisitos	Foi determinado quem define os requisitos?
	Existem mecanismos para determinar o caminho de ação quando é descoberto que um ou mais dos requisitos funcionais não podem ser alcançados?
	Existem mecanismos para determinar o caminho de ação quando o cliente insiste?
	Existe um procedimento para definir o curso de ação se, durante a fase de planejamento de requisitos, for descoberto que cumprir a demanda do cliente levará a consequências negativas e possivelmente contraditórias e ilegais?
	Existe um mecanismo para priorizar os requisitos?
Fase de *Design*	As condições foram definidas para determinar o nível de compromisso com o desenvolvimento de uma abordagem de reutilização?
	As condições foram definidas para determinar o nível de compromisso com o desenvolvimento de toda a metodologia de padrões, como o Padrão de *Design* na análise e *design* orientados a objetos (OOAD)?
	Seria fácil orientar-se nos diagramas de *design*?

Continua na próxima página...

Continuação do Quadro 1

Fase	**Pergunta**
	Você acha que deve usar mais diagramas de *design*?
	Alguém que não é especialista consegue se orientar nos diagramas?
	Os diagramas são suficientemente claros?
	Há atenção suficiente aos detalhes?
	O nível de detalhamento é suficiente?
Fase de Desenvolvimento	As condições foram definidas para selecionar a linguagem de codificação?
	Existem condições para integrar desenvolvedores jovens e menos experientes?
	Existem condições para integrar as capacidades avançadas da gestão de código? Tempo? Parâmetros ambientais?
	É fácil se orientar no código que você escreveu?
	Os comentários têm clareza suficiente?
	O código é compatível com os diagramas de *design*?
Fase de Teste e Verificação	O nível de erros permitidos foi determinado antes da transferência do produto para o teste de aceitação?
	Existe um mecanismo que distingue entre um produto de qualidade e um produto correto?
	Existe um mecanismo para determinar se os testes necessários foram realizados?
	Existe um conjunto de testes obrigatórios?
	As métricas para determinar o nível de treinamento necessário para o testador estão definidas?

Fonte: Lurie e Mark (2016).

3.3.2 Ética como Motor da Qualidade em Projetos Ágeis de *Software*

Abdulhalim, Lurie e Mark (2018) investigaram dois times de desenvolvimento que atuavam em projetos semelhantes e relataram associação entre o emprego de ferramentas éticas no planejamento ágil e a melhoria de indicadores de prazo, funcionalidade e custo. O resultado deve ser interpretado no contexto do estudo, não como prova de que a adoção do EDSD produzirá automaticamente os mesmos efeitos em qualquer equipe ou projeto.

No contexto deste trabalho, a contribuição do EDSD consiste em oferecer perguntas que podem ser incorporadas à rotina de avaliação, registradas pela plataforma e discutidas pelas partes envolvidas. O EthicalSoft Compliance não mede diretamente tempo de mercado, defeitos funcionais ou custo total de propriedade. A plataforma registra respos-

tas declaradas, calcula os indicadores definidos no trabalho e apresenta evidências para apoiar a reflexão ética e a governança do projeto, conforme a proposta de Lurie e Mark (2016).

3.3.3 Quadro Ético para Engenharia de Requisitos

O trabalho proposto por Biable, Garcia e Midekso (2023) apresenta-se como um *framework* ético para a engenharia de requisitos, uma fase fundamental no ciclo de desenvolvimento de *software*, caracterizada pela coleta, análise e especificação de requisitos que definirão o comportamento e as funcionalidades do sistema a ser desenvolvido. No entanto, essa etapa também é propensa a problemas éticos que podem comprometer a integridade e a qualidade do *software* final, e questões como falta de comprometimento ético, identificação incorreta de requisitos e práticas proibidas por parte de *stakeholders* são recorrentes.

Este *framework* busca integrar as responsabilidades profissionais dos engenheiros de *software* ao processo de engenharia de requisitos, abordando preocupações específicas identificadas no campo. Seu diferencial está na forma como estrutura essas preocupações por meio de 110 cláusulas adaptadas dos códigos de ética da ACM/IEEE, fornecendo orientações práticas para que engenheiros possam tomar decisões mais responsáveis.

3.3.3.1 Identificação dos Problemas na Engenharia de Requisitos

Uma das principais categorias de preocupações éticas identificadas é a lacuna de conhecimento entre os envolvidos no processo de engenharia de requisitos. Essa lacuna pode ser observada tanto em profissionais técnicos, como engenheiros de *software* e desenvolvedores, quanto em *stakeholders* não técnicos, incluindo clientes, gerentes e patrocinadores do projeto.

Os profissionais técnicos, por vezes, não executam suas funções conforme esperado, negligenciando a aplicação de padrões éticos e de técnicas adequadas de engenharia de requisitos. Essa falta de comprometimento pode se manifestar na incapacidade de identificar e utilizar metodologias, ferramentas e *frameworks* adequados para a especificação dos requisitos, além de falhas na comunicação com os demais *stakeholders*.

Já as partes envolvidas não técnicas frequentemente realizam atividades não permitidas durante o processo de engenharia de requisitos, como esconder informações críticas, relutar em participar ativamente das sessões de elicitação e modificar requisitos de forma inadequada. Esses comportamentos prejudicam a integridade do processo de desenvolvimento e a qualidade dos artefatos produzidos. O Quadro 2 apresenta as cláusulas exemplares propostas por Biable, Garcia e Midekso (2023) para abordar questões éticas relacionadas à categoria "lacuna de conhecimento".

Quadro 2 – Ética para Lacunas de Conhecimento.

Categoria de Ética em Engenharia de Requisitos	**Preocupações**	**Cláusulas**
1. Ética para Lacunas de Conhecimento	1.1 Não perceber o benefício da automação	1.1 Informar adequadamente todos os participantes sobre o propósito do projeto, possíveis conflitos de interesse e benefícios antecipados.
	1.2 Falta de experiência	1.3 Fornecer serviços de engenharia de requisitos com honestidade, reconhecendo limitações pessoais e buscando oportunidades para aprimoramento.
	1.3 Falha na comunicação	1.5 Não discriminar na seleção de *stakeholders* e assegurar o engajamento dos mais relevantes para o processo.
	1.4 Falta de identificação e uso de ferramentas e metodologias	1.8 Atualizar continuamente o conhecimento sobre métodos, ferramentas e padrões relevantes para o desenvolvimento de *software*.
		1.9 Esforçar-se para melhorar continuamente o conhecimento do processo de engenharia de requisitos.

Fonte: Biable, Garcia e Midekso (2023).

3.3.3.2 Problemas de Identificação de Requisitos

A identificação de requisitos é outro ponto crítico, envolvendo a coleta de informações que definem o comportamento esperado do *software*. Problemas frequentes incluem a definição de requisitos incompletos, ambíguos ou irrelevantes, além da omissão de funcionalidades necessárias e da falta de documentação adequada. Esses problemas são amplamente influenciados pela falta de processos estruturados e pela falta de clareza na comunicação entre os envolvidos.

Estudos mostram que a coleta inadequada de requisitos compromete diretamente a qualidade do *software* final, resultando em sistemas que não atendem plenamente às necessidades dos usuários. A omissão de requisitos e o foco exclusivo em requisitos funcionais em detrimento dos não funcionais representam um risco significativo para o sucesso do projeto. Revisão sistemática recente sobre reúso de *software* com conformidade à privacidade

na fase de requisitos identificou que a integração de considerações de privacidade desde os estágios iniciais do desenvolvimento é fundamental para evitar riscos de conformidade que são muito mais custosos de corrigir em fases posteriores (GUBER; REINHARTZ-BERGER, 2024). O Quadro 3 apresenta as cláusulas exemplares elaboradas por Biable, Garcia e Midekso (2023) para abordar questões éticas na identificação de requisitos.

Quadro 3 – Ética para Identificação de Requisitos.

Categoria de Ética em Engenharia de Requisitos	**Preocupações**	**Cláusulas**
2. Ética para Identificação de Requisitos	2.1 Requisitos Incompletos	2.1 Esforçar-se para compreender completamente os requisitos e suas especificações.
	2.2 Requisitos insuficientes	2.2 Garantir uma clara distribuição de responsabilidades na elicitação, verificação e validação dos requisitos.
		2.3 Cada engenheiro de *software* deve ser responsável por detectar, corrigir e relatar erros nos requisitos.
		2.4 Todo engenheiro de *software* é responsável por detectar, corrigir e relatar erros nos requisitos de *software* e documentos relacionados.
	2.3 Omissão de Requisitos	2.5 Garantir a inclusão de todos os requisitos necessários mediante validação adequada.
		2.6 Faça validação e revisão adequada dos requisitos.
	2.4 Falta de Atenção aos Requisitos Não Funcionais	2.7 Seja racional para manter os benefícios sociais fora dos requisitos.
		2.8 Os engenheiros de *software* devem considerar as normas éticas, legais e regulatórias pertinentes.
		2.9 Facilitar a detecção precoce de preocupações éticas durante a fase de *design*.

Continua na próxima página...

Continuação do Quadro 3

Categoria	**Preocupações**	**Cláusulas**
	2.5 Não Identificar os Verdadeiros Requisitos que Devem Ser Feitos	2.10 Faça com que os clientes considerem mais cuidadosamente cada requisito, o que muitas vezes esclarece quaisquer suposições ocultas que possam ter.
		2.11 Faça com que os clientes considerem mais cuidadosamente cada requisito, o que muitas vezes esclarece quaisquer suposições ocultas que possam ter.
		2.12 Faça com que os desenvolvedores tomem decisões corretas de *design* e dediquem níveis apropriados de esforço às diferentes partes do produto de *software*.

Fonte: Biable, Garcia e Midekso (2023).

3.3.3.3 Qualidade dos Requisitos e Impacto Ético

A qualidade dos requisitos é uma dimensão essencial que impacta a segurança, a privacidade e a usabilidade dos sistemas desenvolvidos. O estudo de Biable, Garcia e Midekso (2023) destacou que, em muitos casos, os problemas de qualidade dos requisitos surgem da falta de consideração por aspectos como segurança de dados e proteção da privacidade dos usuários, o que pode resultar na exposição de informações sensíveis e no desenvolvimento de sistemas que não atendem aos padrões mínimos de segurança e confiabilidade.

As cláusulas propostas nessa categoria focam em melhorar a capacidade dos engenheiros de *software* de identificar e implementar requisitos que atendam aos padrões de qualidade esperados, especialmente em termos de segurança e privacidade, assegurando que os dados dos usuários sejam devidamente protegidos ao longo do ciclo de vida do *software*. O Quadro 4 apresenta cláusulas exemplares de Biable, Garcia e Midekso (2023) que orientam os engenheiros a criar requisitos seguros e confiáveis, além de proteger a privacidade e a confidencialidade das informações dos clientes.

Quadro 4 – Ética para Qualidade dos Requisitos.

Categoria de Ética em Engenharia de Requisitos	**Preocupações**	**Cláusulas**
3. Ética para Qualidade dos Requisitos	3.1 Questões de Segurança	3.1 Melhorar a capacidade de criar requisitos seguros e confiáveis que serão fundamentais para o SDLC.
		3.3 Identificar os requisitos de segurança conforme as políticas e procedimentos da empresa.
		3.4 Com base nos requisitos de segurança identificados asseguramos as questões de confidencialidade das palavras-passe e outras informações.
		3.5 Siga cuidadosamente as implementações dos princípios de segurança.
	3.2 Questões de Privacidade	3.6 Proteger a privacidade e a confidencialidade das informações dos clientes é uma responsabilidade ética do engenheiro de *software*.
		3.7 Todas as precauções devem ser tomadas para garantir a proteção de dados dos *stakeholders*.

Fonte: Biable, Garcia e Midekso (2023).

3.3.3.4 Relutância em Fornecer Requisitos

Durante a fase de elicitação de requisitos, é comum que *stakeholders* demonstrem relutância em fornecer informações detalhadas e precisas. Esse comportamento pode ser motivado por preocupações com a complexidade dos requisitos ou resistência a mudanças durante o desenvolvimento do projeto. A falta de colaboração efetiva nesse estágio pode levar à coleta de requisitos imprecisos, resultando em um *software* que não atende plenamente às necessidades dos usuários finais.

O *framework* propõe cláusulas que visam mitigar esses problemas, destacando a importância de envolver *stakeholders* relevantes desde o início do projeto e garantir que as comunicações sejam claras, abertas e direcionadas à compreensão mútua das necessidades e expectativas. O Quadro 5 apresenta as cláusulas exemplares desenvolvidas por Biable,

Garcia e Midekso (2023), que enfatizam a necessidade de participação voluntária e a criação de um ambiente colaborativo e transparente.

Quadro 5 – Ética para Relutância.

Categoria de Ética em Engenharia de Requisitos	**Preocupações**	**Cláusulas**
4. Ética para Relutância	4.1 Relutância em fornecer requisitos	4.1 A participação dos *stakeholders* deve ser voluntária, e todos os envolvidos devem ser informados sobre os objetivos e métodos do projeto.
		4.2 Para obter requisitos, seja uma pessoa autorizada, comunique-se de maneira adequada, seja aberto com os clientes e obtenha o consentimento correspondente.
		4.3 Recusar-se a participar de qualquer atividade em que existam potenciais conflitos de interesse não divulgados.
		4.4 Esteja cuidadosamente ciente dos processos de engenharia de requisitos. Os processos de engenharia de requisitos estão sujeitos a padrões éticos, que promovem e garantem o respeito a todas as partes interessadas e protegem a sua saúde e direitos.
		4.5 Durante os processos de engenharia de requisitos, cada potencial participante deve ser adequadamente informado sobre os objetivos do projeto, métodos utilizados; fontes de dados, quaisquer possíveis conflitos de interesse, os benefícios esperados, os direitos e deveres dos mesmos.

Continua na próxima página...

Continuação do Quadro 5

Categoria	**Preocupações**	**Cláusulas**
	4.2 Requisitos excessivamente complexos	4.6 Todos os participantes têm obrigações éticas para criar um ambiente de trabalho colaborativo e transparente.
		4.7 Os participantes técnicos e não técnicos de um projeto devem ser responsáveis por qualquer falha ocorrida durante os processos de engenharia de requisitos.
	4.3 Resistência a mudanças	4.8 Evitar solicitar aos engenheiros de *software* a realização de atividades que contrariem padrões éticos estabelecidos.
		4.9 Estabelecer a base para acordo entre os clientes e os engenheiros de requisitos sobre o que os artefatos de engenharia de requisitos devem fazer.

Fonte: Biable, Garcia e Midekso (2023).

3.3.3.5 Atividades Proibidas e Não Permitidas

A última categoria aborda práticas proibidas durante a engenharia de requisitos. Entre as principais preocupações estão a desonestidade, a inserção de requisitos maliciosos, o plágio e a recusa em cumprir acordos estabelecidos. Essas práticas violam diretamente os padrões éticos da engenharia de *software* e podem resultar em graves consequências para a confiabilidade e a aceitação do *software* desenvolvido.

O *framework* sugere cláusulas específicas que orientam os engenheiros a evitar comportamentos que comprometam a ética profissional e a integridade do projeto, promovendo um ambiente de trabalho que respeite os direitos e a segurança de todos os *stakeholders*. O Quadro 6 apresenta cláusulas exemplares criadas por Biable, Garcia e Midekso (2023) que visam prevenir práticas desonestas e assegurar a responsabilidade ética em relação a todos os envolvidos.

Quadro 6 – Ética para Atividades Proibidas.

Categoria de Ética em Engenharia de Requisitos	**Preocupações**	**Cláusulas**
5. Ética para Atividades Proibidas	5.1 Desonestidade	5.1 Seja honesto e cumpra com suas responsabilidades profissionais com seu empregador.
		5.2 Tentar garantir a inclusão justa dos benefícios de todas as partes interessadas e da segurança da sociedade.
	5.2 Cuidados e Prevenção	5.3 Evite danos éticos decorrentes de negligência e assegure acordos justos sobre a propriedade dos artefatos de requisitos.
		5.4 Garantir que haja um acordo justo em relação à propriedade de quaisquer artefatos de engenharia de requisitos para os quais um engenheiro de *software* tenha contribuído.
	5.3 Requisitos Maliciosos	5.5 Evitar a incorporação deliberada de requisitos que possam causar danos aos *stakeholders* e à sociedade.
		5.6 Não incorporar conscientemente questões antiéticas nos requisitos.
	5.4 Plágio	5.7 Evitar práticas enganosas ou desonestas e relatar violações éticas significativas detectadas durante o processo.
		5.8 Expressar preocupações às pessoas envolvidas quando forem detectadas violações éticas significativas, a menos que isso seja impossível, contraproducente ou perigoso.
	5.5 Recusa de acordos	5.9 Não só uma remuneração justa, mas também uma distribuição justa dos riscos e benefícios potenciais.

Continua na próxima página...

Continuação do Quadro 6

Categoria	Preocupações	Cláusulas
		5.10 Durante os processos de engenharia de requisitos, cada potencial participante deve ser adequadamente informado sobre os objetivos do projeto, os métodos utilizados, as fontes de dados, quaisquer possíveis conflitos de interesse, os benefícios previstos, os direitos e deveres dos mesmos.
		5.11 Priorize os requisitos com base no seu grau de necessidade.
		5.12 Elaborar os acordos em conjunto com fornecedores e clientes.

Fonte: Biable, Garcia e Midekso (2023).

3.3.3.6 Validação do *Framework* Proposto

O *framework* proposto foi submetido a um rigoroso processo de validação, envolvendo 40 especialistas da indústria de *software* com vasta experiência em desenvolvimento e ética na engenharia de *software*. A validação considerou aspectos de validade de face, conteúdo e construto, além da consistência interna das cláusulas. A avaliação demonstrou que o *framework* é claro, aplicável e ajustado para a prática diária dos engenheiros de *software*.

Os especialistas consideraram que as cláusulas propostas mapeiam adequadamente as preocupações éticas identificadas e oferecem suporte prático para a mitigação de problemas comuns na engenharia de requisitos. A confiabilidade do *framework* foi avaliada como satisfatória, mostrando que as orientações fornecidas são consistentes e relevantes para os desafios enfrentados na prática.

3.3.4 Comparação entre os Modelos Éticos

Embora ambos os modelos compartilhem o objetivo de integrar a ética de maneira mais eficiente no desenvolvimento de *software*, eles se diferenciam em abrangência e foco. O modelo de Biable, Garcia e Midekso (2023) é particularmente adequado para equipes de engenharia de requisitos que enfrentam desafios éticos específicos nessa fase do processo de desenvolvimento. Ao concentrar-se nos aspectos de identificação e qualidade dos requisitos, o *framework* é uma ferramenta poderosa para prevenir problemas que poderiam comprometer a segurança e confiabilidade do *software* logo no início do projeto. Por outro lado, o EDSD de Lurie e Mark (2016) tem aplicação mais ampla, sendo útil para equipes

que precisam de uma integração contínua de questões éticas ao longo de todo o ciclo de vida do produto.

A aplicação do *framework* de Biable, Garcia e Midekso (2023) garante que as questões éticas sejam tratadas de maneira mais direta durante a fase de engenharia de requisitos, enquanto o EDSD se apresenta como um modelo genérico e abrangente. Entretanto, a implementação de cada um traz desafios. O *framework* de Biable, com suas 110 cláusulas detalhadas, pode parecer complexo e difícil de adotar por completo em ambientes de desenvolvimento mais ágeis. Já o EDSD, embora forneça uma metodologia prática, exige uma mudança cultural significativa nas equipes de desenvolvimento para que as perguntas éticas sejam integradas de forma eficaz.

3.4 Governança, Transparência e Dívida Técnica e Ética

A literatura analisada até aqui evidencia que ética, qualidade e engenharia de requisitos não podem ser compreendidas isoladamente da estrutura de governança de TI das organizações. *Frameworks* como COBIT 2019 e ITIL 4 oferecem princípios e objetivos que tratam de criação de valor, gestão de riscos e engajamento de *stakeholders*, e ressaltam a importância da transparência em processos e decisões de TI. No entanto, esses *frameworks* atuam predominantemente em nível organizacional e de gestão de serviços, oferecendo poucas diretrizes específicas para o registro e a visibilização de decisões éticas e técnicas no dia a dia das equipes de desenvolvimento.

3.4.1 Governança de TI e Transparência em Projetos de *Software*

O COBIT 2019 estrutura a governança de TI a partir de objetivos de governança e gestão, incluindo processos, estruturas organizacionais, fluxos de informação, cultura e comportamento, pessoas e competências (ISACA, 2018). O ITIL 4 complementa essa abordagem ao tratar governança, criação de valor, colaboração e melhoria contínua em serviços de TI (AXELOS, 2019). Esses referenciais fornecem fundamentos para transparência, monitoramento e comunicação das decisões em projetos de *software*.

O COBIT 2019 é organizado em seis componentes de governança: processos, estruturas organizacionais, princípios e políticas, fluxos de informação, cultura e comportamento, e pessoas e competências, incorporando explicitamente “cultura, ética e comportamento” como uma dimensão que precisa ser gerida de forma integrada aos demais elementos da governança (ISACA, 2018). Essa arquitetura demonstra que a dimensão ética não é secundária na concepção do *framework*, mas constitui parte integrante do sistema que orienta como decisões são tomadas, comunicadas e monitoradas ao longo do tempo em uma organização de TI.

O ITIL 4, de forma complementar, descreve um *Service Value System* orientado à co-criação de valor em serviços de TI, articulando princípios orientadores, governança, cadeia de valor e um conjunto de práticas de gestão. Guias práticos destacam princípios como "colaborar e promover visibilidade" e "focar no valor", que enfatizam o compartilhamento de informações e a participação de múltiplos *stakeholders* nas decisões, bem como o uso de métricas e painéis para monitorar níveis de serviço, mudanças e relacionamentos. O ITIL 4 introduz ainda o conceito de *value streams*, que representam sequências de atividades que entregam valor a clientes e *stakeholders*, exigindo que cada etapa seja documentada, monitorada e continuamente melhorada.

3.4.2 Dívida Técnica, Dívida Ética e Opacidade nas Decisões

A literatura sobre dívida técnica a descreve como o resultado de decisões de *design* ou implementação que trazem benefícios de curto prazo, como rapidez na entrega, ao custo de aumentar a complexidade e os riscos de manutenção futura. Gibson et al. (2022) argumentam que a dívida técnica não é apenas um problema econômico ou de qualidade, mas também uma questão ética, uma vez que decisões que acumulam dívidas sem a devida avaliação e comunicação podem transferir riscos injustos para usuários, clientes e futuras equipes de manutenção. Ghanbari (2024) reforça essa perspectiva ao caracterizar a dívida técnica como uma "espada de dois gumes" que pode comprometer a responsabilidade moral dos engenheiros de *software*, especialmente quando decisões são tomadas sob pressão sem transparência adequada sobre consequências e *trade-offs*.

Pesquisa sobre dívida técnica autodeclarada em projetos de aprendizado de máquina identificou que boa parte dos débitos nestes sistemas está relacionada a escolhas conscientes de simplificação em modelos e *pipelines* de dados, que raramente são documentadas ou comunicadas a gestores e *stakeholders* não técnicos, gerando opacidade sobre riscos que afetam diretamente a equidade e a confiabilidade dos sistemas em produção (GIBSON et al., 2022). Levantamento do estado atual das ferramentas de gestão de dívida técnica mostrou que a maioria delas opera apenas no nível de código ou de gerenciamento de backlog, sem oferecer mecanismos para rastrear as motivações éticas por trás das decisões que geraram as dívidas, o que dificulta a prestação de contas e a governança dessas escolhas ao longo do ciclo de vida do *software* (GOMES; LOIOLA; SANTOS, 2024).

Em todos esses casos, a literatura converge na ideia de que a opacidade das decisões, a falta de registro, justificativa e comunicação estruturada dos motivos que levam à omissão de práticas de qualidade ou à priorização de certos requisitos, contribui diretamente para o acúmulo de dívidas difíceis de identificar e corrigir no futuro. Princípios de transparência e prestação de contas, discutidos no contexto de sistemas de decisão algorítmica e de governança de IA, reforçam que a visibilidade sobre os processos decisórios não é apenas um requisito regulatório, mas um componente estrutural da responsabilidade ética em engenharia de *software* (HARESAMUDRAM; LARSSON; HEINTZ, 2023).

Esse quadro se agrava com o uso crescente de assistentes de código baseados em LLMs. Quando artefatos são gerados com apoio de IA sem que haja registro da ferramenta utilizada, do escopo da sugestão, da revisão humana realizada e das decisões de aceite ou rejeição, cria-se um tipo específico de opacidade que prejudica a rastreabilidade e a prestação de contas sobre os artefatos entregues. A literatura recente sobre riscos de assistentes de código aponta que a ausência de supervisão humana estruturada, a falta de testes equivalentes ao código escrito por humanos e a exposição de dados sensíveis a ferramentas externas de IA são fontes relevantes de dívida ética que precisam ser tratadas por mecanismos explícitos de registro e auditoria.

3.4.3 Integração entre Governança, Modelos Éticos e Prática de Projeto

Ao confrontar os *frameworks* de governança de TI (COBIT 2019, ITIL 4) com os quadros éticos específicos para desenvolvimento de *software* (EDSD, Biable et al.), emerge uma lacuna importante na literatura e na prática. De um lado, COBIT 2019 e ITIL 4 fornecem princípios e objetivos que enfatizam valor, gestão de riscos, engajamento de *stakeholders* e visibilidade sobre processos e serviços, sugerindo o uso de métricas, relatórios e painéis para apoiar a governança. De outro, EDSD e o *framework* de Biable, Garcia e Midekso (2023) detalham perguntas, cláusulas e estruturas que ajudam equipes a refletir sobre dilemas éticos concretos em cada fase do ciclo de vida e da engenharia de requisitos, promovendo maior conscientização e responsabilidade nas decisões técnicas.

Apesar dessa complementaridade conceitual, faltam instrumentos que conectem diretamente essas duas camadas: mecanismos que traduzam as respostas éticas e as percepções de risco das equipes em informações estruturadas, indicadores e evidências capazes de ser consumidas por gestores, comitês de governança e demais partes interessadas. Em particular, a literatura revisada não apresenta soluções amplamente adotadas que permitam: (i) registrar de forma sistemática as respostas a quadros éticos como EDSD e Biable *et al.*; (ii) consolidar essas respostas em indicadores de conformidade declarada e de dívida técnica e ética por projeto, etapa e papel; (iii) comparar percepções entre equipes técnicas e *stakeholders* de negócio; e (iv) expor essas informações em *dashboards*, relatórios e trilhas de auditoria que dialoguem com objetivos de governança definidos em *frameworks* como COBIT 2019 e ITIL 4. Essa lacuna é ainda mais evidente em projetos que utilizam ferramentas de IA assistiva no desenvolvimento, nos quais a necessidade de rastreabilidade, supervisão humana e registro de evidências é especialmente crítica (MOHAMED; ASSI; GUIZANI, 2025).

Essa lacuna é particularmente crítica em ambientes ágeis, nos quais decisões importantes são tomadas em ciclos curtos, muitas vezes com documentação mínima, o que amplia o risco de acúmulo de dívida técnica e ética sem que haja visibilidade adequada para as

instâncias de governança. Embora estudos sobre ética como motor da qualidade mostrem que a integração de ferramentas éticas em projetos ágeis pode reduzir tempo de mercado, defeitos funcionais e custo de propriedade, eles ainda não propõem, de forma detalhada, como os resultados dessas práticas podem ser integrados em sistemas de governança e indicadores corporativos. Diante disso, justifica-se a concepção de soluções que atuem especificamente como ponte entre quadros éticos aplicados no nível de projeto e *frameworks* de governança de TI, fornecendo uma infraestrutura de transparência, rastreabilidade e monitoramento contínuo da dívida técnica e ética em projetos de *software*.

A operacionalização das obrigações profissionais pode ser apoiada por práticas organizacionais que definam responsabilidades e mantenham a participação do cliente ou de seu representante autorizado ao longo do desenvolvimento. Quando o Scrum é efetivamente adotado, a responsabilidade pela maximização do valor e pela gestão do *Product Backlog* pertence ao *Product Owner*; o cliente ou seu representante somente exerce essa função quando ela lhe é formalmente atribuída (SCHWABER; SUTHERLAND, 2020). Independentemente do método, devem permanecer identificados os responsáveis por esclarecer necessidades, ordenar prioridades, avaliar resultados e decidir sobre a aceitação, sem transferir ao cliente a responsabilidade técnica coletiva da equipe. Essas práticas constituem referências de gestão às quais uma plataforma de avaliação ética pode oferecer apoio informacional; não são, por si mesmas, funcionalidades do EthicalSoft Compliance.

Nesse contexto, a visibilidade compartilhada constitui também uma condição ética. Informações fragmentadas podem ocultar riscos, impedir questionamentos e concentrar conhecimento em poucos participantes. Uma base comum de responsabilidades, critérios, resultados, riscos, justificativas e evidências permite que os participantes autorizados identifiquem divergências, comuniquem problemas e prestem contas sobre o que foi decidido, com quais elementos e por quem. Essa compreensão converge com a transparência dos artefatos prevista no Scrum e com as cláusulas 2.06 e 2.07 do Código ACM/IEEE-CS, que orientam o registro, a documentação e a comunicação tempestiva de problemas relevantes ao cliente ou empregador (SCHWABER; SUTHERLAND, 2020; ACM; IEEE COMPUTER SOCIETY, 1999).

O EthicalSoft Compliance procura apoiar essa responsabilidade compartilhada ao reunir registros de avaliação ética que poderiam permanecer dispersos entre pessoas, etapas e documentos. Sua contribuição não consiste em garantir, por si só, determinado padrão profissional, mas em oferecer meios para registrar, compartilhar, acompanhar e discutir as evidências relacionadas aos critérios definidos para o projeto. As informações efetivamente disponibilizadas e os respectivos controles são demonstrados no Capítulo 5.

Visibilidade compartilhada não significa acesso irrestrito a todos os dados. Informações pessoais, respostas confidenciais, credenciais e conteúdos protegidos devem observar privacidade, finalidade, necessidade e controle de acesso. O objetivo é assegurar que os participantes autorizados disponham das informações necessárias às suas responsabilidades,

preservando as restrições éticas e legais aplicáveis.

3.5 Desenvolvimento de *Software* Assistido por Inteligência Artificial

Modelos de linguagem podem apoiar atividades de engenharia de *software*, como explicação, geração e revisão de artefatos textuais ou de código. Entretanto, a utilidade desses sistemas depende do contexto e não elimina riscos de respostas incorretas, reprodução de vieses, exposição de informações e dificuldade de rastrear a origem de uma sugestão. Revisões da literatura também apontam que a avaliação dessas ferramentas varia entre tarefas, conjuntos de dados e critérios, o que desaconselha tratá-las como fonte autônoma de decisão (HOU et al., 2024; MOHAMED; ASSI; GUIZANI, 2025).

Do ponto de vista da governança ética, a utilização de assistentes de IA pode introduzir incorporação de código sem revisão humana adequada, ausência de rastreabilidade dos artefatos gerados, exposição de dados sensíveis a serviços externos, dependência excessiva de sugestões automatizadas e dificuldade de responsabilização. A rastreabilidade entre solicitação, resposta gerada e artefato resultante torna-se especialmente relevante quando o conteúdo produzido influencia decisões de desenvolvimento (WANG et al., 2025).

Esses riscos se conectam à dívida ética e à opacidade das decisões. Quando não há registro da ferramenta utilizada, do escopo da sugestão, da revisão humana e das razões para aceite ou rejeição, torna-se difícil atribuir responsabilidade e reconstruir a origem de um artefato. Por essa razão, o uso assistivo adotado neste trabalho permanece vinculado às funções efetivamente implementadas e não é apresentado como mecanismo autônomo de julgamento.

No EthicalSoft Compliance, a inteligência artificial possui função assistiva e interpretativa. Ela pode gerar explicações contextuais a partir de dados já calculados pela aplicação, mas não define pesos, não altera respostas e não participa do cálculo determinístico dos indicadores. Quando a chamada externa falha, o resultado numérico permanece disponível. Essa separação permite distinguir evidência registrada, cálculo reproduzível e texto explicativo. A própria utilização de IA no processo de desenvolvimento também pode ser objeto dos questionários éticos da plataforma.

3.6 Conceitos de Arquitetura e Modelagem para o Artefato

A arquitetura de *software* organiza responsabilidades, dependências e mecanismos de comunicação entre os componentes de um sistema. Na arquitetura hexagonal, também denominada *Ports and Adapters*, o núcleo da aplicação é isolado de tecnologias externas

por meio de portas que expressam contratos e adaptadores que realizam integrações. Essa separação busca reduzir o acoplamento entre regras de negócio, interface, persistência e serviços externos (COCKBURN, 2005; MARTIN, 2019).

A modelagem por casos de uso descreve interações entre atores externos e o sistema com a finalidade de alcançar objetivos identificáveis. Esse recurso auxilia a delimitação do comportamento esperado e a comunicação de requisitos funcionais, mas não substitui o modelo de domínio, o modelo de dados ou a descrição da arquitetura. Neste trabalho, os casos de uso são empregados para representar responsabilidades e fluxos funcionais centrais. A definição conceitual é apresentada nesta fundamentação, enquanto o levantamento dos casos, os diagramas e sua correspondência com a implementação são documentados nos capítulos metodológico e de desenvolvimento e nos apêndices (PRESSMAN; MAXIM, 2021).

O modelo de domínio representa conceitos, atributos e relações relevantes do problema, sem se confundir com as classes concretas da implementação ou com o esquema físico do banco de dados. O Modelo Entidade-Relacionamento, por sua vez, descreve entidades persistidas, chaves e relacionamentos do armazenamento relacional. A distinção permite examinar separadamente a compreensão conceitual do domínio e as decisões de persistência adotadas no artefato (LARMAN, 2007).

A persistência poliglota emprega mecanismos de armazenamento distintos para grupos de dados com características e padrões de acesso diferentes. Essa abordagem não pressupõe a superioridade geral de um paradigma: a escolha depende da estrutura das informações, das consultas, das garantias de consistência e dos custos operacionais introduzidos pela arquitetura (LANDUYT et al., 2023). Em bancos documentais, a flexibilidade estrutural também requer controle da evolução dos documentos, pois esquemas mantidos apenas de forma implícita no código podem produzir versões incompatíveis e dificultar a manutenção (KLETTKE et al., 2016; MEURICE; CLEVE, 2017). A aplicação concreta desses critérios ao EthicalSoft Compliance é apresentada no Capítulo 5.

3.7 Síntese da Literatura e Lacuna de Pesquisa

A literatura examinada indica que princípios éticos, práticas de qualidade e mecanismos de governança costumam ser tratados em níveis distintos. Os códigos profissionais estabelecem deveres; os modelos éticos orientam perguntas e decisões; e as estruturas de governança enfatizam controle, rastreabilidade e prestação de contas. Permanece, contudo, uma dificuldade operacional para registrar essas dimensões de forma contínua e relacioná-las às etapas, aos papéis e às evidências de um projeto de *software*.

O EthicalSoft Compliance procura atuar nesse intervalo ao configurar questionários por fase e papel, calcular indicadores, registrar respostas e eventos, apresentar painéis, emitir documentos e oferecer apoio interpretativo. A plataforma não certifica a correção

moral de um produto, não substitui julgamento profissional e não demonstra conformidade legal. Seu escopo é fornecer um mecanismo técnico de acompanhamento, documentação e sinalização de possíveis não conformidades dentro do modelo configurado.

Permanecem fora do escopo a validação normativa independente, a auditoria jurídica, a avaliação causal de mudanças de comportamento e a comprovação de efetividade em organizações reais. Essas delimitações orientam a metodologia de demonstração e a interpretação dos resultados apresentados nos capítulos seguintes.

4 METODOLOGIA

Este capítulo descreve o percurso metodológico adotado na pesquisa. Inicialmente, apresenta-se a caracterização do estudo e as tecnologias empregadas. Em seguida, são descritas as etapas de concepção e construção do artefato, o modelo de avaliação, o protocolo de demonstração, as ameaças à validade e as condições de reprodução da versão examinada.

4.1 Caracterização da Pesquisa

Esta pesquisa é de natureza aplicada, pois busca produzir conhecimento orientado à solução de um problema concreto por meio da construção de um artefato de *software*. Quanto aos objetivos, possui caráter exploratório e descritivo. É exploratória ao organizar e operacionalizar conceitos de ética, qualidade e governança ainda dispersos no contexto estudado, e descritiva ao documentar requisitos, arquitetura, indicadores, comportamento e resultados do artefato. Essas classificações seguem a distinção entre finalidade prática da pesquisa, aproximação exploratória do problema e descrição sistemática de suas características (GIL, 2022).

A abordagem combina elementos qualitativos e quantitativos. A dimensão qualitativa está presente na interpretação da literatura, na definição dos requisitos e na análise dos critérios éticos. A dimensão quantitativa corresponde ao cálculo determinístico dos indicadores e à apresentação dos valores produzidos no cenário simulado. Esses valores não formam uma amostra destinada à inferência estatística.

O procedimento principal é a *Design Science Research* (DSR), por envolver identificação do problema, definição dos objetivos da solução, projeto e desenvolvimento do artefato, demonstração, avaliação e comunicação dos resultados (PEFFERS et al., 2007). A DSR é adequada quando o conhecimento é produzido e examinado por meio de um artefato construído para atuar em determinado contexto. A avaliação deve explicitar o artefato, o ambiente, os critérios e as evidências, pois diferentes métodos avaliam propriedades distintas e sustentam conclusões de alcance diferente (PRAT; COMYN-WATTIAU; AKOKA, 2015).

Neste trabalho, o artefato é uma plataforma *web* e sua avaliação ocorre por verificação técnica e demonstração estruturada em cenário simulado. Portanto, os resultados permitem analisar o funcionamento e a coerência interna da plataforma diante das entradas configuradas, mas não comprovam sua efetividade em uso organizacional real.

4.2 Tecnologias Adotadas

As tecnologias foram selecionadas de acordo com os requisitos do artefato, a separação entre interface, regras de negócio e infraestrutura e a necessidade de reproduzir o ambiente de execução. A adoção de uma tecnologia não é apresentada como superioridade universal, mas como uma decisão de projeto sujeita a benefícios e custos.

O *back-end* foi desenvolvido em Java 21 com Spring Boot 3.4.2 e Spring Web MVC. O Java 21 corresponde à versão utilizada pelo projeto, enquanto o Spring Boot fornece mecanismos de inicialização, injeção de dependências, validação e acesso a dados. O Spring Web MVC foi empregado para expor os serviços por meio de uma API REST (ORACLE, 2023; SPRING, 2025).

A interface foi implementada com Angular 22 e Angular Material. O Angular fornece recursos para roteamento, formulários, cliente HTTP, injeção de dependências e renderização no servidor. O Apache ECharts, integrado por meio do *ngx-echarts*, foi empregado na construção dos gráficos e painéis analíticos (ANGULAR, 2026b; ANGULAR, 2026a).

O PostgreSQL 16 foi adotado para o núcleo relacional da aplicação, incluindo dados que exigem transações, restrições e relações consistentes. O Flyway registra as migrações e torna a evolução do esquema reproduzível. O MongoDB 8 foi empregado para documentos auxiliares com estrutura variável, como respostas, modelos, configurações, registros documentais e *caches*. O uso conjunto caracteriza persistência poliglota e introduz custos adicionais de implantação, observabilidade, consistência entre armazenamentos, *backup* e recuperação. A adequação dessa abordagem depende da carga e do modelo de acesso, não de uma vantagem geral de um paradigma de persistência sobre outro (LANDUYT et al., 2023). As decisões aplicadas ao artefato são retomadas no Capítulo 5 (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2023; MONGODB, 2026b).

A integração com modelos de linguagem foi implementada com Spring AI e o provedor Groq. A abstração de modelo desacopla o serviço de aplicação do provedor e permite aplicar tratamento de falhas. A inteligência artificial possui função exclusivamente assistiva e não interfere no cálculo dos indicadores (SPRING AI, 2025a; SPRING AI, 2025b).

FreeMarker e OpenHTMLtoPDF 1.0.10 foram utilizados na geração de documentos em PDF. Os modelos FreeMarker separam o conteúdo dos documentos de sua renderização, enquanto o OpenHTMLtoPDF converte HTML e CSS em PDF sem exigir navegador em execução (OPENHTMLTOPDF, 2024).

JUnit 5 e Mockito foram utilizados na suíte automatizada de testes unitários. O JaCoCo foi empregado na medição de cobertura e o SonarQube Cloud, na análise estática. Esses mecanismos produzem evidências sobre os fluxos examinados, mas não substituem testes de integração nem testes abrangentes de segurança, desempenho, usabilidade ou aceitação (JUNIT, 2026; JACOCO, 2026; OFFUTT, 1991; PEZZÈ; RUBINOV; WUTTKE, 2013).

Docker Compose, Nginx e Ngrok foram empregados na execução e disponibilização

do ambiente. O Docker Compose descreve serviços, redes e volumes; o Nginx entrega os artefatos da interface e atua como *proxy*; e o Ngrok foi utilizado quando houve necessidade de acesso temporário ao ambiente (DOCKER, 2026). Apache Commons CSV e Gson foram usados, respectivamente, na geração dos formatos CSV e JSON empregados nas exportações.

4.3 Etapas da Pesquisa

A pesquisa foi conduzida como um processo único, iniciado pela delimitação do problema e encerrado pela comunicação dos resultados. A Figura 2 representa o fluxo integral adotado.

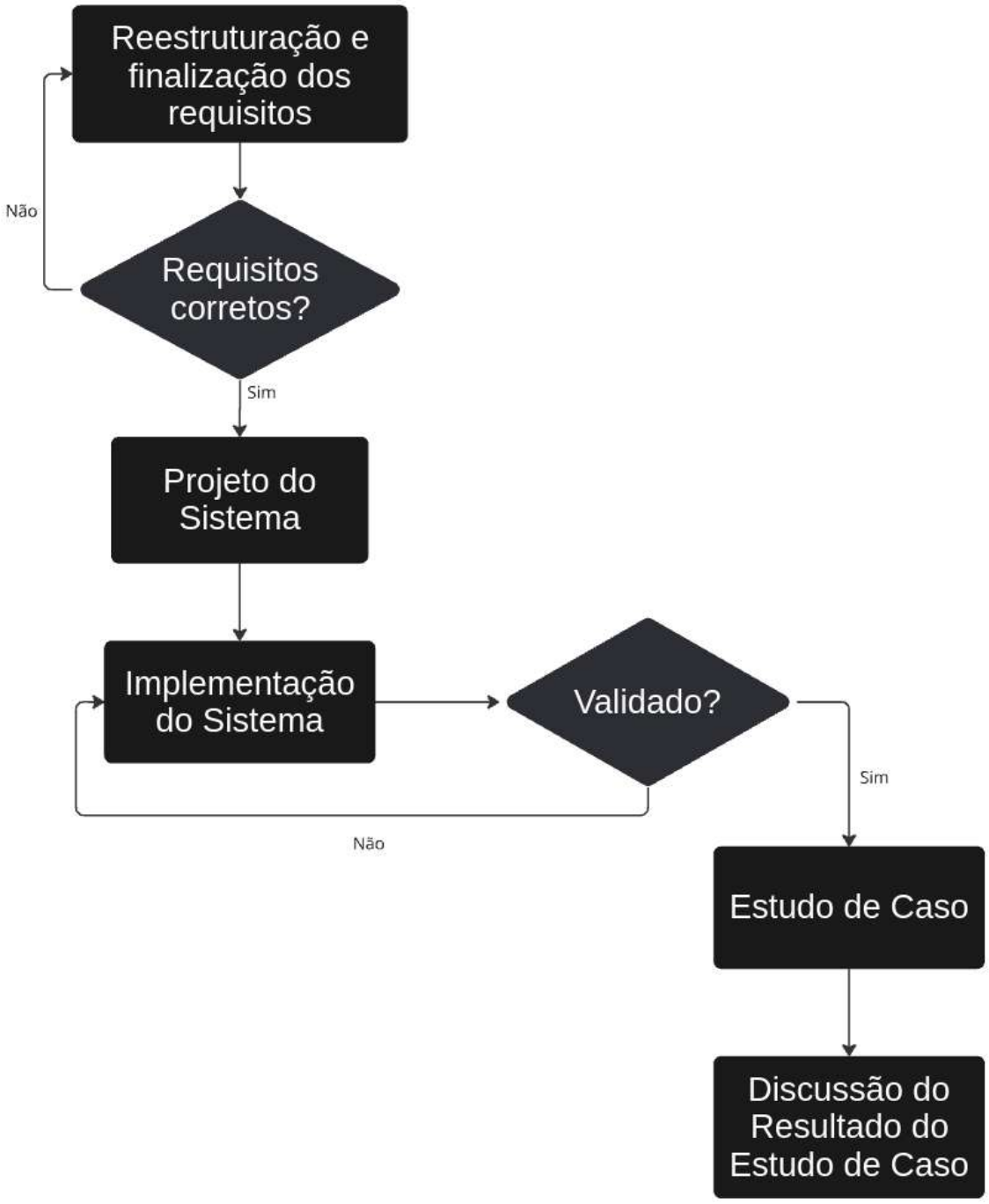

Figura 2 – Fluxo integral da pesquisa.

Fonte: O autor.

4.3.1 Identificação do Problema e Revisão da Literatura

O processo começou pela delimitação do problema e pela revisão da literatura sobre ética, qualidade, governança e desenvolvimento de *software*. Foram examinados códigos profissionais, modelos voltados à incorporação de questões éticas no ciclo de vida, mecanismos de governança e riscos associados ao desenvolvimento assistido por IA. Essa etapa permitiu identificar a dificuldade de converter princípios e perguntas éticas em re-

gistros contínuos, indicadores e evidências relacionadas às etapas e aos participantes de um projeto.

4.3.2 Elicitação de Requisitos e Operacionalização dos Modelos Éticos

A partir da literatura, foram identificadas capacidades necessárias para transformar princípios e perguntas éticas em critérios observáveis. Essas capacidades deram origem aos requisitos funcionais e não funcionais, aos casos de uso, às regras de negócio, aos papéis e às regras de configuração do artefato. As especificações completas são apresentadas nos Apêndices A e B.

Os elementos selecionados do EDSD e do quadro de Biable, Garcia e Midekso (2023) foram operacionalizados em fases, questionários, papéis, pesos, classificações de perguntas e critérios de resposta. O EDSD contribuiu com a distribuição das questões pelo ciclo de vida e pelos participantes, enquanto o quadro de Biable, Garcia e Midekso (2023) contribuiu com categorias e cláusulas direcionadas à engenharia de requisitos. A plataforma não reproduz integralmente os dois modelos nem atribui validade universal às configurações resultantes; ela fornece mecanismos para que os instrumentos sejam configurados de acordo com o contexto do projeto.

4.3.3 Projeto e Prototipação do Artefato

Na etapa de projeto, foram definidos a arquitetura, o modelo de domínio, os casos de uso, a persistência, as interfaces e as integrações externas. Foram elaborados esboços internos de baixa fidelidade para orientar os fluxos de autenticação, configuração de projetos, resposta a questionários e consulta aos painéis. Esses esboços não foram submetidos a validação formal e não foram preservados como evidência de pesquisa; por isso, o trabalho avalia somente as interfaces efetivamente implementadas e registradas no Capítulo 5.

4.3.4 Implementação e Verificação Técnica

A implementação produziu a interface *web*, a API, os mecanismos de persistência, os cálculos, os painéis, os documentos e a integração assistiva de IA. A verificação técnica incluiu uma suíte automatizada de testes unitários, executados com JUnit 5 e Mockito, além de análise estática com SonarQube Cloud, medição de cobertura com JaCoCo e execução de vinte testes funcionais manuais. A inspeção das classes confirmou que os componentes sob teste foram executados em escopo isolado, com substituição de suas dependências; não foram identificados testes automatizados que inicializassem a colaboração entre componentes ou recursos reais de infraestrutura (OFFUTT, 1991; PEZZÈ; RUBINOV; WUTTKE, 2013). Os casos, o ambiente, a classificação, os resultados e as

evidências disponíveis são apresentados no Apêndice D, com distinção entre procedimento, evidência, resultado e limitação (RUNESON; HÖST, 2009).

4.4 Modelo de Avaliação e Indicadores

O modelo de avaliação organiza os resultados nos níveis de participante, questionário, etapa e projeto. Os pesos atribuídos aos representantes e aos questionários são parâmetros configuráveis. As equações desta seção definem o cálculo reproduzível dos indicadores; o Capítulo 6 apresenta somente os valores obtidos no cenário simulado.

4.4.1 Indicador de Conformidade do Participante (ICP)

O Indicador de Conformidade do Participante representa a proporção de respostas conformes de um representante em relação às perguntas elegíveis para os papéis exercidos no questionário. Perguntas não vinculadas aos papéis do representante não compõem o denominador. O cálculo é formalizado pela Equação 4.1.

$$\mathrm{ICP}_r = \frac{Q_{\mathrm{conformes},r}}{Q_{\mathrm{elegiveis},r}} \times 100 \tag{4.1}$$

Em que ICP_r representa o indicador do representante r; $Q_{\mathrm{conformes},r}$ corresponde ao total de respostas conformes; e $Q_{\mathrm{elegiveis},r}$ corresponde ao total de perguntas elegíveis para esse representante. A Equação 4.1 somente é aplicada quando $Q_{\mathrm{elegiveis},r} > 0$; representantes sem perguntas elegíveis não integram esse cálculo.

4.4.2 Indicador Sistêmico Ético do Questionário (ISEQ)

O Indicador Sistêmico Ético do Questionário consolida os ICPs dos representantes elegíveis mediante média ponderada pelo peso atribuído a cada participante. O cálculo é formalizado pela Equação 4.2.

$$\mathrm{ISEQ}_q = \frac{\sum_{r=1}^{n} (\mathrm{ICP}_r \times P_r)}{\sum_{r=1}^{n} P_r} \tag{4.2}$$

Em que ISEQ_q representa o indicador do questionário q; ICP_r é o indicador do representante r; P_r é o peso desse representante; e n é a quantidade de representantes elegíveis considerados. Para que a Equação 4.2 seja definida, exige-se $n \geq 1$ e $\sum_{r=1}^{n} P_r > 0$.

4.4.3 Indicador Sistêmico Ético do Projeto (ISEP)

O Indicador Sistêmico Ético do Projeto consolida os ISEQs dos questionários concluídos, considerando o peso atribuído a cada questionário. O cálculo é formalizado pela Equação 4.3.

$$\text{ISEP} = \frac{\sum_{q=1}^{m}(\text{ISEQ}_q \times P_q)}{\sum_{q=1}^{m} P_q} \tag{4.3}$$

Em que ISEP representa o indicador do projeto; ISEQ_q é o indicador do questionário q; P_q é o peso do questionário; e m corresponde à quantidade de questionários concluídos considerados. Para que a Equação 4.3 seja definida, exige-se $m \geq 1$ e $\sum_{q=1}^{m} P_q > 0$.

4.4.4 Indicador de Ética por Etapa (IEM)

O Indicador de Ética por Etapa representa a proporção de respostas conformes de um representante entre as perguntas elegíveis vinculadas a uma etapa específica. Uma pergunta associada a mais de uma etapa é contabilizada em cada uma delas. O cálculo é formalizado pela Equação 4.4.

$$\text{IEM}_{r,e} = \frac{Q_{\text{conformes},r,e}}{Q_{\text{elegiveis},r,e}} \times 100 \tag{4.4}$$

Em que $\text{IEM}_{r,e}$ representa o indicador do representante r na etapa e; $Q_{\text{conformes},r,e}$ é o total de respostas conformes; e $Q_{\text{elegiveis},r,e}$ é o total de perguntas elegíveis vinculadas à etapa. A Equação 4.4 somente é aplicada quando $Q_{\text{elegiveis},r,e} > 0$.

4.4.5 Pesos, Faixas e Limiar Operacional

Os pesos atribuídos aos representantes e aos questionários definem sua influência nos indicadores ponderados. Esses parâmetros são configurados pelo administrador e devem ser acompanhados de justificativa compatível com o contexto do projeto. Os resultados de ICP, ISEQ, ISEP e IEM são classificados de acordo com o intervalo percentual em que se situam, conforme as faixas apresentadas no Quadro 7.

Quadro 7 – Faixas de classificação dos indicadores.

Faixa	**Intervalo**	**Interpretação operacional**
Faixa A	90% a 100%	Resultado situado no intervalo mais elevado do modelo.
Faixa B	75% a 89,9%	Resultado igual ou superior ao limiar adotado no cenário.
Faixa C	60% a 74,9%	Resultado inferior ao limiar adotado no cenário.
Faixa D	45% a 59,9%	Resultado que demanda revisão das não conformidades registradas.
Faixa E	Abaixo de 45%	Resultado situado no intervalo mais baixo do modelo.

Fonte: O autor.

O valor de 75% é adotado como limiar operacional configurável para o cenário de demonstração e corresponde ao limite inferior da faixa B. Ele não representa ponto de corte universal nem medida cientificamente validada de comportamento ético. Sua função é permitir que o sistema classifique resultados e acione regras de acompanhamento de forma transparente. Organizações que adotem o artefato devem revisar esse parâmetro conforme contexto, risco, política interna e evidências próprias, registrando a justificativa da configuração utilizada.

4.5 Protocolo de Demonstração e Avaliação

O cenário SelecIA RH representa uma plataforma fictícia de recrutamento apoiada por inteligência artificial. Foram configuradas quatro iterações e cinco papéis simulados: gerente de projeto, analista de qualidade, desenvolvedor, *stakeholder* e cliente. Cada perfil recebeu peso previamente definido e respondeu aos instrumentos correspondentes às fases e aos papéis configurados. As respostas foram elaboradas para produzir situações contrastantes, incluindo uma primeira iteração com não conformidades e iterações posteriores com melhoria dos indicadores. Os dados são sintéticos e não foram coletados de participantes ou de uma organização real.

A construção dos dados foi manual e conduzida pelo próprio pesquisador. Os temas das perguntas derivam dos elementos do EDSD e do quadro de Biable, Garcia e Midekso (2023) operacionalizados no instrumento; os cinco papéis foram escolhidos para representar perspectivas gerenciais, de qualidade, de desenvolvimento, de parte interessada e de cliente. As respostas foram distribuídas segundo as regras de elegibilidade da aplicação e configuradas para exercitar casos contrastantes de conformidade, divergência entre participantes e acionamento das faixas. Não foi utilizada base empírica, amostra de organização real nem distribuição probabilística. Os pesos foram definidos exclusivamente para exercitar os cálculos ponderados e não representam uma hierarquia profissional validada. A matriz de configuração e os resultados esperados são apresentados no Apêndice E.

A execução seguiu o seguinte roteiro: criação do projeto; configuração de fases, iterações, papéis e pesos; associação dos questionários; registro das respostas; encerramento dos questionários; cálculo dos indicadores; inspeção dos painéis; emissão dos documentos; e verificação dos registros produzidos. Foram observados os valores de 5,36%, 92,14%, 80,00% e 100,00% nas quatro iterações, com ISEP final de 75,91%, classificado na faixa B, e desvio-padrão da equipe de 37,64%.

Os critérios observados foram: execução dos fluxos previstos; correspondência entre entradas configuradas e resultados calculados; geração das visualizações; emissão dos documentos de acordo com o limiar; e preservação dos registros necessários à rastreabilidade. Esses valores demonstram que o sistema processou as entradas e aplicou as regras configuradas. Eles não constituem evidência de mudança de comportamento, redução de

incidentes éticos ou aceitação por usuários reais. A especificação completa do cenário é apresentada no apêndice correspondente.

4.6 Ameaças à Validade e Limitações

4.6.1 Validade de Construção

Os indicadores representam conformidade declarada dentro do modelo configurado e não medem diretamente a correção moral de um produto. Os pesos, as perguntas e o limiar operacional influenciam os valores obtidos. A correspondência entre os construtos teóricos e os indicadores foi sustentada pelos modelos examinados, mas não foi validada por especialistas externos nesta etapa.

4.6.2 Validade Interna

As entradas e as respostas do cenário foram elaboradas para exercitar comportamentos específicos da aplicação. Dessa forma, a progressão observada decorre das condições definidas na simulação e não permite inferir relação causal entre uso da plataforma e melhoria ética de uma equipe.

4.6.3 Validade Externa

A utilização de um único cenário fictício, sem organização ou participantes externos, limita a generalização. Os resultados não demonstram efetividade em diferentes setores, metodologias, tamanhos de equipe ou níveis de risco.

4.6.4 Confiabilidade

A versão, o *commit*, as entradas do cenário, as regras de cálculo e os procedimentos de execução foram registrados para favorecer a repetição da demonstração. Entretanto, integrações externas, configurações de ambiente e respostas de modelos de linguagem podem variar. A elaboração dos dados pelo próprio pesquisador também introduz risco de viés na seleção das situações demonstradas.

Não foram executados testes automatizados de integração nem suítes abrangentes de contrato de API, automação de interface de ponta a ponta, avaliação formal de usabilidade, ensaios de desempenho sob carga, testes de penetração ou aceitação com usuários externos. Assim, os resultados demonstram o comportamento isolado dos componentes e os fluxos funcionais manuais examinados, mas não autorizam concluir que o sistema esteja integralmente coberto, livre de defeitos, seguro sob ataque ou validado em contexto organizacional real.

4.7 Disponibilidade e Reprodutibilidade do Artefato

O código-fonte e as instruções de execução estão disponíveis no repositório EthicalSoft Compliance, no endereço <https://github.com/yvesls/ethicalsoft-compliance>. A versão examinada foi identificada como `0.0.1-SNAPSHOT`, na *branch* `main`, com referência ao *commit* `4175115d`. Os requisitos de execução, as variáveis de ambiente, o contrato resumido da API e os procedimentos de inicialização são apresentados no apêndice de instalação e reprodução. Credenciais, *tokens* e chaves de API não integram o repositório e devem ser configurados externamente.

5 DESENVOLVIMENTO DO ARTEFATO

Este capítulo documenta a aplicação dos conceitos apresentados na fundamentação e das decisões definidas na metodologia. São descritos a arquitetura implementada, o modelo de domínio, a persistência, os componentes de interface e servidor, a integração assistiva de inteligência artificial, os mecanismos de geração de documentos, a infraestrutura e as evidências de verificação técnica.

5.1 Visão Geral e Arquitetura Implementada

O EthicalSoft Compliance foi implementado como uma aplicação *web* composta por interface Angular, API Spring Boot, bancos PostgreSQL e MongoDB e integrações com serviços externos. A Figura 3 apresenta os componentes e seus mecanismos de comunicação.

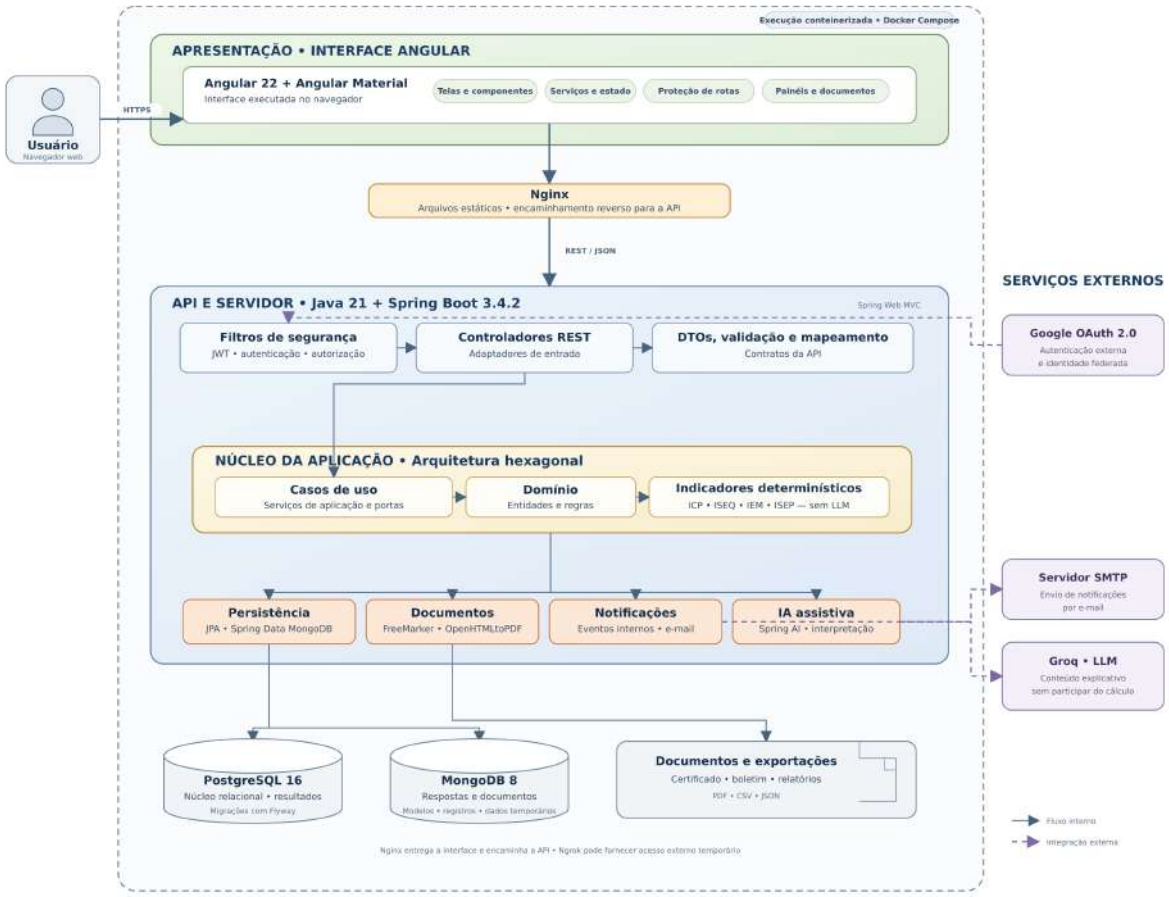

Figura 3 – Arquitetura implementada no EthicalSoft Compliance.

Fonte: O autor.

O *frontend* é executado no navegador e concentra autenticação, configuração de projetos, preenchimento de questionários, painéis e emissão de documentos. A comunicação com o *backend* ocorre por requisições REST em JSON. No *backend*, os *controllers* atuam como adaptadores de entrada, os casos de uso coordenam as regras de aplicação e as portas de saída abstraem persistência, envio de e-mail, autenticação externa, exportações, geração de documentos e integração com modelos de linguagem.

O PostgreSQL mantém o núcleo relacional e os resultados consolidados. O MongoDB armazena documentos aninhados ou variáveis, registros documentais e *caches*. A geração de PDF utiliza *templates* FreeMarker convertidos pelo OpenHTMLtoPDF. A integração de IA emprega o Spring AI com o provedor Groq e permanece separada da lógica determinística dos indicadores.

5.1.1 Casos de Uso Centrais

O modelo funcional representa o conjunto de atores e funcionalidades implementadas. A Figura 4 apresenta esse conjunto em um único diagrama, enquanto as narrativas completas são mantidas no Apêndice A.

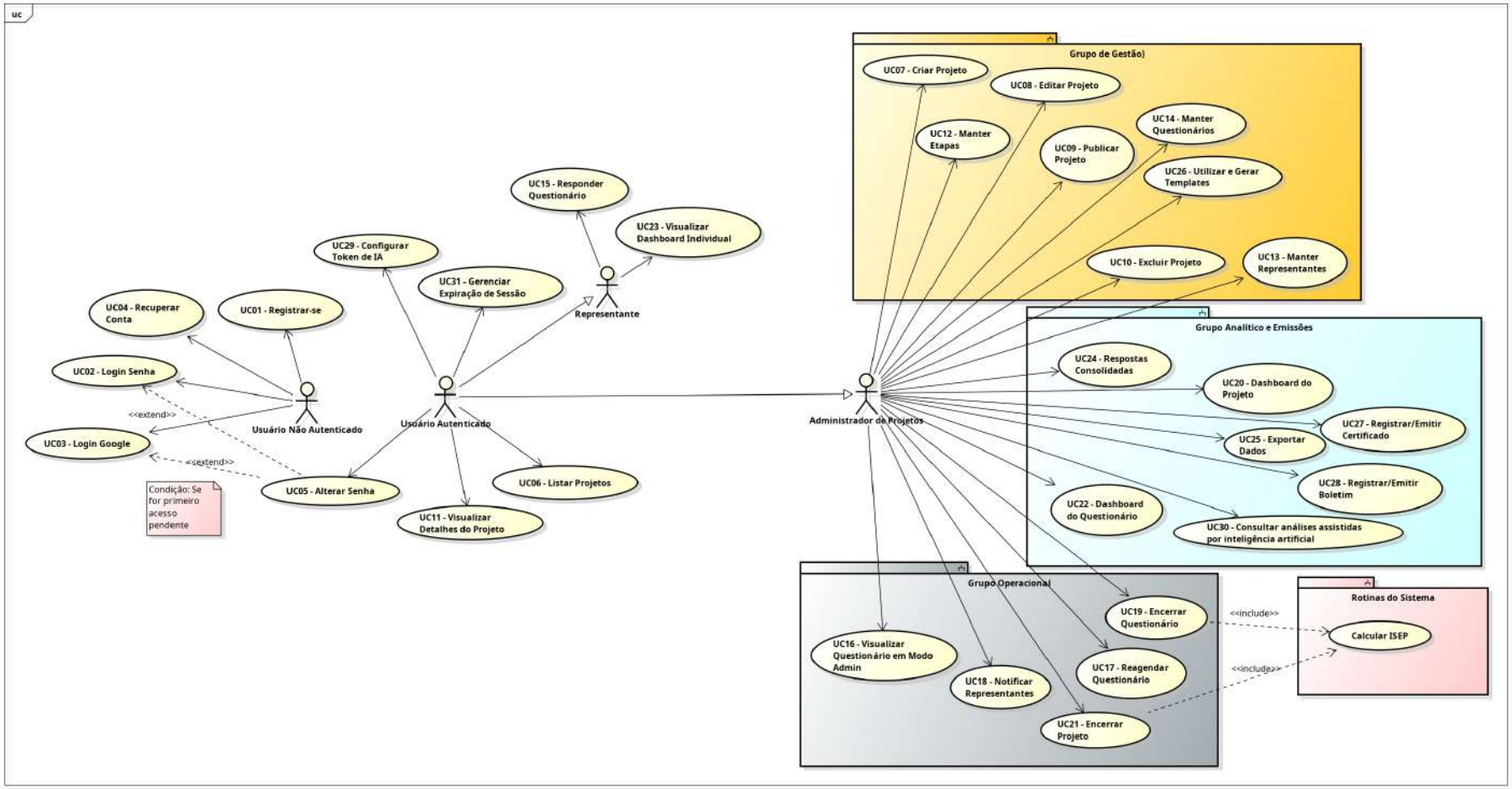

Figura 4 – Diagrama de casos de uso do EthicalSoft Compliance.

Fonte: O autor.

Entre os fluxos centrais, o UC07 — Criar Projeto materializa a configuração de tipo, etapas, iterações, representantes, pesos e questionários. Sua implementação é evidenciada pelo assistente de criação, pelas estratégias específicas para projetos iterativos e em cascata e pelos testes funcionais TF05 e TF06.

O UC15 — Responder Questionário reúne as regras de elegibilidade, salvamento de rascunho, submissão, justificativas, evidências e anexos. O fluxo é implementado pela tela sequencial de respostas e pelos serviços de persistência documental, com evidências nos testes TF07 e TF08.

O UC19 — Encerrar Questionário bloqueia novas respostas, consolida os resultados e aciona o cálculo do ISEQ. A execução manual e automática é registrada pela aplicação e verificada nos testes TF09 e TF10. Os demais casos e suas relações com regras de negócio e testes encontram-se nos apêndices de especificação e rastreabilidade.

5.2 Modelo de Domínio e Persistência

5.2.1 Modelo de Domínio

O modelo de domínio foi construído a partir dos requisitos e representa os principais conceitos envolvidos no acompanhamento da conformidade ética. A Figura 5 distingue essa visão conceitual do esquema relacional apresentado posteriormente.

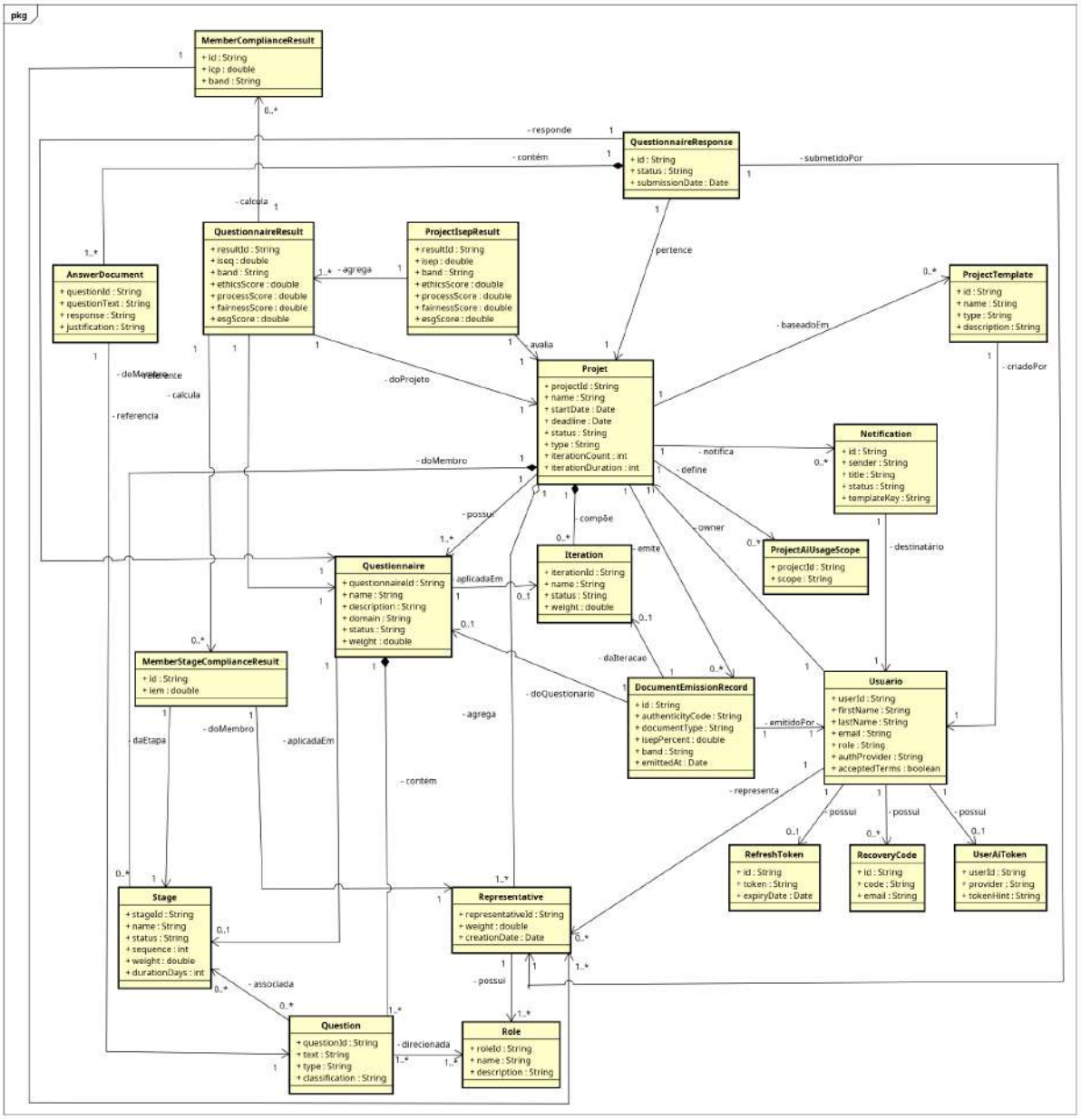

Figura 5 – Modelo conceitual de domínio do EthicalSoft Compliance.

Fonte: O autor.

O Projeto constitui o agregado central, a partir do qual são organizados questionários, etapas, iterações, representantes e resultados. O Questionário representa o instrumento aplicado em determinado contexto, enquanto Pergunta descreve cada critério de avaliação e seus vínculos com papéis e etapas. As respostas registram a participação de cada representante e podem incluir justificativas, evidências e anexos.

Etapa e Iteração permitem representar projetos em cascata e iterativos. Representante expressa a participação de um usuário em determinado projeto, enquanto Papel registra as responsabilidades exercidas. Os resultados do questionário e do projeto consolidam ICP, ISEQ, IEM e ISEP e fornecem dados para painéis e documentos. *Templates*, notificações, registros de emissão, configurações de documentos e *caches* constituem elementos de apoio.

5.2.2 Modelo Entidade-Relacionamento

O Modelo Entidade-Relacionamento descreve o núcleo persistido no PostgreSQL e apresenta tabelas, atributos, chaves e relacionamentos. A Figura 6 foi gerada a partir do esquema efetivamente utilizado. A tabela técnica *flyway_schema_history* aparece no esquema por registrar migrações, mas não integra o domínio da aplicação.

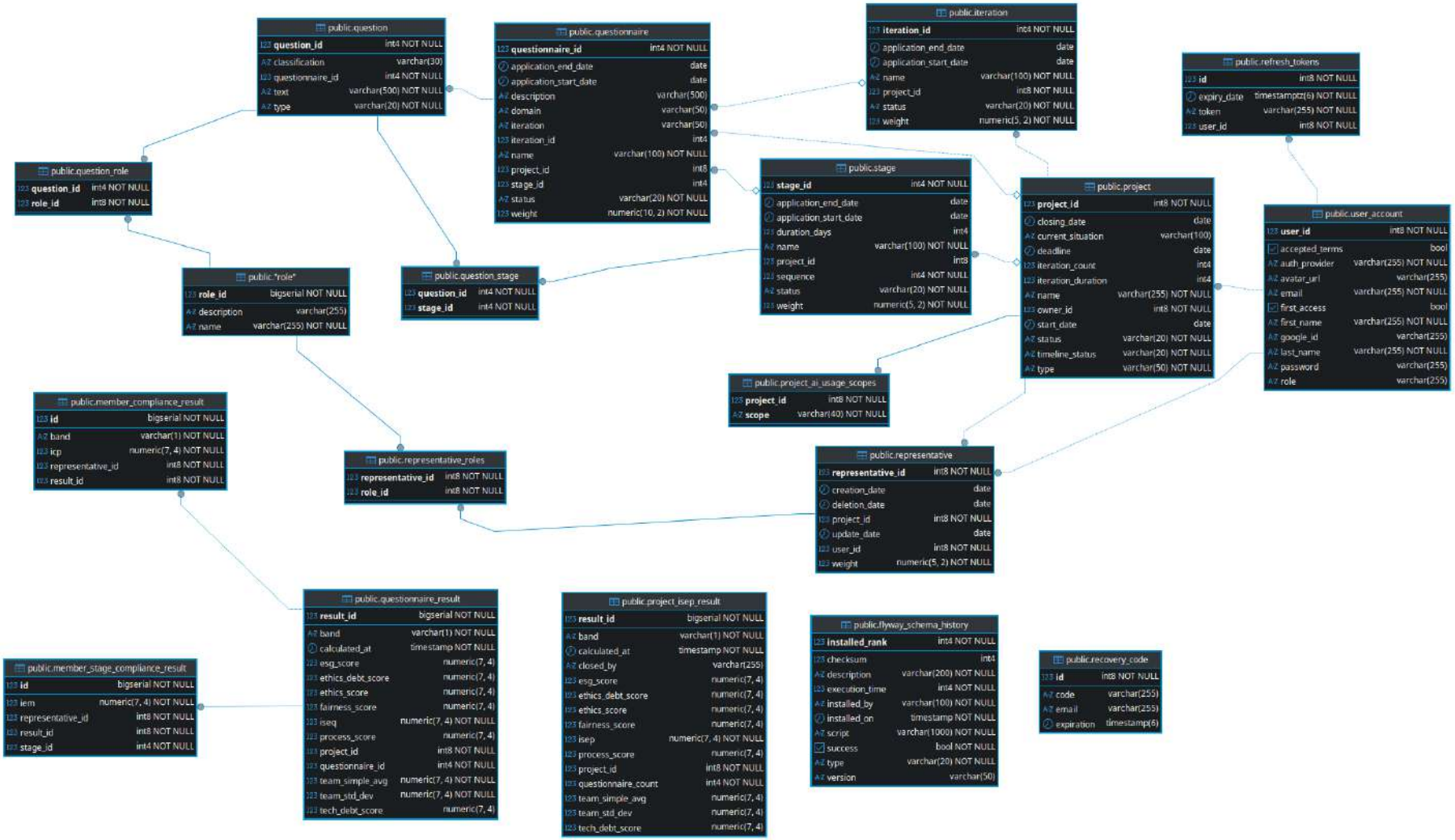

Figura 6 – Modelo Entidade-Relacionamento do núcleo relacional do EthicalSoft Compliance.

Fonte: O autor.

O núcleo relacional é formado por *user_account*, *project*, *representative*, *role*, *stage*, *iteration*, *questionnaire* e *question*. As tabelas associativas *representative_roles*, *question_role* e *question_stage* representam relações muitos-para-muitos. Os resultados dos questionários são mantidos em `questionnaire_result`. Os resultados dos projetos são mantidos em `project_isep_result`. O detalhamento por participante é mantido em `member_compliance_result`. O detalhamento por etapa é mantido em `member_stage_compliance_result`. Tabelas adicionais dão suporte a escopos de uso de IA, sessões e recuperação de conta.

O DDL impõe chaves estrangeiras nas relações centrais entre usuário, projeto, representante, papel, etapa, iteração, questionário e pergunta. Alguns identificadores existentes nas tabelas de resultado são associados pela camada de aplicação e não possuem, na versão examinada do DDL, uma restrição de chave estrangeira correspondente. Essa distinção é relevante para não atribuir ao banco garantias de integridade que não estejam declaradas no esquema.

5.2.3 Modelo Documental do MongoDB

O modelo documental foi levantado a partir das classes anotadas com *@Document* e dos índices declarados pela aplicação. Diferentemente do MER, que representa o esquema físico relacional, o Quadro 8 descreve as *collections*, os principais agregados e as associações lógicas identificadas na versão examinada.

Quadro 8 – Modelo documental das *collections* do MongoDB.

Collection	**Estrutura documental principal**	**Associações e índices**
`ai_generation_cache`	*hash*, operação, idioma, conteúdo, projeto, questionário e data de criação.	Índice único por *hash*; referências lógicas ao projeto e ao questionário.
`document_emission_records`	Código de autenticidade, tipo, instantâneos do projeto e do questionário, emissor, data, *hash*, resultado, faixa e versão do *template*.	Índice único por código de autenticidade; referências lógicas às entidades relacionais associadas.
`notifications`	Remetente e destinatário incorporados, título, conteúdo, estado, datas, *template* e evento de governança.	Índice composto por destinatário e estado; referência lógica ao *template*.

Continua na próxima página...

Continuação do Quadro 8

Collection	**Estrutura documental principal**	**Associações e índices**
`notification_templates`	Chave, papéis autorizados, destinatários, título, corpo, rota e canais.	Índice único por chave.
`pdf_document_configs`	Chave, título, identificação institucional, rodapé, resumo, ações corretivas, endereço de validação e emissor.	Índice único por chave.
`project_templates`	Dados gerais e listas aninhadas de etapas, questionários, perguntas, iterações e representantes.	Índice por usuário proprietário; referências lógicas a usuário, papéis e etapas.
`question_metadata`	Pergunta, domínio, tema, peso e indicação de criticidade.	Índice único por pergunta; referência lógica à pergunta relacional.
`questionnaire_responses`	Projeto, questionário, representante, etapa, estado e lista de respostas com justificativas, evidências e anexos.	Referências lógicas às entidades relacionais e identificadores incorporados em cada resposta.
`translation_cache`	*hash*, idioma, texto original, tradução e data de criação.	Índice único por *hash*.
`user_ai_tokens`	Usuário, *token* cifrado, indicação mascarada, provedor e datas.	Índice único por usuário; referência lógica à conta relacional.
`user_language_preferences`	Usuário, idioma e data de atualização.	Índice único por usuário; criação sob demanda e referência lógica à conta relacional.

Fonte: O autor.

A Figura 7 registra as *collections* presentes na instância utilizada durante a demonstração. A captura evidencia dez *collections* criadas naquele momento. O levantamento das classes também identifica *user_language_preferences*; como essa *collection* não aparece na captura, ela é tratada como estrutura prevista para criação sob demanda, e não como *collection* comprovadamente presente no instante registrado.

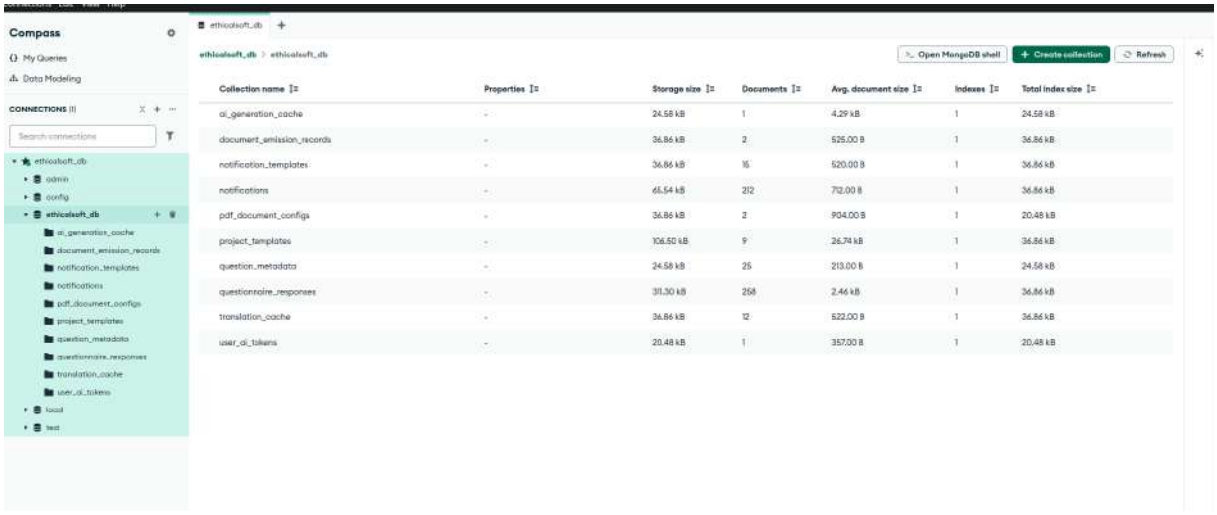

Figura 7 – *Collections* do MongoDB presentes na instância utilizada.

Fonte: O autor.

As estruturas de *project_templates* e *questionnaire_responses* incorporam listas e subdocumentos manipulados como agregados. Em *notifications*, os dados do remetente e do destinatário são preservados como instantâneos do momento do registro. Os índices descritos no Quadro 8 correspondem às anotações encontradas nas classes; não foram identificadas migrações documentais ou índices adicionais definidos separadamente por scripts.

5.2.4 Justificativa da Persistência Poliglota

A camada de persistência do EthicalSoft Compliance utiliza PostgreSQL 16 e MongoDB 8 para atender a grupos de dados com características distintas. A adoção conjunta desses mecanismos caracteriza a persistência poliglota. A escolha não se baseia na premissa de que um paradigma seja universalmente superior, mas na correspondência entre estrutura, padrão de acesso, garantias de consistência e custo operacional de cada grupo de dados (LANDUYT et al., 2023).

O PostgreSQL foi adotado para o núcleo relacional e transacional da plataforma. Nesse grupo estão usuários, projetos, representantes e papéis. Também são mantidos nesse banco etapas, iterações, questionários e perguntas instanciados, além dos resultados calculados. Essas informações possuem relacionamentos estáveis e restrições de integridade representadas no esquema da Figura 6. As transações oferecidas pelo banco permitem agrupar operações em unidades atômicas, evitando que etapas intermediárias sejam tratadas como alterações concluídas (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2026b).

O MongoDB foi utilizado para agregados documentais cujo formato contém listas aninhadas ou atributos auxiliares. Os *templates* reúnem, em uma unidade, etapas, questionários, perguntas, iterações e representantes sugeridos; as respostas detalhadas agregam justificativas, evidências e anexos; e registros de emissão, notificações, configurações e *caches* possuem ciclos próprios de criação e consulta. A incorporação de dados relacionados

em um documento é adequada quando essas informações são recuperadas ou atualizadas conjuntamente e oferece atomicidade no nível do documento (MONGODB, 2026a). Essa justificativa não se estende aos questionários e às perguntas já instanciados em projetos, que permanecem no PostgreSQL conforme o esquema implementado.

A flexibilidade documental não significa ausência de estrutura. O MongoDB oferece validação de esquema para controlar tipos, valores e campos dos documentos (MONGODB, 2026c). Entretanto, na versão examinada, a estrutura das *collections* é imposta pelas classes do Spring Data MongoDB e pelos índices nelas anotados; não foram identificadas regras declarativas de validação, versionamento geral dos documentos nem migrações para versões legadas. Essa condição pode permitir incompatibilidades quando as classes evoluem e, por isso, constitui limitação técnica e oportunidade de evolução (KLETTKE et al., 2016; MEURICE; CLEVE, 2017).

Também seria possível utilizar somente o PostgreSQL. O tipo *jsonb* permite armazenar documentos aninhados, consultar seu conteúdo e criar índices, preservando a flexibilidade estrutural sem introduzir um segundo sistema gerenciador de banco de dados (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2026a). Essa alternativa reduziria a complexidade de implantação, observabilidade, *backup* e restauração, além de facilitar operações que envolvam simultaneamente dados relacionais e documentais. A existência de dados variáveis, portanto, não torna o MongoDB obrigatório.

Na arquitetura implementada, o MongoDB separa o ciclo de evolução dos agregados documentais do esquema relacional e preserva *templates* com composições diferentes sem exigir que toda variação seja representada por novas tabelas e relações. Em contrapartida, dois serviços precisam ser configurados, protegidos, monitorados, atualizados e incluídos nos procedimentos de *backup* e recuperação. Os índices também são administrados separadamente, e as associações entre os armazenamentos não são protegidas por chaves estrangeiras.

As referências a projetos, questionários, perguntas, representantes e usuários presentes nos documentos são referências lógicas. A camada de aplicação deve verificar os registros correspondentes no PostgreSQL, mas essa verificação não equivale a uma restrição imposta pelo banco documental. Também não foi identificado, no material examinado, um mecanismo que assegure atomicidade distribuída, compensação ou reconciliação para operações que gravem nos dois bancos. Consequentemente, falhas entre essas gravações podem exigir tratamento específico pela aplicação, o que permanece como limitação da versão avaliada.

No escopo deste trabalho, o PostgreSQL constitui a fonte de verdade para identidades, vínculos, estado dos projetos, questionários instanciados e indicadores consolidados. O MongoDB constitui a fonte de verdade para o conteúdo integral dos *templates*, das respostas detalhadas, dos registros documentais e das configurações nele armazenadas. Instantâneos de nomes e dados relacionais mantidos nos documentos preservam o con-

texto histórico, mas não substituem o estado corrente das entidades relacionais. Os *caches* armazenam conteúdo derivado e não são tratados como fonte de verdade.

O Quadro 9 sintetiza essa distribuição e explicita a responsabilidade de cada armazenamento.

Quadro 9 – Responsabilidade dos armazenamentos no modelo poliglota.

Grupo de dados	**PostgreSQL**	**MongoDB**	**Papel no modelo**
Núcleo relacional	`user_account`, `project`, `representative`, `role`, `stage`, `iteration`, `questionnaire` e `question`	–	Fonte de verdade para identidades, vínculos e estado transacional.
Resultados consolidados	`questionnaire_result`, `member_compliance_result`, `member_stage_compliance_result` e `project_isep_result`	–	Fonte de verdade para os indicadores determinísticos.
Respostas detalhadas	–	`questionnaire_responses`	Fonte de verdade para respostas, justificativas, evidências e anexos; os identificadores são confrontados com o núcleo relacional.
Templates e metadados	`questionnaire` e `question` instanciados	`project_templates` e `question_metadata`	O PostgreSQL mantém as instâncias do projeto; o MongoDB mantém o agregado reutilizável e os metadados auxiliares.
Documentos e comunicação	–	`document_emission_records`, `notifications`, `notification_templates` e `pdf_document_configs`	Fonte de verdade para os respectivos documentos; instantâneos não substituem o estado relacional corrente.

Continua na próxima página...

Continuação do Quadro 9

Grupo de dados	**PostgreSQL**	**MongoDB**	**Papel no modelo**
Caches	–	`ai_generation_cache` e `translation_cache`	Conteúdo derivado, utilizado para reduzir chamadas e não tratado como fonte de verdade.
Preferências e credenciais	`user_account` identifica o usuário	`user_ai_tokens` e `user_language_preferences`	O PostgreSQL mantém a identidade; o MongoDB mantém as configurações específicas do usuário.

Fonte: O autor.

Assim, a persistência poliglota é justificada pela combinação entre relações estáveis e operações transacionais no PostgreSQL e agregados documentais com ciclos próprios de evolução no MongoDB. A decisão permanece tecnicamente condicionada ao reconhecimento da alternativa com *jsonb*, aos custos de operar dois bancos e à necessidade de controlar a evolução e a consistência dos documentos.

5.3 *Front-end*

Esta seção apresenta as principais interfaces do EthicalSoft Compliance e sua relação com os requisitos e casos de uso centrais do artefato. As telas foram agrupadas conforme os objetivos de interação dos usuários: acesso à plataforma, configuração de projetos, aplicação dos questionários, consulta aos resultados e emissão de documentos. Em vez de documentar todas as páginas da aplicação, são apresentadas as interfaces necessárias para demonstrar as decisões de interação, as regras de negócio e a continuidade entre as operações. As telas complementares são disponibilizadas no Apêndice G.

5.3.1 Organização da Interface e Navegação

A interface *web* foi desenvolvida com Angular 22 e Angular Material, com suporte à renderização no servidor. O projeto foi organizado por domínio e funcionalidade nas estruturas *features*, *shared*, *core* e *environments*. A pasta *core* concentra elementos de infraestrutura, como *guards*, *interceptors* e serviços base; *features* agrupa os módulos de negócio; *shared* reúne componentes, *pipes* e contratos reutilizáveis; e *environments* mantém configurações variáveis por ambiente.

A aplicação possui rotas públicas para cadastro, *login* e recuperação de conta e rotas protegidas para projetos, painéis e configurações. O Angular Router utiliza carregamento

sob demanda por funcionalidade. A autenticação pode ocorrer por e-mail e senha ou Google OAuth. Nos dois fluxos, o *backend* emite o *token* utilizado pela sessão da aplicação.

A interface oferece português do Brasil, inglês e espanhol por meio da biblioteca *@ngx-translate*. Textos explicativos e mensagens auxiliares são centralizados em recursos próprios, e a preferência de idioma é mantida entre sessões por uma API específica.

5.3.2 Fluxos de Interação da Aplicação

5.3.2.1 Autenticação e Acesso aos Projetos

O fluxo de entrada reúne autenticação por e-mail e senha, acesso por conta Google, cadastro e recuperação de conta. A Figura 8 apresenta as interfaces que materializam esses casos de uso.

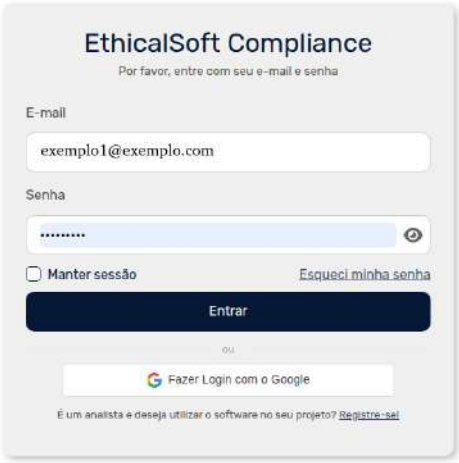

Figura 8 – Fluxo de autenticação e recuperação de acesso.

Fonte: O autor.

A interface demonstra os pontos de entrada implementados e a continuidade entre autenticação, cadastro e recuperação de acesso. Sua apresentação comprova a disponibilidade funcional desses caminhos, mas não constitui avaliação de usabilidade, acessibilidade ou segurança do mecanismo de autenticação.

Após a autenticação, o usuário acessa a relação paginada dos projetos aos quais está vinculado. A Figura 9 apresenta a tela com os filtros por nome, código, tipo e situação.

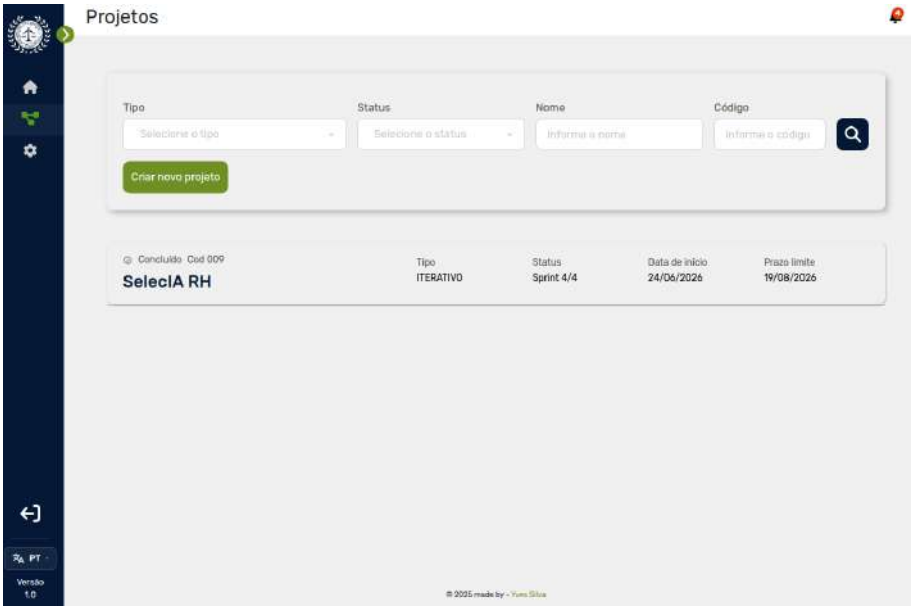

Figura 9 – Listagem e filtragem dos projetos acessíveis ao usuário.

Fonte: O autor.

A listagem funciona como ponto de acesso aos fluxos de gestão e participação. As ações disponíveis dependem do vínculo e do perfil do usuário no projeto, preservando a distinção entre operações administrativas e operações atribuídas aos representantes.

5.3.2.2 Configuração e Gerenciamento do Projeto

O fluxo de criação do projeto materializa as regras de configuração que antecedem a avaliação ética. Conforme apresentado na Figura 10, o administrador informa os dados gerais, define etapas ou iterações, configura papéis, associa representantes e distribui as perguntas. A divisão em cinco painéis procura reduzir a quantidade de informações solicitadas em uma única tela e impedir que o projeto seja publicado sem os elementos necessários ao cálculo dos indicadores. A seleção entre os modelos cascata e iterativo não produz apenas uma alteração visual: determina a organização dos questionários, a recorrência das perguntas, as datas de aplicação e os pesos utilizados na consolidação do resultado.

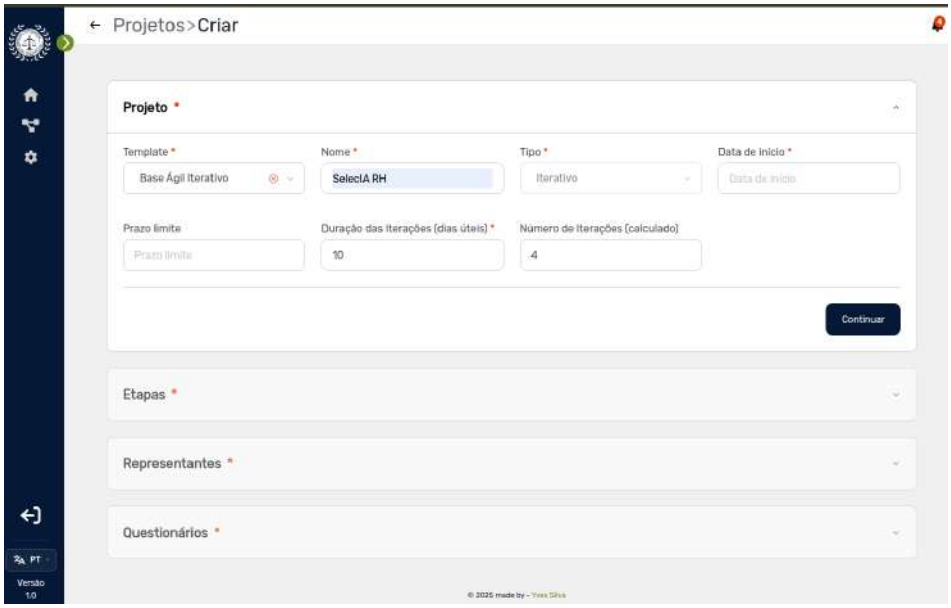

Figura 10 – Assistente de criação e configuração do projeto.

Fonte: O autor.

Os formulários organizam a configuração de tipo, cronograma, etapas, iterações, representantes, papéis, pesos e perguntas. A tela complementar de seleção do tipo de projeto é apresentada na Figura 87, no Apêndice G.

Depois da criação, a página de detalhes centraliza o acesso aos questionários, aos painéis e às ações de publicação e encerramento. A Figura 11 apresenta esse ponto de continuidade do fluxo administrativo.

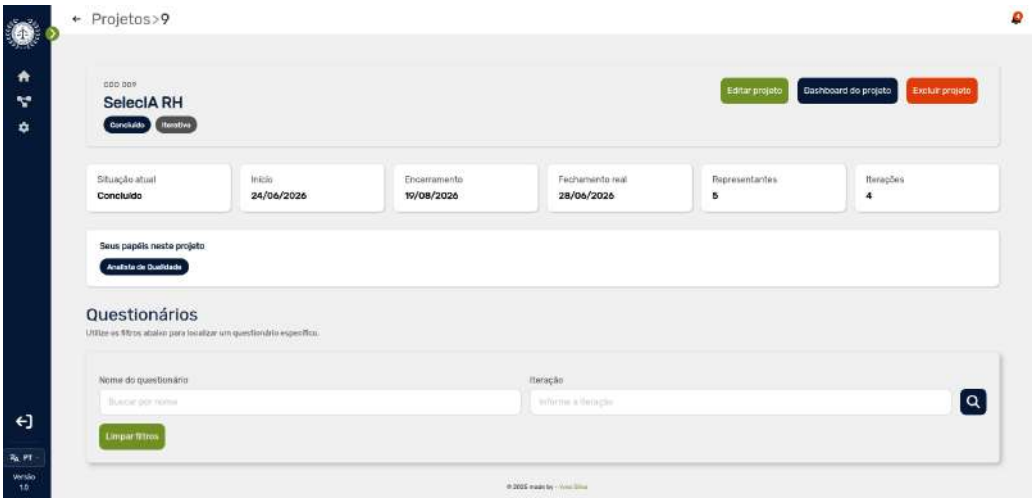

Figura 11 – Detalhes do projeto e ações administrativas disponíveis.

Fonte: O autor.

As operações de gestão permanecem condicionadas ao perfil administrativo e ao estado do projeto. A interface evidencia a reunião das ações implementadas, enquanto a verificação de suas regras de permissão e transição é registrada nos casos de teste do Apêndice D.

5.3.2.3 Aplicação dos Questionários e Registro de Evidências

A interface de resposta ao questionário corresponde ao principal ponto de coleta das evidências utilizadas pela plataforma. A Figura 12 apresenta uma pergunta em foco, as alternativas disponíveis e os campos condicionais de evidência e justificativa. Uma resposta afirmativa exige o registro de uma evidência, enquanto uma resposta negativa exige sua justificativa. Essa regra procura impedir que a conformidade seja declarada apenas por meio de uma seleção binária sem informação complementar.

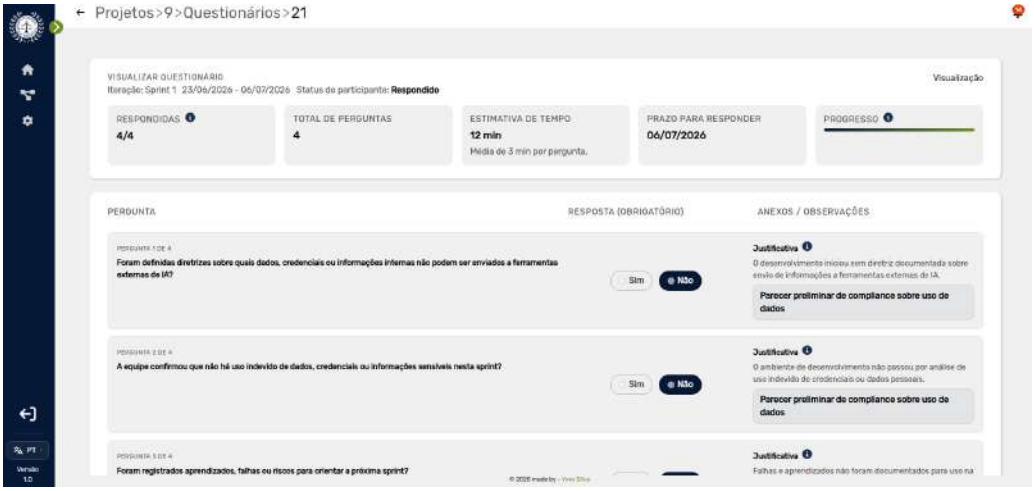

Figura 12 – Resposta ao questionário e registro condicionado de evidência ou justificativa.

Fonte: O autor.

O salvamento em rascunho permite retomar o questionário antes da submissão final. A existência dos campos e das validações comprova o mecanismo de registro implementado, mas não comprova a veracidade, a suficiência ou a autenticidade da evidência fornecida; seu conteúdo precisa ser examinado posteriormente pelos responsáveis pelo projeto.

5.3.2.4 Painéis, Documentos e Configurações

A aplicação disponibiliza três níveis complementares de visualização: individual, questionário e projeto. O painel individual permite ao participante acompanhar seu ICP sem expor diretamente os resultados dos demais. O painel do questionário apresenta o ISEQ e as divergências entre papéis e etapas. O painel do projeto consolida os questionários no ISEP e apresenta a evolução ao longo do ciclo. Neste capítulo, as interfaces são descritas como componentes implementados. Os valores produzidos por elas e sua interpretação são examinados no Capítulo 6.

A Figura 13 apresenta a organização do painel consolidado e a disposição dos componentes analíticos.

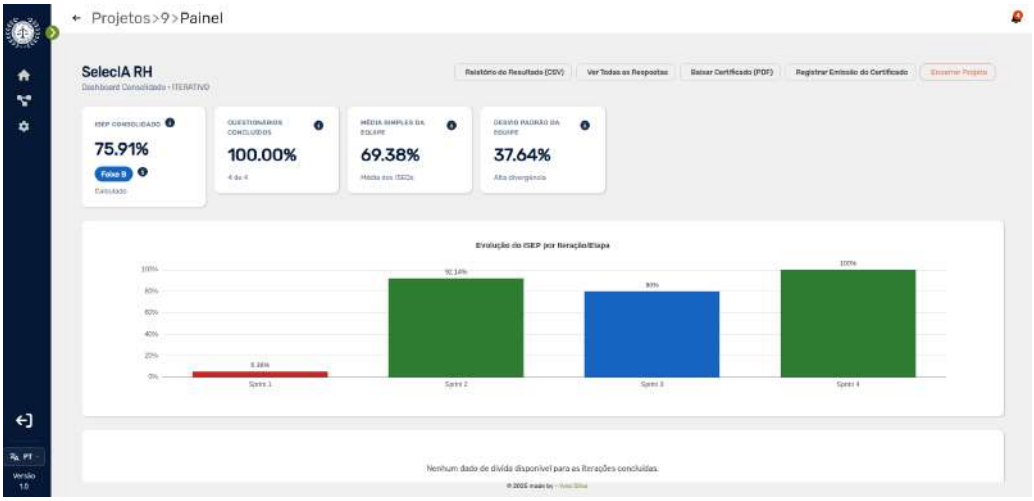

Figura 13 – Organização da interface do painel consolidado do projeto.

Fonte: O autor.

Os gráficos são renderizados com *ngx-echarts*, que integra o Apache ECharts. Os componentes são reutilizados entre os painéis de projeto, questionário e participante. O Quadro 10 relaciona os componentes principais às respectivas finalidades.

Quadro 10 – Componentes de visualização e contextos de uso.

Componente	**Tipo**	**Contexto de uso**
isep-evolution-chart	Barras	Evolução por iteração ou etapa.
role-stage-bar-chart	Barras agrupadas	Comparação entre papéis e etapas.
individual-radar-chart	Radar	Análise individual do participante.
debt-evolution-chart	Linha ou área	Evolução das medidas de dívida.
band-distribution-chart	Pizza ou barras	Distribuição por faixas.
governance-dimensions-chart	Radar ou barras	Dimensões de governança.
role-stage-heatmap	Mapa de calor	Conformidade por papel e etapa.
word-cloud-widget	Nuvem de palavras	Termos presentes nas justificativas.

Fonte: O autor.

Os documentos condicionados às faixas de classificação são disponibilizados para consulta e descarga na interface apresentada na Figura 14. Os valores, as regras de emissão e o alcance desses documentos são discutidos na Seção 6.6.

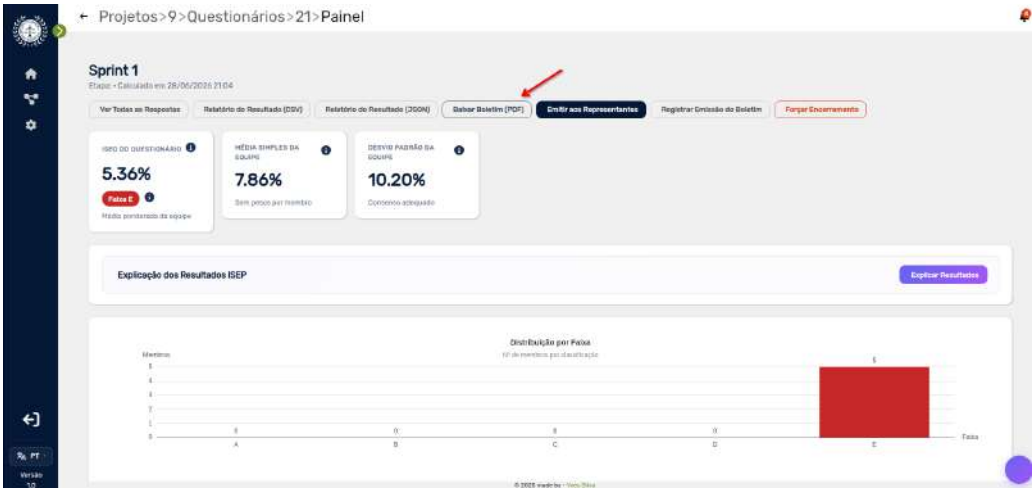

Figura 14 – Consulta e acesso aos documentos de conformidade gerados.

Fonte: O autor.

Em relação à visibilidade compartilhada discutida na Seção 3.4, o comportamento implementado é delimitado pelos vínculos e perfis dos usuários. A listagem da Figura 9 apresenta os projetos acessíveis ao usuário autenticado; o assistente da Figura 10 permite ao administrador configurar papéis, representantes, etapas e questionários; e a página da Figura 11 reúne o andamento e as ações autorizadas do projeto. Os representantes registram respostas, evidências ou justificativas pela interface da Figura 12; os resultados são apresentados nos níveis individual, de questionário e de projeto, conforme exemplificado pela Figura 13; e os documentos emitidos podem ser consultados pela interface da Figura 14. Esses comportamentos foram exercitados, respectivamente, nos casos TF05, TF07, TF10, TF12, TF13, TF14, TF15 e TF16 do Apêndice D.

Esses recursos não implementam gestão de *Product Backlog*, objetivo do produto, critérios de aceitação, *Definition of Done* ou atribuição do papel de *Product Owner*. Eles fornecem apoio informacional à avaliação ética e à rastreabilidade das decisões registradas. O acesso tampouco é irrestrito: as operações administrativas dependem do perfil e do estado do projeto, e o painel individual permite ao participante acompanhar o próprio resultado sem expor diretamente os valores nominais dos demais. A plataforma, portanto, amplia a visibilidade das informações autorizadas, mas não garante a qualidade ou a veracidade das evidências fornecidas nem substitui as práticas de engenharia e de gestão adotadas pela organização.

A área de configurações pessoais permanece entre as telas complementares do Apêndice G. Essa interface permite alterar a senha e gerenciar a credencial individual utilizada pelos recursos assistivos de IA, conforme a Figura 88.

5.4 *Back-end*

O *backend* foi desenvolvido em Java 21 com Spring Boot 3.4.2 e Spring Web MVC. O pacote raiz separa domínio, aplicação, adaptadores e infraestrutura. A camada de domínio concentra entidades e regras; a camada de aplicação contém casos de uso e portas; os adaptadores de entrada expõem a API; e os adaptadores de saída implementam persistência e integrações.

O padrão *Strategy* foi empregado na criação de projetos para encapsular as diferenças entre estruturas em cascata e iterativas. Essa decisão evita concentrar regras específicas em condicionais no caso de uso principal e permite evoluir cada estratégia separadamente.

5.4.1 API REST

Os *controllers* são organizados por responsabilidade. Autenticação trata *login* local, OAuth, cadastro e recuperação de conta. Projetos concentram criação, edição, publicação e encerramento. Questionários coordenam aplicação, respostas e consolidação. Há ainda controladores para painéis, documentos, *tokens* de IA, internacionalização, *templates*, notificações e dados do usuário. O contrato resumido dos *endpoints* encontra-se no apêndice de instalação e reprodução.

5.4.2 Exemplo de Fluxo de Requisição

Como exemplo verificável da integração entre interface, aplicação e persistência, o UC15 — Responder Questionário inicia-se quando o *frontend* carrega a participação do representante autenticado pelo *endpoint* de respostas. Durante o preenchimento, o rascunho é enviado por uma requisição `PUT` para `</api/questionnaires/{questionnaireId}/responses/draft>`; a submissão definitiva utiliza uma requisição `POST` na rota-base do mesmo recurso. O contrato dessas operações é apresentado no Quadro 26, no Apêndice F.

O *token* da sessão acompanha a requisição, o adaptador de entrada identifica o usuário e o caso de uso verifica seu vínculo, sua elegibilidade e o estado do questionário. As respostas detalhadas, incluindo justificativas e evidências, são persistidas na *collection questionnaire_responses*, conforme a distribuição apresentada no Quadro 9. Após o encerramento, os cálculos determinísticos produzem os resultados consolidados mantidos no PostgreSQL. Esses resultados são recuperados pelos *endpoints* dos painéis descritos no Quadro 27. Assim, o fluxo relaciona a interface da Figura 12, a arquitetura da Figura 3, o contrato da API e os dois mecanismos de persistência sem atribuir ao modelo de linguagem participação no cálculo.

5.4.3 Autenticação e Autorização

No fluxo local, o serviço de *tokens* gera *tokens* de acesso e renovação assinados. No fluxo Google, o *backend* valida o *token* emitido pelo provedor e gera uma sessão própria da aplicação. Os *endpoints* públicos restringem-se aos fluxos de autenticação e às requisições de *preflight* CORS. As demais operações exigem autenticação e são submetidas às regras de perfil e vínculo com o projeto.

5.5 Integração Assistiva de Inteligência Artificial

O módulo de IA foi implementado com Spring AI e o provedor Groq. Cada usuário pode cadastrar sua credencial, armazenada de forma criptografada e associada à própria conta. Em tempo de execução, a aplicação resolve o cliente do modelo de acordo com o usuário autenticado, sem compartilhar a credencial entre contas.

Conforme a operação solicitada, são enviados ao serviço externo os indicadores já calculados e os conteúdos textuais necessários à interpretação, como justificativas e o contexto do questionário. As respostas produzidas incluem explicação do índice, síntese das justificativas, relatório textual de riscos e respostas ao *chat* contextual. O conteúdo retornado é apresentado como apoio interpretativo e não como evidência primária.

As respostas são armazenadas em *ai_generation_cache*, com chave derivada dos parâmetros relevantes da operação. Um mecanismo de contingência mantém o restante da aplicação disponível quando o provedor falha. A indisponibilidade do modelo não impede a consulta dos valores numéricos, pois pesos, respostas, faixas e indicadores são processados exclusivamente pela lógica de negócio. A IA não altera respostas, não define parâmetros e não participa da emissão determinística dos documentos.

No *frontend*, o conteúdo é apresentado por um componente de *chat* e por *widgets* analíticos. As respostas progressivas utilizam transmissão em fluxo, e o conteúdo textual é renderizado por um *pipe* específico. Essa integração permanece separada dos componentes responsáveis pelo cálculo e pela persistência dos resultados.

5.6 Geração de Documentos, Notificações e Exportações

O sistema emite o Certificado de Ciência Ética e Conformidade Declarada e o Boletim de Não Conformidade Ética. Os dados são injetados em *templates* FreeMarker e convertidos de HTML para PDF pelo OpenHTMLtoPDF. A separação entre *template* e código permite alterar o conteúdo visual sem modificar a lógica de cálculo.

Cada emissão gera um registro documental com código de autenticidade, tipo, escopo, data, usuário emissor, índices, faixa, versão do *template* e *hash* dos dados. Esse registro preserva um instantâneo necessário à trilha de auditoria. Notificações internas e e-mails

são acionados por eventos, como atribuição de projeto, proximidade de prazo, encerramento e não conformidade.

Os painéis disponibilizam exportação dos dados analíticos em CSV e JSON. A geração utiliza Apache Commons CSV e Gson, enquanto os documentos de conformidade são disponibilizados em PDF.

5.7 Infraestrutura e Implantação

A infraestrutura é conteinerizada e orquestrada por Docker Compose. O ambiente inclui *backend*, Nginx, PostgreSQL e MongoDB em rede própria. O Nginx entrega os arquivos do *frontend* e atua como *proxy* para as chamadas da API. O acesso externo temporário pode ser realizado por túnel Ngrok.

A configuração utiliza variáveis de ambiente para credenciais e endereços externos. Migrações Flyway inicializam e atualizam o esquema relacional. Os pré-requisitos, as variáveis e o procedimento de execução local são apresentados no apêndice de instalação e reprodução.

5.8 Verificação e Qualidade do Código

A verificação automatizada utiliza JUnit 5 e Mockito, complementados por JaCoCo 0.8.11 e SonarQube Cloud. A suíte prioriza cálculo dos indicadores, estratégias de criação de projeto, autenticação, exportação e casos de uso relacionados a questionários e expiração de sessão.

A execução da versão `0.0.1-SNAPSHOT`, realizada em 30 de junho de 2026, resultou em 43 testes executados, sem falhas, erros ou testes ignorados, em 23,1 segundos. O relatório JaCoCo, sobre 192 classes analisadas, registrou cobertura de 12% das instruções e 7% dos desvios condicionais. Os pacotes com maior cobertura foram `<application.service.strategy.project_strategy>`, `<application.usecase.notification.command>`, `<domain.notification>` e `<domain.isep>`.

A inspeção das 11 classes responsáveis pelos 43 casos automatizados identificou 43 testes unitários e nenhum teste de integração. Em todas as classes, apenas o componente sob teste foi inicializado como implementação real, enquanto repositórios, portas, gerenciadores e demais dependências foram substituídos por objetos simulados com Mockito. Nenhuma classe inicializou contexto Spring, banco de dados, servidor ou colaboração entre componentes reais. A classificação completa e os resultados por classe são apresentados no Apêndice D (OFFUTT, 1991; PEZZÈ; RUBINOV; WUTTKE, 2013).

A análise estática da *branch* `main`, no *commit* `4175115d`, registrou 15 novas ocorrências no período observado, duplicação de 0,59% sobre 5,4 mil linhas novas e nenhum *security*

hotspot. A cobertura do JaCoCo não estava integrada ao SonarQube Cloud nessa versão. A Figura 15 apresenta o painel da análise.

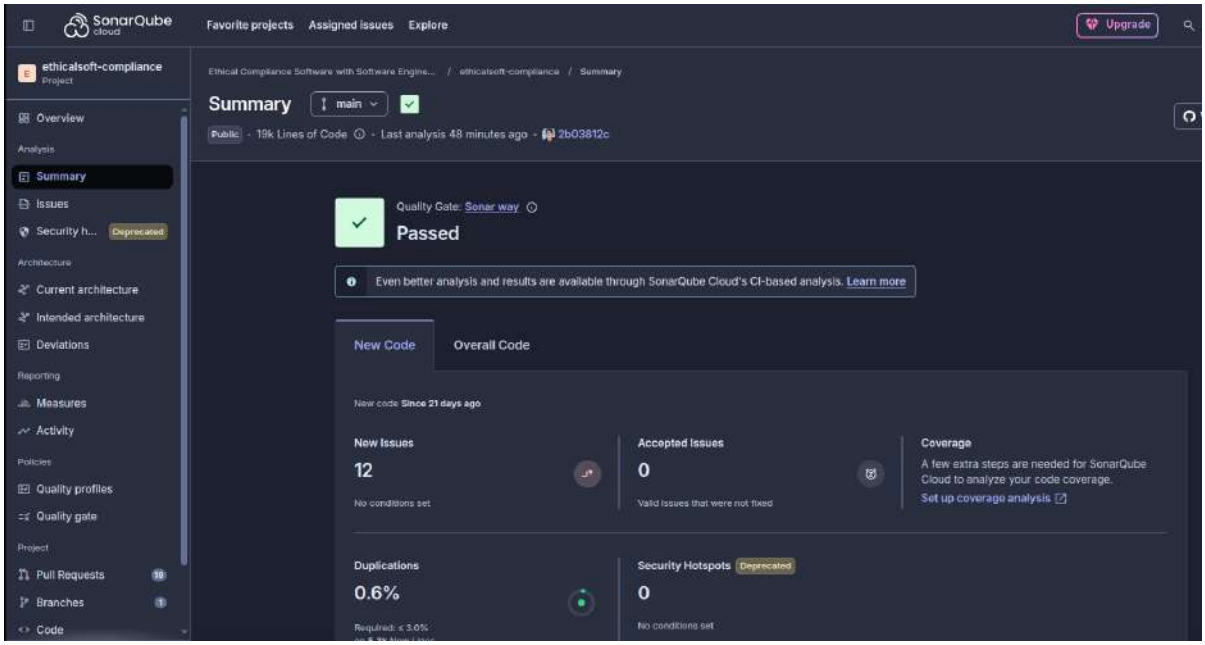

Figura 15 – Resultado da análise estática da *branch* principal no SonarQube Cloud.

Fonte: O autor.

Não foram executados testes automatizados de integração nem suítes abrangentes de contrato de API, automação de interface de ponta a ponta, avaliação formal de usabilidade, ensaios de desempenho sob carga, testes de penetração ou aceitação com usuários externos. Assim, os resultados demonstram o comportamento isolado dos componentes e os fluxos funcionais manuais examinados, mas não autorizam concluir que o sistema esteja integralmente coberto, livre de defeitos, seguro sob ataque ou validado em contexto organizacional real. Os casos de teste e as evidências disponíveis são apresentados no Apêndice D.

6 RESULTADOS E DISCUSSÃO

Este capítulo apresenta e discute as evidências produzidas durante a execução do cenário simulado SelecIA RH. A análise não se limita à demonstração das interfaces: relaciona as operações executadas aos requisitos do artefato, examina os valores produzidos pelos indicadores e discute o que as visualizações permitem identificar. Inicialmente, apresenta-se a correspondência entre objetivos, funcionalidades e evidências. Em seguida, descrevem-se a configuração e a execução do cenário. Por fim, os resultados são confrontados com os conceitos de conformidade ética, governança, rastreabilidade e prestação de contas discutidos na fundamentação teórica. Como os dados são sintéticos e os participantes foram simulados, os resultados demonstram o comportamento e a coerência interna do artefato, mas não comprovam sua efetividade em organizações reais.

6.1 Artefato Produzido e Rastreabilidade dos Requisitos

O resultado técnico da pesquisa é o EthicalSoft Compliance, aplicação *web* que reúne configuração de projetos, questionários por etapa e papel, respostas com evidências, indicadores, painéis, notificações, documentos e registros de emissão. O artefato implementa os fluxos especificados nos casos de uso e aplica as regras de negócio descritas nos apêndices.

O Quadro 11 associa cada objetivo específico à funcionalidade implementada, ao procedimento executado, à evidência examinada, ao resultado observado e à respectiva limitação. Essa relação evita tratar a navegação pelas telas como resultado autossuficiente. A matriz completa entre objetivos, casos de uso, regras de negócio e testes permanece no Apêndice C.

Quadro 11 – Rastreabilidade entre objetivos, execução e resultados observados.

Objetivo específico	**Funcionalidade implementada**	**Procedimento executado**	**Evidência**	**Resultado**	**Limitação**
OE1	Questionários organizados por etapa, iteração, papel e domínio.	Classificação e distribuição das perguntas entre as quatro *sprints* e os representantes elegíveis.	Configuração da Figura 17, respostas registradas e especificações dos apêndices.	Estrutura configurável aplicada durante todo o cenário.	O conjunto de perguntas não foi validado externamente.
OE2	Assistente de projeto, *templates*, pesos, papéis, etapas, iterações e escopos de IA.	Criação do SelecIA RH com quatro *sprints*, cinco representantes e pesos distintos.	Figura 16, Quadro 12 e testes TF05 e TF06.	Parâmetros registrados e utilizados na distribuição e na consolidação.	Não foram examinadas configurações definidas por organizações reais.
OE3	Cálculo de ICP, ISEQ, ISEP e IEM.	Encerramento dos questionários e do projeto após o registro das respostas sintéticas.	Quadro 13 e testes TF10 a TF12.	ISEQs de 5,36%, 92,14%, 80,00% e 100,00%; ISEP de 75,91%.	Os indicadores são medidas operacionais e não universais de comportamento ético.
OE4	Painéis de participante, questionário, projeto e respostas consolidadas.	Consulta aos três níveis de visualização após os cálculos.	Figuras da Seção 6.5 e testes TF15 e TF16.	Visualização de resultados agregados, individuais, temporais e por papel e etapa.	Não houve avaliação formal de usabilidade ou acessibilidade.

Continua na próxima página...

Continuação do Quadro 11

Objetivo específico	**Funcionalidade implementada**	**Procedimento executado**	**Evidência**	**Resultado**	**Limitação**
OE5	Geração do Boletim de Não Conformidade Ética.	Encerramento da *Sprint* 1 abaixo do limiar configurado.	Registro de emissão, interface da Figura 14 e teste TF13.	Boletim condicionado à faixa E gerado e disponibilizado.	A utilidade de seu conteúdo não foi avaliada por usuários externos.
OE6	Geração do Certificado de Ciência Ética e Conformidade Declarada.	Encerramento do projeto com resultado no limiar configurado.	Figura 26 e teste TF14.	Certificado emitido para o ISEP de 75,91%.	O documento não equivale a certificação normativa ou auditoria independente.
OE7	Registro dos metadados das emissões.	Emissão dos documentos e consulta aos registros correspondentes.	*Collection* `<document_emission_records>` e Seção 6.6.	Código, escopo, data, emissor, índice, faixa, versão e *hash* preservados.	O mecanismo de autenticidade não foi submetido a auditoria externa.
OE8	Notificações, encerramentos e registros auditáveis associados aos eventos.	Execução dos eventos de questionário e de emissão previstos no cenário.	Testes TF09, TF13, TF14 e TF20 e *collections* documentais.	Eventos exercitados e registros recuperáveis produzidos.	Não foi demonstrada uma trilha única e completa para todas as categorias de evento.
OE9	Dimensões de governança e mecanismos de transparência e monitoramento.	Confronto entre funcionalidades, resultados e conceitos do COBIT 2019 e do ITIL 4.	Fundamentação teórica e Seção 6.7.	Alinhamento conceitual e funcional explicitado.	Não houve avaliação organizacional formal baseada nesses referenciais.

Continua na próxima página...

Continuação do Quadro 11

Objetivo específico	**Funcionalidade implementada**	**Procedimento executado**	**Evidência**	**Resultado**	**Limitação**
OE10	Fluxo completo de demonstração funcional.	Criação, resposta, encerramento, cálculo, visualização, emissão e recuperação de evidências.	Protocolo da Seção 4.5, Apêndices D e E e este capítulo.	Integração funcional e coerência interna demonstradas no cenário.	Dados sintéticos não permitem inferir efetividade com equipes reais.

Fonte: O autor.

6.2 Configuração do Cenário Simulado

O cenário SelecIA RH representa uma plataforma fictícia de recrutamento e seleção que utiliza IA para analisar currículos, identificar competências, propor ranqueamento e gerar justificativas. O domínio foi escolhido por concentrar riscos relacionados a privacidade, viés, transparência e responsabilização. Os dados são sintéticos e não representam organização, equipe ou produto real.

As entradas foram construídas manualmente pelo pesquisador, sem base empírica ou distribuição probabilística. Os papéis simulados cobrem perspectivas gerenciais, técnicas, de qualidade, de parte interessada e de cliente; as respostas seguem as regras de elegibilidade da plataforma e foram configuradas para produzir situações contrastantes. Os pesos servem para exercitar os cálculos e não constituem hierarquia profissional validada.

O projeto foi configurado como iterativo, com quatro *sprints*. Os pesos dos questionários foram definidos em 1,0, 1,0, 1,5 e 2,0. A Figura 16 registra a configuração que condicionou a consolidação dos resultados.

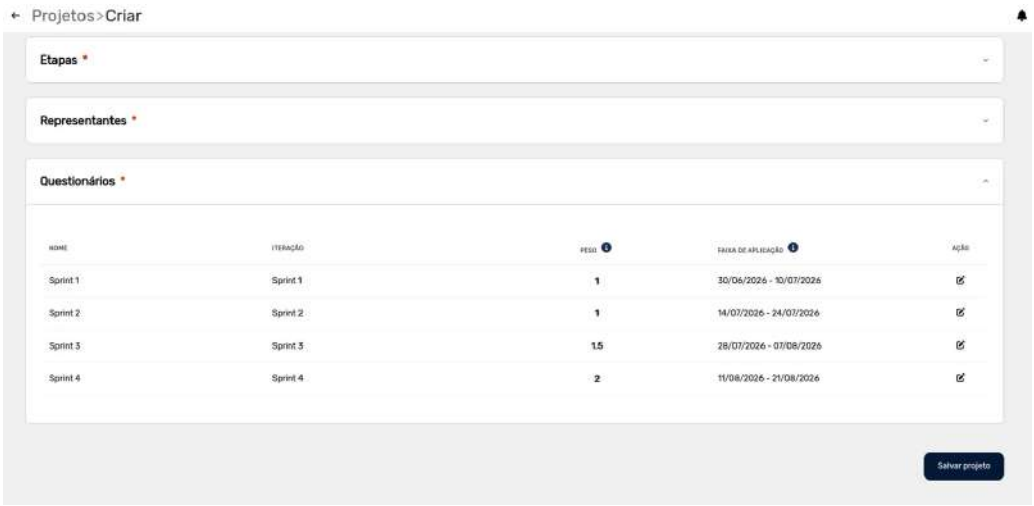

Figura 16 – Pesos das *sprints* e etapas configurados para o cenário simulado.

Fonte: O autor.

A atribuição crescente de pesos foi definida para exercitar a ponderação das iterações e não deriva de dados empíricos sobre projetos de recrutamento. Consequentemente, os valores finais devem ser interpretados em conjunto com essa escolha de configuração.

Foram configurados cinco representantes, com papéis e pesos apresentados no Quadro 12.

Quadro 12 – Representantes, papéis e pesos do cenário simulado.

Representante	**Papel**	**Peso**	**Sigla**
Representante 1	Gerente de Projeto	2,0	GP

Continua na próxima página...

Continuação do Quadro 12

Representante	**Papel**	**Peso**	**Sigla**
Representante 2	Analista de Qualidade	2,0	AQ
Representante 3	Desenvolvedor	3,0	DEV
Representante 4	*Stakeholder*	1,0	STK
Representante 5	Cliente	2,0	CLI

Fonte: O autor.

Os pesos dos representantes também foram definidos para produzir diferenças observáveis entre as contribuições individuais. Eles não expressam uma hierarquia profissional validada nem uma recomendação aplicável a outros projetos.

As perguntas foram classificadas como aplicáveis ao projeto inteiro, recorrentes ou específicas da etapa atual. As perguntas estruturais foram concentradas na primeira *sprint*; as recorrentes foram reaplicadas; e as específicas variaram conforme o foco de cada iteração. A Figura 17 registra a distribuição configurada.

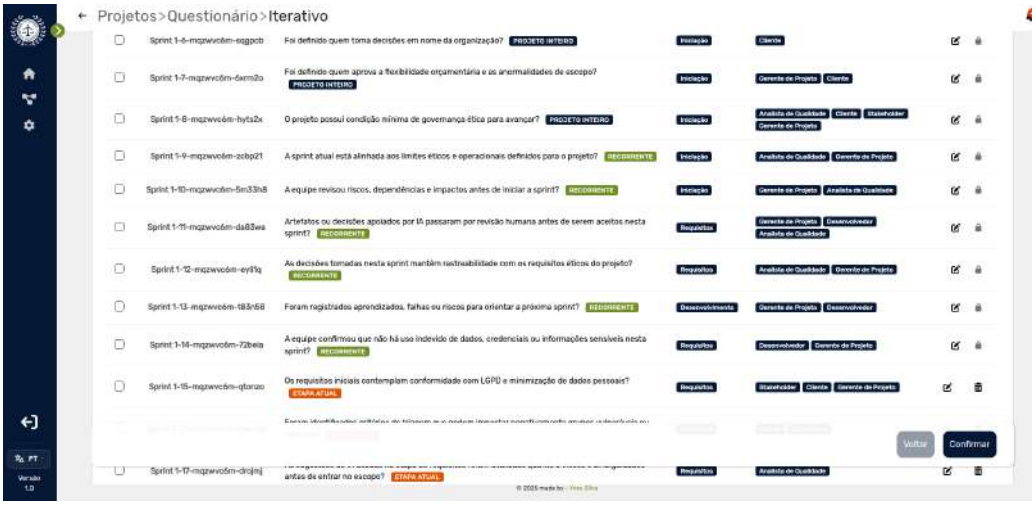

Figura 17 – Distribuição das perguntas por escopo no cenário simulado.

Fonte: O autor.

A configuração demonstra que o artefato diferencia perguntas estruturais, recorrentes e específicas, evitando sua replicação manual entre os questionários. Uma iteração não é tratada como se atravessasse obrigatoriamente todas as etapas: sua cobertura depende dos vínculos configurados entre perguntas, etapas e questionários. A figura não demonstra, por si só, que a distribuição seja suficiente para avaliar projetos reais; essa suficiência exigiria validação externa dos critérios e dos instrumentos.

6.3 Execução Funcional do Cenário

A execução funcional percorreu o ciclo completo previsto para o cenário: criação do projeto, configuração de quatro iterações, associação dos cinco papéis simulados, distribuição das perguntas, registro das respostas, encerramento dos questionários, cálculo dos indicadores, geração das visualizações e emissão dos documentos. O percurso demonstrou que os dados registrados nas interfaces foram processados pelas regras definidas no modelo de avaliação e refletidos nos painéis correspondentes. Essa evidência verifica a integração funcional entre configuração, coleta, cálculo, visualização e geração documental. Entretanto, não avalia facilidade de uso, desempenho sob carga ou aceitação por usuários externos.

Cada representante respondeu somente às perguntas elegíveis para os papéis exercidos. No total, foram registrados 235 itens de resposta. Na primeira *sprint*, as entradas foram configuradas para representar ausência de controles éticos estruturados. Na segunda, foram registradas respostas predominantemente conformes, com ressalva relacionada à exposição de informações nas justificativas geradas por IA. Na terceira, foi mantida uma divergência sobre privacidade durante o desenvolvimento. Na quarta, todas as respostas aplicáveis foram configuradas como conformes.

Durante o preenchimento, uma resposta conforme exigiu evidência, enquanto uma resposta não conforme exigiu justificativa. O rascunho manteve o questionário editável e a submissão final bloqueou alterações. O encerramento consolidou os resultados e atualizou os painéis. A Figura 18 apresenta o registro de uma dessas operações.

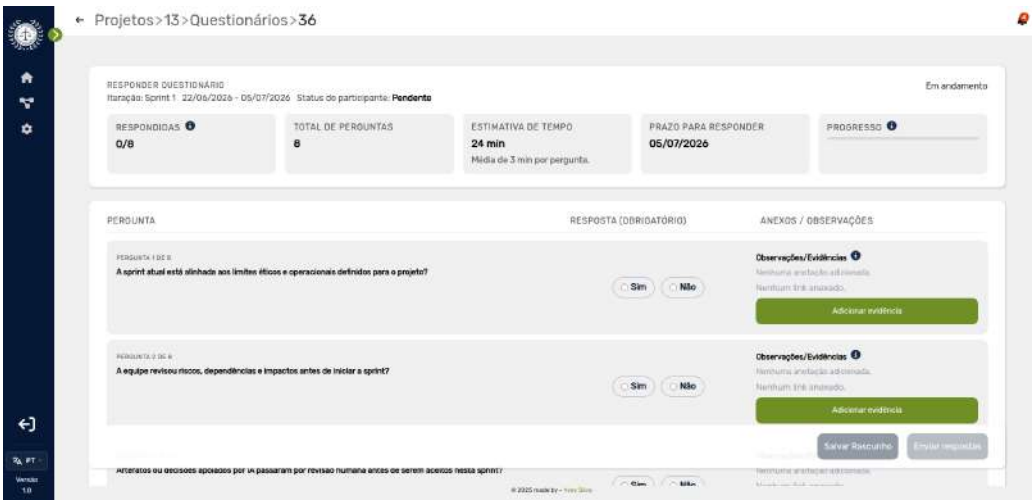

Figura 18 – Registro de resposta e justificativa durante a execução do cenário.

Fonte: O autor.

A captura evidencia a aplicação da regra condicional de preenchimento no cenário. Sua existência não valida a correção do conteúdo informado; demonstra somente que a resposta e seu complemento foram registrados e puderam alimentar as etapas posteriores do

fluxo. Os testes funcionais detalhados no Apêndice D registram os comportamentos esperados e observados, incluindo validações, rascunho, encerramento, cálculos, documentos, exportações e uso assistivo de IA.

6.4 Resultados dos Indicadores

Após o encerramento dos quatro questionários, a aplicação produziu os valores apresentados no Quadro 13. Os cálculos utilizaram as Equações 4.1 a 4.4, definidas na Seção 4.4, sem participação do modelo de linguagem.

Quadro 13 – Matriz dos resultados consolidados por *sprint*.

Métrica	***Sprint* 1**	***Sprint* 2**	***Sprint* 3**	***Sprint* 4**
Peso	1,0	1,0	1,5	2,0
ISEQ	5,36%	92,14%	80,00%	100,00%
Faixa	E	A	B	A
Desvio-padrão	10,20%	19,39%	40,00%	0,00%

Fonte: O autor.

Os valores obtidos nas quatro *sprints* foram 5,36%, 92,14%, 80,00% e 100,00%, respectivamente. Essa sequência não representa uma evolução observada em uma equipe real, pois as respostas foram elaboradas para produzir situações contrastantes no cenário simulado. O resultado demonstra que o indicador responde às alterações nas entradas e aos pesos configurados, diferenciando um cenário crítico de cenários com maior número de respostas conformes.

A primeira *sprint* foi classificada na faixa E, com ISEQ de 5,36%, evidenciando o comportamento esperado do sistema diante de respostas predominantemente negativas. O desvio-padrão de 10,20% indica que, embora os resultados individuais tenham permanecido próximos do nível crítico configurado, não foram idênticos.

Na segunda *sprint*, o ISEQ de 92,14% foi classificado na faixa A. Entretanto, o desvio-padrão de 19,39% e a presença de um participante na faixa D mostram que um indicador agregado elevado pode coexistir com percepções individuais desfavoráveis. Portanto, a leitura do ISEQ isoladamente seria insuficiente para caracterizar o resultado.

A terceira *sprint* reforça essa constatação. Embora tenha alcançado 80,00% e permanecido na faixa B, apresentou desvio-padrão de 40,00%, o maior do cenário. A dispersão decorreu da diferença entre as respostas dos papéis simulados, particularmente na questão relativa ao tratamento de dados e à privacidade. Nesse caso, o valor consolidado indica atendimento ao limiar configurado, mas a divergência interna sinaliza uma questão que deveria ser examinada pelos responsáveis pelo projeto.

A quarta *sprint* atingiu 100,00% e desvio-padrão nulo porque todas as respostas aplicáveis foram configuradas como conformes. Esse resultado verifica o limite superior do cálculo, mas não deve ser interpretado como comprovação de comportamento eticamente perfeito. Em uma aplicação real, a validade dessa classificação dependeria da qualidade das evidências, da independência da revisão e da adequação dos critérios ao contexto organizacional.

O projeto alcançou ISEP de 75,91%, correspondente à faixa B e ao limiar mínimo configurado para o cenário. Esse valor foi influenciado pelos pesos atribuídos às iterações: os resultados mais elevados das etapas finais receberam maior participação no cálculo e compensaram o resultado crítico da primeira *sprint*. O funcionamento confirma que a plataforma aplica a ponderação definida, mas também revela que o resultado final depende das escolhas de configuração. Pesos e limiares diferentes poderiam produzir outra classificação para as mesmas respostas. Por esse motivo, esses parâmetros devem ser transparentes, justificáveis e registrados na trilha de auditoria. O desvio-padrão consolidado da equipe foi de 37,64%, o que reforça a necessidade de examinar a composição do resultado, e não apenas seu valor final.

6.5 Painéis e Visualizações Analíticas

6.5.1 Painel do Questionário

O painel do questionário reúne ISEQ, média simples, desvio-padrão, distribuição por faixas e visualizações complementares. A Figura 19 apresenta a síntese da *Sprint* 2 utilizada para examinar se o resultado agregado permanecia coerente com os dados processados.

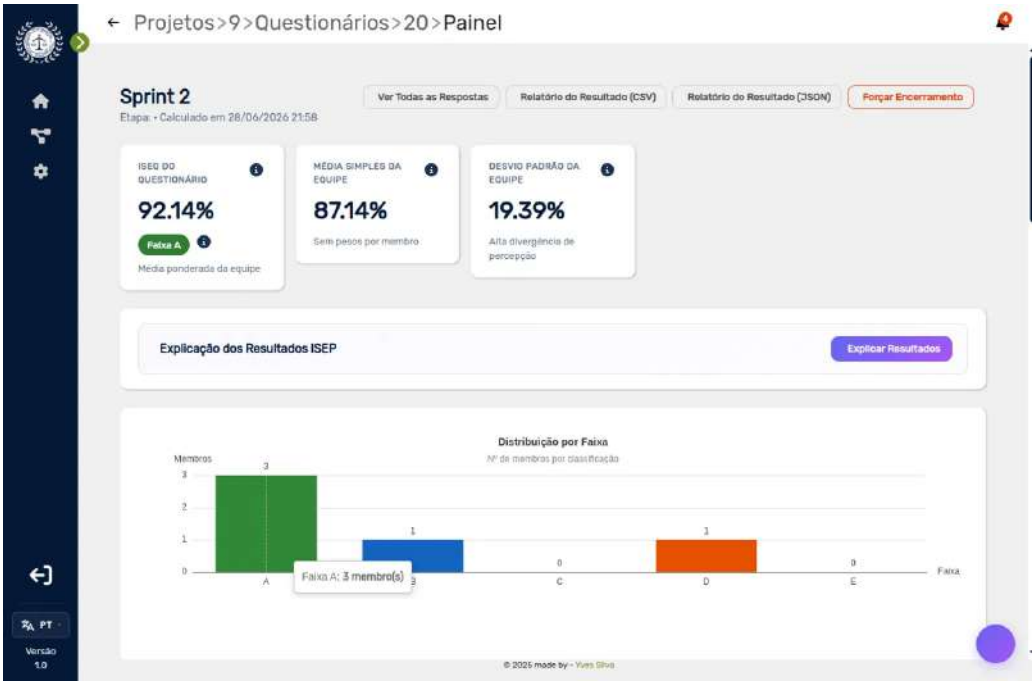

Figura 19 – Síntese dos indicadores do questionário referente à *Sprint* 2.

Fonte: O autor.

A visualização confirma que o ISEQ de 92,14% foi apresentado em conjunto com a faixa e as medidas descritivas. Contudo, o valor agregado não permite identificar isoladamente qual participante ou dimensão produziu a divergência observada.

A Figura 20 detalha os resultados individuais que compõem o mesmo questionário.

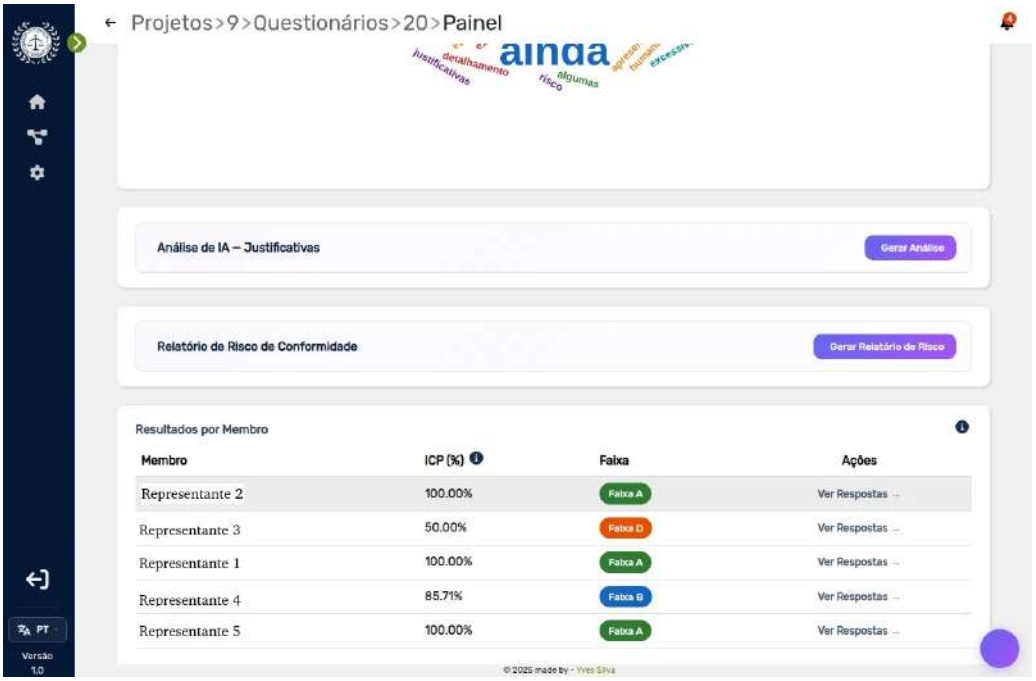

Figura 20 – Resultados individuais que compõem o ISEQ da *Sprint* 2.

Fonte: O autor.

A presença de um participante na faixa D, mesmo com o questionário classificado na faixa A, demonstra que a consolidação pode ocultar avaliações individuais desfavoráveis. A contribuição da visão detalhada consiste em tornar essa composição examinável; ela não determina, sem a leitura das respostas e das evidências, a causa ou a validade da divergência.

O mapa de calor foi examinado para localizar os cruzamentos de papéis e etapas associados às diferenças registradas. A Figura 21 apresenta essa distribuição.

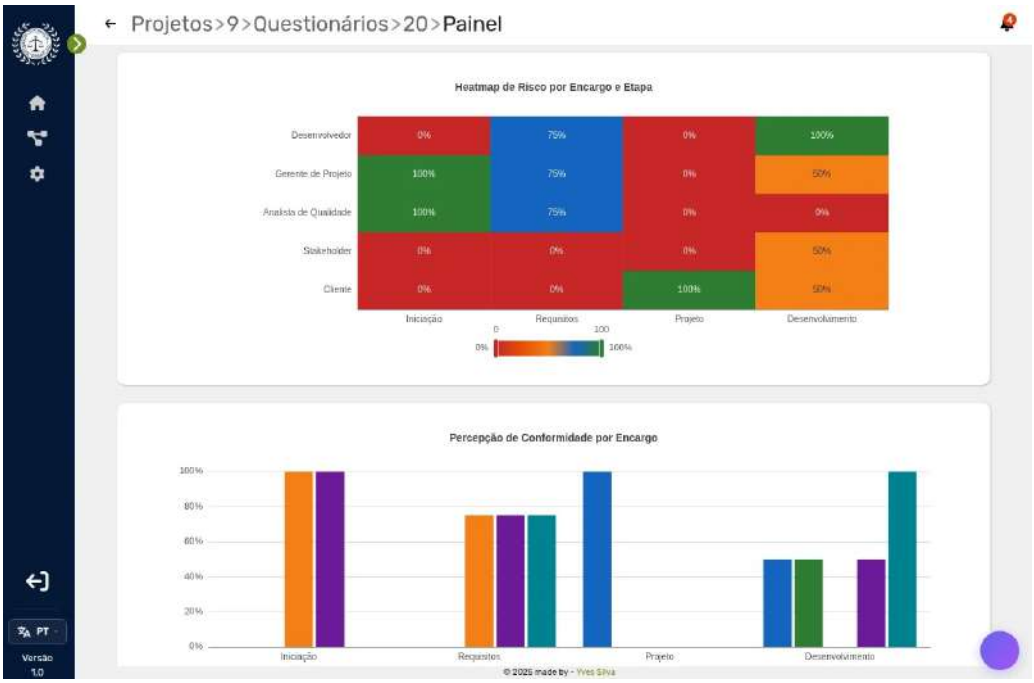

Figura 21 – Distribuição da conformidade por papel e etapa no mapa de calor.

Fonte: O autor.

O mapa de calor permitiu localizar as divergências que permaneceriam ocultas em uma leitura exclusiva do ISEQ. No cenário, os percentuais mais baixos concentraram-se em determinados cruzamentos de papéis e etapas, enquanto papéis técnicos apresentaram resultados superiores em outras dimensões. A contribuição dessa visualização não está em provar a existência de um risco organizacional real, mas em demonstrar como a plataforma pode direcionar a investigação para uma combinação específica de papel, etapa e pergunta. Em uso real, a identificação visual precisaria ser complementada pela análise das justificativas, das evidências e do contexto em que as respostas foram fornecidas.

A nuvem de palavras foi utilizada para explorar os termos recorrentes nas justificativas de não conformidade. A Figura 22 apresenta o resultado dessa agregação textual.

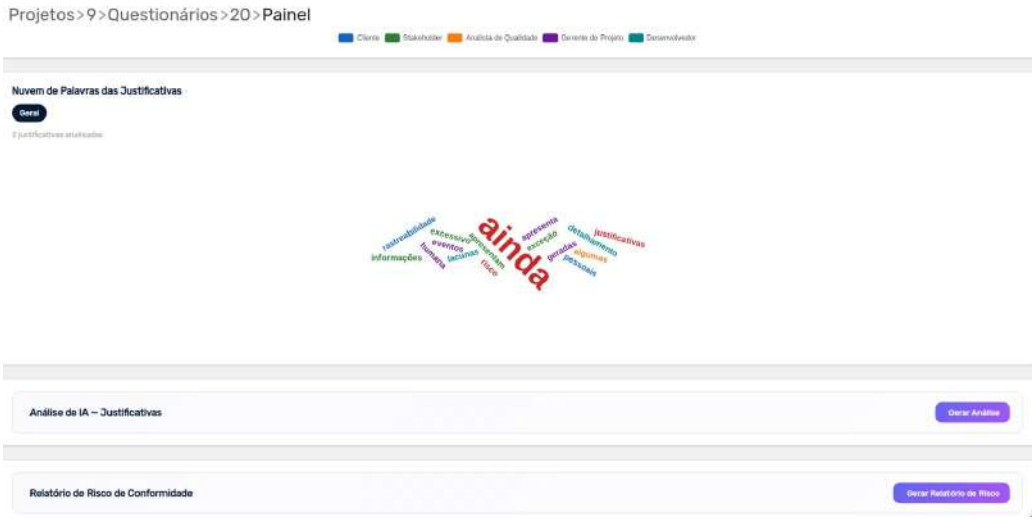

Figura 22 – Termos recorrentes nas justificativas de não conformidade.

Fonte: O autor.

A nuvem de palavras funcionou como mecanismo inicial de exploração do conteúdo textual. Entretanto, frequência não equivale a relevância ética, e palavras isoladas podem perder o contexto da resposta original. Assim, a visualização deve servir como ponto de partida para a consulta às justificativas completas, e não como conclusão autônoma.

6.5.2 Painel Individual do Participante

O painel individual foi consultado para verificar a apresentação do ICP, da evolução e do comparativo anônimo com a equipe. A Figura 23 apresenta essa visão.

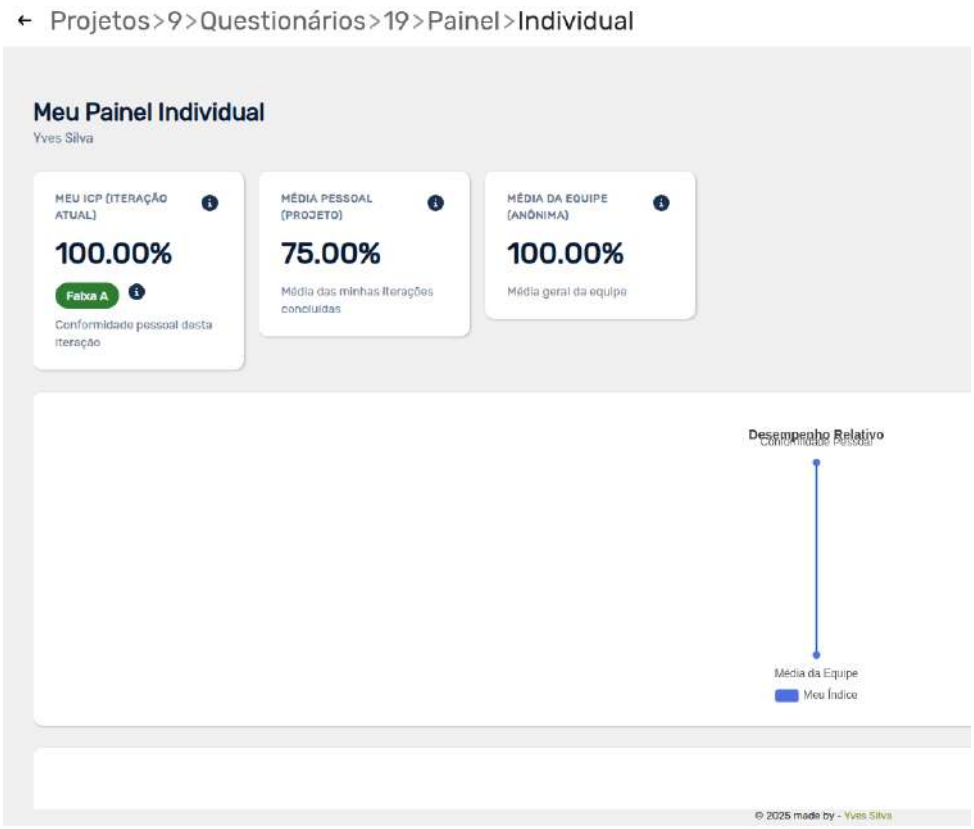

Figura 23 – Indicadores disponibilizados ao participante no cenário simulado.

Fonte: O autor.

A separação da visão individual permite ao participante acompanhar seu resultado sem exibir diretamente os valores nominais dos demais. A demonstração verifica a existência dessa restrição na interface utilizada, mas não substitui uma avaliação específica de privacidade ou de controle de acesso.

6.5.3 Painel Consolidado do Projeto

O painel consolidado foi examinado para verificar a reunião do ISEP, da faixa, do progresso dos questionários, da média da equipe, do desvio-padrão e da evolução por *sprint*. A Figura 24 registra o resultado final do cenário.

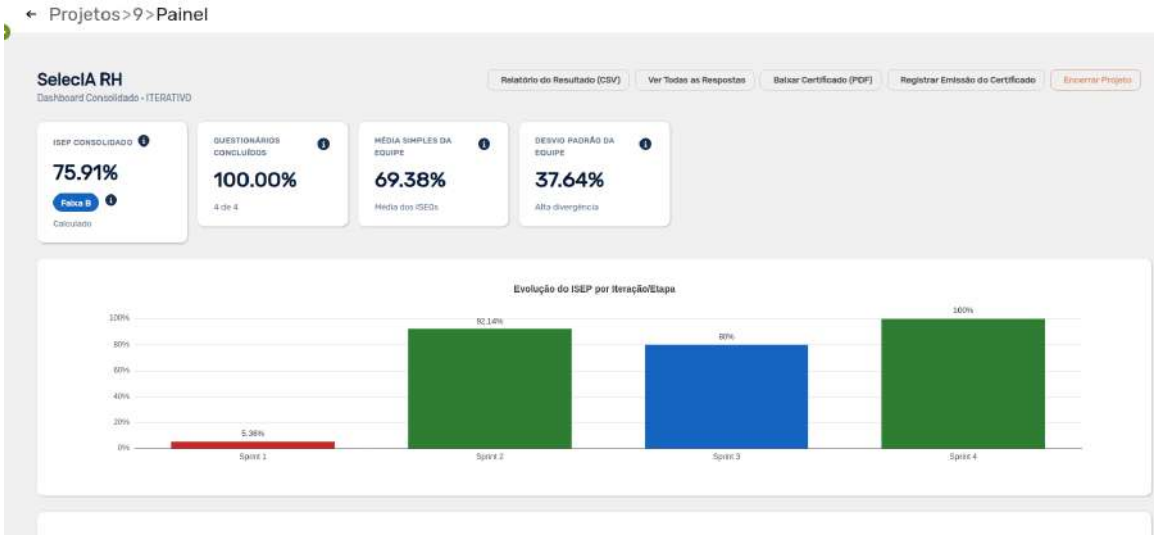

Figura 24 – Resultado consolidado do projeto SelecIA RH.

Fonte: O autor.

O ISEP de 75,91% aparece em conjunto com a evolução das quatro *sprints* e com a dispersão da equipe, permitindo relacionar o valor final à sua composição. A visualização torna perceptível que as etapas finais de maior peso compensaram o resultado crítico inicial. Ela não elimina a dependência dos parâmetros configurados nem transforma o índice em avaliação independente da adequação ética do projeto.

6.5.4 Interpretação Assistida por Inteligência Artificial

O painel assistivo foi acionado depois dos cálculos para verificar a geração de explicações, sínteses de justificativas e relatórios textuais de risco. A Figura 25 apresenta uma das respostas produzidas durante a execução.

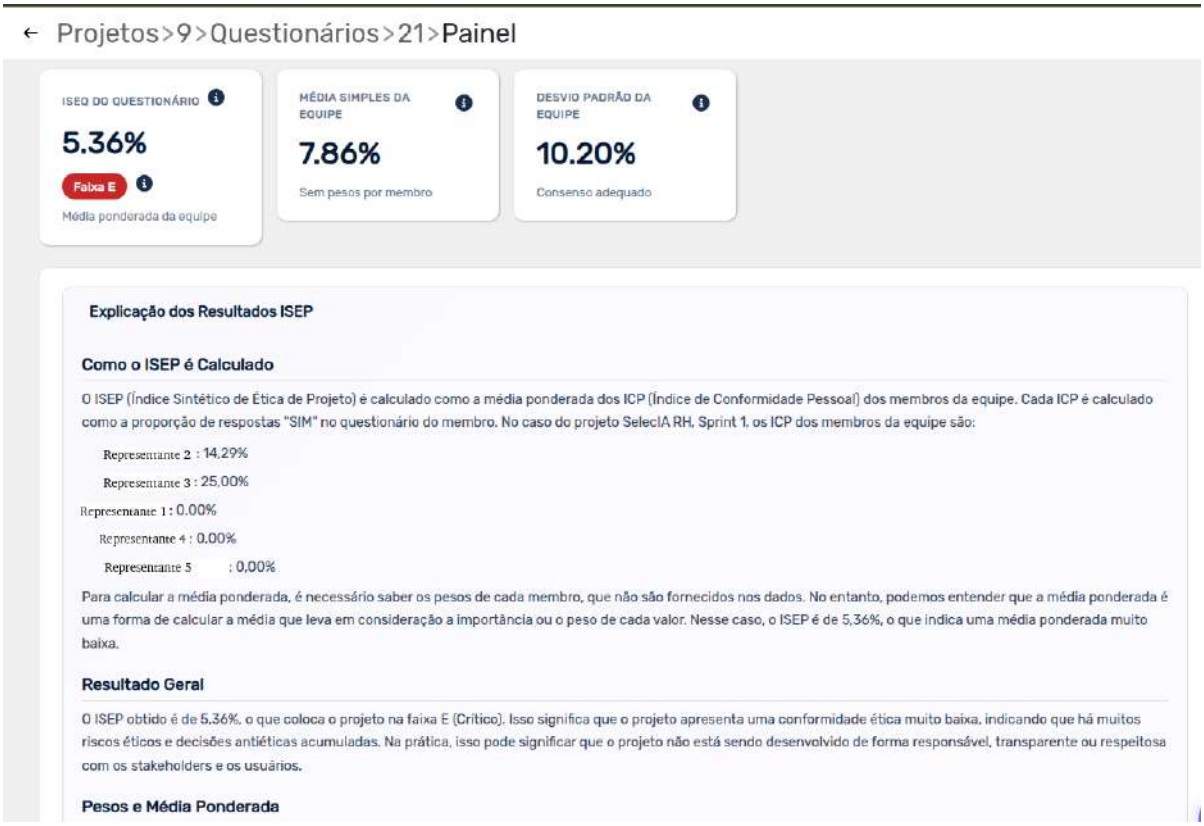

Figura 25 – Interpretação assistida dos indicadores do cenário simulado.

Fonte: O autor.

As explicações geradas acrescentaram uma camada textual aos indicadores calculados pela aplicação. No cenário, o recurso relacionou valores baixos às respostas negativas e destacou questões residuais de privacidade e explicabilidade. Contudo, não foi realizada uma avaliação externa da correção, completude ou utilidade dessas interpretações. Os textos gerados devem ser apresentados como apoio à leitura humana, sem autoridade para alterar respostas, indicadores ou classificações. Como a saída do modelo pode variar ou falhar, ela não constitui resultado reproduzível equivalente aos cálculos determinísticos.

6.6 Documentos Gerados e Trilha de Auditoria

A emissão do Boletim de Não Conformidade Ética na primeira *sprint* e do Certificado de Ciência Ética e Conformidade Declarada ao final do projeto demonstra a execução das regras condicionais associadas às faixas de classificação. A interface de consulta aos documentos foi apresentada na Figura 14; a Figura 26 apresenta o certificado produzido para o resultado final do cenário.

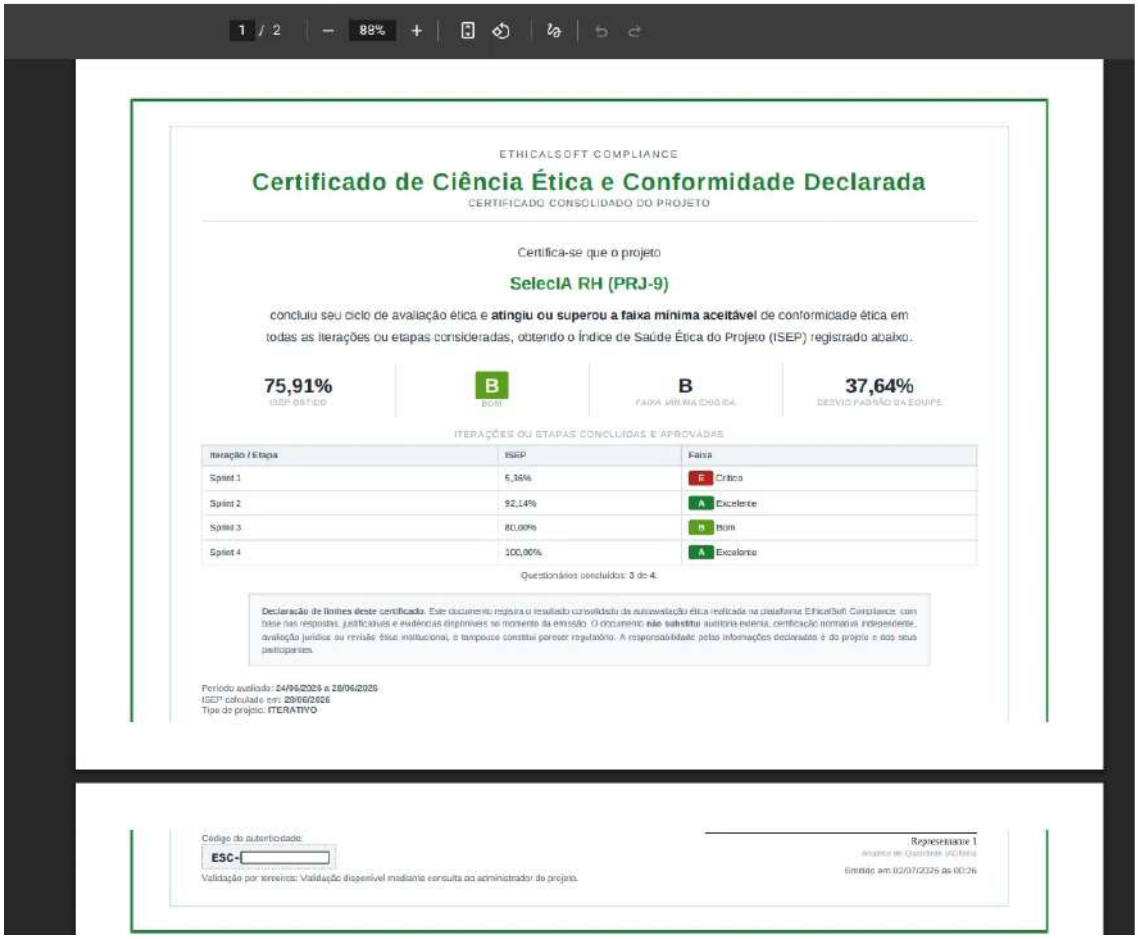

Figura 26 – Certificado emitido para o resultado final do cenário simulado.

Fonte: O autor.

Os documentos preservam valores, datas e identificadores relacionados à avaliação, contribuindo para a rastreabilidade. Entretanto, sua emissão não constitui certificação legal ou garantia de que o *software* seja eticamente adequado. Ela registra o resultado declarado segundo os critérios, pesos e evidências configurados na plataforma.

As emissões registraram código de autenticidade, tipo, escopo, data, usuário emissor, índice, faixa, versão do *template* e *hash* dos dados. Para examinar a recuperação das informações que fundamentaram os resultados, foi utilizada a tela de respostas consolidadas apresentada na Figura 27.

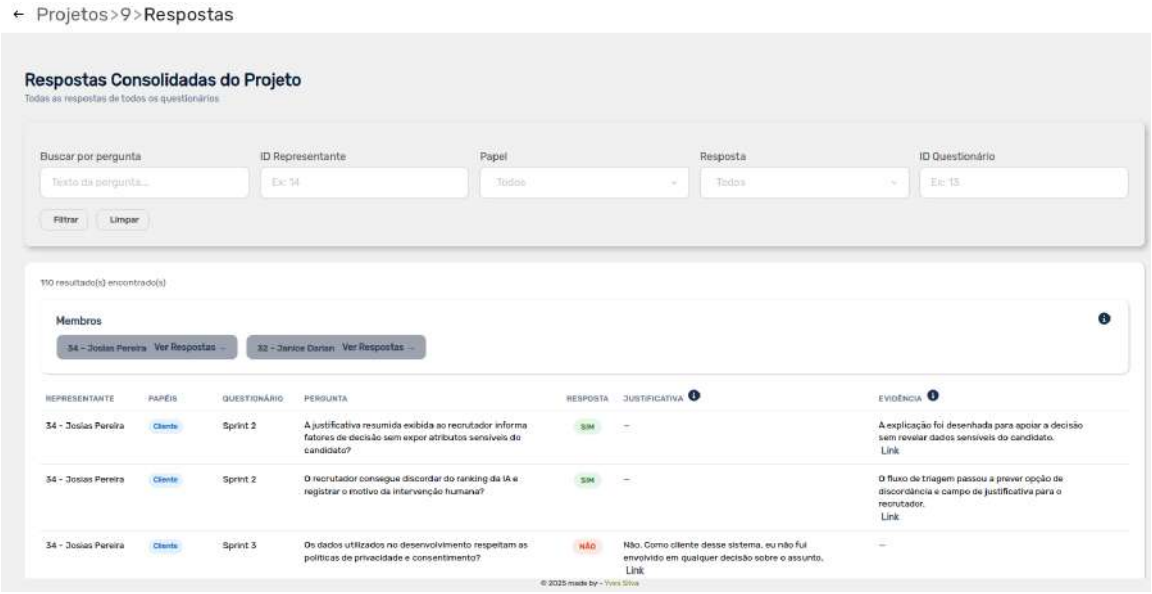

Figura 27 – Consulta às respostas, justificativas e evidências consolidadas.

Fonte: O autor.

A consulta permite recuperar os registros por participante, papel, pergunta e questionário e relacioná-los aos resultados e documentos emitidos. Essa capacidade favorece a reconstrução do percurso declaratório, mas não verifica automaticamente a autenticidade ou a suficiência das evidências anexadas.

6.7 Discussão dos Resultados

Os resultados mostram que princípios e perguntas éticas podem ser operacionalizados em um processo recorrente de registro, cálculo e acompanhamento. A plataforma associa questões a papéis e etapas, exige justificativas ou evidências, preserva resultados individuais e consolidados e registra os documentos emitidos. Desse modo, o artefato aproxima os quadros éticos apresentados na fundamentação de mecanismos de governança como rastreabilidade, monitoramento e prestação de contas (LURIE; MARK, 2016; BIABLE; GARCIA; MIDEKSO, 2023; ISACA, 2018; AXELOS, 2019).

Em resposta à questão de pesquisa, o cenário demonstra que uma plataforma *web* pode operacionalizar critérios de autoavaliação ética ao transformar perguntas configuradas em registros vinculados a projeto, etapa, papel e participante, processá-los por regras determinísticas e preservar os resultados e as emissões correspondentes. A principal contribuição observada não é determinar automaticamente se um projeto é ético, mas oferecer uma estrutura para tornar explícitas as avaliações, divergências e evidências produzidas ao longo do desenvolvimento.

O cenário demonstrou, por exemplo, que um resultado agregado favorável pode ocultar avaliações individuais negativas. A apresentação simultânea do ISEQ, do desvio-padrão,

da distribuição por faixa e do mapa de calor reduz esse problema ao permitir que o responsável examine a composição do resultado. Essa combinação operacionaliza, no contexto do artefato, a distribuição de reflexões ao longo do ciclo defendida pelo EDSD e as categorias éticas associadas aos requisitos discutidas por Biable, Garcia e Midekso (2023), sem validar cientificamente esses modelos originais.

A exigência de evidência e justificativa favorece a rastreabilidade, mas não elimina a natureza autodeclaratória das respostas. A plataforma registra o conteúdo fornecido; ela não verifica automaticamente sua autenticidade, suficiência ou conformidade jurídica. Consequentemente, o uso organizacional exigiria responsáveis pela revisão, critérios de aceitação e procedimentos para contestação ou correção dos registros.

Sob a perspectiva da governança, respostas, pesos, justificativas, registros de emissão e painéis tornam visíveis informações que poderiam permanecer dispersas. Essa característica se relaciona à transparência, à prestação de contas e ao monitoramento discutidos no COBIT 2019 e no ITIL 4 (ISACA, 2018; AXELOS, 2019). O alinhamento observado é conceitual e funcional; não foi realizada avaliação organizacional baseada nesses referenciais.

Os resultados obtidos sustentam a viabilidade técnica do mecanismo proposto no cenário especificado. Não sustentam conclusões sobre mudança de comportamento profissional, prevenção de incidentes, melhoria da cultura ética ou aceitação da ferramenta. Essas questões dependem de avaliações com participantes externos, projetos reais e acompanhamento longitudinal.

6.8 Limitações das Evidências

A avaliação utilizou um único cenário fictício, respostas sintéticas e perfis representados pelo próprio pesquisador. Não houve participação de equipes reais, comparação com outro processo de avaliação, teste de usabilidade com usuários externos ou acompanhamento longitudinal. Os resultados demonstram a execução das regras e a coerência entre entradas, indicadores, painéis e documentos, mas não podem ser generalizados para organizações ou projetos reais. Também permanecem dependentes dos pesos, limiares, perguntas e evidências configurados.

Não foram mensurados incidentes éticos antes e depois do uso, mudança de comportamento, aceitação, desempenho sob carga ou efetividade das notificações. Não foram executados testes automatizados de integração nem suítes abrangentes de contrato de API, automação de interface de ponta a ponta, avaliação formal de usabilidade, ensaios de desempenho sob carga, testes de penetração ou aceitação com usuários externos. Assim, os resultados demonstram o comportamento isolado dos componentes e os fluxos funcionais manuais examinados, mas não autorizam concluir que o sistema esteja integralmente coberto, livre de defeitos, seguro sob ataque ou validado em contexto organizacional real.

A integração externa de IA produz saídas variáveis e não teve sua correção ou utilidade avaliada por especialistas ou usuários externos.

Essas restrições não anulam a demonstração funcional, mas delimitam a conclusão ao funcionamento e à coerência interna do artefato no cenário especificado. A aplicação dos mesmos procedimentos em equipes reais, com critérios revisados por especialistas e observação longitudinal, é necessária para investigar efetividade, validade externa e impacto organizacional.

7 CONCLUSÃO

Este trabalho apresentou o EthicalSoft Compliance, plataforma *web* destinada à autoavaliação ética, à rastreabilidade de decisões e à geração de evidências de conformidade declarada em projetos de *software*. O artefato reuniu questionários configuráveis, indicadores determinísticos, painéis, notificações, documentos, registros de emissão e apoio interpretativo por inteligência artificial.

A demonstração estruturada no cenário simulado SelecIA RH permitiu exercitar o fluxo desde a configuração do projeto até a emissão dos documentos. Os resultados sustentam a conclusão de que a versão examinada executa os comportamentos demonstrados e aplica as regras configuradas. Por outro lado, não permitem afirmar que a plataforma melhora o comportamento ético de equipes, reduz incidentes ou produz conformidade legal ou normativa.

7.1 Atendimento aos Objetivos Específicos

O Quadro 14 retoma OE1 a OE10 e apresenta a classificação, a evidência e a limitação associada a cada objetivo.

Quadro 14 – Atendimento aos objetivos específicos.

Objetivo	**Situação**	**Evidência**	**Limitação**
OE1	Totalmente atendido	Questionários organizados por etapa, iteração, papel e domínio, documentados na Seção 6.3 e nos apêndices.	O conjunto de perguntas não foi validado externamente.
OE2	Totalmente atendido	Configuração de pesos, papéis, etapas, perguntas e escopos de IA, demonstrada na Seção 6.2.	Não foram avaliadas configurações de organizações reais.
OE3	Totalmente atendido	ICP, ISEQ, ISEP e IEM definidos na Seção 4.4 e executados no cenário.	Os indicadores são operacionais e não constituem medida universal de comportamento ético.

Continua na próxima página...

Continuação do Quadro 14

Objetivo	**Situação**	**Evidência**	**Limitação**
OE4	Totalmente atendido	Painéis de questionário, participante e projeto apresentados na Seção 6.5.	Não houve avaliação formal de usabilidade ou acessibilidade.
OE5	Totalmente atendido	Boletim gerado para a *Sprint* 1 e registrado entre os documentos do cenário.	A utilidade das orientações do boletim não foi avaliada por usuários externos.
OE6	Totalmente atendido	Certificado emitido após o ISEP atingir o limiar configurado, conforme Seção 6.6.	O documento representa conformidade declarada e não certificação independente.
OE7	Totalmente atendido	Registro de emissão com código, escopo, data, usuário, índice, faixa, versão e *hash*.	Não foi conduzida auditoria externa do mecanismo de autenticidade.
OE8	Parcialmente atendido	Notificações e registros de emissão foram implementados e exercitados.	Não foi demonstrada uma trilha única e completa para todas as categorias de evento do projeto.
OE9	Parcialmente atendido	Funcionalidades foram relacionadas a transparência, monitoramento e prestação de contas na fundamentação e na discussão.	Não houve avaliação organizacional formal baseada no COBIT 2019 ou no ITIL 4.
OE10	Totalmente atendido	Demonstração reprodutível com quatro *sprints*, cinco papéis e resultados calculados, apresentada nas Seções 4.5 e 6.4.	A avaliação utilizou dados sintéticos e não mediu efetividade com equipes reais.

Fonte: O autor.

Assim, oito objetivos foram classificados como totalmente atendidos e dois como parcialmente atendidos. Nenhum objetivo foi classificado como não atendido dentro do escopo delimitado. As classificações parciais decorrem da ausência de trilha abrangente para todos os eventos e da inexistência de avaliação organizacional formal dos referenciais de governança.

7.2 Contribuições do Trabalho

A primeira contribuição consiste na operacionalização de elementos do EDSD e do quadro de ética para engenharia de requisitos em instrumentos configuráveis vinculados a etapas e papéis. Essa operacionalização aproxima princípios conceituais das rotinas de um projeto sem atribuir à plataforma autoridade para decidir o que é moralmente correto.

A segunda contribuição é o modelo técnico de registro e consolidação. Respostas, justificativas, evidências, pesos e resultados são relacionados em níveis de participante, questionário, etapa e projeto. Os painéis e documentos tornam esses registros acessíveis para acompanhamento e prestação de contas.

A terceira contribuição é a separação entre cálculo determinístico e interpretação assistiva. Os indicadores permanecem disponíveis independentemente do modelo de linguagem, enquanto a IA oferece explicações contextuais sem alterar respostas, pesos ou classificações.

A quarta contribuição é a implementação de um artefato reproduzível, acompanhado por requisitos, casos de uso, regras de negócio, testes, cenário, contrato resumido de API e instruções de execução.

7.3 Limitações

A principal limitação é a utilização de um único cenário simulado, com dados elaborados pelo pesquisador e sem participantes externos. Não foram avaliadas utilidade percebida, aceitação, mudança de comportamento ou redução de incidentes em equipes reais.

Os indicadores dependem das perguntas, dos pesos, dos papéis, das faixas e do limiar configurado. Eles representam conformidade declarada dentro do modelo da plataforma e não certificação moral, legal ou normativa. Também não foi realizada validação por especialistas das perguntas e dos parâmetros adotados.

Não foram executados testes automatizados de integração nem suítes abrangentes de contrato de API, automação de interface de ponta a ponta, avaliação formal de usabilidade, ensaios de desempenho sob carga, testes de penetração ou aceitação com usuários externos. Assim, os resultados demonstram o comportamento isolado dos componentes e os fluxos funcionais manuais examinados, mas não autorizam concluir que o sistema esteja integralmente coberto, livre de defeitos, seguro sob ataque ou validado em contexto organizacional real. A persistência poliglota acrescenta complexidade operacional e exige cuidados com consistência, *backup* e recuperação. A interpretação por IA depende de serviço externo e pode variar entre execuções.

7.4 Trabalhos Futuros

Como continuidade deste trabalho, recomenda-se:

a. conduzir estudos com equipes reais, utilizando critérios de aceitação definidos antes da execução e observação longitudinal;

b. submeter os instrumentos e os indicadores à avaliação de especialistas em ética, engenharia de *software* e governança de TI;

c. avaliar usabilidade, acessibilidade, desempenho, segurança e aceitação da plataforma;

d. ampliar a cobertura automatizada e incluir testes de integração dos cálculos, documentos, bancos de dados e serviços externos;

e. implementar uma trilha unificada para eventos de configuração, resposta, encerramento, cálculo, alteração e emissão;

f. avaliar formalmente o alinhamento do artefato com objetivos do COBIT 2019 e práticas do ITIL 4 em contexto organizacional;

g. comparar configurações e resultados entre projetos, equipes e domínios de risco, sem transformar os indicadores em *ranking* descontextualizado;

h. revisar a necessidade da persistência poliglota diante dos custos de operação e das alternativas documentais disponíveis no PostgreSQL;

i. ampliar os critérios relacionados ao desenvolvimento assistido por IA conforme novas evidências e práticas sejam consolidadas;

j. investigar mecanismos para registrar a origem, a revisão e o destino de sugestões produzidas por modelos de linguagem.

Referências

ABDULHALIM, H.; LURIE, Y.; MARK, S. Ethics as a quality driver in agile software projects. **Journal of Service Science and Management**, v. 11, n. 1, p. 13–25, 2018. DOI: <https://doi.org/10.4236/jssm.2018.111002>.

ACM; IEEE COMPUTER SOCIETY. **Software Engineering Code of Ethics and Professional Practice**. [S.l.], 1999. Versão 5.2. Disponível em: <https://www.computer.org/education/code-of-ethics>. Acesso em: 22 jul. 2026.

ALAMI, A.; ZAHEDI, M.; KRANCHER, O. The role of psychological safety in promoting software quality in agile teams. **Empirical Software Engineering**, v. 29, n. 5, 2024. Artigo 119. DOI: <https://doi.org/10.1007/s10664-024-10512-1>.

ANGULAR. **Dependency injection in Angular**. 2026. Disponível em: <https://angular.dev/guide/di>. Acesso em: 22 jul. 2026.

ANGULAR. **What is Angular?** 2026. Disponível em: <https://angular.dev/overview>. Acesso em: 22 jul. 2026.

AXELOS. **ITIL foundation: ITIL 4 edition**. London: TSO, 2019.

BIABLE, S. E.; GARCIA, V. C.; MIDEKSO, D. Proposed ethical framework for software requirements engineering. **IET Software**, v. 17, p. 526–537, 2023. DOI: <https://doi.org/10.1049/sfw2.12136>.

BRASIL. **Constituição da República Federativa do Brasil de 1988**. Brasília, DF: Presidência da República, 1988. Disponível em: <https://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm>. Acesso em: 22 jul. 2026.

BRASIL. CONSELHO NACIONAL DE EDUCAÇÃO. CÂMARA DE EDUCAÇÃO SUPERIOR. Resolução nº 5, de 16 de novembro de 2016: institui as diretrizes curriculares nacionais para os cursos de graduação na área da computação. **Diário Oficial da União**, Brasília, DF, n. 220, p. 22–24, 2016. Disponível em: <https://www.gov.br/mec/pt-br/cne/pdf/normas-classificadas-por-assunto/diretrizes-curriculares-cursos-de-graduacao/rces005_16.pdf>. Acesso em: 22 jul. 2026.

BRASIL. MINISTÉRIO DO TRABALHO E EMPREGO. **Classificação Brasileira de Ocupações**. Brasília, DF: Ministério do Trabalho e Emprego, 2023. Disponível em: <https://www.gov.br/trabalho-e-emprego/pt-br/servicos/trabalhador/mais-acoes/classificacao-brasileira-de-ocupacoes>. Acesso em: 22 jul. 2026.

BRASIL. MINISTÉRIO DO TRABALHO E EMPREGO. **Regulamentação: listagem das profissões regulamentadas**. Brasília, DF: Ministério do Trabalho e Emprego, 2023. Disponível em: <https://www.gov.br/trabalho-e-emprego/pt-br/assuntos/cbo/regulamentacao>. Acesso em: 22 jul. 2026.

CARVALHO, L. P. et al. Ethics: what is the Brazilian software engineering research scenario? **Journal of Software Engineering Research and Development**, v. 12, n. 1, p. 2:1–2:18, 2024. DOI: <https://doi.org/10.5753/jserd.2024.3395>.

CHEN, A. W. Ethical perception and job satisfaction of software developers. **Online Journal of Applied Knowledge Management**, v. 12, n. 1, p. 34–53, 2024. DOI: <https://doi.org/10.36965/OJAKM.2024.12(1)34-53>.

COCKBURN, A. **Hexagonal architecture**. 2005. Disponível em: <https://alistair.cockburn.us/hexagonal-architecture>. Acesso em: 27 jun. 2026.

CONSELHO FEDERAL DE ENGENHARIA E AGRONOMIA. **Resolução nº 1.156, de 24 de outubro de 2025: consolida normativos acerca das atividades e competências profissionais dos engenheiros da modalidade eletricista**. Brasília, DF: Confea, 2025. Disponível em: <https://normativos.confea.org.br/Ementas/Visualizar?id=82360>. Acesso em: 22 jul. 2026.

CONSELHO FEDERAL DE ENGENHARIA E AGRONOMIA. **Ética profissional**. Brasília, DF: Confea, 2025. Disponível em: <https://www.confea.org.br/atuacao/etica-profissional>. Acesso em: 22 jul. 2026.

DOCKER. **How Compose works**. 2026. Disponível em: <https://docs.docker.com/compose/intro/compose-application-model/>. Acesso em: 22 jul. 2026.

GHANBARI, H. Technical debt, a double-edged sword that can damage moral responsibility. **Journal of Information Technology Teaching Cases**, p. 1–7, 2024. Publicação antecipada online. DOI: <https://doi.org/10.1177/20438869241279439>.

GHANBARI, H.; VARTIAINEN, T.; SIPONEN, M. T. Omission of quality software development practices: a systematic literature review. **ACM Computing Surveys**, v. 51, n. 3, p. 1–27, 2018. DOI: <https://doi.org/10.1145/3177746>.

GIBSON, J. P. et al. Technical debt is an ethical issue. In: **ETHICOMP 2022: Effectiveness of ICT ethics**. Turku, Finland: [s.n.], 2022. p. 258–266. DOI: <https://doi.org/10.21427/Y73Z-6P82>.

GIL, A. C. **Como elaborar projetos de pesquisa**. 7. ed. São Paulo: Atlas, 2022.

GOMES, T. B. S.; LOIOLA, D. A. de M.; SANTOS, A. R. Technical debt tools: a survey and an empirical evaluation. **Journal of Software Engineering Research and Development**, v. 12, n. 1, p. 8:1–8:16, 2024. DOI: <https://doi.org/10.5753/jserd.2024.3591>.

GOTTERBARN, D. Codes of ethics: the conscience of a profession: connecting technology and society. **ACM Inroads**, v. 7, n. 4, p. 33–35, 2016. DOI: <https://doi.org/10.1145/3003437>.

GUBER, J.; REINHARTZ-BERGER, I. Privacy-compliant software reuse in early development phases: a systematic literature review. **Information and Software Technology**, v. 157, p. 107351, 2024. DOI: <https://doi.org/10.1016/j.infsof.2023.107351>.

HALME, E. et al. Ethical user stories: industrial study. In: **CEUR Workshop Proceedings**. [s.n.], 2022. v. 3122. Disponível em: <https://ceur-ws.org/Vol-3122/RE4AI-paper-1.pdf>. Acesso em: 20 jun. 2026.

HALME, E. et al. Making ethics practical: user stories as a way of implementing ethical consideration in software engineering. **Information and Software Technology**, v. 167, p. 107379, 2024. DOI: <https://doi.org/10.1016/j.infsof.2023.107379>.

HARESAMUDRAM, K.; LARSSON, S.; HEINTZ, F. Three levels of AI transparency. **Computer**, v. 56, n. 2, p. 93–100, 2023. DOI: <https://doi.org/10.1109/MC.2022.3213181>.

HOU, X. et al. Large language models for software engineering: a systematic literature review. **ACM Transactions on Software Engineering and Methodology**, v. 33, n. 8, p. 1–79, 2024. DOI: <https://doi.org/10.1145/3695988>.

ISACA. **COBIT 2019 framework: governance and management objectives**. Schaumburg: ISACA, 2018.

JACOCO. **JaCoCo documentation**. 2026. Disponível em: <https://www.jacoco.org/jacoco/>. Acesso em: 27 jun. 2026.

JOHNSON, D. G. **Computer ethics**. 4. ed. Upper Saddle River: Pearson Prentice Hall, 2008.

JUNIT. **JUnit 5 user guide**. 2026. Disponível em: <https://junit.org/junit5/docs/current/user-guide/>. Acesso em: 27 jun. 2026.

KANSTRÉN, T. Towards a deeper understanding of test coverage. **Journal of Software Maintenance and Evolution: Research and Practice**, v. 20, n. 1, p. 59–76, 2008. DOI: <https://doi.org/10.1002/smr.362>.

KLETTKE, M. et al. NoSQL schema evolution and big data migration at scale. In: **Proceedings of the 2016 IEEE International Conference on Big Data**. Washington, DC: IEEE, 2016. p. 2764–2774. DOI: <https://doi.org/10.1109/BigData.2016.7840924>.

KOROBENKO, D.; NIKIFOROVA, A.; SHARMA, R. **Towards a privacy and security-aware framework for ethical AI: guiding the development and assessment of AI systems**. 2024. CoRR, abs/2403.08624. Disponível em: <https://arxiv.org/abs/2403.08624>. Acesso em: 1 jul. 2026.

LANDUYT, D. V. et al. A comparative performance evaluation of multi-model NoSQL databases and polyglot persistence. In: **Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing**. Tallinn: Association for Computing Machinery, 2023. p. 286–293. DOI: <https://doi.org/10.1145/3555776.3577645>.

LARMAN, C. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. Porto Alegre: Bookman, 2007.

LURIE, Y.; MARK, S. Professional ethics of software engineers: an ethical framework. **Science and Engineering Ethics**, v. 22, p. 1577–1591, 2016. DOI: <https://doi.org/10.1007/s11948-015-9665-x>.

MARTIN, R. C. **Arquitetura limpa: o guia do artesão para estrutura e design de software**. Rio de Janeiro: Alta Books, 2019.

MASIERO, P. C. **Ética em computação**. São Paulo: Edusp, 2008.

MEURICE, L.; CLEVE, A. Supporting schema evolution in schema-less NoSQL data stores. In: **Proceedings of the 24th IEEE International Conference on Software Analysis, Evolution and Reengineering**. Klagenfurt: IEEE, 2017. p. 457–461. DOI: <https://doi.org/10.1109/SANER.2017.7884653>.

MITCHELL, A. C.; BALASUBRAMANIAM, D.; FLETCHER, J. Incorporating ethics in software engineering: challenges and opportunities. In: **Asia-Pacific Software Engineering Conference (APSEC), 29.** Los Alamitos: IEEE Computer Society, 2022. p. 90–98. DOI: <https://doi.org/10.1109/APSEC57359.2022.00021>.

MOHAMED, A.; ASSI, M.; GUIZANI, M. The impact of LLM-assistants on software developer productivity: a systematic literature review. **arXiv preprint arXiv:2507.03156**, 2025. DOI: <https://doi.org/10.48550/arXiv.2507.03156>.

MONGODB. **Best Practices for Data Modeling in MongoDB**. 2026. Disponível em: <https://www.mongodb.com/docs/manual/data-modeling/best-practices/>. Acesso em: 22 jul. 2026.

MONGODB. **Data Modeling in MongoDB**. 2026. Disponível em: <https://www.mongodb.com/docs/manual/data-modeling/>. Acesso em: 22 jul. 2026.

MONGODB. **Schema Validation**. 2026. Disponível em: <https://www.mongodb.com/docs/manual/core/schema-validation/>. Acesso em: 22 jul. 2026.

NEIMAT, T. A. **Why IT projects fail**. Sydney: Project Perfect, 2005. Project Perfect White Paper Collection. Disponível em: <http://www.projectperfect.com.au/downloads/Info/info_it_projects_fail.pdf>. Acesso em: 1 jul. 2026.

NUNES, J. H.; SOUSA, T. P. de. Reconhecimento e profissionalismo no campo da tecnologia da informação. **Política & Trabalho: Revista de Ciências Sociais**, n. 46, p. 139–158, 2017. DOI: <https://doi.org/10.22478/UFPB.1517-5901.2017V1N46.27085>.

OFFUTT, A. J. Unit testing versus integration testing. In: **Proceedings of the 1991 IEEE International Test Conference: Test: Faster, Better, Sooner**. Nashville: IEEE Computer Society, 1991. p. 1108–1109.

OPENHTMLTOPDF. **openhtmltopdf**. 2024. Disponível em: <https://github.com/danfickle/openhtmltopdf>. Acesso em: 22 jul. 2026.

ORACLE. **Java Language Updates: Java SE 21**. 2023. Disponível em: <https://docs.oracle.com/en/java/javase/21/language/java-language-changes.html>. Acesso em: 22 jul. 2026.

PEFFERS, K. et al. A design science research methodology for information systems research. **Journal of Management Information Systems**, v. 24, n. 3, p. 45–77, 2007. DOI: <https://doi.org/10.2753/MIS0742-1222240302>.

PEZZÈ, M.; RUBINOV, K.; WUTTKE, J. Generating effective integration test cases from unit ones. In: **Proceedings of the 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation**. Luxembourg: IEEE Computer Society, 2013. p. 11–20. DOI: <https://doi.org/10.1109/ICST.2013.37>.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL 16 Documentation**. 2023. Disponível em: <https://www.postgresql.org/docs/16/>. Acesso em: 22 jul. 2026.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL 16 Documentation: JSON Types**. 2026. Disponível em: <https://www.postgresql.org/docs/16/datatype-json.html>. Acesso em: 22 jul. 2026.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL 16 Documentation: Transactions**. 2026. Disponível em: <https://www.postgresql.org/docs/16/tutorial-transactions.html>. Acesso em: 22 jul. 2026.

PRAT, N.; COMYN-WATTIAU, I.; AKOKA, J. A taxonomy of evaluation methods for information systems artifacts. **Journal of Management Information Systems**, v. 32, n. 3, p. 229–267, 2015. DOI: <https://doi.org/10.1080/07421222.2015.1099390>.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: uma abordagem profissional**. Porto Alegre: AMGH, 2021.

RUNESON, P.; HÖST, M. Guidelines for conducting and reporting case study research in software engineering. **Empirical Software Engineering**, v. 14, n. 2, p. 131–164, 2009. DOI: <https://doi.org/10.1007/s10664-008-9102-8>.

SCHWABER, K.; SUTHERLAND, J. **The Scrum Guide: the definitive guide to Scrum: the rules of the game**. [S.l.], 2020. Disponível em: <https://scrumguides.org/scrum-guide.html>. Acesso em: 22 jul. 2026.

SILVA, R. C. F. da; CARVALHO, J. de S. Por uma formação ética não prescritiva: agência e responsabilidade moral na Computação. In: **Anais da Conferência Latino-Americana de Ética em Inteligência Artificial**. Niterói: Sociedade Brasileira de Computação, 2024. p. 81–84. DOI: <https://doi.org/10.5753/laai-ethics.2024.32457>.

SILVEIRA, A. C. J. da; TONINI, A. M. Análise sobre a regulamentação do profissional do setor de tecnologia da informação e comunicação no Brasil. **Revista HISTEDBR On-line**, Campinas, v. 21, p. e021022, 2021. DOI: <https://doi.org/10.20396/rho.v21i00.8657662>.

SOMMERVILLE, I. **Engenharia de software**. 9. ed. São Paulo: Pearson Education do Brasil, 2011.

SPRING. **Spring Boot Reference Documentation 3.4**. 2025. Disponível em: <https://docs.spring.io/spring-boot/3.4/reference/index.html>. Acesso em: 22 jul. 2026.

SPRING AI. **Chat Model API**. 2025. Disponível em: <https://docs.spring.io/spring-ai/reference/api/chatmodel.html>. Acesso em: 22 jul. 2026.

SPRING AI. **Groq Chat**. 2025. Disponível em: <https://docs.spring.io/spring-ai/reference/api/chat/groq-chat.html>. Acesso em: 22 jul. 2026.

STAHL, B. C. et al. From computer ethics to responsible research and innovation in ICT. **Information & Management**, v. 51, n. 6, p. 810–818, 2014. DOI: <https://doi.org/10.1016/j.im.2014.01.001>.

STANDISH GROUP. **CHAOS Summary 2009**. [S.l.]: Standish Group International, 2009. Disponível em: <https://www.classes.cs.uchicago.edu/archive/2014/fall/51210-1/required.reading/Standish.Group.Chaos.2009.pdf>. Acesso em: 1 jul. 2026.

VAKKURI, V. et al. How do software companies deal with artificial intelligence ethics? A gap analysis. In: **Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering**. Gothenburg: ACM, 2022. p. 100–109. DOI: <https://doi.org/10.1145/3530019.3530030>.

VÁZQUEZ, A. S. **Ética**. Rio de Janeiro: Civilização Brasileira, 2017.

WANG, F. et al. Embedding traceability in large language model code generation: Towards trustworthy AI-augmented software engineering. In: **Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering Companion**. Trondheim: ACM, 2025. p. 1760–1763. DOI: <https://doi.org/10.1145/3696630.3730569>.

Apêndices

APÊNDICE A – Narrativas de Caso de Uso

A.1 UC01 — Registrar-se no Sistema

Ator(es): Novo usuário.

Referência: Não há.

Pré-condição: O usuário não possui sessão autenticada.

Fluxo Principal

[EV] O usuário acessa a tela de *login* e clica em *Registre-se*.

[RS] O sistema exibe a tela de cadastro.

[EV] O usuário preenche os campos **Nome**, **Sobrenome**, **E-mail**, **Senha** e **Confirmar Senha**. [FE01] [FE02] [FE03]

[EV] O usuário lê os termos de participação e marca o aceite. [FE04]

[EV] O usuário clica em **Cadastrar**. [FE05]

[RS] O sistema registra a conta, exibe a mensagem [MSG01] e redireciona o usuário para a tela de *login*.

Fluxo Alternativo

FA01 — *Feedback* de senha durante o preenchimento:

Durante o preenchimento da senha, o sistema valida os critérios mínimos em tempo real.

[RS] O sistema informa ao usuário se a senha atende aos critérios de segurança exigidos.

Fluxo de Exceção

FE01 — E-mail já cadastrado:

No passo 5 do Fluxo Principal, o *backend* rejeita o cadastro por duplicidade.

[RS] O sistema exibe a mensagem [MSG02].

FE02 — Senhas não coincidem:

No passo 3 do Fluxo Principal, os campos **Senha** e **Confirmar Senha** divergem.

[RS] O sistema exibe a mensagem [MSG03].

FE03 — Senha fraca:

No passo 3 do Fluxo Principal, a senha não atende aos critérios mínimos.

[RS] O sistema informa a regra não atendida.

FE04 — Termos não aceitos:

No passo 4 do Fluxo Principal, o usuário não marca o aceite.

[RS] O botão de cadastro permanece desabilitado e o envio não é realizado.

FE05 — Erro no cadastro:

No passo 5 do Fluxo Principal, ocorre falha na operação de registro.

[RS] O sistema exibe a mensagem retornada pela API.

Pós-condição: Uma nova conta foi criada e o usuário pode autenticar-se no sistema.

Mensagens

MSG01: Registrado com sucesso.

MSG02: E-mail já cadastrado.

MSG03: As senhas não coincidem.

A.2 UC02 — Realizar *Login* com E-mail e Senha

Ator(es): Representante e/ou Administrador de projetos.

Referência: BR03.

Pré-condição: O usuário possui cadastro ativo e está na tela de *login*.

Fluxo Principal

[EV] O usuário preenche o campo **E-mail**.

[EV] O usuário preenche o campo **Senha**.

[EV] Opcionalmente, o usuário marca **Manter sessão**.

[EV] O usuário clica em **Entrar**. [FE01]

[RS] O sistema autentica o usuário, armazena *access token* e *refresh token* de acordo com a opção de persistência de sessão e identifica seus papéis.

[RS] Se o usuário estiver com primeiro acesso pendente, o sistema redireciona para **Configurações > Trocar senha**. Caso contrário, redireciona para a página inicial.

Fluxo Alternativo

FA01 — Manter sessão ativa:

No passo 3 do Fluxo Principal, o usuário marca **Manter sessão**.

[RS] O sistema persiste os *tokens* em armazenamento durável e tenta renovar a sessão automaticamente enquanto o *refresh token* for válido.

Fluxo de Exceção

FE01 — Credenciais inválidas:

No passo 4 do Fluxo Principal, o *backend* rejeita a autenticação.

[RS] O sistema exibe a mensagem [MSG01].

Pós-condição: Uma sessão autenticada foi iniciada e o contexto do usuário foi carregado.

Mensagens

MSG01: E-mail ou senha incorretos.

A.3 UC03 — Realizar *Login* com Google

Ator(es): Representante e/ou Administrador de projetos.

Referência: Não há.

Pré-condição: O usuário está sem sessão autenticada e possui conta Google elegível para autenticação.

Fluxo Principal

[EV] O usuário acessa a tela de *login* ou cadastro.

[EV] O usuário clica no botão do Google.

[RS] O sistema inicializa o provedor Google Identity Services e apresenta o fluxo de autenticação do Google.

[EV] O usuário seleciona sua conta Google e concede a autenticação.

[RS] O sistema envia o *token* de identidade ao *backend*, autentica o usuário e armazena os *tokens* da sessão.

[RS] Se houver primeiro acesso pendente, o sistema redireciona para **Configurações > Trocar senha**; caso contrário, redireciona para a página inicial.

Fluxo Alternativo

FA01 — Primeiro acesso via Google:

Após o passo 5 do Fluxo Principal, o sistema identifica que a conta ainda precisa concluir aceite de termos.

[RS] O sistema redireciona o usuário para o fluxo de primeiro acesso em **Configurações**.

Fluxo de Exceção

FE01 — Falha na autenticação federada:

No passo 5 do Fluxo Principal, o *backend* ou o provedor externo rejeita o *token* recebido.

[RS] O sistema exibe a mensagem de erro retornada pela API.

Pós-condição: Uma sessão autenticada por provedor Google foi iniciada.

Mensagens: Não há mensagens fixas além das notificações de erro retornadas pela API.

A.4 UC04 — Recuperar Conta e Redefinir Senha

Ator(es): Usuário sem sessão autenticada.

Referência: Não há.

Pré-condição: O usuário está na tela de *login* e não consegue acessar a conta.

Fluxo Principal

[EV] O usuário clica em **Esqueci minha senha**.

[RS] O sistema exibe a tela de recuperação de conta.

[EV] O usuário informa o e-mail e solicita o envio do código. [FE01]

[RS] O sistema envia um código de verificação e redireciona para a tela de confirmação do código.

[EV] O usuário informa o código recebido. [FE02]

[RS] O sistema valida o código e redireciona para a tela de redefinição de senha.

[EV] O usuário informa a nova senha e sua confirmação. [FE03] [FE04]

[EV] O usuário confirma a redefinição.

[RS] O sistema redefine a senha, exibe a mensagem [MSG01] e redireciona para o *login*.

Fluxo Alternativo

FA01 — Colagem do código completo:

No passo 5 do Fluxo Principal, o usuário cola o código completo de verificação.

[RS] O sistema distribui automaticamente os dígitos nos campos do formulário.

Fluxo de Exceção

FE01 — E-mail inválido ou indisponível:

No passo 3 do Fluxo Principal, o endereço informado não é aceito pelo *backend*.

[RS] O sistema exibe a mensagem correspondente.

FE02 — Código inválido ou expirado:

No passo 5 do Fluxo Principal, o código informado não é validado.

[RS] O sistema exibe a mensagem [MSG02].

FE03 — Senhas não coincidem:

No passo 7 do Fluxo Principal, os campos de senha divergem.

[RS] O sistema exibe a mensagem [MSG03].

FE04 — Senha fora dos critérios de segurança:

No passo 7 do Fluxo Principal, a nova senha é considerada fraca.

[RS] O sistema exibe a regra de validação não atendida.

Pós-condição: A senha do usuário foi redefinida e a conta pode ser acessada normalmente.

Mensagens

MSG01: Senha redefinida com sucesso.

MSG02: Código inválido. Reenvie o código ou informe outro e-mail.

MSG03: As senhas não coincidem.

A.5 UC05 — Alterar Senha e Concluir Primeiro Acesso

Ator(es): Representante e/ou Administrador de projetos autenticado.

Referência: Não há.

Pré-condição: O usuário está autenticado e acessa **Configurações > Trocar senha**.

Fluxo Principal

[EV] O usuário acessa **Configurações**.

[EV] O usuário clica em **Trocar senha**.

[RS] O sistema exibe o formulário com **Senha atual**, **Nova senha** e **Confirmar nova senha**.

[RS] Se o usuário estiver em primeiro acesso, o sistema também exibe *modal* explicativo e o aceite dos termos. [FA01]

[EV] O usuário preenche os campos necessários. [FE01] [FE02] [FE03]

[EV] O usuário confirma a operação.

[RS] O sistema valida a senha atual, redefine a senha com indicador de primeiro acesso e, quando aplicável, marca o aceite de termos.

[RS] O sistema renova o *token* da sessão, exibe a mensagem [MSG01] e redireciona o usuário para **Configurações**.

Fluxo Alternativo

FA01 — Primeiro acesso de usuário autenticado via Google:

No passo 4 do Fluxo Principal, o sistema identifica que o usuário é do provedor Google.

[RS] O sistema exibe apenas o formulário de aceite de termos, sem solicitar senha atual ou nova senha.

[EV] O usuário aceita os termos e confirma.

[RS] O sistema conclui o primeiro acesso, renova a sessão, exibe a mensagem [MSG02] e redireciona para a página inicial.

Fluxo de Exceção

FE01 — Senha atual incorreta:

No passo 7 do Fluxo Principal, a reautenticação falha.

[RS] O sistema exibe a mensagem [MSG03].

FE02 — Nova senha fraca:

No passo 5 do Fluxo Principal, a senha não atende aos critérios mínimos.

[RS] O sistema exibe a regra de validação não atendida.

FE03 — Confirmação divergente:

No passo 5 do Fluxo Principal, **Nova senha** e **Confirmar nova senha** divergem.

[RS] O sistema exibe a mensagem de validação correspondente.

Pós-condição: A senha foi atualizada ou o primeiro acesso foi concluído com aceite de termos.

Mensagens

MSG01: Senha atualizada com sucesso.

MSG02: Termos aceitos com sucesso. Bem-vindo ao sistema.

MSG03: Senha atual não confere.

A.6 UC06 — Listar e Filtrar Projetos

Ator(es): Representante e/ou Administrador de projetos.

Referência: Não há.

Pré-condição: O usuário está autenticado e possui acesso ao menu **Projetos**.

Fluxo Principal

[EV] O usuário acessa **Projetos** pelo menu principal.

[RS] O sistema carrega a listagem paginada de projetos.

[EV] O usuário pode filtrar por **Nome**, **Código**, **Tipo** e ***Status***.

[EV] O usuário aplica os filtros desejados.

[RS] O sistema atualiza a listagem conforme os critérios informados.

[EV] O usuário navega entre as páginas de resultado.

[EV] O usuário seleciona um projeto da lista para visualizar seus detalhes.

Fluxo Alternativo

FA01 — Criar novo projeto a partir da listagem:

Na tela de listagem, um administrador clica em **Criar projeto**.

[RS] O sistema redireciona para o fluxo de criação de projeto.

Fluxo de Exceção

FE01 — Falha ao carregar projetos:

No passo 2 do Fluxo Principal, ocorre erro de comunicação com a API.

[RS] O sistema exibe mensagem de falha de carregamento.

Pós-condição: O usuário visualizou a lista de projetos e pode iniciar navegação para um item específico.

Mensagens: Não há mensagens fixas além das mensagens de erro de carregamento.

A.7 UC07 — Criar Projeto

Ator(es): Administrador de projetos.

Referência: BR02.

Pré-condição: O usuário está autenticado com perfil administrativo.

Fluxo Principal

[EV] O usuário acessa **Projetos** e clica em **Criar projeto**.

[RS] O sistema exibe o seletor de tipo de projeto com as opções **Cascata** e **Iterativo**.

[EV] O usuário escolhe o tipo do projeto. [FA01] [FA02]

[RS] O sistema exibe o formulário correspondente com as seções **Projeto**, **Etapas**, **Representantes** e **Questionários**.

[EV] O usuário preenche os dados básicos do projeto, inclusive *template* quando desejar reutilizar uma estrutura existente. [FA03] [FE01]

[EV] O usuário configura etapas, representantes e questionários. [FE02]

[EV] O usuário clica em **Salvar projeto**.

[RS] O sistema exibe uma confirmação informando que criará o projeto e também um *template* derivado dele.

[EV] O usuário confirma a criação.

[RS] O sistema cria o projeto, gera automaticamente um *template* a partir da configuração informada, exibe a mensagem [MSG01] e redireciona para a listagem de projetos.

Fluxo Alternativo

FA01 — Projeto Cascata:

No passo 3 do Fluxo Principal, o usuário seleciona **Cascata**.

[RS] O sistema exige nome, data de início e prazo limite, e organiza o projeto em etapas com sequência e duração em dias úteis.

FA02 — Projeto Iterativo:

No passo 3 do Fluxo Principal, o usuário seleciona **Iterativo**.

[RS] O sistema exige nome, data de início e duração das iterações, calcula o número de iterações e utiliza questionários associados às iterações.

FA03 — Salvar como rascunho:

Antes do passo 7 do Fluxo Principal, o usuário clica em **Salvar como Rascunho**.

[RS] O sistema salva o projeto com *status* **Rascunho**, exibe a mensagem [MSG02] e permite continuidade futura da configuração.

Fluxo de Exceção

FE01 — Dados básicos insuficientes:

No passo 5 do Fluxo Principal, faltam campos obrigatórios ou há conflito de datas.

[RS] O sistema informa os problemas encontrados e impede a criação.

FE02 — Estrutura inválida do projeto:

No passo 6 do Fluxo Principal, não há etapas, representantes ou perguntas suficientes, ou há inconsistências entre papéis, etapas e questionários.

[RS] O sistema exibe a relação de pendências e impede a criação.

Pós-condição: Um novo projeto foi criado ou salvo como rascunho.

Mensagens

MSG01: Projeto criado e *template* gerado com sucesso.

MSG02: Rascunho salvo com sucesso.

A.8 UC08 — Editar Projeto

Ator(es): Administrador de projetos.

Referência: BR02.

Pré-condição: O usuário está autenticado com perfil administrativo e o projeto existe.

Fluxo Principal

[EV] O usuário acessa os detalhes do projeto e clica em **Editar projeto**.

[RS] Se o projeto estiver em **Rascunho**, o sistema reabre o fluxo de criação com os dados preenchidos; se estiver **Aberto**, o sistema abre a tela dedicada de edição.

[EV] O usuário altera dados do projeto, etapas, representantes e questionários.

[EV] O usuário solicita o salvamento.

[RS] O sistema valida as alterações, consolida o impacto da operação e persiste as mudanças.

[RS] O sistema exibe a mensagem [MSG01] e redireciona para a tela de detalhes do projeto.

Fluxo Alternativo

FA01 — Projeto em rascunho:

No passo 2 do Fluxo Principal, o projeto está em **Rascunho**.

[RS] O sistema reaproveita o mesmo fluxo de configuração utilizado na criação.

Fluxo de Exceção

FE01 — Campos obrigatórios ou estrutura inválida:

No passo 5 do Fluxo Principal, a edição viola regras de consistência.

[RS] O sistema informa as pendências e não conclui o salvamento.

FE02 — Erro ao persistir alterações:

No passo 5 do Fluxo Principal, ocorre erro de *backend*.

[RS] O sistema exibe a mensagem retornada pela API.

Pós-condição: As alterações válidas do projeto foram salvas.

Mensagens

MSG01: Projeto atualizado com sucesso.

A.9 UC09 — Publicar Projeto

Ator(es): Administrador de projetos.

Referência: Não há.

Pré-condição: O projeto está com *status* **Rascunho** e o usuário está na tela de detalhes do projeto.

Fluxo Principal

[EV] O usuário clica em **Publicar projeto**.

[RS] O sistema exibe uma confirmação de publicação.

[EV] O usuário confirma a operação.

[RS] O sistema altera o *status* do projeto para **Aberto**, recarrega os dados e exibe a mensagem [MSG01].

Fluxo Alternativo: Não há.

Fluxo de Exceção

FE01 — Erro na publicação:

No passo 4 do Fluxo Principal, a API rejeita a operação.

[RS] O sistema exibe a mensagem de erro retornada.

Pós-condição: O projeto foi publicado e passa a aceitar o fluxo operacional normal.

Mensagens

MSG01: Projeto publicado com sucesso.

A.10 UC10 — Excluir Projeto

Ator(es): Administrador de projetos.

Referência: Não há.

Pré-condição: O usuário está na tela de detalhes do projeto e o projeto não está concluído.

Fluxo Principal

[EV] O usuário clica em **Excluir projeto**.

[RS] O sistema exibe mensagem de confirmação com o nome do projeto.

[EV] O usuário confirma a exclusão.

[RS] O sistema remove o projeto, exibe a mensagem [MSG01] e redireciona para a listagem de projetos.

Fluxo Alternativo: Não há.

Fluxo de Exceção

FE01 — Erro na exclusão:

No passo 4 do Fluxo Principal, ocorre erro no *backend*.

[RS] O sistema exibe a mensagem retornada pela API.

Pós-condição: O projeto foi removido do sistema.

Mensagens

MSG01: Projeto excluído com sucesso.

A.11 UC11 — Visualizar Detalhes do Projeto

Ator(es): Representante e/ou Administrador de projetos.

Referência: Não há.

Pré-condição: O usuário está autenticado e o projeto existe.

Fluxo Principal

[EV] O usuário seleciona um projeto na listagem.

[RS] O sistema exibe os dados gerais do projeto, seu *status*, situação atual e a lista paginada de questionários.

[EV] O usuário pode filtrar os questionários por **Nome**, **Etapa** ou **Iteração**, conforme o tipo do projeto.

[RS] O sistema atualiza a lista conforme os filtros aplicados.

[EV] O usuário escolhe uma ação disponível para um questionário ou para o projeto.

Fluxo Alternativo

FA01 — Ações disponíveis para representante:

O usuário pode responder o questionário quando estiver associado e houver pendência.

O usuário pode visualizar respostas já enviadas.

O usuário pode acessar seu painel individual quando houver *dashboard* calculado.

FA02 — Ações disponíveis para administrador:

O usuário pode editar, publicar ou excluir o projeto.

O usuário pode copiar o *link* de um questionário.

O usuário pode entrar em modo administrativo de visualização do questionário.

O usuário pode enviar lembretes, reagendar questionários, forçar encerramento e abrir *dashboards*.

Fluxo de Exceção

FE01 — Falha ao carregar projeto ou questionários:

Nos passos 2 ou 4 do Fluxo Principal, ocorre erro de leitura.

[RS] O sistema exibe mensagem de falha e disponibiliza ação de nova tentativa.

Pós-condição: O usuário visualizou o estado operacional do projeto e pode disparar fluxos complementares.

Mensagens: Não há mensagens fixas além das mensagens de erro e confirmação da própria tela.

A.12 UC12 — Manter Etapas

Ator(es): Administrador de projetos.

Referência: BR02.

Pré-condição: O usuário está no formulário de criação ou edição de projeto.

Fluxo Principal

[RS] O sistema exibe a lista de etapas já configuradas.

[EV] O usuário escolhe adicionar, editar ou excluir uma etapa. [FA01] [FA02] [FA03]

[RS] O sistema valida os dados da etapa e atualiza a lista.

Fluxo Alternativo

FA01 — Adicionar etapa:

O usuário clica em **Adicionar etapa**.

[RS] O sistema abre o *modal* correspondente ao tipo do projeto.

[EV] O usuário informa os campos obrigatórios.

[RS] O sistema adiciona a etapa à lista.

FA02 — Editar etapa:

O usuário clica em **Editar** na etapa desejada.

[RS] O sistema abre o *modal* com os dados preenchidos.

[EV] O usuário altera os dados e confirma.

[RS] O sistema atualiza a etapa na lista.

FA03 — Excluir etapa:

O usuário clica em **Excluir** na etapa desejada.

[RS] O sistema remove a etapa da configuração atual.

Fluxo de Exceção

FE01 — Campos inválidos:

No passo 3 do Fluxo Principal, a etapa possui nome inválido, peso inadequado, sequência duplicada ou dados insuficientes.

[RS] O sistema exibe as mensagens de validação e não conclui a operação.

Pós-condição: A estrutura de etapas do projeto foi atualizada.

Mensagens: Não há mensagens globais fixas além das validações do *modal*.

A.13 UC13 — Manter Representantes

Ator(es): Administrador de projetos.

Referência: BR02.

Pré-condição: O usuário está no formulário de criação ou edição de projeto.

Fluxo Principal

[RS] O sistema exibe a lista de representantes configurados.

[EV] O usuário escolhe adicionar, editar ou excluir um representante. [FA01] [FA02] [FA03]

[RS] O sistema atualiza a lista com nome, sobrenome, e-mail, peso, papéis e data de inclusão.

Fluxo Alternativo

FA01 — Adicionar representante:

O usuário clica em **Adicionar representante**.

[RS] O sistema abre um *modal* com os campos obrigatórios.

[EV] O usuário informa **nome**, **sobrenome**, **e-mail**, **peso** e ao menos um encargo.

[RS] O sistema adiciona o representante à lista.

FA02 — Editar representante:

O usuário clica em **Editar** no representante desejado.

[RS] O sistema abre o *modal* com os dados atuais.

[EV] O usuário altera os dados e confirma.

[RS] O sistema atualiza a linha correspondente.

FA03 — Excluir representante:

O usuário remove o representante da configuração atual.

[RS] O sistema atualiza a lista.

Fluxo de Exceção

FE01 — Dados obrigatórios inválidos:

No passo 3 do Fluxo Principal, há campos ausentes ou fora do intervalo aceito.

[RS] O sistema mantém o *modal* aberto e exibe as mensagens de validação.

Pós-condição: A composição de representantes do projeto foi atualizada.

Mensagens: Não há mensagens globais fixas além das validações do *modal*.

A.14 UC14 — Manter Questionários

Ator(es): Administrador de projetos.

Referência: BR02.

Pré-condição: O usuário está no formulário de criação ou edição de projeto.

Fluxo Principal

[RS] O sistema exibe a lista de questionários do projeto.

[EV] O usuário escolhe adicionar, editar ou excluir um questionário. [FA01] [FA02] [FA03]

[RS] O sistema atualiza a configuração do questionário com seu nome, período e perguntas.

Fluxo Alternativo

FA01 — Adicionar questionário:

O usuário clica em **Adicionar questionário**.

[RS] O sistema abre o formulário específico do tipo de projeto.

[EV] O usuário informa os dados do questionário e gerencia suas perguntas.

[RS] O sistema grava o questionário na configuração atual do projeto.

FA02 — Editar questionário:

O usuário clica em **Editar** no questionário desejado.

[RS] O sistema reabre o formulário com os dados atuais.

[EV] O usuário atualiza o questionário e salva.

[RS] O sistema atualiza a configuração do projeto.

FA03 — Gerenciar perguntas do questionário:

No formulário do questionário, o usuário pode **criar pergunta nova**, **reutilizar perguntas existentes**, **editar perguntas** e **remover perguntas** selecionadas.

[RS] O sistema atualiza a lista de perguntas associadas ao questionário.

Fluxo de Exceção

FE01 — Questionário sem perguntas ou dados inválidos:

No passo 3 do Fluxo Principal, o questionário não possui perguntas válidas ou tem campos obrigatórios pendentes.

[RS] O sistema impede o salvamento e informa as inconsistências.

Pós-condição: Os questionários do projeto foram configurados com sua estrutura de aplicação e perguntas.

Mensagens: Não há mensagens globais fixas além das validações e avisos do formulário.

A.15 UC15 — Responder Questionário

Ator(es): Representante.

Referência: Não há.

Pré-condição: O usuário está autenticado, associado ao questionário e possui permissão de resposta.

Fluxo Principal

[EV] Na tela de detalhes do projeto, o usuário clica em **Responder agora**.

[RS] O sistema carrega o questionário, as perguntas, o *status* do participante e o progresso de respostas.

[EV] Para cada pergunta, o usuário informa **Sim** ou **Não**. [FA01] [FA02]

[EV] O usuário pode salvar rascunho a qualquer momento. [FA03]

[EV] Ao concluir, o usuário clica em **Enviar respostas**. [FE01]

[RS] O sistema valida o preenchimento, persiste as respostas, exibe a mensagem [MSG01] e retorna à tela de detalhes do projeto.

Fluxo Alternativo

FA01 — Adicionar evidência ou justificativa:

No passo 3 do Fluxo Principal, o usuário abre o *modal* de anotações da resposta.

[EV] O usuário informa observações e, opcionalmente, *links* de evidência.

[RS] O sistema associa a anotação à resposta.

FA02 — Revisar respostas já preenchidas:

Durante o fluxo, o usuário pode voltar às perguntas anteriores.

[RS] O sistema mantém em memória local e em *cache* de rascunho as respostas em andamento.

FA03 — Salvar rascunho:

O usuário clica em **Salvar rascunho**.

[RS] O sistema salva o andamento localmente e no servidor e exibe a mensagem [MSG02].

Fluxo de Exceção

FE01 — Perguntas não respondidas:

No passo 5 do Fluxo Principal, ainda existem perguntas sem resposta.

[RS] O sistema exibe a mensagem [MSG03] e destaca as pendências.

Pós-condição: As respostas foram submetidas ou permaneceram registradas como rascunho.

Mensagens

MSG01: Respostas salvas com sucesso.

MSG02: Rascunho salvo com sucesso.

MSG03: Existem perguntas não respondidas.

A.16 UC16 — Visualizar Questionário em Modo Administra

Ator(es): Administrador de projetos.

Referência: Não há.

Pré-condição: O usuário está autenticado como administrador e acessa um questionário a partir da tela de detalhes do projeto.

Fluxo Principal

[EV] O administrador clica em **Visualizar questionário**.

[RS] O sistema identifica automaticamente o tipo do projeto e redireciona para o formulário correto do questionário.

[RS] O sistema carrega o questionário em **modo somente leitura**, indicando ao usuário que se trata de visualização administrativa.

[EV] O administrador revisa respostas, observações e anexos existentes.

Fluxo Alternativo: Não há.

Fluxo de Exceção

FE01 — Identificação inválida do projeto ou questionário:

No passo 2 do Fluxo Principal, os parâmetros são inválidos ou o projeto não pode ser carregado.

[RS] O sistema redireciona para a listagem de projetos.

Pós-condição: O questionário foi aberto para consulta sem permitir edição administrativa direta das respostas.

Mensagens: Não há mensagens fixas além dos avisos de contexto da tela.

A.17 UC17 — Reagendar Questionário

Ator(es): Administrador de projetos.

Referência: Não há.

Pré-condição: O usuário está na tela de detalhes do projeto; o questionário está **Pendente**; já existe ao menos um questionário concluído no projeto; o item selecionado é o primeiro questionário pendente da ordem cronológica.

Fluxo Principal

[EV] O usuário clica em **Reagendar** no questionário desejado.

[RS] O sistema abre o *modal* de reagendamento com as datas atuais.

[EV] O usuário informa a nova **data de início** e a nova **data de término**.

[EV] O usuário confirma a operação. [FE01] [FE02]

[RS] O sistema atualiza o período do questionário, recarrega a lista e exibe a mensagem [MSG01].

Fluxo Alternativo

FA01 — Início imediato do questionário:

No passo 3 do Fluxo Principal, a nova data de início é hoje ou data passada.

[RS] O sistema informa que o questionário passará imediatamente a **Em andamento**.

Fluxo de Exceção

FE01 — Ordem de datas inválida:

No passo 4 do Fluxo Principal, a data final é anterior à data inicial.

[RS] O sistema exibe validação no *modal* e impede a confirmação.

FE02 — Erro ao reagendar:

No passo 5 do Fluxo Principal, a API rejeita a operação.

[RS] O sistema exibe a mensagem [MSG02].

Pós-condição: O questionário passou a ter novo período de aplicação.

Mensagens

MSG01: Questionário reagendado com sucesso.

MSG02: Erro ao reagendar o questionário.

A.18 UC18 — Notificar Representantes Pendentes

Ator(es): Administrador de projetos.

Referência: Não há.

Pré-condição: O usuário está na tela de detalhes do projeto e há questionário em andamento com representantes pendentes que possuam e-mail.

Fluxo Principal

[EV] O usuário clica em **Enviar lembrete** para um questionário em andamento.

[RS] O sistema identifica os representantes pendentes aptos a receber notificação.

[EV] O usuário confirma o envio do lembrete.

[RS] O sistema envia as notificações e exibe a mensagem [MSG01].

Fluxo Alternativo: Não há.

Fluxo de Exceção

FE01 — Não há pendentes com e-mail:

No passo 2 do Fluxo Principal, nenhum destinatário elegível é encontrado.

[RS] O sistema exibe a mensagem [MSG02].

FE02 — Erro no envio:

No passo 4 do Fluxo Principal, ocorre falha ao acionar o envio.

[RS] O sistema exibe a mensagem [MSG03].

Pós-condição: Os representantes pendentes elegíveis foram notificados ou o usuário foi informado da impossibilidade de envio.

Mensagens

MSG01: Lembrete enviado com sucesso.

MSG02: Não há representantes pendentes com endereço de e-mail disponível para este questionário.

MSG03: Falha ao enviar o lembrete.

A.19 UC19 — Encerrar Questionário

Ator(es): Administrador de projetos.

Referência: Não há.

Pré-condição: O usuário está na tela de detalhes do projeto ou no *dashboard* do questionário e possui permissão administrativa.

Fluxo Principal

[EV] O usuário clica em **Encerrar questionário**.

[RS] O sistema exibe mensagem de confirmação.

[EV] O usuário confirma.

[RS] O sistema encerra o questionário, calcula o **ISEQ** com as respostas existentes e atualiza o *status* para **Concluído**.

[RS] O sistema exibe a mensagem [MSG01].

Fluxo Alternativo: Não há.

Fluxo de Exceção

FE01 — Erro no encerramento:

No passo 4 do Fluxo Principal, a operação falha.

[RS] O sistema exibe a mensagem [MSG02].

Pós-condição: O questionário foi encerrado e disponibilizou seus *dashboards* e exportações.

Mensagens

MSG01: Questionário encerrado. O ISEQ foi calculado.

MSG02: Erro ao encerrar o questionário. Verifique se ele possui respostas registradas.

A.20 UC20 — Visualizar *Dashboard* Consolidado do Projeto

Ator(es): Administrador de projetos, Analista de qualidade e Gerente de projeto.

Referência: Não há.

Pré-condição: O usuário está autenticado e o projeto possui dados suficientes para o *dashboard* consolidado.

Fluxo Principal

[EV] Na tela de detalhes do projeto, o usuário clica em ***Dashboard* do projeto**.

[RS] O sistema carrega o painel consolidado com **KPIs de ISEP**, progresso de questionários, média da equipe, desvio padrão, dimensões de governança, dívida ética e dívida técnica.

[EV] O usuário pode navegar para respostas consolidadas, exportar CSV ou abrir o *dashboard* de um questionário específico.

[RS] O sistema apresenta também os registros de emissão de certificado já existentes para o projeto.

Fluxo Alternativo

FA01 — Encerrar projeto a partir do *dashboard*:

Se o usuário for administrador e o projeto ainda não estiver encerrado, ele pode clicar em **Encerrar projeto**.

[RS] O sistema segue o fluxo de encerramento do projeto.

Fluxo de Exceção

FE01 — *Dashboard* indisponível:

No passo 2 do Fluxo Principal, não há dados suficientes ou ocorre erro de carregamento.

[RS] O sistema exibe mensagem de indisponibilidade do *dashboard*.

Pós-condição: O usuário visualizou o desempenho consolidado do projeto.

Mensagens: Não há mensagens fixas além dos erros de carregamento.

A.21 UC21 — Encerrar Projeto

Ator(es): Administrador de projetos.

Referência: Não há.

Pré-condição: O usuário está no *dashboard* do projeto e o projeto ainda não possui ISEP consolidado calculado.

Fluxo Principal

[EV] O usuário clica em **Encerrar projeto**.

[RS] O sistema exibe mensagem de confirmação do encerramento consolidado.

[EV] O usuário confirma.

[RS] O sistema calcula o **ISEP consolidado** do projeto, registra data e usuário do encerramento e atualiza o estado do painel.

[RS] O sistema exibe uma confirmação com **ISEP**, **faixa** e **quantidade de questionários** considerados.

Fluxo Alternativo: Não há.

Fluxo de Exceção

FE01 — Erro ao encerrar o projeto:

No passo 4 do Fluxo Principal, a operação falha.

[RS] O sistema exibe a mensagem de erro correspondente.

Pós-condição: O projeto fica encerrado e apto à emissão ou *download* do certificado, quando aplicável.

Mensagens: Não há mensagem fixa além do resumo de sucesso e dos erros de *backend*.

A.22 UC22 — Visualizar *Dashboard* do Questionário

Ator(es): Administrador de projetos, Analista de qualidade e Gerente de projeto.

Referência: Não há.

Pré-condição: O questionário está encerrado e possui ISEQ calculado.

Fluxo Principal

[EV] O usuário abre o *dashboard* do questionário a partir do projeto ou de outro *dashboard*.

[RS] O sistema exibe **KPIs do questionário**, distribuição por faixa, dimensões de governança, *heatmap* por papel e etapa, barra por papel e etapa, nuvem de palavras, indicadores de dívida e *widgets* de IA.

[EV] O usuário pode visualizar respostas consolidadas, exportar CSV, exportar JSON e abrir o detalhamento de respostas por membro.

[RS] O sistema exibe também a área de emissões registradas do boletim de não conformidade.

Fluxo Alternativo

FA01 — Forçar encerramento a partir do *dashboard*:

Se o usuário for administrador e o questionário ainda estiver aberto, ele pode usar a ação de encerramento.

[RS] O sistema segue o fluxo de encerramento do questionário.

Fluxo de Exceção

FE01 — ISEQ indisponível:

No passo 2 do Fluxo Principal, o questionário ainda não possui cálculo disponível ou há erro de consulta.

[RS] O sistema informa que o *dashboard* ainda não pode ser carregado.

Pós-condição: O usuário visualizou os indicadores analíticos do questionário.

Mensagens: Não há mensagens fixas além dos erros de carregamento.

A.23 UC23 — Visualizar *Dashboard* Individual

Ator(es): Representante.

Referência: Não há.

Pré-condição: O usuário está autenticado, é participante do questionário e o *dashboard* individual está disponível.

Fluxo Principal

[EV] Na tela de detalhes do projeto, o usuário clica em **Meu painel**.

[RS] O sistema identifica o representante autenticado no contexto do questionário.

[RS] O sistema exibe indicadores individuais, média da equipe, média histórica e gráfico *radar* comparativo.

Fluxo Alternativo: Não há.

Fluxo de Exceção

FE01 — Representante não encontrado:

No passo 2 do Fluxo Principal, o usuário autenticado não é localizado entre os participantes do questionário.

[RS] O sistema exibe mensagem de erro apropriada.

FE02 — Erro ao carregar o painel:

No passo 3 do Fluxo Principal, ocorre erro de consulta.

[RS] O sistema informa falha no carregamento do painel individual.

Pós-condição: O representante visualizou sua própria posição no contexto do questionário.

Mensagens: Não há mensagens fixas além das mensagens de erro.

A.24 UC24 — Visualizar Respostas Consolidadas

Ator(es): Administrador de projetos, Analista de qualidade e Gerente de projeto.

Referência: Não há.

Pré-condição: Existem respostas registradas no projeto ou no questionário selecionado.

Fluxo Principal

[EV] O usuário clica em **Ver Todas as Respostas** no *dashboard* do projeto ou do questionário.

[RS] O sistema exibe a grade paginada de respostas consolidadas.

[EV] O usuário pode filtrar por **texto da pergunta**, **representante**, **papel**, **resposta** e **identificador do questionário**.

[EV] O usuário aplica os filtros e navega pelas páginas.

[EV] O usuário abre uma resposta específica para ver detalhes completos.

[RS] O sistema exibe o conteúdo da pergunta, a resposta, a justificativa e os *links* anexados.

Fluxo Alternativo: Não há.

Fluxo de Exceção

FE01 — Erro ao carregar as respostas:

Nos passos 2 ou 6 do Fluxo Principal, ocorre erro de consulta.

[RS] O sistema informa a falha de carregamento.

Pós-condição: O usuário visualizou as respostas individuais de forma consolidada.

Mensagens: Não há mensagens fixas além das mensagens de erro.

A.25 UC25 — Exportar Dados Analíticos

Ator(es): Administrador de projetos, Analista de qualidade e Gerente de projeto.

Referência: Não há.

Pré-condição: O usuário está em um *dashboard* com dados carregados.

Fluxo Principal

[EV] O usuário clica em **Relatório do Resultado (CSV)** no *dashboard* do projeto ou do questionário.

[RS] O sistema gera o arquivo CSV e inicia o *download*.

Fluxo Alternativo

FA01 — Exportar JSON do questionário:

No *dashboard* do questionário, o usuário clica em **Relatório do Resultado (JSON)**.

[RS] O sistema gera o arquivo JSON e inicia o *download*.

Fluxo de Exceção

FE01 — Erro ao exportar:

Durante a geração do arquivo, ocorre erro de *backend* ou processamento.

[RS] O sistema exibe mensagem de falha na exportação.

Pós-condição: O arquivo de exportação foi baixado ou o usuário foi informado do erro.

Mensagens: Não há mensagens fixas além das mensagens de erro de exportação.

A.26 UC26 — Utilizar e Gerar *Templates* de Projeto

Ator(es): Administrador de projetos.

Referência: Não há.

Pré-condição: O usuário está criando um projeto ou configurando partes reutilizáveis da sua estrutura.

Fluxo Principal

[EV] Durante a criação de um projeto, o usuário seleciona um *template* disponível para reaproveitar uma configuração anterior.

[RS] O sistema carrega o *template* escolhido e preenche automaticamente os dados compatíveis do formulário.

[EV] O usuário ajusta as informações conforme necessário.

[EV] Ao concluir a criação do projeto, o usuário confirma a operação.

[RS] O sistema cria o projeto e gera automaticamente um novo *template* derivado da configuração criada.

Fluxo Alternativo

FA01 — Reutilizar partes de *template*:

Durante a configuração de etapas, representantes ou perguntas, o usuário escolhe **Selecionar já existente**.

[RS] O sistema exibe um *modal* de seleção de *templates* compatíveis com o tipo de projeto.

[EV] O usuário escolhe o item desejado.

[RS] O sistema importa a parte selecionada para a configuração atual.

Fluxo de Exceção

FE01 — Falha ao carregar *templates*:

No passo 2 do Fluxo Principal ou no passo 2 do Fluxo Alternativo, o catálogo não pode ser carregado.

[RS] O sistema informa o erro e mantém o fluxo sem o reaproveitamento.

FE02 — Falha ao gerar o *template* derivado:

No passo 5 do Fluxo Principal, o projeto é criado, mas a clonagem para *template* falha.

[RS] O sistema exibe a mensagem de erro retornada pela API.

Pós-condição: O usuário reaproveitou *templates* existentes e, ao criar um projeto, gerou automaticamente um novo *template* derivado.

Mensagens: Não há mensagens fixas além das notificações de sucesso e erro do fluxo de criação.

A.27 UC27 — Baixar e Registrar Emissão de Certificado

Ator(es): Administrador de projetos, Analista de qualidade e Gerente de projeto.

Referência: Não há.

Pré-condição: O usuário está no *dashboard* do projeto e o projeto possui **ISEP consolidado** calculado.

Fluxo Principal

[EV] O usuário clica em **Baixar Certificado (PDF)**.

[RS] O sistema gera o PDF do certificado e inicia o *download*. [FE01]

Fluxo Alternativo

FA01 — Registrar emissão do certificado:

No *dashboard* do projeto, o usuário clica em **Registrar Emissão do Certificado**.

[RS] O sistema exibe uma confirmação informando que a ação criará um **registro histórico auditável**.

[EV] O usuário confirma.

[RS] O sistema registra a emissão, grava **código de autenticidade**, atualiza a lista de emissões e exibe a mensagem [MSG01].

Fluxo de Exceção

FE01 — Erro ao gerar certificado:

No passo 2 do Fluxo Principal, ocorre falha na geração do PDF.

[RS] O sistema exibe a mensagem correspondente.

FE02 — Erro ao registrar emissão:

No passo 4 do Fluxo Alternativo, o registro histórico falha.

[RS] O sistema exibe a mensagem correspondente.

Pós-condição: O certificado foi baixado e/ou sua emissão foi registrada historicamente.

Mensagens

MSG01: Emissão registrada. Código: [Código de autenticidade].

A.28 UC28 — Baixar, Registrar e Emitir Boletim de Não Conformidade

Ator(es): Administrador de projetos, Analista de qualidade e Gerente de projeto.

Referência: Não há.

Pré-condição: O usuário está no *dashboard* do questionário e o questionário possui ISEQ calculado.

Fluxo Principal

[EV] O usuário clica em **Baixar Boletim (PDF)**.

[RS] O sistema gera o PDF do boletim e inicia o *download*. [FE01]

Fluxo Alternativo

FA01 — Emitir boletim aos representantes:

O usuário clica em **Emitir aos Representantes**.

[RS] O sistema solicita confirmação do envio por e-mail.

[EV] O usuário confirma.

[RS] O sistema envia o boletim aos representantes elegíveis, registra a operação e exibe a mensagem [MSG01].

FA02 — Registrar emissão do boletim:

O usuário clica em **Registrar Emissão do Boletim**.

[RS] O sistema exibe confirmação do **registro histórico auditável**.

[EV] O usuário confirma.

[RS] O sistema grava a emissão, atualiza a lista de emissões e exibe a mensagem [MSG02].

Fluxo de Exceção

FE01 — Erro ao gerar o boletim:

No passo 2 do Fluxo Principal, ocorre falha na geração do PDF.

[RS] O sistema exibe a mensagem correspondente.

FE02 — Erro ao emitir aos representantes:

No passo 4 do Fluxo Alternativo FA01, a operação falha.

[RS] O sistema exibe a mensagem correspondente.

FE03 — Erro ao registrar emissão:

No passo 4 do Fluxo Alternativo FA02, o registro histórico falha.

[RS] O sistema exibe a mensagem correspondente.

Pós-condição: O boletim foi baixado, registrado e/ou emitido aos representantes, conforme a ação escolhida.

Mensagens

MSG01: Boletim emitido para os representantes elegíveis.

MSG02: Emissão registrada. Código: [Código de autenticidade].

A.29 UC29 — Configurar *Token* Pessoal de IA

Ator(es): Usuário autenticado.

Referência: Não há.

Pré-condição: O usuário está autenticado e acessa **Configurações > *Token* de IA**.

Fluxo Principal

[EV] O usuário acessa a seção ***Token* de IA** nas Configurações.

[RS] O sistema carrega o *status* atual do *token*, o provedor configurado e uma dica parcial do valor já salvo, quando existir.

[EV] O usuário informa um novo *token* pessoal e escolhe se deseja visualizá-lo temporariamente na tela.

[EV] O usuário clica em **Salvar**.

[RS] O sistema persiste o *token*, atualiza o *status* para **Configurado** e exibe a mensagem [MSG01].

Fluxo Alternativo

FA01 — Remover *token* configurado:

Quando já existe *token* salvo, o usuário clica em **Remover**.

[RS] O sistema solicita confirmação.

[EV] O usuário confirma.

[RS] O sistema exclui o *token* salvo, atualiza o *status* para **Não configurado** e exibe a mensagem [MSG02].

Fluxo de Exceção

FE01 — *Token* inválido ou fora do padrão aceito:

No passo 4 do Fluxo Principal, o *token* não atende às validações mínimas de tamanho.

[RS] O sistema impede o envio e exibe as mensagens de validação.

FE02 — Erro ao salvar ou remover:

No passo 5 do Fluxo Principal ou no passo 4 do Fluxo Alternativo, a API falha.

[RS] O sistema exibe a mensagem de erro correspondente.

Pós-condição: O *token* pessoal de IA foi configurado ou removido.

Mensagens

MSG01: *Token* salvo com sucesso.

MSG02: *Token* removido com sucesso.

A.30 UC30 — Consultar análises assistidas por inteligência artificial

Ator(es): Administrador de projetos, Analista de qualidade e Gerente de projeto.

Referência: Não há.

Pré-condição: O usuário está no *dashboard* do questionário e os recursos de IA estão disponíveis para consulta.

Fluxo Principal

[RS] O *dashboard* do questionário exibe os *widgets* de **Explicação dos Resultados**, **Análise de Justificativas**, **Relatório de Risco** e ***Chat* com IA**.

[EV] O usuário aciona um dos *widgets* analíticos ou envia uma pergunta pelo *chat* contextual.

[RS] O sistema envia o contexto do questionário ao serviço de IA e aguarda a resposta.

[RS] O sistema apresenta o texto analítico gerado na própria interface do *widget* ou do *chat*.

Fluxo Alternativo

FA01 — Limpar conversa do *chat*:

No *widget* de *chat*, o usuário clica em **Limpar conversa**.

[RS] O sistema remove o histórico local da conversa e mantém o contexto do questionário para novas perguntas.

Fluxo de Exceção

FE01 — Serviço de IA indisponível:

No passo 3 do Fluxo Principal, o serviço de IA não responde ou retorna erro.

[RS] O sistema exibe a mensagem de indisponibilidade no *widget* correspondente.

Pós-condição: O usuário obteve explicações, análises ou respostas contextuais baseadas nos dados do questionário.

Mensagens: Não há mensagens fixas além dos erros de indisponibilidade do serviço.

A.31 UC31 — Gerenciar Expiração de Sessão

Ator(es): Usuário autenticado.

Referência: Não há.

Pré-condição: O usuário está com uma sessão autenticada prestes a expirar.

Fluxo Principal

[RS] O sistema monitora a expiração do *token* de acesso e agenda um aviso próximo ao vencimento.

[RS] Ao se aproximar da expiração, o sistema exibe um aviso ao usuário. [FA01] [FA02]

[EV] O usuário escolhe a ação disponível no aviso.

[RS] O sistema executa a ação escolhida.

Fluxo Alternativo

FA01 — Estender a sessão:

Quando não há rascunho pendente a ser preservado, o sistema oferece a opção **Estender sessão**.

[EV] O usuário escolhe estender a sessão.

[RS] O sistema utiliza o *refresh token*, agenda novo vencimento e exibe mensagem de sucesso.

FA02 — Salvar rascunho antes do *logout*:

Quando o usuário está respondendo questionário com alterações em andamento, o sistema oferece salvar rascunho antes do encerramento da sessão.

[EV] O usuário confirma o salvamento.

[RS] O sistema salva o rascunho, informa o sucesso e encerra a sessão em seguida.

Fluxo de Exceção

FE01 — Falha ao estender a sessão:

No fluxo FA01, a renovação com *refresh token* falha.

[RS] O sistema informa que não foi possível estender a sessão e agenda o *logout*.

FE02 — Falha ao salvar rascunho antes do *logout*:

No fluxo FA02, ocorre erro ao persistir o rascunho.

[RS] O sistema alerta que os dados não salvos podem ser perdidos e encerra a sessão.

FE03 — Expiração consumada sem ação do usuário:

O usuário não responde ao aviso dentro do período de tolerância.

[RS] O sistema encerra a sessão automaticamente e redireciona para o *login*.

Pós-condição: A sessão foi renovada ou encerrada de forma controlada, com tentativa de preservação do trabalho em andamento quando aplicável.

Mensagens: Não há mensagens fixas além das notificações de sessão exibidas pela interface.

APÊNDICE B – Requisitos e Regras de Negócio

Os requisitos funcionais da aplicação são especificados pelos casos de uso apresentados no Apêndice A. Este apêndice complementa essa especificação com as regras de negócio que restringem e orientam o comportamento dos fluxos implementados. Não são introduzidos novos requisitos nesta seção; preservam-se os casos de uso e as regras levantados para o artefato.

B.1 BR01 – Questionário

O questionário é a principal e única forma avaliativa para mensurar a conformidade das implicações éticas e a qualidade do *software* durante o processo de desenvolvimento. Ele é composto por perguntas de teor ético relacionadas à engenharia de *software* e suas etapas, as quais deverão ser respondidas pelos membros autorizados (vide BR05). As respostas deverão ser fornecidas com as opções "SIM" ou "NÃO", acompanhadas de justificativa escrita. O respondente poderá reforçar sua justificativa com *links* ou arquivos de apoio. O envio de anexos é opcional; apenas um *link* principal é armazenado por resposta, extraído da lista de anexos fornecida. É obrigatório responder a todas as perguntas até o prazo estipulado (vide BR08).

B.1.1 Elaboração e Atribuição de Perguntas

O ADMIN deverá elaborar e atribuir as perguntas do questionário durante a configuração do projeto. Cada etapa do projeto terá um conjunto de perguntas constituindo um questionário. O número e o conteúdo das perguntas poderão variar conforme o representante e sua função no projeto. Os questionários deverão ser personalizados conforme o perfil de cada membro, apresentando perguntas direcionadas aos encargos correspondentes. Cada pergunta poderá ser associada a um ou mais encargos, de modo que apenas os representantes com aquele encargo a visualizem e a respondam. As perguntas também poderão estar associadas a uma ou mais etapas, que atuam como categorias de agrupamento dentro do questionário, utilizadas para o cálculo do IEM (Índice de Etapa por Membro).

B.1.2 Reutilização e Alteração de Questionários

O questionário criado e definido como padrão deverá ser mantido e poderá ser reutilizado em outros projetos (vide BR09). O ADMIN poderá adicionar, remover ou editar

perguntas, desde que as alterações sejam realizadas em iterações ou etapas que ainda não entraram em vigor.

B.1.3 Rascunho

O sistema deverá permitir que o respondente salve um rascunho das respostas antes de submetê-las definitivamente. O rascunho deverá ser enviado ao servidor com *status* “Em Progresso”, persistido localmente no navegador e salvo automaticamente no armazenamento local quando a sessão expirar. O rascunho deverá ser removido do *cache* local após a submissão definitiva bem-sucedida.

B.1.4 Modos de Visualização

O questionário deverá ser exibido em dois modos:

- Resposta: modo de resposta ativo, disponível para respondentes com *status* “Pendente” ou “Em Progresso”.

- Visualização: modo somente leitura, ativado automaticamente quando o respondente já concluiu ou quando o usuário acessar em modo de leitura.

B.1.5 Funcionalidades do Sistema

O sistema deverá permitir as seguintes operações:

- Criação de questionários com perguntas direcionadas ao perfil de cada membro.
- Edição do conteúdo das perguntas em iterações futuras.
- Visualização dos questionários atribuídos por meio de interface clara.
- Exclusão de questionários durante a fase de criação do projeto.
- Importação de questionários de projetos existentes no sistema para facilitar a reutilização (vide BR09).

B.1.6 Barra de Progresso

O sistema deverá exibir o percentual de perguntas respondidas em tempo real.

B.1.7 Validações

O sistema deverá impedir o envio do questionário enquanto houver perguntas sem resposta ou sem a respectiva descrição obrigatória. Ao tentar anexar evidências sem ter selecionado “SIM” ou “NÃO”, o sistema deverá exibir o aviso: “Selecione ’Sim’ ou ’Não’ antes de anexar evidências.”

B.1.8 Notificações

O sistema deverá enviar notificações internas ao ADMIN quando um questionário entrar em atraso e quando o ISEQ de um questionário for calculado. O sistema deverá enviar notificações automáticas aos membros autorizados quando novos questionários forem atribuídos ou quando houver alterações em perguntas ainda não respondidas.

B.1.9 Exportação de Dados

Os dados brutos de respostas e os índices calculados deverão poder ser exportados nos formatos CSV e JSON. A exportação poderá ser anônima ou nominal.

B.1.10 Usabilidade

O sistema deverá fornecer mensagens detalhadas e claras para informar o usuário sobre erros ou configurações incorretas durante a criação e edição do questionário. Essas mensagens deverão especificar o tipo de erro identificado, apresentar orientações para correção e destacar visualmente os campos que necessitam de ajuste.

B.2 BR02 – Projeto

O projeto representa o desenvolvimento de *software* de uma empresa. O ADMIN deverá criar e configurar os projetos no sistema, podendo cadastrar informações essenciais de múltiplos projetos.

B.2.1 Tipos de Estrutura

O projeto deverá ser configurado em uma das seguintes estruturas:

- CASCATA: o projeto é dividido em etapas fixas, cada uma com um questionário associado. As etapas padrão são: Iniciação, Requisitos, Projeto, Desenvolvimento e Testes. O ADMIN poderá adicionar, renomear ou remover etapas.

- ITERATIVO: o projeto é dividido em iterações com ciclos repetidos, cada uma com um questionário associado. O ADMIN deverá definir a duração de cada iteração e o número total de iterações.

B.2.2 *Status* do Projeto

O projeto deverá possuir os seguintes *status*:

- Rascunho: projeto criado, mas ainda não publicado. As configurações poderão ser alteradas livremente.

- Aberto: projeto publicado. Os representantes poderão responder aos questionários ativos. A edição será restrita.

- Arquivado: projeto encerrado antes da conclusão. Não poderá ser reaberto pelo sistema.

- Concluído: projeto encerrado com cálculo do ISEP consolidado.

- Excluído: projeto marcado como excluído. Não será visível nas listagens padrão.

B.2.3 *Status* de Linha do Tempo dos Questionários

Os questionários de um projeto deverão possuir os seguintes *status* independentes do *status* do projeto:

- PENDENTE: o período de aplicação ainda não foi iniciado.

- EM ANDAMENTO: o período de aplicação está ativo.

- CONCLUÍDO: todos os respondentes submeteram ou o questionário foi encerrado forçadamente.

- ATRASADO: o prazo de aplicação expirou sem que todos respondessem.

B.2.4 Dados Necessários para Criação do Projeto

Os seguintes dados deverão ser fornecidos ao configurar um projeto:

- Nome do projeto.

- Tipo de estrutura (CASCATA ou ITERATIVO).

- Data de início.

- Prazo final (opcional).

- *Template* de base (opcional) (vide BR09).

- Para CASCATA: lista de etapas com nome, peso, sequência e duração em dias. As datas de aplicação serão calculadas automaticamente pelo sistema com base na data de início e na duração de cada etapa.

- Para ITERATIVO: número de iterações, duração de cada iteração em dias e lista de iterações com nome, peso e datas de aplicação.

- Questionários com perguntas associadas a encargos e etapas.

- Representantes com nome, e-mail, peso e encargos.

B.2.5 Publicação do Projeto

O projeto poderá ser salvo como Rascunho ou publicado diretamente como Aberto. A publicação poderá ser realizada posteriormente a partir da tela de detalhes do projeto.

B.2.6 Edição de Projeto Aberto

Ao editar um projeto com *status* Aberto, o sistema deverá executar uma simulação prévia (*dry-run*) antes de aplicar as alterações. As mudanças deverão ser simuladas e um resumo das alterações deverá ser retornado ao ADMIN, contendo a lista de adições, remoções, atualizações e avisos. O ADMIN deverá confirmar antes da aplicação definitiva. Alterações que violem regras do sistema deverão ser bloqueadas (vide BR14).

B.2.7 Encerramento do Projeto

O ADMIN poderá encerrar o projeto por meio de ação na tela do *dashboard*. Ao confirmar, o sistema deverá calcular o ISEQ consolidado com os questionários disponíveis e transitar o *status* para Concluído.

B.2.8 Exclusão do Projeto

O ADMIN poderá excluir o projeto a partir da tela de detalhes. A ação será irreversível e exigirá confirmação. Após a exclusão, o sistema deverá navegar para a listagem de projetos.

B.2.9 Configuração das Iterações

Os dados configurados na iteração anterior deverão ser exibidos como padrão para a nova iteração. A cada nova iteração, o ADMIN poderá realizar configurações personalizadas, como adicionar novas perguntas ao questionário e alterar o peso das etapas. Novas perguntas adicionadas pelo administrador deverão ser do tipo Personalizada. Perguntas com classificação Recorrente serão incluídas automaticamente pelo sistema em todos os questionários subsequentes, dispensando inclusão manual iteração a iteração.

B.2.10 Funcionalidades do Sistema

O sistema deverá permitir as seguintes operações:
- Criar novos projetos.
- Alterar as configurações de projetos existentes.
- Visualizar os detalhes de projetos.
- Arquivar projetos quando não forem mais necessários.
- Excluir projetos de forma irreversível mediante confirmação.

B.2.11 Validações

O sistema deverá validar automaticamente as configurações de projetos, evitando inconsistências como prazos incompatíveis ou pesos incoerentes.

B.2.12 Notificações

O sistema deverá enviar notificações internas ao ADMIN quando o projeto entrar em atraso e quando o ISEP consolidado do projeto for calculado.

B.3 BR03 – Membro da Equipe

Os membros da equipe representam as partes envolvidas nos projetos. Cada membro deverá assumir um ou mais encargos, correspondentes às suas responsabilidades dentro do projeto.

B.3.1 Encargos

Os encargos disponíveis no sistema são: Cliente, Consultor, Desenvolvedor, Líder Técnico, Gerente de Projeto, Arquiteto, Analista, Testador, *Designer* e *Stakeholder*. Um representante poderá ter múltiplos encargos dentro do mesmo projeto.

B.3.2 Perfis de Acesso

O sistema deverá possuir dois perfis de acesso:

- ADMIN (Analista de Qualidade): acesso completo ao sistema, incluindo criação e gerenciamento de projetos, publicação, arquivamento, exclusão, encerramento de questionários e acesso a *dashboards* analíticos.

- USER (Representante): acesso restrito à resposta dos questionários dos projetos em que está vinculado e à visualização do *dashboard* individual após a submissão.

B.3.3 Cadastro de Membros

O cadastro dos representantes deverá ser realizado pelo ADMIN durante a criação ou edição do projeto. Para cada representante, deverão ser registrados: primeiro nome e sobrenome, e-mail válido para *login*, peso (que influencia o cálculo do ISEQ, vide BR04) e um ou mais encargos.

B.3.4 Primeiro Acesso

No primeiro acesso, o representante deverá definir sua própria senha por meio do fluxo de primeiro acesso. A senha não deverá ser criada pelo ADMIN.

B.3.5 Vinculação a Projetos

Um representante poderá estar vinculado a múltiplos projetos simultaneamente, com encargos diferentes em cada um.

B.3.6 Peso por Representante

Cada representante deverá possuir um peso numérico utilizado no cálculo ponderado do ISEQ da iteração ou etapa (vide BR04). O peso poderá ser ajustado pelo ADMIN durante a criação ou edição do projeto.

B.4 BR04 – Classificação Ética do Projeto

B.4.1 Descrição

O sistema deverá processar os questionários éticos vinculados ao projeto para apurar os indicadores de conformidade ética dos representantes, dos questionários e do projeto, considerando o tipo de projeto configurado, os papéis exercidos pelos representantes, as etapas relacionadas às perguntas, os pesos definidos para representantes e questionários e as respostas válidas registradas no sistema.

Para projetos do tipo iterativo incremental, o sistema deverá considerar a iteração como unidade organizadora dos questionários, permitindo a existência de diferentes questionários em diferentes *sprints*, cada um com suas próprias perguntas, pesos e período de aplicação.

Para projetos do tipo cascata, o sistema deverá considerar a etapa como unidade organizadora dos questionários, permitindo a existência de questionários associados às etapas do ciclo do projeto, conforme a configuração realizada pelo analista.

O resultado da aplicação de cada questionário não deverá ser tratado como ISEP. O indicador calculado no nível do questionário deverá ser denominado ISEQ, Indicador Sistêmico Ético do Questionário. O ISEP, Indicador Sistêmico Ético do Projeto, deverá ser reservado exclusivamente para o resultado geral consolidado do projeto.

B.4.2 Elementos

B.4.2.1 Projeto

O projeto é a entidade principal da avaliação ética. Ele agrega o tipo de projeto, as etapas, as iterações quando existirem, os representantes, os papéis atribuídos a esses representantes, os questionários, as perguntas e as respostas utilizadas na classificação ética

O projeto poderá ser configurado manualmente ou a partir de *template*. Na criação do projeto, o sistema prevê o preenchimento do tipo, das etapas, dos representantes e dos questionários; além disso, quando o tipo iterativo incremental for selecionado, o sistema também deverá calcular e exibir o número de iterações com base nas datas e na duração informada.

B.4.2.2 Tipo de projeto

O tipo de projeto define a forma de organização dos questionários e da classificação ética. O sistema deverá admitir, no mínimo, os tipos Iterativo Incremental e Cascata.

No tipo Iterativo Incremental, os questionários são organizados por iteração. No tipo Cascata, os questionários são organizados por etapa.

B.4.2.3 Iteração

A iteração é a unidade organizadora dos questionários em projetos do tipo iterativo incremental. Cada iteração representa um ciclo delimitado de aplicação, como uma *sprint*, com período próprio e com um ou mais questionários associados.

Em projetos iterativos, uma mesma iteração poderá conter perguntas relacionadas a diferentes etapas do projeto, pois as etapas funcionam como dimensões analíticas da pergunta e não como uma sequência obrigatória dentro da sprin.

B.4.2.4 Etapa

A etapa representa uma dimensão estrutural do projeto e também uma dimensão de classificação das perguntas. O sistema deverá permitir o cadastro, edição, exclusão e reutilização de etapas na configuração do projeto.

Em projetos do tipo cascata, a etapa também poderá funcionar como unidade organizadora do questionário, uma vez que a visão de manutenção de questionários nesse tipo exibe os questionários por etapa. Em projetos iterativos, a etapa não substitui a iteração como unidade organizadora do questionário, mas continua identificando a qual parte do processo a pergunta se relaciona.

B.4.2.5 Questionário

O questionário é o instrumento de avaliação ética aplicado em um recorte específico do projeto. Ele possui nome, faixa de aplicação, peso e conjunto de perguntas.

Em projetos iterativos, o questionário deverá estar associado a uma iteração. Em projetos do tipo cascata, o questionário deverá estar associado a uma etapa.

B.4.2.6 Pergunta

A pergunta é a unidade mínima de verificação ética. Cada pergunta integra um questionário e poderá estar vinculada a uma ou mais etapas e a um ou mais papéis, conforme a configuração realizada no sistema.

A pergunta poderá ser do tipo Base, quando oriunda do *template* inicial e imutável, ou do tipo Personalizada, quando criada pelo administrador. Toda pergunta possui ainda uma classificação de escopo — Projeto Inteiro, Recorrente ou Etapa Atual — que define

em quais questionários ela deverá ser exibida, conforme BR04.1. A pergunta preserva o vínculo com seus encargos relacionados, isto é, com os papéis e as etapas aplicáveis.

B.4.2.7 Representante

O representante é a pessoa vinculada ao projeto que participa do processo de avaliação ética por meio do registro das respostas aplicáveis ao conjunto de papéis que exerce. Cada representante possui peso próprio e poderá acumular um ou mais papéis no projeto.

O sistema deverá permitir a inclusão de representantes com nome, sobrenome, email, peso e um ou mais encargos, e deverá verificar se os encargos definidos para o projeto estão representados por ao menos um participante, ainda que o avanço possa ocorrer com alerta.

B.4.2.8 Papel

O papel representa a função exercida pelo representante no contexto do projeto. O papel determina a elegibilidade das perguntas que deverão compor o conjunto de avaliação de cada representante.

Um mesmo representante poderá exercer múltiplos papéis simultaneamente. Nessa situação, o conjunto de perguntas elegíveis do representante deverá corresponder à união das perguntas vinculadas a todos os papéis que ele exerce, sem duplicidade quando a mesma pergunta estiver vinculada a mais de um de seus papéis.

B.4.2.9 Resposta

A resposta é o registro produzido pelo representante para uma pergunta elegível em determinado questionário. A resposta deverá manter vínculo com a pergunta, com o questionário, com as etapas associadas e com os papéis relacionados à pergunta, garantindo rastreabilidade para classificação e auditoria.

B.4.2.10 ICP

O ICP, Indicador de Conformidade do Participante, representa o resultado individual do representante em um questionário específico. O cálculo do ICP deverá considerar apenas as perguntas elegíveis para o conjunto de papéis exercidos pelo representante naquele questionário.

B.4.2.11 ISEQ

O ISEQ, Indicador Sistêmico Ético do Questionário, representa o resultado consolidado de um questionário específico. O ISEQ deverá ser apurado a partir da média ponderada dos ICPs dos representantes elegíveis que responderam ao questionário, observando os pesos dos representantes no projeto.

O ISEQ é o indicador aplicável ao resultado de cada questionário, independentemente de o projeto ser iterativo ou cascata. Em projetos iterativos, o questionário estará vinculado a uma iteração; em projetos cascata, o questionário estará vinculado a uma etapa.

B.4.2.12 ISEP

O ISEP, Indicador Sistêmico Ético do Projeto, representa o resultado geral consolidado do projeto. O ISEP deverá ser apurado a partir da consolidação ponderada dos ISEQs dos questionários que compõem o projeto, observando os pesos definidos para esses questionários.

O ISEP não deverá ser utilizado para designar o resultado de um questionário isolado. Seu uso deverá ficar restrito ao nível geral do projeto, como indicador final de classificação ética.

B.4.2.13 Certificação

A certificação representa o resultado institucional da avaliação ética do projeto. Sua emissão deverá considerar o resultado geral do projeto e os critérios mínimos de cobertura exigidos pela política de avaliação.

B.4.2.14 Regras gerais

O sistema deverá classificar eticamente projetos dos tipos Iterativo Incremental e Cascat.

O sistema deverá calcular o ICP por representante e por questionário, considerando somente as perguntas elegíveis para os papéis do representante.

O sistema deverá calcular o ISEQ por questionário, a partir da média ponderada dos ICPs dos representantes elegívei.

O sistema deverá calcular o ISEP do projeto, a partir da consolidação ponderada dos ISEQs dos questionários do projeto.

Em projetos iterativos, os questionários deverão ser organizados por iteraçã.

Em projetos cascata, os questionários deverão ser organizados por etapa.

Uma pergunta poderá estar vinculada a múltiplos papéis e a múltiplas etapa.

Um representante poderá exercer múltiplos papéis simultaneamente.

Perguntas repetidas por acúmulo de papéis do mesmo representante deverão ser consideradas uma única vez no cálculo.

A ausência de perguntas para determinado papel em um questionário não deverá gerar penalização automática ao representante.

A ausência de perguntas para determinada etapa em um questionário não deverá invalidar o cálculo do questionário.

O sistema deverá preservar a rastreabilidade entre questionário, pergunta, etapa, papel, representante e resposta.

B.4.2.15 Regras de elegibilidade

Uma pergunta deverá ser considerada elegível para um representante quando houver interseção entre pelo menos um dos papéis do representante e pelo menos um dos papéis vinculados à pergunta.

Quando o representante possuir mais de um papel, o conjunto de perguntas elegíveis deverá corresponder à união das perguntas vinculadas aos seus papéis, sem duplicidade de contagem.

Perguntas não elegíveis para determinado representante não deverão compor o cálculo do seu ICP.

B.4.2.16 Regras de cobertura por papel e etapa

A existência de um papel no projeto não obriga que ele possua perguntas em todos os questionários. Se um papel não possuir perguntas vinculadas em determinado questionário, ele não deverá compor o denominador do ICP dos representantes naquele questionário.

A existência de uma etapa no projeto também não obriga que ela esteja coberta em todos os questionários. Em projetos iterativos, a ausência de perguntas para determinada etapa em uma *sprint* deverá ser tratada como ausência de cobertura analítica daquela etapa no questionário, sem impedir o cálculo do ISEQ.

Em projetos cascata, a etapa poderá ser a unidade organizadora do questionário. Nessa situação, a etapa de referência do questionário deverá ser mantida como identificação principal do recorte avaliado, sem impedir que perguntas internas do questionário também possuam vínculos analíticos adicionais quando a modelagem do sistema assim permitir.

B.4.2.17 Regras de validade das respostas

O cálculo do ICP deverá utilizar somente respostas válidas registradas para perguntas elegíveis do representante no questionário.

O sistema deverá exigir integridade entre representante, pergunta, papéis associados, etapas relacionadas e questionário de origem, de forma a permitir auditoria e reprocessamento da classificação ética.

B.4.2.18 Cálculo dos indicadores

O ICP do representante em um questionário deverá ser calculado exclusivamente sobre o conjunto de perguntas elegíveis para os papéis que ele exerce. O ISEQ deverá consolidar os ICPs dos representantes elegíveis, observando seus respectivos pesos, e o ISEP deverá consolidar os ISEQs dos questionários do projeto, também de forma ponderada.

As definições matemáticas completas e a interpretação de cada variável encontram-se na Seção 4.4.

B.4.2.19 Regras para certificação

A emissão de certificação ética não deverá depender exclusivamente do resultado numérico de um único questionário, mas do resultado geral consolidado do projeto e do atendimento aos critérios mínimos definidos para cobertura e consistência da avaliação

Quando o processo exigir verificação por faixas mínimas aceitáveis, o sistema deverá impedir a emissão de certificado e permitir a emissão de boletim de não conformidade ética caso os requisitos mínimos não tenham sido alcançados.

B.4.2.20 Redação formal

O sistema deverá processar os questionários éticos do projeto para calcular o Indicador de Conformidade do Participante (ICP) de cada representante, o Indicador Sistêmico Ético do Questionário (ISEQ) de cada questionário e o Indicador Sistêmico Ético do Projeto (ISEP), considerando o tipo de projeto, os pesos definidos para representantes e questionários, os papéis exercidos pelos representantes, as etapas associadas às perguntas e as respostas válidas registradas no sistema.

Nos projetos do tipo Iterativo Incremental, os questionários deverão ser organizados por iteração. Nos projetos do tipo Cascata, os questionários deverão ser organizados por etapa.

O conjunto de perguntas elegíveis de um representante em determinado questionário deverá corresponder à união das perguntas vinculadas aos papéis que ele exerce no projeto, sem duplicidade quando uma mesma pergunta estiver associada a mais de um de seus papéis. O ICP deverá ser calculado exclusivamente sobre esse conjunto elegível.

O ISEQ de cada questionário deverá corresponder à média ponderada dos ICPs dos representantes elegíveis, observando o peso de cada representante no projeto. O ISEP do projeto deverá corresponder à média ponderada dos ISEQs dos questionários que compõem o projeto, observando o peso atribuído a cada questionário.

A ausência de perguntas vinculadas a determinado papel em um questionário não deverá implicar penalização automática ao representante. A ausência de perguntas vinculadas a determinada etapa também não deverá invalidar o cálculo do questionário, devendo ser tratada conforme o tipo de projeto e a cobertura configurada para o recorte avaliad.

A emissão de certificação ética deverá depender do resultado geral do projeto e do atendimento aos critérios mínimos de cobertura e conformidade definidos para a avaliação.

B.4.2.21 Regras de interface e interação

B.4.2.21.1 Regras gerais de edição do projeto

O sistema não deverá permitir alterar a configuração estrutural do projeto após a existência de qualquer resposta registrada para qualquer questionário do projeto.

Quando houver ao menos uma resposta registrada, o projeto deverá ser tratado como versionado para fins de auditoria.

Alterações estruturais somente poderão ocorrer por clonagem de projeto ou criação de nova versão, nunca por edição silenciosa do projeto já respondido.

O sistema deverá exibir bloqueio e mensagem clara quando o usuário tentar editar campos restritos após início das respostas.

B.4.2.21.2 Regras para questionários em projetos iterativos

O usuário não poderá editar o peso de um questionário após ao menos um usuário ter respondido o primeiro questionário vinculado àquela sequência.

O usuário não poderá editar o nome, a faixa de aplicação ou a sequência de um questionário que já possua respostas registradas.

O usuário não poderá remover perguntas de um questionário que já tenha sido respondido por qualquer representante.

O usuário não poderá adicionar novas perguntas a um questionário já respondido, caso isso altere a base de cálculo já aplicada.

O usuário não poderá editar perguntas do tipo Base em nenhuma circunstância, independentemente do *status* do questionário ou do projeto.

O usuário não poderá editar o vínculo de etapa ou papel de perguntas já respondidas.

O usuário não poderá alterar o período de aplicação de um questionário já iniciado.

O sistema deverá bloquear a alteração de peso de questionários futuros quando a alteração puder afetar a proporcionalidade do resultado global já em processamento.

B.4.2.21.3 Regras para questionários em projetos cascata

O usuário não poderá editar o peso de um questionário associado a uma etapa após existir qualquer resposta registrada para esse questionário.

O usuário não poderá alterar a etapa vinculada ao questionário depois do início das respostas.

O usuário não poderá editar perguntas do tipo Base em projetos cascata, independentemente do *status* do questionário.

B.5 BR04.1 — Classificação e Tipo das Perguntas

Descrição

Toda pergunta vinculada a um questionário possui dois atributos independentes: o tipo, que indica a origem da pergunta e determina seu grau de imutabilidade; e a classificação, que define o escopo de exibição da pergunta dentro do projeto.

Tipo da Pergunta

O tipo da pergunta é derivado automaticamente de sua origem. Uma pergunta do tipo Base foi criada pelo *template* inicial do sistema e não pode ter seu conteúdo editado nem ser excluída, independentemente de qualquer configuração posterior. Uma pergunta do tipo Personalizada foi criada pelo administrador do projeto e pode ser editada e excluída livremente.

Classificação da Pergunta

A classificação indica quando a pergunta deve ser exibida no ciclo do projeto. O sistema deverá admitir as seguintes classificações:

Projeto Inteiro: perguntas que tratam de governança geral, políticas de uso de inteligência artificial, definição de responsabilidades, canais de reporte e limites éticos estruturais do projeto. Por representarem decisões que não variam a cada iteração ou etapa, deverão ser exibidas apenas uma vez, preferencialmente no primeiro questionário do projeto, seja ele vinculado à primeira *sprint* do iterativo ou à fase inicial do cascata.

Recorrente: perguntas que funcionam como verificação mínima obrigatória antes de cada nova iteração ou etapa. Verificam o alinhamento ético da *sprint* em relação ao que foi definido no projeto, a identificação de novos riscos desde a iteração anterior, a rastreabilidade atualizada e a revisão humana de artefatos gerados com apoio de inteligência artificial. Quando o administrador criar uma pergunta do tipo Personalizada com classificação Recorrente, o sistema deverá adicioná-la automaticamente a todos os questionários existentes e garantir sua presença nos questionários futuros do projeto.

Etapa Atual: perguntas associadas ao foco da iteração ou etapa em andamento, variando conforme o estado do projeto. Perguntas com essa classificação são sempre do tipo Personalizada, uma vez que perguntas do tipo Base representam diretrizes de alcance geral e não podem ser restritas a uma única iteração específica.

Regras de Combinação

Perguntas do tipo Base podem ser classificadas como Projeto Inteiro ou Recorrente. A combinação de tipo Base com classificação Etapa Atual não é válida e não deverá ser permitida pelo sistema.

Perguntas do tipo Personalizada podem ter qualquer uma das três classificações: Projeto Inteiro, Recorrente ou Etapa Atual.

Estrutura de Exibição por Questionário

O primeiro questionário de um projeto deverá concentrar todas as perguntas com classificação Projeto Inteiro — tanto as do *template* inicial quanto as eventualmente cri-

adas pelo administrador com essa classificação —, além das perguntas Recorrentes e das perguntas de Etapa Atual definidas para o primeiro recorte avaliado.

Os questionários seguintes deverão carregar apenas as perguntas Recorrentes e as perguntas de Etapa Atual associadas ao recorte em questão, evitando a repetição de perguntas de governança estrutural já respondidas.

Vinculação de Etapas e Papéis

A vinculação de etapas às perguntas é recomendativa, não rígida. O sistema deverá permitir o mapeamento de uma pergunta para uma ou mais etapas com base em equivalência funcional, e não em igualdade textual do nome da etapa. A vinculação de papéis também é recomendativa: cada pergunta aceita um ou mais papéis associados, mas o sistema não poderá depender da existência exata dos papéis previstos no *template* de origem.

Ausência de Etapa ou Papel

Se o projeto não possuir etapa equivalente à prevista na pergunta, a pergunta deverá ser ignorada ou remapeada para a fase funcionalmente mais próxima, sem gerar falha de configuração. Se um papel previsto na pergunta não existir no projeto, a pergunta não deverá impedir o uso do questionário; o sistema deverá permitir substituição por papel equivalente ou execução com o conjunto de papéis disponível.

B.6 BR05 – Controle de Acesso ao Sistema

O sistema deverá controlar o acesso por perfis, restringindo as ações conforme as permissões atribuídas a cada usuário.

B.6.1 Perfil ADMIN (Analista de Qualidade)

O ADMIN deverá ter acesso completo ao sistema, incluindo:
- Criar, editar, publicar, arquivar, excluir e encerrar projetos.
- Configurar etapas, iterações, questionários, perguntas e representantes.
- Ajustar pesos de etapas, iterações e representantes.
- Encerrar questionários forçadamente.
- Reagendar questionários.
- Enviar lembretes manuais de resposta.
- Acessar o *dashboard* analítico completo.
- Exportar dados em CSV e JSON.
- Visualizar respostas individuais de qualquer representante.
- Visualizar respostas consolidadas do projeto e dos questionários.
- Acessar e gerenciar *templates*.

B.6.2 Perfil USER (Representante)

O USER deverá ter acesso restrito, incluindo:
- Realizar *login* com o e-mail registrado no projeto.
- Visualizar e responder os questionários dos projetos em que está vinculado.
- Salvar rascunho de respostas.
- Visualizar o *dashboard* individual após submeter o questionário.

B.7 BR06 – Boletim de Não Conformidade Ética

O sistema deverá gerar automaticamente o boletim de não conformidade ética quando os resultados de um ou mais membros estiverem abaixo da faixa mínima aceitável estabelecida na criação do projeto. O boletim deverá apontar os desvios éticos identificados e propor ações corretivas.

B.7.1 Conteúdo do Boletim

O boletim deverá conter:
- Listagem detalhada das perguntas cujas respostas ficaram abaixo do limite mínimo.
- Análise do impacto dos desvios identificados na qualidade ética do projeto.
- Orientações para revisão e correção das ações, com objetivo de atingir a conformidade mínima exigida.

B.7.2 Processo de Revisão

A iteração em que o boletim for gerado não poderá ser concluída enquanto as não conformidades não forem revisadas. Os membros deverão revisar as respostas e submetê-las novamente. Após a revisão, o sistema deverá recalcular os resultados para verificar se a classificação mínima foi alcançada.

B.7.3 Permissões

Apenas o ADMIN poderá gerar e emitir o boletim de não conformidade ética.

B.7.4 Histórico e Notificações

O sistema deverá armazenar todos os boletins gerados para consulta futura. Os membros com respostas identificadas como não conformes deverão receber notificações com o boletim e os prazos para revisão. O sistema deverá registrar a frequência de geração de boletins por projeto, possibilitando a análise de recorrências.

B.8 BR07 – Certificado de Ciência Ética

O sistema deverá emitir o certificado de ciência ética como documento formal que comprova a aprovação da conformidade ética do projeto, com base nas respostas dos questionários atribuídos aos membros.

B.8.1 Conteúdo do Certificado

O certificado deverá conter:

- Resumo dos resultados gerais do projeto, incluindo o ISEP consolidado e os ISEQs das iterações ou etapas concluídas (vide BR04).
- Declaração formal de que o projeto atingiu ou superou a faixa mínima aceitável de conformidade ética.
- Identificação do projeto, data e período da iteração.
- Código único de autenticidade para validação por terceiros.

B.8.2 Permissões

Apenas o ADMIN poderá gerar e emitir o certificado.

B.8.3 Certificado Consolidado

O sistema deverá gerar um certificado consolidado ao final do projeto, contendo o resumo das iterações concluídas e aprovadas ao longo do desenvolvimento.

B.8.4 Armazenamento

O sistema deverá armazenar todos os certificados emitidos, vinculados ao projeto correspondente, para consulta e exportação posterior.

B.9 BR08 – Previsão de Data Limite de Resposta dos Questionários

O sistema deverá armazenar e gerenciar as datas de aplicação dos questionários para cada questionário, etapa e iteração.

B.9.1 Projeto Cascata

As datas de aplicação das etapas deverão ser calculadas automaticamente pelo sistema com base na data de início do projeto e na duração em dias de cada etapa. O *frontend*

deverá enviar apenas a duração de cada etapa durante a criação; as datas resultantes serão retornadas pelo servidor.

B.9.2 Projeto Iterativo

As datas de aplicação de cada iteração deverão ser informadas manualmente pelo ADMIN durante a criação ou edição do projeto.

B.9.3 Reagendamento

O ADMIN poderá reagendar o período de aplicação de um questionário que esteja com *status* PENDENTE, desde que seja o primeiro questionário pendente e haja pelo menos um questionário já concluído. O reagendamento deverá exigir as novas datas de início e fim. O sistema deverá retornar as datas antigas e novas, o novo *status* do questionário e, quando aplicável, as novas datas da etapa ou iteração associada. Quando as novas datas ultrapassarem o prazo do projeto, o sistema deverá exibir um aviso, sem bloquear o reagendamento. O sistema deverá enviar notificações automaticamente aos representantes afetados.

B.9.4 Edição das Datas

Ao editar um projeto Aberto, as datas de aplicação dos questionários pendentes poderão ser alteradas. O resumo de mudanças retornado pela simulação prévia deverá incluir avisos sobre questionários e datas afetados (vide BR14).

B.10 BR09 – *Template*

O sistema deverá permitir a criação, o armazenamento e a reutilização de configurações de projetos por meio de *templates*.

B.10.1 Criação de *Template*

O ADMIN deverá poder criar um *template* a partir de um projeto existente, informando: nome do *template*, descrição e visibilidade (pública ou privada). *Templates* públicos deverão estar disponíveis para todos os ADMINs; *templates* privados deverão ser visíveis apenas ao seu criador.

B.10.2 Estrutura do *Template*

Um *template* poderá conter: etapas com nome, peso, sequência e duração em dias; iterações com nome e peso; representantes com nome, e-mail, peso e encargos; questionários

com perguntas associadas, cada uma com seu tipo (Base ou Personalizada), sua classificação de escopo (Projeto Inteiro, Recorrente ou Etapa Atual), seus papéis vinculados e suas etapas vinculadas; e tipo do projeto (Cascata ou Iterativo).

B.10.3 Tipos de Importação Parcial

O sistema deverá suportar importação parcial de *templates* pelos seguintes tipos: Etapas, Iterações, Representantes, Questionários ou Todos (*template* completo).

B.10.4 Uso na Criação de Projeto

Durante a criação de um novo projeto, o ADMIN poderá selecionar um *template* como base. O *template* deverá pré-popular os campos do formulário de criação. O ADMIN poderá ajustar as configurações antes de salvar.

B.10.5 Listagem de *Templates*

O ADMIN deverá visualizar todos os *templates* disponíveis com nome, descrição e tipo.

B.11 BR10 – Notificações Internas

O sistema deverá registrar e exibir notificações internas para comunicar eventos relevantes de governança ao usuário logado.

B.11.1 Eventos de Notificação

O sistema deverá gerar notificações para os seguintes eventos:

- Questionário Vencido: o prazo de resposta de um questionário foi ultrapassado.
- ISEQ do Questionário Calculado: o ISEQ de um questionário foi calculado após encerramento ou encerramento forçado.
- Projeto Vencido: o prazo do projeto foi ultrapassado.
- ISEP do Projeto Calculado: o ISEP consolidado do projeto foi calculado após encerramento.

B.11.2 Estrutura da Notificação

Cada notificação deverá conter: identificador único, título, mensagem detalhada, *status* (não lido, lido ou excluído), data e hora de criação e tipo do evento.

B.11.3 Gerenciamento

O usuário deverá poder marcar notificações como lidas ou excluí-las. As notificações deverão ser carregadas e atualizadas conforme o usuário interagir com o sistema.

B.12 BR11 – Inteligência Artificial

O sistema deverá integrar recursos de inteligência artificial no *dashboard* do questionário, disponíveis apenas para o ADMIN após o cálculo do ISEP.

B.12.1 Funcionalidades de IA

O sistema deverá disponibilizar os seguintes *widgets* de IA:

- Análise de Justificativas: geração de análise textual das justificativas e evidências fornecidas pelos respondentes. Tempo estimado de processamento: até 15 segundos. Resultado exibido em formato markdown. Acionado manualmente pelo ADMIN.

- Explicação dos Resultados ISEQ: geração de explicação textual dos resultados do ISEQ, faixas e métricas do questionário. Tempo estimado de processamento: até 15 segundos. Resultado exibido em formato markdown. Acionado manualmente pelo ADMIN.

- Relatório de Risco de Conformidade: geração de relatório identificando riscos de conformidade com base nos dados do questionário. Tempo estimado de processamento: até 20 segundos. Resultado exibido em formato markdown. Acionado manualmente pelo ADMIN.

- *Chat* Interativo: interface de *chat* com *streaming* de respostas em tempo real. O ADMIN poderá enviar perguntas livres sobre os dados do questionário. O sistema deverá oferecer perguntas sugeridas pré-definidas. O *chat* deverá manter o histórico da conversa durante a sessão. O *streaming* poderá ser cancelado a qualquer momento e o histórico poderá ser limpo.

B.12.2 Disponibilidade

Os *widgets* deverão ser independentes entre si. Caso o serviço de IA esteja indisponível, o *widget* afetado deverá exibir mensagem de erro sem travar a interface. A falha de um *widget* não deverá afetar os demais.

B.13 BR12 – Indicadores de Dívida Ética e Técnica

O sistema deverá calcular e exibir indicadores de dívida para cada questionário e para o projeto consolidado.

B.13.1 Dívida Ética

O sistema deverá representar o percentual de não conformidade ética do questionário com a seguinte sinalização visual:

- Verde: menos de 10% de dívida.
- Amarelo: entre 10% e 30% de dívida.
- Vermelho: acima de 30% de dívida.

B.13.2 Dívida Técnica

O sistema deverá representar o percentual de dívida técnica identificada no questionário com a seguinte sinalização visual:

- Verde: menos de 15% de dívida.
- Amarelo: entre 15% e 35% de dívida.
- Vermelho: acima de 35% de dívida.

B.13.3 Disponibilidade

Os indicadores deverão estar disponíveis tanto no *dashboard* do questionário quanto no *dashboard* do projeto. O sistema deverá permitir a visualização da evolução desses indicadores ao longo dos questionários.

B.14 BR13 – Dimensões de Governança

O sistema deverá classificar os resultados do ISEP em quatro dimensões de governança:

- Ética: conformidade relacionada a princípios éticos gerais.
- Processo: conformidade relacionada à qualidade e à aderência a processos.
- Fairness: conformidade relacionada à equidade e à imparcialidade.
- ESG: conformidade relacionada a critérios ambientais, sociais e de governança.

As dimensões deverão ser exibidas em formato de barras horizontais com valores percentuais. Quando não houver dados suficientes para calcular uma dimensão, o sistema deverá indicar ausência de valor para a dimensão correspondente. Os *scores* de governança deverão estar disponíveis tanto no *dashboard* do questionário quanto no *dashboard* do projeto.

B.15 BR14 – *Preview* de Alterações

Ao editar um projeto com *status* Aberto, o sistema deverá realizar uma simulação prévia das alterações antes de aplicá-las definitivamente.

B.15.1 Fluxo de Edição com Simulação Prévia

O ADMIN deverá acessar a tela de edição do projeto e realizar as alterações desejadas em etapas, iterações, questionários, representantes, datas ou pesos. Ao confirmar, o sistema deverá enviar os dados ao servidor, que retornará um resumo contendo: contagem de adições, remoções e atualizações por entidade; avisos sobre situações permitidas que possam impactar dados existentes; e bloqueios referentes a alterações que não possam ser aplicadas. O sistema deverá exibir o resumo ao ADMIN para confirmação antes de aplicar as mudanças definitivamente.

B.15.2 Bloqueios de Alteração

Alterações que impactem dados consolidados ou em andamento poderão ser bloqueadas pelo servidor. O sistema deverá exibir os motivos do bloqueio ao ADMIN antes de qualquer modificação ser persistida.

APÊNDICE C – Matriz de Rastreabilidade

A matriz apresentada no Quadro 15 relaciona os objetivos específicos aos casos de uso, às regras de negócio e às evidências utilizadas na avaliação do artefato.

Quadro 15 – Matriz de rastreabilidade entre objetivos, artefatos e evidências.

Objetivo	**Funcionalidades e casos de uso**	**Regras de negócio**	**Evidências**
OE1	Questionários por etapa, iteração, papel e domínio; UC14 e UC15.	BR01, BR04 e BR11.	Estrutura dos questionários, classificação das perguntas e execução do cenário.
OE2	Configuração de projeto, etapas, representantes, perguntas e *templates*; UC07, UC08, UC12, UC13, UC14 e UC26.	BR02, BR03, BR04 e BR09.	Projeto SelecIA RH configurado com quatro *sprints*, cinco representantes e pesos distintos; TF05 e TF06.
OE3	Cálculo de ICP, ISEQ, ISEP e IEM; UC19 e UC21.	BR04.	TF10 a TF12; ISEQs de 5,36%, 92,14%, 80,00% e 100,00%; ISEP de 75,91%.
OE4	Painéis de projeto, questionário, participante e respostas consolidadas; UC20, UC22, UC23 e UC24.	BR04, BR12 e BR13.	Painéis apresentados na Seção 6.5; TF15 e TF16.
OE5	Geração do Boletim de Não Conformidade; UC28.	BR06.	Boletim gerado para a *Sprint* 1; TF13.
OE6	Geração do Certificado de Ciência Ética e Conformidade Declarada; UC27.	BR07.	Certificado emitido após o encerramento do cenário; TF14.

Continua na próxima página...

Continuação do Quadro 15

Objetivo	**Funcionalidades e casos de uso**	**Regras de negócio**	**Evidências**
OE7	Registro de autenticidade, escopo, emissão, índice, faixa e *hash*; UC27 e UC28.	BR06 e BR07.	*Collection document_emission_records* e registros apresentados na Seção 6.6.
OE8	Notificações, reagendamento, encerramento e registros de emissão; UC17, UC18, UC19, UC27 e UC28.	BR06, BR07, BR08 e BR10.	TF09, TF13, TF14 e TF20; *collections notifications* e *document_emission_records*.
OE9	Dimensões de governança e visualizações analíticas.	BR12 e BR13.	Relação conceitual discutida nas Seções 3.4 e 6.7.
OE10	Demonstração funcional do ciclo completo.	Conjunto das regras aplicáveis ao cenário.	Protocolo da Seção 4.5, cenário detalhado no Apêndice E e casos de teste no Apêndice D.

Fonte: O autor.

APÊNDICE D – Casos de Teste e Evidências de Execução

D.1 Testes automatizados

A execução da suíte automatizada na versão 0.0.1-SNAPSHOT, em 30 de junho de 2026, resultou em 43 testes executados, sem falhas, erros ou itens ignorados. O relatório gerado pelo JaCoCo, considerando 192 classes analisadas, registrou 12% de cobertura de instruções, correspondente a 4.085 de 33.577 instruções, e 7% de cobertura de branches, correspondente a 211 de 2.826 ramificações.

Os pacotes com maior cobertura foram `<application.service.strategy.project_strategy>` (88%), `<application.usecase.notification.command>` (100%), `<domain.notification>` (78%) e `<domain.isep>` (52%). Por convenção de projeto, *controllers*, *models*, *repositories*, configurações, utilitários e classes de exceção foram excluídos do escopo de cobertura. A análise estática realizada pelo SonarQube Cloud, sobre o *commit* `4175115d`, registrou 15 novas *issues*, 0 *security hotspots* e índice de duplicação de 0,59%.

O relatório consolidado informa a quantidade total e o resultado da suíte, mas não distingue, isoladamente, o alcance dos níveis de teste (KANSTRÉN, 2008). Por esse motivo, a classificação foi realizada pela inspeção das 11 classes, das dependências substituídas e dos componentes ou recursos reais inicializados. O critério adotado classificou como unitário o caso que executa um único componente real em escopo isolado, com suas dependências substituídas, e como integração o caso que exercita a colaboração entre componentes ou recursos reais. O Quadro 16 apresenta o inventário resultante.

A inspeção identificou 43 testes unitários e nenhum teste de integração. Todas as 11 classes inicializam somente o componente sob teste e utilizam objetos simulados para as dependências. Não foram identificadas inicialização de contexto Spring, conexão com PostgreSQL ou MongoDB, abertura de servidor, uso de contêineres ou colaboração entre componentes reais.

Quadro 16 – Inventário e classificação dos testes automatizados.

Classe ou caso de teste	**Comportamento verificado**	**Classificação**	**Componentes reais inicializados**	**Dependências substituídas**	**Resultado da execução**
`AuthServiceTest` (5 casos)	Autenticação válida e inválida; cadastro válido; rejeição de e-mail existente e de termos não aceitos.	Unitário	`AuthAdapter`.	Repositório de usuários, gerenciador de autenticação e codificador de senhas.	5 de 5 passaram.
`ExportIsepCsv UseCaseTest` (4 casos)	Exportação CSV com e sem anonimização, cabeçalhos por etapa e retorno sem resultados.	Unitário	`ExportIsepCsv UseCase`.	Repositórios de projeto, questionário, representante e etapa; porta de consulta de resultados.	4 de 4 passaram.
`IterativeProject StrategyTest` (1 caso)	Criação de etapas e iterações e encaminhamento da configuração dos questionários em projeto iterativo.	Unitário	`IterativeProject Strategy`.	Portas de escrita de etapas e iterações e porta de criação de questionários.	1 de 1 passou.
`PasswordRecovery ServiceTest` (6 casos)	Solicitação, validação de código e redefinição de senha, incluindo fluxos de sucesso e falha.	Unitário	`PasswordRecovery Adapter`.	Repositórios de usuário e código de recuperação, codificador de senha e caso de uso de notificação.	6 de 6 passaram.
`ProcessExpired ProjectIsep UseCaseTest` (9 casos)	Encerramento, atraso e fechamento forçado de projetos, notificações e consolidação ponderada do ISEP.	Unitário	`ProcessExpired ProjectIsep UseCase`.	Repositório de projeto, portas de consulta e escrita dos resultados ISEQ/ISEP e caso de uso de notificação.	9 de 9 passaram.
`ProcessExpired QuestionnairesIsep UseCaseTest` (4 casos)	Processamento de questionários expirados, ausência de projeto, cálculo do ISEP e tratamento de atraso com notificação.	Unitário	`ProcessExpired QuestionnairesIsep UseCase`.	Repositórios de questionário e resposta, casos de uso de cálculo, notificação e expiração do projeto e porta de resultados.	4 de 4 passaram.

Continua na próxima página...

Continuação do Quadro 16

Classe ou caso de teste	**Comportamento verificado**	**Classificação**	**Componentes reais inicializados**	**Dependências substituídas**	**Resultado da execução**
`QuestionnaireServiceTest` (3 casos)	Vinculação de questionários à iteração ou etapa e tratamento de conjunto nulo.	Unitário	`ProjectQuestionnaireAdapter`.	Repositórios de questionário, pergunta, papel e resposta.	3 de 3 passaram.
`TemplateMongoAdapterTest` (2 casos)	Preservação de pesos, cronogramas, vínculos e representantes em modelos de projetos cascata e iterativo.	Unitário	`TemplateMongoAdapter`.	Repositórios de modelos e projetos e porta do usuário atual.	2 de 2 passaram.
`WaterfallProjectStrategyTest` (1 caso)	Criação de etapas e encaminhamento da configuração dos questionários em projeto cascata.	Unitário	`WaterfallProjectStrategy`.	Porta de escrita de etapas e porta de criação de questionários.	1 de 1 passou.
`SendInternalNotificationUseCaseTest` (2 casos)	Substituição de marcadores na notificação e bloqueio de papel não autorizado.	Unitário	`SendInternalNotificationUseCase`.	Portas de modelo de notificação e notificação interna, criadas manualmente como objetos simulados.	2 de 2 passaram.
`IseqCalculationServiceTest` (6 casos)	Cálculo de ICP e ISEQ, elegibilidade por papel, respostas não informadas e detalhamento analítico por etapa.	Unitário	`IseqCalculationService`.	Repositório de metadados das perguntas.	6 de 6 passaram.

Fonte: O autor.

D.2 Testes funcionais manuais

Esta seção apresenta os testes funcionais manuais executados sobre as principais funcionalidades da plataforma. Cada teste foi conduzido pelo próprio desenvolvedor, com base nos fluxos descritos nas narrativas de caso de uso do Apêndice A.

D.2.1 TF01 — Cadastro de usuário com e-mail duplicado (UC01)

Cenário: tentativa de cadastro com e-mail já registrado no sistema.

Pré-condição: o e-mail informado já possui conta ativa na plataforma.

Execução: o testador acessou a tela de *login* e clicou no *link* "Registre-se", conforme destacado pela seta indicativa na Figura 28. O sistema exibiu o formulário de cadastro, no qual foram preenchidos nome, sobrenome, e-mail, senha e confirmação de senha, com aceite dos termos marcado, conforme a Figura 29. Ao clicar em "Cadastrar", o sistema identificou a duplicidade do e-mail e exibiu o *modal* de erro com a mensagem "Não foi possível concluir o cadastro. Já existe uma conta associada a este e-mail", conforme a Figura 30.

Resultado obtido: sistema rejeitou o cadastro e exibiu mensagem de duplicidade. **Situação: passou.**

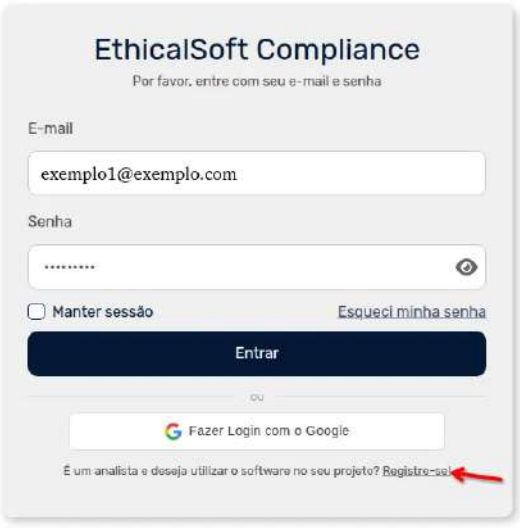

Figura 28 – Tela de *login* com destaque para o *link* de cadastro.
Fonte: O autor.

D.2.2 TF02 — *Login* com credenciais inválidas (UC02)

Cenário: tentativa de autenticação com senha incorreta para e-mail existente.

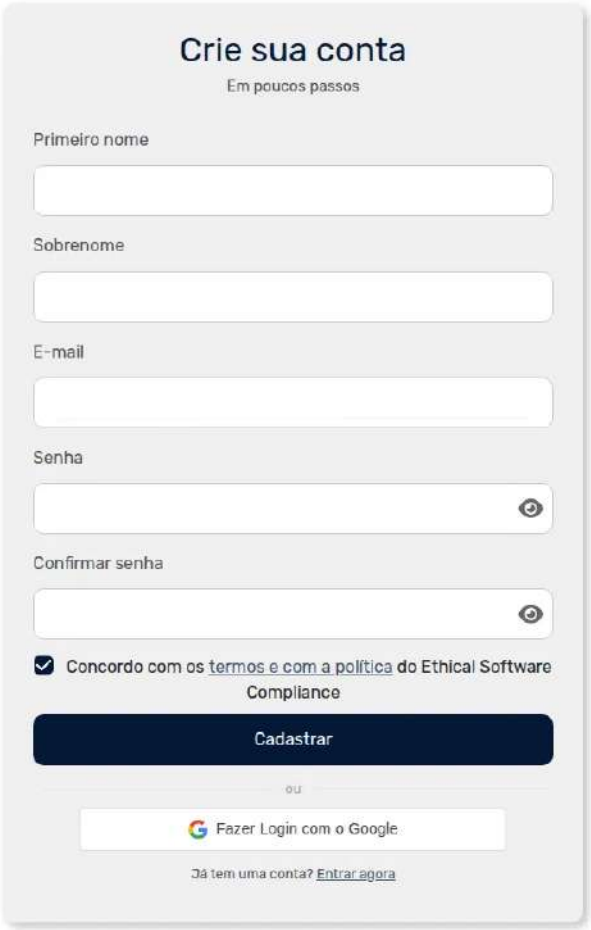

Figura 29 – Formulário de cadastro preenchido com e-mail já existente.
Fonte: O autor.

Pré-condição: o e-mail informado possui conta ativa no sistema.

Execução: o testador preencheu o campo de e-mail com um endereço válido e informou uma senha incorreta, em seguida clicou em "Entrar". O sistema processou a requisição e exibiu o *modal* de erro com a mensagem "Erro 401 — Credenciais inválidas. Verifique seu usuário e senha", conforme a Figura 31.

Resultado obtido: sistema rejeitou a autenticação e exibiu mensagem de credenciais inválidas. **Situação: passou.**

D.2.3 TF03 — *Login* via Google OAuth (UC03)

Cenário: autenticação por meio de conta Google válida vinculada à plataforma.

Pré-condição: o usuário possui conta Google elegível e está sem sessão ativa.

Execução: o testador clicou em "Fazer *Login* com o Google" na tela de *login*. O sistema acionou o provedor Google Identity Services, que abriu o seletor de contas conforme a Figura 32, exibindo as contas disponíveis no navegador. O testador selecionou a conta de um representante cadastrado no projeto SelecIA RH. Após a autenticação federada, o sistema emitiu o JWT próprio da aplicação e redirecionou o usuário autenticado para a

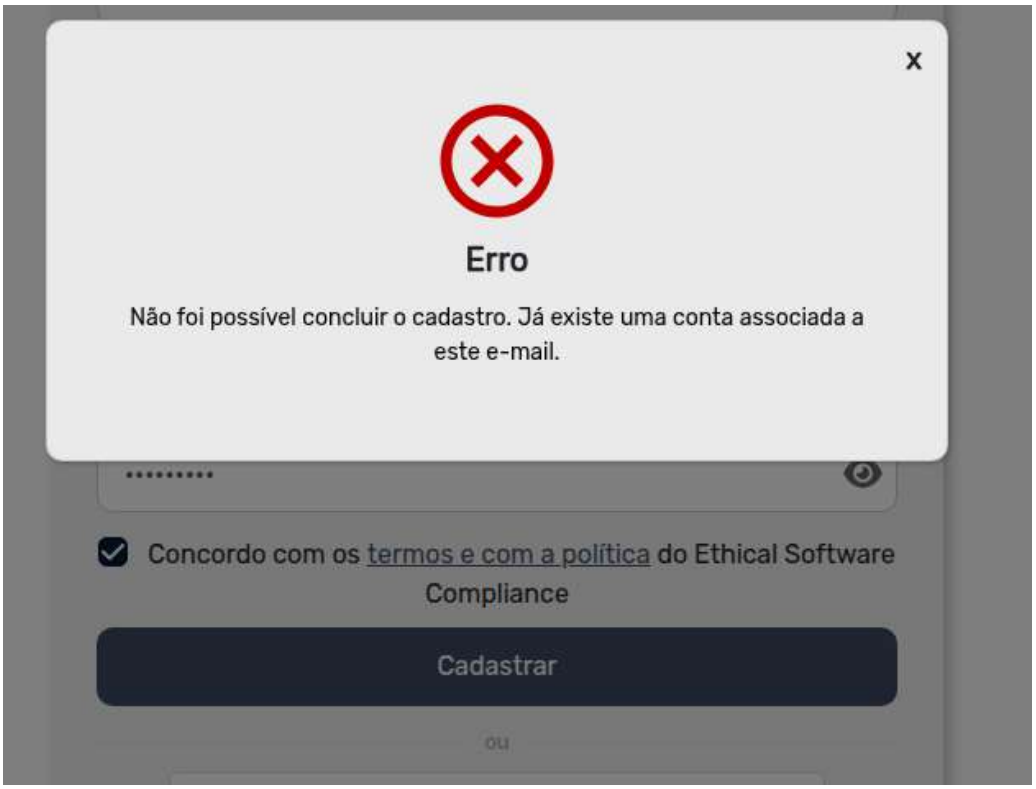

Figura 30 – *Modal* de erro exibido ao tentar cadastrar e-mail duplicado.
Fonte: O autor.

Figura 31 – *Modal* de erro exibido ao informar credenciais inválidas no *login*.
Fonte: O autor.

tela inicial, conforme a Figura 33, que exibe o painel de boas-vindas com a descrição do sistema, os indicadores ICP e ISEQ e os recursos disponíveis na plataforma.

Resultado obtido: autenticação via Google concluída e sessão iniciada corretamente. **Situação: passou.**

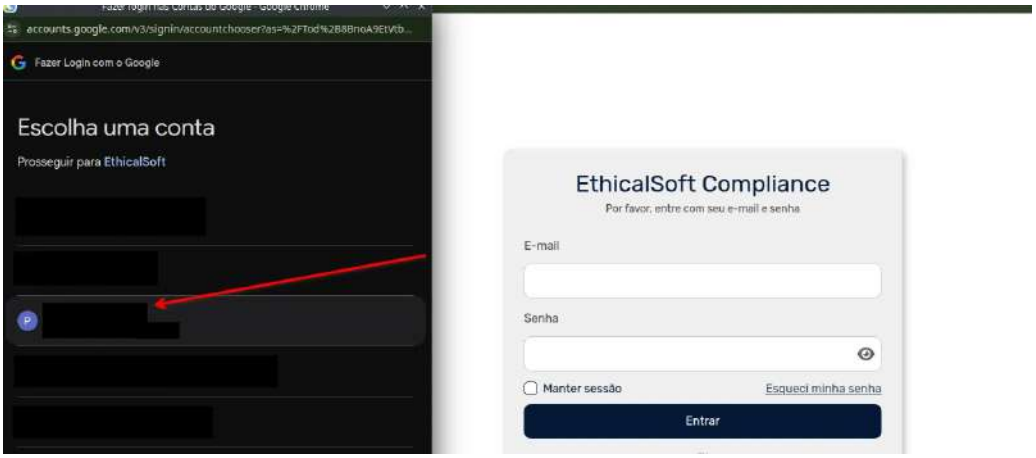

Figura 32 – Seletor de contas Google acionado pelo sistema após clique no botão de autenticação federada.

Fonte: O autor.

Figura 33 – Tela inicial exibida após autenticação bem-sucedida via Google OAuth.

Fonte: O autor.

D.2.4 TF04 — Recuperação de conta por código de e-mail (UC04)

Cenário: solicitação de redefinição de senha por meio do fluxo de recuperação de conta.

Pré-condição: o usuário possui conta ativa e não consegue acessar a plataforma com a senha atual.

Execução: o testador acessou a tela de *login* e clicou em "Esqueci minha senha", conforme destacado pela seta indicativa na Figura 34. O sistema exibiu a tela de recuperação de conta, na qual o testador informou o e-mail e clicou em "Enviar e-mail de recuperação de conta", conforme a Figura 35. A interface também informou que o código seria válido por até 10 minutos. O sistema enviou o e-mail de recuperação para a caixa de entrada do usuário, contendo o código e a indicação de validade de 10 minutos, conforme a Figura 36. De posse do código, o testador foi redirecionado à tela de redefinição de senha, na qual informou a nova senha e sua confirmação, conforme a Figura 37. Após a confirmação, o sistema redefiniu a senha e redirecionou o usuário para a tela de *login*. O navegador ofereceu a opção de salvar as novas credenciais no gerenciador de senhas do Google, conforme visível na Figura 38, indicando que o *login* com as novas credenciais foi reconhecido com sucesso.

Resultado obtido: fluxo completo de recuperação de conta executado em três etapas - solicitação, verificação de código e redefinição - sem erros. **Situação: passou.**

D.2.5 TF05 — Criação de projeto iterativo com *template* (UC07, UC26)

Cenário: criação de projeto do tipo iterativo incremental utilizando o *template* Base Ágil Iterativo como ponto de partida.

Pré-condição: o usuário está autenticado com perfil de administrador e há pelo menos um *template* público disponível na plataforma.

Execução: o testador acessou a tela de listagem de projetos e clicou em "Criar novo projeto", conforme destacado pela seta indicativa na Figura 39. O sistema exibiu o *modal* de seleção de tipo de projeto, no qual o testador selecionou a opção "Iterativo Incremental" e clicou em "Continuar", conforme a Figura 40.

O sistema redirecionou para o formulário de criação, no qual o testador selecionou o *template* "Base Ágil Iterativo", informou o nome "National Geograph", confirmou o tipo iterativo, definiu a data de início como 29/06/2026, o prazo limite como 02/09/2026 e a duração das iterações como 10 dias úteis. O sistema calculou automaticamente o número de iterações como 4, conforme a Figura 41.

Na seção de etapas, o *template* pré-populou a estrutura com cinco etapas - Iniciação, Requisitos, Projeto, Desenvolvimento e Testes - com pesos configurados em 2, 3, 2, 2,5 e 1,5 respectivamente, conforme a Figura 42. Na seção de representantes, o *template* pré-

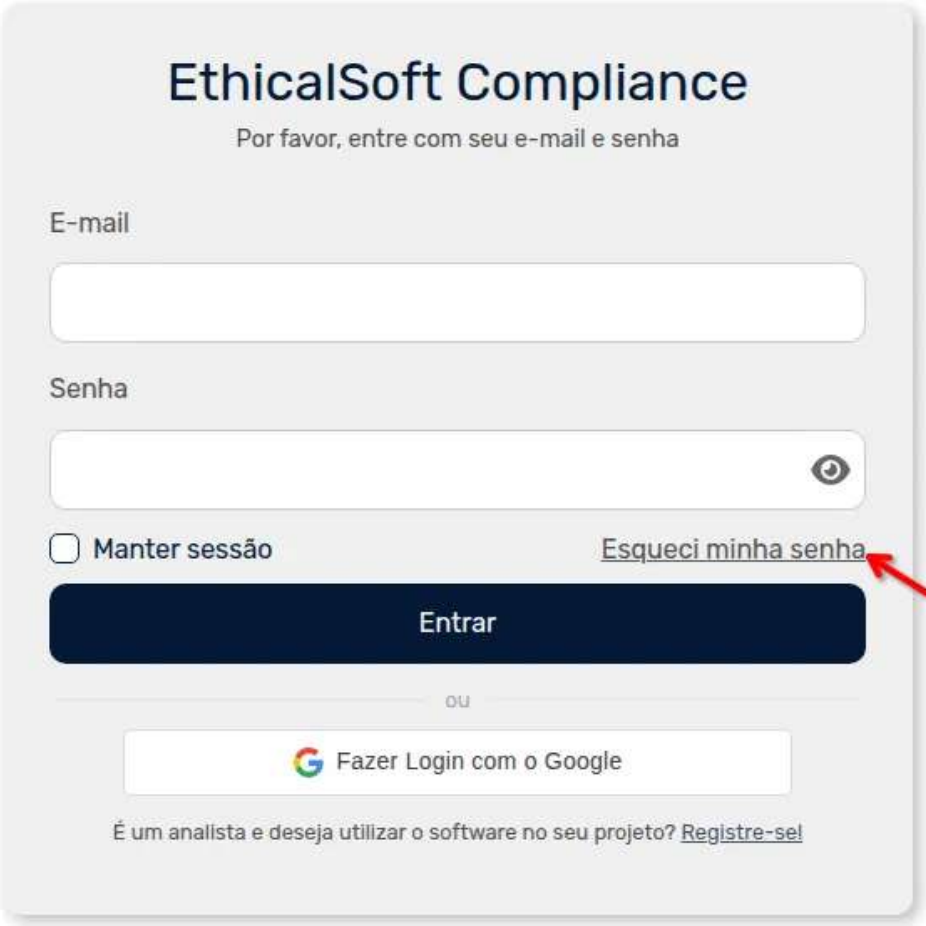

Figura 34 – Tela de *login* com destaque para o *link* de recuperação de senha.
Fonte: O autor.

Figura 35 – Tela de recuperação de conta com campo de e-mail preenchido.
Fonte: O autor.

Figura 36 – E-mail de recuperação recebido com código de verificação.
Fonte: O autor.

Figura 37 – Tela de registro de nova senha após validação do código.
Fonte: O autor.

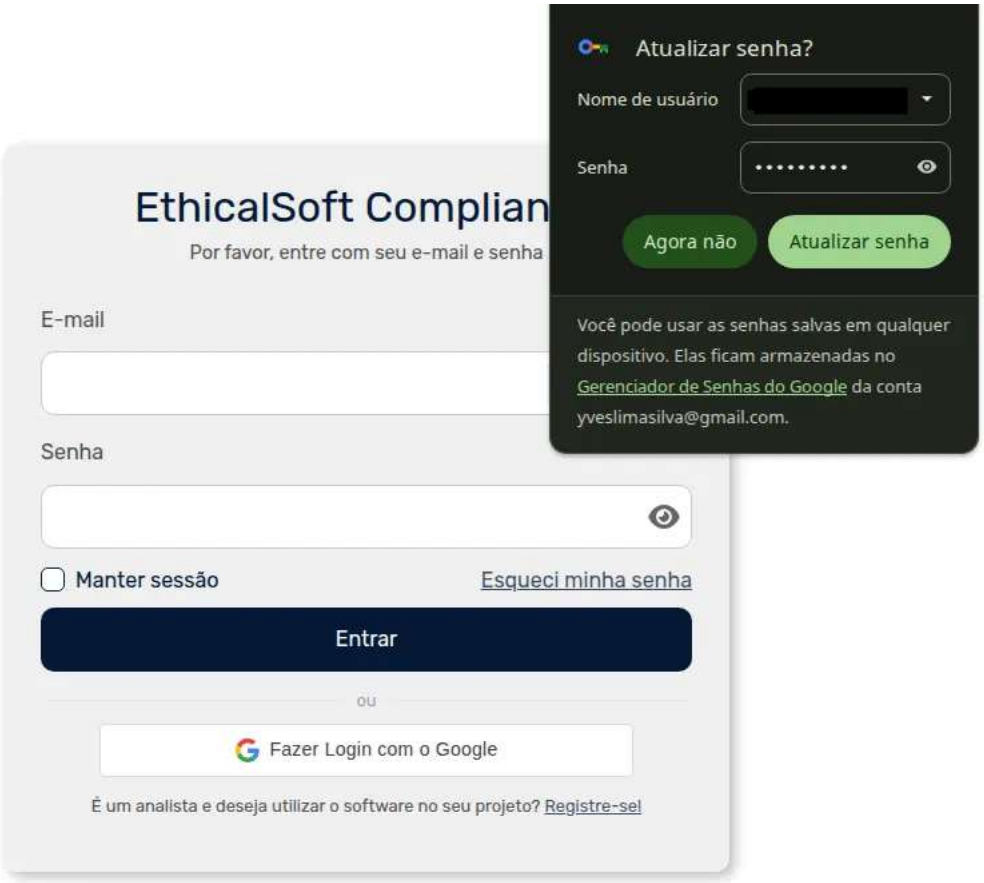

Figura 38 – Tela de *login* com *prompt* do navegador para atualização de senha, confirmando o sucesso da redefinição.
Fonte: O autor.

populou os cinco participantes do projeto com seus respectivos papéis e pesos, conforme a Figura 43.

Na seção de questionários, o sistema exibiu os quatro questionários gerados automaticamente — *Sprint* 1 a *Sprint* 4 - com pesos 1, 1, 1,5 e 2 e as faixas de aplicação calculadas automaticamente com base nas datas e na duração configuradas, conforme a Figura 44. O testador acessou a edição do questionário *Sprint* 1 para verificar as perguntas vinculadas pelo *template*, confirmando a presença de 18 perguntas com as classificações de escopo e os papéis corretamente atribuídos, conforme a Figura 45.

Ao clicar em "Salvar projeto", o sistema exibiu o *modal* de confirmação informando que um *template* seria gerado e que os representantes receberiam e-mail de acesso, conforme a Figura 46. Após a confirmação, o projeto "National Geograph" passou a constar na listagem com *status* "Aberto" e situação "*Sprint* 1/4", conforme a Figura 47, confirmando a criação bem-sucedida.

O sistema enviou automaticamente dois tipos de notificação por e-mail: o primeiro informando a cada representante sua vinculação ao projeto, com *link* direto, período previsto, papéis atribuídos e data do próximo questionário, conforme a Figura 48; e o segundo

notificando a disponibilidade do questionário *Sprint* 1 para resposta, com o período de aplicação e o *link* de acesso, conforme a Figura 49.

Resultado obtido: projeto iterativo criado com estrutura pré-configurada a partir do *template*, quatro questionários gerados com datas calculadas automaticamente, *template* derivado gerado e notificações enviadas aos representantes. **Situação: passou.**

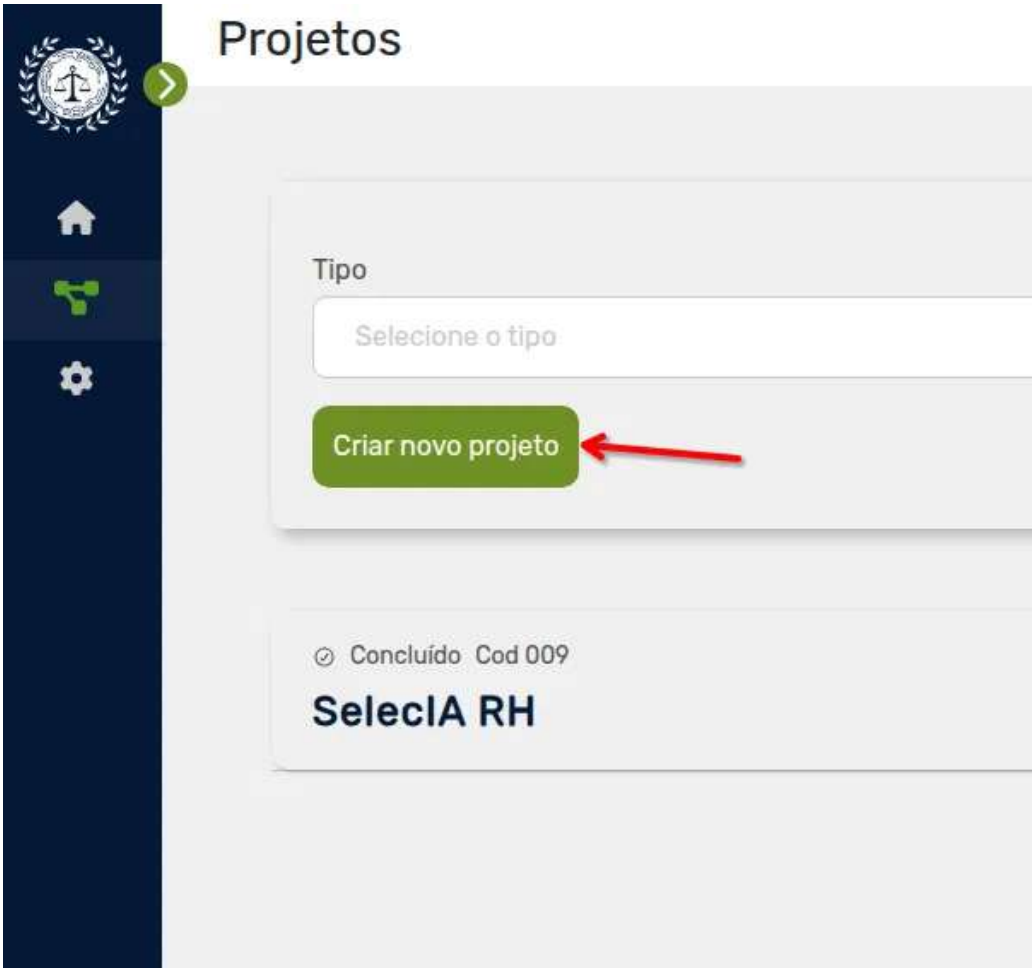

Figura 39 – Tela de listagem de projetos com destaque para o botão de criação.
Fonte: O autor.

D.2.6 TF06 — Criação de projeto cascata com *template* (UC07, UC26)

Cenário: criação de projeto do tipo cascata utilizando o *template* Base Cascata como ponto de partida, com configuração manual das etapas, representantes e questionários.

Pré-condição: o usuário está autenticado com perfil de administrador e há pelo menos um *template* público do tipo cascata disponível na plataforma.

Execução: o testador acessou o formulário de criação e selecionou a opção “Cascata” no *modal* de seleção do tipo de projeto, conforme a Figura 50. O sistema exibiu o formulário de criação, no qual o testador selecionou o *template* “Base Cascata”, informou o nome

Figura 40 – *Modal* de seleção do tipo de projeto com opção Iterativo Incremental selecionada.

Fonte: O autor.

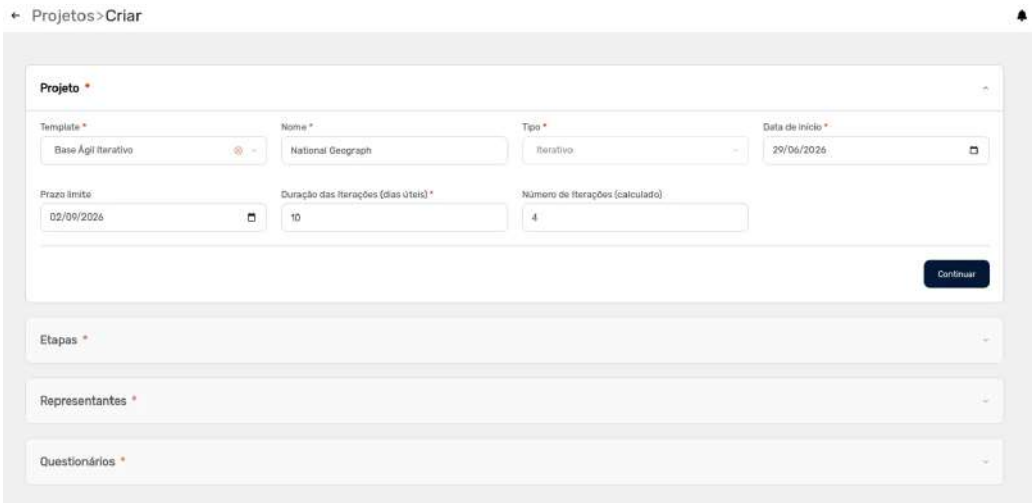

Figura 41 – Formulário de criação com *template* selecionado e número de iterações calculado automaticamente.

Fonte: O autor.

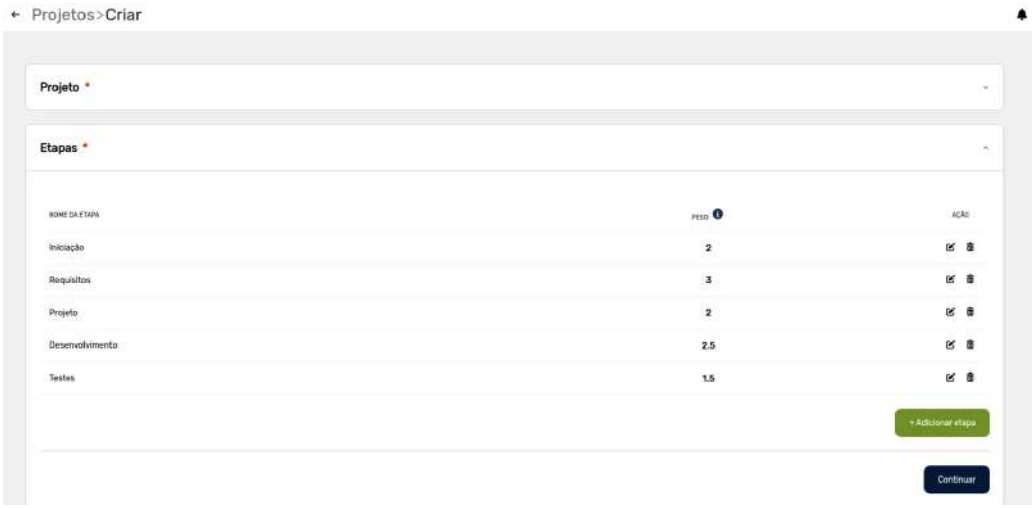

Figura 42 – Seção de etapas pré-configuradas pelo *template* com pesos atribuídos.
Fonte: O autor.

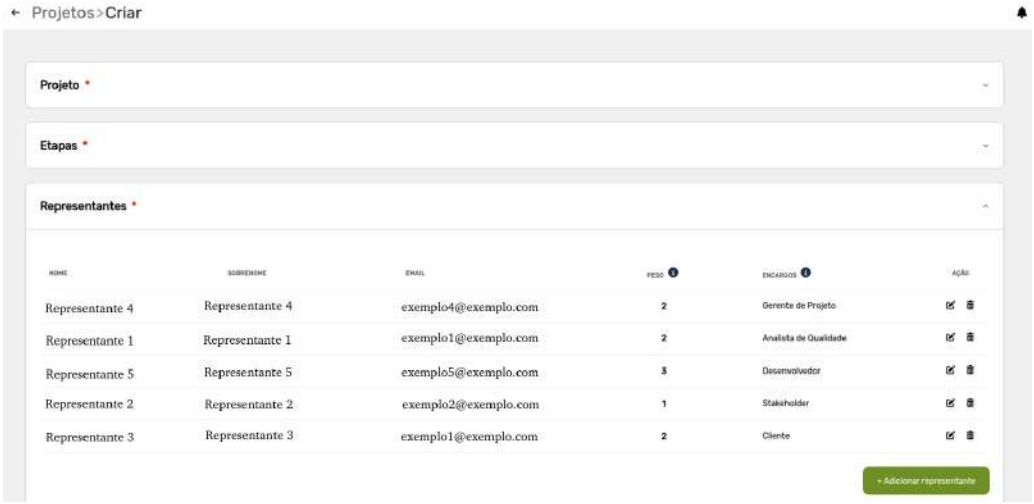

Figura 43 – Seção de representantes pré-configurados pelo *template* com papéis e pesos.
Fonte: O autor.

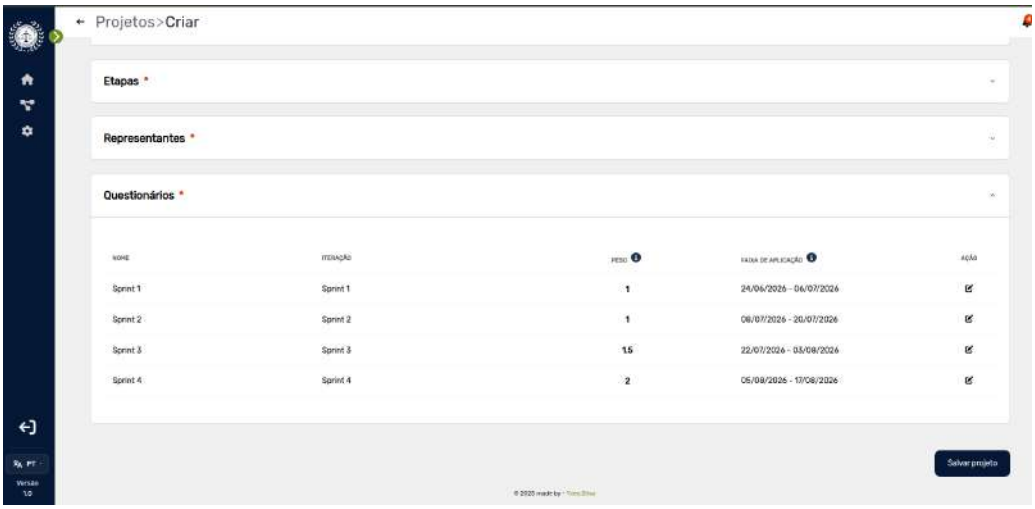

Figura 44 – Seção de questionários com quatro *sprints* geradas e faixas de aplicação calculadas.

Fonte: O autor.

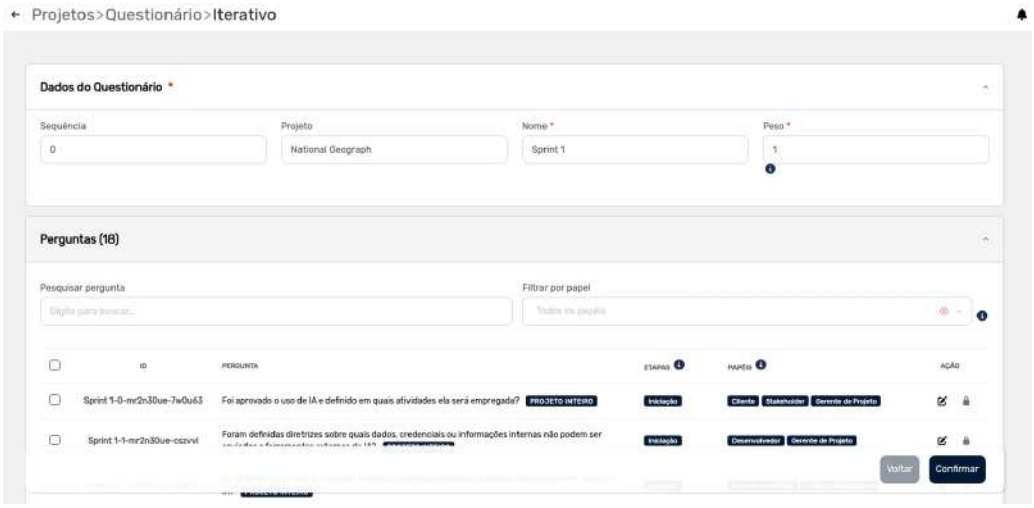

Figura 45 – Edição do questionário *Sprint* 1 exibindo as 18 perguntas vinculadas pelo *template* com classificações e papéis.

Fonte: O autor.

Figura 46 – *Modal* de confirmação de criação do projeto informando geração de *template* e envio de e-mails.

Fonte: O autor.

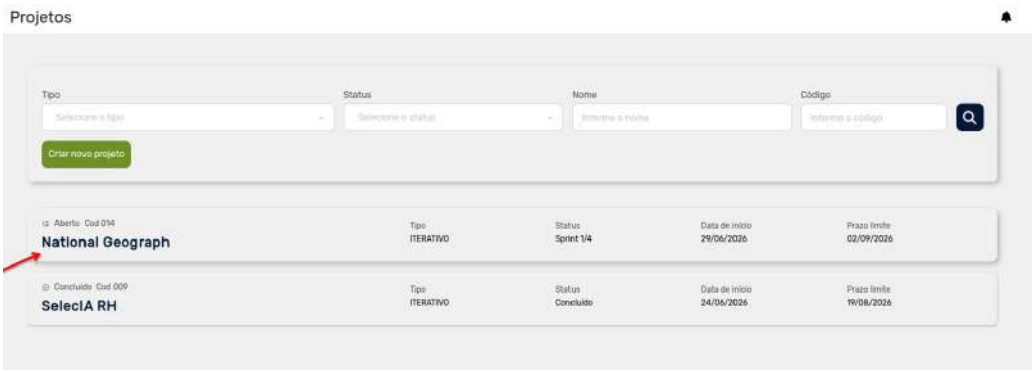

Figura 47 – Listagem de projetos com o projeto National Geograph recém-criado em *status* Aberto.

Fonte: O autor.

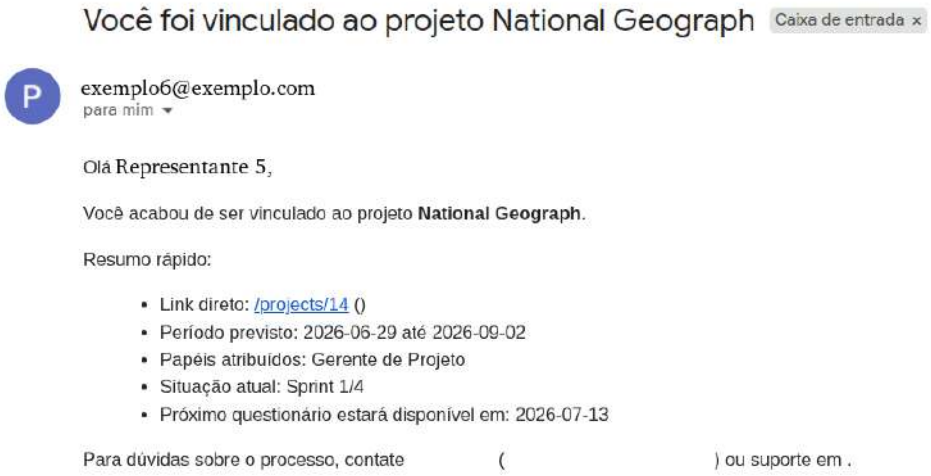

Figura 48 – E-mail de vinculação ao projeto recebido pelo representante com *link*, papéis e situação atual.

Fonte: O autor.

Figura 49 – E-mail de notificação do questionário *Sprint* 1 disponível para resposta.

Fonte: O autor.

"Extensão PopTop", confirmou o tipo cascata, definiu a data de início como 29/06/2026 e o prazo limite como 01/08/2026, conforme a Figura 51.

Na seção de etapas, o *template* pré-populou a estrutura com três etapas — Iniciação, Requisitos e Desenvolvimento — com as respectivas faixas de aplicação calculadas automaticamente pelo sistema com base na data de início e nos períodos configurados no *template*, e com pesos 2, 3 e 2,5, conforme a Figura 52. O testador editou a composição dos representantes diretamente na seção correspondente, definindo três participantes com múltiplos encargos cada: Usuário 1 com os papéis de Analista de Qualidade, Analista de Requisitos e *Designer*; Usuário 2 com os papéis de Desenvolvedor, Responsável do Negócio, Líder de Equipe e Arquiteto de *Software*; e Usuário 3 com os papéis de Cliente, *Stakeholder* e Gerente de Projeto, conforme a Figura 53.

Na seção de questionários, o sistema gerou automaticamente três questionários vinculados às etapas configuradas — Iniciação, Requisitos e Desenvolvimento — com as datas de início e término calculadas sequencialmente com base no prazo do projeto, conforme a Figura 54. Ao clicar em "Salvar projeto", o sistema registrou o projeto e o exibiu na listagem com *status* "Aberto", tipo "CASCATA" e situação "Iniciação", identificado como Cod 015, conforme a Figura 55. O sistema também gerou notificações internas para os representantes vinculados, informando a vinculação ao projeto e a disponibilidade do primeiro questionário, conforme a Figura 56.

Resultado obtido: projeto cascata criado com *template*, etapas e questionários vinculados com datas calculadas automaticamente e representantes configurados com múltiplos encargos. **Situação: passou.**

Figura 50 – *Modal* de seleção do tipo de projeto com opção Cascata selecionada.
Fonte: O autor.

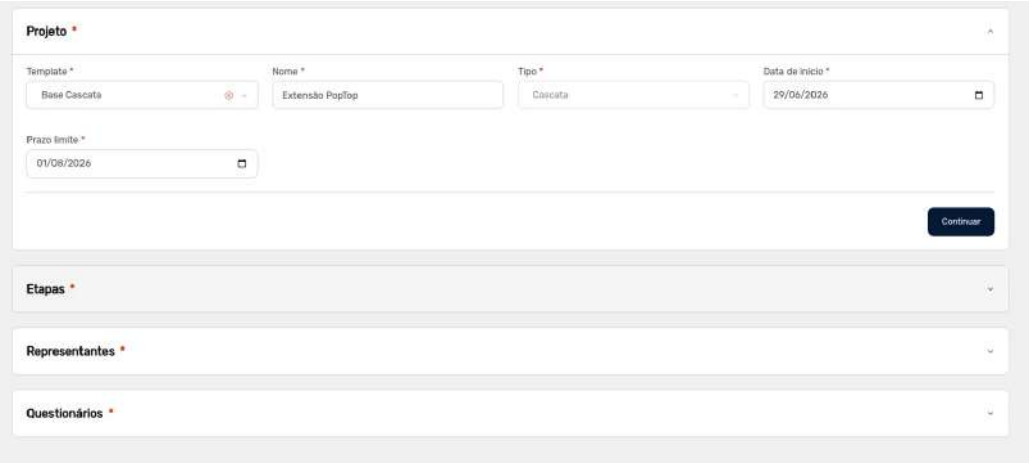

Figura 51 – Formulário de criação com *template* Base Cascata selecionado e dados do projeto preenchidos.

Fonte: O autor.

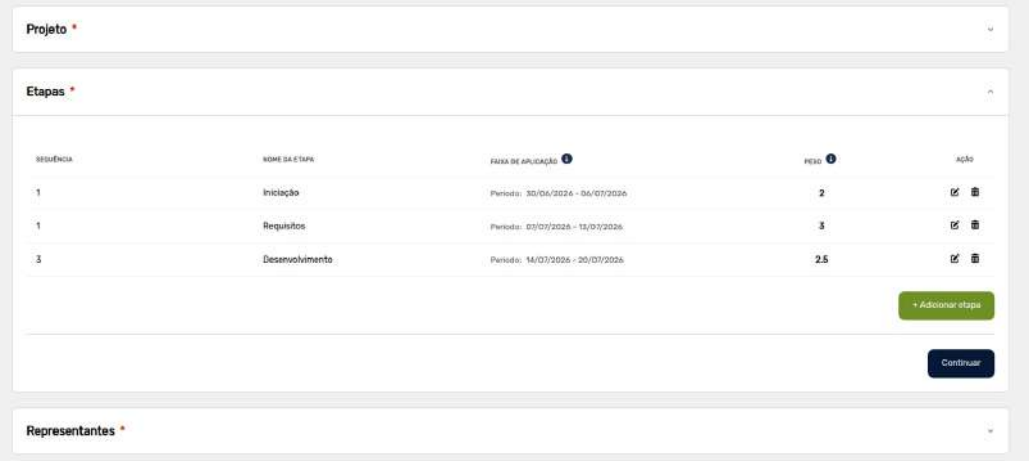

Figura 52 – Seção de etapas com três fases pré-configuradas e faixas de aplicação calculadas automaticamente.

Fonte: O autor.

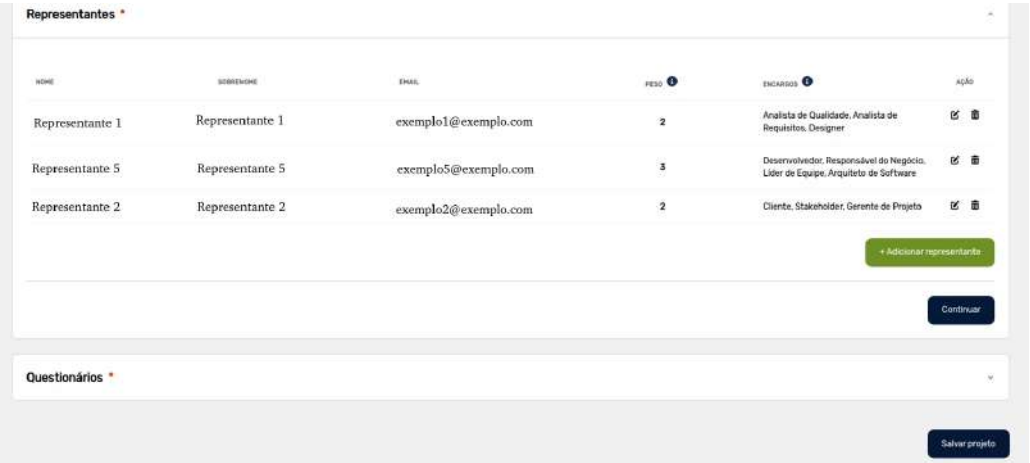

Figura 53 – Seção de representantes com três participantes configurados e múltiplos encargos por participante.

Fonte: O autor.

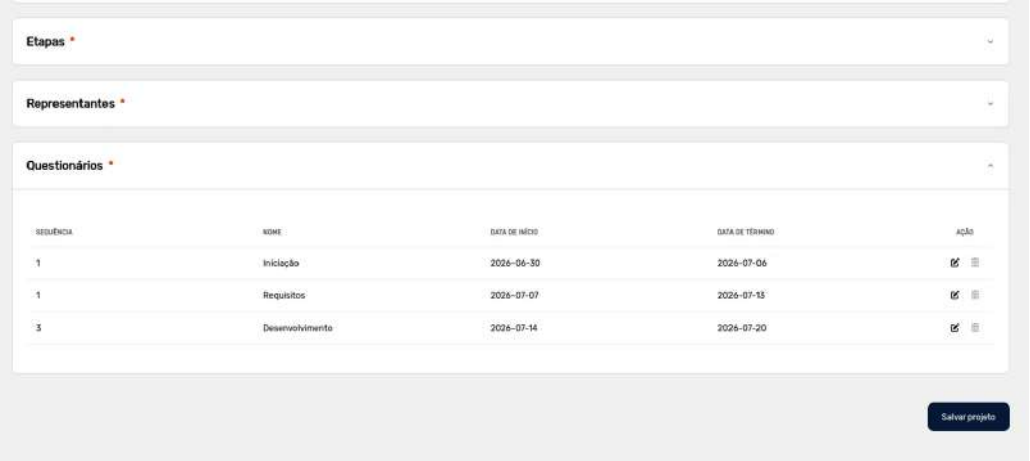

Figura 54 – Seção de questionários com três itens vinculados às etapas e datas sequenciais calculadas.

Fonte: O autor.

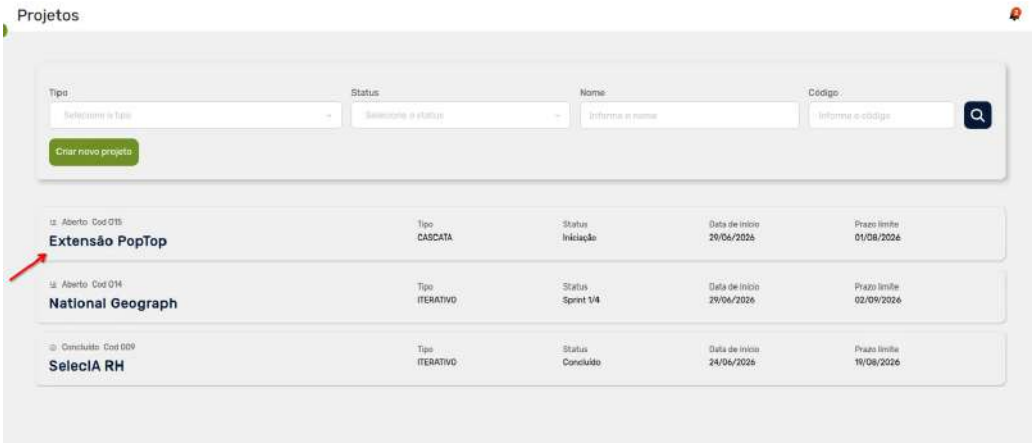

Figura 55 – Listagem de projetos com o projeto Extensão PopTop recém-criado em *status* Aberto do tipo Cascata.

Fonte: O autor.

D.2.7 TF07 — Resposta ao questionário com campo obrigatório ausente (UC15)

Cenário: tentativa de envio de questionário sem preencher o campo de evidência obrigatório para resposta afirmativa.

Pré-condição: o usuário está autenticado como representante, o questionário *Sprint* 1 do projeto 14 está em andamento e o participante possui *status* “Pendente”.

Execução: o testador acessou os detalhes do projeto 14 e verificou o estado do questionário *Sprint* 1, com 0 de 5 respondentes e todos os representantes com *status* “Pendente”, conforme a Figura 57. Ao clicar em “Responder agora”, o sistema carregou o formulário de resposta com 8 perguntas elegíveis para o papel do testador, estimativa de 24 minutos e prazo até 09/07/2026. O testador respondeu as perguntas com “Sim” e tentou enviar sem preencher o campo de evidência obrigatório associado à resposta afirmativa, conforme indicado pelas setas na Figura 58 — o botão “Enviar respostas” permaneceu desabilitado

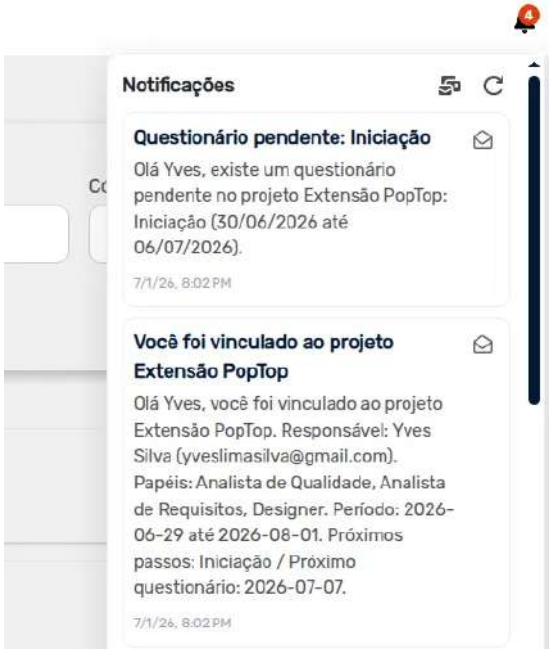

Figura 56 – Painel de notificações internas exibindo vinculação ao projeto e questionário pendente.

Fonte: O autor.

enquanto o campo de evidência da pergunta 2 permanecia em branco. Após preencher as observações e evidências em todas as perguntas, o botão foi habilitado, conforme a Figura 59, e ao clicar em "Enviar respostas" o sistema exibiu o *modal* de confirmação "Respostas enviadas com sucesso", conforme a Figura 60.

Resultado obtido: sistema bloqueou o envio enquanto havia campo obrigatório de evidência não preenchido, liberando o botão somente após o preenchimento completo. **Situação: passou.**

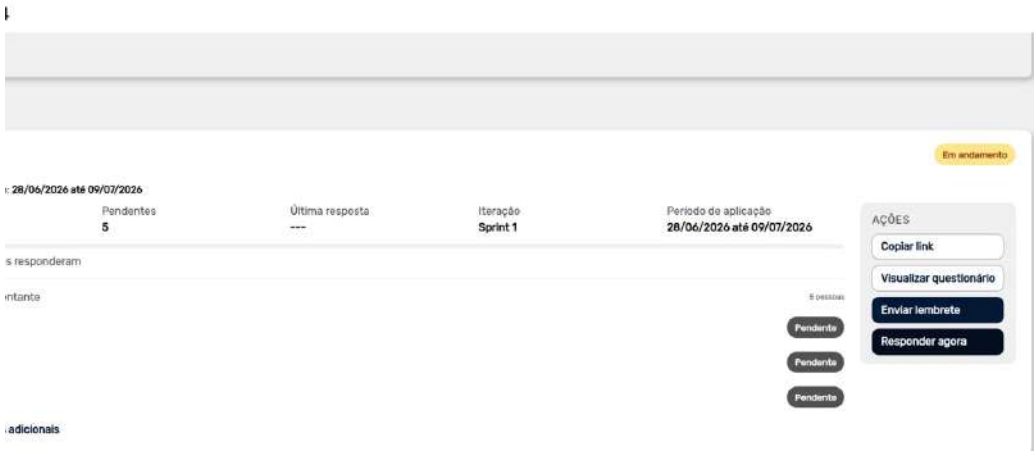

Figura 57 – Tela de detalhes do projeto 14 com o questionário *Sprint* 1 em andamento e todos os representantes com *status* Pendente.

Fonte: O autor.

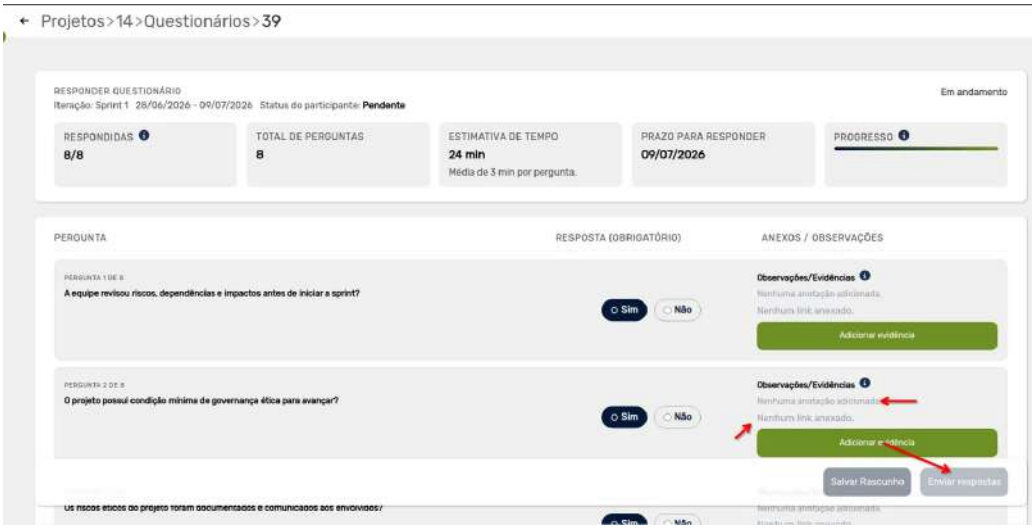

Figura 58 – Tela de resposta ao questionário com botão “Enviar respostas” desabilitado e campo de evidência obrigatório em branco.
Fonte: O autor.

Figura 59 – Tela de resposta com todas as perguntas respondidas e campos de evidência preenchidos, botão de envio habilitado.
Fonte: O autor.

Figura 60 – *Modal* de confirmação exibido após envio bem-sucedido das respostas.
Fonte: O autor.

D.2.8 TF08 — Salvamento de rascunho durante preenchimento (UC15)

Cenário: preenchimento parcial do questionário com salvamento de rascunho e retomada da sessão por outro representante.

Pré-condição: o questionário *Sprint* 1 do projeto 14 está em andamento e o representante possui *status* "Pendente".

Execução: o testador acessou o questionário com 4 perguntas elegíveis para o papel de Desenvolvedor. Após responder 2 das 4 perguntas — registrando "Não" para a pergunta 1 com campo de justificativa habilitado, e "Sim" para a pergunta 2 com campo de evidências habilitado — o testador clicou em "Salvar Rascunho", conforme indicado pela seta na Figura 61. O sistema exibiu o *modal* de confirmação "Rascunho salvo com sucesso", conforme a Figura 62, mantendo o questionário com o estado parcialmente preenchido.

Para verificar a persistência do rascunho entre sessões, o testador encerrou a sessão e realizou novo *login* com as credenciais do representante, conforme a Figura 63. Ao acessar novamente o questionário *Sprint* 1, o sistema carregou o estado anterior com *status* do participante "Em andamento" e progresso de 2/4 respondidas, conforme a Figura 64, confirmando que o rascunho foi persistido corretamente no servidor e recuperado na nova sessão.

Resultado obtido: rascunho salvo com sucesso, estado parcial persistido no servidor e recuperado corretamente na retomada da sessão com o mesmo representante. **Situação: passou.**

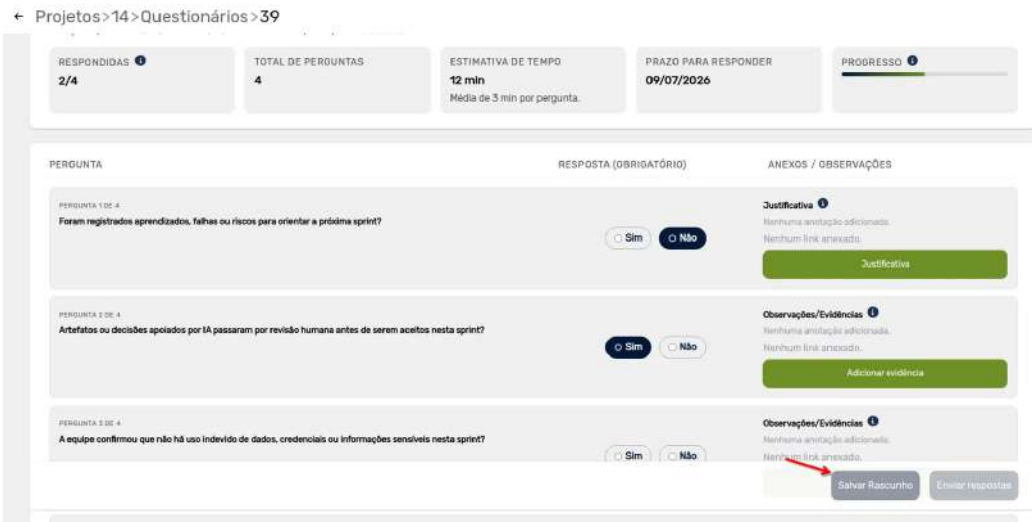

Figura 61 – Tela de resposta ao questionário com duas perguntas respondidas e destaque para o botão "Salvar Rascunho".

Fonte: O autor.

Figura 62 – *Modal* de confirmação exibido após salvamento do rascunho.
Fonte: O autor.

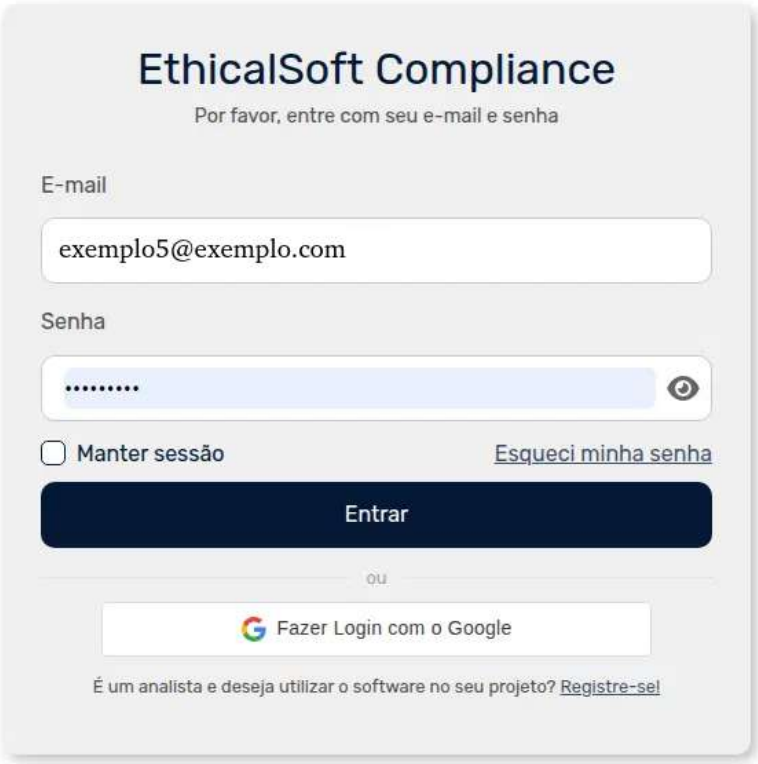

Figura 63 – Tela de *login* com as credenciais do representante para retomada da sessão.
Fonte: O autor.

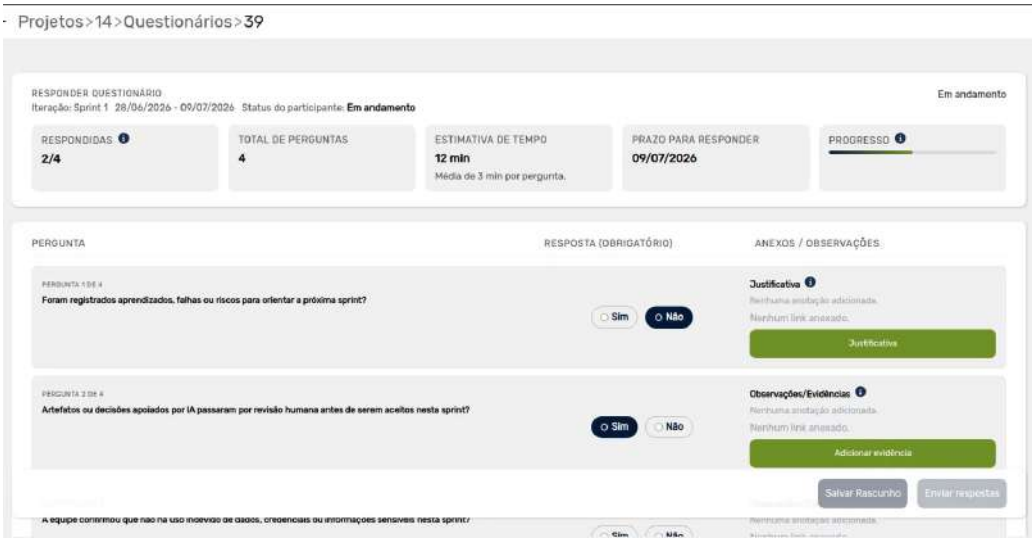

Figura 64 – Questionário retomado com *status* "Em andamento" e progresso de 2/4 respostas preservado.

Fonte: O autor.

D.2.9 TF09 — Encerramento manual de questionário (UC19)

Cenário: encerramento forçado de questionário com todos os representantes já tendo respondido, via ação administrativa.

Pré-condição: o usuário está autenticado como administrador, o questionário *Sprint* 1 do projeto 16 está em andamento e todos os 3 representantes elegíveis já realizaram a submissão.

Execução: o testador acessou os detalhes do projeto 16 e verificou que o questionário *Sprint* 1 apresentava 3 de 3 respondentes com *status* "Respondido" e 0 pendentes, barra de progresso completa e data da última resposta registrada em 01/07/2026 às 20:44, conforme a Figura 65. O testador clicou em "Forçar Encerramento", conforme indicado pela seta na mesma figura. O sistema exibiu o *modal* de confirmação perguntando se o testador desejava encerrar o questionário "*Sprint* 1" e calcular o ISEQ com as respostas existentes, conforme a Figura 66. Após confirmar, o sistema exibiu o *modal* de sucesso com a mensagem "Questionário '*Sprint* 1' encerrado. O ISEQ foi calculado", conforme a Figura 67.

Resultado obtido: questionário encerrado manualmente com sucesso e ISEQ calculado a partir das respostas disponíveis. **Situação: passou.**

D.2.10 TF10 — Cálculo do ISEQ da *Sprint* 1 (UC19, BR04)

Cenário: verificação do cálculo do ISEQ após o encerramento do questionário *Sprint* 1 com maioria das respostas negativas.

Pré-condição: o questionário *Sprint* 1 do projeto 16 foi encerrado com 3 de 3 representantes respondidos.

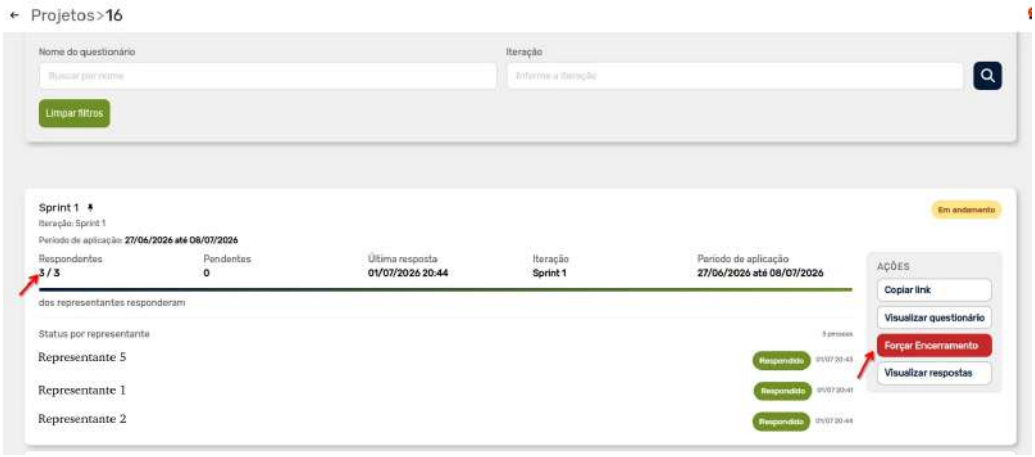

Figura 65 – Tela de detalhes do projeto 16 com *Sprint* 1 com 3/3 respondentes e destaque para o botão de encerramento forçado.

Fonte: O autor.

Figura 66 – *Modal* de confirmação de encerramento do questionário com aviso de cálculo do ISEQ.

Fonte: O autor.

Execução: o testador acessou o *dashboard* do questionário *Sprint* 1 e verificou o ISEQ calculado pelo sistema. O painel exibiu ISEQ de 68,37%, classificado na Faixa C, com média simples da equipe de 64,29% e desvio padrão de 45,55%, indicado como "Alta divergência de percepção". O gráfico de distribuição por faixa mostrou 2 membros na Faixa A e 1 membro na Faixa E, evidenciando a heterogeneidade das respostas, conforme a Figura 68.

Resultado obtido: ISEQ calculado corretamente após o encerramento, com distribuição por faixa e desvio padrão exibidos no painel. **Situação: passou.**

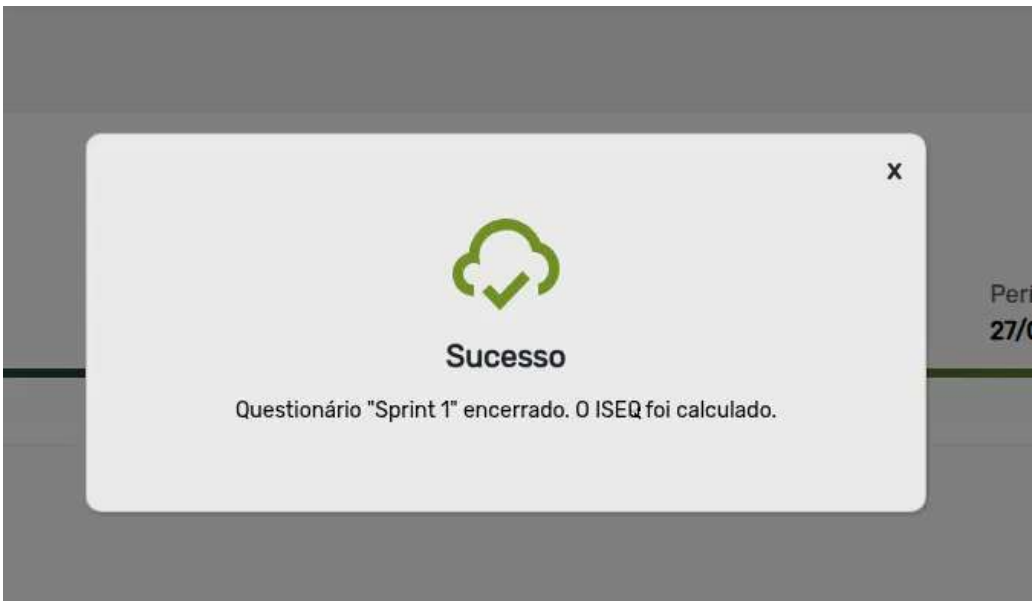

Figura 67 – *Modal* de sucesso confirmando o encerramento do questionário e o cálculo do ISEQ.

Fonte: O autor.

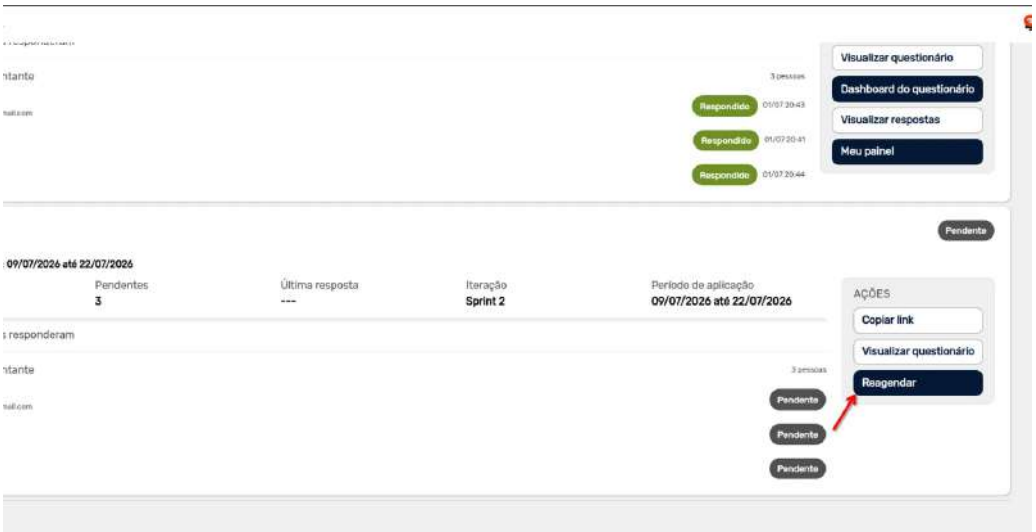

Figura 68 – *Dashboard* do questionário *Sprint* 1 com ISEQ de 68,37%, distribuição por faixa e métricas da equipe.

Fonte: O autor.

D.2.11 TF11 — Cálculo do ISEQ da *Sprint* 2 (UC19, BR04)

Cenário: verificação do cálculo do ISEQ após o encerramento do questionário *Sprint* 2 com maioria das respostas afirmativas.

Pré-condição: o questionário *Sprint* 2 do projeto 16 foi encerrado com todos os representantes elegíveis tendo respondido.

Execução: o testador acessou o *dashboard* do questionário *Sprint* 2 e verificou o ISEQ calculado pelo sistema. O painel exibiu ISEQ de 92,86%, classificado na Faixa A, com média simples da equipe de 91,67% e desvio padrão de 11,79%, indicado como "Consenso adequado", conforme a Figura 69.

Resultado obtido: ISEQ calculado corretamente com classificação na Faixa A, desvio padrão baixo e métricas consolidadas exibidas no painel. **Situação: passou.**

Figura 69 – *Dashboard* do questionário *Sprint* 2 com ISEQ de 92,86%, Faixa A e desvio padrão de 11,79%.

Fonte: O autor.

D.2.12 TF12 — Cálculo do ISEP consolidado (UC21, BR04)

Cenário: verificação do cálculo do ISEP consolidado do projeto após o encerramento de todos os questionários.

Pré-condição: os dois questionários do projeto 16 foram concluídos e o projeto está apto ao encerramento.

Execução: o testador acessou o *dashboard* consolidado do projeto 16 — denominado "Teste" — e verificou o ISEP calculado. O painel exibiu ISEP consolidado de 80,62%, classificado na Faixa B, com 100% de questionários concluídos (2 de 2), média simples da equipe de 80,62% e desvio padrão de 12,25%, indicado como "Consistência aceitável". O gráfico de evolução por iteração mostrou a progressão do ISEQ entre *Sprint* 1 (68,37%) e *Sprint* 2 (92,86%), conforme a Figura 70.

Resultado obtido: ISEP consolidado calculado corretamente com evolução histórica por iteração e métricas de consistência exibidas no *dashboard*. **Situação: passou.**

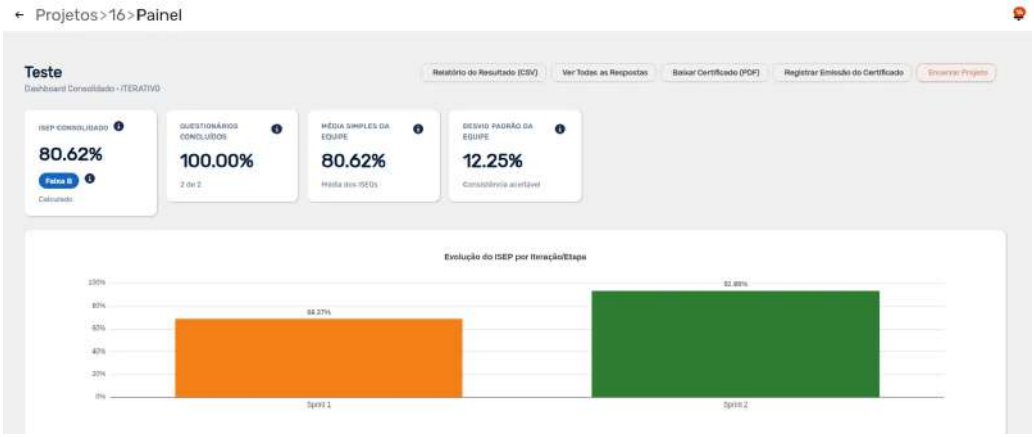

Figura 70 – *Dashboard* consolidado do projeto com ISEP de 80,62%, Faixa B e evolução por *sprint*.

Fonte: O autor.

D.2.13 TF13 — Geração do boletim de não conformidade (UC28, BR06)

Cenário: verificação da geração e envio automático do Boletim de Não Conformidade Ética quando o ISEQ de um questionário fica abaixo do limiar configurado.

Pré-condição: o questionário *Sprint* 1 do projeto National Geograph foi encerrado com ISEQ de 29,49%, abaixo do limiar mínimo de 75% (Faixa B).

Execução: após o encerramento do questionário, o sistema gerou automaticamente o Boletim de Não Conformidade Ética e o enviou por e-mail ao representante. O e-mail recebido exibiu o título "Boletim de Não Conformidade Ética: *Sprint* 1", o ISEP do escopo de 29,49%, a faixa obtida E e a faixa mínima exigida B, com informação de que o documento integral em PDF estava anexado e recomendação de revisão das respostas não conformes antes do encerramento da etapa, conforme a Figura 71.

Resultado obtido: boletim gerado automaticamente e enviado por e-mail com o PDF anexado e as informações do questionário e dos índices obtidos. **Situação: passou.**

D.2.14 TF14 — Emissão do Certificado de Ciência Ética e Conformidade Declarada (UC27, BR07)

Cenário: emissão do Certificado de Ciência Ética e Conformidade Declarada após o projeto atingir o ISEP consolidado acima do limiar configurado.

Pré-condição: o projeto SelecIA RH foi encerrado com ISEP consolidado de 75,91%, acima do limiar de 75% (Faixa B).

Execução: o testador acessou o *dashboard* consolidado do projeto e clicou em "Baixar Certificado (PDF)". O sistema iniciou o *download* do arquivo certificado-9.pdf, conforme a Figura 72. O certificado gerado exibiu o título "Certificado de Ciência Ética e Conformidade Declarada", o nome do projeto SelecIA RH (PRJ-9), o ISEP obtido de 75,91%,

Figura 71 – E-mail de boletim de não conformidade recebido pelo representante com ISEQ, faixa obtida e faixa exigida.

Fonte: O autor.

a faixa B, a faixa mínima exigida B, o desvio padrão da equipe de 37,64%, a relação das quatro *sprints* com seus respectivos ISEQ e faixas — *Sprint* 1 em 5,36% (E), *Sprint* 2 em 92,14% (A), *Sprint* 3 em 80,00% (B) e *Sprint* 4 em 100,00% (A) — e o código de autenticidade, conforme a Figura 73.

Resultado obtido: certificado emitido em PDF com os indicadores consolidados, a relação das *sprints* e o código de autenticidade. **Situação: passou.**

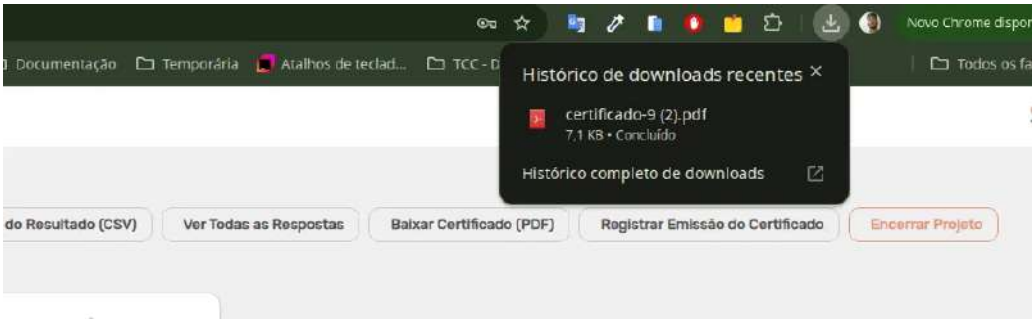

Figura 72 – Histórico de *downloads* do navegador exibindo o arquivo PDF do certificado em *download*.

Fonte: O autor.

D.2.15 TF15 — *Dashboard* do questionário com *heatmap* e nuvem de palavras (UC22)

Cenário: verificação dos componentes analíticos do painel de questionário após o encerramento de uma *sprint*.

Pré-condição: o questionário 20 do projeto 9 foi encerrado e possui ISEQ calculado.

Figura 73 – Certificado de Ciência Ética e Conformidade Declarada emitido para o projeto SelecIA RH com ISEP 75,91% e código.
Fonte: O autor.

Execução: o testador acessou o *dashboard* do questionário 20 e verificou os componentes exibidos. O gráfico de distribuição por faixa apresentou 3 membros na Faixa A, 1 na Faixa B e 1 na Faixa D. O *heatmap* de risco por encargo e etapa cruzou os cinco papéis — Desenvolvedor, Gerente de Projeto, Analista de Qualidade, *Stakeholder* e Cliente — com as quatro etapas — Iniciação, Requisitos, Projeto e Desenvolvimento — exibindo percentuais de conformidade em escala de cor contínua do vermelho (0%) ao verde (100%), conforme a Figura 74. O painel exibiu também a nuvem de palavras gerada a partir de 2 justificativas analisadas, com destaque para os termos “ainda”, “risco”, “justificativas”, “detalhamento” e “rastreabilidade”, além dos *widgets* de IA para geração de análise de justificativas e relatório de risco de conformidade, conforme a Figura 75.

Resultado obtido: todos os componentes do *dashboard* do questionário carregados e exibidos corretamente, incluindo distribuição por faixa, *heatmap* e nuvem de palavras. **Situação: passou.**

D.2.16 TF16 — *Dashboard* individual do representante (UC23)

Cenário: verificação do painel individual de um representante após o encerramento de um questionário.

Pré-condição: representante autenticado e o questionário *Sprint* 1 do projeto 16 foi encerrado com seu ICP calculado.

Execução: o testador acessou o painel individual do questionário 46 e verificou os indicadores exibidos. O painel apresentou ICP da iteração atual de 100,00%, classificado na Faixa A, média pessoal do projeto de 87,50% e média da equipe de 64,29%, conforme a Figura 76. O gráfico de evolução de conformidade exibiu 100% na *Sprint* 1 e 75% na *Sprint* 2, com interpretação textual indicando que o ICP atual é de 100,00% — Faixa A

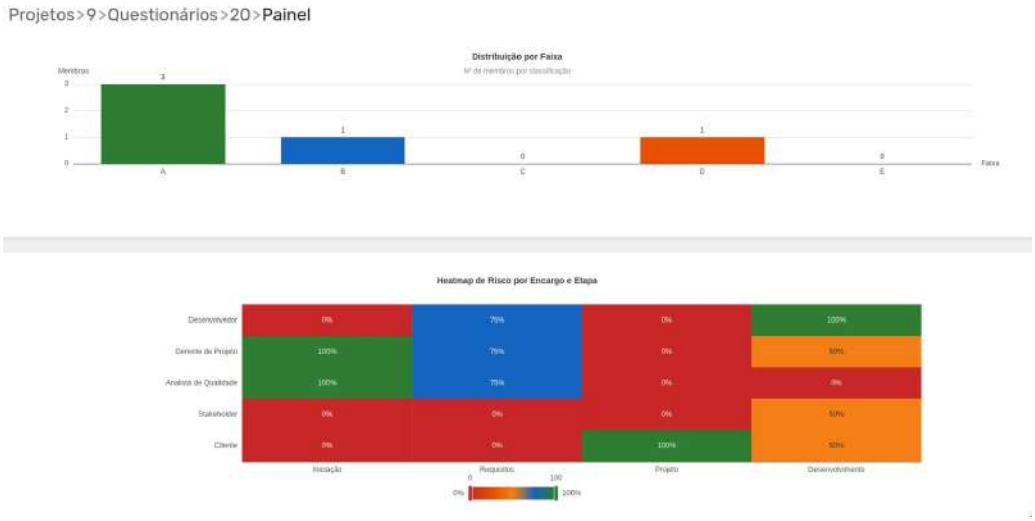

Figura 74 – *Dashboard* do questionário 20 com gráfico de distribuição por faixa e *heatmap* de risco por encargo e etapa.

Fonte: O autor.

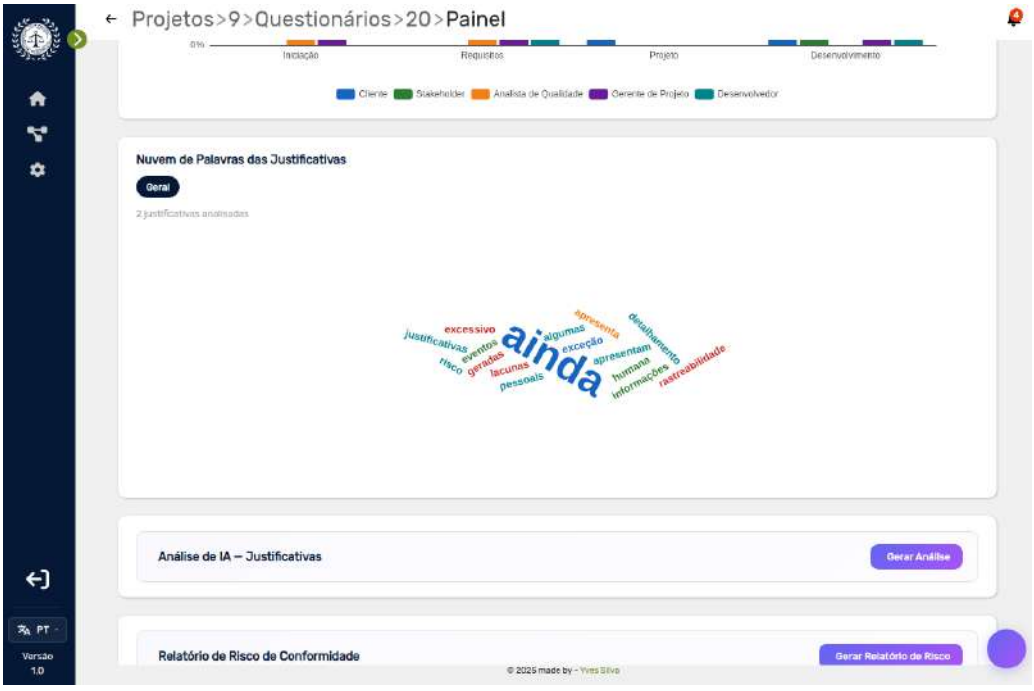

Figura 75 – *Dashboard* do questionário 20 com nuvem de palavras das justificativas e *widgets* de análise de IA.

Fonte: O autor.

— e que o representante está acima da média da equipe de 64,29%, conforme a Figura 77.

Resultado obtido: painel individual exibido com ICP, médias e gráfico de evolução histórica do representante. **Situação: passou.**

Figura 76 – Painel individual do representante com ICP 100%, média pessoal e média da equipe.

Fonte: O autor.

Figura 77 – Gráfico de evolução de conformidade do representante e interpretação textual do desempenho.

Fonte: O autor.

D.2.17 TF17 — Exportação de CSV do *dashboard* do projeto (UC25)

Cenário: exportação dos dados analíticos do projeto em formato CSV a partir do *dashboard* consolidado.

Pré-condição: o projeto SelecIA RH está encerrado e o *dashboard* consolidado está disponível.

Execução: o testador clicou em "Relatório do Resultado (CSV)", conforme indicado pela seta na Figura 78. O sistema iniciou o *download* do arquivo isep-projeto-9.csv. A Figura 79 exibe o conteúdo do arquivo aberto em planilha, com as colunas Projeto, Questionário, Iteração/Etapa, Calculado em, ID Membro, Nome Membro, ICP (%), Faixa ICP,

ISEQ Iteração (%), Faixa ISEP, e as métricas de dívida ética, dívida técnica e IEM por etapa, com 20 linhas de dados correspondentes aos cinco representantes nas quatro *sprints* do projeto.

Resultado obtido: arquivo CSV gerado e *download* iniciado corretamente, com todos os campos de indicadores e identificadores esperados. **Situação: passou.**

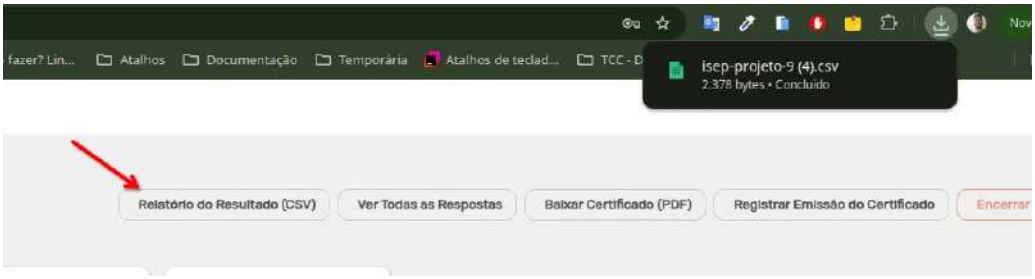

Figura 78 – *Dashboard* do projeto com destaque para o botão de exportação CSV e *download* iniciado no navegador.

Fonte: O autor.

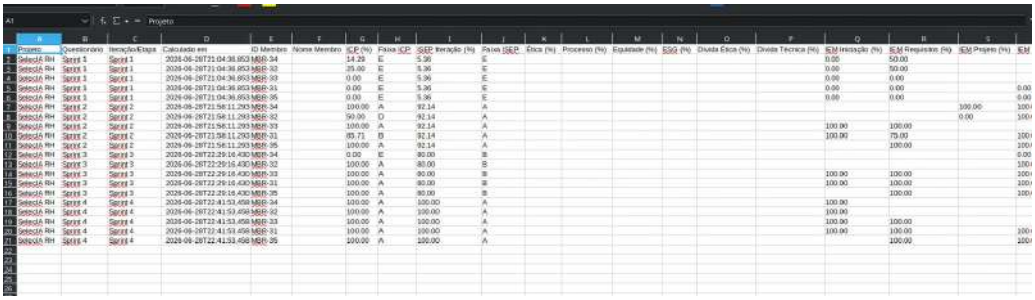

Figura 79 – Arquivo CSV aberto em planilha com os dados analíticos do projeto por representante e por *sprint*.

Fonte: O autor.

D.2.18 TF18 — Exportação de JSON do *dashboard* do questionário (UC25)

Cenário: exportação dos dados analíticos de um questionário em formato JSON a partir do *dashboard* do questionário.

Pré-condição: o questionário 20 do projeto 9 está encerrado e o *dashboard* do questionário está disponível.

Execução: o testador clicou em "Relatório do Resultado (JSON)" no *dashboard* do questionário. O sistema iniciou o *download* do arquivo iseq-questionario-20.json com 2.454 bytes, conforme a Figura 80.

Resultado obtido: arquivo JSON gerado e *download* iniciado corretamente. **Situação: passou.**

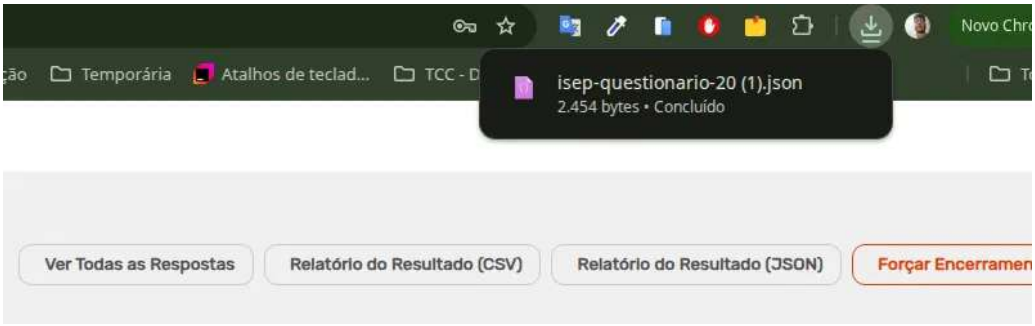

Figura 80 – Histórico de *downloads* do navegador exibindo o arquivo JSON do questionário em *download*.

Fonte: O autor.

D.2.19 TF19 — Configuração e uso do *token* de IA (UC29, UC30)

Cenário: configuração do *token* pessoal de IA e posterior utilização dos recursos analíticos do *dashboard*, com persistência segura do *token* e geração de explicação dos resultados.

Pré-condição: o usuário está autenticado, acessa Configurações > *Token* de IA e, em seguida, possui um questionário com *dashboard* disponível para consulta.

Execução: O testador acessou a tela Configurações > *AI-Token*, conforme a Figura 82, na qual o sistema exibiu o *card Token* de IA com orientações sobre o uso da chave pessoal. Na sequência, retornou ao *dashboard* do questionário, exibido na Figura 81, e acionou o *widget* Explicação dos Resultados ISEQ. O sistema carregou a análise contextual do resultado, apresentou a média ponderada da equipe, os valores individuais de ICP e o texto explicativo do painel, interpretando o ISEQ obtido na faixa crítica e descrevendo as implicações do resultado para o projeto. A interface exibiu a análise com sucesso, sem falhas de comunicação com o serviço de IA, confirmando o funcionamento integrado entre a configuração do *token* e o consumo dos recursos analíticos.

Resultado obtido: *token* salvo com sucesso e análise gerada corretamente no *widget* de explicação dos resultados. **Situação: passou.**

D.2.20 TF20 — Reagendamento de questionário pendente (UC17)

Cenário: reagendamento do primeiro questionário pendente após a conclusão de ao menos um questionário no projeto.

Pré-condição: o usuário está autenticado como administrador, o questionário *Sprint* 1 foi concluído e o questionário *Sprint* 2 está com *status* “Pendente”.

Execução: após o encerramento do *Sprint* 1, o testador verificou na tela de detalhes do projeto 16 que o *Sprint* 2 aparecia com *status* “Pendente”, 0 de 3 respondentes e a ação “Reagendar” disponível no painel de ações, conforme indicado pela seta na Figura 83. O testador clicou em “Reagendar” e informou o novo período. O sistema exibiu o *modal* de

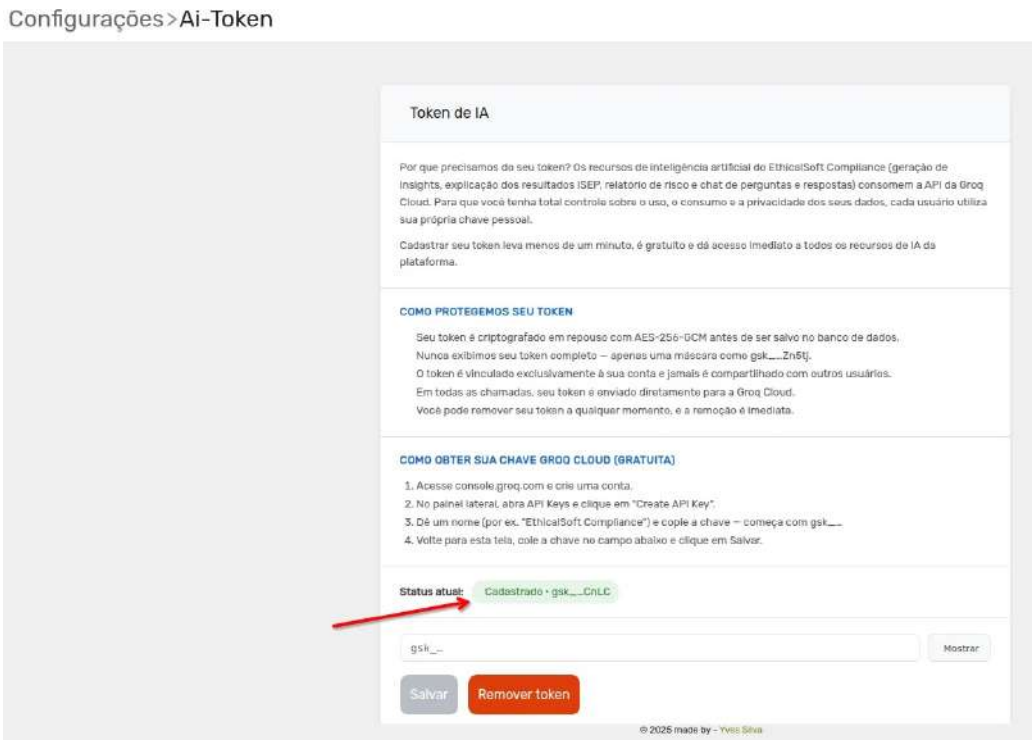

Figura 81 – *Dashboard* do questionário com *widget* de Explicação dos Resultados ISEQ.
Fonte: O autor.

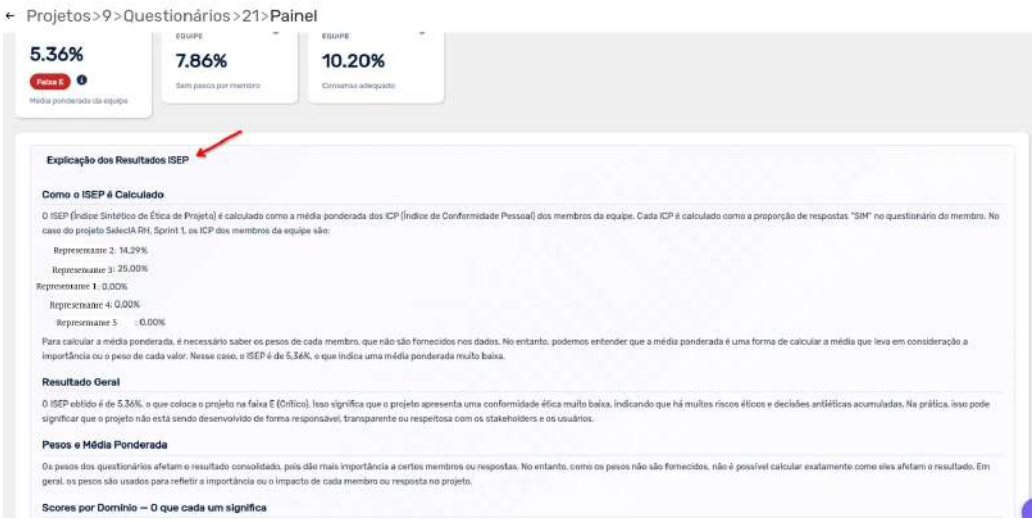

Figura 82 – Tela de configuração do *token* de IA com *status* cadastrado e opções de salvar/remover.

Fonte: O autor.

sucesso com a mensagem “Questionário ’*Sprint* 2’ reagendado com sucesso. Novo período: 30/06/2026 até 13/07/2026”, conforme a Figura 84.

Após o reagendamento, a tela de detalhes exibiu o *Sprint* 2 com *status* atualizado para “Em andamento”, período de aplicação ajustado para 30/06/2026 até 13/07/2026 e o botão “Responder agora” disponível, confirmando a transição de *status*, conforme as setas indicativas na Figura 85. Abaixo, o *Sprint* 1 aparecia com *status* “Concluído” e todos os representantes com registro de resposta, evidenciando a pré-condição necessária para o reagendamento. O sistema enviou automaticamente e-mail de notificação aos representantes informando o reagendamento, com o período anterior e o novo período, conforme a Figura 86.

Resultado obtido: questionário reagendado com sucesso, período atualizado, *status* transitado para “Em andamento” e notificação enviada por e-mail aos representantes. **Situação: passou.**

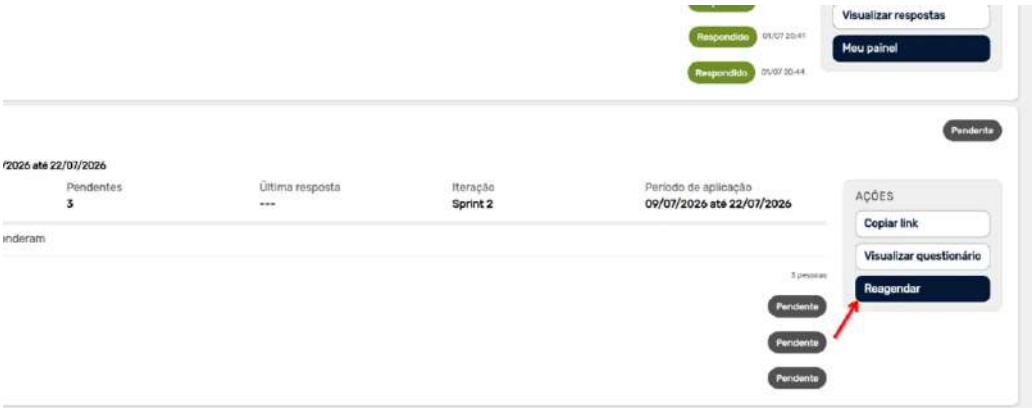

Figura 83 – Tela de detalhes do projeto 16 com *Sprint* 2 pendente e destaque para o botão de reagendamento.

Fonte: O autor.

Figura 84 – *Modal* de sucesso confirmando o reagendamento do questionário *Sprint* 2 com o novo período.

Fonte: O autor.

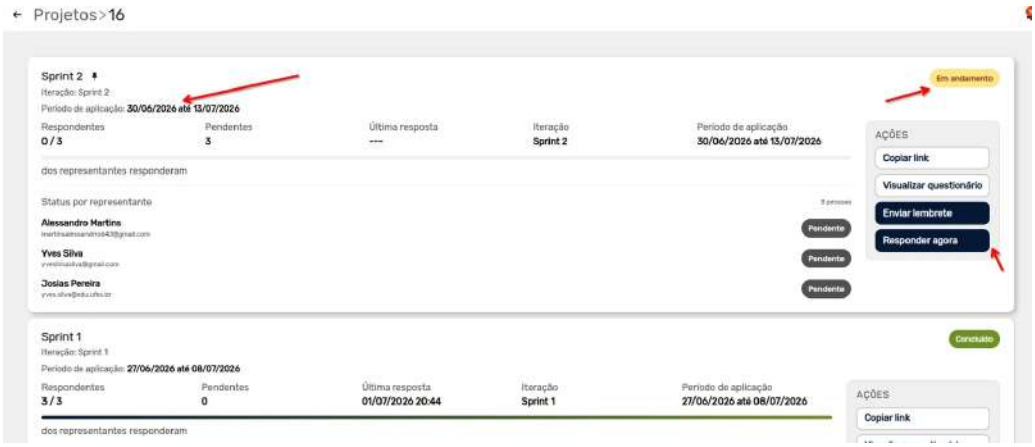

Figura 85 – Tela de detalhes do projeto 16 após reagendamento, com *Sprint* 2 em andamento e novo período exibido.

Fonte: O autor.

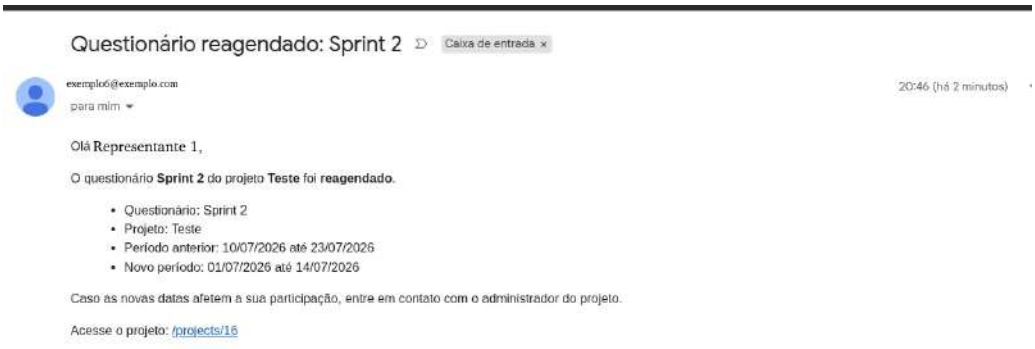

Figura 86 – E-mail de notificação de reagendamento recebido pelo representante com período anterior e novo período.

Fonte: O autor.

APÊNDICE E – Especificação do Cenário Demonstrativo SelecIA RH

E.1 Descrição do cenário

O SelecIA RH é um sistema fictício de apoio ao recrutamento e seleção que emprega inteligência artificial para analisar currículos, identificar competências, propor ranqueamento de candidatos e gerar justificativas resumidas. O cenário foi escolhido por reunir riscos éticos representativos do uso de IA em processos que envolvem pessoas: privacidade de dados, vieses algorítmicos, falta de transparência, responsabilização e discriminação indireta. Os dados são inteiramente simulados e não representam equipe, organização ou produto real.

E.2 Configuração do projeto

Tipo: Iterativo Incremental — 4 *sprints*
***Template* de base:** Base Ágil Iterativo (público)
Limiar mínimo de conformidade: 75% (faixa B)

Quadro 17 – Representantes, papéis e pesos.

Representante	**Papel**	**Peso**
Representante 1	Gerente de Projeto	2,0
Representante 2	Analista de Qualidade	2,0
Representante 3	Desenvolvedor	3,0
Representante 4	*Stakeholder*	1,0
Representante 5	Cliente	2,0

Fonte: O autor.

Quadro 18 – Questionários e pesos.

Questionário	**Peso**	**Foco temático**
Sprint 1	1,0	Governança inicial, LGPD, política de triagem com IA
Sprint 2	1,0	Ranqueamento assistido por IA, explicabilidade, rastreabilidade

Continua na próxima página...

Continuação do Quadro 18

Questionário	**Peso**	**Foco temático**
Sprint 3	1,5	Desenvolvimento, privacidade, revisão de código
Sprint 4	2,0	Testes de viés, validação com *stakeholders*, encerramento ético

Fonte: O autor.

Quadro 19 – Peso por Etapas.

Etapa	**Peso**
Iniciação	2,00
Requisitos	3,00
Projeto	2,00
Desenvolvimento	2,50
Testes	1,50

Fonte: O autor.

E.3 Distribuição e classificação das perguntas

As perguntas foram organizadas em três classificações de escopo, conforme descrito nesta seção.

Perguntas de projeto inteiro: concentradas na *Sprint* 1. Tratam de governança estrutural: aprovação do uso de IA, diretrizes sobre dados sensíveis, definição de responsável pela revisão humana, documentação de valores e riscos éticos, canal de reporte anônimo, definição de autoridade decisória e aprovação orçamentária.

Perguntas recorrentes: presentes em todas as *sprints*. Verificam alinhamento ético da *sprint* com os limites definidos, revisão de riscos antes de iniciar, revisão humana de artefatos gerados com IA, rastreabilidade das decisões e ausência de uso indevido de dados sensíveis.

Perguntas da etapa atual: variam por *sprint*.

Sprint 1: LGPD e minimização de dados, identificação de critérios de triagem com impacto em grupos vulneráveis, avaliação de vieses em sugestões de IA.

Sprint 2: capacidade de discordância do recrutador sobre o *ranking*, rastreabilidade entre critérios e decisão final, controle de exposição de atributos sensíveis nas justificativas.

Sprint 3: identificação de código gerado com IA para revisão antes de *merge*, documentação de *trade-offs* técnicos com justificativa ética, respeito às políticas de privacidade nos dados de desenvolvimento.

Sprint 4: submissão de artefatos de IA aos mesmos critérios de aceite do restante do *software*, testes específicos para detecção de vieses, compartilhamento dos resultados com

stakeholders para validação, registro de falhas e alucinações de IA, verificação de condições mínimas de encerramento ético.

E.4 Respostas configuradas por *sprint*

***Sprint* 1 — ISEQ: 5,36% — Faixa E** Cenário configurado para refletir ausência de governança ética inicial. As perguntas de projeto inteiro foram respondidas predominantemente com "Não" por todos os papéis elegíveis, indicando ausência de documentação de valores, de mapeamento de riscos e de processos de revisão humana. A única resposta afirmativa confirmada para todos os papéis elegíveis foi a pergunta sobre a identificação de critérios de triagem com potencial impacto negativo em grupos vulneráveis, o que, no contexto, evidenciou o risco sem mitigá-lo.

***Sprint* 2 — ISEQ: 92,14% — Faixa A** As lacunas identificadas na *Sprint* 1 foram corrigidas sistematicamente. As perguntas recorrentes e as específicas da *sprint* foram respondidas afirmativamente pela maioria dos representantes. A divergência concentrou-se no papel de *Stakeholder*, que registrou resposta negativa para a pergunta sobre o controle de exposição de atributos sensíveis nas justificativas geradas pela IA, sinalizando risco residual em explicabilidade.

***Sprint* 3 — ISEQ: 80,00% — Faixa B** A maioria das respostas foi afirmativa. A divergência ocorreu no papel de Cliente, que registrou resposta negativa para a pergunta sobre respeito às políticas de privacidade no tratamento de dados de candidatos durante o desenvolvimento. Os demais representantes responderam afirmativamente, gerando desvio padrão de 40%.

***Sprint* 4 — ISEQ: 100% — Faixa A** Todas as respostas configuradas como afirmativas para todos os representantes elegíveis. Desvio padrão: 0%.

E.5 Resultados esperados e obtidos

Quadro 20 – Resultados esperados e obtidos no cenário SelecIA RH.

Sprint	**Peso**	**ISEQ esperado**	**ISEQ obtido**	**Faixa esperada**	**Faixa obtida**	**Situação**
Sprint 1	1,0	5,36%	5,36%	E	E	Conforme
Sprint 2	1,0	92,14%	92,14%	A	A	Conforme
Sprint 3	1,5	80,00%	80,00%	B	B	Conforme
Sprint 4	2,0	100,00%	100,00%	A	A	Conforme

Continua na próxima página...

Continuação do Quadro 20

Sprint	**Peso**	**ISEQ esperado**	**ISEQ obtido**	**Faixa esperada**	**Faixa obtida**	**Situação**
ISEP consolidado	—	75,91%	75,91%	B	B	Conforme

Fonte: O autor.

APÊNDICE F – Contrato da API, Instruções de Execução e Identificação da Versão

Quadro 21 – Identificação da versão entregue.

Item	**Valor**
Nome do projeto	`ethicalsoft-compliance`
Versão	`0.0.1-SNAPSHOT`
Commit avaliado (SonarQube)	`4175115d`
Branch principal	`main`
Data da última execução dos testes	30 de junho de 2026
Resultado da suíte automatizada	43 testes, 0 falhas, 0 erros
Repositório da solução	<https://github.com/yvesls/ethicalsoft-compliance>

Fonte: O autor.

F.1 Pré-requisitos de execução

Java 21 (JDK, via *toolchain* Maven configurado).
Maven 3.9+.
Docker e Docker Compose para PostgreSQL 16, MongoDB 8 e Nginx.
Node.js 20+ e Angular CLI 22+ para o *frontend*.
Conta no Groq Cloud com *token* de API, opcional para execução sem IA.

Quadro 22 – Variáveis de ambiente obrigatórias do *backend*.

Variável	**Descrição**
SPRING_DATASOURCE_URL	URL JDBC do PostgreSQL, por exemplo jdbc:postgresql://localhost:5432/ethicalsoft.
SPRING_DATASOURCE_USERNAME	Usuário do PostgreSQL.
SPRING_DATASOURCE_PASSWORD	Senha do PostgreSQL.

Continua na próxima página...

Continuação do Quadro 22

Variável	**Descrição**
SPRING_DATA_MONGODB_URI	URI de conexão do MongoDB, por exemplo mongodb://localhost:27017/ethicalsoft.
JWT_SECRET	Chave secreta para assinatura dos *tokens* JWT.
GOOGLE_CLIENT_ID	*Client ID* do provedor Google Identity Services.
SPRING_MAIL_HOST	Servidor SMTP para envio de e-mails.
SPRING_MAIL_USERNAME	Usuário do servidor SMTP.
SPRING_MAIL_PASSWORD	Senha do servidor SMTP.
ENCRYPTION_KEY	Chave de criptografia dos *tokens* pessoais de IA.

Fonte: O autor.

F.2 Instruções de execução local

Backend

bash

./mvnw clean install

./mvnw spring-boot:run

A API estará disponível em http://localhost:8080. As migrações Flyway são executadas automaticamente na inicialização.

Frontend

bash

npm install

ng serve

O *frontend* estará disponível em http://localhost:4200. A URL base da API é configurável no arquivo environments/environment.ts.

Com Docker Compose

bash

docker compose up -d

Sobe PostgreSQL, MongoDB e Nginx. O *backend* e o *frontend* podem ser executados fora do contêiner durante o desenvolvimento.

F.3 Contrato resumido da API REST

Os *endpoints* estão organizados por domínio funcional. Todos os *endpoints*, exceto os de autenticação, exigem o cabeçalho Authorization: Bearer <*token*>.

Quadro 23 – Autenticação (/api/auth).

Método	**Rota**	**Descrição**
POST	`</login>`	*Login* com e-mail e senha
POST	`</login/google>`	*Login* com *token* Google
POST	/register	Cadastro de novo usuário
POST	/recover	Solicitação de código de recuperação
POST	/verify-code	Validação do código de recuperação
POST	/reset-password	Redefinição de senha

Fonte: O autor.

Quadro 24 – Projetos (/api/projects).

Método	**Rota**	**Descrição**
GET	/	Listagem paginada com filtros
POST	/	Criação de projeto
GET	/{id}	Detalhes do projeto
PUT	/{id}	Edição do projeto (com *dry-run*)
DELETE	/{id}	Exclusão do projeto
POST	/{id}/publish	Publicação do projeto
POST	/{id}/close	Encerramento e cálculo do ISEP

Fonte: O autor.

Quadro 25 – Questionários (/api/projects/{projectId}/questionnaires).

Método	**Rota**	**Descrição**
GET	/	Listagem dos questionários do projeto
POST	/{id}/close	Encerramento e cálculo do ISEQ
POST	/{id}/reschedule	Reagendamento do questionário

Fonte: O autor.

Quadro 26 – Respostas (/api/questionnaires/{questionnaireId}/responses).

Método	**Rota**	**Descrição**
GET	/my	Carrega resposta do representante autenticado
POST	/	Submissão definitiva das respostas
PUT	/draft	Salvamento de rascunho

Fonte: O autor.

Quadro 27 – *Dashboard* (`/api/dashboard`).

Método	**Rota**	**Descrição**
GET	/project/{id}	Painel consolidado do projeto
GET	/questionnaire/{id}	Painel analítico do questionário
GET	/individual/{questionnaireId}	Painel individual do representante autenticado

Fonte: O autor.

Quadro 28 – Documentos (/api/documents).

Método	**Rota**	**Descrição**
GET	/project/{id}/certificate	*Download* do certificado em PDF
POST	/project/{id}/certificate/register	Registro auditável da emissão do certificado
GET	/questionnaire/{id}/bulletin	*Download* do boletim em PDF
POST	/questionnaire/{id}/bulletin/register	Registro auditável da emissão do boletim
POST	/questionnaire/{id}/bulletin/emit	Envio do boletim por e-mail aos representantes

Fonte: O autor.

Quadro 29 – Inteligência artificial (/api/ai).

Método	**Rota**	**Descrição**
POST	/explain/{questionnaireId}	Explicação dos resultados ISEQ
POST	/analyze/{questionnaireId}	Análise das justificativas
POST	/risk/{questionnaireId}	Relatório de risco de conformidade
POST	`</chat/{questionnaireId}>`	*Chat* contextual com *streaming*
GET	`</token>`	*Status* do *token* pessoal configurado
POST	`</token>`	Salvar *token* pessoal
DELETE	`</token>`	Remover *token* pessoal

Fonte: O autor.

Quadro 30 – *Templates* (`/api/templates`).

Método	**Rota**	**Descrição**
GET	/	Listagem dos *templates* disponíveis
GET	/{id}	Detalhes do *template*

Fonte: O autor.

Quadro 31 – Exportação (/api/export).

Método	**Rota**	**Descrição**
GET	/project/{id}/csv	Exportação CSV do painel do projeto
GET	/questionnaire/{id}/json	Exportação JSON do painel do questionário

Fonte: O autor.

APÊNDICE G – Telas Complementares da Aplicação

Este apêndice reúne as interfaces complementares aos fluxos centrais apresentados na Seção 5.3.2. As telas de autenticação, acesso aos projetos, criação, gerenciamento, resposta aos questionários, painéis e documentos permanecem no Capítulo 5, no qual são relacionadas às regras e às decisões implementadas.

G.1 Seleção do Tipo de Projeto

Antes do preenchimento do assistente de criação, o administrador seleciona a estrutura em cascata ou iterativa. Essa escolha condiciona a organização das etapas ou iterações, dos questionários e dos respectivos pesos, conforme a Figura 87.

Figura 87 – Seleção da estrutura do projeto antes da configuração.

Fonte: O autor.

G.2 Configurações da Conta

A área de configurações permite alterar a senha e gerenciar a credencial pessoal empregada nas funções assistivas de IA, como apresenta a Figura 88.

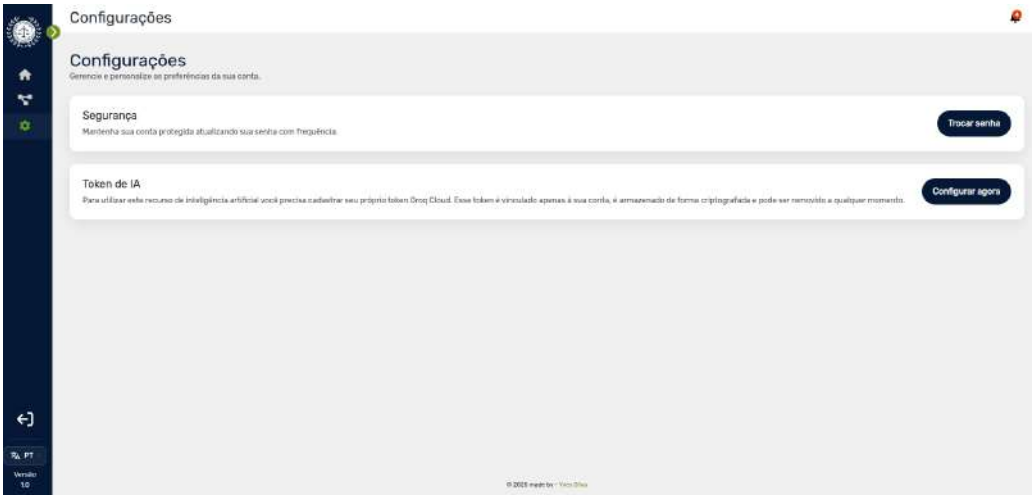

Figura 88 – Configurações da conta e gerenciamento da credencial de IA.

Fonte: O autor.