

Gerenciamento e Análise Comparativa de Organizações na Maturidade em Práticas Contínuas

Pedro Antônio Félix Borges

Relatório Técnico - IC-PFG-26-05

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.

Gerenciamento e Análise Comparativa de Organizações na Maturidade em Práticas Contínuas

Pedro Antônio Félix Borges*

Resumo

A adoção de práticas de Engenharia de Software Contínua (ESC) tem se tornado cada vez mais importante para organizações que buscam aumentar a qualidade do software e reduzir o tempo de entrega de novas funcionalidades. Nesse contexto, o instrumento Zeppelin foi proposto para avaliar o nível de adoção dessas práticas e, posteriormente, recebeu extensões capazes de gerar recomendações personalizadas com base no perfil e no diagnóstico de cada organização. Entretanto, as soluções existentes ainda eram voltadas principalmente a experimentos de pesquisa, não oferecendo uma plataforma integrada para a aplicação do instrumento e o acompanhamento dos resultados.

Este trabalho apresenta o desenvolvimento de uma plataforma web para a aplicação do Zeppelin, permitindo o gerenciamento de organizações, a realização de avaliações, a visualização de diagnósticos e recomendações, além do acompanhamento da evolução dessas organizações ao longo do tempo. No contexto deste projeto, o foco concentrou-se no desenvolvimento do módulo de gerenciamento de organizações e das funcionalidades de análise, incluindo mecanismos de comparação entre ciclos de avaliação e de *benchmarking* entre organizações com perfis semelhantes.

Como resultado, a plataforma passou a oferecer uma experiência integrada para o uso do instrumento Zeppelin, ampliando suas capacidades por meio de funcionalidades que facilitam a interpretação dos resultados, o acompanhamento da evolução organizacional e a comparação com outras organizações, contribuindo para a adoção de práticas contínuas de desenvolvimento de software.

1 Introdução

A adoção de práticas contínuas tornou-se um dos principais fatores para aumentar a qualidade e a velocidade de desenvolvimento de software [9]. Práticas como Integração Contínua (CI), Entrega e Implantação Contínua (CD) permitem reduzir o tempo entre alterações no código e a disponibilização aos usuários, além de aumentar a confiabilidade do processo de desenvolvimento por meio da automação de testes, *builds* e implantações [5].

Entretanto, a adoção dessas práticas não depende apenas de ferramentas tecnológicas, mas também de mudanças organizacionais, culturais e de processo [9]. Dessa forma, avaliar o nível de maturidade de uma organização em relação à Engenharia de Software Contínua

*Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

(ESC) torna-se uma atividade importante para identificar pontos fortes, oportunidades de melhoria e apoiar a evolução contínua das equipes de desenvolvimento.

Nesse contexto, o instrumento Zeppelin foi proposto como mecanismo para diagnosticar o nível de adoção de práticas contínuas em organizações, por meio de um conjunto estruturado de questões fundamentadas nos modelos Engenharia de Software Contínua e *Stairway to Heaven* [3]. Posteriormente, o instrumento foi refinado e expandido com um mecanismo de recomendações personalizadas, permitindo não apenas identificar o nível de maturidade da organização, mas também sugerir ações para sua evolução [1].

Apesar dessas contribuições, as soluções existentes ainda eram voltadas principalmente para avaliações conceituais de pesquisa, não oferecendo uma plataforma integrada que permitisse gerenciar organizações, realizar avaliações, acompanhar a evolução dos resultados ao longo do tempo e comparar organizações com perfis semelhantes de forma prática.

Diante desse cenário, este trabalho apresenta o desenvolvimento de uma plataforma web¹ para aplicação do instrumento Zeppelin. A aplicação foi construída a partir de um módulo *back-end* semente previamente existente (ainda que incipiente) e expandida para oferecer uma interface completa para gerenciamento de organizações, realização de avaliações, visualização de resultados, acompanhamento da evolução temporal, *benchmarking* entre organizações e acesso às recomendações personalizadas propostas na literatura.

O desenvolvimento da plataforma foi realizado em equipe, sendo este Projeto Final de Graduação dedicado às funcionalidades implementadas pelo autor. Especificamente, este trabalho aborda o desenvolvimento do módulo de gerenciamento de organizações e das funcionalidades de análise da evolução temporal e de *benchmarking* entre organizações.

Este relatório está organizado da seguinte forma. A Seção 2 apresenta a fundamentação teórica necessária para a compreensão do trabalho. A Seção 3 descreve os trabalhos relacionados e o contexto da solução proposta. A Seção 4 apresenta uma visão geral da arquitetura e da interface da aplicação. A Seção 5 descreve as funcionalidades desenvolvidas neste projeto. Por fim, a Seção 6 apresenta as conclusões e possíveis trabalhos futuros.

2 Fundamentação Teórica

As práticas contínuas são um dos pilares do desenvolvimento moderno de software. Ao longo dos últimos anos, essas práticas estabeleceram tanto fundamentos tecnológicos quanto hábitos culturais que possibilitaram ciclos de entrega mais curtos, maior qualidade de software e melhor alinhamento entre as equipes de desenvolvimento e de negócio.

Esta seção apresenta uma breve introdução aos conceitos de Integração Contínua (CI), Entrega Contínua e Implantação Contínua (CD) [6], Engenharia de Software Contínua (ESC), *Stairway to Heaven* (StH) e *Eye of ESC*.

2.1 Integração Contínua

Integração Contínua [6] é uma prática de desenvolvimento de software na qual os desenvolvedores integram frequentemente suas alterações de código em um repositório compar-

¹Código-fonte disponível em: <https://github.com/brenorobazza/zeppelin>

tilhado por meio de um processo automatizado. A integração, de fato, ocorre apenas após a execução bem-sucedida das etapas de *build* e dos testes automatizados. Para sua adoção eficaz, é necessário um alto nível de cobertura de testes automatizados, usualmente focado em testes de unidade, além de um processo automatizado de *build*.

2.2 Entrega e Implantação Contínua

Entrega Contínua [6] surgiu com o objetivo de acelerar o processo de lançamento de um produto. Consiste em um pipeline de *builds* e testes, agora em níveis mais altos, como integração e sistema, que busca garantir a qualidade do software, automatizando ao máximo o processo de entrega.

A principal diferença entre Entrega Contínua e Implantação Contínua (referido como CD ao longo deste relatório) é o nível de automação do processo de entrega [9]. Na Implantação Contínua, toda o pipeline é automatizada: após um *merge*, se todas as etapas da pipeline forem executadas com sucesso, a nova versão é automaticamente disponibilizada em produção. Já na Entrega Contínua, apesar de a pipeline também ser automatizada, ainda há uma etapa manual para aprovar ou iniciar a implantação.

Os principais requisitos para a adoção de CD incluem a construção de um pipeline de implantação robusto, além dos mesmos requisitos exigidos pela Integração Contínua.

2.3 Engenharia de Software Contínua

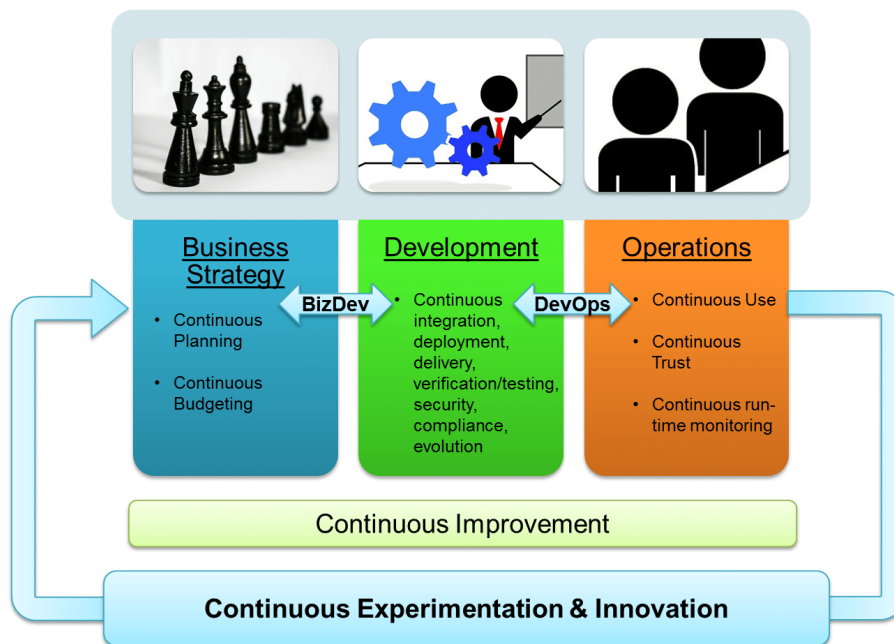


Figura 1: Engenharia de Software Contínua (adaptado de [5])

Engenharia de Software Contínua (ESC) é um conceito que reúne práticas contínuas apresentadas anteriormente, como CI e CD, entre outras, visando estabelecer um fluxo contínuo ao longo de todo o ciclo de desenvolvimento de software [5].

Além da adoção dessas práticas, o ESC também aborda a forma como as diferentes áreas envolvidas no ciclo de desenvolvimento de software interagem entre si, como mostra a Figura 1. Essas áreas são tradicionalmente divididas em três grandes domínios: Negócio, Desenvolvimento e Operação. Dessa forma, o ESC engloba práticas como CI/CD, aborda-gens *DevOps* (integração entre desenvolvimento e operação) [2], *BizDev* (integração entre desenvolvimento e negócio) e Continuous Experimentation, que promove um ciclo contínuo de coleta de *feedback* de usuários e *stakeholders* para apoiar a evolução e a inovação contínua do software [5].

2.4 Stairway to Heaven

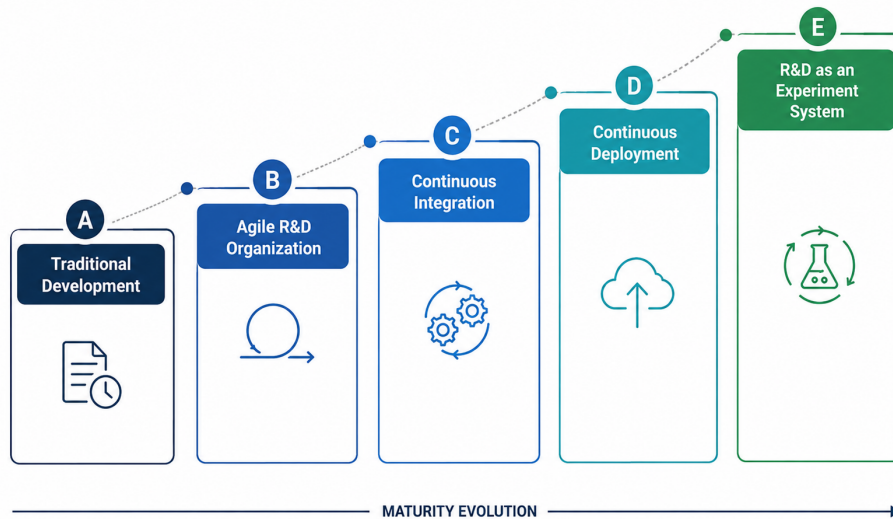


Figura 2: Stairway to Heaven (adaptado de [8])

O modelo *Stairway to Heaven* (StH), proposto por Olsson, Alahyari e Bosch [8], organiza a adoção de práticas contínuas em cinco níveis evolutivos e cumulativos, representando a evolução das organizações desde processos tradicionais de desenvolvimento até um ambiente baseado em experimentação contínua. A Figura 2 apresenta uma visão geral do modelo.

Level A – Desenvolvimento Tradicional

Representa organizações que ainda utilizam modelos tradicionais de desenvolvimento, normalmente baseados em ciclos longos, como Cascata ou processos iterativos de ciclos longos. Nesse nível, o grau de automação é reduzido e o *feedback* dos usuários só ocorre após longos períodos de desenvolvimento.

Level B – Organização de Pesquisa e Desenvolvimento Ágil

Neste nível, as organizações já adotam metodologias ágeis. Entretanto, seus processos de integração, testes e *deploy* ainda dependem fortemente de atividades manuais.

Level C – Integração Contínua

As organizações passam a adotar práticas de Integração Contínua, automatizando *builds* e testes. Entretanto, o processo de *deploy* permanece manual.

Level D – Implantação Contínua

Neste estágio ocorre a adoção de Implantação Contínua, automatizando também o processo de entrega da aplicação.

Level E – Pesquisa e Desenvolvimento como um Sistema de Experimentos

No nível mais avançado do modelo, o *deploy* deixa de ser o objetivo final e passa a ser apenas uma etapa intermediária. O foco passa a ser um ciclo contínuo de experimentação, permitindo coletar rapidamente *feedback* dos usuários para evoluir o produto de forma iterativa.

2.5 Eye of ESC

Johanssen et al. (2018) [7] conduziram um estudo com 24 profissionais de 17 empresas, com o objetivo de compreender como a Engenharia de Software Contínua é percebido e implementado na indústria.

Como resultado, os autores observaram que a adoção completa de ESC ainda representa um desafio para a maioria das organizações. Com o objetivo de fornecer uma visão mais ampla desse cenário, foi proposto o modelo Eye of ESC, composto por nove dimensões que representam áreas fundamentais para a adoção de Engenharia de Software Contínua:

- Usuário (*User*);
- Gestão de Software (*Software Management*);
- Desenvolvimento (*Development*);
- Código (*Code*);
- Qualidade (*Quality*);
- Negócio (*Business*);
- Desenvolvedor (*Developer*);
- Operação (*Operation*);
- Conhecimento (*Knowledge*).

3 Contexto da Solução Proposta

3.1 Instrumento Zeppelin

dos Santos Júnior et al. (2021) propuseram o Zeppelin [3], um instrumento para avaliar o nível de adoção de práticas de Engenharia de Software Contínua (ESC) e do modelo

Stairway to Heaven (StH). O instrumento é composto por 76 questões, cada uma respondida utilizando cinco níveis de adoção:

- **Não Adotado (0%);**
- **Abandonado (10%);**
- **Nível de Projeto/Produto (30%);**
- **Nível de Processo (60%);**
- **Institucionalizado (100%).**

As questões estão distribuídas em oito categorias, que representam dimensões da ESC e são mapeadas proporcionalmente aos níveis do modelo *Stairway to Heaven*. O instrumento foi desenvolvido para aplicação em organizações e deve ser respondido por profissionais com amplo conhecimento sobre o processo de desenvolvimento de software da empresa avaliada.

Posteriormente, dos Santos Júnior et al. (2022) [4] realizaram um refinamento, reduzindo o número de questões de 76 para 71 e ampliando a validação por meio da aplicação em um número maior de organizações brasileiras.

3.2 Recomendações Baseadas no Zeppelin

Posteriormente, Araújo (2025) [1] conduziu um estudo com o objetivo de fornecer recomendações personalizadas para a adoção de práticas de CI/CD, de acordo com o contexto e o nível de maturidade identificados pelo Zeppelin.

Como resultado, os autores calibraram os pesos atribuídos às questões, considerando diferentes perfis organizacionais, e desenvolveram recomendações práticas e personalizadas para apoiar a evolução das organizações em práticas contínuas.

Durante a avaliação da proposta, 60% dos participantes reconheceram que as recomendações recebidas poderiam contribuir para melhorar a adoção de práticas contínuas, enquanto 68,5% afirmaram que pretendiam colocá-las em prática.

3.3 Aplicação Web Proposta

Entretanto, embora o Zeppelin e o trabalho de Araújo forneçam mecanismos para diagnóstico e geração de recomendações, ainda não existia uma plataforma web que integrasse essas funcionalidades, permitindo o gerenciamento de organizações, o acompanhamento da evolução das avaliações e o *benchmarking* entre empresas. Nesse contexto, este projeto apresenta o desenvolvimento dessa plataforma.

4 Visão Geral da Aplicação

A aplicação desenvolvida neste projeto consiste em uma plataforma web para a aplicação do instrumento Zeppelin.

Para seu desenvolvimento, utilizou-se como base um projeto previamente desenvolvido por dos Santos Júnior, cujo objetivo era automatizar a análise dos estudos iniciais realizados com o instrumento Zeppelin. A solução original consistia em um back-end capaz de receber, como entrada, os resultados de questionários em formato de planilha, processar os diagnósticos e persistir os resultados em um banco de dados.

A partir dessa base, a aplicação foi expandida para uma plataforma web completa, permitindo que organizações realizem avaliações diretamente na interface, acompanhem sua evolução ao longo do tempo, comparem seus resultados com diferentes perfis organizacionais e recebam recomendações personalizadas para a evolução de práticas contínuas.

Na plataforma, cada avaliação realizada por uma organização é denominada um ciclo. Um ciclo corresponde ao preenchimento completo ou parcial de um questionário Zeppelin em determinado momento, armazenando informações como a data de criação, as respostas fornecidas e os resultados gerados. Dessa forma, a evolução de uma organização pode ser acompanhada por meio da comparação entre diferentes ciclos ao longo do tempo.

A aplicação adota uma estrutura multi-organização. Um mesmo usuário pode estar associado a uma ou mais organizações e alternar entre elas durante a utilização da plataforma. Todas as avaliações, análises, recomendações e comparações são sempre associadas simultaneamente a um usuário e a uma organização específicos.

Esta seção apresenta uma visão geral da arquitetura e da interface da aplicação. Os módulos que compõem o sistema serão detalhados nas próximas seções.

4.1 Visão Geral da Arquitetura

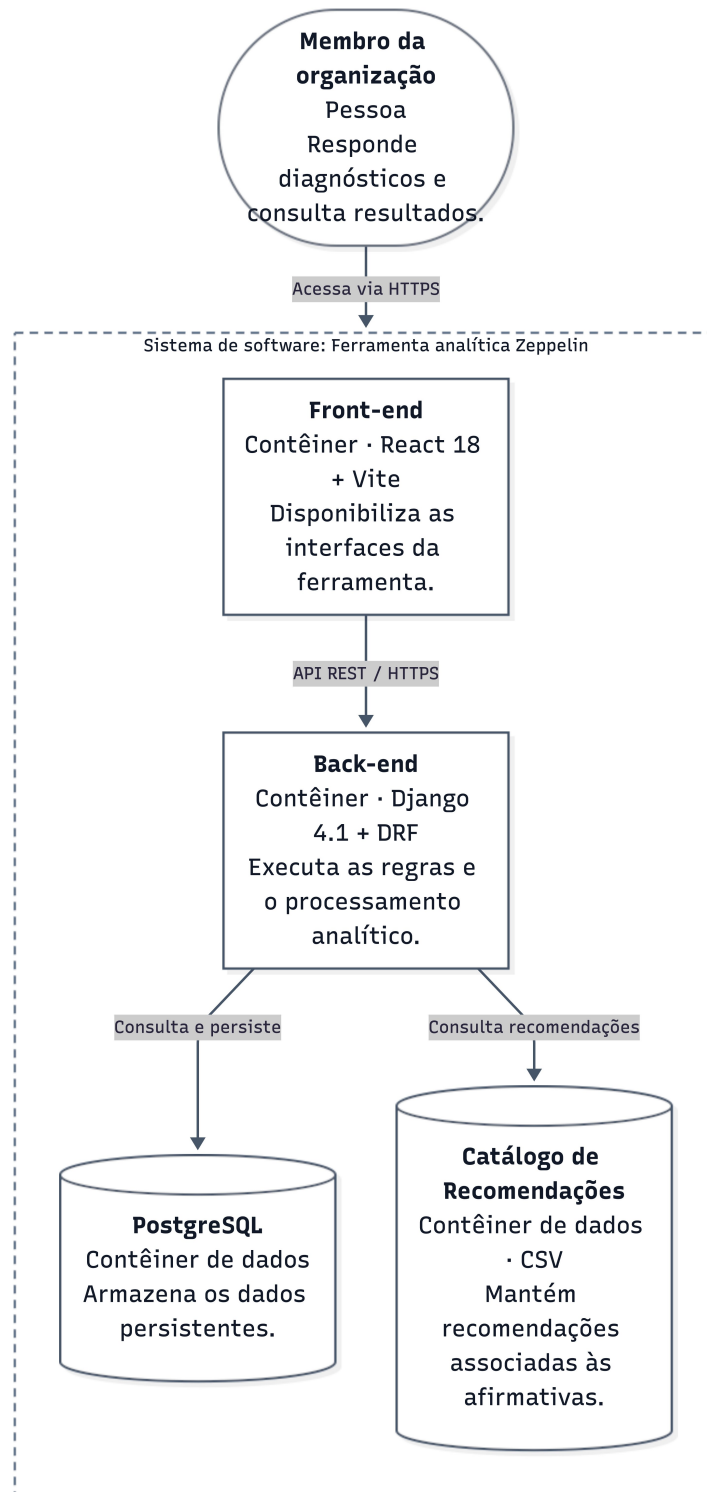


Figura 3: Visão geral da arquitetura da plataforma.

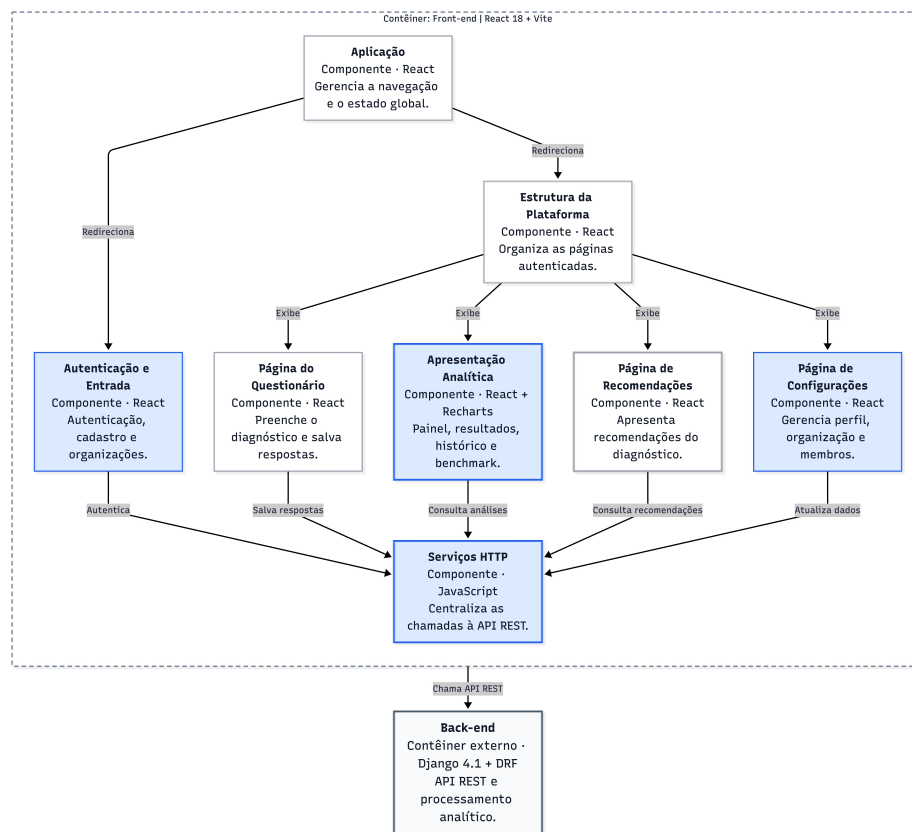


Figura 4: Componentes do Front-End da plataforma. Em azul são destacados os componentes desenvolvidos no escopo deste PFG.

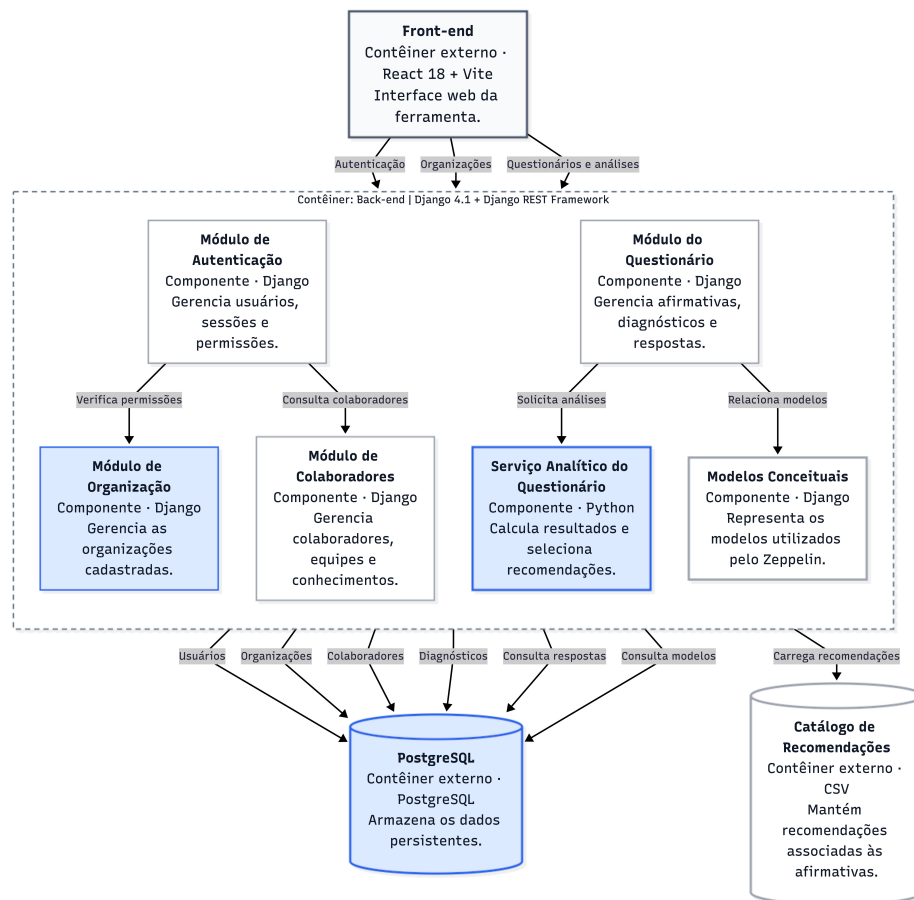


Figura 5: Componentes do Back-End da plataforma. Em azul são destacados os componentes desenvolvidos no escopo deste PFG.

As Figuras 3, 4 e 5 apresentam diferentes níveis de detalhamento da arquitetura da plataforma desenvolvida. Nas Figuras 4 e 5, os componentes destacados em azul correspondem aos módulos e funcionalidades implementados no escopo deste Projeto Final de Graduação.

A Figura 3 apresenta uma visão geral da arquitetura, evidenciando sua organização em três camadas principais: Front-end, Back-end e Dados. Essa separação favorece a modularização da aplicação, permitindo que cada camada possua responsabilidades bem definidas.

A Figura 4 detalha a arquitetura da camada de Front-end. Nela são apresentados os principais componentes responsáveis pela interação com o usuário, incluindo a interface de autenticação, o módulo de avaliação, as funcionalidades analíticas, o gerenciamento de organizações e o adaptador responsável pela comunicação com a API do back-end.

A Figura 5 apresenta a arquitetura do Back-end, evidenciando os módulos responsáveis pela autenticação, gerenciamento do questionário, processamento das funcionalidades analíticas e acesso à camada de dados. Também são representados o banco PostgreSQL e o catálogo de recomendações em formato CSV utilizados pela aplicação.

A seguir, são descritas as principais tecnologias e responsabilidades de cada camada da arquitetura.

Front-end. A camada de apresentação foi desenvolvida utilizando React e Vite.js. Ela é responsável pela interface com o usuário, pela renderização das páginas da aplicação e pela comunicação com o back-end por meio de requisições HTTP. Nessa camada encontram-se os módulos responsáveis pelo preenchimento dos questionários, pelo gerenciamento das organizações, pela visualização das recomendações e pelas funcionalidades analíticas da plataforma.

Back-end. O back-end foi desenvolvido utilizando Django e concentra toda a lógica de negócio da aplicação. Entre suas responsabilidades estão o processamento dos questionários, o cálculo dos diagnósticos, o gerenciamento das organizações, a geração das análises e recomendações e a disponibilização das funcionalidades por meio de uma API REST.

Camada de Dados. A camada de dados é composta por um banco PostgreSQL, responsável pela persistência das informações da aplicação, e por um catálogo de recomendações em formato CSV utilizado pelo módulo analítico para enriquecer as recomendações apresentadas aos usuários.

Autenticação e CI/CD. A aplicação também conta com um sistema de autenticação baseado em OAuth2, responsável pelo controle de acesso às funcionalidades do sistema. Além disso, foi desenvolvido um pipeline de CI/CD para automatizar a execução de testes, *builds* e processos de implantação, contribuindo para a adoção das práticas contínuas apresentadas na Seção 2.

4.2 Interface da Aplicação

Embora a arquitetura apresentada na Figura 3 organize a aplicação em módulos funcionais, a interação do usuário ocorre por meio de sete páginas principais. A Tabela 1 apresenta a correspondência entre cada página da interface e o módulo arquitetural ao qual ela pertence.

Nas subseções seguintes é apresentada uma visão geral de cada módulo da aplicação por meio de suas respectivas páginas.

Página	Módulo
Dashboard	Análise
Assessment	Questionário
Results	Análise
Recommendations	Recomendações
Evolution	Análise
Benchmark	Análise
Settings	Organização

Tabela 1: Correspondência entre as páginas da interface e os módulos da arquitetura.

4.2.1 Módulo Questionário

O módulo do Questionário, ilustrado nas Figuras 6 e 7, é representado pela página *Assessment*, responsável por reunir todos os ciclos pertencentes à organização atualmente selecionada. Nessa página, o usuário pode visualizar avaliações anteriores, acompanhar o status de cada ciclo e iniciar um novo preenchimento do instrumento Zeppelin.

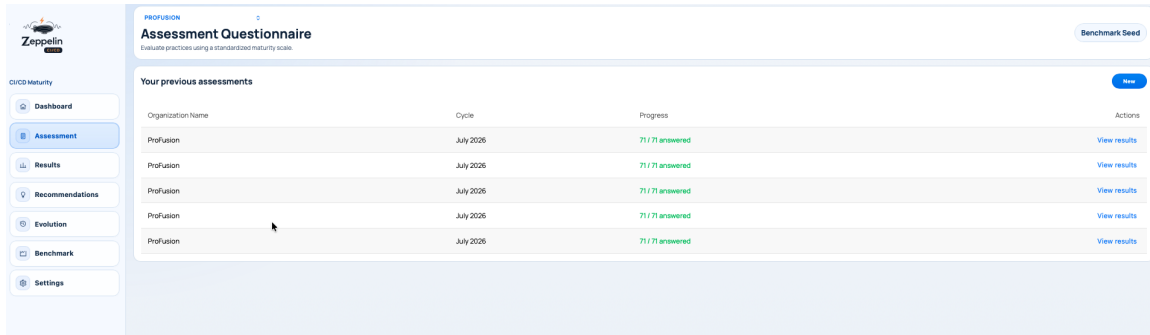


Figura 6: Página Assessment contendo os ciclos de avaliação da organização selecionada.

Ao selecionar um ciclo existente ou iniciar uma nova avaliação, o usuário é direcionado para a interface de preenchimento do questionário Zeppelin.

Zeppelin

CICD Maturity

- Dashboard
- Assessment
- Results
- Recommendations
- History
- Benchmark
- Settings

Sign out

PROFUSION

Assessment Questionnaire

Evaluate practices using a standardized maturity scale.

Pedro

Agile R&D Organization 0 / 26

Continuous Integration 0 / 16

Continuous Deployment 0 / 17

R&D as an Experiment System 0 / 13

Agile R&D Organization Progress

Question 1 of 26

Exit and save

[AO.01]

Papéis envolvidos no processo de desenvolvimento ágil (e.g., Scrum Master, Product Owner, Desenvolvedor e Tester) existem na organização.

0% - Not adopted
A prática não é adotada.

10% - Abandoned
A prática foi tentada, mas abandonada.

30% - Realized at project/product level
Adotado em alguns projetos ou produtos específicos.

60% - Realized at process level
Padronizado como um processo dentro da organização.

100% - Institutionalized
Totalmente institucionalizado e enraizado na cultura da empresa.

Figura 7: Interface do questionário Zeppelin.

4.2.2 Módulo Recomendações

A página *Recommendations*, ilustrada nas Figuras 8 e 9, disponibiliza recomendações personalizadas para auxiliar a organização na evolução da adoção de práticas contínuas. As recomendações são geradas considerando simultaneamente o diagnóstico obtido pelo Zeppelin e o perfil organizacional informado no cadastro da empresa.

A interface organiza as recomendações por categorias e apresenta orientações específicas para as dimensões com maior potencial de evolução.

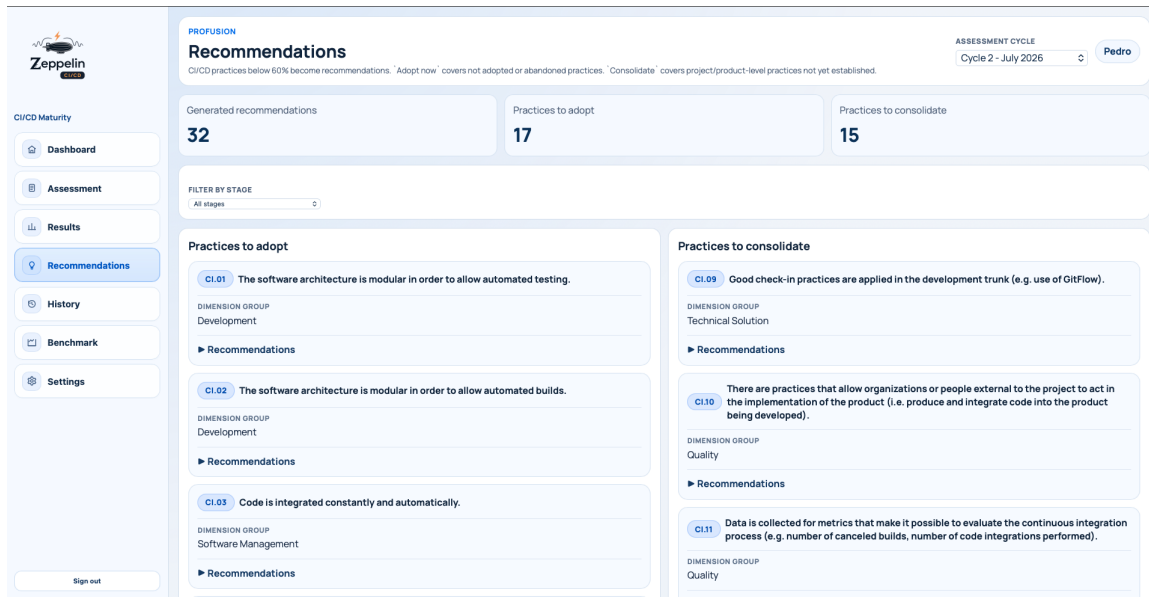


Figura 8: Página de recomendações personalizadas.

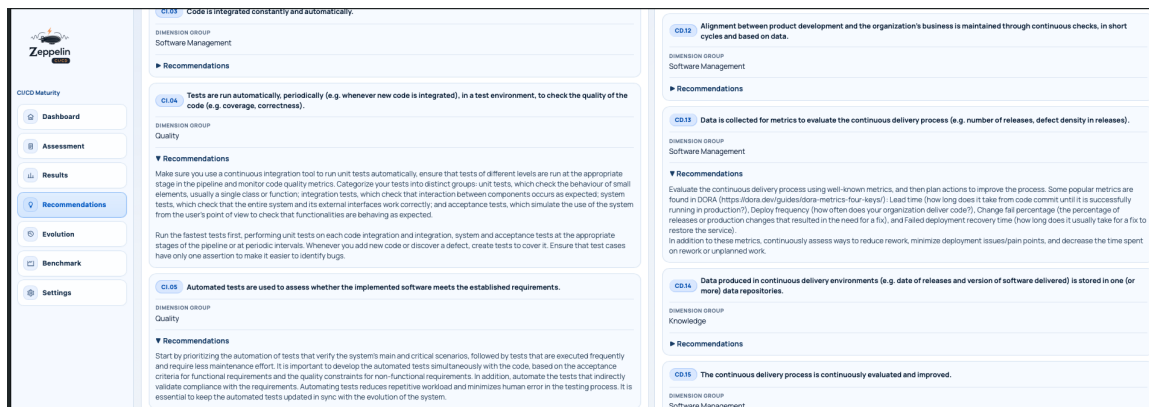


Figura 9: Exemplo de recomendações apresentadas ao usuário.

4.2.3 Módulo Análise

O módulo de Análise reúne as funcionalidades responsáveis por transformar os diagnósticos produzidos pelo Zeppelin em informações que auxiliem as organizações na compreensão de seu nível de maturidade, no acompanhamento de sua evolução e na comparação com outras organizações.

Esse módulo engloba quatro páginas da aplicação: *Dashboard*, *Results*, *Evolution* e *Benchmark*. Cada uma delas atende a um objetivo específico no processo de análise.

Dashboard O *Dashboard*, ilustrado na Figura 10, é a página inicial da aplicação. Seu objetivo é fornecer uma visão resumida da situação atual da organização, apresentando indicadores do ciclo selecionado, a pontuação geral, o progresso no preenchimento do questionário e informações relevantes sobre o diagnóstico mais recente.

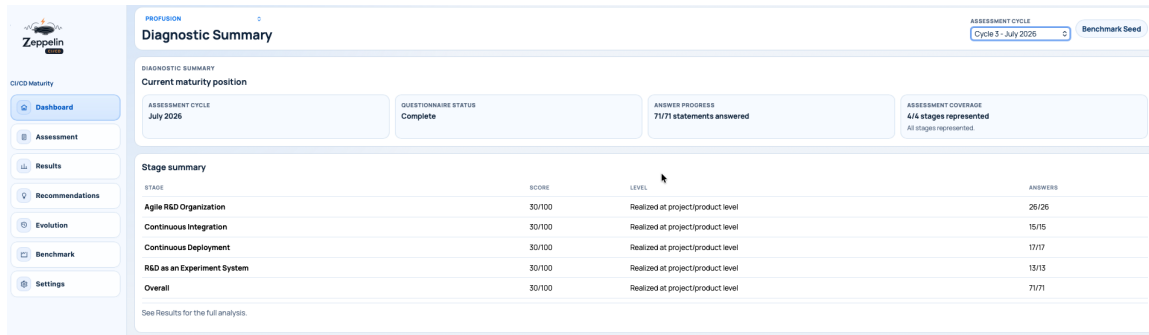


Figura 10: Página Dashboard.

Results A página *Results*, ilustrada na Figura 11, apresenta os resultados detalhados do diagnóstico realizado pelo Zeppelin. Nela é possível visualizar a pontuação geral da organização, os resultados por estágio do modelo *Stairway to Heaven* (StH), as dimensões da Engenharia de Software Contínua (ESC) e o nível de adoção obtido em cada uma delas, o que permite uma análise detalhada dos pontos fortes e das oportunidades de melhoria da organização.

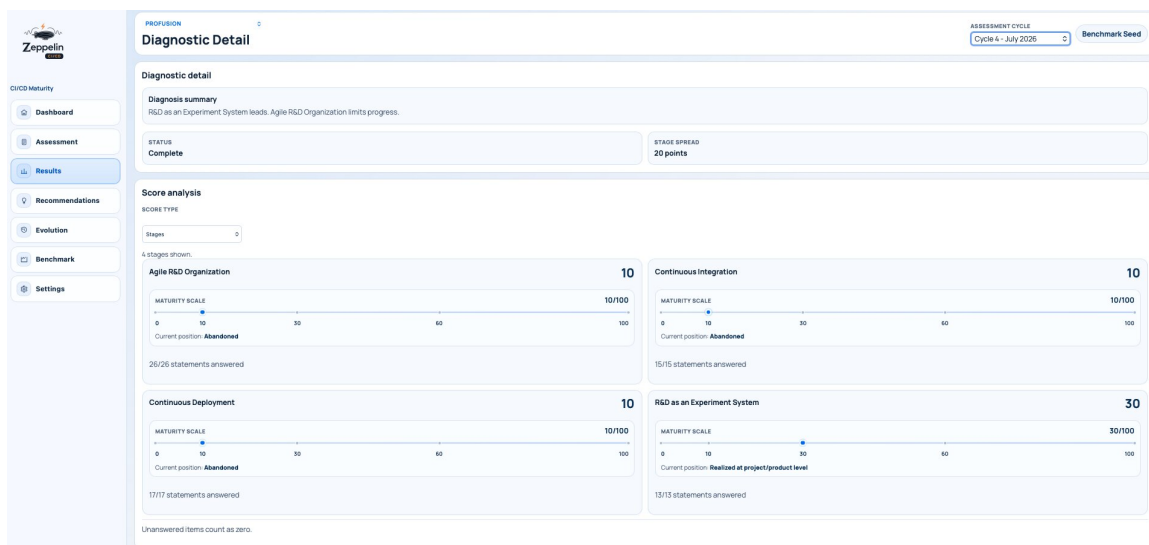


Figura 11: Página Results.

Evolution A página *Evolution*, ilustrada na Figura 12, permite acompanhar a evolução histórica da organização por meio da comparação entre diferentes ciclos. Os resultados são apresentados por meio de gráficos e indicadores que permitem identificar avanços ou regressões nas dimensões avaliadas ao longo do tempo.

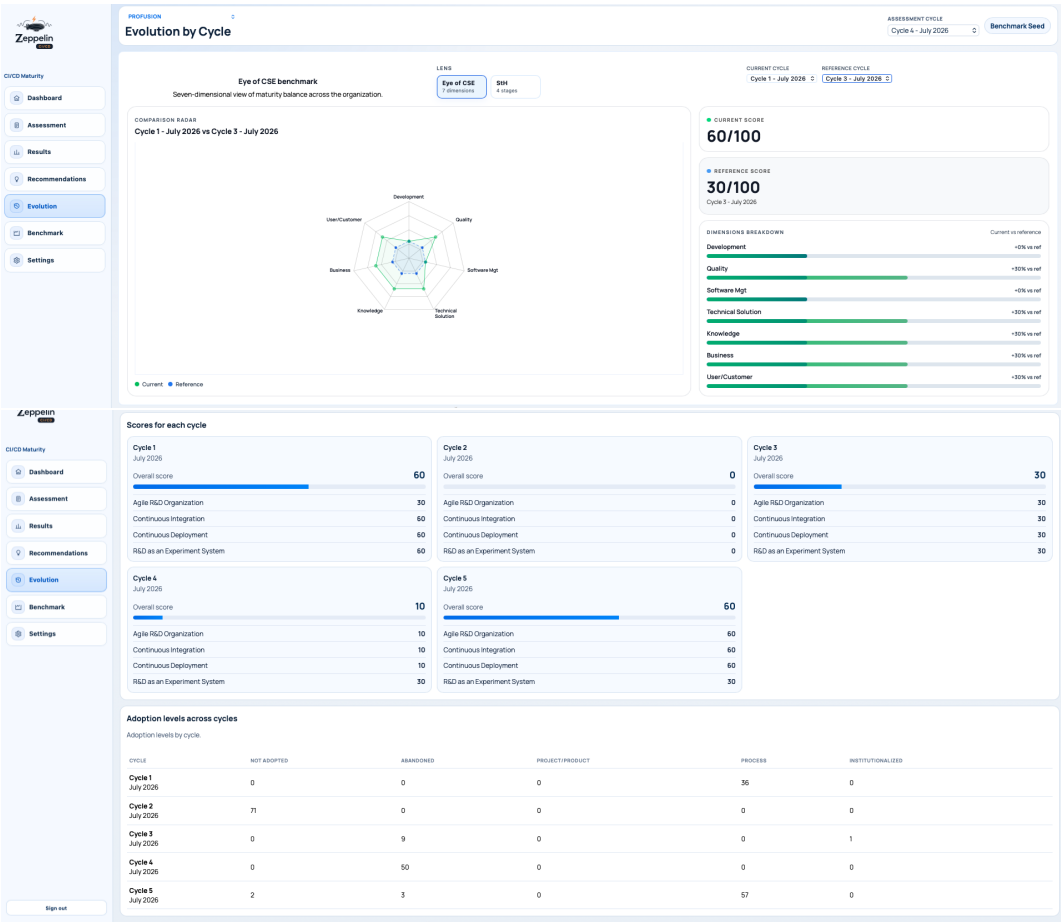


Figura 12: Página Evolution.

Benchmark A página *Benchmark*, ilustrada na Figura 13, permite comparar os resultados da organização com dados agregados de outras empresas cadastradas na plataforma. O usuário pode aplicar filtros com base em características organizacionais, como o setor de atuação, o que permite comparar sua situação atual com a de diferentes grupos por meio de gráficos de radar.

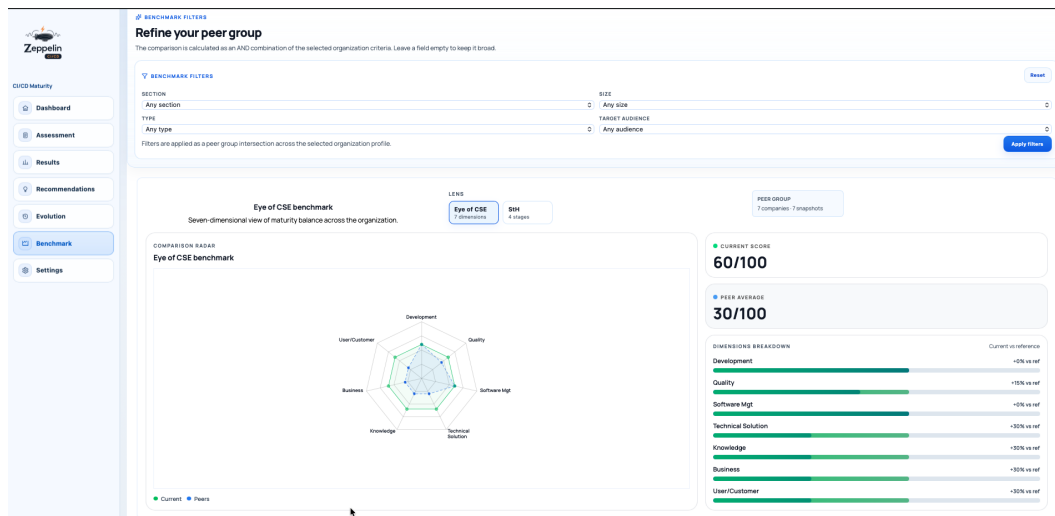


Figura 13: Página Benchmark.

4.2.4 Módulo Organização

O módulo de Organização é acessado pela página *Settings*, ilustrada na Figura 14. Nela estão concentradas as funcionalidades relacionadas ao gerenciamento de empresas às quais o usuário pertence, permitindo criar novas organizações, ingressar em organizações existentes, desvincular-se de organizações e atualizar informações do perfil, como o nome de usuário.

Embora todos esses módulos componham a plataforma desenvolvida, este trabalho concentra-se principalmente nas funcionalidades implementadas pelo autor, relacionadas ao gerenciamento de empresas e aos recursos de análise, especialmente as de evolução histórica e comparação entre organizações (*Benchmark*). Essas contribuições são detalhadas na próxima seção.

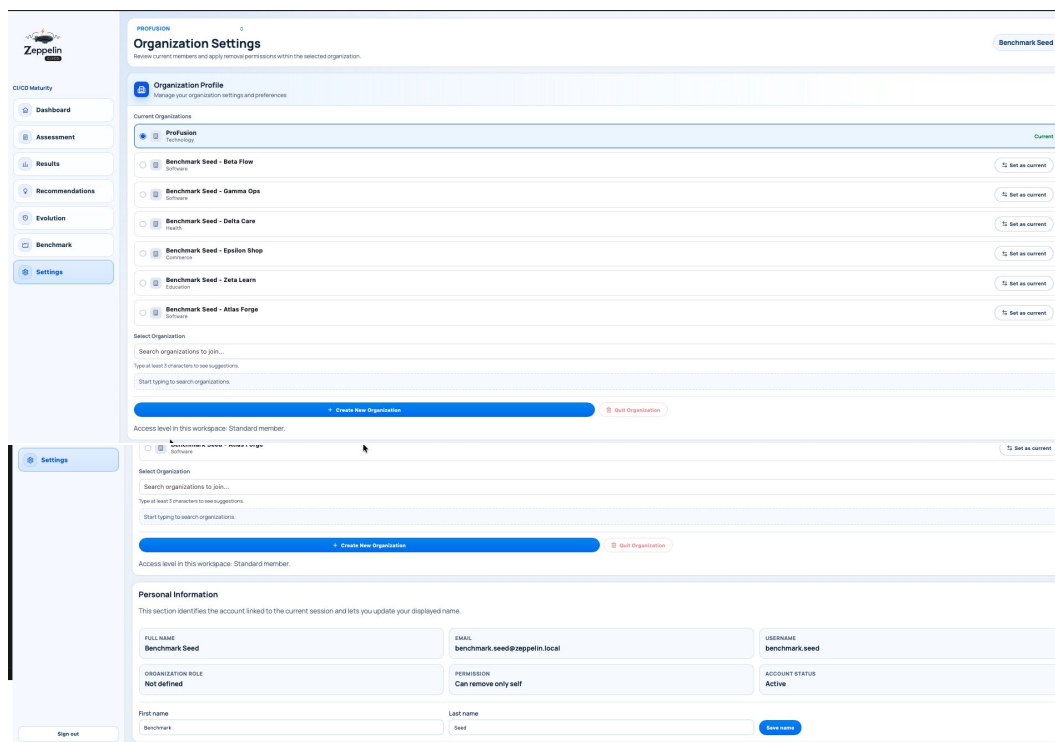


Figura 14: Página Settings.

5 Funcionalidades Desenvolvidas

Esta seção apresenta as principais funcionalidades implementadas² durante o desenvolvimento deste trabalho. O foco concentrou-se no gerenciamento de organizações e nas funcionalidades de análise, permitindo acompanhar a evolução dessas organizações ao longo do tempo e compará-las com outras empresas cadastradas na plataforma.

5.1 Módulo de Organização

O módulo de Organização é um dos componentes centrais da plataforma, uma vez que todas as avaliações, resultados, recomendações e análises são associados a uma organização. Dessa forma, esse módulo é responsável por gerenciar todo o ciclo de vida das organizações e controlar qual delas será utilizada como contexto da aplicação.

Para garantir a consistência dos dados, definiu-se como regra de negócio que todo usuário deve estar vinculado a pelo menos uma organização. Assim, durante o cadastro, após informar seus dados básicos, o usuário é direcionado para a tela apresentada na Figura 15, onde pode optar por ingressar em uma organização já existente ou cadastrar uma nova organização.

²Código-fonte disponível em: <https://github.com/brenorobazza/zeppelin>

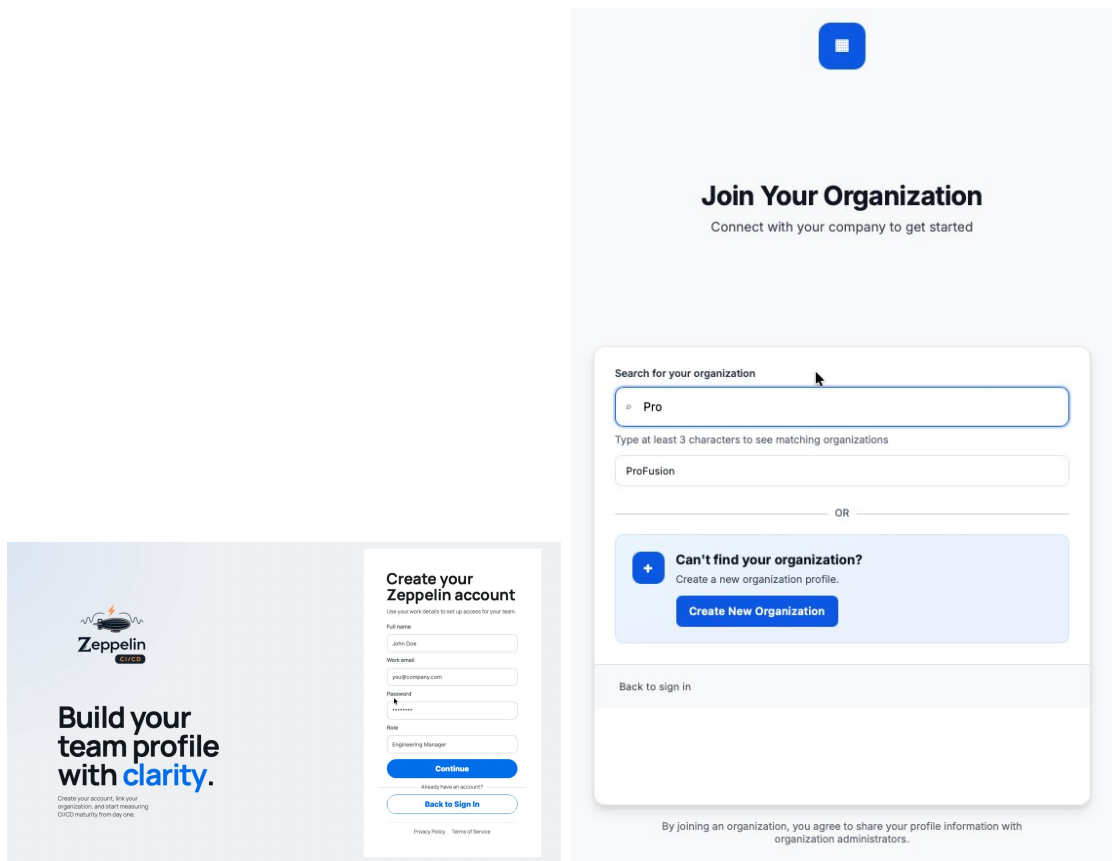


Figura 15: Fluxo de cadastro do usuário e associação inicial a uma organização.

5.1.1 Cadastro de Organizações

Quando o usuário opta por cadastrar uma nova organização, é apresentado um formulário composto por oito campos, ilustrado na Figura 16. Essas informações têm duas finalidades principais: fornecer os dados necessários à personalização das recomendações propostas por Araújo et al. (2025) e servir de base para as funcionalidades de análise e *benchmarking* implementadas na plataforma.

Os campos disponíveis são:

- **Nome (Obrigatório):** nome da organização.
- **País (Obrigatório):** seleção do país, com Brasil como valor padrão.
- **Estado:** obrigatório apenas quando o país selecionado é Brasil.
- **Tipo de Organização (Obrigatório):** Privado, Público, Híbrido, Startup, Software House ou TI.

Organization Registration ~5 minutes

Progress Step 1 of 1

Organization Details

Please provide information about your organization and software development practices.

Organization Name *

Organization Test

In which country is your organization located? *

If multiple countries, indicate your work unit's country

Brazil

In which Brazilian state is your organization located? (required only if country is Brazil)

If multiple locations, indicate your work unit's state

Sao Paulo

Organization Type *

Private

Organization Sector *

Technology

Organization Size (number of employees) *

200

What is the main target audience of the software developed?

B2C

How many years has your organization been developing software?

10

Back Create Organization

Your information is secure and will be used only for research purposes.

Figura 16: Cadastro de Organização

- **Setor da Organização (Obrigatório):** Financeiro, Saúde, Tecnologia, Indústria, Varejo ou Educação.
- **Número de Empregados (Obrigatório):** número de colaboradores da organização.
- **Público-Alvo (Opcional):** B2B, B2C, B2G, Interno ou Misto.
- **Tempo de Desenvolvimento de Software (Opcional):** tempo, em anos, desde o início do desenvolvimento de software pela organização.

5.1.2 Gerenciamento de Organizações

As funcionalidades de gerenciamento de organizações estão concentradas na página *Settings* ilustrada na Figura 14. Nela, o usuário pode visualizar todas as organizações às quais está vinculado e selecionar qual delas será a organização ativa.

A organização selecionada define o contexto da aplicação. Todas as informações apresentadas nas páginas de *Dashboard*, *Assessment*, *Results*, *Evolution*, *Benchmark* e *Recommendations* passam a considerar exclusivamente os dados da organização ativa. Essa seleção também pode ser realizada diretamente no cabeçalho da aplicação, permitindo alternar rapidamente entre organizações durante a navegação.

Além da seleção da organização ativa, o módulo permite ingressar em organizações previamente cadastradas por meio de um campo de busca com autocompletar e cadastrar

novas organizações por meio do formulário descrito anteriormente.

Também foi implementada a funcionalidade de desvinculação de organizações. Entretanto, como regra de negócio, essa operação somente é permitida quando o usuário permanece associado a pelo menos outra organização, o que impede que haja usuários sem organização válida na plataforma.

Essas funcionalidades tornam o gerenciamento das organizações simples e centralizado, além de fornecerem a base necessária para todas as demais funcionalidades da aplicação, como avaliações, análises, recomendações e comparações entre organizações.

5.2 Evolução Temporal

A funcionalidade de Evolução Temporal está localizada no topo da página *Evolution* ilustrada na Figura 12. Seu objetivo é permitir que a organização acompanhe sua evolução ao longo do tempo por meio da comparação entre dois ciclos distintos.

A comparação é realizada por meio de um gráfico de radar, que pode ser visualizado tanto nas dimensões do modelo *Eye of ESC* quanto nos estágios do modelo *Stairway to Heaven (StH)*. Além da representação gráfica, são apresentados indicadores da pontuação geral e das respectivas dimensões avaliadas, permitindo uma análise mais detalhada da evolução da organização.

Para garantir que as comparações representem corretamente a maturidade da organização, foram implementadas duas regras de negócio principais:

- Apenas ciclos concluídos podem ser comparados, uma vez que ciclos incompletos não possuem todas as respostas necessárias para o cálculo correto das métricas e poderiam gerar interpretações incorretas da evolução da organização;
- Caso a organização possua menos de dois ciclos concluídos, o componente entra automaticamente no estado de *Dados Insuficientes*, indicando que ainda não há informações suficientes para realizar uma comparação temporal.

Essas regras garantem que todas as comparações apresentadas pela plataforma sejam realizadas com dados consistentes, evitando conclusões equivocadas decorrentes de avaliações incompletas.

5.3 Benchmark

A funcionalidade de *Benchmark*, ilustrada na Figura 13, tem como objetivo permitir que uma organização compare seu nível de adoção de práticas contínuas ao de outras empresas cadastradas na plataforma.

Para isso, a página disponibiliza quatro filtros que podem ser utilizados individualmente ou em conjunto para definir o perfil das organizações a serem comparadas. Todos os filtros utilizam as informações coletadas durante o cadastro da organização (Seção 5.1.1):

- **Section:** setor de atuação da organização;
- **Type:** tipo da organização;

- **Size:** quantidade de colaboradores, discretizada nas faixas 1–10, 11–50, 51–200, 201–500, 501–1000 e mais de 1000 funcionários;
- **Target Audience:** público-alvo da organização.

Após a definição dos filtros, a aplicação identifica todas as organizações que atendem aos critérios selecionados. Para cada organização encontrada, considera-se apenas o ciclo concluído mais recente. Em seguida, esses resultados são agregados para gerar o perfil médio do grupo selecionado, que é comparado ao ciclo atual da organização por meio de um gráfico de radar.

Durante o desenvolvimento dessa funcionalidade foram definidos três estados possíveis:

- **Dados insuficientes:** ocorre quando a organização atual ainda não possui nenhum ciclo concluído, impossibilitando qualquer comparação;
- **Coorte insuficiente:** para preservar o anonimato das organizações cadastradas, os resultados agregados são apresentados apenas quando pelo menos cinco organizações satisfazem os filtros selecionados. Caso contrário, o sistema informa que o grupo encontrado é insuficiente para gerar uma comparação anônima;
- **Comparação disponível:** quando ambas as condições anteriores são satisfeitas, é apresentado um gráfico de radar que compara a organização atual ao grupo filtrado.

A adoção dessas regras permite que a funcionalidade forneça comparações representativas sem comprometer a privacidade das organizações participantes. Além disso, o uso de filtros permite comparar empresas de perfis semelhantes, tornando os resultados mais relevantes para apoiar a tomada de decisão.

O componente de visualização em radar foi desenvolvido de forma reutilizável, sendo compartilhado entre as funcionalidades de Evolução Temporal e *Benchmark*. Essa abordagem reduziu a duplicação de código e garantiu uma experiência visual consistente entre as duas funcionalidades.

As funcionalidades apresentadas nesta seção ampliam significativamente as capacidades originalmente disponibilizadas pelo instrumento Zeppelin. Além de possibilitar o gerenciamento de organizações, a plataforma passou a oferecer mecanismos para o acompanhamento da evolução temporal das avaliações e para a comparação entre organizações com perfis semelhantes, fornecendo informações que apoiam a tomada de decisão e o planejamento da adoção de práticas contínuas.

6 Conclusão e Trabalhos Futuros

Este trabalho apresentou o desenvolvimento de funcionalidades para uma plataforma web³ baseada no instrumento Zeppelin, cujo objetivo é apoiar organizações na avaliação da adoção de práticas de Engenharia de Software Contínua.

³Código-fonte disponível em: <https://github.com/brenorobazza/zeppelin>

Partindo de um back-end já existente, foi desenvolvida uma aplicação web que centraliza a execução das avaliações, o gerenciamento de organizações, a visualização dos resultados e a disponibilização de recomendações personalizadas. No contexto deste trabalho, o foco esteve no desenvolvimento do módulo de gerenciamento de organizações e das funcionalidades de análise, responsáveis por acompanhar a evolução temporal das avaliações e possibilitar comparações entre organizações por meio da funcionalidade de *Benchmark*.

Além da implementação das funcionalidades, foram definidas regras de negócio para garantir a consistência dos dados apresentados, como validações de comparações temporais e mecanismos de anonimização nas comparações entre organizações. Essas decisões contribuíram para tornar a plataforma mais robusta e adequada para utilização em ambientes organizacionais.

Como resultado, a plataforma passou a oferecer uma experiência mais completa para o uso do instrumento Zeppelin, permitindo não apenas a realização de diagnósticos, mas também o acompanhamento da evolução das organizações e a análise de seus resultados em diferentes contextos.

Embora a plataforma já ofereça suporte às principais funcionalidades propostas, diversas melhorias podem ser incorporadas em trabalhos futuros. Uma das principais limitações do projeto está relacionada à validação da aplicação em ambientes reais. A realização de estudos com empresas utilizando a plataforma por períodos prolongados permitiria identificar *bugs*, validar as funcionalidades implementadas e capturar cenários de uso não previstos durante o desenvolvimento.

Outra possibilidade é a evolução da funcionalidade de acompanhamento temporal. Atualmente, a comparação é realizada entre dois ciclos específicos. Como melhoria, poderia ser disponibilizada uma visualização contínua da evolução da organização ao longo de todos os ciclos, utilizando gráficos de linhas ou outras representações temporais. Essa funcionalidade permitiria identificar tendências, períodos de evolução e regressão, facilitando a interpretação dos resultados.

Também podem ser exploradas novas funcionalidades de análise, como métricas estatísticas adicionais, novos indicadores de maturidade organizacional e formas mais avançadas de *benchmarking*, ampliando o suporte à tomada de decisão das organizações.

Ademais, uma possibilidade seria a incorporação de mecanismos inteligentes para a geração automática de recomendações, com base em modelos de Inteligência Artificial. Dessa forma, seria possível produzir recomendações ainda mais personalizadas, considerando o histórico da organização, sua evolução ao longo do tempo e o contexto específico de atuação.

Referências

- [1] Aline Dias de Araújo. Diagnosis and recommendations for ci/cd practices based on the zeppelin instrument. Dissertação de mestrado, Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP, Brazil, 2025.
- [2] Bruno B. N. de França, Hugo Jeronimo, and Guilherme H. Travassos. Characterizing devops by hearing multiple voices. In *Proceedings of the XXX Brazilian Symposium on*

- Software Engineering (SBES '16)*, pages 53–62, New York, NY, USA, 2016. Association for Computing Machinery.
- [3] Paulo Sérgio dos Santos Júnior, Monalessa Perini Barcellos, and Fabiano Borges Ruy. Tell me: Am i going to heaven? a diagnosis instrument of continuous software engineering practices adoption. In *Proceedings of the 25th International Conference on Evaluation and Assessment in Software Engineering (EASE '21)*, pages 30–39. Association for Computing Machinery, 2021.
 - [4] Paulo Sérgio dos Santos Júnior, Monalessa Perini Barcellos, Fabiano Borges Ruy, and Marcelo S. Omêna. 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 '22)*, pages 279–288, New York, NY, USA, 2022. Association for Computing Machinery.
 - [5] Brian Fitzgerald and Klaas-Jan Stol. Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, 123:176–189, 2017.
 - [6] Jez Humble and David Farley. *Continuous delivery: reliable software releases through build, test, and deployment automation*. Pearson Education, 2010.
 - [7] Jan Ole Johanssen, Anja Kleebaum, Barbara Paech, and Bernd Bruegge. Practitioners’ eye on continuous software engineering: An interview study. In *Proceedings of the 2018 International Conference on Software and System Process (ICSSP '18)*, pages 41–50, New York, NY, USA, 2018. Association for Computing Machinery.
 - [8] Helena Holmström Olsson and Jan Bosch. Climbing the ”stairway to heaven”: Evolving from agile development to continuous deployment of software. In *Continuous Software Engineering*, pages 15–27. Springer, 2014.
 - [9] André Portela and Breno Bernard Nicolau de França. Empirical evidence on technical challenges when adopting continuous practices. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*, pages 11–20, 2023.