

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI  
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**ADAPTAÇÕES DO SISTEMA CURRICULAR:** Estratégias, Desafios e Resultados

**Natascha Annette Soares Schlegel**

Diamantina

2023

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI  
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS

**ADAPTAÇÕES DO SISTEMA CURRICULAR:** Estratégias, Desafios e Resultados

**Natascha Annette Soares Schlegel**

Orientador(a):

**Alessandro Vivas Andrade**

Co-orientador(a):

**Luciana Pereira de Assis**

Trabalho de Conclusão de Curso apresentado ao  
Curso de Bacharelado em Sistemas de Informa-  
ção, como parte dos requisitos exigidos para a  
obtenção do título de Bacharel.

Diamantina

2023



**ADAPTAÇÕES DO SISTEMA CURRICULAR:** Estratégias, Desafios e Resultados

**Natascha Annette Soares Schlegel**

Orientador(a):

**Alessandro Vivas Andrade**

Co-orientador(a):

**Luciana Pereira de Assis**

Trabalho de Conclusão de Curso apresentado ao  
Curso de Bacharelado em Sistemas de Informa-  
ção, como parte dos requisitos exigidos para a  
conclusão do curso.

APROVADO em \_\_\_\_ / \_\_\_\_ / \_\_\_\_.

---

Me. Clayton Brás de Souza Lopes – Universidade Federal dos Vales do Jequitinhonha e  
Mucuri

---

Prof. Dra. Luciana Pereira de Assis – Universidade Federal dos Vales do Jequitinhonha e  
Mucuri

---

Prof. Dr. Alessandro Vivas Andrade – Universidade Federal dos Vales do Jequitinhonha e  
Mucuri

Dedico esse trabalho a minha mãe, a flor mais  
bonita do meu jardim.

## **AGRADECIMENTO**

Agradeço primeiramente os meus pais, Eliene e Ernesto, que me incentivaram e encorajaram durante todo o meu percurso estudantil, sem os quais eu não teria chegado até aqui.

Aos meus orientadores, Luciana e Alessandro, pela atenção e tempo que me dedicaram, me guiando e incentivando o meu progresso. A Clayton, pelo apoio na realização deste trabalho.

Um agradecimento especial aos meus amigos e colegas de curso, Amanda, Christian, Hugo e Paulo, que me acompanharam durante todos esses anos de estudo, compartilhando experiências e aprendizados que moldaram minha trajetória acadêmica e pessoal.

Aos meus amigos Gabriel, Hellen e Raabe, que estiveram ao meu lado nos momentos desafiantes da graduação. A Beatriz, pelas palavras certas nos momentos certos.

E por fim, aos professores do curso de Sistemas de Informação, que contribuíram diretamente para minha formação acadêmica, instruindo minha compreensão na área para os desafios que virão pela frente.

## RESUMO

No ensino superior, a construção da grade curricular desempenha um papel fundamental no sucesso acadêmico dos estudantes. Grades curriculares mal estruturadas podem trazer dificuldades, aumentando o risco de reprovação e suas consequências. O Sistema Curricular distribui as disciplinas que o estudante ainda não cursou nos períodos disponíveis utilizando o modelo matemático desenvolvido para resolver o Problema de Balanceamento de Currículo com Dados de Retenção, que procura distribuir as disciplinas minimizando o número total de créditos por períodos, a distância entre disciplinas interconectadas e o acumulado de índice de retenção, criando uma grade equilibrada que respeite as restrições pedagógicas e seja coerente com os objetivos do estudante. Após a disponibilização do sistema para testes e a realização de uma pesquisa de opinião pública, foi percebido que o sistema possui algumas queixas em relação a usabilidade no processo de inserção de disciplinas. Com o objetivo de estudar as queixas relatadas e propor uma solução afim de tornar a experiência do usuário mais agradável, as alterações foram desenvolvidas e apresentadas nesse trabalho.

**Palavras-chave:** Desenvolvimento Web, Balanceamento de Currículo Acadêmico, Usabilidade.

## ABSTRACT

In higher education, the construction of the curriculum plays a fundamental role in the academic success of students. Poorly structured curricula can cause difficulties, increasing the risk of failure and its consequences. The Curricular System distributes the subjects that the student has not yet taken in the available periods using the mathematical model developed to solve the Curriculum Balancing Problem with Retention Data, which seeks to distribute the subjects minimizing the total number of credits per period, the distance between interconnected subjects and the accumulated retention rate, creating a balanced schedule that respects pedagogical restrictions and is coherent with the student's objectives. After making the system available for testing and carrying out a public opinion survey, it was noticed that the system has some complaints regarding usability in the course insertion process. With the aim of studying the reported complaints and proposing a solution to make the user experience more pleasant, the changes were developed and presented in this work.

**Keywords:** Web development, Academic Curriculum Balancing, Usability.



## LISTA DE FIGURAS

2.1	Tela Inicial do Sistema Curricular. Fonte: Captura de Tela . . . . .	7
3.1	Arquitetura Cliente-Servidor. Fonte: Elaborada pelo autor . . . . .	11
3.2	Protocolo HTTP. Fonte: Gourley et al. (2002) . . . . .	12
3.3	Arquitetura MTV. Fonte: Elaborada pelo autor . . . . .	14
4.1	Captura de tela da página original. Fonte: Elaborada pelo autor . . . . .	18
4.2	Fluxograma Antigo. Fonte: Elaborada pelo autor . . . . .	19
4.3	Fluxograma Novo. Fonte: Elaborada pelo autor . . . . .	20
5.1	Página de administração da grade. Fonte: Captura de Tela . . . . .	23
5.2	Página para adicionar disciplinas cursadas. Fonte: Captura de Tela . . .	24
5.3	Aviso de erro. Fonte: Captura de Tela . . . . .	25
5.4	Chamada de requisição. Fonte: Captura de tela . . . . .	25
5.5	Adicionar disciplinas não cursadas. Fonte: Captura de Tela . . . . .	26
5.6	Aviso de sucesso. Fonte: Captura de Tela . . . . .	26
5.7	Aviso de erro. Fonte: Captura de Tela . . . . .	26
5.8	Chamada assíncrona para fixar período. Fonte: Captura de Tela . . . . .	26
5.9	Administração de Grade. Fonte: Captura de Tela . . . . .	27
5.10	Página Disciplina A Cursar. Fonte: Captura de Tela . . . . .	28
5.11	Administração de disciplina de Grade. Fonte: Captura de Tela . . . . .	29
5.12	Administração de disciplina de Grade. Fonte: Captura de Tela . . . . .	29

## LISTA DE TABELAS

3.1	Fonte: Texto Adaptado de Gourley et al. (2002) . . . . .	12
4.1	Fonte: Adaptado de Lopes (2022) . . . . .	17

## LISTA DE SIGLAS

AJAX - Asynchronous Javascript and XML.

API - Application Programming Interface.

BACF - Balancing Academic Curriculum With Subjects Failure Data.

BACP - Balanced Academic Curriculum Problem.

CSP - Constraint Satisfaction Problem.

CSS - Cascading Style Sheets.

HTTP - Hypertext Transfer Protocol.

HTML - HyperText Markup Language.

JSON - JavaScript Object Notation.

MILP - Mixed Integer Linear Programming.

MIQP - Mixed-Integer Quadratic Programming.

MTV - Model-Template-View.

OECD - Organisation for Economic Co-operation and Development.

ORM - Object-Relational Mapper.

PNE - Plano Nacional de Educação.

RBCB - Relevance Based Curriculum Balancing.

RIA - Rich Internet Application.

SGBD - Sistema de Gerenciamento de Banco de Dados.

SQL - Structured Query Language.

UEQ - User Experience Questionnaire.

UFVJM - Universidade Federal dos Vales do Jequitinhonha e Mucuri.

## SUMÁRIO

<b>1 INTRODUÇÃO.</b>	1
1.1 Objetivos	3
1.1.1 Objetivos Gerais	3
1.1.2 Objetivos Específicos	3
1.2 Metodologia	3
<b>2 REFERENCIAL TEÓRICO</b>	5
2.1 Introdução	5
2.2 Balanceamento de Currículo Acadêmico com Dados de Retenção	6
2.3 Trabalhos Relacionados	8
<b>3 FERRAMENTAS E TECNOLOGIAS</b>	10
3.1 Framework Web Django	10
3.2 Arquitetura Cliente-Servidor	11
3.3 Protocolo HTTP	11
3.4 Arquitetura MTV	13
3.5 Mapeamento Objeto-Relacional	14
3.6 Funções Assíncronas	15
<b>4 MELHORIAS PROPOSTAS</b>	16
4.1 Justificativa	16
4.2 Diagrama de Fluxo de Usuário	18
4.3 Implementação	19
<b>5 RESULTADO</b>	23
<b>6 CONCLUSÃO</b>	30
<b>REFERÊNCIAS</b>	32
<b>APÊNDICE A – CÓDIGO-FONTE</b>	37

## 1 INTRODUÇÃO.

A Universidade desempenha um papel importante de transformação social ao criar e disseminar conhecimento. O aumento dos investimentos em educação refletem em uma mão de obra mais qualificada ([VIANA; LIMA, 2010](#)), ao mesmo tempo que a transformação das condições produtivas promovida pela pesquisa e aumento de competitividade têm o potencial de gerar lucro e superávits econômicos, resultando em mais empregos e renda, sustentando a sociedade e contribuindo para o desenvolvimento local e nacional ([BURON, 2016](#); [TARTARUGA, 2010](#)).

O governo brasileiro investe cerca de 5% do PIB ([INEP, 2023](#)) na educação básica e superior, com a expectativa de que os resultados desse investimento se traduzam em um aumento significativo da presença da população jovem nas escolas e universidades, pretendendo ter até 2024 um terço da população entre 18 e 24 anos matriculada no ensino superior. Em 2021, 20,1% da população nessa faixa etária estava cursando ensino superior e 5,4% já havia concluído seus estudos ([INEP, 2022](#)). Nesse mesmo ano, 21% dos brasileiros entre 25 a 34 anos possuíam ensino superior ([OECD, 2022](#)). Observando a entrada de estudantes em 2017 em cursos presenciais das instituições públicas, após cinco anos, 19,3% desses estudantes se formaram no tempo ideal, 39,4% evadiram do curso e 41,3% mantiveram vínculo com a universidade após o tempo ideal, considerados como retidos ([SEMESP, 2023](#)), portanto 80,7% dos alunos acarretaram prejuízo para a universidade.

Apesar das percentagens alarmantes de desistência, é importante considerar que as porcentagens de evasão registram a saída de estudantes de um determinado curso, sem considerar os passos seguintes, sendo possível em que o estudante evada para trocar de curso ou instituição e não abandonando o ensino superior. Apesar disso, os dados demonstram que apenas 1 em cada 5 estudantes se graduam no tempo ideal, 2 ficam retidos e 2 evadem.

A retenção de estudantes tem impactos significativos, tanto para os estudantes que acabam estendendo o tempo necessário para concluir sua graduação e adiar sua entrada no mercado de trabalho ([ARAÚJO et al., 2021](#)), quanto para as instituições de ensino que precisam lidar com imprevistos custosos no processo educacional ([GARCIA et al., 2020](#)). Já a evasão, representa um desperdício de tempo e recursos investidos e a quebra da expectativa dos retornos financeiros, tanto por parte do estudante quanto do país que dirigiu recursos para a educação superior.

O desempenho acadêmico ruim e a reprovação em disciplinas são fatores que precedem a retenção dos estudantes e podem, eventualmente, levar à evasão (PEREIRA et al., 2015; CUNHA et al., 2023; AMORIM, 2016). A relação entre reprovações e retenção foi estudada por Garcia et al. (2020), que afirma que estudantes com uma reprovação tem 2,3 vezes mais chance de retenção, duas a cinco reprovações o risco é de 4,8 vezes, e seis ou mais reprovações o risco é de 41,7 vezes.

O currículo acadêmico, responsável por estruturar o percurso educacional, inclui também a grade curricular do curso, que lista as disciplinas cursadas pelo aluno até sua graduação. Em muitos cursos, os alunos têm liberdade para montar sua grade curricular, desde que respeitem as regras pedagógicas do curso. Castro e Manzano (2001) propuseram o Problema de Balanceamento de Currículo Acadêmico (BACP - Balanced Academic Curriculum Problem) que organiza as disciplinas minimizando o número de períodos necessários para a conclusão do curso, seguindo as regras pedagógicas de pré-requisitos e números mínimos e máximos de créditos por período.

O Balanceamento de Currículo Acadêmico com Dados de Retenção (BACF - Balancing Academic Curriculum With Subjects Failure Data), desenvolvido a partir do BACP por Lopes (2022), distribui as disciplinas minimizando o número de períodos necessários para graduação, equilibrando os níveis de retenção e favorecendo a proximidade de disciplinas com temas correlatos, buscando assim diminuir as chances de reprovação.

Para que o uso do balanceamento se tornasse de amplo acesso, Lopes (2022) desenvolveu uma aplicação Web denominada "Curricular". O sistema é composto por dois módulos distintos: um destinado aos gestores, que permite adicionar, modificar e excluir informações relacionadas aos cursos, como por exemplo, disciplinas, pré-requisitos, cargas de créditos e grades curriculares, e outro voltado para os alunos, que, ao selecionarem seu curso em uma lista, terão acesso às informações adicionadas pelos gestores. O sistema foi apresentado aos estudantes do curso de Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri e aplicado um questionário mensurando a qualidade do sistema através da experiências do usuário. Nesse questionário identificou-se um problema de usabilidade no fluxo de inserção das disciplinas que o usuário necessita cursar. O propósito desse trabalho é compreender o desenvolvimento do sistema, elaborar um protótipo que indique melhorias quantos aos problemas relatados, implementar as alterações necessárias e documentar o processo.

## **1.1 Objetivos**

### **1.1.1 Objetivos Gerais**

Entender os problemas de usabilidade relatados na página de inclusão das disciplinas que serão cursadas pelos alunos e desenvolver uma nova interface, visando promover agilidade no processo e melhorias na experiência do usuário.

### **1.1.2 Objetivos Específicos**

Com o intuito de alcançar o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- I. Realizar uma revisão literária sobre o software Curricular e o problema de usabilidade a ser tratado.
- II. Criar protótipos de tela com as alterações propostas.
- III. Desenvolver as alterações no sistema Curricular.
- IV. Realizar testes de regressão no software adaptado.

## **1.2 Metodologia**

O trabalho iniciou-se na leitura dos artigos e trabalhos que envolvessem o tema do Problema de Balanceamento de Currículo com Dados de Retenção e o sistema Curricular. Foi então criado um protótipo da interface no software Figma<sup>1</sup> e iniciado o desenvolvimento das telas estáticas no front-end e adaptação dessas telas no sistema Curricular.

Dado que o projeto se fundamentou em um sistema já existente, as tecnologias a serem utilizadas já estavam definidas. No desenvolvimento do sistema foi usado a linguagem de programação Python<sup>2</sup>, o Framework Django<sup>3</sup> e o SGBD (Sistema de Gerenciamento de Banco de Dados) PostgreSQL<sup>4</sup>. Além disso, o sistema também incorporou funcionalidades relacionadas

---

<sup>1</sup><https://help.figma.com/hc/en-us>

<sup>2</sup><https://docs.python.org/3/>

<sup>3</sup><https://docs.djangoproject.com/en/4.2/>

<sup>4</sup><https://www.postgresql.org/docs/>

à biblioteca Gurobi<sup>5</sup>, responsável pelo balanceamento, e o *message broker* Redis<sup>6</sup>, responsável pelos processos assíncronos de balanceamento.

---

<sup>5</sup><https://www.gurobi.com/documentation/>

<sup>6</sup><https://docs.redis.com/latest/index.html>

## 2 REFERENCIAL TEÓRICO

Neste capítulo serão abordados alguns tópicos relacionados ao sistema curricular. Na primeira seção, serão exploradas as razões que motivaram a construção do software. Em seguida será descrito os objetivos do BACF que fundamentam esse sistema e por fim, será apresentado uma análise do material bibliográfico existente sobre o software.

### 2.1 Introdução

De acordo com [Biggs \(1987\)](#), há diversos fatores que influenciam o nível de esforço dedicado pelos estudantes ao aprendizado. Os fatores de presságio, que representam o contexto anterior ao ato de estudar, são classificados em duas categorias: pessoais e situacionais. Os fatores pessoais envolvem aspectos como personalidade, capacidade cognitiva e outras questões, como problemas emocionais ou financeiros, enquanto que os fatores situacionais estão relacionados à estrutura da organização educacional em que o estudante está inserido, incluindo a didática do professor, o clima institucional e a estrutura do curso.

Os fatores de presságio influenciam então no processo de estudo, definindo a abordagem escolhida pelo estudante e sua motivação. Como explica [Gomes \(2011\)](#), estudantes com motivações extrínsecas, como obter aprovação em uma disciplina, tendem a adotar uma abordagem superficial, caracterizada por uma postura passiva, em que o estudante procura memorizar o conteúdo apenas para reproduzi-lo em uma avaliação. Por outro lado, estudantes com motivações intrínsecas, que buscam adquirir uma compreensão abrangente sobre um conteúdo, tendem a adotar uma abordagem profunda, construindo significados pessoais sobre o objeto de estudo e formando relações com conhecimentos aprendidos anteriormente.

Através do estudo realizado por [Lizzio et al. \(2002\)](#), foi percebida uma conexão recíproca entre a escolha de uma abordagem superficial e a percepção de sobrecarga, indicando que além das percepções influenciarem as abordagens escolhidas pelos alunos, as abordagens escolhidas afetam a percepção da carga de trabalho. A sobrecarga nos estudos tem consequências como a redução do desempenho, o esgotamento e o aumento dos níveis de ansiedade e depressão ([HERNESNIEMI et al., 2017](#)).

A Organização para a Cooperação e Desenvolvimento Econômico (OCDE), em [2020](#), publicou um estudo sobre sobrecarga curricular, termo utilizado para descrever situações em



que há uma quantidade inadequada de informações ou atividades. Neste estudo, a sobrecarga pode ser vista por quatro perspectivas diferentes, sendo uma delas a sobrecarga percebida, conceito que representa a sobrecarga sentida quando a quantidade excessiva de conteúdo necessita de um espaço de tempo maior que o disponível pra ser absorvido. Nessa circunstância, a presença de disciplinas densas ou difíceis em uma estrutura curricular pode intensificar a sensação de sobrecarga atrapalhando os estudantes na compreensão do conteúdo da forma adequada.

A carga de trabalho numa grade curricular é, geralmente, medida na distribuição de créditos. Quanto maior a quantidade de créditos maior deve ser o tempo e esforço dedicado pelo aluno. Inicialmente, para evitar a sobrecarga de seus alunos, as instituições de ensino optam por restringir a quantidade mínima e máxima de créditos que o estudante pode cursar em um período.

Castro e Manzano (2001) veem a carga acadêmica como fator fundamental que influencia o sucesso acadêmico. Por isso, com o intuito de reduzir os efeitos da sobrecarga, desenvolveram o BACP, distribuindo de forma equilibrada as disciplinas entre períodos acadêmicos. Baseando-se no BACP, Lopes (2022) criou um novo problema, o BACF, abordado no próximo capítulo. Após o desenvolvimento do modelo matemático para resolução do problema, Lopes (2022) fundamentou a criação de um sistema que incorporasse o modelo.

Por fim, em que pese a expressividade deste problema nas realidades acadêmicas, são escassos os trabalhos da literatura que resultam na construção de interfaces de usuário para a construção de grades curriculares balanceadas. Portanto, o estudo deste trabalho justifica-se pela oportunidade em disponibilizar um sistema, que tenha um novo modelo de otimização multiobjetivo embutido mais abrangente do que os apresentados até então na literatura, para suporte ao desenvolvimento de grades curriculares otimizadas. (LOPES, 2022, p. 19)

O sistema foi então implementado, exibido na figura 2.1, tornando-se uma plataforma voltada para a comunidade acadêmica, capaz de gerenciar e disponibilizar grades curriculares mais adequadas aos esforços do estudante. Com o objetivo de se tornar acessível para qualquer aluno, o sistema oferece a possibilidade de inserção de grades curriculares de qualquer universidade que o utilize.

## **2.2 Balanceamento de Currículo Acadêmico com Dados de Retenção**

O BACF foi construído a partir do BACP, incorporando um objetivo descrito no trabalho de Ünal e Uysal (2014) para aproximar disciplinas que são relacionadas entre si e uma restrição descrita no trabalho de Rosas et al. (2011), favorecendo a inclusão de disciplinas em um pe-

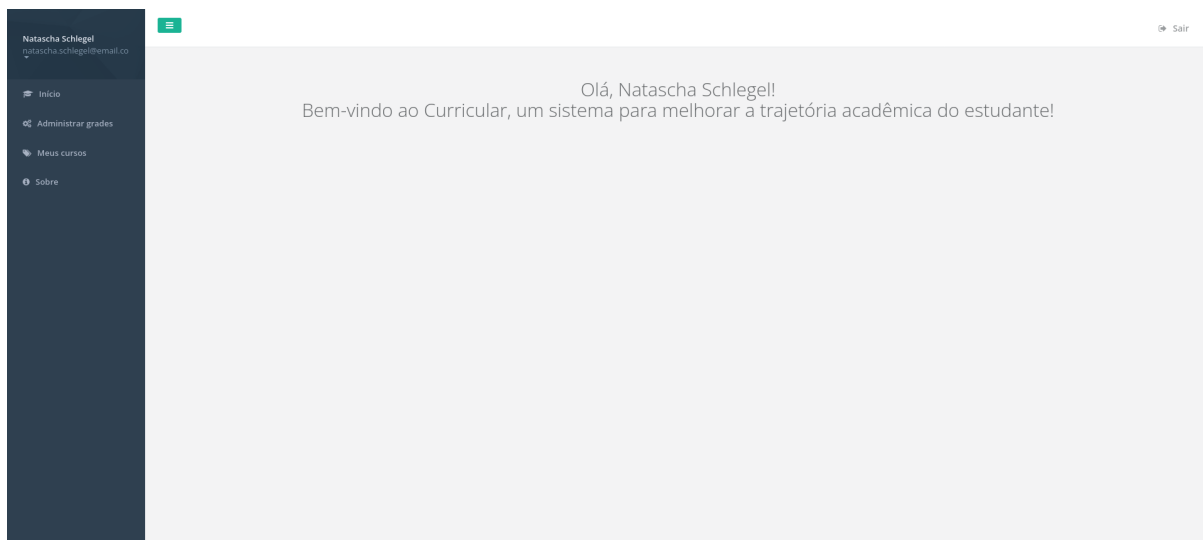


Figura 2.1: Tela Inicial do Sistema Curricular. Fonte: Captura de Tela

ríodo específico. [Lopes \(2022\)](#) então adicionou mais dois objetivos: garantir que as disciplinas cursadas no ciclo básico permaneçam nesses períodos no balanceamento e minimizar o acumulado dos índices de retenção do período, justificando que estes ainda não haviam recebido a devida consideração.

Originalmente, [Castro e Manzano \(2001\)](#) apresentaram o BACP, resolvendo-o utilizando Programação por Restrições e Programação Inteira, ao utilizar um algoritmo que testa todas as possibilidades possíveis, considerando a solução ótima aquela que respeita todas as restrições e tenha o melhor resultado do objetivo da função. O modelo apresentado pelos autores visa minimizar a soma da quantidade de créditos entre os períodos acadêmicos, estabelecendo um número inteiro limite para a maior carga de créditos possível entre todos os períodos, restringindo o espaço de busca do algoritmo. A medida que é encontrado um valor de soma menor que o limite, esse valor se torna o novo limite e, por fim, quando o problema não encontra valores menores que o limite definido, a solução encontrada é considerada uma solução ótima ([LOPES, 2022](#) apud [CASTRO; MANZANO, 2001](#)).

[Rosas et al. \(2011\)](#) utilizaram um algoritmo genético, pertencente a classe de algoritmos evolutivos que buscam resolver problemas de otimização, para balancear a grade curricular. A população (solução) é representada por um vetor em que a posição indica a disciplina e o valor da posição indica o período a que lhe foi atribuído. Já que a população é um indivíduo único, através da mutação, altera-se o período respeitando os pré-requisitos e demais restrições. O currículo é avaliado por uma função de aptidão que soma os erros absolutos e quanto menor

o valor, melhor balanceado esse currículo está. Na função objetivo desse modelo foi adicionada uma variável de decisão na qual é possível definir um período fixo para uma disciplina específica.

[Ünal e Uysal \(2014\)](#) implementaram uma nova abordagem baseada no BACP e utilizando Programação Linear Inteira Mista (MILP - Mixed Integer Linear Programming) para desenvolver o Balanceamento de Currículo Baseado em Relacionamento (RBCB - Relevance Based Curriculum Balancing). Além da uniformização da carga de créditos entre os períodos, é objetivo desse modelo diminuir a distância entre períodos das disciplinas que tenham em seus conteúdos didáticos alguma conexão, esta representada com uma pontuação de 0 a 9. Quanto maior sua conexão e maior a distância entre as disciplinas, maior será o custo imputado. O objetivo é reduzir este custo, consequentemente, aproximando as disciplinas inter-relacionadas ao longo da grade curricular.

Em suma, podemos listar os objetivos do BACF como:

- minimizar o número total de créditos por períodos;
- minimizar a distância entre disciplinas interconectadas;
- e minimizar o acumulado de índice de retenção.

Se tratando de um problema multiobjetivo, em que há mais de um objetivo para ser otimizado, o problema utiliza Programação Inteira Quadrática Mista e o método da soma ponderada, resultando em uma fronteira de pareto, em que há múltiplas soluções ótimas e a solução escolhida é feita conforme a necessidade ([KONAK et al., 2006](#)).

## 2.3 Trabalhos Relacionados

Dois trabalhos embasaram a proposta deste trabalho: a dissertação de [Lopes \(2022\)](#), que apresenta o BACF e o Curricular e o trabalho de [Rodrigues \(2022\)](#), que propõe uma interface para sistemas que utilizem o BACF.

Na dissertação de [Lopes \(2022\)](#), são explicados alguns problemas de otimização combinatória, sendo eles o Problema de Atribuição Generalizada, o Problema Quadrático de Atribuição Generalizada e a Programação Inteira Quadrática Mista juntamente com a Programação por Restrições. É explicado que a fronteira de Pareto é a solução ótima em uma otimização

multiobjetivo e descrito o estado da arte dos modelos de balanceamento de currículos, explorando as técnicas e abordagens utilizadas. A dissertação apresenta os resultados das hipóteses apresentadas, contribuindo para a validação do modelo proposto. E por fim, é apresentado a análise de uma pesquisa de usabilidade aplicada aos usuários do sistema, avaliando a percepção deles na utilização do sistema, fundamentando pontos de melhoria que podem vir a ser desenvolvidas.

O trabalho de [Rodrigues \(2022\)](#) enfatiza a ausência de um sistema que interaja com modelos criados para resolver o BACP. Seu objetivo é, portanto, proporcionar um sistema front-end ideal para qualquer aplicação que utilize modelos para resolver esse problema. Para isso, ele desenvolve várias páginas estáticas que, ao interagir com o usuário, recebem as informações necessárias para executar o processo de balanceamento. Além disso, apresenta também uma extensa documentação sobre ferramentas que podem ser utilizadas para a publicação do sistema em um servidor.

### 3 FERRAMENTAS E TECNOLOGIAS

Nesse capítulo, será abordado algumas ferramentas e tecnologias utilizadas durante o desenvolvimento: o Framework Web Django, a arquitetura cliente-servidor, o protocolo HTTP, a arquitetura MTV, o mapeamento objeto-relacional e funções assíncronas. É importante ressaltar que os tópicos podem estar relacionados entre si, mas não são necessariamente interdependentes.

#### 3.1 Framework Web Django

Segundo [Vincent \(2018\)](#), um Framework Web é uma coleção de ferramentas que simplificam diversos desafios frequentemente encontrados no desenvolvimento de uma aplicação acessada pelo navegador. O autor enfatiza que atualmente a maioria dos desenvolvedores optam por utilizar Frameworks. De acordo com [Rachapudi et al. \(2020\)](#), os Frameworks Web auxiliam em um desenvolvimento mais rápido e organizado, com segurança aprimorada e acesso fácil a bibliotecas externas. Por outro lado, são presentes problemas como restrições próprias do Framework, problemas de desempenho, redução da customização e custos adicionais ao implementar em Frameworks que demandam desenvolvedores com habilidades específicas e escassas.

O Framework Django começou a ser desenvolvido em 2003 por Adrian Holovaty e Simon Willison, na época desenvolvedores responsáveis pela criação e manutenção de sites de notícias que exigiam o desenvolvimento de recursos ou aplicações inteiras em prazos muito curtos ([HOLOVATY, 2008](#)). Ao migrar do PHP para Python procurando um desenvolvimento mais rápido, a equipe que reutilizava constantemente partes do código de programação, começou a elaborar um Framework, tornando-o público dois anos depois. ([FOUNDATION, 2023](#)).

[Holovaty \(2008\)](#) explica que o Django foi inicialmente implementado em cenários reais, em vez de criado com fins comerciais ou acadêmicos, direcionando o objetivo para solucionar problemas enfrentados pelos próprios desenvolvedores, recebendo atualizações contantes e buscando garantir a otimização do tempo de desenvolvimento, a facilidade na manutenção de aplicativos e o desempenho consistente mesmo sob cargas de trabalho intensas. [Vincent \(2018\)](#) detalha que o Django possui suporte para recursos como autenticação de usuários, roteamento de URL, interface administrativa, conexão em diferentes bancos de dados, entre outros.

### 3.2 Arquitetura Cliente-Servidor

O Django, sendo um Framework voltado para aplicações Web, adota a arquitetura cliente-servidor, comumente utilizada em sistemas distribuídos. A arquitetura é dividida em duas partes distintas: os servidores, provedores de recursos ou serviços, e os clientes, requerentes desses recursos ou serviços. Essa arquitetura permite distribuir a capacidade de processamento, já que inúmeros usuários ou clientes podem fazer requisições ao servidor, o qual responde com as respostas referentes a cada uma das requisições, como mostrado na figura 3.1. Ao mesmo tempo, essa estratégia reduz a carga no servidor, já que algumas interações do usuário podem ser processadas diretamente no navegador (OLUWATOSIN, 2014).

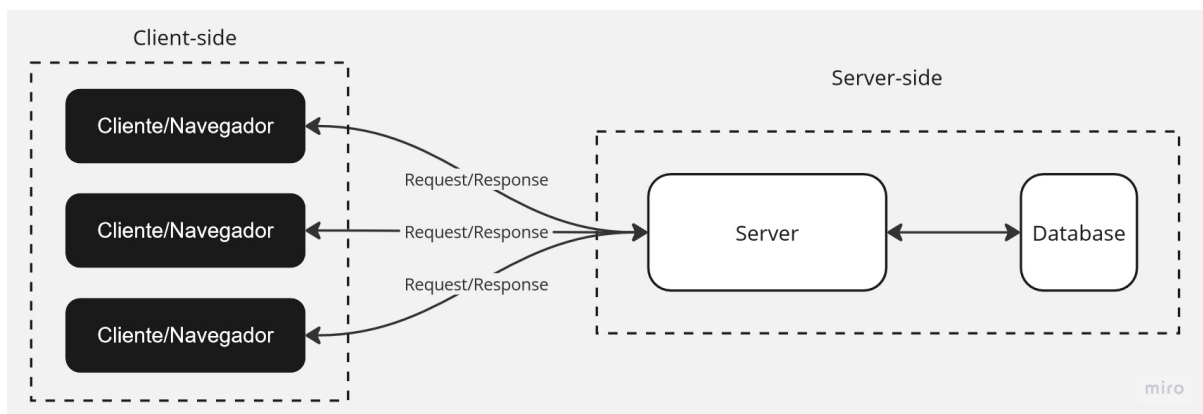


Figura 3.1: Arquitetura Cliente-Servidor. Fonte: Elaborada pelo autor

Taneja e Gupta (2014) especifica que o lado do cliente (client-side) é responsável por renderizar a interface, lidar com as interações do usuário, realizar requisições síncronas e as- síncronas buscando ou enviando informações. O lado do servidor (server-side) fica então res- ponsável por receber as requisições do navegador, tratar de acordo com sua lógica de negócios, manipular dados no banco de dados e responder a requisição com os dados solicitados.

### 3.3 Protocolo HTTP

Como explicado anteriormente, a arquitetura cliente-servidor é geralmente usada ao se desenvolver uma aplicação web. A interação entre o cliente e o servidor é realizada através do protocolo HTTP, em que o cliente envia uma solicitação para um recurso específico e o servi- dor responde com o resultado da solicitação. As solicitações (request) e respostas (response) incluem uma linha de início (start line) com informações essenciais para identificar o tipo da

mensagem e seu propósito, um cabeçalho (header) com informações sobre o tipo de conteúdo, tamanho do conteúdo, cookies, informações de autenticação, entre outros e, opcionalmente, um corpo (body) contendo dados da solicitação ou da resposta (DOCS, 2023), exemplificado na figura 3.2.

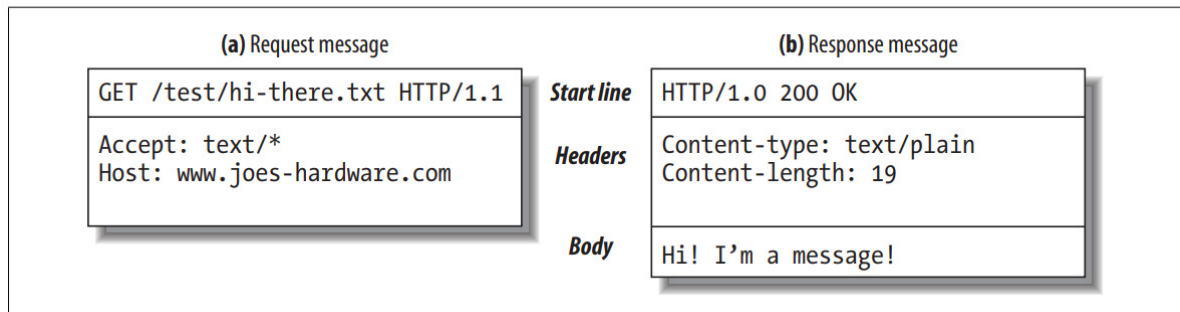


Figura 3.2: Protocolo HTTP. Fonte: Gourley et al. (2002)

O método HTTP, presente no header da solicitação, determina o tipo de operação que está sendo solicitada. Os métodos mais comuns são GET (para obter um recurso), POST (para enviar dados), PUT (para criar ou atualizar um recurso) e DELETE (para deletar um recurso). No header da resposta também está presente o código de status, um número de três dígitos, que determina o resultado da solicitação, as categorias dos códigos são mostradas na tabela 3.1

Código	Significado
1XX	O servidor recebeu e está prosseguindo com a solicitação
2XX	Sucesso na solicitação
3XX	O recurso foi movido de lugar
4XX	Há um erro na solicitação
5XX	Erro no servidor

Tabela 3.1: Fonte: Texto Adaptado de Gourley et al. (2002)

Na solicitação, a linha de início inclui o método, a URL do destino e versão do protocolo HTTP, no cabeçalho estão presentes informações necessárias para a transmissão da mensagem, e o corpo transmite os dados enviados do cliente para o servidor. Já na resposta, a linha de início inclui a versão do protocolo HTTP, código de status e frase que descreve o estado da operação legível para humanos, no cabeçalho são enviados informações que descrevem aos dados retornados e no corpo os dados requisitados (GOURLEY et al., 2002).

A figura 3.2 ilustra uma mensagem de solicitação à esquerda e à direita uma mensagem de resposta a solicitação. Na figura, uma solicitação é feita ao servidor por meio do método GET, requisitando um recurso localizado no caminho /tests/hi-there.txt usando o protocolo HTTP/1.1. O servidor então responde usando o protocolo HTTP/1.0 indicando o sucesso da operação pelo código de status 200 e a frase descritiva, e os dados solicitados no corpo (body) da resposta.

### 3.4 Arquitetura MTV

Para [Noriega et al. \(2021\)](#), padrões de projeto são soluções previamente estabelecidas para solucionar problemas comuns no desenvolvimento de software. Essas soluções são amplamente aceitas e seguem princípios sólidos de design que visam a modularidade, reutilização de código e facilidade de manutenção. Dentre os padrões de projeto, existem os padrões de projeto arquitetural, responsáveis por especificar a estrutura fundamental de uma aplicação ([BUSCH-MANN et al., 2013](#)). Ao separar a lógica de acesso a dados, a lógica de negócios e a lógica de apresentação, o Django segue a arquitetura MTV (Model-Template-View), um padrão de projeto arquitetural que procura dividir as responsabilidades entre três camadas principais, sendo projetado para incentivar o baixo acoplamento e a separação rigorosa entre as partes da aplicação, tornando possível a alteração de uma parte da aplicação sem afetar as outras ([HOLOVATY, 2008](#)).

Na arquitetura MTV, o Model representa a camada de acesso a dados, definindo a estrutura dos dados e oferecendo uma camada de abstração para interação com o banco de dados, o Template representa a camada de apresentação, sendo um arquivo HTML que renderiza as informações do servidor utilizando placeholders e a View representa a lógica de negócio, recebendo requisições, tratando e organizando dados do Model, controlando o que é apresentado ao usuário e renderizando Templates ou outros tipos de saída.

O processamento de uma requisição inicia quando o navegador envia uma solicitação ao servidor. A solicitação é recebida no módulo URLconfig, onde as URLs configuradas são mapeadas com suas respectivas Views. A View então recebe as informações enviadas pelo navegador, as processa, busca ou salva informações no banco de dados e, por fim, retorna uma resposta. A resposta pode assumir diferentes formas, por exemplo, se a requisição for para exibir uma página HTML, a View irá renderizar um Template com as informações necessárias



e enviá-lo como resposta. A View não se limita a retornar páginas HTML, podendo retornar arquivos, mensagens de erros, dados em formatos JSON, XML, entre outros.

A figura 3.3 ilustra como as camadas interagem, tanto entre si, dentro da própria aplicação quanto com o banco de dados e o navegador.

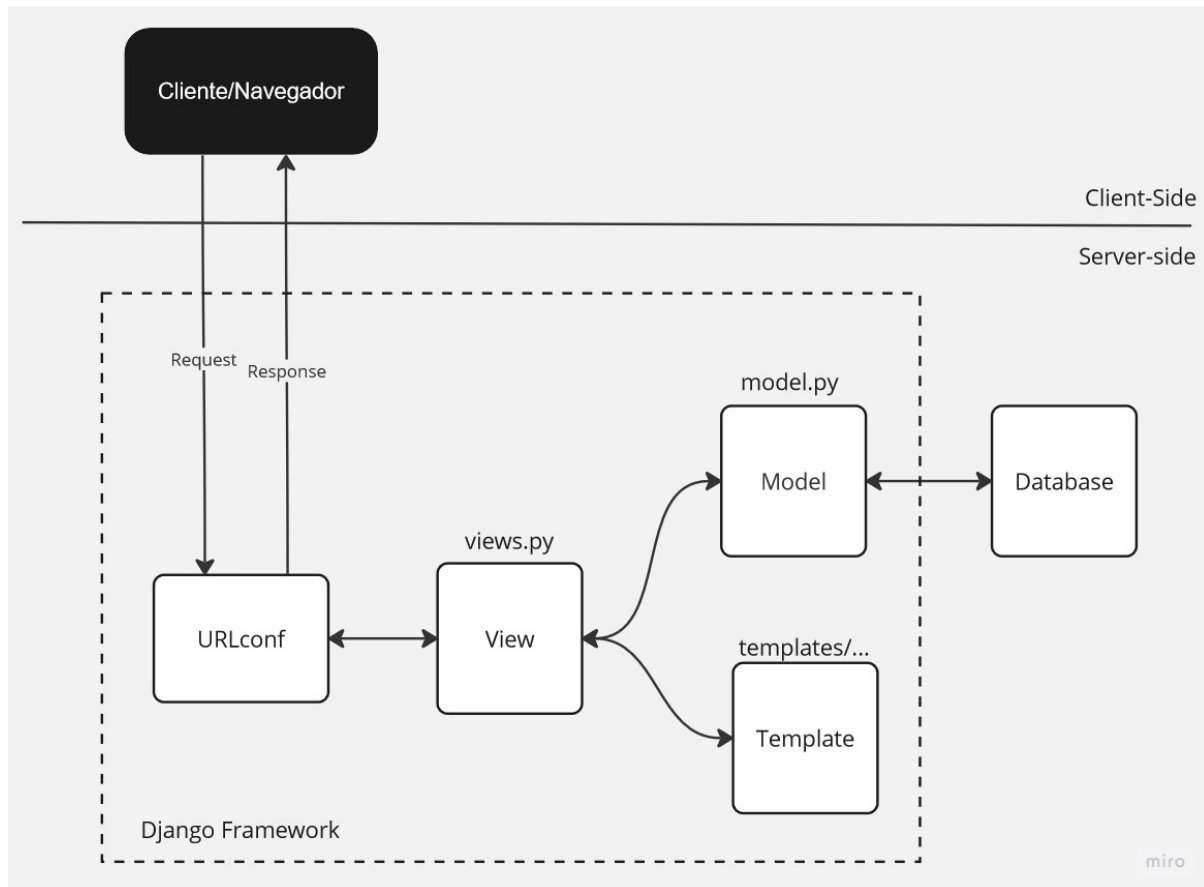


Figura 3.3: Arquitetura MTV. Fonte: Elaborada pelo autor

### 3.5 Mapeamento Objeto-Relacional

[Shaw et al. \(2021\)](#) explica que em aplicações Web desenvolvidas apenas com Python, a interação com o banco de dados é geralmente realizada através de comandos SQL (Structured Query Language), uma linguagem desenvolvida para gerenciamento de dados relacionais armazenados em uma estrutura de tabelas. No entanto, à medida que a aplicação escala e o número de tabelas aumenta, os comandos SQL podem se tornar complexos e difíceis de manter. Para tratar essa questão, alguns Frameworks Web oferecem a técnica de mapeamento objeto-relacional, que abstrai a interação com o banco de dados utilizando objetos e métodos próprios do Framework, tornando o processo mais intuitivo e simples.

Um mapeamento objeto-relacional permite que o desenvolvedor defina a estrutura do banco de dados usando linguagens de programação. Na ORM do Django, esse comandos mapeiam as tabelas através de classes e variáveis. Esse processo, conhecido como migração, converte as classes em comandos SQL e serve como um histórico de alterações e versionamento. Outra ferramenta do Django ORM é a interação com o banco de dados através de querySets, uma API baseada em métodos de classe capaz de adicionar, modificar, excluir e filtrar informações do banco de dados ([FOUNDATION, 2023](#)).

### 3.6 Funções Assíncronas

[Moore et al. \(2008\)](#) relata que a transição da WEB 1.0 para 2.0 foi impulsionada pela criação de aplicações Web que oferecem qualidade e funcionalidades semelhantes às aplicações de desktop, desencadeada por dois fatores: a popularização do Javascript e do AJAX. O Javascript permitiu que as interfaces tivessem comportamentos dinâmicos e interativos, enquanto o AJAX popularizou o uso de requisições assíncronas, possibilitando atualizações em tempo real sem a necessidade do recarregamento da página.

Ao contrário das funções síncronas, que executam suas operações sequencialmente, as funções assíncronas permitem que o programa continue sua execução enquanto uma operação assíncrona é processada. No Javascript, a declaração de uma função assíncrona é realizada através da adição da palavra `async` antes da declaração da função e a requisição a essa função é feita adicionando a palavra `await` antes da chamada. As funções assíncronas são executadas usando Promises, objetos que representam uma eventual conclusão de uma operação assíncrona. Quando uma Promise é chamada, ela é removida do Call Stack, onde estão sendo interpretadas as funções síncronas, e passa a outra thread, aguardando sua conclusão. Assim que a Promise é cumprida, ela é movida para o Event Loop, que avalia esporadicamente se há funções sendo executadas no Call Stack. Quando não há, envia a Promise para o Call Stack para retornar o resultado da execução.

## 4 MELHORIAS PROPOSTAS

### 4.1 Justificativa

Após a disponibilização para testes do Curricular e através de um questionário aplicado aos potenciais usuários do sistema, foram coletadas respostas relacionados a usabilidade do sistema Curricular. Para isso foi utilizado o Questionário de Experiência de Usuário (UEQ - User Experience Questionnaire), projetado por [Laugwitz et al. \(2008\)](#). O questionário avalia o sistema sobre duas dimensões: pragmática e hedônica. A dimensão pragmática avalia aspectos funcionais e utilitários de um produto, focando no desempenho do produto, eficiência e confiabilidade, enquanto a dimensão hedônica avalia aspectos emocionais e experienciais de um produto avaliando a estética, apelo sensorial e experiência geral ao utilizar um produto ([HASSENZAHN et al., 2011](#)).

O questionário foi desenvolvido de forma que há dois adjetivos opostos em cada extremidade de uma escala com sete níveis e o usuário escolhe a opção que melhor descreve sua percepção. Os seguintes aspectos do software são avaliados: atratividade, controle, eficiência, inovação, transparência e estimulação. Esses aspectos são organizados de forma que cada um dos aspectos possui uma ou mais duplas de adjetivos antônimos entre si. O valor das opções na escala começam em -3 e sobem até 3, sendo que os adjetivos que descrevem aspectos ruins tem valor -3 e os adjetivos que descrevem aspectos bons tem valor 3, no total há 26 duplas de adjetivos sendo testadas.

Além do questionário UEQ, foi adicionado ao formulário duas perguntas que poderiam ser respondidas sem limite de caracteres, sendo elas: "Na sua opinião, o software Curricular pode contribuir para a melhoria da trajetória acadêmica do estudante? Por quê?" e "Caso queira, dê mais opiniões sobre o software Curricular".

Após o recolhimento das respostas, foram calculados as médias das duplas de adjetivos, e posteriormente são agrupados os adjetivos de mesma categoria e calculada a média da categoria. Segundo [Lopes \(2022\)](#), todas as médias da categorias foram acima de 0.8 pontos, o que indica uma avaliação positiva do sistema.

Como mostrado na tabela 4.1, a categoria Transparência, que procura entender se a ferramenta é facilmente entendida e utilizada, não foi classificada como excelente. Analisando as respostas escritas, nota-se algumas oportunidades de melhoria na interface do sistema. Nota-

Escala	Média	Comparação com benchmark	Interpretação
Atratividade	2,18	Excelente	No intervalo dos 10% melhores resultados
Transparência	1,62	Acima da média	No intervalo dos 25% melhores resultados
Eficiência	2,05	Excelente	No intervalo dos 10% melhores resultados
Controle	1,84	Excelente	No intervalo dos 10% melhores resultados
Estimulação	2,17	Excelente	No intervalo dos 10% melhores resultados
Inovação	1,95	Excelente	No intervalo dos 10% melhores resultados

Tabela 4.1: Fonte: Adaptado de [Lopes \(2022\)](#)

velmente, três dessas respostas coincidem para uma mesma sugestão de aprimoramento.

- “Na opção de escolher as disciplinas que já foram concluídas, talvez um layout checkbox [sic] seria melhor.”
- “Na parte de selecionar as disciplinas, deveria ter uma caixa de marcar para ser mais ágil na seleção das mesmas.”
- “O software cumpre o que promete. O que precisa, talvez, é só fazer alguns ajustes como colocar caixas de marcação para selecionar as disciplinas. Mas muito show o software.”

O problema relatado nas respostas se refere a inserção de disciplinas que ainda não foram cursadas pelo usuário. Originalmente, a inserção era feita uma disciplina por vez, como demonstrado na figura 4.1, exigindo que o usuário preenchesse o formulário e clicasse em "Salvar" para cada disciplina não cursada, a página então recarrega exibindo um aviso se foi possível adicionar a disciplina não cursada, caso o usuário quera sair desse fluxo após adicionar todas as disciplinas não cursadas, precisa clicar no botão "Voltar".

Com base nas respostas coletadas no questionário e nas observações feitas, justifica-se a reformulação do fluxo que trata da inserção de disciplinas ainda não cursadas. Para isso, foi desenvolvido um diagrama retratando o comportamento anterior e um diagrama com as possíveis alterações para solucionar o problema em questão.

A captura de tela mostra uma interface web com o título "Administração de disciplinas da grade" no topo. No canto superior direito, há um botão verde com o texto "← Voltar". Abaixo, há uma seção intitulada "Selecionar disciplinas não cursadas". Dentro desta seção, há um formulário com os seguintes campos: "Grade" (com o valor "Grade Natascha - SIN2021 - Sistemas de Informação - UFVJM"), "Disciplina:" (com o valor "ALGORITMOS E ESTRUTURA DE DADOS I - COM001 - Sistemas de Informação - UFVJM"), "Período fixo:" (com o valor "Não") e "Período em que a disciplina deve ser fixada:" (um campo vazio). No canto inferior esquerdo do formulário, há um botão verde com o texto "✓ Salvar".

Figura 4.1: Captura de tela da página original. Fonte: Elaborada pelo autor

## 4.2 Diagrama de Fluxo de Usuário

Através do diagrama de fluxo do usuário, demonstrados nas figuras 4.2 e 4.3, é possível descrever o comportamento implementado anteriormente, as diferenças entre o antigo e o novo comportamento e quais melhorias o novo comportamento pode trazer ao usuário. O diagrama descreve o fluxo de forma que os cartões verdes representam componentes que levam o usuário de uma página a outra, os cartões rosas representam ações realizadas dentro de uma mesma página, como o preenchimento de formulários, e os triângulos azuis representam decisões do usuários.

O diagrama novo propõe a criação de duas novas páginas substituindo a página desenvolvida anteriormente, adaptando a tela para que seja feita primeiramente a inserção das disciplinas que já foram cursadas em um único formulário, e ao clicar para salvar o formulário, o usuário é direcionado à uma segunda página, contendo as disciplinas que não foram marcadas como cursadas pelo usuário. Nessa página, o usuário pode confirmar suas escolhas anteriores (observando se nessa lista está presente alguma disciplina que já foi cursada) e caso queira, definir um período fixo para alguma disciplina.

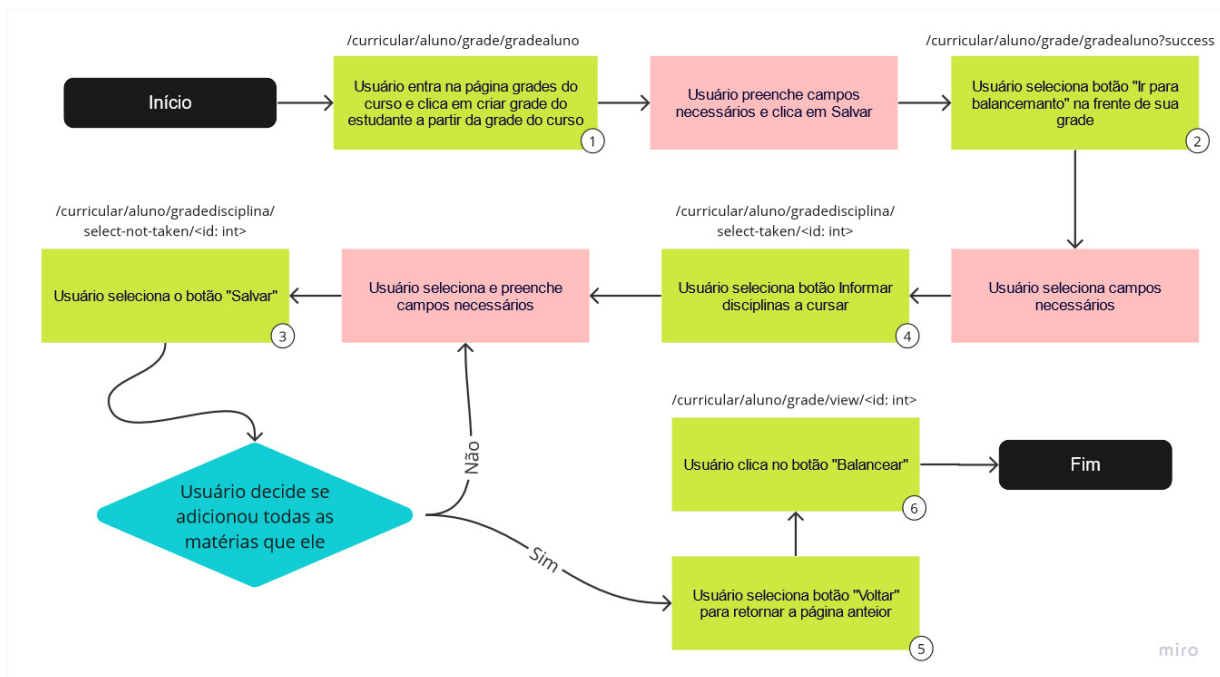


Figura 4.2: Fluxograma Antigo. Fonte: Elaborada pelo autor

### 4.3 Implementação

Após compreender o contexto no qual o sistema foi implementado no capítulo dois, as ferramentas e tecnologias utilizadas no capítulo três e apresentar as alterações propostas nas seções anteriores deste capítulo, nessa seção será abordado a codificação.

As principais implementações realizadas foram a criação de quatro novas views, dois templates, um model, dois arquivos Javascript, além de alterações ocasionais em arquivos já existentes.

O terceiro passo presente na figura 4.3 usa o template select-taken.html, o arquivo javascript select-taken.js e a view aluno\_gradedisciplina\_select\_taken. O quarto passo usa o template select-not-taken.html, o arquivo javascript select-not-taken.js e a view aluno\_gradedisciplina\_select\_not\_taken.

Como explicado anteriormente, a inserção de disciplinas não cursadas era feita individualmente no formulário, e da mesma forma no banco de dados, significando que o sistema não tinha conhecimento de quais disciplinas foram cursadas pelo aluno, apenas as disciplinas não cursadas. De certa forma não havia prejuízo na falta dessa informação, já que o objetivo era balancear exatamente as disciplinas não cursadas, porém para a implementação da ideia apre-

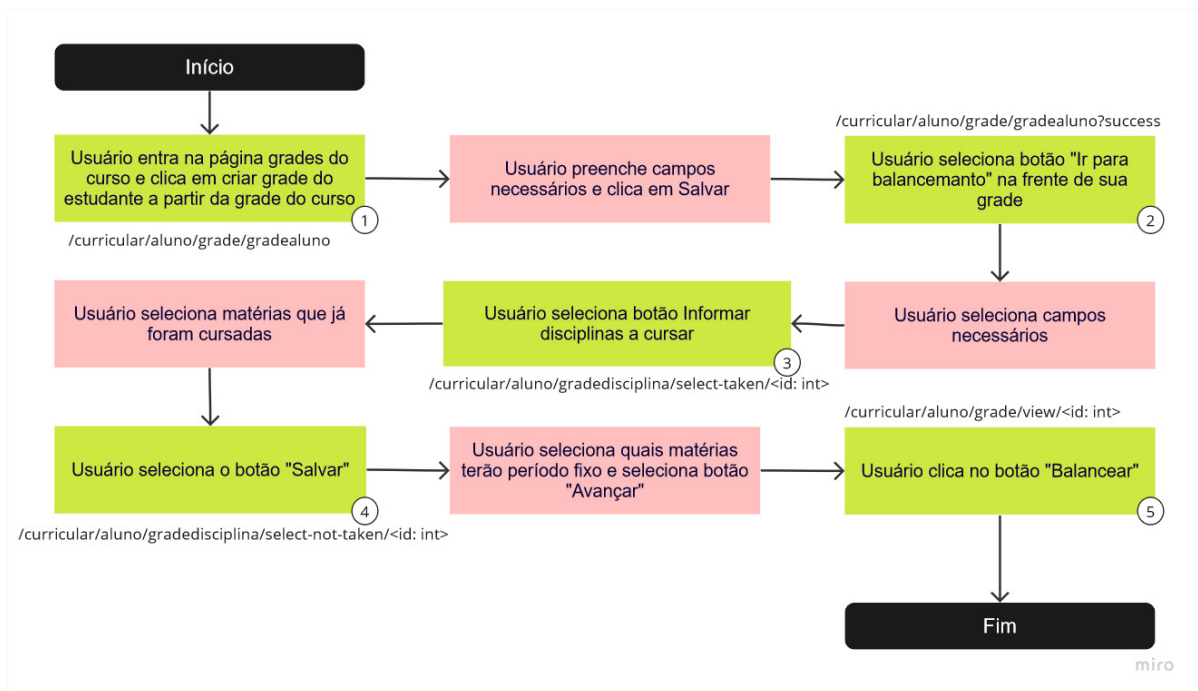


Figura 4.3: Fluxograma Novo. Fonte: Elaborada pelo autor

sentada nesse trabalho, foi necessário modificar esse comportamento adicionando uma nova página para a inserção de disciplinas cursadas pelo aluno e a adição de uma nova tabela no banco de dados que recebe as disciplinas que foram cursadas.

A página para inserção de disciplinas cursadas recebe do banco de dados todas as disciplinas pertencentes a grade curricular selecionada pelo aluno e através de um checkbox, o usuário poderá selecionar as disciplinas que foram cursadas por ele. Nesse momento é enviado uma requisição assíncrona à view `aluno_gradedisciplina_checkrequisites` com o id da disciplina para que seja buscado no banco de dados informações sobre pré-requisitos, retornando três informações: uma lista com os ids das disciplinas que são pré-requisitos, uma lista com os nomes das disciplinas que são pré-requisitos e uma lista com o id das disciplinas que possuem a disciplina enviada como pré-requisito.

Usando Javascript, a página dispõe de comportamentos que previnem uma interação errada do usuário, com os seguintes comportamentos esperados:

- Ao carregar a página, as disciplinas que já foram adicionadas anteriormente como cursadas tem seu checkbox selecionado.
- Caso o usuário clique em uma disciplina em que o pré-requisito não tenha sido selecio-

nado, seu checkbox é desmarcado e é avisado ao usuário o motivo.

- Como as disciplinas estão divididas em períodos, caso o usuário clique em Marcar Todas as Disciplinas, os checkboxes das disciplinas do período são marcados e é feita a verificação se existe pré-requisitos não marcados, caso exista a disciplina em questão é desmarcada e é avisado ao usuário o motivo.
- Caso o usuário marque uma disciplina e seu pré-requisito na sequência correta, mas posteriormente clique para desmarcar o pré-requisito, automaticamente é desmarcado também a disciplina.

Ao fim da marcação, ao clicar em Avançar o usuário será redirecionado a segunda página do fluxo, no qual será listado todas as disciplinas que não foram cursadas pelo usuário. O usuário pode então verificar se todas as disciplinas presentes não foram cursadas, e caso note a falta de alguma ou perceba que uma disciplina que já foi cursada está presente, o usuário pode voltar a página anterior e verificar se o checkbox está marcado. Além disso o usuário pode fixar em um período em uma determinada disciplina através da coluna Período Fixo. Nas linhas dessa coluna estão presentes um checkbox e um dropdown, o estado do dropdown está coerente com a marcação do checkbox, caso o checkbox não esteja selecionado, o dropdown deve estar inativo, caso o checkbox seja marcado, o dropdown deve se tornar disponível ao usuário.

Caso o usuário queira definir um período fixo clicando no checkbox, ele deve clicar no checkbox e selecionar o número referente ao período no dropdown, ao selecionar esse número, é feita uma chamada assíncrona através da view `aluno_gradedisciplina_checkrequisites`, caso o período informado possa ser fixado na disciplina, a API retornará o status code 200, de sucesso, caso não possa, retornará o status code 422, informando que não há erro na sintaxe da requisição, mas não foi possível processar as instruções. Com essa informação, o Javascript retorna ao usuário a resposta através de um aviso. Na segunda página estão presentes os seguintes comportamentos:

- Ao carregar a página, todos os elementos dropdown estão desativados.
- Ao fixar um período, o sistema informa ao usuário o resultado da solicitação.
- Caso o usuário desista do período fixo e clique para desmarcar o checkbox, a opção selecionada voltara a ser padrão (Selecione o semestre fixo) e o período fixo é apagado



no banco de dados.

Ao clicar no botão de avançar, o usuário é direcionado a primeira tela do fluxo, na qual pode definir os termos do seu balanceamento e clicar no botão Balancear.

## 5 RESULTADO

Para demonstrar os resultados das modificações propostas, será feito o caminho de um aluno adicionando suas disciplinas. Para isso, supõe-se que o aluno esteja no quarto período e tenha cursado as seguintes disciplinas: Fundamentos de Sistemas de Informação, Introdução à Logica Computacional, Leitura e Produção de Textos, Matemática Discreta, Metodologia do Trabalho e da Pesquisa Científica e Tecnológica, Administração I, Fundamentos de Economia, Teoria da Computação.

A captura de tela mostra a interface de administração da grade curricular. No topo, há o título "Administração de grades" e um botão "Voltar" verde. Abaixo, a seção "Detalhes da grade curricular" contém informações sobre a instituição (Universidade Federal dos Vales do Jequitinhonha e Mucuri - UFVJM), o curso (Sistemas de Informação - SIUFVJM), o currículo (SIN2021), o nome da grade (Grade Natascha), o número de períodos disponíveis (6) e o número de períodos já passados (3). O status é "Aguardando balanceamento".

Abaixo, a seção "Configurações do balanceamento" contém sete opções de configuração, todas com um menu suspenso "Sim":

- Balancear a carga de créditos?
- Reduzir as localizações de disciplinas retentoras em um mesmo período?
- Reduzir as distâncias entre disciplinas inter-relacionadas?
- Aplicar restrição de ciclo básico?
- Aplicar restrição de período fixo?
- Aplicar restrição de carga máxima?
- Aplicar restrição de quantidade máxima de disciplinas?
- Aplicar restrição de pré-requisitos?

No rodapé da seção, há um botão azul "+ Informar disciplinas a cursar".

Figura 5.1: Página de administração da grade. Fonte: Captura de Tela

Guiando pelo fluxograma da figura 4.3, o caminho começa na página de Administração de grades, mostrada na figura 5.1. Nessa página, o aluno clica em "Informar disciplinas a cursar" e é direcionado à página seguinte, mostrada na figura 5.2, terceiro passo do fluxograma no intuito de adicionar as disciplinas cursadas por ele.

Enquanto está definindo as disciplinas que já cursou, acidentalmente clica na disciplina Algoritmos E Estrutura De Dados II, que tem como pré-requisito a disciplina Algoritmos E Estrutura De Dados I que não foi cursada por ele. Então o sistema informa ao usuário que não é possível fazer a marcação dessa disciplina, como demonstrado na figura 5.3.

Clicando na disciplina Algoritmos E Estrutura De Dados II, podemos ver a chamada assíncrona feita e a resposta na figura 5.4. É feito a requisição enviando o id da disciplina e a resposta contém os ids das disciplinas que são pré-requisitos, o nome dessas disciplinas e o ids das disciplinas que tem como pré-requisito a disciplina enviada, nesse caso, AEDS II.

Feita todas as marcações e clicando em avançar, o usuário é levado à próxima página,

Administração de disciplinas cursadas

Voltar

Selecionar disciplinas cursadas

1º período ▾

☐ Selecionar todas as disciplinas

☐ Fundamentos De Matemática - MAT001

☒ Fundamentos De Sistemas De Informação - COM040

☐ Inglês Instrumental - COM043

☒ Introdução À Logica Computacional - MAT007

☒ Leitura E Produção De Textos - COM059

2º período ▾

☐ Selecionar todas as disciplinas

☐ Algoritmos E Estrutura De Dados I - COM001

☐ Cálculo Diferencial E Integral I - MAT003

☒ Matemática Discreta - MAT006

☒ Metodologia Do Trabalho E Da Pesquisa Científica E Tecnológica - COM060

Figura 5.2: Página para adicionar disciplinas cursadas. Fonte: Captura de Tela

demonstrado na figura 5.5, quarto passo do fluxograma, em que são listadas todas as disciplinas que não foram cursadas por ele.

Caso o usuário deseje fixar a disciplina Banco de Dados I no 5º período, ele clica no checkbox e seleciona 5 no dropdown. Ao fazer isso, o sistema retorna o aviso de sucesso ao adicionar no banco de dados, mostrado na figura 5.6.

Depois de fixar a disciplina de Banco de Dados I no 5º período, o usuário deseja fixar também a disciplina Banco de Dados II no 5º período, porém isso não é possível, já que seu pré-requisito Banco de Dados I está fixado nesse período. O sistema então avisa o usuário sobre a impossibilidade de inserção no banco de dados, mostrado na figura 5.7.

Na figura 5.8 podemos ver a requisição enviada para fixar um período.

Após definir os períodos fixos, o usuário clica em avançar e retornará para a página de Administração de grades, mostrada na figura 5.9, último passo do fluxograma, agora com o botão de "Balancear" e "Apagar Disciplinas A Cursar" visíveis.

Para avaliar a eficácia das alterações realizadas e a inclusão das disciplinas não cursadas de forma apropriada, verificamos a página "Disciplinas a Serem Cursadas", representada na

3º período ▾

☐ Selecionar todas as disciplinas

☒ Administração I - COM007

☐ Algoritmos E Estrutura De Dados II - COM004

Você precisa ter cursado a disciplina ALGORITMOS E ESTRUTURA DE DADOS I para cursar essa disciplina.

☒ Fundamentos De Economia - COM008

☐ Geometria Analítica E Álgebra Linear - MAT002

☐ Organização E Arquitetura De Computadores - COM005

☒ Teoria Da Computação - COM006

Figura 5.3: Aviso de erro. Fonte: Captura de Tela

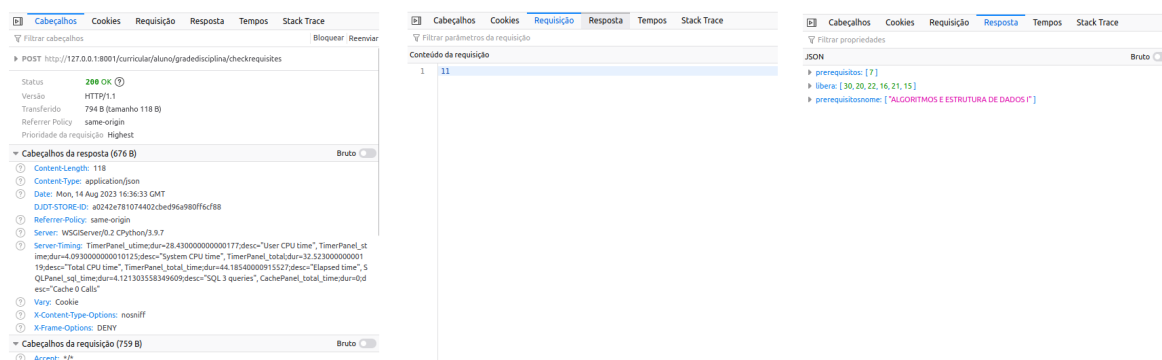


Figura 5.4: Chamada de requisição. Fonte: Captura de tela

figura 5.10, na qual todas as disciplinas não cursadas foram devidamente adicionadas.

Na figura 5.11, a disciplina Banco de Dados I, em que foi definido o período fixo com sucesso, mostra o período fixo marcado como Sim, e o número do período fixado 5.

A figura 5.12 mostra a disciplina Banco de Dados II, em que não foi possível fixar o período, mostrando o período fixo marcado como Não.

## Administração de disciplinas não cursadas

[Voltar](#)

### Selecionar disciplinas cursadas

Código	Nome	Período Fixo
COM011	ADMINISTRAÇÃO II	<input type="checkbox"/> <span>Selecione o semestre fixo</span>
COM001	ALGORITMOS E ESTRUTURA DE DADOS I	<input type="checkbox"/> <span>Selecione o semestre fixo</span>
COM004	ALGORITMOS E ESTRUTURA DE DADOS II	<input type="checkbox"/> <span>Selecione o semestre fixo</span>
COM009	ALGORITMOS E ESTRUTURA DE DADOS III	<input type="checkbox"/> <span>Selecione o semestre fixo</span>
COM015	BANCO DE DADOS I	<input type="checkbox"/> <span>Selecione o semestre fixo</span>

Figura 5.5: Adicionar disciplinas não cursadas. Fonte: Captura de Tela

COM015	BANCO DE DADOS I	<input checked="" type="checkbox"/>	<span>5</span>
<span>Disciplina Adicionada</span>			

Figura 5.6: Aviso de sucesso. Fonte: Captura de Tela

COM015	BANCO DE DADOS I	<input checked="" type="checkbox"/>	<span>5</span>
COM021	BANCO DE DADOS II	<input type="checkbox"/>	<span>Selecione o semestre fixo</span>

Operação não realizada! Tentativa de fixação da disciplina BANCO DE DADOS II no período 4, que é anterior ou igual ao período fixado para a disciplina BANCO DE DADOS I, seu pré-requisito.

Figura 5.7: Aviso de erro. Fonte: Captura de Tela

Cabeçalhos

Cookies

Requisição

Resposta

Tempos

Stack Trace

Filtrar cabeçalhos

Bloquear

Reenviar

POST http://127.0.0.1:8000/curricular/aluno/gradedisciplina/checksemester

Status: 200 OK

Versão: HTTP/1.1

Transferido: 283 B (tamanho 51 B)

Referer Policy: same-origin

Prioridade da requisição: Highest

Cabeçalhos da resposta (232 B)

Content-Length: 51

Content-Type: application/json

Date: Tue, 26 Sep 2023 23:39:36 GMT

Referer Policy: same-origin

Server: WSGIServer/0.2 CPython/3.9.7

X-Content-Type-Options: nosniff

X-Frame-Options: DENY

Cabeçalhos da requisição (754 B)

Accept: \*/\*

Accept-Encoding: gzip, deflate, br

Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3

Connection: keep-alive

Content-Length: 77

Content-Type: application/json

Cookie: csrftoken=uRc2TtnK8Rr6q\*1uGyocpz2M9f3EDwscq2qUgt6MH633a8OwONqnFWFRuJsh

Cabeçalhos

Cookies

Requisição

Resposta

Tempos

Stack Trace

Filtrar parâmetros da requisição

Bruto

JSON

boolPeriodo: "5"

gradeDisciplinaId: "5"

gradeId: "4"

numeroPeriodo: "5"

Cabeçalhos

Cookies

Requisição

Resposta

Tempos

Stack Trace

Filtrar propriedades

Bruto

JSON

message: "Disciplina Adicionada"

status: 200

Figura 5.8: Chamada assíncrona para fixar período. Fonte: Captura de Tela

Administração de grades

← Voltar

Detalhes da grade curricular

Instituição: Universidade Federal dos Vales do Jequitinhonha e Mucuri - UFVJM

Nome: Grade Natascha

Períodos disponíveis: 6

Status: Aguardando balanceamento

Curso: Sistemas de Informação - SIUFVJM

Grade do curso: Grade Oficial

Número de períodos que já se passaram desde seu ingresso no curso: 3

Currículo: SIN2021

Configurações do balanceamento ⓘ

Balancear a carga de créditos?

Sim

Reduzir as localizações de disciplinas retentoras em um mesmo período?

Sim

Reduzir as distâncias entre disciplinas inter-relacionadas?

Sim

Aplicar restrição de ciclo básico?

Sim

Aplicar restrição de período fixo?

Sim

Aplicar restrição de carga máxima?

Sim

Aplicar restrição de quantidade máxima de disciplinas?

Sim

Aplicar restrição de pré-requisitos?

Sim

Balancear

+ reformar disciplinas a cursar

Apagar disciplinas a cursar

Figura 5.9: Administração de Grade. Fonte: Captura de Tela



















































































































Administração de disciplinas da grade					
Disciplinas da grade					
<div>Grade:</div> <div>Escolha a grade</div> <div>50 resultados por página</div> <div>Pesquisar</div>					
Nome	↑ ↓ Grade do estudante	↑ ↓ Grade do curso	↑ ↓ Curso	↑ ↓ Ação	
ADMINISTRAÇÃO II - COM011	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
ALGORITMOS E ESTRUTURA DE DADOS I - COM001	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
ALGORITMOS E ESTRUTURA DE DADOS II - COM004	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
ALGORITMOS E ESTRUTURA DE DADOS III - COM009	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
BANCO DE DADOS I - COM015	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
BANCO DE DADOS II - COM021	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
COMPORTAMENTO ORGANIZACIONAL - COM030	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
CÁLCULO DIFERENCIAL E INTEGRAL I - MAT003	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
DIREITO LEGISLAÇÃO EM INFORMÁTICA - COM013	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
EMPREENDEDORISMO - COM035	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
ENGENHARIA DE SOFTWARE I - COM023	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
ENGENHARIA DE SOFTWARE II - COM027	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
ENGENHARIA WEB - COM020	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
ESTATÍSTICA - MAT004	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
FUNDAMENTOS DE CONTABILIDADE - COM012	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
FUNDAMENTOS DE MATEMÁTICA - MAT001	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
GEOMETRIA ANALÍTICA E ÁLGEBRA LINEAR - MAT002	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
GERÊNCIA DE PROJETOS DE SOFTWARE - COM033	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
GESTÃO DE SISTEMAS DE INFORMAÇÃO - COM018	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
INFORMÁTICA E SOCIEDADE - COM025	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
INGLÊS INSTRUMENTAL - COM043	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
INTELIGÊNCIA ARTIFICIAL - COM024	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
INTERFACE HOMEM MÁQUINA - COM028	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
LINGUAGENS DE PROGRAMAÇÃO - COM014	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES - COM005	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
PESQUISA OPERACIONAL - COM017	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
PROGRAMAÇÃO ORIENTADA A OBJETOS - COM016	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
PROJETO ORIENTADO I (TCC) - COM036	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
PROJETO ORIENTADO II (TCC) - COM038	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
REDES DE COMPUTADORES I - COM022	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
REDES DE COMPUTADORES II - COM026	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
SEGURANÇA E AUDITORIA DE SISTEMAS DE INFORMAÇÃO - COM034	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
SISTEMAS DE APOIO À DECISÃO - COM019	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
SISTEMAS DE COMPUTAÇÃO - COM002	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
SISTEMAS DISTRIBUÍDOS - COM032	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
SISTEMAS OPERACIONAIS - COM010	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
TEORIA GERAL DOS SISTEMAS - COM003	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
TRABALHO COOPERATIVO APOIADO POR COMPUTADOR - COM029	Grade Natascha	Grade Oficial	Sistemas de Informação - UFVJM	  	
Mostrando de 1 até 38 de 38 registros				Anterior 1 Próximo	

Figura 5.10: Página Disciplina A Cursar. Fonte: Captura de Tela

## Administração de disciplinas da grade

[← Voltar](#)

### Detalhes da disciplina da grade

**Instituição:** Universidade Federal dos Vales do Jequitinhonha e Mucuri - UFVJM

**Curso:** Sistemas de Informação - SIUFVJM

**Grade do estudante:** Grade Natascha

**Grade do curso:** Grade Oficial

**Disciplina:** BANCO DE DADOS I - COM015

**Créditos:** 4

**Retenção:** 26,0%

**Ciclo básico:** Não

**Período fixo:** Sim

**Período fixado:** 5

**Disciplinas anteriores do relacionamento:** ALGORITMOS E ESTRUTURA DE DADOS II - COM004

**Disciplinas posteriores do relacionamento:** BANCO DE DADOS II - COM021

**Pré-requisitos:** ALGORITMOS E ESTRUTURA DE DADOS II - COM004

**Disciplinas dependentes do pré-requisito:** BANCO DE DADOS II - COM021

Figura 5.11: Administração de disciplina de Grade. Fonte: Captura de Tela

### Detalhes da disciplina da grade

**Instituição:** Universidade Federal dos Vales do Jequitinhonha e Mucuri - UFVJM

**Curso:** Sistemas de Informação - SIUFVJM

**Grade do estudante:** Grade Natascha

**Grade do curso:** Grade Oficial

**Disciplina:** BANCO DE DADOS II - COM021

**Créditos:** 4

**Retenção:** 24,0%

**Ciclo básico:** Não

**Período fixo:** Não

**Disciplinas anteriores do relacionamento:** BANCO DE DADOS I - COM015

**Pré-requisitos:** BANCO DE DADOS I - COM015

Figura 5.12: Administração de disciplina de Grade. Fonte: Captura de Tela



## 6 CONCLUSÃO

A carga de trabalho inadequada prejudica à assimilação do conteúdo pelos alunos, aumentando as chances de reprovação e, conseqüentemente, contribuindo para a retenção ou até mesmo a desistência do curso, acarretando conseqüências significativas para todos os envolvidos no processo educacional.

Uma das medidas possíveis para sanar o problema é a criação de grades curriculares adequadas. Através do modelo proposto por [Lopes \(2022\)](#) para resolver o Problema de Balanceamento de Currículo Acadêmico, promover um equilíbrio na carga de trabalho, colabora para um ambiente mais propício ao sucesso acadêmico.

O Sistema Curricular, projetado de forma acessível para ser adotado por qualquer instituição de ensino superior, tem o potencial de oferecer apoio ao estudante ao longo de sua jornada acadêmica, tornando a experiência educacional mais agradável, eficaz e enriquecedora, ao mesmo tempo que auxilia as instituições no enfrentamento dos desafios relacionados à retenção e evasão estudantil.

Neste trabalho, foram identificados e implementados melhorias de usabilidade no processo de inserção das disciplinas necessárias para balanceamento das grades curriculares. Para alcançar esse objetivo, foi necessário compreender o planejamento inicial e o desenvolvimento do sistema Curricular, para então encontrar uma solução que integrasse ao sistema sem causar impactos em outras partes do mesmo. Ao fixar o período de uma disciplina não cursada, há uma complexa comparação com dados existentes no banco de dados para garantir a conformidade com o projeto pedagógico, requerendo a adaptação de um algoritmo originalmente desenvolvido.

A forma escolhida para simplificar e agilizar o processo foi utilizar chamadas de funções assíncronas feitas pelo navegador ao back-end avaliando se as escolhas do usuário condizem com as condições esperadas pelo curso na realidade. Ao avaliar se o aluno não está cometendo o erro de adicionar uma disciplina sem ter cursado seu pré-requisito e avaliar se o período fixo escolhido não causa atrito com outras disciplinas, assegura-se que não irá ocorrer erros no balanceamento das grades curriculares, nem propor ao usuário uma solução incompatível com o projeto pedagógico do curso.

Trabalhos futuros relacionados ao sistema curricular podem direcionar seus esforços

para resolver outras questões de usabilidade identificados pelos usuários na dissertação de [Lopes \(2022\)](#) ou introduzir novas funcionalidades que ampliem as funcionalidades oferecidas aos usuários.

## REFERÊNCIAS

- AMORIM, E. d. M. Evasão escolar no ensino superior: um estudo de caso na universidade federal dos vales do jequitinhonha e mucuri. 2016. Disponível em: [http://acervo.ufvjm.edu.br/jspui/bitstream/1/1171/1/evandro\\_merces\\_amorim.pdf](http://acervo.ufvjm.edu.br/jspui/bitstream/1/1171/1/evandro_merces_amorim.pdf). Acesso em: 2023-10-23.
- ARAÚJO, A. C. P. L. d.; MARIANO, F. Z.; OLIVEIRA, C. S. d. Determinantes acadêmicos da retenção no ensino superior. **Ensaio: Avaliação e Políticas Públicas em Educação**, Fundação CESGRANRIO, v. 29, n. 113, p. 1045–1066, Oct 2021. ISSN 0104-4036. Disponível em: <https://doi.org/10.1590/S0104-403620210002902255>. Acesso em: 2023-10-23.
- BIGGS, J. B. **Study Process Questionnaire Users Manual**. Estados Unidos: Australian Council For Educational, 1987. ISBN 9780864310026. Disponível em: [https://www.researchgate.net/publication/242530091\\_Study\\_Process\\_Questionnaire\\_Manual](https://www.researchgate.net/publication/242530091_Study_Process_Questionnaire_Manual). Acesso em: 2023-10-23.
- BURON, R. M. O papel da universidade na formação do perfil profissional. **Salão do Conhecimento**, v. 2, n. 2, set. 2016. Disponível em: <https://publicacoeseventos.unijui.edu.br/index.php/salaconhecimento/article/view/7307>. Acesso em: 2023-10-23.
- BUSCHMANN, F.; MEUNIER, R.; ROHNERT, H.; SOMMERLAD, P.; STAL, M. **Pattern-Oriented Software Architecture, A System of Patterns**. Wiley, 2013. (Pattern-Oriented Software Architecture). ISBN 9781118725269. Disponível em: [https://books.google.com.br/books?id=j\\\_ahu\\\_BS3hAC](https://books.google.com.br/books?id=j\_ahu\_BS3hAC). Acesso em: 2023-10-23.
- CASTRO, C.; MANZANO, S. Variable and value ordering when solving balanced academic curriculum problems. **Departamento de Informática, Universidad Técnica Federico Santa Maria**, p. 12, 2001. Disponível em: <https://doi.org/10.48550/arXiv.cs/0110007>. Acesso em: 2023-10-23.
- CUNHA, J. a. P. A.; SANTOS, L. G. D.; TAVARES, T. M. A. u.; QUERINO, J. D. J.; UJO, D. C. S. A. D. A.; BARROS, I. M. D. C.; MESQUITA, A. R.; BRITO, G. D. C. Fatores associados á retenção e intencionalidade de evasão nos cursos de farmácia de uma universidade pública do nordeste brasileiro. **Educação em Revista**, Faculdade de Educação da

Universidade Federal de Minas Gerais, v. 39, p. e36898, 2023. ISSN 0102-4698. Disponível em: <<https://doi.org/10.1590/0102-469836898>>. Acesso em: 2023-10-23.

DOCS, M. W. **Uma visão geral do HTTP**. 2023. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>>. Acesso em: 2023-10-23.

FOUNDATION, D. S. **Django documentation**. [S.l.], 2023. Disponível em: <<https://buildmedia.readthedocs.org/media/pdf/django/4.2.x/django.pdf>>. Acesso em: 2023-10-23.

GARCIA, L. e. M.; LARA, D. F.; ANTUNES, F. Análise da retenção no ensino superior: Um estudo de caso em um curso de sistemas de informação. **Revista da Faculdade de Educação**, v. 34, p. 151–38, 12 2020. Disponível em: <<https://doi.org/10.30681/21787476.2020.34.1538>>. Acesso em: 2023-10-23.

GOMES, C. M. A. Abordagem profunda e abordagem superficial à aprendizagem: diferentes perspectivas do rendimento escolar. **Psicologia: Reflexão e Crítica**, Curso de Pós-Graduação em Psicologia da Universidade Federal do Rio Grande do Sul, v. 24, n. 3, p. 438–447, 2011. ISSN 0102-7972. Disponível em: <<https://doi.org/10.1590/S0102-79722011000300004>>. Acesso em: 2023-10-23.

GOURLEY, D.; TOTTY, B.; SAYER, M.; AGGARWAL, A.; REDDY, S. **HTTP: The Definitive Guide: The Definitive Guide**. O'Reilly Media, 2002. (Definitive Guides). ISBN 9781449379582. Disponível em: <<https://books.google.com.br/books?id=qEoOl9bcV\cC>>. Acesso em: 2023-10-23.

HASSENZAHN, M.; DIEFENBACH, S.; GÖRITZ, A. Needs, affect, and interactive products-facets of user experience. **Interacting with Computers**, v. 22, p. 353–362, 01 2011. Disponível em: <<https://doi.org/10.1016/j.intcom.2010.04.002>>. Acesso em: 2023-10-23.

HERNESNIEMI, E.; RATY, H.; KASANEN, K.; XUEJIAO, C.; HONG, J.; KUITTINEN, M. Perception of workload and its relation to perceived teaching and learning environments among finnish and chinese university students. **International Journal of Higher Education**, v. 6, p. 42, 09 2017. Acesso em: 2023-10-23.

HOLOVATY, A. **The definitive guide to Django**: Web development done right. Berkeley, Calif: Apress, 2008. ISBN 9781430203315. Disponível em: <<https://link.springer.com/book/10.1007/978-1-4302-0331-5>>. Acesso em: 2023-10-23.

INEP. **Relatório do 4o Ciclo de Monitoramento das Metas do Plano Nacional de Educação – 2022**. Inep/MEC – Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira, 2022. 1–576 p. Disponível em: <[https://download.inep.gov.br/publicacoes/institucionais/plano\\_nacional\\_de\\_educacao/relatorio\\_do\\_quarto\\_ciclo\\_de\\_monitoramento\\_das\\_metas\\_do\\_plano\\_nacional\\_de\\_educacao.pdf](https://download.inep.gov.br/publicacoes/institucionais/plano_nacional_de_educacao/relatorio_do_quarto_ciclo_de_monitoramento_das_metas_do_plano_nacional_de_educacao.pdf)>. Acesso em: 2023-10-23.

INEP, I. N. d. E. e. P. E. A. T. Painel de monitoramento do pne. 2023. Disponível em: <<https://www.gov.br/inep/pt-br/aceso-a-informacao/dados-abertos/inep-data/painel-de-monitoramento-do-pne>>. Acesso em: 2023-10-23.

KONAK, A.; COIT, D. W.; SMITH, A. E. Multi-objective optimization using genetic algorithms: A tutorial. **Reliability Engineering and System Safety**, v. 91, n. 9, p. 992–1007, 2006. Disponível em: <<https://ideas.repec.org/a/eee/reensy/v91y2006i9p992-1007.html>>. Acesso em: 2023-10-23.

LAUGWITZ, B.; HELD, T.; SCHREPP, M. Construction and evaluation of a user experience questionnaire. In: . [s.n.], 2008. v. 5298, p. 63–76. ISBN 978-3-540-89349-3. Disponível em: <[https://doi.org/10.1007/978-3-540-89350-9\\_6](https://doi.org/10.1007/978-3-540-89350-9_6)>. Acesso em: 2023-10-23.

LIZZIO, A.; WILSON, K.; SIMONS, R. University students' perceptions of the learning environment and academic outcomes: Implications for theory and practice. **Studies in Higher Education**, v. 27, p. 27–52, 02 2002. Disponível em: <<https://doi.org/10.1080/03075070120099359>>. Acesso em: 2023-10-23.

LOPES, C. B. de S. Abordagem multiobjetivo para o problema de balanceamento de currículo acadêmico. Diamantina, Julho 2022. Disponível em: <[http://acervo.ufvjm.edu.br/jspui/bitstream/1/2993/1/clayton\\_bras\\_souza\\_lopes.pdf](http://acervo.ufvjm.edu.br/jspui/bitstream/1/2993/1/clayton_bras_souza_lopes.pdf)>. Acesso em: 2023-10-23.

MOORE, D.; BUDD, R.; WRIGHT, W. **Professional Python Frameworks: Web 2.0 Programming with Django and Turbogears**. Wiley, 2008. ISBN 9780470245347. Disponível em: <<https://books.google.com.br/books?id=za4iT3GVf6oC>>. Acesso em: 2023-10-23.

NORIEGA, A. R.; MARTÍNEZ-RAMÍREZ, Y.; ENEZ, S. J.; SOTO, J.; EREZ, J. F. P. inde: A software to generate an mvc architecture based on the er model. **Computer Applications in Engineering Education**, 09 2021. Disponível em: <<https://doi.org/10.1002/cae.22455>>. Acesso em: 2023-10-23.

OECD. **Curriculum Overload**. [s.n.], 2020. 98 p. Disponível em: <<https://doi.org/10.1787/3081ceca-en>>. Acesso em: 2023-10-23.

OECD. **Education at a Glance 2022**. [s.n.], 2022. 462 p. Disponível em: <<https://doi.org/10.1787/3197152b-en>>. Acesso em: 2023-10-23.

OLUWATOSIN, S. Client-server model haroon. In: . [s.n.], 2014. Disponível em: <<https://api.semanticscholar.org/CorpusID:41223947>>. Acesso em: 2023-10-23.

PEREIRA, A. S.; CARNEIRO, T. C. J.; BRASIL, G. H.; CORASSA, M. A. d. C. Fatores relevantes no processo de permanência prolongada de discentes nos cursos de graduação presencial: um estudo na universidade federal do espírito santo. **Ensaio: Avaliação e Políticas Públicas em Educação**, Fundação CESGRANRIO, v. 23, n. 89, p. 1015–1039, Oct 2015. ISSN 0104-4036. Disponível em: <<https://doi.org/10.1590/S0104-40362015000400009>>. Acesso em: 2023-10-23.

RACHAPUDI, V.; NITISH, N.; SAMAIKYA, S.; SATHVIK, U. P.; DEVI, S. A. Performance comparison of applications with and without web frameworks. **International Journal of Advanced Trends in Computer Science and Engineering**, v. 9, p. 1020–1028, 04 2020. Disponível em: <<https://doi.org/10.30534/ijatcse/2020/19922020>>. Acesso em: 2023-10-23.

RODRIGUES, R. dos S. Desenvolvimento de sistema web para gestão de currículo acadêmico. Diamantina, 2022. Disponível em: <[https://facet.ufvjm.edu.br/wp-content/uploads/decom-tcc/2021-2/TCC\\_Ramon\\_dos\\_Santos\\_Rodrigues.pdf](https://facet.ufvjm.edu.br/wp-content/uploads/decom-tcc/2021-2/TCC_Ramon_dos_Santos_Rodrigues.pdf)>. Acesso em: 2023-10-23.

ROSAS, L.; FLORES, J. e. L.; ZANELLA, V. Evolutionary strategies for the academic curriculum balanced problem. In: . [s.n.], 2011. Disponível em: <<https://doi.org/10.1787/3081ceca-en10.5220/0003723805340538>>. Acesso em: 2023-10-23.

SEMESP, I. Mapa do ensino superior no brasil. In: . [s.n.], 2023. Disponível em: <<https://www.semesp.org.br/mapa/edicao-13/>>. Acesso em: 2023-10-23.

SHAW, B.; BADHWAR, S.; BIRD, A.; S, B.; GUEST, C. **Web Development with Django: Learn to build modern web applications with a Python-based framework**. Packt Publishing, 2021. ISBN 9781839213779. Disponível em: <<https://books.google.com.br/books?id=qksgEAAQBAJ>>. Acesso em: 2023-10-23.

TANEJA, S.; GUPTA, P. R. Python as a tool for web server application development. In: . [s.n.], 2014. Disponível em: <<https://api.semanticscholar.org/CorpusID:63219796>>. Acesso em: 2023-10-23.

TARTARUGA, I. G. P. e. As inovações nos territórios e o papel das universidades: Notas preliminares para o desenvolvimento territorial no estado do rio grande do sul -brasil. In: . [s.n.], 2010. Disponível em: <[https://www.researchgate.net/publication/270580102\\_As\\_inovacoes\\_nos\\_territorios\\_e\\_o\\_papel\\_das\\_universidades\\_notas\\_preliminares\\_para\\_o\\_desenvolvimento\\_territorial\\_no\\_Estado\\_do\\_Rio\\_Grande\\_do\\_Sul](https://www.researchgate.net/publication/270580102_As_inovacoes_nos_territorios_e_o_papel_das_universidades_notas_preliminares_para_o_desenvolvimento_territorial_no_Estado_do_Rio_Grande_do_Sul)>. Acesso em: 2023-10-23.

VIANA, G.; LIMA, J. F. d. Capital humano e crescimento econômico. **Interações (Campo Grande)**, Universidade Católica Dom Bosco, v. 11, n. 2, p. 137–148, Jul 2010. ISSN 1518-7012. Disponível em: <<https://doi.org/10.1590/S1518-70122010000200003>>. Acesso em: 2023-10-23.

VINCENT, W. S. **Django for Beginners**: Build websites with python & django. Estados Unidos: WelcomeToCode, 2018. 315 p. ISBN 9781735467207. Acesso em: 2023-10-23.

ÜNAL, Y. Z.; UYSAL Özgür. A new mixed integer programming model for curriculum balancing: Application to a turkish university. **European Journal of Operational Research**, v. 238, n. 1, p. 339–347, 2014. ISSN 0377-2217. Disponível em: <<https://doi.org/10.1016/j.ejor.2014.03.015>>. Acesso em: 2023-10-23.

## APÊNDICE A – CÓDIGO-FONTE

### Template select-not-taken.html

```
1  {% extends 'aluno/base/base.html' %}
2  {% block pageTitle %} Administração de disciplinas cursadas {% endblock %}
3  {% block content %}
4      {% load static %}
5      <input type="hidden" id="gradeId" name="gradeId" value="{{ grade }}" />
6      <div class="row wrapper page-heading">
7          <div class="col-lg-12">
8              <h2>Administração de disciplinas não cursadas</h2>
9              <div class="pull-right">
10                 <a class="btn btn-sm btn-primary pull-right" title="Voltar" href="{% url
11                     ↪ 'aluno_gradedisciplina_select_taken' grade %}">
12                     <i class="fa fa-arrow-left"></i> Voltar </a>
13             </div>
14         </div>
15     </div>
16     <div class="row">
17         <div class="col-md-12">
18             <form action="{% request.get_full_path %}" method="POST">
19                 <div class="ibox float-e-margins">
20                     {% csrf_token %}
21                     <div class="ibox-title">
22                         <h5>
23                             <strong>Selecionar disciplinas cursadas</strong>
24                         </h5>
25                         <div class="row">
26                             <div class="col-xs-12 col-sm-12 col-md-12">
27                                 <div class="ibox-content">
28                                     <div class="form-group">
29                                         <table class="table">
30                                             <thead>
31                                                 <tr>
32                                                     <th scope="col">Código</th>
33                                                     <th scope="col">Nome</th>
34                                                     <th scope="col">Período Fixo</th>
35                                                 </tr>
36                                             </thead>
37                                             <tbody>
38                                                 <tr>
39                                                     <th scope="row">{{ disciplina.disciplina.codigo
40                                                         ↪ }}</th>
41                                                     <td><span>{{ disciplina.disciplina.nome }} </span>
42                                                     <td style="width: 410px;">
43                                                         <div style="display: inline-flex; gap: 30px;">
44                                                             <div class="form-check">
```



```

44         <input class="form-check-input"
45             ↪ type="checkbox"
46             data-subjectid="{ disciplina.id
47             ↪ }"
48             id="checkbox-{{ disciplina.id }}">
49         <label class="form-check-label"
50             ↪ for="flexCheckDefault">
51         </label>
52     </div>
53     <select class="form-control"
54         data-subjectid="{ disciplina.id }"
55         style="width: 360px;"
56         id="select-{{ disciplina.id }}"
57         ↪ name="curriculo_id"
58         onchange="setarSemestre(this)">
59         <option selected hidden
60             ↪ value="0">Selecione o semestre fixo
61         </option>
62         {% for semestre in semestresDisponiveis %}
63             <option value="{ semestre }">{{
64                 ↪ semestre }}</option>
65         {% endfor %}
66     </select>
67
68     </div>
69     <div class="alert alert-warning alert-dismissible"
70         role="alert" id="alert-{{ disciplina.id }}"
71         ↪ style="margin: 7px 0; display: none">
72     <button type="button" class="close"
73         ↪ data-dismiss="alert"
74         aria-label="Close"><span
75             aria-hidden="true">&times;</span>
76     </button>
77 </div>
78 </td>
79 </tr>
80 {% endfor %}
81 </tbody>
82 </table>
83 </div>
84 </div>
85 </div>
86 </div>
87 <button type="submit" style="margin-bottom: 40px" href="" class="btn btn-sm btn-primary
88     ↪ pull-left">
89     <i class="fa fa-check"></i> Salvar
90 </button>

```

```

84         </form>
85     </div>
86 </div>
87 <script src="{% static '/js/select-not-taken.js' %}"></script>
88 {% endblock %}

```

## select-taken.html

```

1  {% extends 'aluno/base/base.html' %}
2  {% block pageTitle %} Administração de disciplinas cursadas {% endblock %}
3  {% block content %}
4      {% load static %}
5      <input type="hidden" id="disciplinasJaCursadas" value="{% disciplinasJaCursadas %}">
6      <div class="row wrapper page-heading">
7          <div class="col-lg-12">
8              <h2>Administração de disciplinas cursadas</h2>
9              <div class="pull-right">
10                 <a class="btn btn-sm btn-primary pull-right" title="Voltar" href="{% url
11                     ↳ 'aluno_grade_view' grade %}">
12                     <i class="fa fa-arrow-left"></i> Voltar </a>
13                 </div>
14             </div>
15         <div class="row">
16             <div class="col-md-12">
17                 <form action="{% request.get_full_path %}" method="POST">
18                     <div class="ibox float-e-margins">
19                         {% csrf_token %}
20                         <div class="ibox-title">
21                             <h5>
22                                 <strong>Selecionar disciplinas cursadas</strong>
23                             </h5>
24                             <div class="row">
25                                 <div class="col-xs-12 col-sm-12 col-md-12">
26                                     <div class="ibox-content">
27                                         <div class="form-group">
28                                             <div> {% regroup disciplinas|dictsort:'periodo' by periodo as
29                                             ↳ listaSemestres %} {% csrf_token %} {% for semestre in
30                                             ↳ listaSemestres %} <p>
31                                                 <h3 data-toggle="collapse" href="#semestergroup-{{
32                                             ↳ semestre.grouper }}" role="button" aria-expanded="{% if
33                                             ↳ semestre.grouper == 1 %}true{% else %}false{% endif %}"
34                                             ↳ aria-controls="semestergroup" style="font-size: medium;
35                                             ↳ font-weight: normal">
36                                                 {{ semestre.grouper }}º período <svg
37                                             ↳ xmlns="http://www.w3.org/2000/svg" width="16"
38                                             ↳ height="16" fill="currentColor" class="bi
39                                             ↳ bi-chevron-down" viewBox="0 0 16 16">

```

```

31         <path fill-rule="evenodd" d="M1.646 4.646a.5.5 0 0 1 .708
        ↪ 0L8 10.29315.646-5.647a.5.5 0 0 1 .708.708l-6 6a.5.5 0
        ↪ 0 1-.708 0l-6-6a.5.5 0 0 1 0-.708z" />
32     </svg>
33 </h3>
34 </p>
35 <div class="collapse{% if semestre.grouper == 1 %} in{% endif
    ↪ %}" id="semestergroup-{{ semestre.grouper }}">
36     <div class="card card-body">
37         <div class="form-check">
38             <br>
39             <input type="checkbox" name="checkAll"
                ↪ class="form-check-input"
                ↪ id="flexCheckDefaultSelectAll{{
                ↪ semestre.grouper }}"
                ↪ data-semestergroup="semestergroup-{{
                ↪ semestre.grouper }}" onclick="
                ↪ seleccionarTodasAsDisciplinas(this);
                ↪ checarPreRequisitosSeSeleccionarTodasAsMaterias(this)">
40             <label class="form-check-label" style="font-size:
                ↪ small; font-weight: bold"
                ↪ for="flexCheckDefaultSelectAll{{
                ↪ semestre.grouper }}"> Seleccionar todas as
                ↪ disciplinas </label>
41         </div> {% for disciplina in semestre.list %} <div>
42             <input type="checkbox" id="flexCheckDefault{{
                ↪ semestre.grouper }}{{ disciplina.disciplina.nome
                ↪ }}" data-subjectid="{{ disciplina.disciplina.id
                ↪ }}" name="disciplinasCursadas" value="{{
                ↪ disciplina.id }}"
                ↪ onclick="checaSeExistePreRequisito(this)">
43             <label for="scales" style="font-size: small;
                ↪ font-weight: normal">{{ disciplina.disciplina.nome
                ↪ | title }} - {{ disciplina.disciplina.codigo }}
44         </div>
45         <div id="alert{{disciplina.disciplina.id}}"></div> {%
            ↪ endfor %} <br>
46     </div>
47 </div>
48 {% endfor %}
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 <button type="submit" style="margin-bottom: 40px" href="" class="btn btn-sm btn-primary
    ↪ pull-left">

```

```

57         <i class="fa fa-check"></i> Salvar</button>
58     </form>
59 </div>
60 </div>
61 <script src="{% static '/js/select-taken.js' %}"></script>
62 {% endblock %}

```

## Model GradeDisciplinaCursada

```

1 class GradeDisciplinaCursada(Model):
2     id = AutoField(primary_key=True)
3     gradeDisciplina = ForeignKey('GradeDisciplina', null=True, on_delete=models.PROTECT)
4     grade = ForeignKey('Grade', null=True, on_delete=models.PROTECT)
5     created_at = DateTimeField(auto_now_add=True)
6     updated_at = DateTimeField(auto_now=True)

```

## View aluno\_gradedisciplina\_select\_not\_taken

```

1 def aluno_gradedisciplina_select_not_taken(request, grade_id=None):
2     if request.user.is_authenticated and request.user.groups.filter(name='aluno'):
3         grade = Grade.objects.filter(id=grade_id).first()
4         if request.method == 'POST':
5             disciplinasASeremCursadas = GradeDisciplina.objects.filter(grade_id=grade.gradeOriginal.id) |
6                 ↳ .exclude(id__in=GradeDisciplinaCursada.objects.values('gradeDisciplina_id')) |
7                 ↳ .filter(grade_id=grade_id).exclude(id__in=GradeDisciplinaCursar.objects |
8                 ↳ .values('gradeDisciplina_id')).filter(grade_id=grade_id))
9
10            for disciplina in disciplinasASeremCursadas:
11                gradeDisciplinaIdTest = GradeDisciplina.objects.filter(id=disciplina.id).get()
12                gradeId = Grade.objects.filter(id=grade_id).get()
13                gradeDisciplinaCursar = GradeDisciplinaCursar(
14                    grade = gradeId,
15                    gradeDisciplina = gradeDisciplinaIdTest,
16                    periodoFixo = False,
17                    periodo = None,
18                )
19                try:
20                    gradeDisciplinaCursar.save()
21                except Exception as e:
22                    return print(e)
23            return redirect('aluno_grade_view', grade.id)
24
25        if request.method == 'GET':
26            GradeDisciplinaCursar.objects.filter(grade_id=grade_id).delete()
27
28            grade = Grade.objects.filter(id=grade_id).first()
29            inicio = Grade.objects.filter(id=grade_id).values_list('periodoCronologico', flat=True).first()

```

```

27     fim = Grade.objects.filter(id=grade_id).values_list('periodosRestantes', flat=True).first()
28     semestresDisponiveis = list(range(inicio, inicio + fim))
29     disciplinas = GradeDisciplina.objects.filter(grade_id=grade.gradeOriginal.id)\
30         .exclude(id__in=GradeDisciplinaCursada.objects.values('gradeDisciplina_id'))\
31         ↪ .filter(grade_id=grade_id)
32
33     template = loader.get_template('aluno/gradedisciplina/select-not-taken.html')
34     context = {'disciplinas': disciplinas, 'grade': grade_id,
35         ↪ 'semestresDisponiveis': semestresDisponiveis}
36     return HttpResponse(template.render(context, request))
37 elif request.user.is_authenticated and request.user.groups.filter(name='gestor'):
38     return render(request, 'admin/home/index.html', {})
39 else:
40     return redirect('login')

```

## aluno\_gradedisciplina\_select\_taken

```

1 def aluno_gradedisciplina_select_taken(request, grade_id=None):
2     if request.user.is_authenticated and request.user.groups.filter(name='aluno'):
3
4         if request.method == 'GET':
5
6             disciplinasASeremCursadas = GradeDisciplina.objects.select_related('disciplina').filter(
7                 grade_id=Subquery(Grade.objects.values('gradeOriginal_id').filter(id=grade_id)))
8
9             disciplinasJaAdicionadas =
10                 ↪ list(GradeDisciplinaCursada.objects.filter(grade_id=grade_id).values_list(
11                     'disciplina_id', flat=True))
12
13             context = {'grade': grade_id, 'disciplinas': disciplinasASeremCursadas,
14                 ↪ 'disciplinasJaCursadas': disciplinasJaAdicionadas}
15             template = loader.get_template('aluno/gradedisciplina/select-taken.html')
16             return HttpResponse(template.render(context, request))
17
18         if request.method == 'POST':
19
20             disciplinasCursadasRequest = list(map(int, request.POST.getlist('disciplinasCursadas')))
21
22             GradeDisciplinaCursada.objects.filter(grade_id=grade_id).exclude(
23                 gradeDisciplina_id__in=disciplinasCursadasRequest).delete()
24
25             gradeOriginal = Grade.objects.values_list('gradeOriginal_id', flat=True).filter(
26                 id=grade_id).first()
27
28             for disciplina in disciplinasCursadasRequest:
29                 if not GradeDisciplinaCursada.objects.filter(gradeDisciplina_id=disciplina,
30                     ↪ grade_id=grade_id).exists():
31                     disciplinaCadastrar = GradeDisciplinaCursada.objects.create(

```

```

29         gradeDisciplina = GradeDisciplina.objects.filter(id=disciplina,
30             ↪ grade_id=gradeOriginal).first(),
31         grade = Grade.objects.filter(id=grade_id).first(),
32         disciplina = Disciplina.objects.filter(id__in=Subquery(GradeDisciplina.objects
33             ↪ .filter(id=disciplina,
34             ↪ grade_id=gradeOriginal).values('disciplina_id'))).first()
35     )
36     try:
37         disciplinaCadastrar.save()
38     except Exception as e:
39         return print(e)
40
41     return redirect('aluno_gradedisciplina_select_not_taken', grade_id)
42
43 elif request.user.is_authenticated and request.user.groups.filter(name='gestor'):
44     return render(request, 'admin/home/index.html', {})
45 else:
46     return redirect('login')

```

## aluno\_gradedisciplina\_checkrequisites

```

1 def aluno_gradedisciplina_checkrequisites(request):
2     if request.method == 'POST':
3         idSubject = json.loads(request.body)
4         prerequisitos =
5             ↪ list(Requisito.objects.filter(gradeDisciplina__disciplina__id=idSubject).values_list(
6                 'gradeDisciplinaRequisito__disciplina__id', flat=True))
7         libera =
8             ↪ list(Requisito.objects.filter(gradeDisciplinaRequisito__disciplina__id=idSubject).values_list(
9                 'gradeDisciplinaRequisito__disciplina__nome', flat=True))
10        prerequisitosnome =
11            ↪ list(Requisito.objects.filter(gradeDisciplina__disciplina__id=idSubject).values_list(
12                'gradeDisciplinaRequisito__disciplina__nome', flat=True))
13
14        return JsonResponse({'prerequisitos': prerequisitos, 'libera': libera, 'prerequisitosnome':
15            ↪ prerequisitosnome})
16    return JsonResponse({})

```

## aluno\_gradedisciplina\_checksemester

```

1 def aluno_gradedisciplina_checksemester(request):
2     payload = json.loads(request.body)
3     gradeId = payload['gradeId']
4     gradeDisciplinaId = payload['gradeDisciplinaId']
5     numeroPeriodo = payload['numeroPeriodo']
6     boolPeriodo = payload['boolPeriodo']

```

```

7
8     grade = Grade.objects.filter(id=gradeId).first()
9     GradeDisciplinaCursar.objects.filter(gradeDisciplina_id=gradeDisciplinaId).delete()
10
11     if request.method == 'POST':
12         gradesDisciplinasCursar = GradeDisciplinaCursar.objects.filter(grade_id=gradeId)
13         gradeDisciplinaCursar = GradeDisciplinaCursar.objects.filter(grade_id=gradeId,
14             ↳ gradeDisciplina_id=gradeDisciplinaId).first()
15
16     if gradeDisciplinaCursar != None:
17         response = {
18             'message': 'Disciplina Já está cadastrada',
19             'status': 422
20         }
21         return JsonResponse(response, status=422)
22
23     gradeDisciplinaOriginal = GradeDisciplina.objects.filter(
24         id=gradeDisciplinaId).first()
25     periodoFixo = False
26     periodo = None
27
28     if numeroPeriodo == 0 and boolPeriodo == 's':
29         response = {
30             'message': 'Não foi selecionado nenhum semestre',
31             'status': 422
32         }
33         return JsonResponse(response, status=422)
34
35     if boolPeriodo == 'S' or boolPeriodo == 's':
36
37         if bool(numeroPeriodo) == False:
38             response = {
39                 'message': 'Operação não realizada! Para disciplinas de período fixo, é necessário
40                 ↳ informar a qual período a disciplina deverá pertencer!',
41                 'status': 422
42             }
43             return JsonResponse(response, status=422)
44
45     periodoCadastro = numeroPeriodo
46     periodo = (
47         int(periodoCadastro) - grade.periodoCronologico) + 1
48
49     if periodo < 1:
50         response = {
51             'message': 'Operação não realizada! O período informado já foi concluído!',
52             'status': 422
53         }
54         return JsonResponse(response, status=422)
55
56     if periodo > grade.periodosRestantes:

```

```

54     response = {
55         'message': 'Operação não realizada! O período informado é superior à quantidade de
        ↳ períodos disponíveis para a conclusão do curso!',
56         'status': 422
57     }
58     return JsonResponse(response, status=422)
59
60     requisitos = Requisito.objects.filter(gradeDisciplina_id=gradeDisciplinaId)
61     if requisitos.exists() and periodo == 1:
62         response = {
63             'message': 'Operação não realizada! Esta disciplina possui pré-requisito e, portanto, não
        ↳ pode ser cadastrada neste período!',
64             'status': 422
65         }
66         return JsonResponse(response, status=422)
67
68     for requisito in requisitos:
69         for gradeDisciplinaCursarVerificar in gradesDisciplinasCursar:
70             if gradeDisciplinaCursarVerificar.gradeDisciplina.id ==
        ↳ requisito.gradeDisciplinaRequisito.id:
71                 if gradeDisciplinaCursarVerificar.periodoFixo:
72                     if periodo <= gradeDisciplinaCursarVerificar.periodo:
73                         response = {
74                             'message': 'Operação não realizada! Tentativa de fixação da disciplina ' +
        ↳ gradeDisciplinaOriginal.disciplina.nome + ' no período ' + str(
75                             periodoCadastro) + ', que é anterior ou igual ao período fixado para a
        ↳ disciplina ' +
        ↳ gradeDisciplinaCursarVerificar.gradeDisciplina.disciplina.nome +
        ↳ ', seu pré-requisito.',
76                             'status': 422
77                         }
78                         return JsonResponse(response, status=422)
79
80                     elif (
81                         periodo - gradeDisciplinaCursarVerificar.periodo) >
        ↳ DIFERENCAMAXIMAPERIODOSPONTUACAORELACAOPREREQUISITO:
82                         response = {
83                             'message': 'Operação não realizada! A diferença máxima entre períodos das
        ↳ disciplinas ' + gradeDisciplinaOriginal.disciplina.nome + 'e' +
        ↳ gradeDisciplinaCursarVerificar.gradeDisciplina.disciplina.nome + ',
        ↳ que possuem relação de pré-requisitos, deve ser de, no máximo,' +
        ↳ DIFERENCAMAXIMAPERIODOSPONTUACAORELACAOPREREQUISITO + ' períodos,
        ↳ conforme definição do curso.',
84                             'status': 422
85                         }
86                         return JsonResponse(response, status=422)
87
88     requisitos = Requisito.objects.filter(
89         gradeDisciplinaRequisito_id=gradeDisciplinaId).all()

```



```

90
91     if requisitos.exists() and periodo == grade.periodosRestantes:
92         response = {
93             'message': 'Operação não realizada! Tentativa de fixação de disciplina que é pré-requisito
↪ no último período!',
94             'status': 422
95         }
96         return JsonResponse(response, status=422)
97
98     for requisito in requisitos:
99         for gradeDisciplinaCursarVerificar in gradesDisciplinasCursar:
100             if gradeDisciplinaCursarVerificar.gradeDisciplina.id == requisito.gradeDisciplina.id:
101                 if gradeDisciplinaCursarVerificar.periodoFixo:
102                     if periodo >= gradeDisciplinaCursarVerificar.periodo:
103                         response = {
104                             'message': 'Operação não realizada! Tentativa de fixação da disciplina' +
↪ gradeDisciplinaOriginal.disciplina.nome + ' em período posterior ou
↪ igual ao período da disciplina' +
↪ gradeDisciplinaCursarVerificar.gradeDisciplina.disciplina.nome + ',
↪ que é sua dependente.',
105                             'status': 422
106                         }
107                         return JsonResponse(response, status=422)
108
109                     elif (
110                         gradeDisciplinaCursarVerificar.periodo - periodo) >
↪ DIFERENCAMAXIMAPERIODOSPONTUACAORELACAOPREREQUISITO:
111                         response = {
112                             'message': 'Operação não realizada! A diferença máxima entre períodos das
↪ disciplinas ' + gradeDisciplinaOriginal.disciplina.nome + 'e' +
↪ gradeDisciplinaCursarVerificar.gradeDisciplina.disciplina.nome + ',
↪ que possuem relação de pré-requisitos, deve ser de, no máximo,' +
↪ DIFERENCAMAXIMAPERIODOSPONTUACAORELACAOPREREQUISITO + ' períodos,
↪ conforme definição do curso.',
113                             'status': 422
114                         }
115                         return JsonResponse(response, status=422)
116
117     relacionamentos = Relacionamento.objects.filter(
118         gradeDisciplinaRelacionamentoPosterior_id=gradeDisciplinaId)
119
120     if relacionamentos.exists() and periodo == 1:
121         response = {
122             'message': 'Operação não realizada! Esta disciplina é dependente de outra e, portanto, não
↪ pode ser cadastrada neste período!.',
123             'status': 422
124         }
125         return JsonResponse(response, status=422)
126

```

```

127     for relacionamento in relacionamentos:
128         for gradeDisciplinaCursarVerificar in gradesDisciplinasCursar:
129             if gradeDisciplinaCursarVerificar.gradeDisciplina.id ==
130                 ↳ relacionamento.gradeDisciplinaRelacionamentoAnterior.id:
131                 if gradeDisciplinaCursarVerificar.periodoFixo and periodo <
132                     ↳ gradeDisciplinaCursarVerificar.periodo and relacionamento.relacionamento >=
133                     ↳ PONTUACAOINICIALRELACAO:
134                         response = {
135                             'message': 'Operação não realizada! A disciplina' +
136                                 ↳ gradeDisciplinaOriginal.disciplina.nome + ' não pode estar localizada em
137                                 ↳ período anterior à da disciplina ' +
138                                 ↳ gradeDisciplinaCursarVerificar.gradeDisciplina.disciplina.nome + ', pois é
139                                 ↳ inter-relacionada com esta!',
140                             'status': 422
141                         }
142                     return JsonResponse(response, status=422)
143
144     relacionamentos = Relacionamento.objects.filter(
145         gradeDisciplinaRelacionamentoAnterior_id=gradeDisciplinaId).all()
146
147     if relacionamentos.exists() and periodo == grade.periodosRestantes:
148         response = {
149             'message': ' Operação não realizada! Tentativa de fixação no último período de disciplina
150             ↳ da qual outras dependem!',
151             'status': 422
152         }
153         return JsonResponse(response, status=422)
154
155     for relacionamento in relacionamentos:
156         for gradeDisciplinaCursarVerificar in gradesDisciplinasCursar:
157             if gradeDisciplinaCursarVerificar.gradeDisciplina.id ==
158                 ↳ relacionamento.gradeDisciplinaRelacionamentoPosterior.id and
159                 ↳ relacionamento.relacionamento >= PONTUACAOINICIALRELACAO:
160                 if gradeDisciplinaCursarVerificar.periodoFixo and periodo >
161                     ↳ gradeDisciplinaCursarVerificar.periodo:
162                     response = {
163                         'message': 'Operação não realizada! A disciplina ' +
164                             ↳ gradeDisciplinaCursarVerificar.gradeDisciplina.disciplina.nome + ' é
165                             ↳ inter-relacionada com a disciplina' +
166                             ↳ gradeDisciplinaOriginal.disciplina.nome + ', o que impede esta de estar
167                             ↳ localizada em um período posterior ao da disciplina ' +
168                             ↳ gradeDisciplinaCursarVerificar.gradeDisciplina.disciplina.nome,
169                         'status': 422
170                     }
171                     return JsonResponse(response, status=422)
172
173     periodoFixo = True
174
175     gradeDisciplinaCursar = GradeDisciplinaCursar(

```

```

160         grade=grade,
161         gradeDisciplina=gradeDisciplinaOriginal,
162         periodoFixo=periodoFixo,
163         periodo=periodo,
164     )
165     try:
166         gradeDisciplinaCursar.save()
167     except Exception as e:
168         return print(e)
169     response = {
170         'message': 'Disciplina Adicionada',
171         'status': 200
172     }
173     return JsonResponse(response, status=200)
174

```

## Javascript select-taken.js

```

1  window.addEventListener('load', function() {
2      let disciplinasJaCursadas =
3      ↪ JSON.parse((document.getElementById('disciplinasJaCursadas')).defaultValue)
4      let todasAsCheckbox = document.querySelectorAll('input[name="disciplinasCursadas"]');
5      todasAsCheckbox.forEach(checkbox => {
6          disciplinasJaCursadas.forEach(disciplina => {
7              let disciplinaGradeId = parseInt(checkbox.dataset.subjectid)
8              if (disciplinaGradeId === disciplina) {
9                  checkbox.checked = true;
10             }
11         });
12     });
13
14     async function checaSeExistePreRequisito(disciplina) {
15         let gradeDisciplinaId = disciplina.dataset.subjectid;
16         let CSRFToken = document.cookie.match(/csrftoken=(\[~ ;]+\)/)[1];
17         let response = await fetch("/curricular/aluno/gradedisciplina/checkrequisites", {
18             method: 'post',
19             headers: {
20                 "X-CSRFToken": CSRFToken
21             },
22             body: gradeDisciplinaId
23         }).then((response) => {
24             return response.json();
25         })
26         DesmarcaSeRequisitoNaoEstiverMarcado(gradeDisciplinaId, response, disciplina);
27         DesmarcaDisciplinaLiberadaSeForDesmarcado(gradeDisciplinaId, response, disciplina);
28     }
29

```

```

30 function DesmarcaSeRequisitoNaoEstiverMarcado(gradeDisciplinaId, resposta, disciplina) {
31     let todasAsCheckbox = document.querySelectorAll('input[name="disciplinasCursadas"]');
32     todasAsCheckbox.forEach(checkbox => {
33         resposta.prerequisitos.forEach(prerequisitoId => {
34             let disciplinaGradeId = parseInt(checkbox.dataset.subjectid)
35             if (disciplinaGradeId === prerequisitoId && !checkbox.checked){
36                 disciplina.checked = false;
37                 AvisaOUusuarioSobreRequisito(gradeDisciplinaId, resposta.prerequisitosnome);
38             }
39         });
40     });
41 }
42
43 function DesmarcaDisciplinaLiberadaSeForDesmarcado(idSubject, resposta, subject) {
44     let todasAsCheckbox = document.querySelectorAll('input[name="disciplinasCursadas"]');
45     todasAsCheckbox.forEach(checkbox => {
46         resposta.libera.forEach(disciplinaLiberada => {
47             let disciplinaGradeId = parseInt(checkbox.dataset.subjectid)
48             if (disciplinaGradeId === disciplinaLiberada) {
49                 if (!subject.checked && checkbox.checked) {
50                     checkbox.checked = false;
51                     disciplina = document.querySelector('[data-subjectid="' + disciplinaLiberada + '"');
52                     checaSeExistePreRequisito(disciplina);
53                 }
54             }
55         });
56     });
57 }
58
59 function selecionarTodasAsDisciplinas(checkboxMarcarTodas) {
60     let todasAsCheckbox = document.getElementById(checkboxMarcarTodas.dataset.semestergroup)
61     .querySelectorAll('input[name="disciplinasCursadas"]');
62     todasAsCheckbox.forEach(checkbox => {
63         checkbox.checked = checkboxMarcarTodas.checked;
64     });
65 }
66
67 function AvisaOUusuarioSobreRequisito(gradeDisciplinaId, preRequisitoNome) {
68     if ($('#alert' + gradeDisciplinaId).find('div').length == 0) {
69         let aviso = '<div class="alert alert-warning alert-dismissible" role="alert">' +
70             'Você precisa ter cursado a disciplina ' + preRequisitoNome + ' para cursar essa disciplina.
71             ↳ <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span
72             ↳ aria-hidden="true">&times;</span></button>' +
73             '</div>';
74         $('#alert' + gradeDisciplinaId).html(aviso).delay(3000).fadeOut();
75     } else {
76         $('#alert' + gradeDisciplinaId).show().delay(3000).fadeOut();
77     }
78 }

```

```

77
78 async function checarPreRequisitosSeSelecionarTodasAsMaterias(checkboxMarcarTodas) {
79     if (checkboxMarcarTodas.checked) {
80         let todasAsCheckbox = document.getElementById(checkboxMarcarTodas.dataset.semestergroup)
81             .querySelectorAll('input[name="disciplinasCursadas"]');
82         for (let checkbox of todasAsCheckbox) {
83             await checaSeExistePreRequisito(checkbox);
84         }
85     }
86 }

```

## select-not-taken.js

```

1 document.addEventListener('DOMContentLoaded', () => {
2     let selects = document.querySelectorAll('select');
3     selects.forEach(select => {
4         select.disabled = true;
5     });
6 });
7
8 let todasAsCheckbox = document.querySelectorAll('input[type="checkbox"]');
9 todasAsCheckbox.forEach(checkbox => {
10     checkbox.addEventListener("change", async () => {
11         let id = checkbox.dataset.subjectid;
12         let select = document.querySelector(`#select-${id}`);
13         let selectedIndex = select.selectedIndex;
14
15         if (checkbox.checked) {
16             if (selectedIndex !== 0) {
17                 await setarSemestre(select);
18             } else {
19                 select.disabled = false;
20             }
21         } else {
22             if (selectedIndex !== 0) {
23                 await deletarDoBancoDeDados(select);
24                 select.selectedIndex = 0;
25             }
26             select.disabled = true;
27         }
28     });
29 });
30
31 async function deletarDoBancoDeDados(select) {
32     let url = "/curricular/aluno/gradedisciplina/deletesubject";
33
34     let gradeId = (document.querySelector("#gradeId")).value;
35     let gradeDisciplinaId = select.getAttribute('data-subjectid');

```

```

36     let csrftoken = document.cookie.match(/csrftoken=(\[^\s\;]+\))/[1];
37
38     let payload = {
39         "gradeDisciplina_id": gradeDisciplinaId,
40         "grade_id": gradeId
41     };
42
43     await fetch(url, {
44         method: "POST",
45         headers: {
46             "Content-Type": "application/json",
47             "X-CSRFToken": csrftoken
48         },
49         body: JSON.stringify(payload)
50     })
51 }
52
53 async function setarSemestre(disciplina) {
54
55     let gradeId = document.querySelector("#gradeId").value;
56     let numeroPeriodo = disciplina.value;
57     let gradeDisciplinaId = disciplina.dataset.subjectid;
58
59     let csrftoken = document.cookie.match(/csrftoken=(\[^\s\;]+\))/[1];
60     let checkbox = $('input[id^="checkbox-" + gradeDisciplinaId + "']');
61
62     let payload = {
63         "gradeId": gradeId,
64         "gradeDisciplinaId": gradeDisciplinaId,
65         "numeroPeriodo": numeroPeriodo,
66         "boolPeriodo": "S"
67     };
68
69     let url = "/curricular/aluno/gradedisciplina/checksemester";
70     await fetch(url, {
71         method: "POST",
72         headers: {
73             "Content-Type": "application/json",
74             "X-CSRFToken": csrftoken
75         },
76         body: JSON.stringify(payload)
77     }).then(response => {
78         return response.json();
79     }).then(json => {
80         avisaUsuarioErroDeCadastroDePeriodoFixo(json.message, gradeDisciplinaId);
81         if (json.status !== 200){
82             disciplina.selectedIndex = 0;
83             disciplina.disabled = true;
84             checkbox[0].checked = false;

```

```
85         }
86     });
87 }
88
89 function avisaUsuarioErroDeCadastroDePeriodoFixo(message, subjectId){
90     $('#alert-' + subjectId).css('display','inline-flex').text(message).show().delay(3000).fadeOut();
91 }
92
```

## **AUTORIZAÇÃO**

Autorizo a reprodução e/ou divulgação total ou parcial do presente trabalho, por qualquer meio convencional ou eletrônico, desde que citada a fonte.

Diamantina, \_\_\_\_ / \_\_\_\_ / \_\_\_\_.

---

Natascha Annette Soares Schlegel

[natascha.schlegel@ufvjm.edu.br](mailto:natascha.schlegel@ufvjm.edu.br)

Universidade Federal dos Vales do Jequitinhonha e Mucuri

MGC 367, Km 583 - nº 5000, Diamantina - MG, 39100-000





**UFVJM**