

Engenharia Reversa no Site Cronos do IFPR

André Ricardo Zavan¹, Helon Bentes Bastos Xavier¹

¹Instituto Federal do Paraná (IFPR)
Paranavaí, PR – Brasil

andre.zavan@ifpr.edu.br, helonbentes51@hotmail.com

Nos últimos anos, os sistemas web têm se tornado essenciais no apoio às atividades acadêmicas, administrativas e corporativas, proporcionando agilidade na execução de tarefas e centralizando o acesso às informações. No meio educacional, essas plataformas desempenham papel estratégico na gestão de cursos, matrículas, eventos e atividades complementares, permitindo que alunos, professores e gestores interajam de forma eficiente. Entretanto, muitos desses sistemas apresentam limitações relacionadas à escalabilidade, manutenibilidade e, sobretudo, à falta de documentação técnica formal. A ausência de registros estruturados dificulta o entendimento do funcionamento interno e compromete tanto a manutenção quanto a evolução de longo prazo. Nesse contexto, a aplicação de técnicas de engenharia reversa torna-se uma estratégia relevante para compreender, registrar e representar sistemas já desenvolvidos, extraíndo seus requisitos e modelando-os de maneira a auxiliar futuros desenvolvedores [Chikofsky and Cross 1990].

O presente trabalho toma como estudo de caso o sistema Cronos, utilizado pelo Instituto Federal do Paraná (IFPR) - Campus Paranavaí para o gerenciamento das Atividades Complementares. Embora seja muito usado pelos cursos da instituição, o Cronos não conta com documentação técnica, o que representa um risco para sua continuidade. Considerando que professores, alunos e desenvolvedores entram e saem da instituição levando consigo o conhecimento acumulado, a inexistência de documentação amplia a dificuldade de manutenção. De acordo com [Sommerville 2011], a documentação de requisitos e do comportamento do software é fundamental para a comunicação entre os envolvidos, servindo de base para testes, validações e futuras evoluções. [Pressman 2011] complementa que o software moderno, pela sua crescente complexidade, deve ser sistematicamente arquitetado e mantido sob princípios sólidos de engenharia.

Dessa forma, documentar o Cronos por meio de técnicas de engenharia reversa apresenta benefícios institucionais, técnicos e educacionais, contribuindo para a preservação do conhecimento, para a sustentabilidade do sistema e para o fortalecimento da prática acadêmica em engenharia de software. Na fundamentação teórica, destacam-se os conceitos de Engenharia de Software, Engenharia de Requisitos e Engenharia Reversa. A primeira, segundo [Sommerville 2011], abrange todas as fases do ciclo de vida do software, da especificação à manutenção. A Engenharia de Requisitos é a disciplina responsável por elicitar, analisar e documentar necessidades dos *stakeholders*, abrangendo requisitos funcionais — que descrevem serviços e comportamentos que o sistema deve oferecer — e requisitos não funcionais, que tratam de atributos de qualidade como desempenho, segurança, confiabilidade, compatibilidade e usabilidade.

Já a Engenharia Reversa é definida por [Chikofsky and Cross 1990] como o processo de analisar um sistema para identificar seus componentes e suas inter-relações, criando representações em nível superior de abstração. Outro conceito relevante é o de

sistemas legados, termo utilizado para descrever softwares antigos ou já consolidados, que continuam em operação por serem críticos para os processos institucionais, mas que muitas vezes apresentam código obsoleto, ausência de documentação e dificuldades de manutenção. [Cognin 1999] ressalta que os sistemas legados, apesar de sua importância, impõem altos custos de atualização e evolução, pois a falta de registros técnicos dificulta a compreensão de sua lógica e arquitetura. [Schirru et al. 2014] destacam ainda que os sistemas legados representam não apenas um desafio técnico, mas também organizacional e legal, dado que muitas instituições dependem de sua continuidade para garantir serviços essenciais.

Nesse sentido, a aplicação de técnicas de engenharia reversa torna-se uma abordagem eficaz para compreender, registrar e dar suporte à manutenção de sistemas legados, prolongando sua vida útil e reduzindo riscos de descontinuidade. Além disso, a modelagem com *Unified Modeling Language* (UML) possibilita a geração de diversos diagramas, dos quais, para a documentação estão sendo utilizados os diagramas de classes e de casos de uso, que cumprem papéis distintos: enquanto os diagramas de classes oferecem uma visão detalhada da estrutura do sistema, o que é útil para desenvolvedores, os diagramas de casos de uso facilitam a comunicação com *stakeholders*, representando de forma simplificada as interações entre usuários e o sistema.

Considerando esse conjunto de conceitos, torna-se evidente a necessidade de aplicar a engenharia reversa em sistemas institucionais como o Cronos. A utilização da UML e da elicitação de requisitos, discutidas na fundamentação teórica, orientam diretamente a análise do sistema em questão, permitindo transformar observações práticas em documentação estruturada. O objetivo principal deste trabalho é aplicar técnicas de engenharia reversa ao Cronos, de modo a extrair e registrar seus requisitos funcionais, não funcionais e regras de negócio, além de produzir diagramas de casos de uso e diagramas de classes em UML. Entre os objetivos específicos estão o levantamento detalhado da interface e dos fluxos de navegação, a análise do comportamento do sistema, a modelagem dos requisitos, a elaboração de documentação técnica estruturada e a validação parcial dos artefatos produzidos.

A abordagem adotada é qualitativa, descritiva e documental, baseada na observação direta da interface e funcionalidades do Cronos. Por meio dessa análise, foi possível identificar como os diferentes perfis de usuários — alunos, professores e membros da secretaria — interagem com a plataforma e quais são as restrições institucionais que norteiam seu funcionamento. O desenvolvimento do trabalho tem seguido um processo organizado em etapas.

Primeiramente, iniciou-se a extração de requisitos, observando as ações permitidas aos usuários, as limitações, os processos de autenticação e os mecanismos de validação das horas complementares. Foram identificados até o momento 44 requisitos funcionais, distribuídos por módulos, como módulo de horas complementares, inventário, etc. Os requisitos funcionais identificados são funcionalidades como autenticação, recuperação de senha, cadastro de alunos e acompanhamento de certificados, validação de horas, visualização de grupos de atividades complementares, etc. Foram ainda extraídos 14 requisitos não funcionais, relacionados ao desempenho, à segurança por meio de processos de autenticação e criptografia, à compatibilidade com navegadores e dispositivos, bem como à usabilidade. Além disso, foram mapeadas 7 regras de negócio, como os tipos

de atividades aceitas, os limites de carga horária, os critérios de aprovação pela secretaria e os papéis atribuídos a alunos, professores e membros da secretaria.

Com base nesses dados, foram elaborados os diagramas de caso de uso e de classes (Apêndice A) utilizando a ferramenta PlantUML, que permite a construção visual de representações estruturadas. Os diagramas de caso de uso ilustram as interações de cada perfil de usuário com o sistema, como cadastro e validação de certificados, enquanto os diagramas de classes apresentam a estrutura estática do Cronos, com classes, atributos, métodos e relacionamentos. A documentação foi organizada em um *template* padronizado, seguindo boas práticas de elicitação e registro, de modo a garantir clareza, consistência, completude e organização, critérios estes recomendados na literatura de Engenharia de Software [Sommerville 2011, Pressman 2011, Valente 2024].

Os resultados parciais já demonstram que a aplicação da engenharia reversa tem se mostrado eficaz na produção de documentação técnica para sistemas legados ou que não tiveram registros formais durante seu desenvolvimento. A geração de requisitos bem estruturados, regras de negócio e diagramas UML, oferece uma base sólida para manutenção e evolução do Cronos, além de representar um exercício acadêmico importante para estudantes de Engenharia de Software.

Em resumo, a engenharia reversa aplicada ao Cronos não apenas resolve um problema prático de manutenibilidade e evolução do sistema, mas também fortalece práticas de engenharia de requisitos e demonstra o valor da modelagem e da documentação. Essa experiência confirma que a documentação técnica, mesmo quando construída a posteriori, é capaz de promover benefícios institucionais e acadêmicos duradouros, ampliando a clareza sobre o sistema, facilitando sua manutenção e evolução e garantindo que o conhecimento sobre seu funcionamento permaneça registrado de forma acessível.

Referências

- Chikofsky, E. J. and Cross, II, J. H. (1990). Reverse engineering and design recovery: A taxonomy. *IEEE Software*, 7(1):13–17.
- Cognin, S. (1999). Sistemas legados: a engenharia reversa pode ser a solução? Dissertação de mestrado em engenharia de produção, Universidade Federal de Santa Catarina, Florianópolis.
- Pressman, R. S. (2011). *Engenharia de Software: Uma Abordagem Profissional*. AMGH Editora Ltda., 7 edition.
- Schirru, L., Fleury, A., and Yu, A. S. O. (2014). A Engenharia Reversa de Software no Brasil: Análise sob o Ponto de Vista Legal e Técnico. In *Anais do XI Encontro Acadêmico de Propriedade Intelectual, Inovação e Desenvolvimento (ENAPID)*, Florianópolis, SC.
- Sommerville, I. (2011). *Engenharia de Software*. Pearson Addison Wesley, São Paulo, 9 edition.
- Valente, M. T. (2024). Capítulo 3: Requisitos — engenharia de software moderna. <https://engsoftmoderna.info/cap3.html>. Acesso em: 28 ago. 2024.

APÊNDICE A – Requisitos e Diagramas - Engenharia Reversa Cronos

<https://github.com/Helon21/artefatos-cronos>