

Uma Ferramenta Analítica para Resultados do Instrumento Zeppelin

Alexandre Leandro Seixas Adler

Relatório Técnico - IC-PFG-26-04

Projeto Final de Graduação

2026 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Uma Ferramenta Analítica para Resultados do Instrumento Zeppelin

Alexandre Leandro Seixas Adler

Orientador: Breno Bernard Nicolau de França

Instituto de Computação, Universidade Estadual de Campinas, 13083-852, Campinas, SP, Brasil.

Resumo

A adoção de práticas de Engenharia de Software Contínua tem se tornado relevante para organizações que buscam melhorar continuamente seus processos de desenvolvimento de *software*, reduzir riscos, ampliar a previsibilidade das entregas e responder de forma mais adequada a mudanças de contexto. Nesse cenário, instrumentos de diagnóstico, como o Zeppelin, permitem avaliar o grau de adoção de práticas contínuas e apoiar a identificação de forças, fragilidades e oportunidades de evolução. Entretanto, a utilidade prática desses instrumentos depende não apenas da coleta das respostas, mas também da forma como seus resultados são organizados, interpretados e apresentados aos usuários.

Este trabalho apresenta o desenvolvimento de uma ferramenta analítica voltada à apresentação dos resultados do instrumento Zeppelin. A ferramenta concentra-se em transformar as saídas do diagnóstico em visualizações, recomendações organizadas e componentes de consulta histórica por ciclos de avaliação. Com isso, busca apoiar a leitura dos resultados sob diferentes perspectivas analíticas, como estágios, dimensões, processos, elementos e práticas avaliadas, sem modificar a estrutura de avaliação do instrumento.

O desenvolvimento envolveu revisão bibliográfica sobre Engenharia de Software Contínua, práticas contínuas e instrumentos de diagnóstico de maturidade, análise da estrutura do Zeppelin, definição dos requisitos da ferramenta, concepção das visões analíticas, prototipação e implementação da interface *web*. Como contribuição, o trabalho disponibiliza uma camada analítica mais clara e navegável para os resultados do Zeppelin, apoiando a interpretação do diagnóstico, a consulta a recomendações e o planejamento de ações de melhoria contínua em organizações de *software*.

Sumário

1	Introdução	3
2	Fundamentação Teórica	4
2.1	Engenharia de Software Contínua	5
2.2	Práticas Contínuas: Integração, Entrega e Implantação	5
2.3	Modelos em Engenharia de Software Contínua	6
2.4	Instrumentos de Diagnóstico e Instrumento Zeppelin	7
3	Trabalhos Correlatos	9
4	Concepção da Solução Proposta	10
4.1	Requisitos e Escopo	11
4.2	Fluxo de Uso	13
4.3	Regras de Diagnóstico e Recomendações	14
5	Desenvolvimento da Ferramenta	16
5.1	Arquitetura da Solução	16
5.2	Tecnologias Utilizadas	20
5.3	Estrutura Analítica dos Resultados	21
5.4	Interface e Funcionalidades	22
5.4.1	Painel	23
5.4.2	Resultados	24
5.4.3	Recomendações	26
5.4.4	Interpretação histórica	27
5.5	Disponibilização do código-fonte	28
6	Discussão	29
7	Conclusão e Trabalhos Futuros	30
	Agradecimentos	31
	Apêndice A – Capturas Complementares da Tela de Resultados	32
	Referências	35

1 Introdução

Sistemas de *software* estão presentes em grande parte das atividades organizacionais, econômicas e sociais. Nesse cenário, organizações precisam lidar com demandas cada vez mais dinâmicas, ciclos de entrega mais curtos, maior exigência de qualidade e necessidade constante de adaptação. O desenvolvimento de *software*, portanto, deixa de ser compreendido apenas como uma sequência de etapas isoladas e passa a exigir práticas capazes de integrar planejamento, construção, validação, entrega, operação, monitoramento e aprendizado contínuo.

Nesse contexto, a Engenharia de Software Contínua, do inglês *Continuous Software Engineering* (CSE), surge como uma abordagem voltada à organização de práticas e ferramentas contínuas ao longo do ciclo de vida do *software*. Essa perspectiva busca tornar o desenvolvimento mais rápido, iterativo, integrado e alinhado ao negócio, compreendendo o processo de *software* como um fluxo contínuo, e não como um conjunto de atividades discretas e desconectadas [3].

Entre as práticas associadas à CSE, destacam-se a Integração Contínua, a Entrega Contínua e a Implantação Contínua. A Integração Contínua refere-se à integração frequente de alterações de código, apoiada por automação de *build*, testes e mecanismos de *feedback*. Já a Entrega Contínua e a Implantação Contínua buscam tornar o processo de liberação de *software* mais confiável, frequente e automatizado, reduzindo riscos e aumentando a previsibilidade das entregas [4, 8].

Apesar da relevância dessas práticas, sua adoção organizacional não é homogênea. Diferentes organizações podem apresentar níveis distintos de adoção de práticas de CSE, inclusive com avanços em algumas dimensões e fragilidades em outras. Esse cenário torna importante o uso de instrumentos capazes de diagnosticar o estado atual das práticas adotadas e de apoiar a identificação de oportunidades de evolução [2].

O instrumento Zeppelin foi proposto com esse objetivo, permitindo avaliar de forma eficiente o grau de adoção de práticas de Engenharia de Software Contínua em organizações. O instrumento organiza práticas relacionadas à evolução em CSE e permite obter uma visão panorâmica da organização, considerando diferentes perspectivas de análise. Em aplicações anteriores, o Zeppelin foi utilizado para identificar práticas adotadas, lacunas e possíveis áreas de melhoria em organizações de *software* [2].

Trabalhos posteriores também exploraram o uso do Zeppelin como base para diagnóstico e recomendações, especialmente no contexto de práticas de Integração Contínua e Implantação Contínua. Nesse sentido, o trabalho de Araújo propõe recomendações personalizadas com base nos resultados gerados pelo instrumento, com o objetivo de apoiar organizações na evolução de suas práticas de CI/CD [1].

Entretanto, a existência de um instrumento diagnóstico e de recomendações associadas não elimina a necessidade de uma apresentação clara e estruturada dos resultados. Para que os dados obtidos sejam efetivamente úteis, é necessário que sejam organizados de forma compreensível, permitindo ao usuário identificar o panorama geral do diagnóstico, as forças, os gargalos, as recomendações e os possíveis caminhos de evolução. Assim, a forma de apresentação dos resultados torna-se parte relevante do apoio à análise e à tomada de decisão.

Diante desse contexto, este Projeto Final de Graduação tem como objetivo desenvolver uma ferramenta analítica para apoiar a apresentação dos resultados do instrumento Zeppelin. O foco do trabalho não está na definição de novas regras de avaliação, na alteração da estrutura do instrumento ou na criação de um novo modelo de maturidade, mas na construção de uma camada analítica e visual para organizar os resultados já produzidos pelo Zeppelin.

A ferramenta proposta busca transformar os dados do diagnóstico em informações mais acessíveis e interpretáveis, permitindo ao usuário visualizar o estado geral de um ciclo de avaliação, identificar forças e gargalos, consultar recomendações, acessar ciclos anteriores e observar informações históricas associadas às avaliações. Além disso, a ferramenta permite explorar os resultados sob diferentes perspectivas analíticas, como estágios, dimensões, processos, elementos e práticas avaliadas.

A principal contribuição deste trabalho consiste no desenvolvimento da camada analítica, voltada à apresentação dos resultados do Zeppelin. Essa camada envolve principalmente o painel, a tela de resultados detalhados, a organização das recomendações e componentes específicos de interpretação histórica, voltados a apoiar a leitura dos resultados entre os ciclos de avaliação. Dessa forma, a ferramenta pretende ampliar a utilidade prática do diagnóstico ao oferecer uma apresentação mais clara e navegável dos resultados, apoiando a análise e o planejamento de ações de melhoria contínua em organizações de *software*.

Metodologicamente, este trabalho envolve revisão bibliográfica sobre Engenharia de Software Contínua, práticas contínuas e instrumentos de diagnóstico de maturidade; análise da estrutura do instrumento Zeppelin; definição dos requisitos da ferramenta; concepção das visões analíticas; prototipação e desenvolvimento da ferramenta analítica; e análise funcional do artefato desenvolvido. A avaliação deste trabalho fica concentrada no funcionamento da ferramenta e na forma como ela organiza e apresenta os resultados do Zeppelin. Assim, não são afirmados impactos práticos em organizações reais, como melhoria de maturidade ou aumento de eficiência, uma vez que esses efeitos não foram medidos durante o projeto.

O restante deste relatório está organizado da seguinte forma. A Seção 2 apresenta a fundamentação teórica, abordando Engenharia de Software Contínua, práticas contínuas de integração, entrega e implantação, modelos de evolução em CSE, instrumentos de diagnóstico de maturidade e o instrumento Zeppelin. A Seção 3 discute trabalhos correlatos relacionados à avaliação de práticas contínuas, modelos de maturidade e apoio à evolução organizacional. A Seção 4 apresenta a concepção da ferramenta analítica, incluindo requisitos, escopo, fluxo de uso e regras de diagnóstico e recomendações. A Seção 5 descreve o desenvolvimento da ferramenta, contemplando arquitetura, tecnologias utilizadas, estrutura analítica dos resultados e interface. A Seção 6 discute os principais aspectos do artefato desenvolvido, suas limitações e seu alinhamento ao problema proposto. Por fim, a Seção 7 apresenta as conclusões e indica possibilidades de trabalhos futuros.

2 Fundamentação Teórica

A fundamentação teórica reúne os principais conceitos necessários para compreender o contexto deste trabalho. Inicialmente, discute-se a Engenharia de Software Contínua como

uma abordagem abrangente para organizar práticas contínuas no desenvolvimento de *software*. Em seguida, são apresentadas práticas contínuas associadas à integração, à entrega e à implantação de *software*. Na sequência, são discutidos modelos de evolução em Engenharia de Software Contínua, especialmente aqueles utilizados como base para o instrumento Zeppelin. Por fim, são abordados os instrumentos de diagnóstico de maturidade e o próprio Zeppelin, que constitui a base dos resultados apresentados pela ferramenta desenvolvida neste Projeto Final de Graduação.

2.1 Engenharia de Software Contínua

A Engenharia de Software Contínua, do inglês *Continuous Software Engineering* (CSE), pode ser compreendida como uma abordagem que busca organizar práticas, processos e ferramentas de forma contínua ao longo do ciclo de vida do *software*. Essa perspectiva está relacionada à necessidade de estabelecer um fluxo mais integrado entre as demandas dos usuários, o desenvolvimento do produto, a sua entrega e a sua operação. Assim, a CSE não se limita a uma única prática técnica, mas envolve um conjunto mais amplo de atividades contínuas que conectam aspectos de negócio, desenvolvimento, qualidade, operação e aprendizado organizacional [3].

A adoção de CSE está associada a mudanças na forma como as organizações planejam, desenvolvem, validam, entregam e monitoram *software*. Nesse contexto, práticas como planejamento contínuo, integração contínua, entrega contínua, implantação contínua, experimentação contínua, monitoramento e aprendizado a partir de *feedback* passam a compor uma visão mais integrada do processo de desenvolvimento. Essa visão reforça que o *software* não deve ser tratado apenas como um produto entregue em grandes ciclos isolados, mas como um artefato em evolução constante, influenciado por dados, *feedback* dos usuários e necessidades de negócio [3, 5].

Dessa forma, a CSE amplia a visão tradicional de desenvolvimento de *software* ao propor maior continuidade entre diferentes atividades e áreas da organização. Em vez de considerar planejamento, codificação, testes, implantação e operação como etapas separadas, a abordagem contínua busca reduzir barreiras entre essas atividades e favorecer ciclos mais curtos de aprendizado e adaptação. Essa característica é importante para organizações que precisam responder rapidamente a mudanças de mercado, corrigir falhas com agilidade e evoluir seus produtos de forma incremental.

No contexto deste trabalho, a CSE é utilizada como base conceitual para compreender o instrumento Zeppelin. Isso ocorre porque o Zeppelin não avalia apenas uma prática isolada, mas sim um conjunto de práticas relacionadas à evolução da organização em direção a formas mais contínuas de desenvolvimento de *software*. Assim, compreender CSE é essencial para interpretar os resultados produzidos pelo instrumento e para justificar a necessidade de uma ferramenta que organize esses resultados de forma analítica.

2.2 Práticas Contínuas: Integração, Entrega e Implantação

A Integração Contínua, do inglês *Continuous Integration* (CI), é uma prática de desenvolvimento de *software* baseada na integração frequente de alterações de código em um

repositório compartilhado. Em geral, essa prática envolve o uso de controle de versão, automação de *build*, execução de testes automatizados e mecanismos de *feedback* rápido para os desenvolvedores. O objetivo é identificar problemas de integração o mais cedo possível, evitando que falhas se acumulem e sejam descobertas apenas em fases tardias do desenvolvimento [8].

A CI contribui para aumentar a previsibilidade do processo de desenvolvimento, pois permite que cada alteração seja validada de forma mais rápida e sistemática. Quando bem adotada, essa prática reduz o risco de conflitos entre alterações de código, melhora a visibilidade sobre o estado do projeto e favorece uma cultura de integração frequente. Além disso, a Integração Contínua serve como base para práticas mais avançadas, como Entrega Contínua e Implantação Contínua, uma vez que a liberação frequente de *software* depende de um processo confiável de integração e validação.

A Entrega Contínua, do inglês *Continuous Delivery*, está relacionada à capacidade de manter o *software* em um estado potencialmente implantável. Isso significa que as alterações realizadas no sistema devem passar por um processo automatizado de construção, validação e preparação, de modo que possam ser liberadas com segurança quando houver decisão organizacional para isso. A Entrega Contínua busca reduzir riscos no processo de liberação e tornar as entregas mais frequentes, previsíveis e confiáveis [4].

A Implantação Contínua, do inglês *Continuous Deployment*, representa um passo além da Entrega Contínua. Nesse caso, alterações aprovadas pelo *pipeline* automatizado podem ser implantadas automaticamente em ambiente de produção ou em ambientes equivalentes, sem necessidade de uma decisão manual para cada liberação. Assim, enquanto a Entrega Contínua garante que o *software* está pronto para implantação, a Implantação Contínua também automatiza a etapa de disponibilização da nova versão [8].

Essas práticas dependem de uma base técnica e organizacional madura. Para que integração, entrega e implantação ocorram de forma contínua, é necessário contar com testes automatizados confiáveis, ambientes consistentes, controle de configuração, *pipelines* bem definidos, monitoramento adequado e mecanismos de recuperação em caso de falhas. Dessa forma, sua adoção envolve não apenas automação, mas também mudanças na cultura de desenvolvimento, qualidade e operação.

No contexto deste Projeto Final de Graduação, essas práticas são relevantes porque aparecem tanto como parte da evolução de organizações em direção à CSE quanto como estágios analisados pelo Zeppelin. Além disso, ajudam a compreender recomendações associadas a práticas de CI/CD, como as propostas baseadas nos resultados gerados pelo instrumento [1].

2.3 Modelos em Engenharia de Software Contínua

A adoção de práticas de Engenharia de Software Contínua pode ser analisada por meio de modelos que descrevem a evolução das organizações em direção a formas mais contínuas de desenvolvimento, entrega e aprendizado. Esses modelos são importantes porque permitem compreender que a maturidade em CSE não depende apenas da adoção isolada de ferramentas, mas também da evolução gradual de práticas técnicas, organizacionais e de negócio.

Entre esses modelos, destaca-se o *Stairway to Heaven*, que descreve uma trajetória evolutiva em direção a formas mais contínuas de desenvolvimento de *software* [6, 7]. Essa trajetória é organizada em quatro estágios: Organização Ágil, do inglês *Agile Organization*; Integração Contínua, do inglês *Continuous Integration*; Implantação Contínua, do inglês *Continuous Deployment*; e Pesquisa e Desenvolvimento como Sistema de Inovação, do inglês *R&D as Innovation System* [2].

O estágio Organização Ágil representa uma base inicial para a adoção de práticas ágeis e para a organização do trabalho de desenvolvimento. Esse estágio envolve aspectos como colaboração, organização das equipes, foco na entrega de valor e adaptação do processo de desenvolvimento. Ele serve de base para a adoção de práticas contínuas mais avançadas.

O estágio Integração Contínua está associado à integração frequente de alterações de código, à automação de *builds*, a testes e a mecanismos de *feedback* rápido. Nesse estágio, a organização busca reduzir problemas de integração e aumentar a visibilidade do estado do *software* ao longo do desenvolvimento.

O estágio de Implantação Contínua envolve práticas de automação da entrega e da implantação de *software*. Ele considera a capacidade da organização de disponibilizar alterações de forma frequente, confiável e controlada, o que apoia ciclos mais curtos de entrega e de resposta a mudanças.

Por fim, o estágio Pesquisa e Desenvolvimento como Sistema de Inovação está relacionado ao uso de dados, *feedback* e experimentação para apoiar a evolução contínua de produtos e processos. Nesse estágio, a organização passa a utilizar informações provenientes do uso do *software*, dos usuários e do ambiente operacional para orientar decisões de melhoria e inovação.

Além do *Stairway to Heaven*, o Zeppelin também se apoia no *Eye of CSE*, que permite analisar a adoção de práticas contínuas a partir de diferentes dimensões organizacionais [5]. Dessa forma, a maturidade em CSE não é observada apenas por meio de uma sequência de estágios, mas também por perspectivas complementares, como desenvolvimento, qualidade, operação, negócio, conhecimento e usuário/cliente.

Essa combinação de modelos é importante para este trabalho, pois fundamenta as diferentes visões analíticas que a ferramenta deve apresentar. Ao organizar os resultados por estágios, dimensões, práticas avaliadas e níveis de adoção, a ferramenta permite que o usuário compreenda o diagnóstico de forma mais detalhada, identificando forças, fragilidades e oportunidades de evolução.

2.4 Instrumentos de Diagnóstico e Instrumento Zeppelin

Instrumentos de diagnóstico de maturidade são utilizados para avaliar o estado atual de práticas, processos ou capacidades de uma organização. Em Engenharia de Software, esse tipo de instrumento pode apoiar a identificação de práticas já consolidadas, de práticas parcialmente adotadas e de lacunas que dificultam a evolução do processo de desenvolvimento. No contexto de práticas contínuas, essa avaliação é especialmente relevante, pois a adoção de CSE pode ocorrer de forma desigual entre diferentes áreas, estágios e dimensões da organização.

O Zeppelin é um instrumento diagnóstico voltado à avaliação da adoção de práticas de

Engenharia de Software Contínua em organizações. Sua finalidade é apoiar a identificação do estado atual da organização em relação a práticas contínuas, permitindo observar pontos fortes, lacunas e oportunidades de melhoria [2]. A estrutura do instrumento combina modelos conceituais relacionados à CSE, como o *Stairway to Heaven* e o *Eye of CSE*, permitindo analisar os resultados por estágios de evolução, dimensões organizacionais e níveis de adoção [6, 5, 2].

No questionário, as práticas são associadas aos estágios Organização Ágil, Integração Contínua, Implantação Contínua e Pesquisa e Desenvolvimento como Sistema de Inovação. As respostas são classificadas por níveis de adoção, como não adotada (*Not adopted*), abandonada (*Abandoned*), realizada no nível de projeto/produto (*Realized at project/product level*), realizada no nível de processo (*Realized at process level*) e institucionalizada (*Institutionalized*). Dessa forma, o instrumento fornece uma base estruturada para compreender como as práticas contínuas estão distribuídas na organização avaliada [2].

Os resultados produzidos pelo Zeppelin constituem a base de entrada para a ferramenta analítica desenvolvida neste trabalho. A partir deles, torna-se possível construir visões voltadas à leitura do diagnóstico por estágios, dimensões, práticas avaliadas, recomendações e à consulta histórica de ciclos. Assim, a forma de apresentação dos resultados também passa a influenciar a utilidade prática do diagnóstico, pois visualizações, agrupamentos e consultas históricas podem facilitar a interpretação dos dados e apoiar a definição de ações de melhoria.

A Figura 1 resume a relação entre a base conceitual de Engenharia de Software Contínua, os modelos utilizados pelo Zeppelin, os resultados do diagnóstico e a ferramenta analítica proposta neste trabalho.

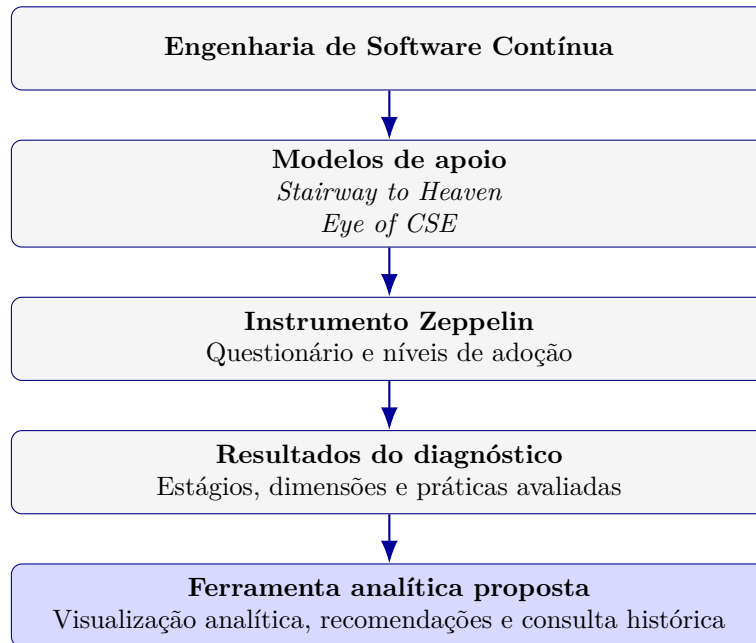


Figura 1: Relação conceitual entre Engenharia de Software Contínua, Zeppelin e a ferramenta analítica proposta. Fonte: elaboração própria, com base nos trabalhos sobre *Stairway to Heaven*, *Eye of CSE*, Zeppelin e recomendações associadas ao instrumento [6, 5, 2, 1].

3 Trabalhos Correlatos

Esta seção apresenta trabalhos diretamente relacionados à proposta deste Projeto Final de Graduação. Diferentemente da fundamentação teórica, que discute conceitos e modelos utilizados para compreender a área, o foco aqui está em estudos que utilizam o instrumento Zeppelin ou que propõem formas de interpretar seus resultados para apoiar a evolução de práticas contínuas em organizações de *software*.

Dos Santos Júnior et al. [2] utilizam o instrumento Zeppelin para obter uma visão panorâmica da adoção de práticas de Engenharia de Software Contínua em organizações brasileiras. O trabalho mostra o uso do Zeppelin como instrumento de diagnóstico organizacional, permitindo observar diferentes aspectos da maturidade em práticas contínuas. Sua contribuição para este projeto está na base diagnóstica utilizada: os resultados produzidos pelo Zeppelin são o ponto de partida para as análises apresentadas pela ferramenta. Este Projeto Final de Graduação não altera o instrumento nem redefine seus critérios de avaliação, mas atua sobre a forma como seus resultados são organizados e apresentados ao usuário.

A dissertação de Araújo [1] também se relaciona diretamente a este projeto, pois utiliza o Zeppelin como base para diagnóstico e recomendações voltadas a práticas de Integração Contínua e Implantação Contínua. Esse trabalho aproxima os resultados do instrumento de uma perspectiva mais prática, ao associar o diagnóstico a recomendações que podem orientar a evolução das organizações avaliadas. Neste Projeto Final de Graduação, essas informações

são organizadas e apresentadas em uma ferramenta analítica, implementada como interface *web*, para facilitar a leitura dos resultados, das recomendações e dos componentes específicos de interpretação histórica.

A Tabela 1 resume a relação entre os trabalhos considerados e a proposta deste Projeto Final de Graduação.

Trabalho	Foco principal	Relação com este projeto
dos Santos Júnior et al. [2]	Uso do Zeppelin para diagnosticar a adoção de práticas de Engenharia de Software Contínua em organizações brasileiras.	Fornece a base diagnóstica utilizada neste projeto. Este PFG atua após a aplicação do instrumento, organizando seus resultados em uma ferramenta analítica.
Araújo [1]	Geração de recomendações personalizadas a partir dos resultados do Zeppelin, com foco em Integração Contínua e Implantação Contínua.	Relaciona-se à interpretação prática dos resultados e às recomendações. Este projeto diferencia-se por concentrar-se na apresentação analítica dessas informações em uma interface <i>web</i> navegável.

Tabela 1: Comparação entre trabalhos relacionados e a proposta deste Projeto Final de Graduação.

A partir dessa comparação, observa-se que os trabalhos anteriores contribuem em etapas anteriores à proposta deste projeto. Dos Santos Júnior et al. [2] consolidam o Zeppelin como instrumento de diagnóstico, enquanto Araújo [1] associa os resultados do instrumento à geração de recomendações. Este Projeto Final de Graduação atua em uma etapa posterior e complementar: a construção de uma ferramenta analítica, implementada como interface *web*, para organizar e apresentar os resultados do Zeppelin de forma mais acessível, estruturada e útil ao planejamento de evolução das organizações avaliadas.

4 Concepção da Solução Proposta

A concepção da solução proposta parte das saídas do instrumento Zeppelin e define como organizar esses dados em uma ferramenta analítica. A solução foi pensada para apoiar tanto a coleta de dados quanto a leitura posterior do diagnóstico, reunindo, em um único fluxo, o vínculo com uma organização, a criação de ciclos de avaliação, o preenchimento do questionário e a consulta aos resultados analíticos.

Na prática, a ferramenta atua como uma camada de apoio ao uso do Zeppelin. Os dados do instrumento são preservados em sua estrutura original, enquanto a interface organiza esses resultados em painéis, tabelas, recomendações e recursos de consulta histórica por ciclos. Com isso, o usuário pode consultar diferentes ciclos de avaliação, observar variações nos níveis de adoção e explorar os resultados por estágios, dimensões e práticas avaliadas.

A concepção da solução foi dividida em três aspectos principais: requisitos e escopo, fluxo de uso e regras de diagnóstico e recomendações. Essa divisão permite separar o que a ferramenta deve oferecer ao usuário, como a interação ocorre ao longo do sistema e de que maneira os resultados do Zeppelin são tratados para fins de apresentação.

4.1 Requisitos e Escopo

A ferramenta foi concebida para apoiar o uso do Zeppelin em um ambiente *web*, reunindo funcionalidades de cadastro, vínculo com organizações, criação de ciclos de avaliação, preenchimento do questionário e consulta dos resultados. Entretanto, nem todas essas funcionalidades constituem a contribuição principal deste Projeto Final de Graduação. Parte delas compõe a infraestrutura necessária para o funcionamento da ferramenta e foi desenvolvida no contexto mais amplo do projeto.

No escopo deste trabalho, a contribuição principal concentra-se na camada analítica de apresentação dos resultados do Zeppelin. Essa camada inclui a organização dos resultados em painéis, visualizações detalhadas, recomendações e componentes específicos de interpretação histórica. Funcionalidades como autenticação, vínculo com organização, criação de ciclos, preenchimento do questionário, recursos de *benchmarking* e determinadas funcionalidades gerais de histórico são consideradas funcionalidades de suporte ou partes desenvolvidas no contexto do projeto por outros integrantes.

Dessa forma, esta seção apresenta tanto o escopo geral da ferramenta quanto a delimitação da contribuição deste trabalho. Essa distinção é importante porque o sistema faz parte de um contexto de desenvolvimento mais amplo, enquanto este Projeto Final de Graduação se concentra na construção e organização da camada analítica responsável por tornar os resultados do Zeppelin mais acessíveis e interpretáveis.

A Tabela 2 resume os principais requisitos funcionais considerados na concepção da ferramenta e indica sua relação com o escopo deste trabalho.

Requisito funcional	Descrição	Relação com este trabalho
Autenticação de usuário	Permitir que o usuário crie uma conta, realize login e acesse a ferramenta de forma autenticada.	Funcionalidade de suporte ao sistema.
Vínculo com organização	Permitir que o usuário se associe a uma organização existente ou cadastre uma nova organização para realizar avaliações.	Funcionalidade de suporte ao sistema.
Seleção da organização ativa	Permitir que o usuário escolha a organização utilizada como contexto para carregar ciclos, respostas e resultados analíticos.	Funcionalidade de suporte ao sistema.
Criação e retomada de ciclos	Permitir a criação de novos ciclos de avaliação, a consulta de ciclos anteriores e a retomada de ciclos ainda não concluídos.	Funcionalidade de suporte ao fluxo de avaliação.
Resposta guiada ao questionário	Permitir o preenchimento do questionário Zeppelin por meio de uma navegação estruturada, com salvamento das respostas ao longo do processo.	Funcionalidade de suporte à geração dos dados analisados.
Recursos de <i>benchmarking</i>	Permitir comparações complementares entre resultados, quando disponíveis, como apoio adicional à interpretação do diagnóstico.	Funcionalidade desenvolvida no contexto mais amplo do projeto.
Resumo diagnóstico geral	Apresentar uma visão geral do ciclo de avaliação, incluindo estado do questionário, cobertura das respostas e leitura inicial dos resultados.	Parte da camada analítica deste trabalho.
Visualização detalhada dos resultados	Organizar os resultados do diagnóstico por níveis de adoção, estágios, dimensões, processos, elementos e práticas avaliadas.	Parte central da contribuição deste trabalho.
Recomendações	Apresentar recomendações associadas aos resultados do diagnóstico, organizadas de forma a apoiar a priorização de ações de melhoria.	Parte da camada analítica deste trabalho.
Histórico de avaliações	Permitir a consulta de ciclos anteriores e apoiar a leitura da evolução dos resultados de uma mesma organização.	Contribuição parcial deste trabalho, limitada a componentes específicos de interpretação histórica.
Gestão básica de perfil e membros	Permitir operações de apoio, como edição de dados do usuário, visualização de membros vinculados e gerenciamento básico de vínculo com organizações.	Funcionalidade de suporte ao sistema.

Tabela 2: Principais requisitos funcionais considerados na concepção da ferramenta e sua relação com este trabalho.

Além dos requisitos funcionais, a concepção da ferramenta considera aspectos não funcionais relacionados à clareza da interface, organização visual dos dados e facilidade de navegação. Como os resultados do Zeppelin podem ser analisados por diferentes perspectivas, a interface deve evitar que o usuário dependa apenas de dados brutos ou tabelas extensas para interpretar o diagnóstico. Também é importante que a solução mantenha separadas as responsabilidades de coleta de respostas, processamento dos dados e apresentação das visualizações, favorecendo a manutenção e a evolução futura do sistema.

O escopo geral da ferramenta inclui a coleta das respostas, a organização por ciclos de avaliação, a visualização do diagnóstico geral e detalhado, a consulta a recomendações, recursos de *benchmarking* e a consulta histórica dos resultados por ciclo. No entanto, o escopo específico deste Projeto Final de Graduação está concentrado na apresentação analítica dos resultados produzidos pelo Zeppelin, especialmente por meio do painel, da tela de resultados, da organização das recomendações e de componentes específicos de interpretação histórica. Por outro lado, não fazem parte do escopo deste trabalho a criação de um novo instrumento diagnóstico, a alteração dos níveis de adoção do Zeppelin, a redefinição das regras conceituais do instrumento ou a validação de impacto real em organizações.

4.2 Fluxo de Uso

O fluxo de uso da ferramenta foi definido para conduzir o usuário desde o acesso inicial até a leitura dos resultados do diagnóstico. A navegação começa pela autenticação, passa pela definição do contexto organizacional e pela seleção de um ciclo de avaliação, chegando às telas voltadas à análise dos resultados, às recomendações e à consulta de ciclos anteriores.

Ao acessar a ferramenta, o usuário realiza a autenticação ou cria uma nova conta. Em seguida, deve estar vinculado a uma organização, seja por associação a uma organização já existente, seja pelo cadastro de uma nova organização. Essa etapa é necessária porque os ciclos de avaliação e os resultados analisados são tratados dentro de um contexto organizacional específico.

Com a organização definida, o usuário pode iniciar um novo ciclo de avaliação ou retomar um ciclo ainda não concluído. Neste trabalho, um ciclo de avaliação corresponde a uma execução do questionário Zeppelin associada a uma organização em um determinado momento. Cada ciclo reúne as respostas registradas naquela avaliação e permite que seus resultados sejam consultados posteriormente, inclusive para apoiar a comparação com avaliações realizadas em outros momentos.

O questionário é apresentado de forma guiada, permitindo que as respostas sejam registradas progressivamente ao longo do preenchimento. Após a conclusão do ciclo, os dados coletados passam a alimentar as visões analíticas disponibilizadas pela ferramenta. A partir dessas respostas, o usuário pode consultar o painel principal, visualizar resultados detalhados por níveis de adoção, estágios, dimensões, processos, elementos e práticas avaliadas, além de acessar recomendações associadas ao diagnóstico e consultar o histórico de ciclos.

Esse encadeamento mostra como o preenchimento do instrumento se conecta às telas de análise. O questionário, os ciclos de avaliação e as visões analíticas são organizados em um mesmo ambiente, permitindo que cada avaliação gere uma base consultável para interpretação dos resultados, apoio à leitura entre ciclos e planejamento de ações futuras.

A Figura 2 apresenta o fluxo principal de interação previsto para a ferramenta.

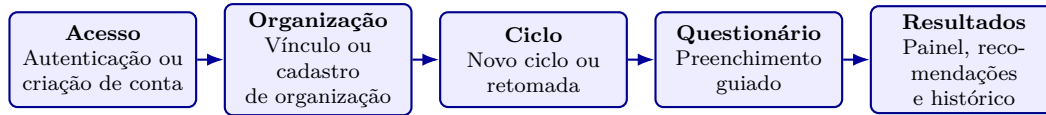


Figura 2: Fluxo principal de uso da ferramenta. Fonte: elaboração própria.

4.3 Regras de Diagnóstico e Recomendações

As regras utilizadas pela ferramenta partem das respostas registradas no questionário Zeppelin. Cada resposta é associada a um nível de adoção predefinido, e esses níveis possuem valores percentuais utilizados para compor as análises apresentadas ao usuário. A ferramenta trabalha com a estrutura do instrumento como entrada, organizando os dados coletados em visões que facilitam a interpretação do diagnóstico.

Os níveis de adoção considerados seguem a estrutura utilizada pelo Zeppelin: não adotada (*Not adopted*), abandonada (*Abandoned*), realizada no nível de projeto/produto (*Realized at project/product level*), realizada no nível de processo (*Realized at process level*) e institucionalizada (*Institutionalized*). Na ferramenta, esses níveis são convertidos em valores numéricos e utilizados em cálculos agregados. A pontuação de um conjunto de respostas é calculada a partir da mediana dos valores associados aos níveis de adoção, permitindo gerar leituras por ciclo de avaliação e por diferentes recortes analíticos.

Além dos níveis de adoção, a ferramenta considera o tipo de organização avaliada no processo de pontuação. Perfis como *startup*, organização com departamento de TI e organização cujo produto principal é *software* podem receber pesos distintos por meio de um mapeamento utilizado no *backend*. Essa diferenciação busca tornar a leitura do diagnóstico mais adequada ao contexto organizacional avaliado.

A organização dos resultados ocorre em diferentes níveis de análise. A ferramenta apresenta uma visão geral do ciclo, resultados por estágios de evolução, dimensões, processos e elementos. Essa divisão evita que o diagnóstico fique restrito a uma pontuação única, permitindo observar quais partes da organização apresentam maior consolidação e quais ainda possuem menor grau de adoção.

Para apoiar a interpretação dos resultados, a ferramenta identifica forças e gargalos com base no percentual bruto associado ao nível de adoção informado em cada prática. Práticas com percentual igual ou superior a 60% são consideradas indícios de maior consolidação, enquanto práticas com percentual inferior a 60% são tratadas como possíveis gargalos ou oportunidades de melhoria. Esse critério é aplicado diretamente sobre o percentual do nível de adoção, sem depender da pontuação calibrada pelo tipo de organização.

As recomendações são geradas principalmente a partir de práticas de Integração Contínua e Implantação Contínua com baixa adoção. Uma prática pode ser considerada candidata a recomendação quando pertence ao escopo de CI/CD e apresenta percentual inferior a 60%. A ferramenta organiza essas recomendações em trilhas de priorização, como adotar agora (*Adopt now*) e consolidar (*Consolidate*). Com isso, o diagnóstico passa a apoiar não

apenas a leitura do estado atual, mas também a identificação de possíveis próximos passos.

Em ciclos de avaliação ainda incompletos, a ferramenta também pode tratar respostas ausentes como valor zero em determinados cálculos do diagnóstico parcial. Esse comportamento permite apresentar uma leitura provisória do ciclo, mas reforça que resultados parciais devem ser interpretados com cautela até a conclusão do questionário.

A organização dessas regras permite transformar as respostas individuais do questionário em uma leitura mais estruturada do diagnóstico. O resultado final não se limita à exibição das respostas preenchidas, mas apresenta agrupamentos, destaques e recomendações que apoiam a análise da situação atual e o planejamento de ciclos futuros de melhoria.

A Tabela 3 resume as principais regras consideradas na organização dos resultados e recomendações.

Aspecto	Tratamento na ferramenta
Níveis de adoção	As respostas são associadas a níveis predefinidos, representados por valores percentuais usados nos cálculos do diagnóstico.
Resultados agregados	Os níveis são convertidos em valores numéricos e utilizados para gerar leituras agregadas por ciclo e por diferentes recortes analíticos.
Tipo de organização	A pontuação pode considerar pesos diferentes de acordo com o perfil da organização avaliada, como <i>startup</i> , organização com departamento de TI ou organização cujo produto principal é <i>software</i> .
Estágios de evolução	Os resultados são agrupados conforme os estágios considerados pelo Zeppelin, permitindo observar a evolução da organização em práticas de CSE.
Perspectivas complementares	As respostas também são organizadas por dimensões, processos e elementos, facilitando a identificação de áreas com maior ou menor adoção.
Forças	Práticas com percentual igual ou superior a 60% são tratadas como indícios de maior consolidação.
Gargalos	Práticas com percentual inferior a 60% são tratadas como possíveis fragilidades ou oportunidades de melhoria.
Recomendações	Recomendações são associadas principalmente a práticas de CI/CD com adoção inferior a 60%, sendo organizadas em trilhas de priorização.
Histórico de ciclos	Resultados de ciclos diferentes podem ser consultados para apoiar a leitura da evolução da organização ao longo do tempo.

Tabela 3: Principais regras consideradas na organização do diagnóstico e das recomendações.

5 Desenvolvimento da Ferramenta

5.1 Arquitetura da Solução

A arquitetura da ferramenta foi documentada conforme o modelo C4 em dois níveis. Primeiro, apresenta-se a organização dos contêineres da solução e, em seguida, os contêineres de *front-end* e *back-end* são detalhados por meio de diagramas de componentes.

A Figura 3 apresenta a arquitetura em nível de contêineres. O membro da organização utiliza o *front-end*, desenvolvido com *React* 18 e *Vite*, para responder aos diagnósticos e consultar os resultados. Por meio de uma *API REST* sobre HTTPS, o *front-end* comunica-se com o *back-end*, implementado com *Django* 4.1 e *Django REST Framework*. O *back-end* executa as regras de negócio e o processamento analítico, persiste as informações no *PostgreSQL* e consulta o catálogo de recomendações em formato *CSV*.

O detalhamento do *front-end* é apresentado na Figura 4. A Aplicação gerencia a navegação e o estado global, enquanto a Estrutura da Plataforma organiza as páginas autenticadas. Os demais componentes tratam a autenticação, o preenchimento dos diagnósticos e o gerenciamento do perfil e da organização.

Em azul, destacam-se o Módulo de Apresentação Analítica, a Página de Recomendações e os Serviços HTTP, desenvolvidos ou adaptados total ou parcialmente neste trabalho. A contribuição abrangeu a apresentação dos resultados, as recomendações e a integração dessas funcionalidades com a *API REST*. Na área de histórico, foram desenvolvidos apenas alguns componentes. Embora o *benchmark* também pertença ao Módulo de Apresentação Analítica, essa funcionalidade não foi desenvolvida no escopo deste trabalho. O *back-end* é representado fora da caixa tracejada como um único contêiner externo.

O detalhamento do *back-end* é apresentado na Figura 5. O Módulo de Autenticação gerencia usuários, sessões e permissões, relacionando-se com os módulos de Organização e de Colaboradores. O Módulo do Questionário controla afirmativas, diagnósticos e respostas, além de solicitar o processamento dos resultados e relacionar as afirmativas aos modelos conceituais.

O Serviço Analítico do Questionário e os Modelos Conceituais, destacados em azul, representam as principais contribuições deste trabalho no *back-end*. O primeiro calcula pontuações, compara ciclos e seleciona recomendações. O segundo representa as taxonomias *Stairway to Heaven*, *Eye of CSE*, *Continuous Star* e *CSE Framework*.

O *front-end*, o *PostgreSQL* e o Catálogo de Recomendações aparecem como contêineres externos. O *front-end* acessa os componentes pela *API REST*, o *PostgreSQL* mantém os dados persistentes e o catálogo em formato *CSV* fornece as recomendações associadas às afirmativas.

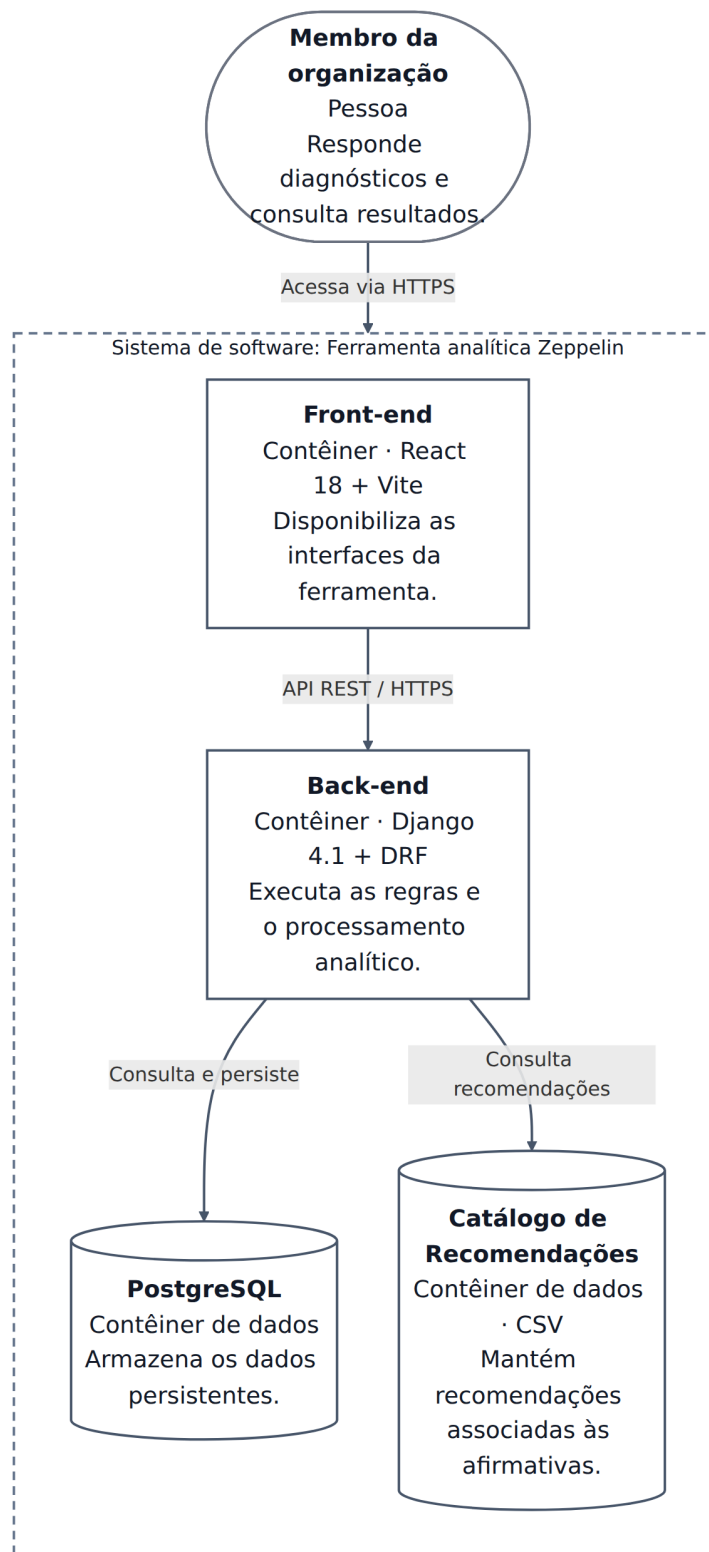


Figura 3: Diagrama de contêineres da ferramenta analítica Zeppelin. Fonte: elaboração da equipe do projeto.

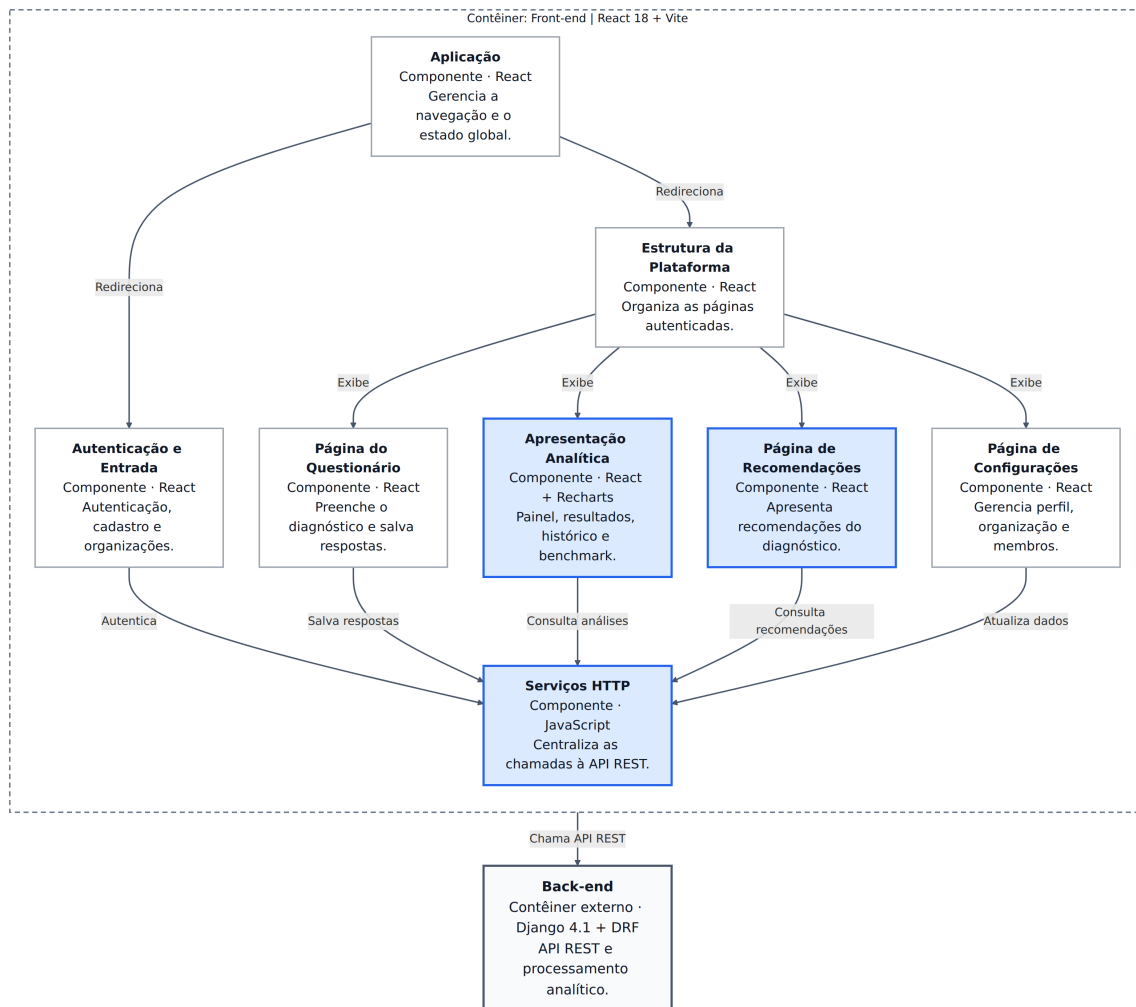


Figura 4: Componentes do contêiner de *front-end*. Em azul, os componentes desenvolvidos ou adaptados neste trabalho. Fonte: elaboração da equipe do projeto.

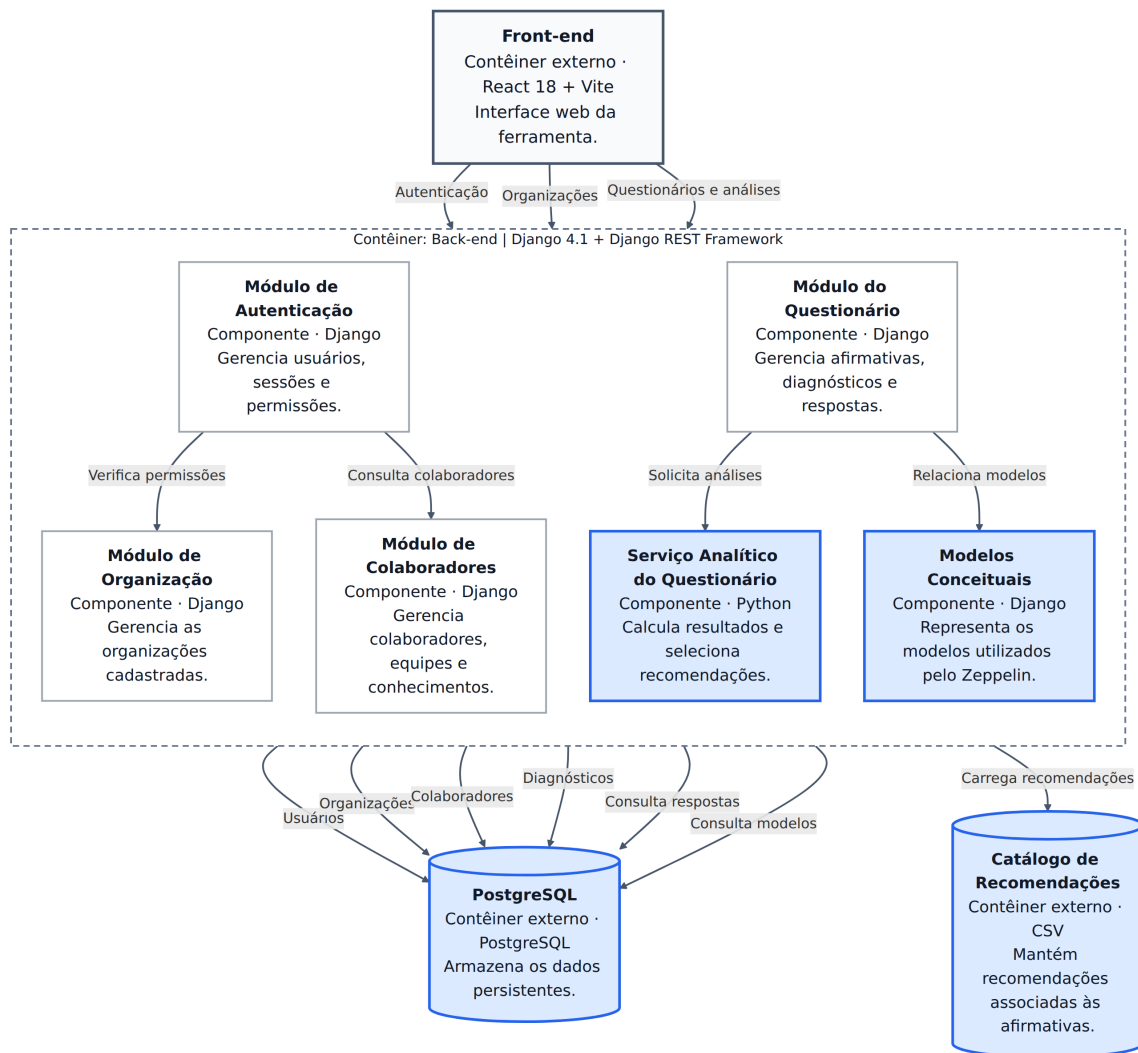


Figura 5: Componentes do contêiner de *back-end* e suas relações externas. Em azul, os elementos desenvolvidos ou adaptados neste trabalho. Fonte: elaboração da equipe do projeto.

5.2 Tecnologias Utilizadas

O desenvolvimento da ferramenta utilizou um conjunto de tecnologias voltadas à construção de uma solução *web* com separação entre a interface, a *API* e a persistência de dados. Essa organização permitiu dividir o sistema em uma camada de apresentação (*frontend*), responsável pela interação com o usuário, e uma camada de *backend*, responsável pelo processamento das respostas e pela geração de dados analíticos.

No *frontend*, a ferramenta utiliza a biblioteca *React* para a construção das páginas e dos componentes da interface. Essa tecnologia foi utilizada para estruturar as telas analíticas, controlar estados locais, organizar componentes reutilizáveis e apresentar os resultados ao usuário. O projeto também utiliza *Vite* como ferramenta de desenvolvimento e *build*, facilitando a execução local da ferramenta e o empacotamento do *frontend*.

A apresentação visual dos resultados combina componentes em *React*, *CSS* próprio e recursos gráficos implementados diretamente na interface. Algumas visualizações foram construídas usando *SVG* personalizado, permitindo representar dados analíticos de forma adaptada às necessidades da ferramenta. Bibliotecas auxiliares também são utilizadas para elementos visuais, como ícones e componentes de navegação.

No *backend*, a ferramenta utiliza *Django* como *framework* principal. Essa camada concentra os modelos de dados, a integração com o banco de dados e as regras responsáveis pela geração dos resultados analíticos. Para a exposição das informações ao *frontend*, o projeto utiliza *Django REST Framework*, permitindo a criação de *endpoints REST* consumidos pelas telas da ferramenta.

A camada analítica foi implementada principalmente com *Django ORM* e recursos da linguagem *Python*. As respostas registradas no questionário são consultadas no banco de dados, agrupadas conforme o contexto da análise e processadas para gerar indicadores, distribuições, recomendações e dados históricos por ciclo. Esse processamento ocorre no *backend*, mantendo o *frontend* concentrado na apresentação dos dados já organizados.

O banco de dados principal previsto para a ferramenta é o *PostgreSQL*, utilizado para persistir informações como usuários, organizações, ciclos de avaliação, respostas, afirmações do instrumento e níveis de adoção. O projeto também possui suporte a *SQLite* em ambiente local de desenvolvimento. Para a organização do ambiente de execução, utilizam-se *Docker* e *Docker Compose*, o que permite separar em contêineres distintos o *frontend*, o *backend* e o banco de dados.

A Tabela 4 resume as principais tecnologias utilizadas no projeto e seus respectivos papéis na ferramenta.

Camada	Tecnologias	Papel na ferramenta
<i>Frontend</i>	<i>React, Vite, CSS e SVG</i>	Construção das telas, componentes de interface, visualizações customizadas e empacotamento do <i>frontend</i> .
<i>Backend e API</i>	<i>Django e Django REST Framework</i>	Estruturação dos modelos, regras de negócio e <i>endpoints REST</i> consumidos pelo <i>frontend</i> .
Processamento analítico	<i>Python e Django ORM</i>	Consulta das respostas armazenadas e implementação das regras de agregação, cálculo e organização dos resultados.
Persistência de dados	<i>PostgreSQL e SQLite</i>	Armazenamento das informações da ferramenta, com <i>PostgreSQL</i> como banco de dados principal e <i>SQLite</i> como alternativa local de desenvolvimento.
Ambiente de execução	<i>Docker e Docker Compose</i>	Organização dos serviços da ferramenta, separando <i>frontend</i> , <i>backend</i> e banco de dados durante a execução.
Suporte da ferramenta	Autenticação, documentação de <i>API</i> e configuração de ambiente	Recursos de apoio ao funcionamento geral do sistema, sem compor diretamente o núcleo da camada analítica.

Tabela 4: Principais tecnologias utilizadas no desenvolvimento da ferramenta.

Entre essas tecnologias, as mais diretamente relacionadas à contribuição deste trabalho são *React*, *Django*, *Django REST Framework*, *Django ORM*, *Python* e *PostgreSQL*. Essas ferramentas sustentam o fluxo principal da camada analítica, desde a consulta das respostas registradas até a apresentação dos resultados, recomendações e componentes de interpretação histórica na interface.

5.3 Estrutura Analítica dos Resultados

A estrutura analítica dos resultados foi construída com base nas respostas registradas no questionário Zeppelin. Cada resposta representa a associação entre uma organização, um ciclo de avaliação, uma afirmação do instrumento e um nível de adoção. A partir dessa estrutura, a ferramenta transforma respostas individuais em recortes agregados, permitindo que o diagnóstico seja analisado sob diferentes perspectivas.

Os principais dados de entrada da camada analítica são as respostas do questionário, os ciclos de avaliação, as afirmações avaliadas e os níveis de adoção. As afirmações estão relacionadas à estrutura do instrumento, incluindo estágios, dimensões, processos e elementos de modelos teóricos sobre Engenharia de Software Contínua (Seção 2). Com isso, uma resposta registrada pelo usuário deixa de ser apenas um valor isolado e passa a contribuir para diferentes leituras do diagnóstico.

Os níveis de adoção utilizados pela ferramenta seguem a escala do Zeppelin: não adotada (*Not adopted*), abandonada (*Abandoned*), realizada no nível de projeto/produto (*Realized at project/product level*), realizada no nível de processo (*Realized at process level*) e institucionalizada (*Institutionalized*). Esses níveis possuem percentuais associados, respectivamente, a 0, 10, 30, 60 e 100, conforme descrito em [1]. Além desses valores, a ferramenta também considera pesos calibrados conforme o tipo de organização avaliada, permitindo ajustar a

pontuação para contextos como *startups*, organizações cujo produto principal é *software* e organizações com departamento de TI.

O cálculo principal das pontuações utiliza a mediana dos pesos associados às respostas, e não uma média simples. Essa escolha reduz o efeito de valores extremos e resulta em uma leitura agregada mais estável do conjunto de respostas, dado que a escala original é ordinal. Em alguns cálculos, quando o total esperado de práticas é informado e o ciclo ainda está incompleto, respostas ausentes podem ser tratadas como zero. Esse comportamento permite gerar leituras parciais do diagnóstico, mas exige cuidado na interpretação de ciclos ainda não concluídos.

Com base nessas regras, a ferramenta organiza os resultados em diferentes recortes analíticos. A pontuação geral resume a situação do ciclo selecionado. Os resultados por estágio permitem observar a evolução da organização em relação aos estágios considerados pelo Zeppelin. A distribuição por níveis de adoção indica quantas práticas foram classificadas em cada nível. Já os recortes por dimensões, processos e elementos permitem detalhar o diagnóstico em partes menores da estrutura do instrumento.

Além das pontuações agregadas, a estrutura analítica também fornece informações auxiliares para apoiar a interpretação do diagnóstico. Entre elas estão a identificação de forças e gargalos, a seleção de práticas candidatas a recomendações e a organização de dados históricos por ciclo de avaliação. Essas informações são geradas a partir das mesmas respostas do questionário, mas são apresentadas em telas específicas da ferramenta, descritas na subseção seguinte.

No caso das forças e gargalos, a ferramenta utiliza o percentual bruto do nível de adoção da prática. Práticas com percentual igual ou superior a 60% são tratadas como indícios de maior consolidação, enquanto práticas com percentual inferior a 60% são consideradas possíveis fragilidades ou oportunidades de melhoria. Esse critério não depende diretamente da pontuação calibrada pelo tipo de organização.

As recomendações e as informações históricas também derivam da estrutura analítica, mas não são detalhadas nesta subseção para evitar sobreposição com a descrição da interface. Em termos gerais, as recomendações partem de práticas com baixa adoção, especialmente no escopo de CI/CD, enquanto os dados históricos são organizados por ciclos de avaliação de uma mesma organização. A forma como essas informações são apresentadas ao usuário é descrita na Subseção 5.4.

Essa estrutura permite que os resultados do Zeppelin sejam apresentados de forma mais interpretável. Em vez de exibir apenas respostas individuais, a ferramenta organiza o diagnóstico em pontuações, agrupamentos e recortes analíticos. Com isso, o usuário pode compreender melhor o estado atual da organização e identificar pontos que merecem atenção em ciclos futuros.

5.4 Interface e Funcionalidades

A interface da ferramenta foi organizada para transformar os dados analíticos gerados pelo *backend* em telas de leitura e interpretação dos resultados. No escopo deste trabalho, o foco está nas funcionalidades do painel, da tela de resultados, das recomendações e dos componentes específicos de interpretação histórica. Outras partes do sistema, como auten-

ticação, cadastro, vínculo com a organização e preenchimento do questionário, compõem o fluxo geral da ferramenta, mas atuam principalmente como suporte à geração dos dados analisados.

5.4.1 Painel

O painel funciona como a tela inicial de leitura do diagnóstico. Seu objetivo é apresentar uma visão resumida do ciclo de avaliação selecionado, ou seja, da avaliação realizada para uma organização em um determinado momento. Com isso, o usuário pode compreender rapidamente o estado geral da avaliação antes de acessar análises mais detalhadas.

Na versão atual da ferramenta, essa tela apresenta informações sintéticas sobre o ciclo selecionado, como o estado do questionário, o progresso das respostas e a cobertura dos estágios avaliados. Quando o ciclo está completo, o painel também exibe uma tabela sintética com as pontuações por estágio e a pontuação geral do ciclo. Caso o ciclo ainda esteja em andamento, a tela indica que é necessário concluir a avaliação para visualizar as pontuações.

Essa separação permite que o painel atue como uma visão inicial do ciclo, enquanto a aba de resultados concentra a análise mais detalhada e navegável das pontuações e demais recortes do diagnóstico.

A Figura 6 apresenta o painel da ferramenta, com os cartões de resumo e a tabela sintética de pontuações por estágio.

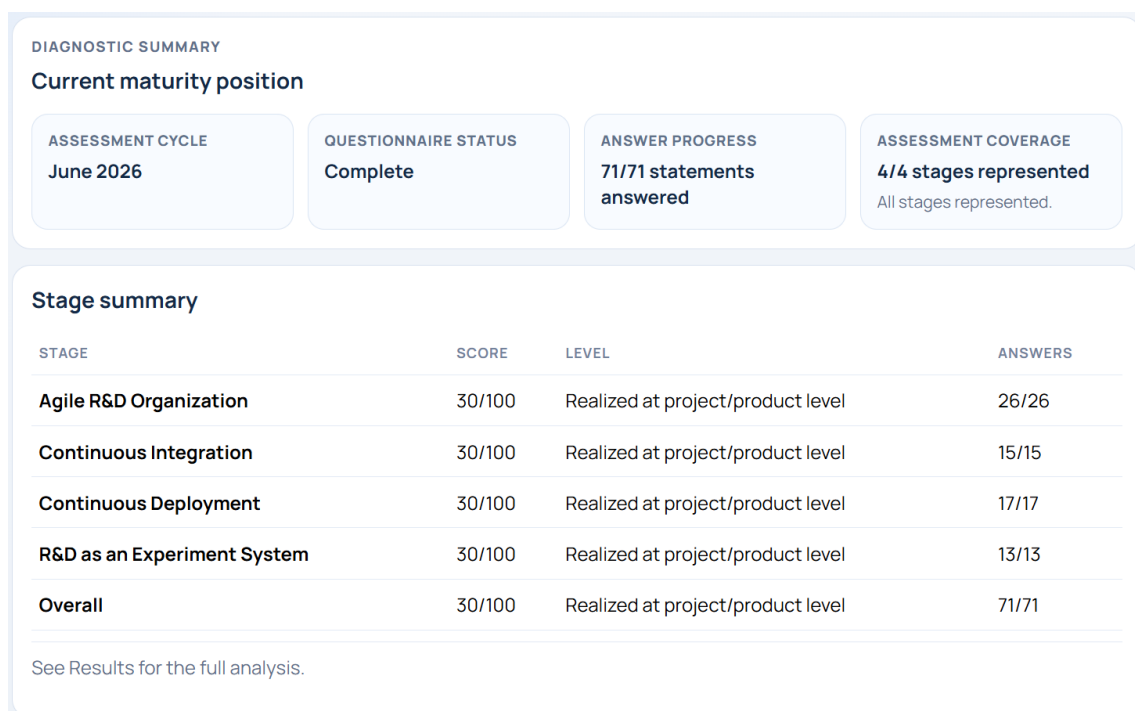


Figura 6: Painel com resumo do ciclo de avaliação e tabela sintética de pontuações por estágio. Fonte: elaboração própria.

5.4.2 Resultados

A tela de resultados é responsável por apresentar o diagnóstico sob diferentes perspectivas analíticas. Enquanto o painel oferece uma visão resumida do ciclo, a aba de resultados concentra as pontuações finais e os recortes detalhados da avaliação, permitindo explorar os dados por pontuação geral, estágios, grupos de dimensões, níveis de adoção, dimensões, processos e elementos. Esses recortes retomam a estrutura conceitual apresentada na Subseção 2.3, na qual os estágios estão associados ao modelo *Stairway to Heaven* e as dimensões ao *Eye of CSE*, ambos utilizados pelo Zeppelin como referência para organização do diagnóstico.

A primeira parte da tela apresenta um detalhamento diagnóstico do ciclo selecionado. Esse bloco inclui uma síntese textual do resultado, informações de estado e a dispersão entre estágios, que representa a diferença entre a maior e a menor pontuação obtida entre os estágios avaliados. Com isso, o usuário pode compreender o estado geral da avaliação antes de navegar pelos recortes específicos.

Em seguida, a tela exibe a seção de análise de pontuação, apresentada na interface como *Score analysis*. Esse componente não apresenta todas as pontuações simultaneamente, mas oferece um seletor que permite alternar entre diferentes tipos de análise: pontuação geral, pontuações por estágio e pontuações por grupos de dimensões. No caso dos estágios, a interface apresenta cartões com o nível de maturidade, a pontuação obtida, o nível correspondente e a quantidade de respostas consideradas.

A Figura 7 apresenta a análise de pontuações por estágio, uma das visualizações disponíveis na seção de análise de pontuação.

Outro recorte importante da tela é a distribuição dos níveis de adoção entre as afirmações avaliadas. A interface mostra quantas práticas foram classificadas em cada nível, como não adotadas (*Not adopted*), abandonadas (*Abandoned*), realizadas no nível de projeto/produto (*Realized at project/product level*), realizadas no nível de processo (*Realized at process level*) e institucionalizadas (*Institutionalized*). Além da distribuição geral, a ferramenta também cruza esses níveis com os estágios de evolução utilizados pelo Zeppelin, derivados do modelo *Stairway to Heaven*. Com isso, torna-se possível observar como a adoção se distribui ao longo da trajetória de evolução considerada pelo instrumento.

A Figura 8 apresenta esse recorte, combinando uma tabela de distribuição com uma visualização gráfica por estágio.

A tela também apresenta uma visão geral por dimensão. Conforme discutido na Subseção 2.3, as dimensões estão associadas ao *Eye of CSE* e são utilizadas pelo Zeppelin como uma forma complementar de analisar a adoção de práticas contínuas. Nesse bloco, as dimensões são organizadas em uma matriz, com a distribuição das práticas pelos estágios de evolução e a quantidade de práticas associadas a cada dimensão. Ao lado da matriz, a interface apresenta um radar de adoção por dimensão, que oferece uma leitura visual comparativa sobre quais dimensões apresentam maior ou menor desempenho relativo.

A Figura 9 apresenta essa visão por dimensões, incluindo a matriz e o radar de adoção.

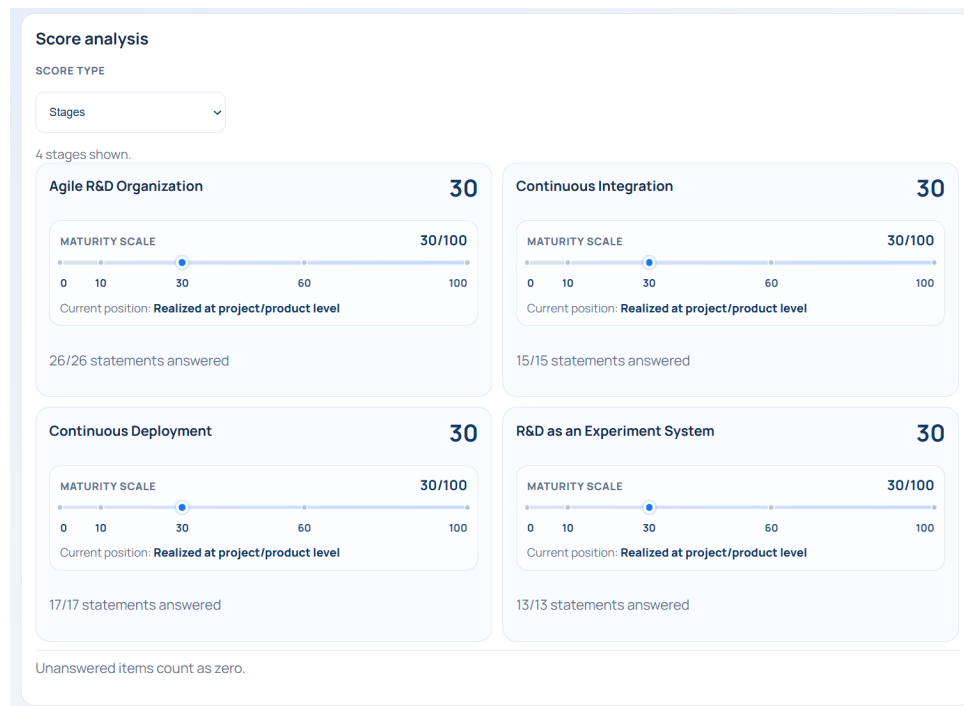


Figura 7: Análise de pontuações por estágio na tela de resultados. Fonte: elaboração própria.

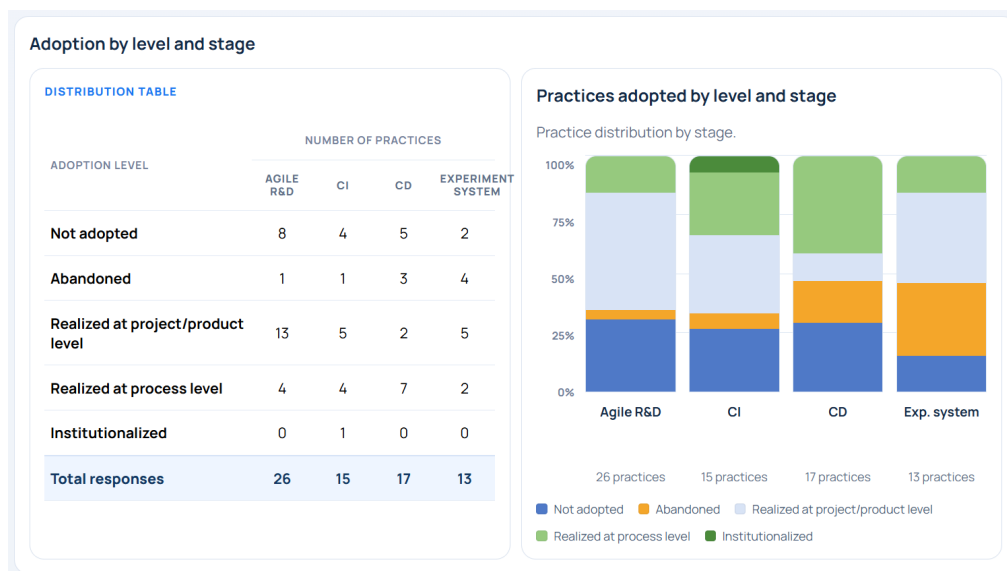


Figura 8: Distribuição dos níveis de adoção por estágio de evolução utilizado no instrumento. Fonte: elaboração própria.



Figura 9: Visão geral de adoção por dimensão, com matriz de distribuição e radar comparativo. Fonte: elaboração própria.

Além dos recortes apresentados nas figuras principais, a tela de resultados também inclui outras visualizações complementares, como detalhamento diagnóstico, pontuação geral, pontuações por grupos de dimensões, distribuição geral dos níveis de adoção, adoção por processo, adoção por elemento e filtros para selecionar dimensões ou elementos específicos. Esses recursos permitem aprofundar a análise quando o usuário deseja investigar partes mais específicas do diagnóstico.

Para evitar excesso de figuras no corpo principal do relatório, as demais capturas da tela de resultados são apresentadas no Apêndice A. No texto principal, foram mantidas apenas as três visualizações mais representativas da aba de resultados: análise de pontuações por estágio, adoção por nível e estágio, e visão geral por dimensão.

5.4.3 Recomendações

A tela de recomendações organiza práticas de baixa adoção em grupos de prioridade. O objetivo é transformar parte do diagnóstico em uma leitura mais acionável, indicando práticas que podem ser adotadas ou fortalecidas em ciclos futuros de melhoria.

Na versão atual da ferramenta, a própria interface exibe a regra utilizada para interpretar as recomendações. Práticas em 0% ou 10%, associadas aos níveis não adotada (*Not adopted*) e abandonada (*Abandoned*), são classificadas como adotar agora (*Adopt now*). Práticas em 30%, associadas ao nível realizada no nível de projeto/produto (*Realized at project/product level*), são classificadas como consolidar (*Consolidate*). Práticas em 60% ou 100% são consideradas mais consolidadas e não entram como prioridade nessa lista.

A interface apresenta a quantidade total de recomendações geradas, a quantidade de

práticas a adotar e a quantidade de práticas a consolidar. Também permite filtrar as recomendações por estágio e visualizar, para cada prática, o código da afirmação, o texto associado, o grupo de dimensão e a orientação vinculada à recomendação.

A Figura 10 mostra a tela de recomendações, com a separação entre práticas a adotar e a consolidar.

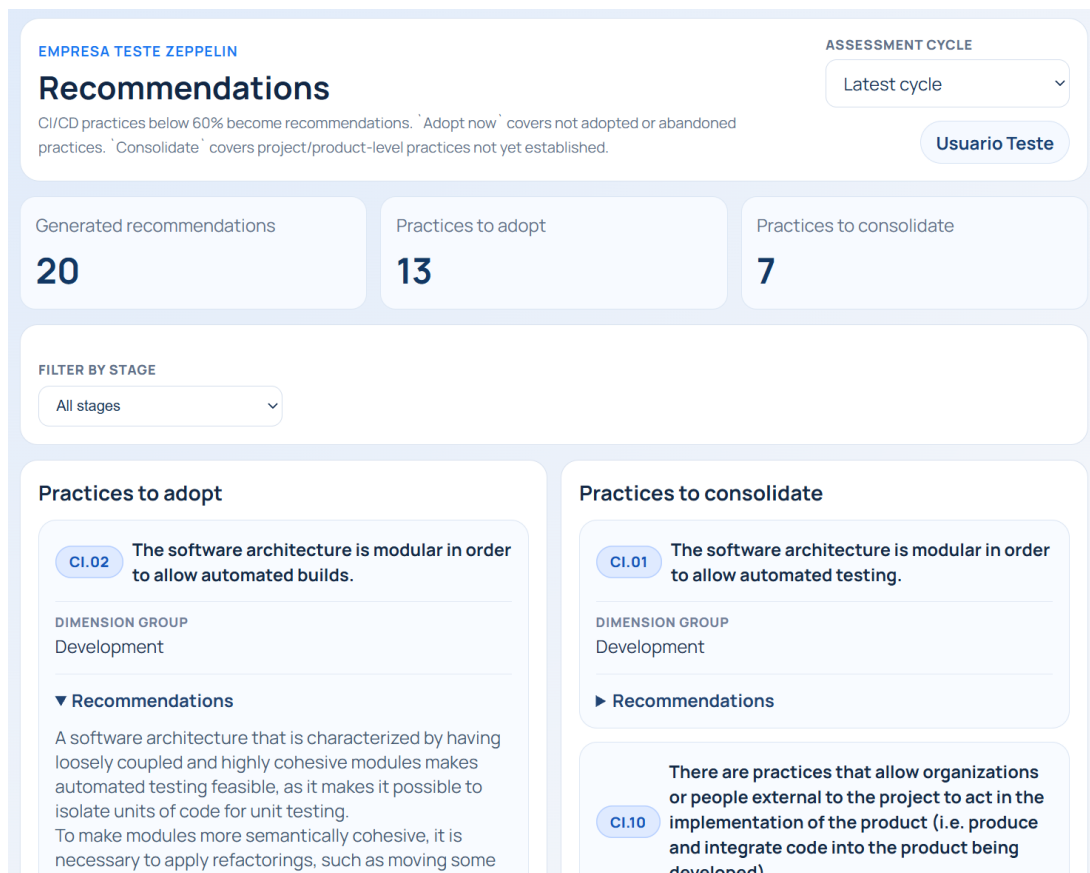


Figura 10: Tela de recomendações organizada em práticas a adotar e práticas a consolidar. Fonte: elaboração própria.

5.4.4 Interpretação histórica

A funcionalidade de histórico permite consultar ciclos anteriores e apoiar a leitura de mudanças entre avaliações de uma mesma organização. No escopo deste trabalho, a contribuição relacionada ao histórico concentrou-se em componentes específicos de interpretação, voltados a apresentar, de forma resumida, a evolução dos resultados entre ciclos.

Atualmente, esses componentes incluem cartões com as pontuações de cada ciclo e uma tabela com a distribuição dos níveis de adoção ao longo das avaliações registradas. Os cartões apresentam a pontuação geral e as pontuações por estágio de cada ciclo, permitindo comparar visualmente a situação de diferentes avaliações. Já a tabela de níveis de adoção

mostra como as respostas se distribuem entre categorias como não adotada, abandonada, realizada no nível de projeto/produto, realizada no nível de processo e institucionalizada.

Esses recursos ajudam o usuário a perceber mudanças gerais nos resultados, como o aumento ou a redução de práticas em determinados níveis de adoção. No entanto, essa funcionalidade deve ser entendida como apoio inicial à interpretação histórica, e não como uma análise estatística avançada, uma comparação completa entre ciclos ou uma validação de impacto organizacional.

A Figura 11 apresenta os componentes de histórico utilizados para apoiar a leitura das pontuações e da distribuição dos níveis de adoção entre ciclos.

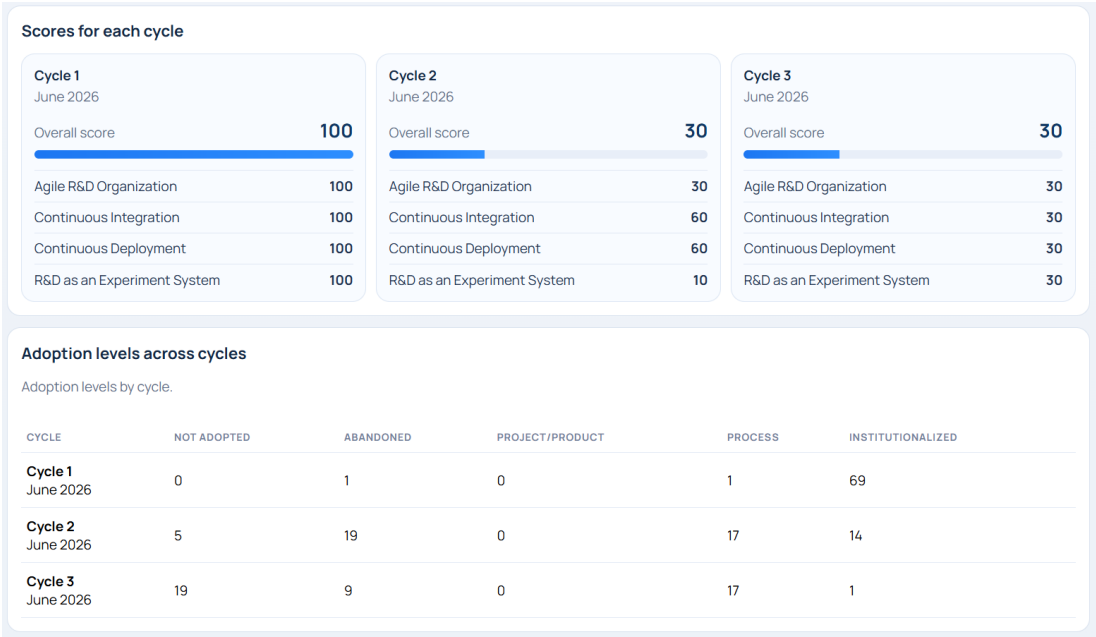


Figura 11: Componentes de interpretação histórica com pontuações por ciclo e distribuição dos níveis de adoção entre avaliações. Fonte: elaboração própria.

As funcionalidades descritas nesta subseção formam a camada de interface responsável por tornar os resultados do Zeppelin mais acessíveis ao usuário. O painel oferece uma visão inicial do ciclo; a tela de resultados concentra os recortes analíticos detalhados; a tela de recomendações organiza práticas que merecem atenção; e os componentes históricos oferecem uma leitura resumida das pontuações e da distribuição dos níveis de adoção entre os ciclos. Em conjunto, essas telas reforçam o objetivo da ferramenta de transformar os dados do instrumento em informações mais claras e úteis para análise.

5.5 Disponibilização do código-fonte

O código-fonte da ferramenta desenvolvida neste Projeto Final de Graduação está disponível em repositório eletrônico, permitindo a consulta à implementação, à organização dos componentes e aos artefatos técnicos produzidos durante o desenvolvimento:

<https://github.com/brenorobazza/zeppelin>

6 Discussão

O desenvolvimento da ferramenta permitiu transformar os resultados do instrumento Zeppelin em uma interface mais organizada para a leitura e a interpretação do diagnóstico. Em vez de apresentar apenas respostas individuais ou uma pontuação geral, a solução estrutura os dados em diferentes perspectivas, como visão geral do ciclo, resultados por níveis de adoção, estágios, dimensões, recomendações e componentes de interpretação histórica.

Um dos principais ganhos da ferramenta está na separação entre a coleta de respostas, o processamento analítico e a apresentação visual. O *backend* concentra as regras de agregação, cálculo e organização dos dados, enquanto o *frontend* consome essas informações e as apresenta em telas específicas. Essa divisão favorece a manutenção da solução e evita que a interface assuma a responsabilidade pela lógica principal do diagnóstico.

A tela de painel contribui para uma leitura inicial do ciclo avaliado, permitindo observar rapidamente o estado do questionário, o progresso das respostas e a posição geral da organização nos estágios considerados. Já a tela de resultados detalhados amplia essa leitura ao organizar o diagnóstico em recortes complementares. Essa organização reduz a dependência de uma única métrica e permite ao usuário identificar diferenças entre estágios e dimensões.

A tela de recomendações também contribui para tornar o diagnóstico mais acionável. Ao agrupar práticas de menor adoção em trilhas como adotar agora (*Adopt now*) e consolidar (*Consolidate*), a ferramenta aproxima os resultados do instrumento de possíveis ações de melhoria. Ainda assim, essas recomendações não devem ser interpretadas como prescrições definitivas. Elas dependem das regras implementadas, do escopo considerado e da qualidade das respostas fornecidas no questionário.

Outro ponto relevante diz respeito aos componentes de interpretação histórica. Esses recursos ajudam o usuário a observar, de forma resumida, mudanças entre ciclos de avaliação, principalmente por meio das pontuações de cada ciclo e da distribuição dos níveis de adoção ao longo das avaliações. No entanto, essa funcionalidade deve ser entendida como apoio inicial à leitura histórica dos resultados, e não como uma análise estatística avançada, uma comparação completa entre ciclos ou uma validação de impacto organizacional.

Apesar dos avanços obtidos, algumas limitações devem ser consideradas. A ferramenta depende diretamente da estrutura e da qualidade dos dados gerados pelo Zeppelin. Caso as respostas estejam incompletas, inconsistentes ou representem apenas uma visão parcial da organização, os resultados apresentados também devem ser interpretados com cautela.

Outra limitação está relacionada à validação do artefato. Este trabalho avalia principalmente o funcionamento da ferramenta e a forma como os resultados são organizados e apresentados. Não foram medidos impactos práticos em organizações reais, como melhoria de maturidade, redução de falhas, aumento de eficiência ou mudança efetiva nos processos de desenvolvimento. Portanto, a contribuição deve ser entendida como uma melhoria na apresentação e interpretação dos resultados do Zeppelin, e não como comprovação de impacto organizacional.

Também é importante destacar que a ferramenta não propõe um novo instrumento di-

agnóstico nem altera a estrutura conceitual do Zeppelin. Os níveis de adoção, os estágios, as dimensões e as práticas avaliadas continuam a ser definidos pelo instrumento. A contribuição deste trabalho está na construção de uma camada analítica e visual que torna esses resultados mais acessíveis, navegáveis e úteis para apoiar a análise.

De forma geral, a ferramenta atende ao objetivo proposto ao organizar os resultados do Zeppelin em uma interface *web* com diferentes níveis de leitura. O artefato desenvolvido contribui para aproximar o diagnóstico de uma experiência de uso mais prática, permitindo que o usuário consulte o estado geral da avaliação, explore recortes detalhados, visualize recomendações e acompanhe informações históricas por ciclo. Ainda assim, a evolução da solução depende de validações futuras com usuários e de melhorias incrementais nas funcionalidades analíticas.

7 Conclusão e Trabalhos Futuros

Este Projeto Final de Graduação teve como objetivo desenvolver uma ferramenta analítica para a apresentação dos resultados do instrumento Zeppelin. A proposta concentrou-se em tornar os resultados do diagnóstico mais acessíveis e interpretáveis, organizando-os em uma interface *web* com diferentes níveis de leitura.

No escopo deste trabalho, foram desenvolvidas principalmente as telas e os componentes da camada analítica, com destaque para o painel, os resultados detalhados, as recomendações e os componentes específicos de interpretação histórica. Em conjunto, essas funcionalidades permitem que o usuário consulte o estado geral de um ciclo de avaliação, explore os resultados por diferentes recortes analíticos, visualize práticas recomendadas e receba apoio para interpretar as mudanças entre ciclos de avaliação.

A ferramenta desenvolvida preserva a estrutura do instrumento e atua na forma como seus resultados são apresentados ao usuário. Em vez de depender apenas de respostas individuais ou de tabelas pouco contextualizadas, a ferramenta organiza os dados em painéis, gráficos, tabelas e componentes de apoio à interpretação. Com isso, o diagnóstico pode ser analisado de forma mais clara, favorecendo a identificação de pontos fortes, fragilidades e possíveis prioridades de evolução.

Apesar dos resultados obtidos, o trabalho apresenta limitações. A ferramenta depende da qualidade e da completude das respostas registradas no questionário, e ciclos incompletos podem gerar leituras parciais. Além disso, a camada analítica depende de regras e agregações pré-definidas pelo instrumento e pela implementação atual, sem substituir a interpretação especializada dos resultados. Este trabalho também não realizou validação empírica com organizações reais ou usuários finais, de modo que não são afirmados impactos práticos como aumento de maturidade, redução de falhas ou melhoria efetiva nos processos de desenvolvimento.

Como trabalhos futuros, uma primeira possibilidade é realizar avaliações com usuários, incluindo pesquisadores, consultores ou representantes de organizações que utilizem o Zeppelin. Essa avaliação poderia investigar se as visualizações propostas facilitam a compreensão do diagnóstico e apoiam a priorização de ações de melhoria.

Também podem ser exploradas melhorias na apresentação das recomendações, como

filtros mais refinados, explicações mais detalhadas sobre a origem de cada recomendação e mecanismos para acompanhar o estado das ações sugeridas. O uso de técnicas de inteligência artificial pode ser investigado futuramente como apoio à explicação ou à priorização de recomendações, sem fazer parte da implementação atual.

Outro ponto de evolução está relacionado à segurança e à privacidade da ferramenta, especialmente por ela lidar com respostas associadas a organizações e ciclos de avaliação. Embora o sistema utilize autenticação e organize os dados por usuário, organização e ciclo, trabalhos futuros poderiam aprofundar os mecanismos de controle de acesso, a validação de permissões por organização, a proteção de variáveis sensíveis, a auditoria de *endpoints* e os testes de segurança.

Por fim, a ferramenta pode ser estendida com funcionalidades de exportação de relatórios, integração com bases de dados externas e importação mais flexível de questionários. Essas extensões poderiam ampliar o uso prático do sistema e apoiar novas pesquisas sobre a adoção de práticas de Engenharia de Software Contínua.

De modo geral, o trabalho atingiu seu objetivo ao desenvolver uma ferramenta analítica capaz de organizar e apresentar os resultados do Zeppelin em uma camada mais clara e navegável. A contribuição principal reside na forma como os dados do instrumento são estruturados e disponibilizados ao usuário, o que apoia a interpretação do diagnóstico e o planejamento de ações de melhoria contínua em organizações de *software*.

Agradecimentos

Agradeço ao professor Breno Bernard Nicolau de França pela orientação ao longo deste Projeto Final de Graduação, pelo acompanhamento do desenvolvimento do trabalho e pelos direcionamentos na definição do escopo, da estrutura do relatório e da ferramenta desenvolvida.

Agradeço também à Alline Dias de Araújo, cuja dissertação de mestrado contribuiu para a compreensão do uso do instrumento Zeppelin no contexto de diagnóstico e recomendações, especialmente em práticas relacionadas a CI/CD. Suas contribuições e *feedbacks* ajudaram na compreensão da proposta e na organização das ideias relacionadas à ferramenta.

Também agradeço ao Paulo Sérgio Santos Júnior pelas contribuições associadas ao instrumento Zeppelin, por sua tese de doutorado, que serviu como referência importante para este trabalho, e pelos *feedbacks* fornecidos durante o desenvolvimento da proposta.

Por fim, agradeço a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.

Apêndice A

Capturas Complementares da Tela de Resultados

Este apêndice apresenta capturas complementares da tela de resultados da ferramenta. As figuras reunidas aqui mostram recortes adicionais da interface que não foram incluídos no corpo principal do relatório, como detalhamento diagnóstico, pontuação geral, pontuações por grupos de dimensões, distribuição geral dos níveis de adoção, adoção por processo e adoção por elemento.

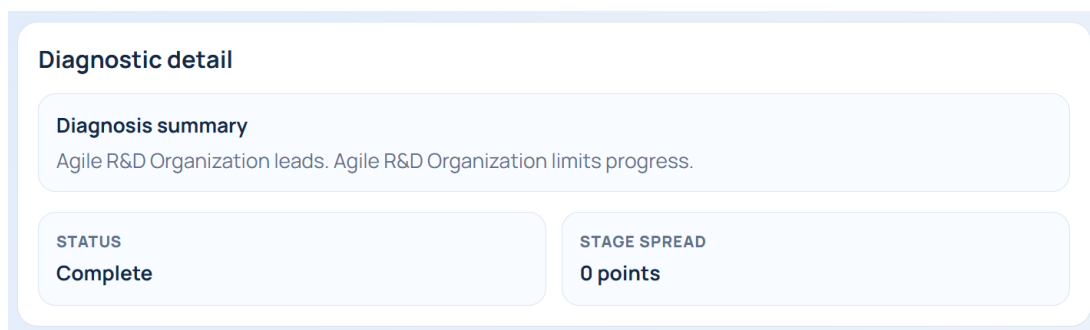


Figura A.1: Detalhamento diagnóstico do ciclo avaliado. Fonte: elaboração própria.

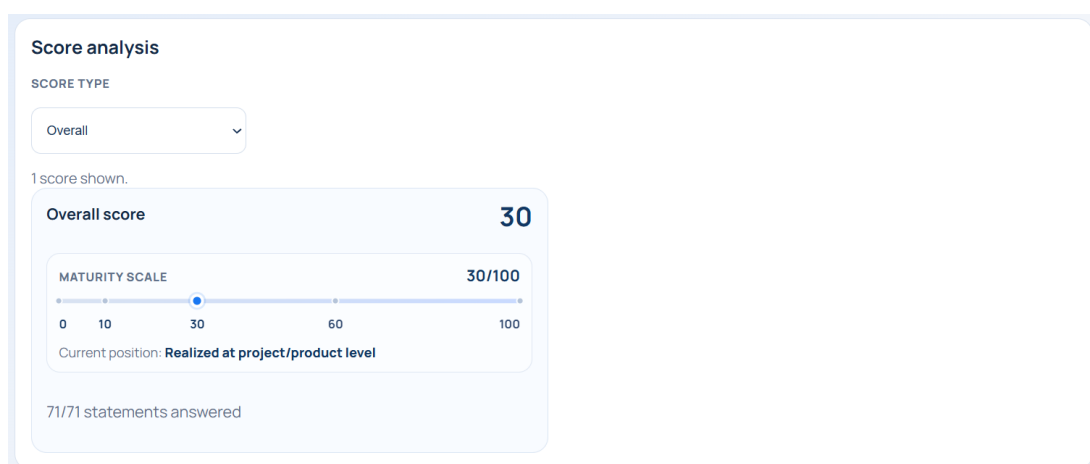


Figura A.2: Análise da pontuação geral do ciclo avaliado. Fonte: elaboração própria.

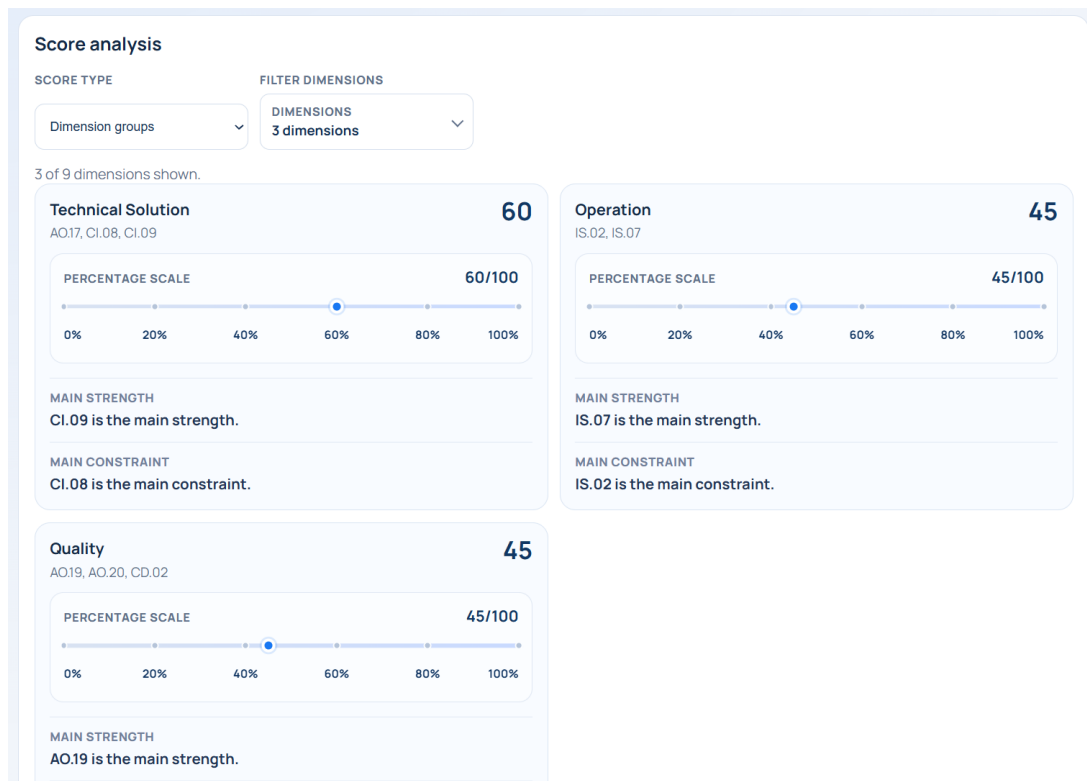


Figura A.3: Análise de pontuações por grupos de dimensões. Fonte: elaboração própria.

Adoption level across assessment items		
ADOPTION LEVEL	STATEMENTS	SHARE
Not adopted	19	27%
Abandoned	9	13%
Realized at project/product level	25	35%
Realized at process level	17	24%
Institutionalized	1	1%

Figura A.4: Distribuição geral dos níveis de adoção entre os itens avaliados. Fonte: elaboração própria.

Process adoption by stage					
PROCESSES	STAGES OF STH				PROCESS ADOPTION
	Agile Organization	CI	CD	R&D as Innovation System	
Business Alignment	20%	-	34%	24%	29%
Continuous Planning, Monitoring and Control	25%	27%	37%	32%	29%
Continuous Quality Assurance	28%	39%	23%	30%	32%
Continuous Improvement & Innovation	18%	30%	35%	31%	30%
Continuous Knowledge Management	12%	-	5%	26%	19%
Continuous Software Measurement	30%	22%	23%	45%	29%
Degree of adoption	23%	31%	31%	29%	28%

Figura A.5: Adoção por processo e estágio. Fonte: elaboração própria.

Element adoption by dimension

FILTER DIMENSIONS

DIMENSIONS

2 dimensions

▼

FILTER ELEMENTS

ELEMENTS

4 elements

▼

4 of 28 elements shown across 1 dimensions.

DIMENSION	ELEMENT	STAGES OF STH				ELEMENT ADOPTION
		Agile Organization	CI	CD	R&D as Innovation System	
Development	Continuous planning activities	15%	0%	-	-	13%
	Continuous requirements engineering	30%	-	-	-	30%
	Focus on Feature	30%	-	-	-	30%
	Modularized architecture and design	-	15%	60%	-	30%
Degree of adoption		19%	10%	60%	-	20%

Figura A.6: Adoção por elemento e dimensão. Fonte: elaboração própria.

Referências

- [1] A. D. Araujo. *Diagnóstico e recomendações para práticas de CI/CD com base no instrumento Zeppelin*. Dissertação de Mestrado, Instituto de Computação, Universidade Estadual de Campinas, Campinas, 2025.
- [2] P. S. S. dos Santos Júnior et al. Flying over Brazilian organizations with Zeppelin: a preliminary panoramic picture of continuous software engineering. In *Proceedings of the XXXVI Brazilian Symposium on Software Engineering (SBES)*. New York: ACM, 2022.
- [3] B. Fitzgerald and K.-J. Stol. Continuous Software Engineering: A Roadmap and Agenda. *Journal of Systems and Software*, 123:176–189, 2017.
- [4] J. Humble and D. Farley. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Boston: Addison-Wesley, 2010.
- [5] J. O. Johanssen et al. Practitioners’ eye on continuous software engineering: an interview study. In *Proceedings of the 2018 International Conference on Software and System Process (ICSSP)*. New York: Association for Computing Machinery, 2018.
- [6] H. H. Olsson and J. Bosch. Climbing the Stairway to Heaven: a multiple-case study exploring barriers in the transition from agile development towards continuous deployment of software. In *Proceedings of the 38th EUROMICRO Conference on Software Engineering and Advanced Applications*, 2014.
- [7] P. S. dos Santos, M. P. Barcellos, and R. F. Calhau. Am I going to Heaven? First step climbing the Stairway to Heaven model: results from a case study in industry. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*, 2020.
- [8] M. Shahin, M. A. Babar, and L. Zhu. Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. *IEEE Access*, 5:3909–3943, 2017.