



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

HEMILLAINY SUELLEN SOUSA SANTOS

**"UFCG Pro":
EVOLUÇÃO DO SISTEMA DE CONTROLE ACADÊMICO DA
UFCG**

CAMPINA GRANDE - PB

2022

HEMILLAINY SUELLEN SOUSA SANTOS

**"UFCG PRO":
EVOLUÇÃO DO SISTEMA DE CONTROLE ACADÊMICO DA
UFCG**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharela em
Ciência da Computação.**

Orientador: Professor Dr. João Arthur Brunet Monteiro

CAMPINA GRANDE - PB

2022

HEMILLAINY SUELLEN SOUSA SANTOS

"UFCG Pro":

**EVOLUÇÃO DO SISTEMA DE CONTROLE ACADÊMICO DA
UFCG**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharela em
Ciência da Computação.**

BANCA EXAMINADORA:

Professor Dr. João Arthur Brunet Monteiro

Orientador – UASC/CEEI/UFCG

Professor Dr. Thiago Emmanuel Pereira da Cunha Silva

Examinador – UASC/CEEI/UFCG

Professor Dr. Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 02 de Setembro de 2022.

CAMPINA GRANDE - PB

"UFCG Pro": Evolução do Sistema de Controle Acadêmico da UFCG

Trabalho de Conclusão de Curso

Hemillainy Suellen Sousa Santos, João Arthur Brunet

hemillainy.santos@ccc.ufcg.edu.br, joao.arthur@computacao.ufcg.edu.br

Universidade Federal de Campina Grande

Campina Grande, Paraíba

RESUMO

A UFCG Pro é uma ferramenta open source que foi criada como uma extensão para os navegadores Chrome e Firefox com o intuito de fornecer funcionalidades extras para o sistema de Controle Acadêmico da UFCG. O objetivo principal deste trabalho foi evoluir a UFCG Pro para que atendesse ainda mais as expectativas da comunidade acadêmica, em particular, dentro do contexto do curso de Ciência da Computação. Este documento descreve não somente o levantamento das necessidades/requisitos com a comunidade acadêmica, como também a implementação e validação da ferramenta.

PALAVRAS-CHAVE

controle-acadêmico, navegador, extensão, vue.js

REPOSITÓRIO

<https://github.com/hemillainy/ufcg-pro>

1. INTRODUÇÃO

A comunidade acadêmica da UFCG, em particular professores e alunos, usam o Sistema de Controle Acadêmico (SCA) para consulta e organização das informações relacionadas às atividades de ensino e estágio. Por exemplo, através do sistema os professores podem registrar aulas, frequência dos alunos e notas dos alunos. Os alunos, por sua vez, além de poder visualizar essas informações, podem realizar matrícula e consultar histórico escolar.

Embora o SCA seja um sistema estável e que supre boa parte das necessidades da comunidade acadêmica, ainda há algumas lacunas que podem ser exploradas. Nesse contexto, a UFCG Pro [1] é uma extensão que foi desenvolvida em 2019 para prover funcionalidades extras ao Controle Acadêmico. Em particular, essa extensão permite aos professores realizar controle de presença de modo interativo e preenchimento automático de notas a partir do upload de um arquivo csv. Para os alunos, permite planejar a matrícula antecipadamente, exportar horário, visualizar a próxima aula na página inicial do SCA, entre outras funcionalidades.

Mesmo diante da criação da extensão, os usuários ainda apontam lacunas que poderiam ser exploradas no SCA e, portanto, implementadas na UFCG Pro como evolução dessa extensão. Neste contexto, neste Trabalho de Conclusão de Curso inicialmente investigamos junto à comunidade acadêmica quais seriam os possíveis pontos de melhoria e quais entre esses eram passíveis de implementação dadas as restrições de tempo e tecnologias. Após essa etapa inicial, evoluímos a implementação da UFCG Pro para incluir algumas das funcionalidades levantadas com a comunidade. Por fim, essas funcionalidades foram validadas através de testes unitários e exploratórios com alguns potenciais usuários da ferramenta.

Este documento está estruturado da seguinte maneira: na próxima seção, descreveremos detalhadamente a investigação

inicial que revelou pontos de melhoria a serem implementados na UFCG Pro e nas seções seguintes apresentaremos as funcionalidades que foram implementadas e algumas considerações sobre o processo de desenvolvimento.

2. INVESTIGAÇÃO DETALHADA DO PROBLEMA

Como o SCA é um sistema utilizado por milhares de usuários, é natural que haja expectativas quanto à sua evolução. Com o intuito de entender com maior profundidade quais necessidades os usuários gostariam que fossem atendidas pelo SCA, realizamos um levantamento com a comunidade acadêmica do curso de Ciência da Computação para identificar quais funcionalidades esses potenciais usuários gostariam que fossem implementadas no SCA.

Nessa fase de investigação inicial, elaboramos um questionário cuja principal questão foi: *Quais funcionalidades você gostaria que estivessem presentes no Controle Acadêmico?* Ao todo 31 pessoas responderam a essa pergunta, o que nos proporcionou dados relevantes sobre a demanda de evolução da UFCG Pro. A análise manual desses dados revelou 9 funcionalidades passíveis de implementação dadas as restrições de tecnologias. Essas funcionalidades revelam problemas de usabilidade no SCA, que ainda não foram solucionados pela implementação inicial da UFCG Pro. Por exemplo, um professor relatou dificuldade em realizar o registro das aulas, pois no total é necessário registrar 30 aulas semestralmente de forma manual para cada disciplina, no entanto, esses registros não costumam sofrer grandes alterações a cada semestre, portanto trata-se de **um trabalho manual e repetitivo** que pode ser automatizado.

Outro problema, dessa vez apontado por alguns alunos, foi a dificuldade de realizar uma estimativa do Coeficiente de Rendimento Acadêmico (CRA) ao final de cada período letivo. Tipicamente, os alunos criam planilhas eletrônicas para realizar a simulação do CRA.

Todavia, é necessário adicionar manualmente todas as notas das disciplinas já aprovadas, além das disciplinas cursadas no momento em conjunto com a média esperada ao final do período. Por ser **manual**, essa atividade é **custosa e passível de erros**. Ainda que a planilha guarde o histórico, a cada semestre novo o aluno deve adicionar mais disciplinas ao registro. Por fim, trata-se de uma informação que está fora do ambiente do controle acadêmico, além de não ser concebida utilizando dados oficiais.

Por fim, dois outros problemas do SCA foram identificados nesta investigação inicial, quais sejam: dificuldade em visualizar e compreender o horário semanal e a impossibilidade de realizar o download das notas em formato de planilha. Assim como o problema citado anteriormente, ambos são resolvidos através de inserção manual em planilhas eletrônicas.

3. SOLUÇÃO: EVOLUÇÃO DA UFCG Pro

Como solução para os problemas expostos acima, desenvolvemos uma evolução da extensão UFCG Pro que implementa as funcionalidades descritas abaixo.

3.1 Funcionalidades

Simulador de CRA

O CRA é utilizado em diversas atividades no ambiente acadêmico, sobretudo para seleção para projetos, monitoria, PIBIC, entre outras. Portanto, é importante que o aluno consiga realizar um planejamento de suas notas, de modo a monitorar as variações de seu CRA. Isso pode ser feito através da funcionalidade Simulador de CRA. Essa funcionalidade permite ao usuário simular seu CRA com base nas médias anteriores obtidas automaticamente do sistema e nas médias estimadas para as disciplinas cursadas no período atual. A automação desse processo facilita o monitoramento do CRA por parte do aluno, pois otimiza a coleta das notas de todas as disciplinas já cursadas.

A funcionalidade está disponível na página *Turmas em curso* (Figura 1), onde é exibida uma seção com a listagem das disciplinas cursadas no momento, na qual é possível inserir uma média estimada que será utilizada na simulação do CRA. Após o aluno inserir uma nota, o simulador irá calcular qual será o CRA do aluno caso ele obtenha a média que foi fornecida.

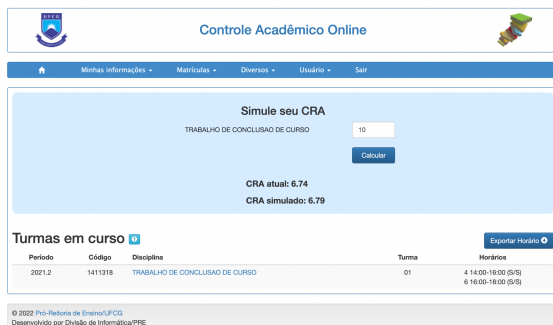


Figura 1: Simulação Automática de CRA

Exibição do horário do aluno em formato de calendário

Atualmente, os alunos consideram que o SCA apresenta o horário em um formato pouco compreensível. Isto porque o formato escolhido não segue o padrão de calendário. Tipicamente os alunos resolvem esse problema por meio de aplicações externas ou planilhas eletrônicas, evidenciando a necessidade latente de uma melhoria dentro do ambiente do SCA.

Para exemplificar o problema, a figura abaixo apresenta um exemplo de horário onde as disciplinas são exibidas de forma linear.

	Quarta-feira	Sexta-feira
14.00-16.00	TRABALHO DE CONCLUSÃO DE CURSO (S/S)	
16.00-18.00		TRABALHO DE CONCLUSÃO DE CURSO (S/S)

Figura 2: Tabela com o horário do aluno

Nossa investigação inicial apontou a necessidade de apresentar esse horário como um calendário em forma de matriz, onde a primeira linha possui todos os dias da semana e a primeira coluna possui os horários de aula mais comuns.

Essa é uma forma mais adequada para apresentar o horário, pois é o padrão utilizado

para exibição desse tipo de informação. A Figura 3 apresenta como o horário será exibido após o aluno clicar no botão *Alterar Visualização*.

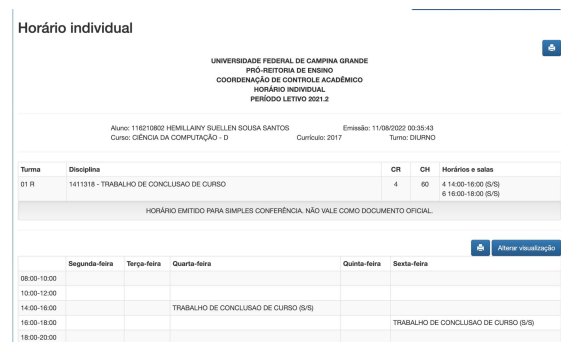


Figura 3: Visualização do Horário em Formato de Calendário

Apesar de termos implementado essa funcionalidade, a visualização padrão foi mantida, portanto, o aluno pode escolher o modo de exibição do seu horário.

Download das notas em formato de planilha

As notas em algumas disciplinas são compostas muitas vezes por diversas atividades além das habituais provas, como mini-testes e projetos práticos, por exemplo. Por isso, é bastante comum os alunos criarem planilhas para fazer o acompanhamento de suas notas ao longo do semestre, de modo que possam planejar melhor a pontuação que deve ser obtida nas próximas atividades, de modo que seja aprovado na disciplina.

No entanto, esse também é um **trabalho manual**, no qual o aluno deve criar a planilha e ir preenchendo-a com as notas que vão sendo obtidas. Por isso, alguns usuários informaram na nossa pesquisa que gostariam que o SCA oferecesse a possibilidade de download de suas notas.

A Figura 4 apresenta a funcionalidade que foi implementada para resolver esse problema. Adicionamos um novo botão **Baixar notas**, que após ser clicado faz o download de uma planilha com as notas do aluno na disciplina em questão.



Figura 4: Tela de notas com o novo botão para download de notas

A Figura 5 apresenta um exemplo da planilha que é baixada para o aluno.

trabalho_de_conclusao_de_curso_notas

	nota 1	nota 2	nota 3	media parcial	exame final	media final	obs
1							

Figura 5: Planilha de notas

A planilha do exemplo está vazia pois ainda não havia nenhuma nota no sistema no momento do download. A planilha é nomeada no formato <nome da disciplina>_notas.csv, de modo que fique fácil para o aluno ministrar as planilhas de diversas disciplinas.

Preenchimento automatizado do registro de aulas

Como mencionado na seção 2, todo semestre é necessário que os professores realizem, de forma manual, o registro de todas as aulas ministradas, registros esses que não costumam sofrer grandes alterações com o passar do tempo. A Figura 6 mostra a página *Registro de aulas*. Nela, o professor precisa adicionar uma nova linha na tabela para cada novo registro de aula, para em seguida preencher os campos com as respectivas informações.

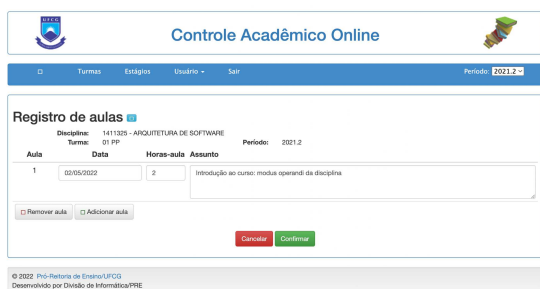


Figura 6: Tela de registro de aulas

Buscando automatizar esse processo, implementamos a funcionalidade que permite que o professor possa preencher o registro de aulas através do upload de um arquivo CSV. As Figuras 7 e 8 apresentam a nova funcionalidade. Inicialmente (Figura 7), há apenas 1 aula registrada. Em seguida, após selecionar um arquivo CSV e clicar em preencher, os registros são preenchidos automaticamente.

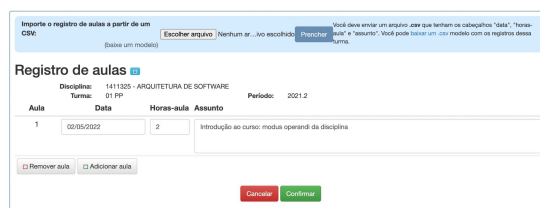


Figura 7: Registro de aulas antes do preenchimento automático via CSV

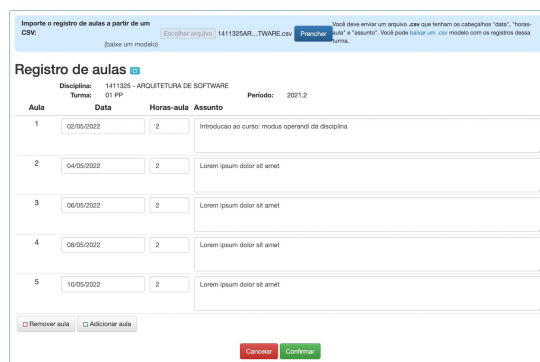


Figura 8: Registro de aulas após o preenchimento automático

A Figura 9 apresenta a planilha que foi utilizada para realizar o preenchimento do registro do exemplo.

1411325ARQUITETURADESOFTWARE

aula	data	horas-aula	assunto
1	02/05/2022	2	Introdução ao curso: modos operandi da disciplina
2	04/05/2022	2	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
3	06/05/2022	2	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
4	08/05/2022	2	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Figura 9: Planilha com os registros de aulas

3.2 Arquitetura da Solução

A UFCG Pro foi desenvolvida utilizando uma arquitetura de componentes. Para isto, foi utilizado o framework Vue.js [4]. Para cada

página que desejamos injetar um novo código, criamos dois arquivos: um *component.vue* e um *injector.js*. Além desses dois arquivos, há um script *background.js*. O *component.vue* é responsável por definir o que será exibido na tela e a forma que os dados serão tratados; o *injector* define em qual ponto do DOM o *component* será injetado. O *background.js* é quem identifica as páginas nas quais o *injector* será aplicado. A Figura 10 apresenta a arquitetura geral da UFCG Pro.

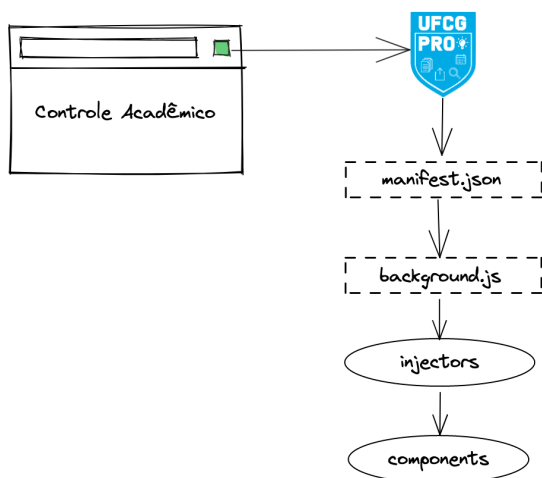


Figura 10: Arquitetura geral

3.3 O Processo de Implementação

Após analisar manualmente as respostas obtidas na pesquisa, fizemos um levantamento de quais funcionalidades conseguiríamos implementar, levando em consideração as restrições de tecnologias. Foram definidas 9 funcionalidades implementáveis, das quais priorizamos a implementação de 5, com base na importância para o usuário e tempo disponível para o desenvolvimento.

A priorização das tarefas foi feita utilizando *planning poker* com a *pontuação Fibonacci* [15] e utilizamos o Trello para realizar o registro do ciclo de vida das atividades de implementação.

4. DESAFIOS

O desenvolvimento open source costuma trazer consigo alguns desafios, entre eles podemos citar: curva de aprendizado das tecnologias e compreensão da arquitetura e implementação do projeto. Neste trabalho em específico, os maiores desafios enfrentados foram encontrar funcionalidades passíveis de implementação e em seguida entender como implementá-las em um projeto já existente.

Optamos por iniciar a implementação das tarefas que obtiveram uma pontuação mais baixa no *planning poker* pois isso geralmente significa que a implementação seria mais fácil. No entanto, devido à falta de um conhecimento mais aprofundado sobre como o projeto estava implementado, algumas pontuações foram estimadas erroneamente, o que atrasou o processo de desenvolvimento, pois foi necessário mais tempo que o esperado para a implementação das tarefas. Devido a isso, não conseguimos implementar uma das funcionalidades que havíamos planejado, falaremos a respeito disso no tópico sobre os trabalhos futuros.

4.1 Pull Requests

Abrimos um *Pull Request* (PR) ao final da implementação de cada nova funcionalidade, os quais passaram por todo processo de revisão, correção e aprovação pelo criador e desenvolvedor da ferramenta. Conseguimos obter aprovação em todos os PR's que foram abertos, o deploy ficou sob responsabilidade do criador da ferramenta.

4.2 Trabalhos Futuros

Podemos citar duas que podem ser trabalhadas futuramente: para os alunos, o download de seu fluxograma, semelhante à planilha da Figura 11 e para os professores, o preenchimento automatizado do plano de curso via upload de uma planilha. Essa segunda sugestão estava no nosso backlog mas não conseguimos implementá-la devido às restrições de tempo.

	A	B	C	D	E	F	G	H	I	
1	1º Período	2º Período	3º Período	4º Período	5º Período	6º Período	7º Período	8º Período	9º Período	
1	FMCC1	FMCC2	Álgebra Linear	Análise I / Probabilidade	Estatística Aplicada	Metodologia Científica	ATAC	Projeto em Computação I	Projeto em Computação II	
2	Programação I	Cálculo I	Lógica para Computação	PIBET	Análise de Sistemas	Programação Concorrente	Computadores	FTCC	Trabalho de Conclusão de Curso	
3	LPI	Programação II	Cálculo II	PLP	Engenharia de Software	Inteligência Artificial	Recuperação de Informação e Bases de Dados II	Governança de TI	Gestão de Projetos II	
4	IC	LPI2	EDA	Banco de Dados	Redes de Computadores	Gestão de Informação II	Empreendedorismo em Software II	Segurança de Sistemas II	Identificação e Verificação de Software II	
5	LPI (G)	Álgebra Vetorial (G)	LEDA	OAC	SD	Prática de Ensino II	Física Geral II (G)	Informática e Sociedade II	Sistemas de Informação geográfico II	
6			Teoria dos Grafos	LOAC	Teoria da Computação					
7										
8										
9										
10										
11	LEGENDA									
12	Obrigatório / Ap									
13	Opt. Geral / Ap									
14	Opt. Específica / Ap									
15	Faltas									
16	Maturidade									
17			5							
18			5							
19			0							

Figura 11: Planilha com o fluxograma do aluno

A UFCG Pro no momento já é uma ferramenta bastante completa que oferece diversas novas funcionalidades para os usuários. Partindo desse ponto, está ficando um pouco mais difícil encontrar novas funcionalidades passíveis de implementação, portanto é crucial que a comunidade contribua constantemente expondo quais necessidades ainda não são atendidas pelo SCA e pelo UFCG Pro até o momento. Acreditamos que as próximas implementações na ferramenta surgirão a partir do feedback e contribuição dos usuários.

6. REFERÊNCIAS

- [1] OLIVEIRA JÚNIOR, Luciano de. **"UFCG Pro": uma ferramenta para melhorar o uso do Controle Acadêmico da Universidade Federal de Campina Grande**. 2019. 12f. (Trabalho de Conclusão de Curso - Artigo), Curso de Bacharelado em Ciência da Computação, Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande – Paraíba - Brasil, 2019. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/20435/>
- [2] UBAH, Kingsley. **Desenvolvendo Extensões para o Google Chrome**. Disponível em: <https://code.tutsplus.com/pt/tutorials/developing-google-chrome-extensions--net-33076/>
- [3] **Como criar uma extensão do Google Chrome**. Disponível em: <https://dev.to/aryclenio/como-criar-uma-extensao-do-google-chrome-2o61/>
- [4] **Vue.js**. Disponível em: <https://vuejs.org/>
- [5] **Manual do Aluno**. Disponível em: <https://pre.ufcg.edu.br/pre/manual-do-aluno/>

[6] **CLI**. Disponível em: <https://vercel.com/docs/cli/>

[7] **Node.removeChild - APIs da Web | MDN**. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/API/Node/removeChild/>

[8] **Remove all the child elements of a DOM node in JavaScript**. Disponível em: <https://www.geeksforgeeks.org/remove-all-the-child-elements-of-a-dom-node-in-javascript/#:~:text=Child%20nodes%20can%20be%20removed/>

[9] **How To Add Table Rows & Cells In Javascript (Simple Examples)**. Disponível em: <https://code-boxx.com/add-table-rows-cells-javascript/>

[10] **Traversing an HTML table with JavaScript and DOM Interfaces - Web APIs | MDN**. Disponível em:

<https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Traversing_an_HTML_table_with_JavaScript_and_DOM_Interfaces/>

[12] **How to Add Styles to an Element in JavaScript**. Disponível em: <https://www.javascripttutorial.net/dom/css/add-styles-to-an-element/>

[13] **Class and Style Bindings | Vue.js**. Disponível em: <https://vuejs.org/guide/essentials/class-and-style.html#binding-html-classes/>

[14] **Using v-model for Two-Way Binding in Vue.js**. Disponível em: <https://www.digitalocean.com/community/tutorial-s/vuejs-v-model-two-way-binding/>

[15] BRASILEIRO, Roberto. **Planning Poker: Tudo que você NÃO sabia sobre a melhor técnica de estimativa**. Disponível em: <https://www.metodoagil.com/planning-poker/#fibonacci/>

[16] **HTML DOM Document Objects**. Disponível em: <https://www.w3schools.com/jsref/dom_obj_document.asp/>

[17] **HTMLTableRowElement.insertCell() - Web APIs | MDN**. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/API/HTMLTableRowElement/insertCell/>