

Em Direção ao Desenvolvimento de uma Ferramenta de Gerenciamento de Rubricas de Avaliação

Matheus Lemes Fialho¹, Maicon Bernardino¹, Erik Fontella¹, Fábio Paulo Basso¹

¹Universidade Federal do Pampa (UNIPAMPA)
Avenida Tiarajú, 810, Bairro Ibirapuitã – Alegrete – RS – Brazil

{matheusfialho,erikfontella}
.aluno@unipampa.edu.br, fabiobasso@unipampa.edu.br, bernardino@acm.org

Abstract. *Despite the importance of rubrics for fair assessment, existing software tools for their management often present technical and usability limitations. This paper presents the engineering behind RubricPro, a web tool we designed and developed to address these gaps. We detail its development process, design decisions, and its architecture, based on the SPA and BaaS pattern and documented with the C4 model. The tool was validated through iterative usability studies, which demonstrated a positive evolution, with the average System Usability Scale (SUS) score increasing from 66.25 to 71.25. The results confirm the effectiveness of the engineering approach and position RubricPro as a case study in creating modern educational software.*

Resumo. *Apesar da importância das rubricas para uma avaliação justa, as ferramentas de software para seu gerenciamento frequentemente apresentam limitações técnicas e de usabilidade. Este artigo apresenta a engenharia por trás da RubricPro, uma ferramenta web que projetamos e implementamos para superar tais lacunas. Detalhamos seu processo de desenvolvimento, as decisões de projeto e sua arquitetura, baseada no padrão SPA e BaaS e documentada com o modelo C4. A ferramenta foi validada por meio de estudos iterativos de usabilidade, que demonstraram uma evolução positiva, com o score médio do System Usability Scale (SUS) aumentando de 66,25 para 71,25. Os resultados confirmam a eficácia da abordagem de engenharia e posicionam a RubricPro como um caso de estudo na criação de software educacional moderno.*

1. Introdução

Segundo Biagiotti (2005), a avaliação dos discentes exige atenção e cautela, devendo, portanto, evidenciar clareza e equidade. Fernandes (2021) acrescenta que as rubricas são ferramentas eficazes para avaliação em diferentes níveis educacionais. Uma rubrica de avaliação é composta por um conjunto de critérios, cada um com descritores que definem os níveis de desempenho e qualidade do critério avaliado [Brookhart 2013]. As rubricas avaliativas contribuem para a entrega de um *feedback* claro ao discente, reduzindo a subjetividade do processo avaliativo [Biagiotti 2005, Fernandes 2021].

Ferramentas de apoio devem suportar tanto rubricas analíticas (que avaliam critérios individualmente) quanto holísticas (que avaliam o trabalho como um todo) [Brookhart 2013], além de oferecer flexibilidade, padronização e, crucialmente, o reúso de artefatos [Biagiotti 2005]. No entanto, uma análise do estado da arte e da prática

[Coelho 2024] revela um cenário de ferramentas com interfaces antiquadas, funcionalidades restritas e baixa flexibilidade [4Teachers.org 2024, Clever Prototypes 2024, Tech4Learning 2024].

Para endereçar as limitações identificadas, este trabalho apresenta a engenharia por trás da RubricPro, uma ferramenta web que oferece funcionalidades para a criação e personalização de rubricas, aplicação de avaliações e envio de *feedback* aos discentes. O conceito da ferramenta também prevê, para futuras versões, o compartilhamento de modelos entre usuários e um sistema de busca por *tags*, visando criar um ambiente onde o processo avaliativo seja mais claro e eficaz.

Este estudo está estruturado da seguinte forma: a Seção 2 discute trabalhos relacionados e ferramentas correlatas. A Seção 3 detalha a concepção, engenharia de requisitos, decisões de projeto, os protótipos da ferramenta RubricPro e seus resultados preliminares. Por fim, a Seção 4 apresenta as considerações finais e trabalhos futuros envolvendo a ferramenta.

2. Ferramentas e Trabalhos Relacionados

A seção aborda pesquisas sobre ferramentas e tecnologias envolvendo rubricas de avaliação. Estes trabalhos e ferramentas foram obtidos por meio de uma pesquisa *ad-hoc* na literatura cinzenta (*gray literature*) [Garousi et al. 2019], via Google.

Coelho (2024) apresenta uma comparação de ferramentas para rubricas, analisando suas características e limitações. Utilizando uma metodologia bibliográfica e qualitativa, o estudo mapeou e fez a análise comparativa das diversas ferramentas, destacando suas funções e restrições. Lobato *et al.* (2009) desenvolveram um módulo integrado ao LabSQL para gerenciamento de rubricas, que melhorou a precisão da avaliação e ajudou a identificar dificuldades dos discentes. No entanto, eles destacaram que a interface do protótipo, por sua complexidade, teve baixa aceitação.

Dornisch e McLoughlin (2006) exploraram as limitações de ferramentas online para a criação de rubricas, ressaltando que bancos de rubricas e geradores online frequentemente apresentam problemas como falta de flexibilidade para edição, descritores pouco claros e navegação complexa. O artigo também enfatiza a importância de rubricas bem estruturadas, com critérios relevantes e descritores claros, para garantir avaliações consistentes e *feedback* significativo.

As três pesquisas evidenciam tanto os benefícios quanto as limitações das ferramentas de gerenciamento de rubricas avaliativas, fornecendo uma base para a importância de novas soluções e para a melhoria das existentes, priorizando a usabilidade, a colaboração e a customização. Realizou-se também uma busca por ferramentas com funcionalidades semelhantes à ferramenta foco deste estudo de caso (ver Tabela 1). As limitações identificadas, como interfaces antiquadas e restrições funcionais (*e.g.*, foco exclusivo na criação e não na aplicação de rubricas), reforçaram a necessidade de uma solução como a *RubricPro*.

Considerando as limitações nas ferramentas apresentadas, nossa ferramenta visa fornecer uma interface intuitiva e adaptada às necessidades dos usuários, modelada para melhorar a experiência e a colaboração entre docentes, tanto na criação quanto na aplicação das rubricas.

Tabela 1. Síntese das Ferramentas e Tecnologias Correlatas.

Ferramenta	Pontos fortes	Limitações
Quick Rubric [Clever Prototypes 2024]	<ul style="list-style-type: none"> • Criação rápida e fácil de rubricas de avaliação; • Interface intuitiva para customização de critérios e níveis de desempenho; • Compartilhar rubricas por meio de uma URL personalizada; • Disponibilidade gratuita. 	<ul style="list-style-type: none"> • Disponível apenas em inglês; • Limitada à criação de rubricas e não à sua aplicação; • O cálculo da pontuação da rubrica é baseado na quantidade de níveis de desempenho e na pontuação máxima e mínima fornecidas.
RubiStar [4Teachers.org 2024]	<ul style="list-style-type: none"> • Criação de rubricas de avaliação baseadas em diversos tipos de modelos; • Base de dados própria com modelos pré-definidos; • Disponibilidade gratuita. 	<ul style="list-style-type: none"> • Interface desatualizada que pode ser menos intuitiva para os usuários atuais; • Disponível apenas em inglês e espanhol; • Limitada à criação de rubricas e não à sua aplicação; • O sistema de busca entre as rubricas criadas por outros usuários na ferramenta pode fornecer várias rubricas em branco.
Rubric Maker [Tech4Learning 2024]	<ul style="list-style-type: none"> • Interface intuitiva que facilita a utilização do usuário; • Oferece recomendações baseadas nos tópicos e subtópicos pré-definidos na ferramenta; • Possui um plano gratuito que permite a criação de rubricas; • Diversas funcionalidades de customização. 	<ul style="list-style-type: none"> • Disponível apenas em inglês; • Grande parte das funcionalidades de customização limitadas aos planos pagos; • Recursos avançados, como compartilhamento de rubricas, utilização de <i>templates</i> e adição de outros usuários, disponíveis apenas nos planos pagos.
RubricPro (Versão Alfa)	<ul style="list-style-type: none"> • Implementação inicial do ciclo completo de funcionalidades (criação, aplicação, feedback). 	<ul style="list-style-type: none"> • Baixa usabilidade percebida em estudos iniciais (SUS de 66,25); • Fluxo de navegação e autenticação apresentavam problemas.
RubricPro (Versão Beta)	<ul style="list-style-type: none"> • Manutenção do ciclo completo de funcionalidades com foco em UX; • Flexibilidade na criação de avaliações (individuais/grupos); • Interface moderna com usabilidade validada como “Boa” (SUS de 71,25). • Gerenciamento de turmas com importação de alunos via CSV (Modelo); 	<ul style="list-style-type: none"> • Compartilhamento e Busca de modelos de rubricas é uma funcionalidade futura; • Integração dinâmica com AVAs (além de CSV) não implementada; • Acesso e funcionalidades específicas para discentes não disponíveis na versão atual; • Módulos de estatísticas, gamificação e IA em desenvolvimento/planejamento.

3. Ferramenta Proposta

3.1. Contexto do Projeto

A ferramenta RubricPro, cujo ciclo de concepção, desenvolvimento e avaliação é o foco deste trabalho, foi desenvolvida como um sistema web para melhorar o processo avaliativo, reduzir a subjetividade e oferecer uma ferramenta customizável para os métodos de avaliação dos docentes.

Para atingir seus objetivos, a RubricPro foi projetada com um conjunto de funcionalidades centrais. Estas incluem um sistema para o gerenciamento de turmas e discentes, um editor para a criação de modelos de rubricas altamente personalizáveis, e mecanismos para a configuração e aplicação de avaliações, seja para indivíduos ou para grupos. Adicionalmente, a ferramenta contempla funcionalidades de colaboração, como o compartilhamento e a busca de modelos de rubricas por meio de *tags*, e um sistema para automatizar o envio de notificações por *e-mail* contendo a avaliação detalhada e o *feedback* do docente ao discente.

3.2. Engenharia de Requisitos

As funcionalidades da ferramenta foram sumarizadas em 20 Requisitos de Software (RS), especificados na forma de histórias de usuários (*User Stories - US*), majoritariamente para docentes, com exceção de duas US (US06 e US13) para discentes e sete (7) Requisitos Não Funcionais (RNF), destacando a usabilidade e escalabilidade, visando assegurar uma experiência de usuário positiva e um sistema robusto que suporte o crescimento e a diversidade de uso. Para priorização dos requisitos funcionais, utilizou-se o método *MoSCoW*, uma técnica amplamente empregada que categoriza os requisitos em quatro níveis de prioridade [Clegg and Barker 1994].

A ferramenta visa auxiliar os docentes com requisitos como a integração com Ambientes Virtuais de Aprendizagem (AVA) (US17) e a melhoria da experiência do usuário (RNF01, RNF02 e RNF07). Além disso, a ferramenta promove a colaboração entre docentes, por meio do compartilhamento e busca de modelos de rubricas, com foco na disseminação de conhecimento. Os requisitos funcionais podem ser vistos na Tabela 2 e os requisitos não funcionais na Tabela 3.

3.3. Decisões de Projeto

Nesta seção discutimos cada uma das Decisões de Projeto (DP) estabelecidas durante o planejamento da ferramenta.

- DP1. Arquitetura de Software:** A arquitetura adotada foi *Single Page Application* (SPA), que segue o modelo Cliente-Servidor. Segundo Agouf *et al.* (2023), a abordagem Cliente-Servidor envolve uma estrutura distribuída, em que os clientes enviam requisições de conteúdo ou serviços aos servidores. Neste caso, nossa função será a de servidor, sendo responsáveis pelo processamento de dados, lógica de negócio e persistência de informações [Agouf et al. 2023]. Em conjunto, a arquitetura SPA trata as páginas de maneira dinâmica. Após o primeiro carregamento por parte do cliente, os dados são atualizados por meio de requisições ao servidor, sem a necessidade de recarregar a página. Isso torna a ferramenta mais fluida e rápida, gerando uma melhor experiência de usuário. Essa arquitetura simplifica a manutenção e facilita a escalabilidade do sistema [Kornienko et al. 2021];
- DP2. Ferramenta Web:** Por conta da maior versatilidade e facilidade de acesso, a solução foi desenvolvida para a plataforma web;
- DP3. Persistência de Dados:** Foi decidida a utilização de um banco de dados não relacional (NoSQL) por meio do serviço *Google Firebase*. A alta escalabilidade e disponibilidade oferecidas por um banco de dados não relacional são mais adequadas à ferramenta, devido à necessidade de customização nas rubricas de avaliação e à possível escalabilidade do sistema;

Tabela 2. Lista das Estórias de Usuários da RubricPro.

ID	Estórias de Usuários	P	S
US01	Como Docente eu quero gerenciar minhas turmas para que eu possa organizar as disciplinas ativas.	MUST	✓
US02	Como Docente eu quero adicionar discentes as minhas turmas para que eu possa avaliá-los posteriormente.	MUST	✓
US03	Como Docente eu quero criar rubricas de avaliação customizadas para que eu possa definir os pontos concretos na avaliação de um discente.	MUST	✓
US04	Como Docente eu quero gerenciar as rubricas de avaliação de cada turma para que eu possa verificar o quadro geral das avaliações.	MUST	✓
US05	Como Docente eu quero avaliar cada discente usando as rubricas criadas para que o discente receba o devido <i>feedback</i> de sua avaliação.	MUST	✓
US06	Como Discente eu quero visualizar a minha nota perante uma rubrica de avaliação para que eu possa compreender os pontos nos quais fui avaliado.	MUST	✓
US07	Como Docente eu quero notificar os discentes sobre suas notas perante uma rubrica de avaliação para que eu possa manter os mesmos informados quanto ao andamento de suas avaliações.	SHOULD	✓
US08	Como Docente eu quero verificar o desempenho da turma em cada rubrica de avaliação para que eu possa compreender possíveis necessidades de mudança nos métodos de ensino.	COULD	✗
US09	Como Docente eu quero gerenciar minhas avaliações para que eu possa alterar dados que tenham sido afetados por fatores externos.	SHOULD	✓
US10	Como Docente eu quero buscar modelos de rubricas a partir de <i>tags</i> para que o tempo de busca seja mais flexível e reduzido.	SHOULD	✗
US11	Como Docente eu quero enviar uma nova notificação da reavaliação realizada para um discente para que eu possa notificar o mesmo de maneira individual ao reavaliar seu trabalho.	SHOULD	✓
US12	Como Docente eu quero criar grupos para agrupar as avaliações para que eu possa aplicar uma rubrica de avaliação em um grupo de discentes.	SHOULD	✓
US13	Como Discente eu quero notificar o docente que não estou de acordo com a avaliação atribuída para que eu possa recorrer e discutir sobre uma reavaliação.	SHOULD	✗
US14	Como Docente eu gostaria de descrever um parecer na avaliação para que o discente compreenda melhor como foi feita sua avaliação.	SHOULD	✓
US15	Como Docente eu quero escrever anotações na avaliação, exclusiva aos docentes, para que eu tenha um controle e organização mais efetivo das minhas decisões avaliativas.	SHOULD	✓
US16	Como Docente eu quero compartilhar minha avaliação com outros docentes para que o mesmo também possa aplicar esta avaliação em caso de disciplinas compartilhadas.	COULD	✓
US17	Como Docente eu quero exportar os dados das avaliações realizadas para que eu possa importar estes dados em ambientes virtuais de aprendizagem.	SHOULD	✓
US18	Como Docente eu quero refazer a avaliação de um discente para que eu possa corrigir enganos ocorridos na avaliação inicial.	MUST	✓
US19	Como Docente eu quero visualizar dicas flutuantes referentes as funcionalidades da ferramenta para que a minha experiência na ferramenta se torne mais efetiva.	SHOULD	✗
US20	Como Docente eu quero confirmar alterações e exclusões na customização de rubricas para que eu possa evitar erros e exclusões indesejadas ao confeccionar uma rubrica.	MUST	✓

Legenda: ✓= US Implementado. — ✗= US em andamento. — ✗= US não iniciado.

Tabela 3. Lista dos Requisitos Não Funcionais da RubricPro.

ID	Requisitos Não Funcionais
RNF01	A ferramenta deve ter uma interface intuitiva e amigável que auxilie o docente.
RNF02	A ferramenta deve ser responsiva, funcionando corretamente em diferentes tamanhos de tela de computadores.
RNF03	O tempo de carregamento das páginas da ferramenta deve ser inferior a 5 segundos.
RNF04	A ferramenta deve suportar um mínimo de 1000 usuários simultâneos sem degradação de performance.
RNF05	A ferramenta deve garantir a privacidade e segurança dos dados dos usuários, em conformidade com a LGPD.
RNF06	A ferramenta deve ter capacidade de integração com o AVA Moodle e outras plataformas educacionais.
RNF07	A ferramenta deve disponibilizar indicadores visuais, textuais e figurativos para garantir acessibilidade.

DP4. Modelo de Negócio: O modelo de negócio adotado é o Software como Serviço (SaaS), em que Ouchaou *et al.* (2022) destacam que toda a infraestrutura necessária para a disponibilização do sistema é nossa responsabilidade, permitindo que a ferramenta seja utilizada pelos clientes como um serviço na web;

DP5. Múltiplos Idiomas: A aplicação oferecerá suporte a múltiplos idiomas, visando uma maior aceitação e utilização da ferramenta em diferentes regiões. Atualmente, estando disponível em português, inglês e espanhol. Essa escolha busca otimizar o tempo de manutenção no futuro, caso o projeto se expanda internacio-

nalmente [Reynolds 2020];

DP6. Linguagem: Para o desenvolvimento da ferramenta, decidimos a utilização da linguagem *JavaScript*, utilizando o superconjunto *TypeScript* agregado com a biblioteca *Svelte*. Optamos por essas tecnologias devido à facilidade de tipagem oferecida pelo *TypeScript* [Scarsbrook et al. 2023] e à simplicidade que o *Svelte* [Svelte 2024] traz ao código, além de melhorar o tempo de resposta;

DP7. Framework: Para melhor efetividade do *Svelte*, utilizamos o *Framework SvelteKit*, cujo objetivo é o desenvolvimento de ferramentas web robustas e de alto desempenho, permitindo uma criação rápida, flexível e simples [Svelte 2024].

3.4. Modelagem e Prototipação

Pressman e Maxim (2019) comentam que protótipos das interfaces devem ser produzidos como parte da modelagem de requisitos, auxiliando na avaliação inicial e garantindo que o *design* do sistema esteja consistente, intuitivo e efetivo. Para cumprir esse propósito, foi utilizada a ferramenta Figma¹, que permite a elaboração de protótipos simples até os de alta fidelidade, e também uma pré-visualização do *design* como um *website* estático.

O desenvolvimento da RubricPro ocorreu em duas fases principais. A primeira, denominada RubricPro Alfa, foi prototipado em 2024, forneceu uma base funcional para o sistema, a partir de análises da versão inicial e de *feedbacks* coletados, identificou-se a necessidade de um *redesign* completo para aprimorar a usabilidade e a experiência do usuário. A segunda versão, denominada RubricPro Beta, foi o resultado desse projeto de *redesign*. Os novos protótipos foram construídos com foco em um *design* mais moderno e uma arquitetura de navegação mais intuitiva.

Como resultado desse processo, o protótipo e a implementação foram significativamente aprimorados. A Figura 1 ilustra o estado atual da versão Beta, apresentando as interfaces de gerenciamento de turmas e de criação de rubricas.

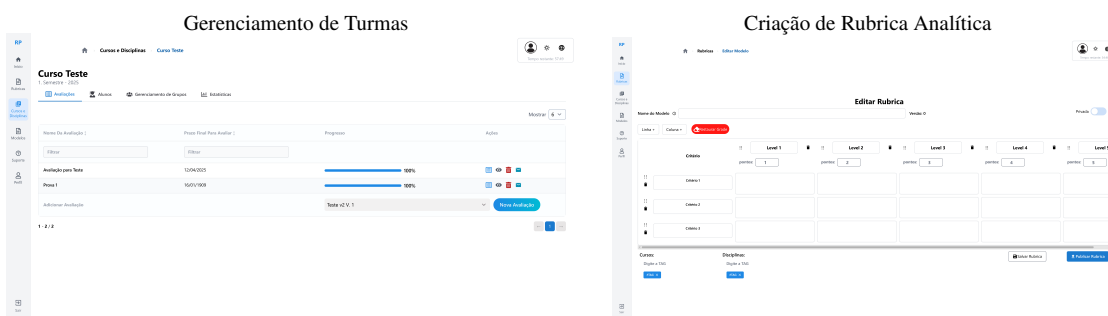


Figura 1. Rubric Pro Beta

3.5. Arquitetura do Software

Para documentar e comunicar a arquitetura da RubricPro, adotou-se o modelo C4 (*Context, Containers, Components, and Code*), que permite visualizar o sistema em diferentes níveis de abstração [Vázquez-Ingelmo et al. 2020, Brown 2020]. Esta abordagem facilita o entendimento do sistema tanto do ponto de vista do usuário quanto da perspectiva técnica de implementação. As seções seguintes detalham os dois primeiros níveis da arquitetura.

¹Figma: <https://www.figma.com/pt-br/>

3.5.1. Nível 1: Diagrama de Contexto do Sistema

O Diagrama de Contexto, apresentado na Figura 2, delimita a fronteira do sistema RubricPro e suas interações de alto nível. Neste nível, o sistema é tratado como uma “caixa preta”. O Professor é o único usuário humano que interage diretamente com o sistema, utilizando-o para gerenciar todo o ciclo de vida das avaliações. A RubricPro depende de quatro sistemas externos para operar: (i) **Firestore Authentication**: Utilizado como um provedor de identidade externo para gerenciar o login dos usuários via Google; (ii) **Firestore Firestore**: Banco de dados principal da aplicação, persistindo todas as informações em um modelo NoSQL; (iii) **Serviço de Email**: Serviço responsável pelo envio de notificações de *feedback* aos discentes; (iv) **Sistema de Chatbot de Ajuda**: Uma API que fornece suporte ao usuário, respondendo a perguntas sobre o funcionamento do sistema. Este diagrama estabelece o escopo e as principais dependências tecnológicas do projeto.

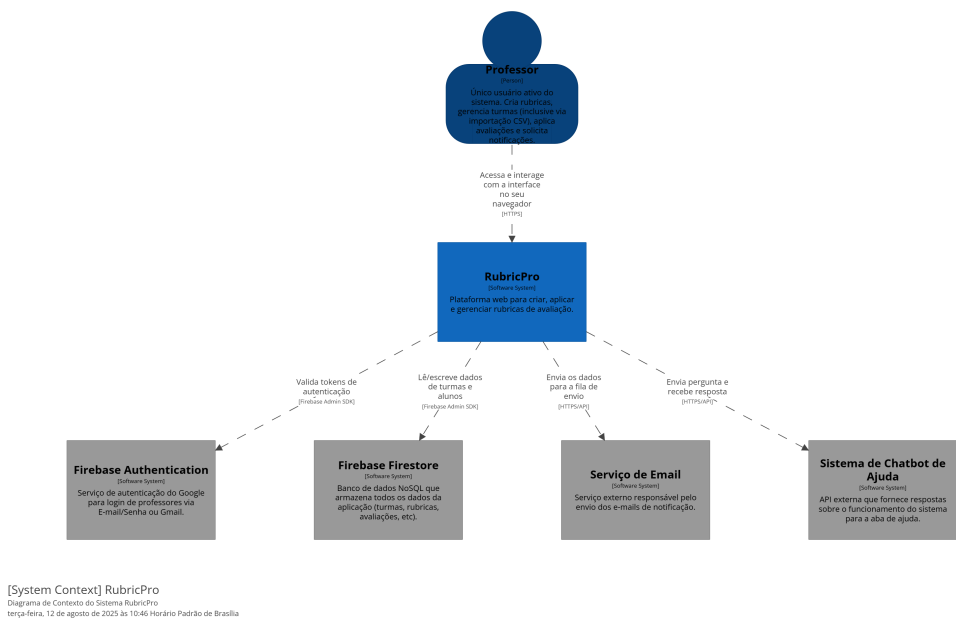


Figura 2. Nível 1 (Contexto) da Arquitetura C4 da RubricPro.

3.5.2. Nível 2: Diagrama de Contêineres

A Figura 3 amplia o sistema RubricPro, revelando seus principais blocos de construção arquiteturais, ou contêineres. A implementação segue uma arquitetura cliente-servidor moderna, composta por dois contêineres principais: (i) **Single-Page Application (SPA)**: É o contêiner do lado do cliente, uma aplicação web construída com SvelteKit que roda diretamente no navegador do usuário. A SPA é responsável por toda a interface do usuário, interatividade e renderização dinâmica do conteúdo; (ii) **Backend API (SvelteKit Server)**: É o contêiner do lado do servidor, também desenvolvido com SvelteKit. Ele centraliza a lógica de negócio, processa as requisições da SPA, valida a autenticação dos usuários com o *Firestore Authentication* e orquestra a comunicação com os demais serviços (*Firestore*,

Email, Chatbot). Este modelo desacoplado permite que a interface (SPA) e a lógica de negócio (*Backend API*) sejam desenvolvidas e mantidas de forma independente, além de garantir uma experiência de usuário fluida, característica das SPAs.

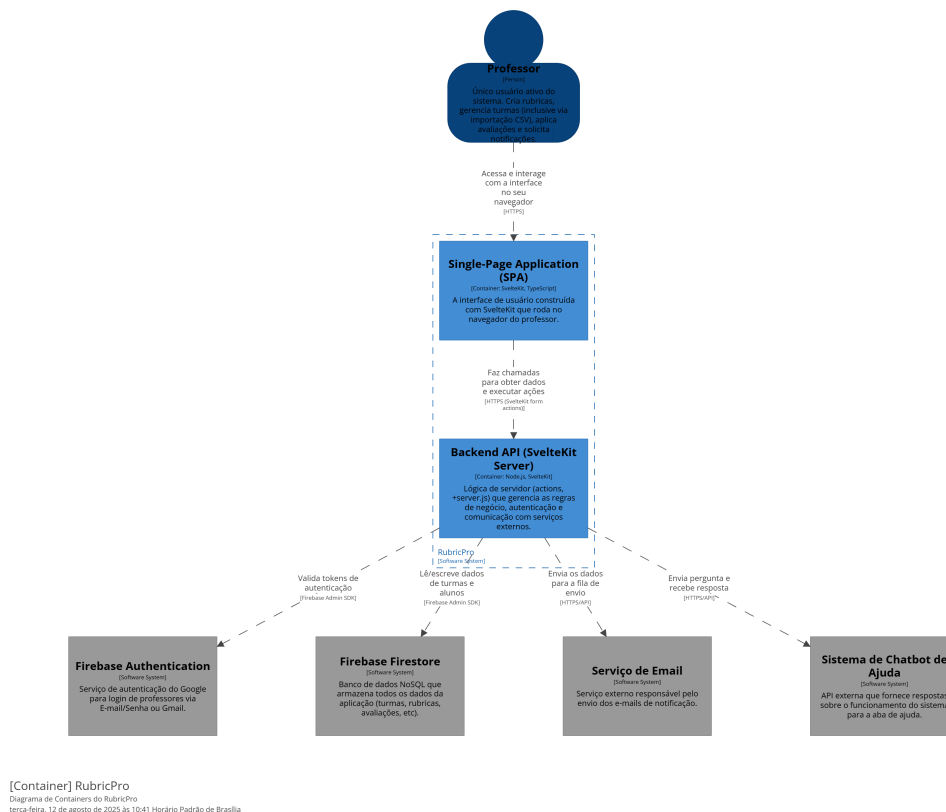


Figura 3. Nível 2 (Contêineres) da Arquitetura C4 da RubricPro.

3.6. Resultados Preliminares

Para validar a eficácia das decisões de arquitetura e tecnologia, a ferramenta foi submetida a um processo de avaliação iterativo, que incluiu um grupo focal com protótipos e um estudo de caso comparativo entre as versões Alfa e Beta. Os principais resultados que validam o artefato são sumarizados a seguir. Maiores detalhes sobre a metodologia e resultados do grupo focal foram aceitos para publicação no Simpósio Brasileiro de Informática na Educação (SBIE).

A validação quantitativa demonstrou uma evolução positiva na usabilidade percebida pelos docentes. A pontuação média do *System Usability Scale* (SUS), um indicador padrão de usabilidade, aumentou de **66,25 (versão Alfa)** para **71,25 (versão Beta)**, movendo a classificação da ferramenta de “Aceitável” para “Boa” e superando a média de referência de 68 [Brooke 1996].

Qualitativamente, a análise do *feedback* dos docentes confirmou essa evolução. As discussões transitaram de críticas sobre problemas básicos de usabilidade na versão Alfa para sugestões de refinamentos e funcionalidades avançadas na versão Beta, indicando a maturação e a estabilidade do sistema. Adicionalmente, um estudo com 33 discentes

revelou uma percepção majoritariamente positiva sobre a clareza do *feedback* recebido por meio da ferramenta. Estes resultados validam que a arquitetura e as tecnologias escolhidas foram capazes de suportar o desenvolvimento de uma ferramenta não apenas funcional, mas também com uma experiência de usuário positiva e em clara evolução.

4. Considerações Finais

Este artigo detalhou o processo de engenharia de software por trás da ferramenta Rubric-Pro, uma solução desenvolvida para superar as limitações técnicas e de usabilidade de ferramentas de rubricas existentes. As principais contribuições deste trabalho são: (i) a apresentação de uma arquitetura de software moderna, baseada no padrão SPA e BaaS (Backend as a Service); (ii) a documentação de um processo de desenvolvimento estruturado, com engenharia de requisitos e um registro de decisões de projeto; (iii) a validação do artefato por meio de estudos que demonstraram uma evolução positiva na usabilidade, confirmando a eficácia das escolhas de engenharia. A RubricPro se apresenta, portanto, como um caso de estudo da aplicação de tecnologias web contemporâneas para a criação de software educacional robusto e centrado no usuário.

Como trabalhos futuros, o foco técnico será na expansão das funcionalidades da ferramenta. Planeja-se a implementação dos requisitos de colaboração, como o compartilhamento de modelos de rubricas, o que exigirá o projeto de um modelo de permissões e o desenvolvimento de novas APIs no *backend*. Outra frente de pesquisa e desenvolvimento envolve a integração de modelos de Inteligência Artificial para auxiliar na geração de *feedback* textual, além da condução de testes de carga para avaliar a escalabilidade da arquitetura proposta sob uso intensivo.

Agradecimentos

Os autores agradecem à Universidade Federal do Pampa (UNIPAMPA) e à Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) pelo apoio financeiro e institucional que tornaram esta pesquisa possível.

Referências

- 4Teachers.org (2024). RubiStar. Disponível em: <http://rubistar.4teachers.org/index.php>. Acesso em: 12/08/2025.
- Agouf, N. J., Labsari, S., Ducasse, S., Etien, A., and Anquetil, N. (2023). A visualization for client-server architecture assesment. In *2023 IEEE Working Conference on Software Visualization (VISSOFT)*, pages 1–11. IEEE.
- Biagiotti, L. C. M. (2005). Conhecendo e aplicando rubricas em avaliações. In *Congresso Brasileiro de Educação a Distância*, volume 12, pages 1–9.
- Brooke, J. (1996). Sus: A quick and dirty usability scale. *Usability Evaluation in Industry*, pages 189–194.
- Brookhart, S. M. (2013). *How to create and use rubrics for formative assessment and grading*. Ascd.
- Brown, S. (2020). *The C4 model for visualising software architecture*. Leanpub. Acessado em: 12/08/2025.

- Clegg, D. and Barker, R. (1994). *Case Method Fast-Track: A Rad Approach*. Addison-Wesley Longman Publishing Co., Inc., USA.
- Clever Prototypes (2024). Quick Rubric. Disponível em: <https://www.quickrubric.com>. Acesso em: 12/08/2025.
- Coelho, I. M. W. S. (2024). Estudo comparativo de ferramentas para construção de e-rubricas aplicadas à avaliação. *Ensino e Tecnologia em Revista*, 7(3):100–115.
- Dornisch, M. M. and McLoughlin, A. S. (2006). Limitations of web-based rubric resources: Addressing the challenges. *Practical Assessment, Research & Evaluation*, 11(1):3.
- Fernandes, D. (2021). Rubricas de avaliação. *Critério*, 1(2):3.
- Garousi, V., Felderer, M., and Mäntylä, M. V. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology*, 106:101–121.
- Kornienko, D. V., Mishina, S. V., and Melnikov, M. O. (2021). The single page application architecture when developing secure web services. *Journal of Physics: Conference Series*, 2091(1):012065.
- Lobato, A. S., Brito, S., Souza, D., and Favero, E. L. (2009). Um sistema gerenciador de rubricas para apoiar a avaliação em ambientes de aprendizagem. *Anais do XX Simpósio Brasileiro de Informática na Educação, Florianópolis/SC*.
- Ouchaou, L., Nacer, H., and Labba, C. (2022). Towards a distributed saas management system in a multi-cloud environment. *Cluster Computing*, 25(6):4051–4071.
- Pressman, R. and Maxim, B. (2019). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education.
- Reynolds, C. (2020). The benefits of translating your website into other languages. Disponível em: <https://tech.co/news/benefits-translating-website-languages-2015-07>. Acesso em: 21/06/2025.
- Scarsbrook, J. D., Utting, M., and Ko, R. K. L. (2023). Typescript's evolution: An analysis of feature adoption over time. In *2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR)*, pages 109–114.
- Svelte (2024). Introduction, Docs, SvelteKit. Disponível em: <https://kit.svelte.dev/docs/introduction#sveltekit-vs-svelte>. Acesso em: 12/08/2025.
- Tech4Learning (2024). Rubric Maker. Disponível em: <https://rubric-maker.com>. Acesso em: 12/08/2025.
- Vázquez-Ingelmo, A., García-Holgado, A., and García-Peñalvo, F. J. (2020). C4 model in a software engineering subject to ease the comprehension of uml and the software. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 919–924.