



SCHOOLPLANNER: um aplicativo para tarefas escolares

Leandro A. da SILVA¹; Paulo C. dos SANTOS²

RESUMO

O software SchoolPlanner apresenta uma solução digital para lidar com a falta de organização e motivação dos alunos em completar suas tarefas escolares, afetando seu desempenho acadêmico. O objetivo foi utilizar a tecnologia de forma positiva, integrando um sistema de gamificação ao software de gerenciamento escolar para motivar os alunos a concluírem suas tarefas e se envolver mais ativamente no ambiente escolar. O projeto utilizou conceitos de Engenharia de Software, como arquitetura, prototipagem e análise de requisitos, e segue a metodologia Scrum para um desenvolvimento ágil. Embora haja limitações, os resultados até o momento foram satisfatórios.

Palavras-chave: Escola, Gamificação, Software.

1. INTRODUÇÃO

SchoolPlanner é um software que possui como público-alvo os estudantes, principalmente aqueles que enfrentam dificuldades em completar suas tarefas escolares devido à falta de organização ou motivação. Com base em algumas pesquisas, houve a proposta da criação de um software de gerenciamento escolar integrado a um sistema de gamificação. Esse sistema de gamificação incentiva os alunos a completarem suas tarefas, fornecendo pontos ao cumpri-las.

Um grande problema identificado nas escolas foi a falta de organização e motivação dos alunos para concluírem suas tarefas escolares. Sendo assim, o objetivo do software é utilizar a tecnologia, integrando um sistema de gamificação, a fim de motivar os alunos a concluírem suas tarefas e se envolver mais ativamente no ambiente escolar.

Para o desenvolvimento do software, foram seguidos importantes conceitos teóricos da Engenharia de Software, como arquitetura de software, prototipagem de telas, levantamento e análise de requisitos. A união desses conceitos contribuem para a criação organizada de um software robusto e de fácil manutenção (Pressman, 2009).

Além dos conceitos abordados, foram feitas pesquisas relacionadas ao uso da tecnologia nas escolas. Por exemplo, um estudo na Universidade de Harvard, demonstrou que os jogos educacionais melhoram a motivação dos alunos, aumentando seu engajamento e interesse nas atividades escolares (GEE, 2017). Outro estudo realizado na Universidade de Rochester, demonstra que a gamificação é uma forma eficaz de motivar os alunos, pois o uso de jogos em sala de aula pode melhorar o desempenho acadêmico dos alunos e aumentar a participação na aula (DICHEVA, 2015).

¹ Discente, IFSULDEMINAS - Campus Muzambinho. E-mail: leandro.adrian@alunos.ifsuldeminas.edu.br.

² Orientador, IFSULDEMINAS - Campus Muzambinho. E-mail: paulo.santos@muz.ifsuldeminas.edu.br.

2. MATERIAL E MÉTODOS

O desenvolvimento do software seguiu conceitos presentes na Engenharia de Software, como arquitetura de software, prototipagem, levantamento, análise e modelagem UML, metodologia Scrum e documentação de requisitos, além de diversas pesquisas em relação ao uso da tecnologia no ambiente escolar.

O processo de desenvolvimento foi dividido da seguinte maneira: primeiramente, foi feito o levantamento dos requisitos, que consistem nas necessidades dos usuários que o software deve atender. Um exemplo de requisito é o gerenciamento de tarefas, que inclui funções como adicionar, editar, consultar e excluir tarefas. Para esta etapa, foram utilizados o Google Drive e Notion. Em seguida, foi realizada a modelagem (UML) do diagrama de casos de uso, utilizando o Visual Paradigm. Depois, iniciou-se a prototipação front-end das páginas principal e secundária da aplicação, utilizando HTML, CSS, Bootstrap e Javascript.

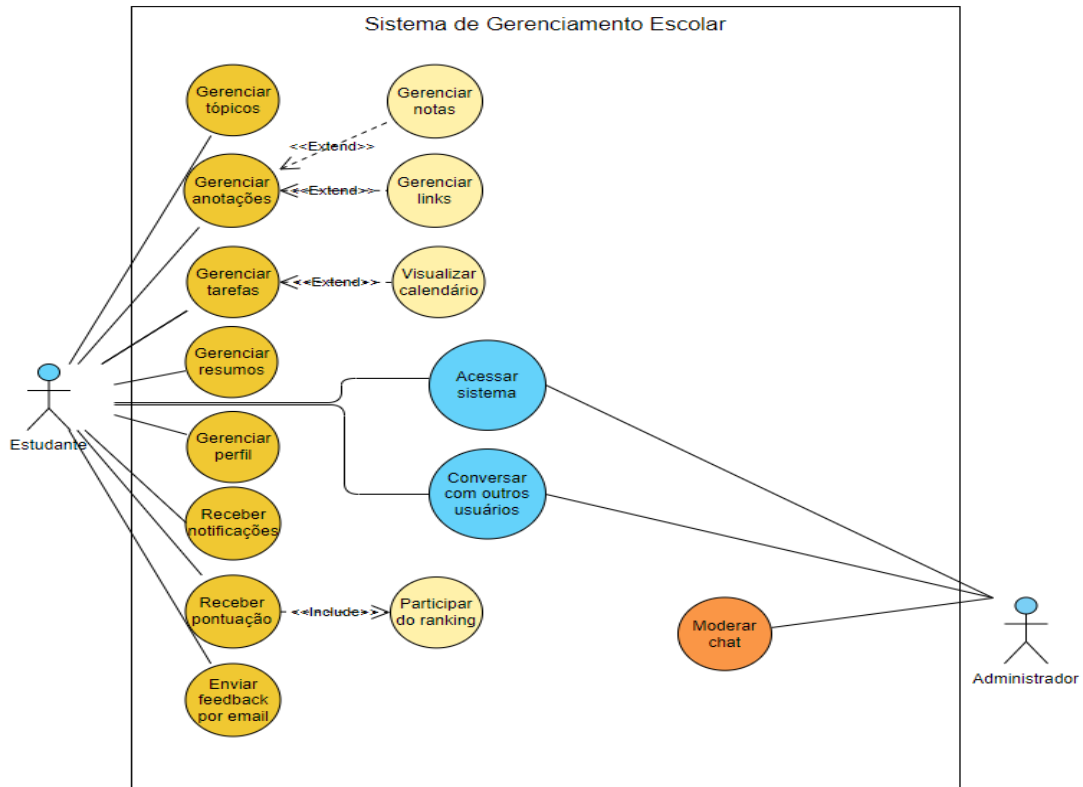
Após a criação das interfaces, foram criados os diagramas de classe e atividades para cada requisito funcional do software, também através do Visual Paradigm. Depois foi feita a definição do banco de dados, através do diagrama de classes, utilizando MySQL Workbench. Posteriormente, foi feita a integração do projeto com o Framework CodeIgniter e criação do back-end.

Durante o desenvolvimento do software, é comum que ocorram diversos empecilhos, o que pode afetar o produto final, por isso, houve a utilização da metodologia Scrum, que permite uma abordagem ágil e flexível para tratar erros e fazer alterações necessárias. Essa abordagem foi feita entre aluno e orientador, sendo aplicada durante as aulas da disciplina de Projeto de Software Orientado a Objetos. Por ser uma abordagem que consiste em ciclos curtos, a entrega das funcionalidades do sistema foi feita de forma incremental, sendo realizados ajustes ao longo do processo de desenvolvimento, ou seja, caso ocorresse alguma alteração no diagrama de casos de uso, o documento de requisitos seria alterado no mesmo momento.

3. RESULTADOS E DISCUSSÃO

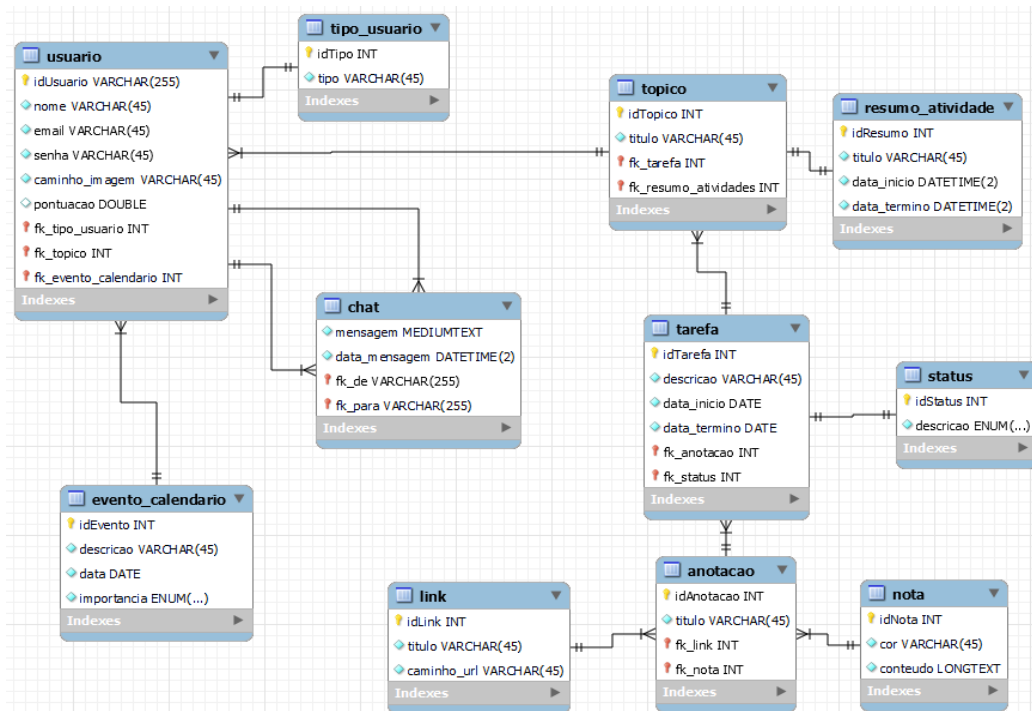
A utilização de modelos de processo de desenvolvimento de software, como a metodologia Scrum, juntamente com técnicas de modelagem de software, como a documentação de requisitos, diagrama de caso de uso e a modelagem de banco de dados, foi essencial para garantir a construção rápida e eficiente e do sistema de gerenciamento escolar. As figuras 1 e 2 demonstram respectivamente, o diagrama de caso de uso e o diagrama de modelagem de banco de dados.

Figura 1: diagrama de casos de uso



Fonte: dos autores (2023)

Figura 2: diagrama de banco de dados



Fonte: dos autores (2023)

4. CONCLUSÃO

Em conclusão, o projeto SchoolPlanner apresentou uma solução para lidar com a falta de organização e motivação dos alunos em concluir suas tarefas escolares. Com a utilização dos artefatos de software, o desenvolvimento do projeto foi bastante satisfatório e produtivo. O website ainda está em período de prototipação e no futuro será validado por outros usuários. Através das validações, serão realizados ajustes para melhor atender às necessidades das pessoas.

REFERÊNCIAS

DICHEVA, D., Dichev, C., Agre, G., & Angelova, G. (2015). **Gamification in education: A systematic mapping study**. *Educational Technology & Society*, 18(3), 75-88.

GEE, J. P., Squire, K., Halverson, R., & Moeller, K. (2017). **Motivation and engagement in the digital classroom**. *Harvard Educational Review*, 87(3), 409-434.

PRESSMAN, R.S. **Engenharia de Software**. 7ª ed. McGraw-Hill, 2011.